

EMBEDDED VISION

LICENSE, DISCLAIMER OF LIABILITY, AND LIMITED WARRANTY

By purchasing or using this book (the “Work”), you agree that this license grants permission to use the contents contained herein, but does not give you the right of ownership to any of the textual content in the book or ownership to any of the information or products contained in it. *This license does not permit uploading of the Work onto the Internet or on a network (of any kind) without the written consent of the Publisher.* Duplication or dissemination of any text, code, simulations, images, etc. contained herein is limited to and subject to licensing terms for the respective products, and permission must be obtained from the Publisher or the owner of the content, etc., in order to reproduce or network any portion of the textual material (in any media) that is contained in the Work.

MERCURY LEARNING AND INFORMATION (“MLI” or “the Publisher”) and anyone involved in the creation, writing, or production of the companion disc, accompanying algorithms, code, or computer programs (“the software”), and any accompanying Web site or software of the Work, cannot and do not warrant the performance or results that might be obtained by using the contents of the Work. The author, developers, and the Publisher have used their best efforts to insure the accuracy and functionality of the textual material and/or programs contained in this package; we, however, make no warranty of any kind, express or implied, regarding the performance of these contents or programs. The Work is sold “as is” without warranty (except for defective materials used in manufacturing the book or due to faulty workmanship).

The author, developers, and the publisher of any accompanying content, and anyone involved in the composition, production, and manufacturing of this work will not be liable for damages of any kind arising out of the use of (or the inability to use) the algorithms, source code, computer programs, or textual material contained in this publication. This includes, but is not limited to, loss of revenue or profit, or other incidental, physical, or consequential damages arising out of the use of this Work.

The sole remedy in the event of a claim of any kind is expressly limited to replacement of the book, and only at the discretion of the Publisher. The use of “implied warranty” and certain “exclusions” vary from state to state, and might not apply to the purchaser of this product.

**EMBEDDED
VISION**
An Introduction

S. R. Vijayalakshmi, PhD
S. Muruganand, PhD



MERCURY LEARNING AND INFORMATION

Dulles, Virginia
Boston, Massachusetts
New Delhi

Copyright ©2020 by MERCURY LEARNING AND INFORMATION LLC. All rights reserved.

Original title and copyright: *Embedded Vision*. Copyright ©2019 by Overseas Press India Private Limited. All rights reserved.

This publication, portions of it, or any accompanying software may not be reproduced in any way, stored in a retrieval system of any type, or transmitted by any means, media, electronic display or mechanical display, including, but not limited to, photocopy, recording, Internet postings, or scanning, without prior permission in writing from the publisher.

Publisher: David Pallai
MERCURY LEARNING AND INFORMATION
22841 Quicksilver Drive
Dulles, VA 20166
info@merclearning.com
www.merclearning.com
(800) 232-0223

S. R. Vijayalakshmi and S. Muruganand. *Embedded Vision: An Introduction*.
ISBN: 978-1-68392-457-9

The publisher recognizes and respects all marks used by companies, manufacturers, and developers as a means to distinguish their products. All brand names and product names mentioned in this book are trademarks or service marks of their respective companies. Any omission or misuse (of any kind) of service marks or trademarks, etc. is not an attempt to infringe on the property of others.

Library of Congress Control Number: 2019937247
192021321 This book is printed on acid-free paper in the United States of America.

Our titles are available for adoption, license, or bulk purchase by institutions, corporations, etc. For additional information, please contact the Customer Service Dept. at (800) 232-0223 (toll free).

All of our titles are available in digital format at academiccourseware.com and other digital vendors. Companion disc files for this title are available by contacting info@merclearning.com. The sole obligation of MERCURY LEARNING AND INFORMATION to the purchaser is to replace the disc, based on defective materials or faulty workmanship, but not based on the operation or functionality of the product.

Contents

<i>Preface</i>	<i>xvii</i>
Chapter 1 Embedded Vision	1
1.1 Introduction to Embedded Vision	1
1.2 Design of an Embedded Vision System	5
Characteristics of Embedded Vision System Boards Versus Standard Vision System Boards	6
Benefits of Embedded Vision System Boards	7
Processors for Embedded Vision	7
High Performance Embedded CPU	9
Application Specific Standard Product (ASSP) in Combination with a CPU	10
General Purpose Cpus	10
Graphics Processing Units with CPU	10
Digital Signal Processors with Accelerator(s) and a CPU	11
Field Programmable Gate Arrays (FPGAs) with a CPU	12
Mobile “Application Processor”	13
Cameras/Image Sensors for Embedded Vision	14
Other Semiconductor Devices for Embedded Vision	15
Memory	15
Networking and Bus Interfaces	16
1.3 Components in a Typical Vision System	16
Vision Processing Algorithms	17
Embedded Vision Challenges	18
1.4 Applications for Embedded Vision	19
Swimming Pool Safety System	20
Object Detection	20
Video Surveillance	21
Gesture Recognition	21
Simultaneous Localization and Mapping (SLAM)	21
Automatic Driver Assistance System (ADAS)	21
Game Controller	22
Face Recognition for Advertising Research	23
Mobile Phone Skin Cancer Detection	23
Gesture Recognition for Car Safety	23
Industrial Applications for Embedded Vision	23

Medical Applications for Embedded Vision	25
Automotive Applications for Embedded Vision	26
Security Applications for Embedded Vision	26
Consumer Applications for Embedded Vision	27
Machine Learning in Embedded Vision Applications	28
1.5 Industrial Automation and Embedded Vision: A Powerful Combination	28
Inventory Tracking	29
Automated Assembly	30
Automated Inspection	31
Workplace Safety	32
Depth Sensing	33
1.6 Development Tools for Embedded Vision	34
Both General Purpose and Vendor Specific Tools	35
Personal Computers	35
OpenCV	36
Heterogeneous Software Development in an Integrated Development Environment	36
<i>Summary</i>	36
<i>Reference</i>	37
<i>Learning Outcomes</i>	37
<i>Further Reading</i>	37
Chapter 2 Industrial Vision	39
2.1 Introduction to Industrial Vision Systems	40
PC-Based Vision Systems	46
Industrial Cameras	46
High-Speed Industrial Cameras	47
Smart Cameras	48
2.2 Classification of Industrial Vision Applications	48
Dimensional Quality	49
Surface Quality	53
Structural Quality	55
Operational Quality	56
2.3 3D Industrial Vision	56
Automated Inspection	57
Robotic Guidance	58
3D Imaging	59
3D Imaging Methods	59
3D Inspection	61

3D Processing	63
3D Robot Vision	63
High-Speed Imaging	64
High-Speed Cameras	65
Line Scan Imaging	66
Capture and Storage	67
High-Speed Inspection for Product Defects	68
Labels and Marking	68
Web Inspection	69
High-Speed Troubleshooting	69
Line Scan Technology	70
Contact Image Sensors	72
Lenses	73
Image Processing	74
Line Scan Inspection	74
Tracking and Traceability	76
Serialization	77
Direct Part Marking	77
Product Conformity	78
Systems Integration Challenges	80
2.4 Industrial Vision Measurement	81
Character Recognition, Code Reading, and Verification	83
Making Measurements	84
Pattern Matching	85
3D Pattern Matching	86
Preparing for Measurement	86
Industrial Control	87
Development Approaches and Environments	88
Development Software Tools for Industrial Vision Systems	89
Image Processing and Analysis Tools	90
<i>Summary</i>	91
<i>References</i>	92
<i>Learning Outcomes</i>	92
<i>Further Reading</i>	92
Chapter 3 Medical Vision	93
3.1 Introduction to Medical Vision	94
Advantages of Digital Processing for Medical Applications	95
Digital Image Processing Requirements for Medical Applications	96
Advanced Digital Image Processing Techniques in Medical Vision	96
Image Processing Systems for Medical Applications	97
Stereoscopic Endoscope	98

3.2 From Images to Information in Medical Vision	109
Magnifying Minute Variations	116
Gesture and Security Enhancements	116
3.3 Mathematics, Algorithms in Medical Imaging	117
Artificial Intelligence (AI)	117
Computer-Aided Diagnostic Processing	120
Vision Algorithms for Biomedical	123
Real-Time Radiography	123
Image Compression Technique for Telemedicine	127
Region of Interest	128
Structure Sensitive Adaptive Contrast Enhancement Methods	129
LSPHIT Algorithm for ECG Data Compression and Transmission	130
Retrieval of Medical Images in a PACs	130
Digital Signature Realization Process of DICOM Medical Images	131
Computer Neural Networks (CNNs) in Medical Image Analysis	132
Deep Learning and Big Data	133
3.4 Machine Learning in Medical Image Analysis	135
Convolutional Neural Networks	135
Convolution Layer	136
Rectified Linear Unit (RELU) Layer	137
Pooling Layer	137
Fully Connected Layer	137
Feature Computation	143
Feature Selection	143
Training and Testing: The Learning Process	143
Example of Machine Learning with Use of Cross Validation	144
<i>Summary</i>	145
<i>References</i>	146
<i>Learning Outcomes</i>	146
<i>Further Reading</i>	147
Chapter 4 Video Analytics	149
4.1 Definition of Video Analytics	149
Applications of Video Analytics	151
Image Analysis Software	153
Security Center Integration	156
Video Analytics for Perimeter Detection	157
Video Analytics for People Counting	157
Traffic Monitoring	158
Auto Tracking Cameras for Facial Recognition	158
Left Object Detection	158

4.2 Video Analytics Algorithms	160
Algorithm Example: Lens Distortion Correction	161
Dense Optical Flow Algorithm	162
Camera Performance Affecting Video Analytics	163
Video Imaging Techniques	167
4.3 Machine Learning in Embedded Vision Applications	168
Types of Machine-Learning Algorithms	170
Implementing Embedded Vision and Machine Learning	177
Embedded Computers Make Inroads to Vision Applications	179
4.4 Examples for Machine Learning	180
1. Convolutional Neural Networks for Autonomous Cars	180
2. CNN Technology Enablers	186
3. Smart Fashion AI Architecture	188
4. Teaching Computers to Recognize Cats	190
<i>Summary</i>	194
<i>References</i>	194
<i>Learning Outcomes</i>	195
<i>Further Reading</i>	195
Chapter 5 Digital Image Processing	197
5.1 Image Processing Concepts for Vision Systems	198
Image	198
Signal	199
Systems	199
5.2 Image Manipulations	203
Image Sharpening and Restoration	203
Histograms	203
Transformation	205
Edge Detection	213
Vertical Direction	214
Horizontal Direction	214
Sobel Operator	215
Robinson Compass Mask	217
Kirsch Compass Mask	219
Laplacian Operator	221
Positive Laplacian Operator	222
Negative Laplacian Operator	222
5.3 Analyzing an Image	223
Color Spaces	232
JPEG Compression	234

Pattern Matching	236
Template Matching	239
Template Matching Approaches	240
Motion Tracking and Occlusion Handling	240
Template-Matching Techniques	241
Advanced Methods	243
Advantage	247
Enhancing The Accuracy of Template Matching	247
5.4 Image-Processing Steps for Vision System	247
Scanning and Image Digitalization	247
Image Preprocessing	248
Image Segmentation On Object	248
Description of Objects	249
Classification of Objects	249
<i>Summary</i>	249
<i>References</i>	250
<i>Learning Outcomes</i>	250
<i>Further Reading</i>	251
Chapter 6 Camera—Image Sensor	253
6.1 History of Photography	254
Image Formation on Cameras	256
Image Formation on Analog Cameras	257
Image Formation on Digital Cameras	257
Camera Types and Their Advantages: Analog Versus Digital Cameras	259
Interlaced Versus Progressive Scan Cameras	260
Area Scan Versus Line Scan Cameras	261
Time Delay and Integration (TDI) Versus Traditional Line Scan Cameras	262
Camera Mechanism	263
Perspective Transformation	264
Pixel	265
6.2 Camera Sensor for Embedded Vision Applications	271
Charge Coupled Device (CCD) Sensor Construction	272
Complementary Metal Oxide Semiconductor (CMOS)	
Sensor Construction	273
Sensor Features	276
Electronic Shutter	279
Sensor Taps	281
Spectral Properties of Monochrome and Color Cameras	281
Camera Resolution for Improved Imaging System Performance	288

6.3	Zooming, Camera Interface, and Selection	291
	Optical Zoom	292
	Digital Zoom	292
	Spatial Resolution	296
	Gray-Level Resolution	298
	Capture Boards	302
	Firewire Ieee 1394/IIDC DCAM Standard	302
	Camera Link	302
	GigE Vision Standard	303
	USB—Universal Serial Bus	303
	CoaXPress	303
	Camera Software	304
	Camera and Lens Selection for a Vision Project	305
6.4	Thermal-Imaging Camera	307
	<i>Summary</i>	309
	<i>References</i>	310
	<i>Learning Outcomes</i>	310
	<i>Further Reading</i>	311
Chapter 7	Embedded Vision Processors and Sensors	313
7.1	Vision Processors	313
	Hardware Platforms for Embedded Vision, Image Processing, and Deep Learning	317
	Requirements of Computers for Embedded Vision Application	319
	Processor Configuration Selection	321
7.2	Embedded Vision Processors	321
	Convolution Neural Networks (CNN) Engine	323
	Cluster Shared Memory	323
	Streaming Transfer Unit	323
	Bus Interface	323
	Complete Suite of Development Tools	324
	Intel Movidius Myraid X Vision Processors	324
	Matrox RadientPro CL	325
	Single-Board Computer Raspberry Pi	326
	Nvidia Jetson TX1	326
	Nvidia Jetson Tk1	326
	Beagle board: Beagle bone Black	327
	Orange Pi	327
	ODROID-C2	327
	Banana Pi	327
	CEVA—XM4 Imaging and Vision Processor	328

MAX10 FPGA	329
Vision DSPs for Imaging and Vision	330
Vision Q6 DSP Features and Benefits	331
Vision P6 DSP Features and Benefits	332
Vision P5 DSP Features and Benefits	333
VFPU	333
7.3 Sensors for Applications	338
Sensors for Industrial Applications	338
Sensors for Aviation and Aerospace	340
Sensors for the Automobile Industry	345
Agricultural Sensors	351
Smart Sensors	352
7.4 MEMS	355
NEMS	356
Biosensors	357
Medical Sensors	358
Nuclear Sensors	358
Sensors for Deep-Sea Applications	359
Sensors for Security Applications	362
Selection Criteria for Sensor	365
<i>Summary</i>	366
<i>References</i>	367
<i>Learning Outcomes</i>	367
<i>Further Reading</i>	368
Chapter 8 Computer Vision	369
8.1 Embedded Vision and Other Technologies	370
Robot Vision	371
Signal Processing	372
Image Processing	372
Pattern Recognition and Machine Learning	372
Machine Vision	373
Computer Graphics	373
Artificial Intelligence	374
Color Processing	374
Video Processing	374
Computer Vision Versus Machine Vision	374
Computer Vision Versus Image Processing	376
The Difference Between Computer Vision, Image Processing, and Machine Learning	377

8.2	Tasks and Algorithms in Computer Vision	379
	Image Acquisition	379
	Image Processing	380
	Image Analysis and Understanding	381
	Algorithms	384
	Feature Extraction	385
	Feature Extraction Algorithms	386
	Image Classification	389
	Object Detection	390
	Object Tracking	391
	Semantic Segmentation	392
	Instance Segmentation	393
	Object Recognition Algorithms	393
	SIFT: Scale Invariant Feature Transforms Algorithm	393
	SURF: Speed up Robust Features Algorithm	394
	ORB: Oriented Fast and Rotated Brief Algorithm	395
	Optical Flow and Point Tracking	397
	Commercial Computer Vision Software Providers	397
8.3	Applications of Computer Vision	398
	Packages and Frameworks for Computer Vision	403
8.4	Robotic Vision	404
	Mars Path Finder	406
	Cobots Versus Industrial Robots	406
	Machine Learning in Robots	407
	Sensors in Robotic Vision	408
	Artificial Intelligence Robots	411
	Robotic Vision Testing in the Automotive Industry	411
8.5	Robotic Testing in the Aviation Industry	413
	Robotic Testing in the Electronics Industry	414
	The Use of Drones and Robots in Agriculture	416
	Underwater Robots	417
	Autonomous Security Robots	418
	<i>Summary</i>	419
	<i>References</i>	419
	<i>Learning Outcomes</i>	420
	<i>Further Reading</i>	421
Chapter 9	Artificial Intelligence for Embedded Vision	423
9.1	Embedded Vision-based Artificial Intelligence	424
	AI-Based Solution for Personalized Styling and Shopping	429

AI Learning Algorithms	432
Algorithm Implementation Options	435
AI Embedded in Cameras	439
9.2 Artificial Vision	441
AI for Industries	442
9.3 3D-Imaging Technologies: Stereo Vision, Structured Light, Laser Triangulation, and ToF	444
1. Stereo Vision	444
2. Structured Light	450
3. Laser Triangulation	450
4. Time-of-Flight Camera for 3D Imaging	451
Theory of Operation	452
Working of ToF	455
Comparison of 3D-Imaging Technologies	456
Structured-Light Versus ToF	457
Applications of ToF 3D-Imaging Technology	459
Gesture Applications	460
Non-Gesture Applications	460
Time of Flight Sensor Advantages	461
9.4 Safety and Security Considerations in Embedded Vision Applications	462
Architecture Case Study	466
Choosing Embedded Vision Software	468
<i>Summary</i>	471
<i>References</i>	472
<i>Learning Outcomes</i>	473
<i>Further Readings</i>	473
Chapter 10 Vision-Based Real-Time Examples	475
10.1 Algorithms for Embedded Vision	476
Three Classes	476
Local Operators	478
Global Transformations	480
10.2 Methods and Models in Vision Systems	480
1. Shapes and Shape Models	480
2. Active Shape Model (ASM)	482
3. Clustering Algorithms	483

4. Thinning Morphological Operation	488
5. Hough Transform (HT)	490
10.3 Real-Time Examples	492
1. Embedded-Vision-Based Measurement	492
2. Defect Detection on Hardwood Logs Using Laser Scanning	497
3. Reconstruction of Monocular Fiberscopic Images	498
4. Vision Technologies for Empty Bottle Inspection Systems	500
5. Unmanned Rotorcraft for Ground Target Following Using Embedded Vision	502
6. Automatic Axle-Lifting System Design	513
7. Object Tracking Using an Address Event Vision Sensor	514
8. Using FPGA as an SoC Processor in ADAS Design	516
9. Diagnostic Imaging	522
10. Electronic Pill	523
10.4 Research and Development in Vision Systems	526
Robotic Vision	526
Stereo Vision	527
Vision Measurement	529
Industrial Vision	530
Automobile Industry	531
Medical Vision	531
Embedded Vision System	532
<i>Summary</i>	534
<i>References</i>	534
<i>Learning Outcomes</i>	535
<i>Further Readings</i>	535
Appendix	537
Embedded Vision Glossary	537
Index	551

Preface

Embedded Vision (EV) is an emerging electronics industry technology. It provides visual intelligence to automated embedded products. It combines embedded systems and computer vision and is the integration of a camera and a processing board. Embedded vision integrates computer vision in machines that use algorithms to decode meaning from observed images or video images. It has a wide range of potential applications to industrial, medical, automotive including driverless cars, drones, smart phones, aerospace, defense, agriculture, consumer, surveillance, robotics, and security. It will meet the requirements of algorithms in the computer vision field and the hardware and software requirements of the embedded systems field to give visual talent to end products.

This book is an essential guide for anyone who is interested in designing machines that can see, sense, and build vision-enabled embedded products. It covers a large number of topics encountered in the hardware architecture, software algorithms, applications, advancements in camera, processors, and sensors in the field of embedded vision. Embedded vision systems are built for special applications, whereas PC based systems are usually intended for general image processing.

Chapter 1 discusses introductory points, the design of an embedded vision system, characteristics of an embedded vision system board, processors and cameras for embedded vision, components in a typical vision system and embedded vision challenges. Application areas of embedded vision are analyzed. Development tools for embedded vision are also introduced in this chapter.

Chapter 2 discusses industrial vision. PC based vision systems, industrial cameras, high speed industrial cameras, and smart cameras are discussed in this chapter. The industrial vision applications are classified as dimensional quality inspection, surface quality inspection, structural quality, and operational quality. 3D imaging methods, 3D inspection, 3D processing, 3D robotic vision, capture and storage are analyzed under the heading of 3D industrial vision. 3D pattern matching, development approaches, development software tools, image processing analysis tools are also discussed.

Chapter 3 covers medical vision techniques. Image processing systems for medical applications, from images to information in medical vision, mathematics, algorithms in medical imaging, and machine learning in medical image analysis are discussed. Stereoscopic endoscope, CT, ultrasonic imaging system, MRI, X-ray, PACS, CIA, FIA, ophthalmology, indo cyanine green, automatic classification of cancerous cells, facial recognition to determine pain level, automatic detection of patient activity, peripheral vein imaging and the stereoscopic microscope, CAD processing, radiography, and telemedicine are covered. CNN, machine learning, deep learning and big data are also discussed.

Chapter 4 discusses video analytics. Definitions, applications, and algorithms of video analytics and video imaging are covered. Different types of machine learning algorithms and examples of ML such as CNN for an autonomous car, smart fashion AI architecture, and teaching computer to recognize animals are discussed.

Chapter 5 discusses digital image processing. The image processing concept, image manipulations, image analyzing and image processing steps for an embedded vision system are covered. Image sharpening, histograms, image transformation, image enhancement, convolution, blurring, edge detection are a few image manipulations discussed in this chapter. Frequency domain, transformation, filters, color spaces, jpeg compression, pattern matching, and template matching used to analyze images are discussed.

Chapter 6 discusses the history of photography, camera sensors for embedded vision applications, zooming camera, camera interface and camera selection for vision projects are discussed in this chapter.

Chapter 7 covers embedded vision processors and sensors. This chapter deals with the vision processor selection, embedded vision processor boards, different sensors based on the applications, and MEMS. The options for processor configuration, embedded vision processor boards and sensors suited for different applications are thoroughly discussed.

Chapter 8 discusses computer vision. This chapter compares various existing technologies with *embedded* vision. Tasks and algorithms in computer vision such as feature extraction, image classification, object detection, object tracking, semantic segmentation, instance segmentation, object recognition algorithms, optical flow, and point tracking are discussed. Commercial computer vision software providers are listed and the applications of computer vision and robotic vision are discussed.

Chapter 9 discusses the use of artificial intelligence in embedded vision. Embedded vision based artificial intelligence, artificial vision, 3D imaging technologies, safety & security considerations in EV applications are covered. AI based solutions for personalized styling and shopping, AI embedded cameras and algorithm implementations are discussed. Stereo vision, structured light, laser triangulation, and time of flight techniques for 3D images are compared.

Chapter 10 discusses vision based, real time examples. Algorithms for embedded vision, and methods and models in vision systems are covered. Recent research, applications, and developments in the field of embedded vision systems are analyzed.

EMBEDDED VISION

Overview

Embedded vision is the integration of vision in machines that use algorithms to decode meaning from observing images or videos. Embedded vision systems use embedded boards, sensors, cameras, and algorithms to extract information. Application areas are many, and include automobiles, medical, industry, domestic, and security systems.

Learning Objectives

After reading this the reader will be able to

- differentiate between embedded vision and computer vision,
- define embedded vision system,
- understand embedded vision system design requirements,
- understand application areas of embedded vision, and
- development tools for vision.

1.1 INTRODUCTION TO EMBEDDED VISION

Embedded vision refers to the practical use of computer vision in machines that understand their environment through visual means. *Computer vision* is the use of digital processing and intelligent algorithms to interpret meaning from images or video. Due to the emergence of very powerful, low-cost, and energy efficient processors, it has become possible to incorporate practical computer vision capabilities into embedded systems, mobile devices, PCs, and the cloud. Embedded vision is the integration of computer vision in machines that use algorithms to decode

meaning from observing pixel patterns in images or video. The computer vision field is developing rapidly, along with advances in silicon and, more recently, purpose designed embedded vision processors.

Embedded vision is the extraction of meaning from visual inputs, creating “machines that see and understand.” Embedded vision is now spreading into a very wide range of applications, including automotive driver assistance, digital signage, entertainment, healthcare, and education. Embedded vision is developing across numerous fields including autonomous medical care, agriculture technology, search and rescue, and repair in conditions dangerous to humans. Applications include autonomous machines of many types such as embedded systems, driverless cars, drones, smart phones, and rescue and bomb disarming robots. The term *embedded vision* refers to the use of computer vision technology in embedded systems. Stated another way, *embedded vision* refers to embedded systems that extract meaning from visual inputs. Similar to the way that wireless communication has become pervasive over the past 10 years, embedded vision technology will be very widely deployed in the next 10 years.

Computer (or machine) vision is the field of research that studies the acquisition, processing, analysis, and understanding of real-world visual information. It is a discipline that was established in the 1960s, but has made recent rapid advances due to improvements both in algorithms and in available computing technology. Embedded systems are computer systems with dedicated functions that are embedded within other devices, and are typically constrained by cost and power consumption. Some examples of devices using embedded systems include mobile phones, set top boxes, automobiles, and home appliances. *Embedded vision is an innovative technology in which computer vision algorithms are incorporated into embedded devices to create practical and widely deployable applications using visual data.* This field is rapidly expanding into emerging high-volume consumer applications such as home surveillance, games, automotive safety, smart glasses, and augmented reality.

With the emergence of increasingly capable processors, it's becoming practical to incorporate computer vision capabilities into a wide range of embedded systems, enabling them to analyze their environments via video inputs. Products like game controller and driver assistance systems are raising awareness of the incredible potential of embedded vision technology. As a result, many embedded system designers are beginning to think about implementing embedded vision capabilities. It's clear that embedded

vision technology can bring huge value to a vast range of applications. Two examples are eye's vision-based driver assistance systems, intended to help prevent motor vehicle accidents, and swimming pool safety systems, which help prevent swimmers from drowning.

The term embedded vision implies a hybrid of two technologies, embedded systems and computer vision. An embedded system is a *microprocessor-based system that isn't a general-purpose computer*, whereas computer vision refers to the *use of digital processing and intelligent algorithms to interpret meaning from images or video*. Most commonly defined, an embedded vision system is any microprocessor-based system with image sensor functionality that isn't a standard personal computer. Tablets and smart phones fall into this category, as well as more unusual devices such as advanced medical diagnosis instruments and robots with object recognition capabilities. So, to put it simply, embedded vision refers to *machines that understand their environment through visual means*.

Embedded vision processors are now developed by electronic companies to make computer vision lower in cost, lower in power and ready for smaller, more mobile. Embedded devices and coprocessing chips can be connected to neural networks or neural net processors to add efficient computer vision to machine learning. Two main trends of embedded vision are: Miniaturization of PCs and of cameras, and the possibility for vision systems to be produced affordably and for highly specific applications. Systems of this kind are referred to as embedded vision systems.

Visual inputs are the richest source of sensor information. For more than 50 years, scientists have tried to understand imaging and developed algorithms allowing computers to see with computer vision applications. The first real commercial applications, referred to as machine vision, analyzed fast moving objects to inspect and detect errors in products. Due to improving process power, lower power consumption, better image sensors, and better computer algorithms, vision elevates to a much higher level. Combining embedded systems with computer vision results in embedded vision systems. Embedded vision blocks are shown in Figures 1.1a and 1.1b.

Initially, embedded vision technology was found in complex, expensive systems, for example a surgical robot for hair transplantation or quality control inspection systems for manufacturing. Like wireless communication, embedded vision requires lots of processing power, particularly as applications increasingly adopt high-resolution cameras and

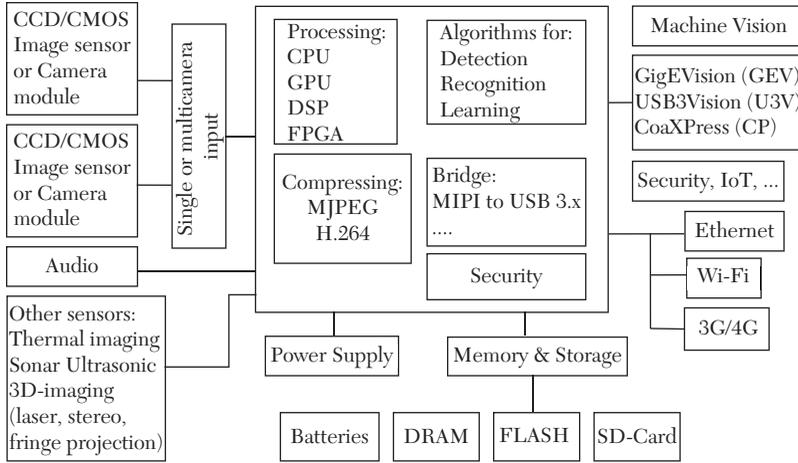


FIGURE 1.1A. Embedded vision system blocks.

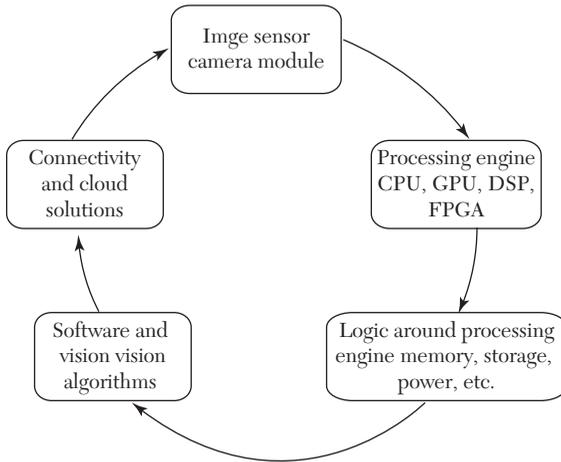


FIGURE 1.1B. Embedded vision block diagram.

make use of multiple cameras. Providing that processing power at a cost low enough to enable mass adoption is a big challenge. This challenge is multiplied by the fact that embedded vision applications require a high degree of programmability. In wireless applications algorithms don't vary dramatically from one cell phone handset to another, but in embedded vision applications there are great opportunities to get better results and enable valuable features through unique algorithms.

With embedded vision, the industry is entering a “virtuous circle” of the sort that has characterized many other digital signal processing application domains. Although there are few chips dedicated to embedded vision applications today, these applications are increasingly adopting high performance, cost effective processing chips developed for other applications, including DSPs, CPUs, FPGAs, and GPUs. As these chips continue to deliver more programmable performance per watt, they will enable the creation of more high volume embedded vision products. Those high-volume applications, in turn, will attract more attention from silicon providers, who will deliver even better performance, efficiency, and programmability.

1.2 DESIGN OF AN EMBEDDED VISION SYSTEM

An embedded vision system consists, for example, of a camera, a so called board level camera, which is connected to a processing board as shown in Figure 1.2. Processing boards take over the tasks of the PC from the classic machine vision setup. As processing boards are much cheaper than classic industrial PCs, vision systems can become smaller and also more cost effective. The interfaces for embedded vision systems are primarily USB or LVDS (Low voltage differential signaling connector).

As like embedded systems, there are popular single board computers (SBC), such as the Raspberry Pi are available on the market for embedded vision product development. Figure 1.3 shows the Raspberry Pi is a mini computer with established interfaces and offers a similar range of features as a classic PC or laptop. Embedded vision solutions can also be implemented with so-called *system on modules (SoM)* or *computer on modules (CoM)*. These modules represent a computing unit. For the adaptation of the desired interfaces to the respective application, a so called individual carrier board is needed. This is connected to the SoM via specific connectors and



FIGURE 1.2. Design of embedded vision system.



FIGURE 1.3. Embedded System Boards

can be designed and manufactured relatively simply. The SoMs or CoMs (or the entire system) are cost effective on the one hand since they are available off-the-shelf, while on the other hand they can also be individually customized through the carrier board. For large manufactured quantities, individual processing boards are a good idea.

All modules, single board computers, and SoMs, are based on a system on chip (SoC). This is a component on which the processor(s), controllers, memory modules, power management, and other components are integrated on a single chip. Due to these efficient components, the SoCs, embedded vision systems have only recently become available in such a small size and at a low cost.

Embedded vision is the technology of choice for many applications. Accordingly, the design requirements are widely diversified. The two interface technologies offered for embedded vision systems in the portfolio are USB3 Vision for easy integration and LVDS for a lean system design. USB 3.0 is the right interface for a simple plug and play camera connection and ideal for camera connections to single board computers. It allows the stable data transfer with a bandwidth of up to 350 MB/s. LVDS-based interface allows a direct camera connection with processing boards and thus also to on board logic modules such as FPGAs (field programmable gate arrays) or comparable components. This allows a lean system design to be achieved and can benefit from a direct board-to-board connection and data transfer. The interface is therefore ideal for connecting to a SoM on a carrier / adapter board or with an individually developed processor unit. It allows stable, reliable data transfer with a bandwidth of up to 252 MB/s.

Characteristics of Embedded Vision System Boards versus Standard Vision System Boards

Most of the previously mentioned single board computers and SoMs do not include the x86 family processors common in standard PCs. Rather, the CPUs are often based on the ARM architecture. The open source Linux operating system is widely used as an operating system in the world of ARM processors. For Linux, there are a large number of open source application programs, as well as numerous freely available program libraries. Increasingly, however, x86-based single board computers are also spreading. A consistently important criterion for the computer is the space available for the embedded system.

For the software developer, the program development for an embedded system is different than for a standard PC. As a rule, the target system does not provide a suitable user interface which can also be used for programming. The software developer must connect to the embedded system via an appropriate interface if available (e.g., network interface) or develop the software on the standard PC and then transfer it to the target system. When developing the software, it should be noted that the hardware concept of the embedded system is oriented to a specific application and thus differs significantly from the universally usable PC. However, the boundary between embedded and desktop computer systems is sometimes difficult to define. Just think of the mobile phone, which on the one hand has many features of an embedded system (ARM-based, single-board construction), but on the other hand can cope with very different tasks and is therefore a universal computer.

Benefits of Embedded Vision System Boards

A single board computer is often a good choice. Single board is a standard product. It is a small compact computer that is easy to use. This is also useful for developers who have had little to do with embedded vision. However, the single board computer is a system that contains unused components, and thus generally does not allow the leanest system configuration. Hence, this is suitable for small to medium quantities. The leanest setup is obtained through a customized system. Here, however, higher integration effort is a factor. This customized system is therefore suitable for large unit numbers. The benefits of embedded vision system boards at a glance are:

- Lean system design
- Light weight
- Cost-effective, because there is no unnecessary hardware
- Lower manufacturing costs
- Lower energy consumption
- Small footprint

Processors for Embedded Vision

This technology category includes any device that executes vision algorithms or vision system control software. The applications represent distinctly different types of processor architectures for embedded vision, and

each has advantages and trade-offs that depend on the workload. For this reason, many devices combine multiple processor types into a heterogeneous computing environment, often integrated into a single semiconductor component. In addition, a processor can be accelerated by dedicated hardware that improves performance on computer vision algorithms.

Vision algorithms typically require high compute performance. And, of course, embedded systems of all kinds are usually required to fit into tight cost and power consumption envelopes. In other digital signal processing application domains, such as digital wireless communications, chip designers achieve this challenging combination of high performance, low cost, and low power by using specialized coprocessors and accelerators to implement the most demanding processing tasks in the application. These coprocessors and accelerators are typically not programmable by the chip user, however. This trade-off is often acceptable in wireless applications, where standards mean that there is strong commonality among algorithms used by different equipment designers.

In vision applications, however, there are no standards constraining the choice of algorithms. On the contrary, there are often many approaches to choose from to solve a particular vision problem. Therefore, vision algorithms are very diverse, and tend to change fairly rapidly over time. As a result, the use of nonprogrammable accelerators and coprocessors is less attractive for vision applications compared to applications like digital wireless and compression centric consumer video equipment. Achieving the combination of high performance, low cost, low power, and programmability is challenging. Special purpose hardware typically achieves high performance at low cost, but with little programmability. General purpose CPUs provide programmability, but with weak performance, poor cost, or energy efficiency.

Demanding embedded vision applications most often use a combination of processing elements, which might include, for example:

- A general purpose CPU for heuristics, complex decision making, network access, user interface, storage management, and overall control
- A high-performance DSP-oriented processor for real time, moderate rate processing with moderately complex algorithms
- One or more highly parallel engines for pixel rate processing with simple algorithms

While any processor can in theory be used for embedded vision, the most promising types today are:

- High-performance embedded CPU
- Application specific standard product (ASSP) in combination with a CPU
- Graphics processing unit (GPU) with a CPU
- DSP processor with accelerator(s) and a CPU
- Field programmable gate array (FPGA) with a CPU
- Mobile “application processor”

High Performance Embedded CPU

In many cases, embedded CPUs cannot provide enough performance or cannot do so at an acceptable price or power consumption levels to implement demanding vision algorithms. Often, memory bandwidth is a key performance bottleneck, since vision algorithms typically use large amounts of memory bandwidth, and don't tend to repeatedly access the same data. The memory systems of embedded CPUs are not designed for these kinds of data flows. However, like most types of processors, embedded CPUs become more powerful over time, and in some cases can provide adequate performance. There are some compelling reasons to run vision algorithms on a CPU when possible. First, most embedded systems need a CPU for a variety of functions. If the required vision functionality can be implemented using that CPU, then the complexity of the system is reduced relative to a multiprocessor solution.

In addition, most vision algorithms are initially developed on PCs using general purpose CPUs and their associated software development tools. Similarities between PC CPUs and embedded CPUs (and their associated tools) mean that it is typically easier to create embedded implementations of vision algorithms on embedded CPUs compared to other kinds of embedded vision processors. In addition, embedded CPUs typically are the easiest to use compared to other kinds of embedded vision processors, due to their relatively straightforward architectures, sophisticated tools, and other application development infrastructure, such as operating systems. An example of an embedded CPU is the Intel Atom E660T.

Application Specific Standard Product (ASSP) in Combination with a CPU

Application specific standard products (ASSPs) are specialized, highly integrated chips tailored for specific applications or application sets. ASSPs may incorporate a CPU, or use a separate CPU chip. By virtue of specialization, ASSPs typically deliver superior cost and energy efficiency compared with other types of processing solutions. Among other techniques, ASSPs deliver this efficiency through the use of specialized coprocessors and accelerators. ASSPs are by definition focused on a specific application, they are usually provided with extensive application software.

The specialization that enables ASSPs to achieve strong efficiency, however, also leads to their key limitation lack of flexibility. An ASSP designed for one application is typically not suitable for another application, even one that is related to the target application. ASSPs use unique architectures, and this can make programming them more difficult than with other kinds of processors. Indeed, some ASSPs are not user programmable.

Another consideration is risk. ASSPs often are delivered by small suppliers, and this may increase the risk that there will be difficulty in supplying the chip, or in delivering successor products that enable system designers to upgrade their designs without having to start from scratch. An example of a vision-oriented ASSP is the PrimeSense PS1080-A2, used in the Microsoft Kinect.

General Purpose CPUs

While computer vision algorithms can run on most general purpose CPUs, desktop processors may not meet the design constraints of some systems. However, x86 processors and system boards can leverage the PC infrastructure for low-cost hardware and broadly supported software development tools. Several Alliance Member companies also offer devices that integrate a RISC CPU core. A general purpose CPU is best suited for heuristics, complex decision making, network access, user interface, storage management, and overall control. A general purpose CPU may be paired with a vision specialized device for better performance on pixel level processing.

Graphics Processing Units with CPU

High-performance GPUs deliver massive amounts of parallel computing potential, and graphics processors can be used to accelerate the portions of

the computer vision pipeline that perform parallel processing on pixel data. While General Purpose GPUs (GPGPUs) have primarily been used for high-performance computing (HPC), even mobile graphics processors and integrated graphics cores are gaining GPGPU capability meeting the power constraints for a wider range of vision applications. In designs that require 3D processing in addition to embedded vision, a GPU will already be part of the system and can be used to assist a general purpose CPU with many computer vision algorithms. Many examples exist of x86-based embedded systems with discrete GPGPUs.

Graphics processing units (GPUs), intended mainly for 3D graphics, are increasingly capable of being used for other functions, including vision applications. The GPUs used in personal computers today are explicitly intended to be programmable to perform functions other than 3D graphics. Such GPUs are termed “general purpose GPUs” or “GPGPUs.” GPUs have massive parallel processing horsepower. They are ubiquitous in personal computers. GPU software development tools are readily and freely available, and getting started with GPGPU programming is not terribly complex. For these reasons, GPUs are often the parallel processing engines of first resort of computer vision algorithm developers who develop their algorithms on PCs, and then may need to accelerate execution of their algorithms for simulation or prototyping purposes.

GPUs are tightly integrated with general purpose CPUs, sometimes on the same chip. However, one of the limitations of GPU chips is the limited variety of CPUs with which they are currently integrated. The limited number of CPU operating systems support the integration. Today there are low-cost, low-power GPUs, designed for products like smart phones and tablets. However, these GPUs are generally not GPGPUs, and therefore using them for applications other than 3D graphics is very challenging. An example of a GPGPU used in personal computers is the NVIDIA GT240.

Digital Signal Processors with Accelerator(s) and a CPU

DSPs are very efficient for processing streaming data, since the bus and memory architecture are optimized to process high-speed data as it traverses the system. This architecture makes DSPs an excellent solution for processing image pixel data as it streams from a sensor source. Many DSPs for vision have been enhanced with coprocessors that are optimized for processing video inputs and accelerating computer vision algorithms. The specialized nature of DSPs makes these devices inefficient for processing

general purpose software workloads, so DSPs are usually paired with a RISC processor to create a heterogeneous computing environment that offers the best of both worlds.

Digital signal processors (“DSP processors” or “DSPs”) are microprocessors specialized for signal processing algorithms and applications. This specialization typically makes DSPs more efficient than general purpose CPUs for the kinds of signal processing tasks that are at the heart of vision applications. In addition, DSPs are relatively mature and easy to use compared to other kinds of parallel processors. Unfortunately, while DSPs do deliver higher performance and efficiency than general purpose CPUs on vision algorithms, they often fail to deliver sufficient performance for demanding algorithms. For this reason, DSPs are often supplemented with one or more coprocessors. A typical DSP chip for vision applications therefore comprises a CPU, a DSP, and multiple coprocessors. This heterogeneous combination can yield excellent performance and efficiency, but can also be difficult to program. Indeed, DSP vendors typically do not enable users to program the coprocessors; rather, the coprocessors run software function libraries developed by the chip supplier. An example of a DSP targeting video applications is the Texas Instruments DM8168.

Field Programmable Gate Arrays (FPGAs) with a CPU

Instead of incurring the high cost and long lead times for a custom ASIC to accelerate computer vision systems, designers can implement an FPGA to offer a reprogrammable solution for hardware acceleration. With millions of programmable gates, hundreds of I/O pins, and compute performance in the trillions of multiply accumulates/sec (tera-MACs), high-end FPGAs offer the potential for highest performance in a vision system. Unlike a CPU, which has to use time slice or multi-thread tasks as they compete for compute resources, an FPGA has the advantage of being able to simultaneously accelerate multiple portions of a computer vision pipeline. Since the parallel nature of FPGAs offers so much advantage for accelerating computer vision, many of the algorithms are available as optimized libraries from semiconductor vendors. These computer vision libraries also include preconfigured interface blocks for connecting to other vision devices, such as IP cameras.

Field programmable gate arrays (FPGAs) are flexible logic chips that can be reconfigured at the gate and block levels. This flexibility enables the user to craft computation structures that are tailored to the application at hand.

It also allows selection of I/O interfaces and on-chip peripherals matched to the application requirements. The ability to customize compute structures, coupled with the massive amount of resources available in modern FPGAs, yields high performance coupled with good cost and energy efficiency. However, using FPGAs is essentially a hardware design function, rather than a software development activity. FPGA design is typically performed using hardware description languages (Verilog or VHDL) at the register transfer level (RTL) a very low-level of abstraction. This makes FPGA design time consuming and expensive, compared to using the other types of processors discussed here.

However using FPGAs is getting easier, due to several factors. First, so called “IP block” libraries—libraries of reusable FPGA design components are becoming increasingly capable. In some cases, these libraries directly address vision algorithms. In other cases, they enable supporting functionality, such as video I/O ports or line buffers. Second, FPGA suppliers and their partners increasingly offer reference designs reusable system designs incorporating FPGAs and targeting specific applications. Third, high-level synthesis tools, which enable designers to implement vision and other algorithms in FPGAs using high-level languages, are increasingly effective. Relatively low-performance CPUs can be implemented by users in the FPGA. In a few cases, high-performance CPUs are integrated into FPGAs by the manufacturer. An example FPGA that can be used for vision applications is the Xilinx Spartan-6 LX150T.

Mobile “Application Processor”

A mobile “application processor” is a highly integrated system-on-chip, typically designed primarily for smart phones but used for other applications. Application processors typically comprise a high-performance CPU core and a constellation of specialized coprocessors, which may include a DSP, a GPU, a video processing unit (VPU), a 2D graphics processor, an image acquisition processor, and so on. These chips are specifically designed for battery-powered applications, and therefore place a premium on energy efficiency. In addition, because of the growing importance of and activity surrounding smart phone and tablet applications, mobile application processors often have strong software development infrastructure, including low-cost development boards, Linux and Android ports, and so on. However, as with the DSP processors discussed in the previous section, the specialized coprocessors found in application processors are usually not user programmable, which limits their utility for vision applications. An example of a mobile application processor is the Freescale i.MX53.

Cameras/Image Sensors for Embedded Vision

While analog cameras are still used in many vision systems, this section focuses on digital image sensors usually either a CCD or CMOS sensor array that operates with visible light. However, this definition shouldn't constrain the technology analysis, since many vision systems can also sense other types of energy (IR, sonar, etc.).

The camera housing has become the entire chassis for a vision system, leading to the emergence of “smart cameras” with all of the electronics integrated. By most definitions, a smart camera supports computer vision, since the camera is capable of extracting application specific information. However, as both wired and wireless networks get faster and cheaper, there still may be reasons to transmit pixel data to a central location for storage or extra processing.

A classic example is cloud computing using the camera on a smart phone. The smart phone could be considered a “smart camera” as well, but sending data to a cloud-based computer may reduce the processing performance required on the mobile device, lowering cost, power, weight, and so on. For a dedicated smart camera, some vendors have created chips that integrate all of the required features.



FIGURE 1.4. Outdoor fixed security camera.

Until recent times, many people would imagine a camera for computer vision as the outdoor security camera shown in Figure 1.4. There are countless vendors supplying these products, and many more supplying indoor cameras for industrial applications. There are simple USB cameras for PCs available and billions of cameras embedded in the mobile phones of the world. The speed and quality of these cameras has risen dramatically supporting 10+ mega pixel sensors with sophisticated image-processing hardware.

Another important factor for cameras is the rapid adoption of 3D-imaging using stereo optics, time-of-flight and structured light technologies. Trendsetting cell phones now even offer this technology, as do the most recent generation of game consoles. Look again at the picture of the outdoor camera and consider how much change is about to happen to computer vision markets as new camera technologies become pervasive.

Charge coupled device (CCD) image sensors have some advantages over CMOS image sensors, mainly because the electronic shutter of CCDs traditionally offers better image quality with higher dynamic range and resolution. However, CMOS sensors now account for more 90% of the market, heavily influenced by camera phones and driven by the technology's lower cost, better integration, and speed.

Other Semiconductor Devices for Embedded Vision

Embedded vision applications involve more than just programmable devices and image sensors; they also require other components for creating a complete system. Most applications require data communications of pixels and/or metadata, and many designs interface directly to the user. Some computer vision systems also connect to mechanical devices, such as robots or industrial control systems.

The list of devices in this “other” category includes a wide range of standard products. In addition, some system designers may incorporate programmable logic devices or ASICs. In many vision systems, power, space, and cost constraints require high levels of integration with the programmable device often into a system-on-a-chip (SoC) device. Sensors to sense external parameters or environmental measurements are discussed in the separate chapter headings.

Memory

Processors can integrate megabytes' worth of SRAM and DRAM, so many designs will not require off-chip memory. However, computer vision algorithms for embedded vision often require multiple frames of sensor data to track objects. Off-chip memory devices can store gigabytes of memory, although accessing external memory can add hundreds of cycles of latency. The systems with a 3D-graphics subsystem will usually already include substantial amounts of external memory to store the frame buffer, textures, Z buffer, and so on. Sometimes this graphics memory is stored in a dedicated, fast memory bank that uses specialized DRAMs.

Some vision implementations store video data locally, in order to reduce the amount of information that needs to be sent to a centralized system. For a solid state, nonvolatile memory storage system, the storage density is driven by the size of flash memory chips. Latest generation NAND chip fabrication technologies allow extremely large, fast and low-power storage in a vision system.

Networking and Bus Interfaces

Mainstream computer networking and bus technology has finally started to catch up to the needs of computer vision to support simultaneous digital video streams. With economies of scale, more vision systems will use standard buses like PCI and PCI Express. For networking, Gigabit Ethernet (GbE) and 10GbE interfaces offer sufficient bandwidth even for multiple high-definition video streams. However, the trade association for Machine Vision (AIA) continues to promote Camera Link, and many camera and frame grabber manufacturers use this interface.

1.3 COMPONENTS IN A TYPICAL VISION SYSTEM

Although applications of embedded vision technologies vary, a typical computer vision system uses more or less the same sequence of distinct steps to process and analyze the image data. These are referred to as a vision pipeline, which typically contains the steps shown in Figure 1.5.

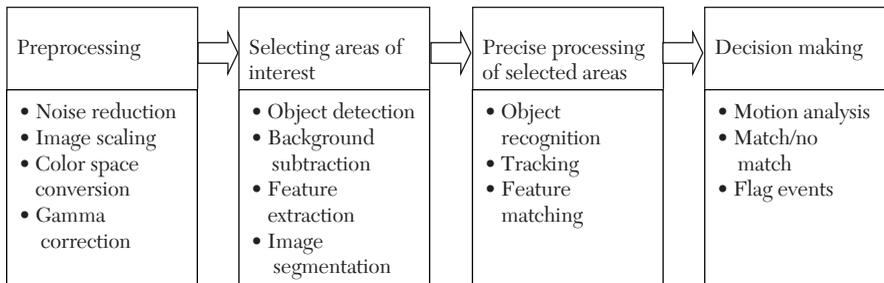


FIGURE 1.5. Vision pipeline.

At the start of the pipeline, it is common to see algorithms with simple data-level parallelism and regular computations. However, in the middle region, the data-level parallelism and the data structures themselves are both more complex, and the computation is less regular and more control-oriented. At the end of the pipeline, the algorithms are more general purpose in nature. Here are the pipelines for two specific application examples: Figure 1.6 shows a vision pipeline for a video surveillance application.

Figure 1.7 shows a vision pipeline for a pedestrian detection application. Note that both pipelines construct an image pyramid and have an object detection function in the center.

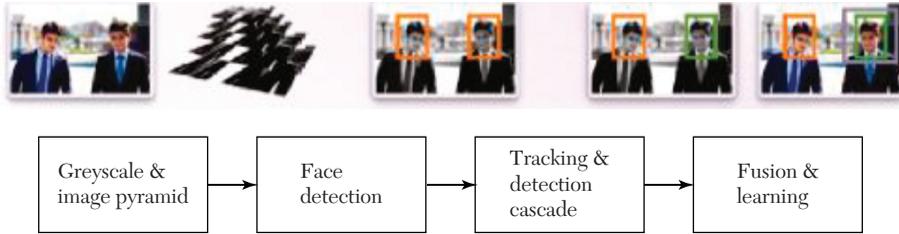


FIGURE 1.6. Vision pipeline for video surveillance application.

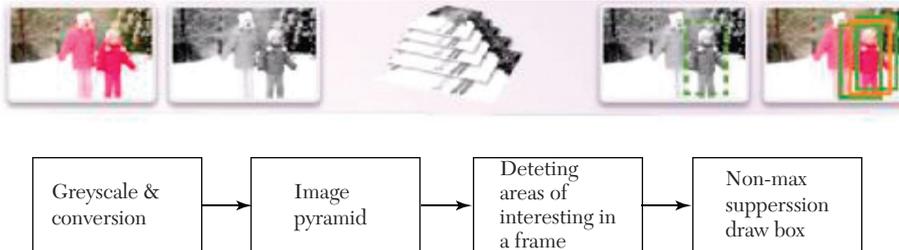


FIGURE 1.7. Vision pipeline for pedestrian detection application

Vision Processing Algorithms

Vision algorithms typically require high computing performance. And unlike many other applications, where standards mean that there is strong commonality among algorithms used by different equipment designers, no such standards that constrain algorithm choice exist in vision applications. On the contrary, there are often many approaches to choose from to solve a particular vision problem. Therefore, vision algorithms are very diverse, and tend to change fairly rapidly over time. And, of course, industrial automation systems are usually required to fit into tight cost and power consumption envelopes.

The rapidly expanding use of vision technology in industrial automation is part of a much larger trend. From consumer electronics to automotive safety systems, today we see vision technology (Figure 1.8) enabling a wide range of products that are more intelligent and responsive than before, and thus more valuable to users. We use the term “embedded vision” to refer to this growing practical use of vision technology in embedded systems, mobile devices, special purpose PCs, and the cloud, with industrial automation being one showcase application.

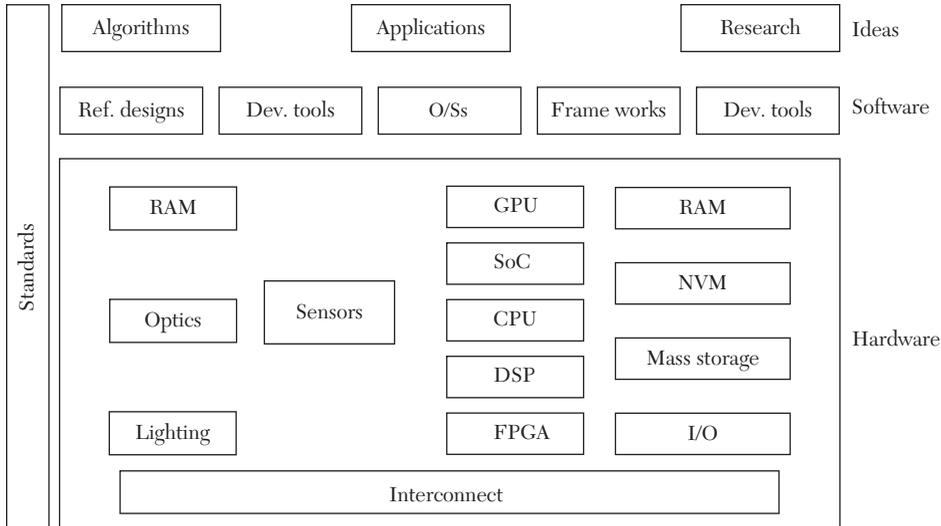


FIGURE 1.8. Vision technology.

Embedded Vision Challenges

Although the rapid progress of technology has made available very powerful microprocessor architectures, implementing a computer vision algorithm on embedded hardware/software platforms remains a very challenging task. Some specific challenges encountered by embedded vision systems include:

i. Power consumption

Vision applications for mobile platforms are constrained by battery capacity, leading to power requirements of less than one Watt. Using more power means more battery weight, a problem for mobile and airborne systems (i.e., drones). More power also means higher heat dissipation, leading to more expensive packaging, complex cooling systems, and faster aging of components.

ii. Computational requirements

Computer vision applications have extremely high computational requirements. Constructing a typical image pyramid for a VGA frame (640x480) requires 10–15 million instructions per frame. Multiply this by 30 frames per second and this will require a processor capable of doing 300–450 MIPS just to handle this preliminary processing step, let alone the more advanced recognition tasks required later in the

pipeline. State-of-the-art, low-cost camera technology today can provide 1080p or 4K video, at up to 120 frames per second. A vision system using such a camera requires compute power ranging from a few Giga Operations per Second (GOPS) to several hundred GOPS.

iii. Memory usage

The various vision processing tasks require large buffers to store processed image data in various formats, and high bandwidth to move this data from memory and between computational units. The on-chip memory size and interconnect has a significant impact on the cost and performance of a vision application on an embedded platform.

iv. Fixed-point algorithm development

Most of the published computer vision algorithms are developed for the computational model of Intel-based workstations where, since the advent of the Intel Pentium in 1993, the cost of double-precision operations is roughly identical to integer or single-precision operations. However, 64-80 bit hardware floating point units massively increase silicon area and power consumption, and software emulation libraries for floating point run slowly. For this reason, algorithms typically need to be refined to use the more efficient fixed-point data arithmetic based on integer types and operands combined with data shifts to align the radix point.

Besides the previously discussed challenges, an embedded vision developer should keep the dynamic nature of the market. The market is changing in an ongoing basis, including applications and use cases, the underlying vision algorithms, the programming models, and the supporting hardware architectures.

Currently, there is a need for standardized vision kernels, algorithm libraries, and programming models. At this time, there are no fully established standards for vision software with efficient hardware implementations. There are a number of likely candidates. OpenCV is a good starting point for reference algorithms and their test benches. Khronos is an emerging standard focused on embedded systems. OpenCL is a software framework to tie together massively parallel heterogeneous computation units.

1.4 APPLICATIONS FOR EMBEDDED VISION

The emergence of practical embedded vision technology creates vast opportunities for innovation in electronic systems and associated

software. In many cases, existing products can be transformed through the addition of vision capabilities. One example of this is the addition of vision capabilities to surveillance cameras, allowing the camera to monitor a scene for certain kinds of events, and alert an operator when such an event occurs. In other cases, practical embedded vision enables the creation of new types of products, such as surgical robots and swimming pool safety systems that monitor swimmers in the water. Some specific applications that use embedded vision include object detection, video surveillance, gesture recognition, Simultaneous Localization and Mapping (SLAM), and Advanced Driver Assistance Systems (ADAS). Let's take a closer look at each one.

Swimming Pool Safety System

While there are bigger markets for vision products, swimming pool safety (as shown in Figure 1.9) is one of those applications that truly shows the positive impact that technological progress can have for society. Every parent will instantly appreciate the extra layer of safety provided by machines that see and understand if a swimmer is in distress. When tragedies can happen in minutes, a vision system shows the true potential of this technology never becoming distracted or complacent in performing the duties of a digital lifeguard.

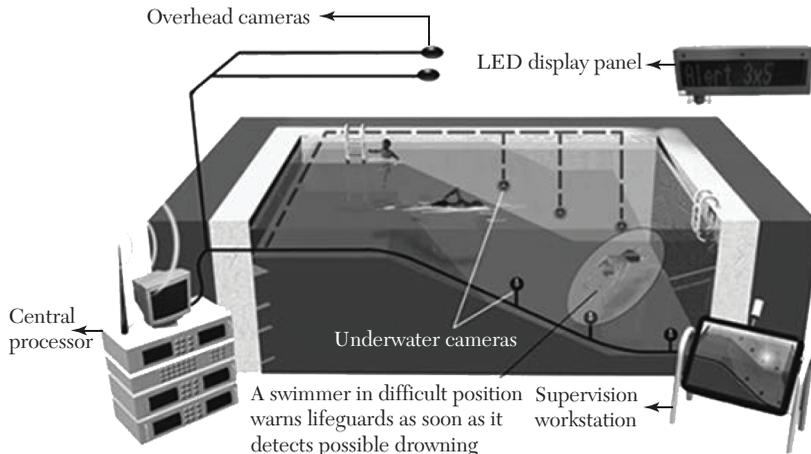


FIGURE 1.9. Pool graphic system

Object Detection

Object detection is at the heart of virtually all computer vision systems. Of all the visual tasks we might ask a computer to perform, the task of

analyzing a scene and recognizing all of the constituent objects remains the most challenging. Furthermore, detected objects can be used as inputs for object recognition tasks, such as instance or class recognition, which can find a specific face, a car model, a unique pedestrian, and so on. Applications include face detection and recognition, pedestrian detection, and factory automation. Even vision applications that are not specifically performing object detection often have some sort of detection step in the processing flow. For example, a movement tracking algorithm uses “corner detection” to identify easily recognizable points in an image and then looks for them in subsequent frames.

Video Surveillance

Growing numbers of IP cameras and the need for surveillance cameras with better video quality is driving the global demand for video surveillance systems. Sending HD resolution images of millions of IP cameras to the cloud will require prohibitive network bandwidth. So intelligence in the camera is needed to filter the video to only transmit the appropriate cases (e.g., only the frames with pedestrian detected, with background subtracted) for further analysis.

Gesture Recognition

Candidates for control by vision-based gesture recognition include automotive infotainment and industrial applications, where touch screens are either dangerous or impractical. For consumer electronics, gaming, and virtual reality, gesture recognition can provide a more direct interface for machine interaction.

Simultaneous Localization and Mapping (SLAM)

SLAM is the capability of mobile systems to create a map of an area they are navigating. This has applications in areas like self-driving cars, robot vacuum cleaners, augmented reality games, virtual reality applications, and planetary rovers.

Automatic Driver Assistance System (ADAS)

ADAS systems are used to enhance and automate vehicle systems to improve safety and driving by, for example, detecting lanes, other cars, road signs, pedestrians, cyclists, or animals in the path of a car. Another example of an emerging high-volume embedded vision application is automotive safety systems based on vision which is shown in Figure 1.10. A few



FIGURE 1.10. Mobileye driver assistance system.

automakers, such as Volvo, have begun to install vision-based safety systems in certain models. These systems perform a variety of functions, including warning the driver (and in some cases applying the brakes) when a forward collision is threatening, or when a pedestrian is in danger of being struck.

Another example of an emerging high-volume embedded vision application is “smart” surveillance cameras, which are cameras with the ability to detect certain kinds of activity. For example, the Archerfish

Solo, a consumer-oriented smart surveillance camera, can be programmed to detect people, vehicles, or other motion in user selected regions of the camera’s field of view.

Game Controller

In more recent decades, embedded computer vision systems have been deployed in applications such as target-tracking for missiles, and automated inspection for manufacturing plants. Now, as lower cost, lower power, and higher performance processors emerge, embedded vision is beginning to appear in high-volume applications. Perhaps the most visible of these is the Microsoft Kinect, a peripheral for the Xbox 360 game console that uses embedded vision to enable users to control video games simply by gesturing and moving their bodies. The success of Microsoft Kinect for the Xbox 360 game console as in Figure 1.11, subsequently expanded to support PCs, as well as the vision support in successor generation consoles from both Microsoft and Sony, demonstrates that people want to control their machines using natural language and gestures. Practical computer



FIGURE 1.11. Microsoft Kinect for Xbox 360, a gesture-based game controller.

vision technology has finally evolved to make this possible in a range of products that extend well beyond gaming. For example, the Microsoft Kinect with 3D motion capture, facial recognition, and voice recognition is one of the fastest selling consumer electronics devices. Such highly visible applications are creating consumer expectations for systems with visual intelligence and

increasingly powerful, low-cost, energy-efficient processors and sensors are making the widespread use of embedded vision practical.

Face Recognition for Advertising Research

An innovated technology tracks the facial responses of Internet users while they view content online. This would allow many companies to monitor Internet user's real-time reactions to their advertisements.

Mobile Phone Skin Cancer Detection

A smart phone application detects signs of skin cancer in moles on the human body. The application allows a person to take a photograph of a mole with the smart phone and receive an instant analysis of its status. Using a complex algorithm the application will tell the person, whether or not the mole is suspicious and advise on whether the person should seek treatment. The application also allows the person to find an appropriate dermatologist in the immediate vicinity. Other revolutionary medical applications utilizing embedded vision include an iPhone app that reads heart rate and a device to assist the blind by using a camera that interprets real objects and communicates them to the user as auditory indication.

Gesture Recognition for Car Safety

In the automotive industry, a new system incorporates gesture and face recognition to reduce distractions while driving. The use of face recognition for security purposes has been well documented; interpreting nods, winks, and hand movements to execute specific functions within the car. For example, a winking of the eye will turn the car radio off and on and by tilting head left or right, the volume will go up or down! Since many road accidents are the result of drivers trying to multitask, this application could potentially save many lives.

Industrial Applications for Embedded Vision

Vision-processing-based products have established themselves in a number of industrial applications. The most prominent one being factory automation where the application is commonly referred to as machine vision. It identifies the primary factory automation sectors as:

- Automotive—motor vehicle and related component manufacturing
- Chemical and Pharmaceutical—chemical and pharmaceutical manufacturing plants and related industries

- Packaging—packaging machinery, packaging manufacturers, and dedicated packaging companies not aligned to any one industry
- Robotics—guidance of robots and robotic machines
- Semiconductors and Electronics—semiconductor machinery makers, semiconductor device manufacturers, electronic equipment manufacturing, and assembly facilities

The primary embedded vision products used in factory automation applications are:

- Smart Sensors—A single unit that is designed to perform a single machine vision task. Smart sensors require little or no configuring and have limited on board processing. Frequently a lens and lighting are also incorporated into the unit.
- Smart Cameras—This is a single unit that incorporates a machine vision camera, a processor and I/O in a compact enclosure. Smart cameras are configurable and so can be used for a number of different applications. Most have the facility to change lenses and are also available with built in LED lighting.
- Compact Vision System—This is a complete machine vision system, not based on a PC, consisting of one or more cameras and a processor module. Some products have an LCD screen incorporated as part of the processor module. This obviates the need to connect the devices to a monitor for set up. The principal feature that distinguishes compact vision systems (CVS) from smart cameras is their ability to take information from a number of cameras. This can be more cost effective where an application requires multiple images.
- Machine Vision Cameras (MV Cameras)—These are devices that convert an optical image into an analogue or digital signal. This may be stored in random access memory, but not processed, within the device.
- Frame Grabbers—This is a device (usually a PCB card) for interfacing the video output from a camera with a PC or other control device. Frame grabbers are sometimes called video capture boards or cards. They vary from being a simple interface to a more complex device that can handle many functions including triggering, exposure rates, shutter speeds, and complex signal processing.

- **Machine Vision Lighting**—This refers to any device that is used to light a scene being viewed by a machine vision camera or sensor. This report considers only those devices that are designed and marketed for use in machine vision applications in an industrial automation environment.
 - **Machine Vision Lenses**—This category includes all lenses used in a machine vision application, whether sold with a camera or as a spare or additional part.
 - **Machine Vision Software**—This category includes all software that is sold as a product in its own right, and is designed specifically for machine vision applications. It is split into:
 - **Library Software**—allows users to develop their own MV system architecture. There are many different types, some offering great flexibility. They are often called SDKs (Software Development Kits).
 - **System Software**—which is designed for a particular application. Some are very comprehensive and require little or no set up.

Medical Applications for Embedded Vision

Embedded vision and video analysis have the potential to become the primary treatment tool in hospitals and clinics, and can increase the efficiency and accuracy of radiologists and clinicians. The high quality and definition of the output from scanners and X-ray machines makes them ideal for automatic analysis, be it for tumor and anomaly detection, or for monitoring changes over a period of time in dental offices or for cancer screening. Other applications include motion analysis systems, which are being used for gait analysis for injury rehabilitation and physical therapy. Video analytics can also be used in hospitals to monitor the medical staff, ensuring that all rules and procedures are properly followed.

For example, video analytics can ensure that doctors “scrub in” properly before surgery, and that patients are visited at the proper intervals. Medical imaging devices including CT, MRI, mammography, and X-ray machines, embedded with computer vision technology and connected to medical images taken earlier in a patient’s life, will provide doctors with very powerful tools to help detect rapidly advancing diseases in a fraction of the time currently required. Computer-aided detection or computer-aided diagnosis (CAD) software is currently also being used in early stage deployments to assist doctors in the analysis of medical images by helping to highlight potential problem areas.

Automotive Applications for Embedded Vision

Vision products in automotive applications can serve to enhance the driving experience by making us better and safer drivers through both driver and road monitoring. Driver monitoring applications use computer vision to ensure that driver remains alert and awake while operating the vehicle. These systems can monitor head movement and body language for indications that the driver is drowsy, thus posing a threat to others on the road. They can also monitor for driver distraction behaviors such as texting, eating, and so on, responding with a friendly reminder that encourages the driver to focus on the road instead.

In addition to monitoring activities occurring inside the vehicle, exterior applications such as lane departure warning systems can use video with lane detection algorithms to recognize the lane markings and road edges and estimate the position of the car within the lane. The driver can then be warned in cases of unintentional lane departure. Solutions exist to read roadside warning signs and to alert the driver if they are not paying attention, as well as for collision damage control, blind spot detection, park and reverse assist, self-parking vehicles, and event data recording. Eventually, this technology will lead cars with self-driving capability; However many automotive industry experts believe that the goal of vision in vehicles is not so much to eliminate the driving experience but to just to make it safer. The primary vision products in the automotive market are camera modules.

Security Applications for Embedded Vision

Vision technology was first deployed for security applications, addressing the fundamental problem that the millions of security video cameras vastly outnumber the availability of humans to monitor them in real time. Moreover, studies have shown that the attention span of a human is only about 20 minutes when watching a camera feed. In most cases, video cameras historically only helped to review the scene after an incident had already occurred. A tireless and ubiquitous embedded vision system solves many of these problems and can even provide an alert to take action in real time.

Vision products have been used in the physical security market for a number of years. The technology is commonly referred to as video content analysis (VCA) or intelligent video in the physical security industry. It is used in virtual tripwire applications, providing perimeter intrusion detection for high-risk facilities such as airports and critical infrastructure.

Typical projects use video analytics to generate real-time alerts when an object is identified and tracked through a predefined area. Operators, based in the central control room, can manage more video streams as they are not required to identify intruders purely from watching the monitors. Instead, the vision product will identify and alert the operator when there is a potential security issue. It is then up to the human operator to decide how best to deal with the threat.

Other applications of vision products in the security space include wrong way detection and trajectory tracking. These types of algorithm can be used in airport exit lanes, where security managers want to identify people entering the restricted area from the main terminal. This responsibility has historically been given to security guards; however, vision technology can reduce security guard costs for airport operators and increase the accuracy of the security system.

Other applications in the security market include behavior recognition algorithms such as “slip and fall” and abandoned/left object detection. These applications are less developed than the perimeter detection systems, but will provide powerful tools for security managers over the coming years.

Consumer Applications for Embedded Vision

The days of the traditional game controller may well be numbered, as sensors and video analytics enable detection of body movements and even facial expressions to become the interface between player and game. Consumer applications for embedded vision products have taken off in a big way since the launch of the first generation Xbox Kinect, the fastest selling consumer electronic device in history. The PlayStation Move quickly followed suit and now software developers are rapidly developing titles that take advantage of the technology. Consumer applications for vision are not limited to the gaming industry, however. Remote control applications are already in use today, via vendors such as GestureTek and PrimeSense (for the Kinect, now owned by Apple), which eliminate the need for a physical remote control.

Furthermore, embedded vision can be used in other household applications such as set top boxes, televisions, PCs, smart phones and tablets, lighting, heating, and air conditioning to further eliminate the need for physical controls. Mobile electronics devices are a particularly compelling vision platform, because they often already contain both front (for “selfies” and video conferencing) and rear mounted (for still and video photography)

high-resolution cameras, along with powerful application processors and abundant DRAM and nonvolatile storage.

In addition, video analytics and intelligent video applications will become integral to the way consumers absorb content. The ability to analyze vast libraries of video content as well as live video, including movies, sports and TV shows in addition to user-generated content, opens up countless possibilities. For example, by metatagging video with information about its content, actors, players, and so on, it will be possible to effectively search the rapidly expanding library of video content available to consumers. This easy search capability will be vital in facilitating a truly personal TV experience, with the consumer able to quickly and accurately search and find all of the content fitting their interests. Virtually any device requiring human input represents an opportunity for vision processing.

Machine Learning in Embedded Vision Applications

Embedded vision systems often have the task of classifying images captured by the camera: on a conveyor belt, for example, in round and square biscuits. In the past, software developers have spent a lot of time and energy developing intelligent algorithms that are designed to classify a biscuit based on its characteristics (features) in type A (round) or B (square). In this example, this may sound relatively simple, but the more complex the features of an object, the more difficult it becomes. Algorithms of machine learning (e.g., Convolutional Neural Networks, CNNs), however, do not require any features as input. If the algorithm is presented with large numbers of images of round and square cookies, together with the information which image represents which variety, the algorithm automatically learns how to distinguish the two types of cookies. If the algorithm is shown a new, unknown image, it decides for one of the two varieties because of its “experience” of the images already seen. The algorithms are particularly fast on graphics processor units (GPUs) and FPGAs.

1.5 INDUSTRIAL AUTOMATION AND EMBEDDED VISION: A POWERFUL COMBINATION

Vision technologies enable machines and robots to adapt to evolving manufacturing line circumstances. In order for manufacturing robots and other industrial automation systems to meaningfully interact with the

objects they are assembling, as well as to cleverly and safely move about in their environments, they must be able to see and understand their surroundings. Cost effective and capable vision processors, fed by depth discerning image sensors and running robust software algorithms, are transforming longstanding autonomous and adaptive industrial automation aspirations into reality.

Automated systems in manufacturing line environments are capable of working more tirelessly, faster, and more precise than humans. However, their success has traditionally been predicated on incoming parts arriving in fixed orientations and locations, thereby increasing manufacturing process complexity. Any deviation in part position or orientation causes assembly failures. Humans use their eyes (along with other senses) and brains to understand and navigate through the world around them. Robots and other industrial automation systems should be able to do the same thing. They can leverage camera assemblies, vision processors, and various software algorithms in order to skillfully adapt to evolving manufacturing line circumstances, as well as to extend vision processing benefits to other areas of the supply chain, such as piece parts and finished goods inventory tracking.

Inventory Tracking

Figure 1.12 shows a pharmaceutical packaging line using vision-guided robots to quickly pick syringes from conveyor belt and place them into packages. Embedded vision innovations can help improve product tracking through production lines and with enhanced storage efficiency. While bar codes and radio frequency identification tags can also help track and route materials, they cannot be used to detect damaged or flawed goods. Intelligent raw material and product tracking and handling in the era of embedded vision will be the foundation for the next generation of inventory management systems, as image sensor technologies continue to mature and as other vision processing components become increasingly integrated. High-resolution cameras can already provide detailed images of work material and inventory tags,



FIGURE 1.12. Vision-guided robots in packaging of pharmaceutical industry.

but complex, real-time software is needed to analyze the images, to identify objects within them, to identify ID tags associated with these objects, and to perform quality checks.

The phrase “real time” can potentially mean rapidly evaluating dozens of items per second. To meet the application’s real-time requirements, various tasks must often run in parallel. On-the-fly quality checks can be used to spot damaged material and to automatically update an inventory database with information about each object and details of any quality issues. Vision systems for inventory tracking and management can deliver robust capabilities without exceeding acceptable infrastructure costs, by integrating multiple complex real-time video analytics extracted from a single video stream.

Automated Assembly

Embedded vision is a key enabling technology for the factory production floor in areas such as raw materials handling and assembly. Cameras find use in acquiring images of, for example, parts or destinations. Subsequent vision processing sends data to a robot, enabling it to perform functions such as picking up and placing a component. As previously mentioned, industrial robots inherently deliver automation benefits such as scalability and repeatability. Adding vision processing to the mix allows these machines to be far more flexible. The same robot can be used with a variety of parts, because it can see which particular part it is dealing with and adapt accordingly.

Factories can also use vision in applications that require high-precision assembly; cameras can “image” components after they are picked up, with slight corrections in the robot position made to compensate for mechanical imperfections and varying grasping locations. Picking parts from a bin also becomes easier. A camera can be used to locate a particular part with an orientation that can be handled by the robotic arm, within a pile of parts.

Depth-discerning 3D vision is a growing trend that can help robots perceive even more about their environments. Cost-effective 3D vision is now appearing in a variety of applications, from vision-guided robotic pin picking to high-precision manufacturing metrology. Latest generation vision processors can now adeptly handle the immense data sets and sophisticated algorithms required to extract depth information and rapidly make decisions. Three-dimensional imaging is enabling vision process tasks that were previously impossible with traditional 2D cameras. Depth

information can be used, for example, to guide robots in picking up parts that are disorganized in a container.

Automated Inspection

An added benefit of using vision for robotic guidance is that the same images can also be used for inline inspection of the parts being handled. In this way, not only robots are more flexible, they can produce higher quality results. This outcome can also be accomplished at lower cost, because the vision system can detect, predict, and prevent “jam” and other undesirable outcomes. If a high degree of accuracy is needed within the robot’s motion, a technique called *visual servo control* can be used. The camera is either fixed to or nearby the robot and gives continuous visual feedback (versus only a single image at the beginning of the task) to enable the robot controller to correct for small errors in movement.

Beyond robotics, vision has many uses and delivers many benefits in automated inspection. It performs tasks such as checking for the presence of components, reading text and bar codes, measuring dimensions and alignment, and locating defects and patterns (Figures 1.13 and 1.14). Historically, quality assurance was often performed by randomly selecting samples from the production line for manual inspection, and then using statistical analysis to extrapolate the results to the larger manufacturing run. This approach leaves unacceptable room for defective parts to cause jams in machines further down the manufacturing line or for defective products to be shipped. Automated inspection, however, can provide 100% quality assurance. And with recent advancements in vision processing performance, automated visual inspection is frequently no longer the limiting factor in manufacturing throughput.



FIGURE 1.13. Three-dimensional imaging is used to measure the shape of a cookie and inspect it for defects.



FIGURE 1.14. Production and assembly applications, such as the system in this winery, need to synchronize a sorting system with the visual inspection process.

The vision system is just one piece of a multistep puzzle and must be synchronized with other equipment and input/output protocols to work well within an application. A common inspection scenario involves separating faulty parts from correct ones as they transition through the production line. These parts move along a conveyor belt with a known distance between the camera and the ejector location that removes faulty parts. As the parts migrate, their individual locations must be tracked and correlated with the image analysis results, in order to ensure that the ejector correctly sorts out failures.

Multiple methods exist for synchronizing the sorting process with the vision system, such as time stamps with known delays and proximity sensors that also keep track of the number of parts that pass by. However, the most common method relies on encoders. When a part passes by the inspection point, a proximity sensor detects its presence and triggers the camera. After a known encoder count, the ejector will sort the part based on the results of the image analysis.

The challenge with this technique is that the system processor must constantly track the encoder value and proximity sensors while simultaneously running image processing algorithms, in order to classify the parts and communicate with the ejection system. This multifunction juggling can lead to a complex software architecture, add considerable amounts of latency and jitter, increase the risk of inaccuracy, and decrease throughput. High-performance processors, such as field programmable gate arrays, are now being used to solve this challenge by providing a hardware timed method of tightly synchronizing inputs and outputs with vision inspection results.

Workplace Safety

Humans are still a key aspect of the modern automated manufacturing environment, adding their flexibility to adjust processes “on the fly.” They need to cooperate with robots, which are no longer confined in cages but share the work space with their human coworkers. Industrial safety is a big challenge, since increased flexibility and safety objectives can be contradictory. A system deployed in a shared work space needs to have a higher level of perception of surrounding objects, such as other robots, work pieces, and human beings.

Three-dimensional cameras help create a reliable map of the environment around the robot. This capability allows for robust detection

of people in safety and warning zones, enabling adaptation of movement trajectories and speeds for cooperation purposes, as well as collision avoidance. They aim to offer a smart and flexible approach to machine safety, necessary for reshaping factory automation.

Depth Sensing

3D cameras can deliver notable advantages over their 2D precursors in manufacturing environments. Several depth sensor technology alternatives exist, each with strengths, shortcomings, and common use cases (Table 1.1). Stereoscopic vision, combining two 2D image sensors, is currently the most common 3D-sensor approach. Passive (i.e., relying solely on ambient light) range determination via stereoscopic vision uses the disparity in viewpoints between a pair of near identical cameras to measure the distance to a subject of interest. In this approach, the centers of perspective of the two cameras are separated by a baseline or interpupillary distance to generate the parallax necessary for depth measurement.

TABLE 1.1. Three-Dimensional Vision Sensor Technology Comparisons

	Stereoscopic vision	Structured light		Time of flight
		Fixed pattern	Programmable pattern	
Depth accuracy	mm to cm <i>difficulty with smooth surface</i>	mm to cm	μm to cm <i>variable patterns and different light sources improve accuracy</i>	mm to cm <i>depends on resolution of sensor</i>
Scanning speed	Medium <i>limited by software complexity</i>	Fast <i>limited by camera speed</i>	Fast/medium <i>limited by camera speed</i>	Fast <i>limited by sensor speed</i>
Distance range	Midrange	Very short to midrange <i>Depends on illumination power</i>	Very short to midrange <i>Depends on illumination power</i>	Short to long range <i>Depends on laser power and modulation</i>

Microsoft Kinect is today's best known structured light-based 3D sensor. The structured light approach, like the time-of-flight technique is an example of an active scanner, because it generates its own electromagnetic radiation and analyzes the reflection of this radiation from the object. Structured light projects a set of patterns onto an object, capturing the resulting image with an offset image sensor. Similar to stereoscopic vision techniques, this approach takes advantage of the known camera-to-projector separation to locate a specific point between them and compute the depth with triangulation algorithms. Thus, image processing and triangulation algorithms convert the distortion of the projected patterns, caused by surface roughness, into 3D information.

An indirect time-of-flight (ToF) system obtains travel time information by measuring the delay or phase shift of a modulated optical signal for all pixels in the scene. Generally, this optical signal is situated in the near-infrared portion of the spectrum so as not to disturb human vision. The ToF sensor in the system consists of an array of pixels, where each pixel is capable of determining the distance to the scene. Each pixel measures the delay of the received optical signal with respect to the sent signal. A correlation function is performed in each pixel, followed by averaging or integration. The resulting correlation value then represents the travel time or delay. Since all pixels obtain this value simultaneously, "snap-shot" 3-D imaging is possible. The 3D-imaging technologies discussed in detail in Section 9.3.

Human need to explore lot of ideas and research in the application of embedded vision. In coming chapter lot of application are discussed. But it is not limited only with these.

1.6 DEVELOPMENT TOOLS FOR EMBEDDED VISION

The software tools (compilers, debuggers, operating systems, libraries, etc.) encompass most of the standard arsenal used for developing real-time embedded processor systems, while adding in specialized vision libraries and possibly vendor specific development tools for software development. On the hardware side, the requirements will depend on the application space, since the designer may need equipment for monitoring and testing real-time video data. Most of these hardware development tools are already used for other types of video system design.

Both General Purpose and Vendor Specific Tools

Many vendors of vision devices use integrated CPUs that are based on the same instruction set (ARM, x86, etc.), allowing a common set of development tools for software development. However, even though the base instruction set is the same, each CPU vendor integrates a different set of peripherals that have unique software interface requirements. In addition, most vendors accelerate the CPU with specialized computing devices (GPUs, DSPs, FPGAs, etc.). This extended CPU programming model requires a customized version of standard development tools. Most CPU vendors develop their own optimized software tool chain, while also working with 3rd-party software tool suppliers to make sure that the CPU components are broadly supported.

Personal Computers

The personal computer is both a blessing and a curse for embedded vision development. Most embedded vision systems and virtually all vision algorithms are initially developed on a personal computer. The PC is a fabulous platform for research and prototyping. It is inexpensive, ubiquitous, and easy to integrate with cameras and displays. In addition, PCs are endowed with extensive application development infrastructure, including basic software development tools, vision specific software component libraries, domain specific tools (such as MATLAB), and example applications. In addition, the GPUs found in most PCs can be used to provide parallel processing acceleration for PC-based application prototypes or simulations.

However, the PC is not an ideal platform for implementing most embedded vision systems. Although some applications can be implemented on an embedded PC (a more compact, lower-power cousin to the standard PC), many cannot, due to cost, size, and power considerations. In addition, PCs lack sufficient performance for many real-time vision applications.

And, unfortunately, many of the same tools and libraries that make it easy to develop vision algorithms and applications on the PC also make it difficult to create efficient embedded implementations. For example vision libraries intended for algorithm development and prototyping often do not lend themselves to efficient embedded implementation.

OpenCV

OpenCV is a free, open-source computer vision software component library for personal computers, comprising over two thousand algorithms. Originally developed by Intel, now maintained by Willow Garage. The OpenCV library, used along with Bradski and Kahler's book, is a great way to quickly begin experimenting with computer vision. However, OpenCV is not a solution to all vision problems. Some OpenCV functions work better than others. And OpenCV is a library, not a standard, so there is no guarantee that it functions identically on different platforms. In its current form, OpenCV is not particularly well suited to embedded implementation. Ports of OpenCV to non-PC platforms have been made, and more are underway, but there's currently little coherence to these efforts.

Heterogeneous Software Development in an Integrated Development Environment

Since vision applications often require a mix of processing architectures, the development tools become more complicated and must handle multiple instruction sets and additional system debugging challenges. Most vendors provide a suite of tools that integrate development tasks into a single interface for the developer, simplifying software development and testing.

Developing embedded vision systems is challenging. One consideration, already mentioned, is that vision algorithms tend to be very computationally demanding. Squeezing them into low-cost, low-power processors typically requires significant optimization work, which in turn requires a deep understanding of the target processor architecture.

Another key consideration is that vision is a system level problem. That is, success depends on numerous elements working together, besides the vision algorithms themselves. These include lighting, optics, image sensors, image preprocessing, and image storage subsystems. Getting these diverse elements working together effectively and efficiently requires multidisciplinary expertise. There are numerous algorithms available for vision functions, so in many cases it is not necessary to develop algorithms from scratch. But picking the best algorithm for the job, and ensuring that it meets application requirements, can be a large project in itself.

Summary

- Embedded vision is the extraction of meaning from visual inputs, creating machines that see and understand.

- Embedded vision is a hybrid of two technologies—embedded systems and computer vision.
- Smart sensor is a single unit designed to perform a single machine vision task.
- Smart camera is a single unit that incorporates a machine vision camera, a processor and I/O in a compact enclosure.
- Processor choice is Embedded CPU, ASSP, GPU, DSP, FPGA depends on application.
- Power consumption memory usage, algorithm development are challenges in embedded vision.

Reference

<https://www.isa.org/standards-and-publications/factory-automation-industrial-automation-and-embedded-vision-a-powerful-combination/>

Learning Outcomes

- 1.1 Define Embedded vision. Draw embedded vision blocks.
- 1.2 What are the benefits of embedded vision system?
- 1.3 List some applications of embedded vision system.
- 1.4 Why industrial automation and embedded vision is called as powerful combination?
- 1.5 Write short note on processors for embedded vision.
- 1.6 What are the components to design embedded vision?
- 1.7 Write about vision pipeline.
- 1.8 Give example for pedestrian detection and video surveillance.
- 1.9 Explain about development tools for embedded vision.

Further Reading

1. *Embedded Computer Vision* by Brainslav Kisacanin, Shuvra S. Bhattacharyya, Sek Chai
2. *Learning Open CV* by Gary Bradski and Adrian Kaehler

INDUSTRIAL VISION

Overview

In an industrial vision camera, sensors and a processor make decisions on what the camera sees and then relays information back to the handling system. It introduces automation into the production process in all levels of product manufacturing in different kind of industries. An industrial inspection system computes information from raw images according to the steps of image acquisition, image processing, feature extraction, and decision making. An industrial camera is designed according to high standards with repeatable performance, and it must be robust to withstand the demands of harsh industrial environments. The industrial vision applications are classified into four types: dimensional quality, surface quality, structural quality, and operational quality.

Learning Objectives

After reading this, the reader will know about

- the need of industrial cameras, trouble shooting, and smart cameras
- classification of industrial vision applications
- development in 3D industrial vision, 3D imaging methods
- inspection, processing, capturing, storage, tracking, measurement automation
- industrial vision measurement techniques
- software tools for industrial vision systems

2.1 INTRODUCTION TO INDUSTRIAL VISION SYSTEMS

Vision systems are now considered to be an essential part of many industrial processes; as they can offer fast, accurate, and reproducible inspection capabilities at a highly competitive cost. For instance, in the food industry, vision technology plays a crucial role for processes where speed and accuracy are extremely important. It helps to secure a competitive advantage for manufacturers. The food product itself is inspected to aid portion control. Vision technology checks quality as well as the packaging and labeling of each product. The pharmaceutical market requires the most demanding vision systems that not only inspect products, but also audit the systems use and configuration. It ensures that the correct drug and dose are delivered with full traceability.

Industrial vision encompasses all technology that allows a machine to see. It's also referred to as computer vision or image processing. However the term *machine vision* has become linked with industrial vision where a camera and processor make a decision on what it sees. Industrial vision relays information back to the handling system. Some applications of industrial vision include automated product inspection, code reading, and vision guided robotics.

Industrial vision requires specific capabilities such as asynchronous triggering and progressive scan. These capabilities ensure that a sharp high quality image is captured at exactly the right time without distortion. These are generally not found in cameras used for broadcast, security, and multimedia. These features find themselves in all cameras developed for industrial vision. A good industrial vision system also includes robust algorithms to measure and identify items in the image and then pass data in a compatible format back to the controlling programmable logic controller (PLC). Industrial vision systems come in many forms. It includes smart cameras which have everything in the camera body along with embedded multicamera systems and PC-based systems for more complex applications.

All industrial vision systems require an element of machine vision software. The software is for camera control or to complete a bespoke (computer program) written or adapted for a specific user or purpose application. For many industrial inspections, an easy to configure machine vision development environment with simple user interfaces is needed. It should allow most cost effective solutions to be deployed. For more bespoke requirements, companies with good software development skills

often use a machine vision software library. With image processing, vision systems should provide measurement capabilities with click user interfaces. The vision systems have their own role in the automation process. They also provide a powerful link with robotics. The term *image processing software* is a very broad area applied to commercially available photo manipulation packages used by photographers. Image processing software, photos, and other image types can be manipulated to improve an image. For example, unwanted components are removed by airbrushing; a form of image processing.

In the context of industrial vision, image processing is used to enhance, filter, mask, and analyze images. Well-known machine or industrial vision software packages such as Common Vision Blox, Scorpion Vision Software, Halcon, Matrox Imaging Library, or Cognex VisionPro are applications that run on Microsoft Windows. They are used to create advanced and powerful automation software that will take image input and output data based on the content of the image. Ultimately, in commercial industrial vision, image processing is used to classify, read characters, and recognize shapes or measures.

Industrial vision systems can introduce automation into the production process at a number of different levels. It simply speeds up the inspection process. It is used as an integral part of a statistical process control system. It can identify when a manufacturing process is moving out of specification. Hence remedial action can be taken before any defective product is manufactured. Online manufacturing inspection systems acquire the images of products and inspect them in real time before providing a decision about product quality. This can be useful in identifying problems and enabling process improvements to stop substandard items getting through to the next stage of production. There are basically three main types of vision systems: smart cameras; compact vision systems, and PC-based systems.

Other areas to consider when specifying industrial vision systems are machine vision illumination products. Machine vision lighting is a critical step. When designing an industrial vision system, getting the illumination geometry and correct wavelength makes the vision system more reliable. It simplifies the inspection task considerably. Selecting the right lens for an industrial vision camera can have a significant effect on the image quality and therefore success of the industrial vision application. With a wide range of lens mounts, sensor resolutions, and sensor sizes it's important to

choose an industrial lens with the correct specifications for the application. The wide variety of available photo electric sensor types with their varying function characteristics makes it possible to solve nearly every detection problem.

Machine vision provides innovative solutions in the direction of industrial automation. A plethora of industrial activities have benefited from the application of machine vision technology on manufacturing processes. These activities include, among others, delicate electronics component manufacturing, quality textile production, metal product finishing, glass manufacturing, machine parts, printing products, granite quality inspection, integrated circuits manufacturing, and many others. Machine vision technology improves productivity and quality management and provides a competitive advantage to industries that employ this technology.

Traditionally, visual inspection and quality control are performed by human experts. Although humans can do the job better than machines in many cases, they are slower than the machines and get tired quickly. Moreover, human experts are difficult to find or maintain in an industry, require training, and their skills may take time to develop. There are also cases for which inspection tends to be tedious or difficult, even for the best-trained experts. In certain applications, precise information must be quickly or repetitively extracted and used (e.g., target tracking and robot guidance). In some environments (e.g., underwater inspection, nuclear industry, chemical industry, etc.) inspection may be difficult or dangerous. Embedded vision may effectively replace human inspection in such demanding cases.

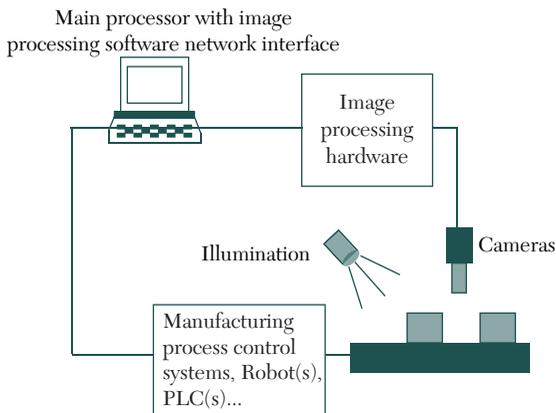


FIGURE 2.1. A typical industrial vision system.

Figure 2.1 illustrates the structure of a typical industrial vision system. First, a computer is employed for processing the acquired images. This is achieved by applying special purpose image processing analysis and classification software. Images are usually acquired by one or more cameras placed at the scene under

inspection. The positions of the cameras are usually fixed. In most cases, industrial automation systems are designed to inspect only known objects at fixed positions.

The scene is appropriately illuminated and arranged in order to facilitate the reception of the image features necessary for processing and classification. These features are also known in advance. When the process is highly time constrained or computationally intensive and exceeds the processing capabilities of the main processor, application specific hardware (e.g., DSPs, ASICs, or FPGAs) is employed to reduce the problem of processing speed. The results of this processing can be used to:

- Control a manufacturing process (e.g., for guiding robot arms placing components on printed circuits, painting surfaces, etc.)
- Propagate to other external devices (e.g., through a network or other type of interface like FireWire) for further processing (e.g., classification)
- Characterize defects of faulty items and take actions for reporting and correcting these faults and replacing or removing defective parts from the production line
- The requirements for the design and development of a successful machine vision system vary depending on the application domain and are related to the tasks to be accomplished, environment, speed, and so on. For example, in machine vision inspection applications, the system must be able to differentiate between acceptable and unacceptable variations or defects in products. While in other applications, the system must enable users to solve the guidance tasks, alignment tasks, measurement, and also the assembly verification tasks.

There exists no industrial vision system capable of handling all tasks in every application field. Only once the requirements of a particular application domain are specified, then appropriate decisions for the design and development of the application can be taken. The first problem to solve in automating a machine vision task is to understand what kind of information the machine vision system is to retrieve and how this is translated into measurements or features extracted from images. For example, it is important to specify in advance what “defective” means in terms of measurements and rules and implement these tasks in software or hardware. Then, a decision has to be made on the kind of measurements

to be acquired (e.g., position or intensity measurements) and on the exact location for obtaining the measurements.

For the system to be reliable, it must reduce “escape rates” (i.e., nonaccepted cases reported as accepted) and “false alarms” (i.e., accepted cases reported as nonaccepted) as much as possible. It is a responsibility of the processing and classification units to maintain system reliability, but the effectiveness of classification depends also on the quality of the acquired images. An industrial vision system must also be robust. Thus, it should adapt itself automatically and achieve consistently high performance despite irregularities in illumination, marking, or background conditions and, accommodate uncertainties in angles, positions, and so on. Robust performance is difficult to achieve. High recognition and classification rates are obtained only under certain conditions of good lighting and low noise. Finally, an industrial vision system must be fast and cost efficient.

The important attributes of an industrial machine vision inspection system are such as, flexibility, efficiency in performance, speed and cost, reliability and robustness. In order to design a system that maintains these attributes it is important to clearly define its required outputs and the available inputs. Typically, an industrial inspection system computes information from raw images according to the following sequence of steps:

1. **Image acquisition:** Images containing the required information are acquired in digital form through cameras, digitizers, and so on.
2. **Image processing:** Once images have been acquired, they are filtered to remove background noise or unwanted reflections from the illumination system. Image restoration may also be applied to improve image quality by correcting geometric distortions introduced by the acquisition system (e.g., the camera).
3. **Feature extraction:** A set of known features, characteristic of the application domain, is computed, probably with some consideration for nonoverlapping or uncorrelated features, so that better classification can be achieved. Examples of such features include size, position, contour measurement via edge detection, and linking, as well as and texture measurements on regions. Such features can be computed and analyzed by statistical or other computing techniques (e.g., neural networks or fuzzy systems). The set of computed features forms the description of the input image.

4. Decision making: Combining the feature variables into a smaller set of new feature variables reduces the number of features. While the number of initial features may be large, the underlying dimensionality of the data, or the intrinsic dimensionality, may be quite small. The first step in decision making attempts to reduce the dimensionality of the feature space to the intrinsic dimensionality of the problem. The reduced feature set is processed further as to reach a decision. This decision, as well as the types of features and measurements (the image descriptions) computed, depends on the application. For example, in the case of visual inspection during production the system decides if the produced parts meet some quality standards by matching a computed description with some known model of the image (region or object) to be recognized. The decision (e.g., model matching) may involve processing with thresholds, statistical, or soft classification.

At the last level of decision making and model matching previously mentioned, there are two types of image (region or object) models that can be used; namely, declarative and procedural. Declarative models consist of constraints on the properties of pixels, objects, or regions and on their relationships. Procedural models are implicitly defined in terms of processes that recognize the images. Both types of models can be fuzzy or probabilistic, involving probabilistic constraints and probabilistic control of syntactic rules respectively. A special category of models is based on neural networks.

Model-based approaches often require that descriptions (e.g., features) of the image at different levels of specificity or detail be matched with one of many possible models of different classes of images. This task can become very difficult and computationally intensive if the models are complex and a large number of models must be considered. In a top-down approach to model matching, a model might guide the generation of appropriate image descriptions rather than first generating the description and then attempting to match it with a model. Another alternative would be to combine top-down and bottom-up processes. The previously described control strategies are simplified when one is dealing with two-dimensional images taken under controlled conditions of good lighting and low noise, as is usually the case in industrial vision applications. Image descriptions and class models are easier to construct in this case and complex model matching can be avoided. Model-based approaches to visual inspection tasks have been applied in a variety of application fields.

PC-Based Vision Systems

PC-based systems can support the most complex image processing capabilities with a versatility that ranges from single PC and single camera to multicomputer, multicamera configurations. Cable and connector products for industrial vision is another important area to consider in most imaging or machine vision applications. A robust interconnect between the industrial camera and the PC or image processor is required. There are not only many camera interfacing standards available but also environmental requirements for machine vision cable such as the need for robotic flex, fire-safe, and resistance to chemical exposure. Standard industrial vision cables as well as custom and specialist cables are all available. PC-based vision systems all require an interface between the camera and computer. In modern systems this is based on a number of industrial vision camera interface standards. Some interfacing standards use the consumer ports that reside inside a PC such as USB or FireWire, whereas others require an additional camera interface card often called a frame grabber.

Industrial Cameras

Cameras are present in every day of our lives. Walking down the street the CCTV (closed circuit television) on buildings is commonplace, while driving speed/safety cameras are clearly evident, and every modern cellphone has a built-in camera. It is obvious these devices exist. They are an integral part of our lives today. However, very few people are aware that there are hundreds of thousands of cameras being used behind the scenes in manufacturing plants worldwide. These industrial cameras, also known as machine vision cameras, are being used to inspect a vast range of products, from headache tablets to shampoo bottles to spark plugs. The list goes on and covers industries like automotive, pharmaceutical, food and beverage, electronics, and print and packaging. An industrial camera is a camera that has been designed with high standards, repeatable performance, and is robust enough to withstand the demands of harsh industrial environments. These are commonly referred to as machine vision cameras as they are used on manufacturing processes for inspection and quality control.

Industrial vision cameras typically conform to a defined standard such as Firewire, GigE Vision, CameraLink, USB, and CoaXPress. The purposes of these standards are to facilitate ease of integration and to ensure future flexibility for camera upgrades.

There are two main types of cameras. They are area scan and line scan. An area scan camera is a CCD/CMOS sensor in a 2D matrix of pixels. This results in an image consisting of pixels in the X and Y direction (i.e., a normal looking image as taken by mobile phone). Industrial area scan camera run from ten to hundreds of frames per second. With a line scan camera, the CCD/CMOS sensor typically contains only a single row of pixels. This means that the object to be captured must be moved under the line scan camera to “build” a 2D matrix image. Line scan cameras run from hundreds to thousands of images per second are ideal for web applications where products are being manufactured continuously, such as paper and textiles and/or where the products are large in size. Machine vision cameras can be combined with illumination, optics, image processing software, and robotics to create fully automated inspection solutions.

High-speed Industrial Cameras

High-speed cameras in industrial applications are used mainly to troubleshoot high-speed events and machinery where speeds are such the human eye can't see what's happening. Hence they are also called troubleshooting cameras. The fastest high-speed cameras tend to have the image memory inside the unit enabling them to record up to 10,000 frames per second. A camera for video troubleshooting applications not only needs speed, but also needs specialist triggering modes. It is important in some applications to know the cause of the triggered event. So pre-event recording is common in high-speed cameras that capture continuously. When the event happens, data from both before and after is stored. Another key feature in troubleshooting cameras is the ability to self-trigger through the lens. This feature monitors a small region of the image for a change and when this happens makes a trigger point. The resulting image sequence includes images before and after the trigger region changed. Modern high-speed troubleshooting cameras are normally controlled via a laptop, which makes them portable and provides the ability to store and play back videos using standard technology.

One challenge with very high-speed cameras is light. With very short exposures, a lot of light is needed to get good quality images. Many of the providers also have a good solution of lighting to address many applications. High-speed cameras without memory also exist but these are generally integrated in PC systems where the image data is recorded in to PC memory rather than the camera memory. The limit here is the interface speed which

tends to typically be ten times slower than cameras with inbuilt memory. These systems tend to be used less in troubleshooting applications rather high-speed inspection applications.

Smart Cameras

Smart cameras combine the sensor, processor, and I/O of a vision system in a compact housing, often no bigger than a standard industrial camera, so that all of the image processing is carried out onboard. By combining all these items in a single package, costs are minimized. These systems are ideal where only one inspection view is required or where no local display or user control is needed. Many smart cameras are offered with additional extension products such as expanded I/O, image display, and control interfaces.

The various types of cameras used in embedded vision applications are elaborately discussed in Chapter 6.

2.2 CLASSIFICATION OF INDUSTRIAL VISION APPLICATIONS

In modern industrial-vision-system research and development, most applications are related to at least one of the following four types of inspection:

1. Inspection of dimensional quality,
2. Inspection of surface quality,
3. Inspection of correct assembling (structural quality), and
4. Inspection of accurate or correct operation (operational quality).

A formalization of this categorization is attempted in the following, by probing further the characteristics of products being inspected. Table 2.1 gathers some of the most ordinary inspected features of products.

TABLE 2.1. Potential Features of Inspected Products

Dimensional	Dimensions, shape, positioning, orientation, alignment, roundness, corners	
Structural	Assembly	Holes, slots, rivets, screws, clamps
	Foreign objects	Dust, bur, swarm
Surface	Pits, scratches, cracks, wear, finish, roughness, texture, seams-folds-laps, continuity	
Operational	Incompatibility of operation to standards and specifications	

Industrial vision applications may also be classified based on features whose measurements do not affect the inspection process (may take any value) allowing the system to be independent on these types of features. The set of such features defines the so-called “degrees of freedom” (DoFs) of the inspection process. Some of the most common DoFs met in the industrial world are concern about shape, geometrical dimensions, intensity, texture, pose, and so on. The DoFs of objects are strongly related to the variances of their characteristics and are considered to be a measure of the flexibility of the vision system. The fewer the DoFs, the stronger the dependency of the inspection system on the application for which it is originally designed. Therefore, systems with low DoFs are less likely to be expandable. However, high levels of variability are characteristic of more general or expandable systems.

To allow many DoFs, the system must employ sophisticated image classification approaches based on carefully selected models and algorithms, as to minimize its dependency on the inspected item and its potential deformations. Moreover, the more the DoFs of a system, the greater its potential for expandability. For example, the system can be enhanced to detect new types of defects if additional image processing and analysis functions are introduced to the system and applied independently from the old ones to capture more image features (e.g., capture surface in addition to dimensional characteristics). The previous considerations concerning the proposed classification based on DoFs reveals a known trade-off in the design of inspection systems between flexibility, complexity, and cost which is not obvious in other classifications.

Table 2.2 illustrates the relationship between DoFs and quality inspection systems developed for applications. Most applications focus on the inspection of a single characteristic (e.g., size). The remaining characteristics (e.g., finish, texture, etc.) can be considered as DoFs for these applications indicating the flexibility of the vision system.

Dimensional Quality

Checking whether the dimensions of an object are within specified tolerances or the objects have the correct shape, are ordinary tasks for industrial vision systems. Such tasks involve inspection of geometrical characteristics of objects in two or three dimensions namely the inspection of dimensional quality.

TABLE 2.2 Classification of Industrial Vision Systems

Quality inspection	Application field	Degrees of freedom (DoFs)
SURFACE	Mini resistor painting	Resistor orientation
	Aluminum sheet casting	Sheet width
	Railroad line inspection	Illumination/rail-foot-head position
	Oil seals	Illumination
	Chicken meat defects	Illumination/skin
	Wafer surface inspection	Distortion/scale/orientation/position
	Surface approximation	Illumination
	Granite surface inspection	Texture
	Directional texture	Illumination/rotation of direction
	Surface roughness	Orientation
	Surface defects	Pose
	Textile seam defects	Translation/rotation
	Internal wood defects	Wood density
	Wood veneer surface	Scale/intensity
	Surface corrosion	Shape
DIMENSIONAL	Machined parts inspection	Scale/translation/orientation
	Solder joints inspection	Orientation/position/size
	External screw threads	Thread position
	Banknotes inspection	Position
	Image segmentation	Shape/texture
	Object classification	Scale/orientation/shape
	Circular parts	Peripheral occlusion
	Packaging	Position/orientation
	Line segment measurement	Orientation/scale
	Fruit harvesting	Maturity/illumination/occlusion
	Packaging	Shape/size/illumination
	Automotive industry	Size/shape/pose
STRUCTURAL	Object assembly	Orientation (limited)
	Railroad parts inspection	Illumination/shape/texture
	Automotive industry	Illumination/position/shape/size
	PCB inspection	Illumination
OPERATIONAL	Laser butt joint welding	Welding path shape/gap size/beam position
	Wrist watch quality	Hands shape/size/orientation/distortion

Various industries are involved in the development of vision systems for automated measurement of dimensional quality. In packaging industry, the tasks vary from measurements of the fill level in bottles, to sell by date inspection and to airbag canister inspection (e.g., online gauging systems that measure height, concentricity, and diameter of airbag canisters). A vision-guided system for the automated supply of packaging machines with paper and foil material system enables the manipulator to locate the proper bobbin, depalletize it, and transfer it to the requesting machine. A vision system is used to determine the correct position of pallets and recognize the arrangement pattern of sacks on the pallets. The system enables a robot mechanism to grasp the sacks and pass them along a rotating cutting disk.

A popular and demanding real-time application is the inspection and classification of solder joints on printed circuit boards (PCBs). A typical inspection system for this application consists of a camera with appropriate illumination placed on top of the PCB conveyor system. Processing PCB images consists of two major stages: First a preprocessing is performed in order to remove noise and make the tracking of solder joints on the image of the PCB easy. Then, the solder joints are classified according to the types of defects. The usual classification is concerned with the quantity of the solder paste placed on a joint. Four classes are defined: good, excess solder, insufficient, and no solder. Simulation results on geometric models of joints have shown that efficient classification can be achieved only by an optimal feature selection, so that the classes do not overlap.

Current research has shown that histogram-based techniques perform better than two and three-dimensional feature-based techniques, both in terms of system and computational complexity. The major problem is that two-dimensional features alone are insufficient for correct classification and an extra classifier is required to separate overlapping classes. A combination of histogram (discussed in digital image processing chapter) and 2D, 3D feature-based techniques can overcome the performance of other techniques relying only on topological features. Many PCB inspection systems rely on neural networks for the design of classifiers that can deal with both distribution (histogram) and topological features of defects. An approach to the problem of cutting two-dimensional stock sheets is machine vision system. It is employed to acquire images of irregularly shaped sheets. Then, a genetic algorithm is applied to generate part layouts that satisfy the manufacturing constraints (e.g., minimization of trim losses). This method is particularly useful for the leather and apparel industries, where irregular parts are commonly used.

An automatic visual system for the location of spherical fruits on trees, under natural conditions is also done by industrial vision system. The system utilizes a laser range finder that provides range and attenuation data of the inspected fruit surface and shape analysis algorithms are employed to process the acquired reflectance and range images to locate the fruit and, finally, to determine the position of the fruit on the tree. This system is embedded in the AGRIBOT integrated robotic system, aimed at the automatic harvesting of fruits.

The problem of measuring line segments is a primary machine vision problem. A heuristic algorithm for line segment measurement is used to assess the efficiency of a machine vision system in accurately measuring properties of line segments, such as length, angle, and straightness. A similar application concerns the detection of circular parts with peripheral defects or irregularities is applied with two-stage Hough transform algorithm for the detection of circular machine parts.

A model-based computer vision system for the estimation of poses of objects in industrial environments, at near real-time rates is also done in vision system. A demanding real-time application is the detection of high-quality printed products. This application deals with products with high degree of resemblance, where minor differences among them makes the application very difficult to cope with, considering its real-time nature. An algorithm based on morphological operations facilitates the detection of flaws at near-pixel resolution. The system is applied for the inspection of banknotes, which is clearly a very delicate application, considering the requirements in the validity of the produced printings.

An interesting application in this category deals with the inspection of screw threads for compliance with manufacturing standards. Edge detection algorithms (based on linear interpolation to the subpixel resolution) are applied to detect regions of interest. Each such region is matched with multiple models of threads, since the dimensions and positions of the inspected threads are allowed to vary. The system has been tested on the production line and has been shown to perform better than other competitive methods, such as manual measurement. Active shape models as the basis of a generic object search technique are employed. The approach is based on the identification of characteristic or “landmark” points (i.e., points that exist in all aspects of the object) in images and on the recording of statistics concerning the relationships between the positions of the landmark points in a set of training examples. The effectiveness of the approach is demonstrated on inspection of automotive brake assemblies.

Surface Quality

Inspecting objects for scratches, cracks, wear, or checking surfaces for proper finish, roughness, and texture are typical tasks of surface quality inspection. Significant labor savings are achieved in textile, wood, and metal industries employing vision systems for fault detection and quality verification. The quality of textile seams is assessed using feature classification based on self-organizing neural networks. The system also enables the expedition of seam quality judgement, compared to human inspection, by locating seams on images of low contrast and then inspecting the waviness of the seam specimens. This information is in fact three-dimensional.

CATALOG is a system for internal wood-log defects detection, based on computer axial tomography (CAT or CT). Sequences of CT image slices are acquired and each one is segmented into two-dimensional regions. Each segmented image slice is analyzed and is characterized as defect-free or defect-like. The correlation of defect-like regions across a CT sequence enables the three-dimensional reconstruction of the log defects. The use of a decision tree in combination with a modular neural network topology is shown to be more efficient than a single large neural network alone for the classification of wood veneer. The design of this topology is based on normalized inter-class variation of features for separating between classes. An improved version of this topology, based on intra-class variation of features, allows for the reduction of the complexity of the neural network topology and results in improved classification accuracy.

Machine vision can also be used for the inspection and visualization of defects on ground or machined components (e.g., cracks, pitting, and changes in material quality). Segmentation techniques are used for the detection of characteristic surface faults (e.g., indentations, protrusions). Similar techniques are applied for the detection of scratches during machine polishing of natural stone. The assessment of surface roughness of machined parts is done by the Fourier transform method for the extraction of roughness measures. Then, neural networks are employed for the classification of surfaces based on roughness. The inspection of defects on objects with directionally textured surfaces (e.g., natural wood, machined surfaces, and textile fabrics) is also possible with vision system. For that, global image restoration scheme based on Fourier transform is applied.

High-frequency Fourier components corresponding to line patterns are discriminated from low-frequency ones corresponding to defective regions. An alternative approach for the inspection of randomly textured color

images is a method that considers both color and texture image properties and introduces a color similarity measure that allows the application of the watershed transform. The problem of recovering depth information for surface approximation of objects is achieved using stereo image pairs, a scanning electron microscope (SEM) and involves computation of disparity estimates utilizing a feature-based stereo algorithm.

The use of a finite-window robust sequential estimator for the detection and analysis of corrosion in range images of gas pipelines is also possible with industrial vision system. Deviations from the robust surface fit (which correspond to statistical outliers) represent potential areas of corrosion. The algorithm estimates surface parameters over a finite sliding window. The technique is shown to be robust in that it estimates the pipeline surface range function in the presence of noise, surface deviations, and changes in the underlying model. Despite the fact that the method exhibits real-time execution capability, it fails to interpret correctly the combinations of high magnitude and high-frequency ripples with large patches of corrosion.

Surface inspection is also applied to the aluminum strip casting process. Infrared (IR) temperature measurements (providing a measure of the distribution of surface temperature) are used to evaluate the quality of aluminum sheets. A two-level process for the inspection of aluminum sheets is done. First, the system inspects both sides of an aluminum sheet and captures images of potential defective areas. These images are then classified according to defect type and stored for review by experts. Machine vision is applied for the inspection of wafer surfaces in Integrated Circuits (IC) production. A fuzzy membership function is used to cope with the wide range of shape variations of the dimple defects.

Potential applications of surface quality inspection also include detection of damages on railroad tracks, where on-board detection and classification of defects is performed in real time. Exhaustive (100%) quality inspection of painting of metal film miniresistors is also possible. In that detection of low-quality products is achieved by the acquisition of a line pattern image of a correctly painted resistor, which is compared with each acquired line pattern image. Inspection of machined parts (e.g., circular oil seals) is done with verification of both surface and dimensional qualities. The center of each circular seal is computed and the intensities of its circumferential pixels are inspected.

In the food industry, the inspection of the quality of goods is of primary interest. An intelligent system for the detection of defects in chicken meat

before packaging is done by a vision system. The system relies on the analysis of chromatic content of chicken images, the extraction of potential defective areas by morphological processing, and their classification according to a predefined list of defects.

Structural Quality

Checking for missing components (e.g., screws, rivets, etc.) on assembled parts or checking for the presence of foreign or extra objects (e.g., leaves, little sticks) are typical tasks of this class of quality inspection. In semiconductor and electronics industries, the tasks vary from connector presence, capacitor polarity checking, Integrated Circuit (IC) pin gauging, IC identification, IC alignment and positioning, to information-gathering tasks such as automatic defect classification on electronic circuit boards, and so on. For example, a connector inspection system is designed to be fast and capable of detecting bent pins on connectors with 20–1000 pins—all done by a vision system. The structural quality of PCB components is efficiently done by the industrial vision system. The inspected objects (electronic components) are assumed to have little variations in size or shape but significant variation in grey-level appearance. Statistical models of the intensity across the objects structure in a set of training examples are built. Subsequently, a multi-resolution search technique is used to locate the model that matches a region of an input image. A fit measure with predictable statistical properties is also used to determine that this region is a valid instance of the model. The method demonstrates failure rates that are acceptable for use in a real environment (i.e., 1 in 1000 samples).

Template matching methods for the detection of anomalies in a car assembly line in real time is another application. Templates corresponding to four image regions of a car are selected by a human supervisor and are analyzed by the system. This work is part of an integrated system for automatic inspection of a complete car assembly line. A second part of the same system aimed at the inspection of the condition of vehicle doors in order to detect whether a vehicle door is open or closed. For that a line-fitting algorithm is applied.

Filtering techniques for detecting rail clamps and neural networks for detecting screws are employed. The detection of wooden ties of rail lines in real time is done in vision system. This system enables the detection of tie boundaries on rail-line images. An adaptive edge detector based on a modified Marr-Hildreth operator is employed to cope with the steep

transitions in the image resulting from wood grain. A stochastic model-based inspection algorithm (based on Bayesian estimation) is used for the detection of assembly errors on rigid objects. The image models describe the appearance of a complex three-dimensional object in a two-dimensional monochrome image. This method is applied for verifying correct assembly in a gear assembly line and a video home system (VHS) cassette production line.

Operational Quality

Inspection of operational quality is related to the verification of correct or accurate operation of the inspected products according to the manufacturing standards. The inspection of laser butt joint welding is done in the industrial vision system. A camera captures the welding seam track and determines the proper welding path and gap size. A noise-eliminating step is applied first. Then, the welding path and gap are calculated on segmented welding images. Segmentation is based on Laplacian filters. The information previously computed enables the control of the laser for the automatic welding of butt joints. Quality inspection of wristwatches is possible with vision system. All inspected watches are first synchronized with a reference clock. Images of watch hands are acquired by a camera and are classified as hour, minute, second, and 24-hour hands. The difficulty of this task stems from the overlapping of hands, as well as from the existence of a curved magnifying glass over the date window of the watch, which corrupts the clarity of captured images of hands. To compensate for such problems, the time that a watch shows is detected and compared with the time of the reference clock using neural network classifiers.

2.3 3D INDUSTRIAL VISION

One of the more recent developments in industrial vision is the wide commercial availability of algorithms and software tools that can process and measure pixels in the third dimension. The most common applications work in two dimensions: X and Y. For example, in the real world this translates to the accurate location of an object within the image or the actual position of a product on a conveyor belt. In a manufacturing environment this works very well when the product type and size is known. As when the height of the product is a fixed value in XY directions as long as the product outline can be recognized and measured. In a scenario where multiple product types on a conveyor belt are presented to the camera, this may be

a problem for traditional 2D systems. If a product height is not expected and cannot be assumed, the system will fail. *A 3D industrial vision system can extrapolate a pixel's position, not just in the X and Y direction, but also in the Z direction.*

3D industrial vision is achieved using a variety of techniques that include, but are not limited to stereo vision, point clouds, or 3D triangulation. Taking stereo vision as an example, this works in the same way that the human brain does, and in fact any animal with two eyes. Images from each eye are processed by the brain and the difference in the images caused by the displacement between the two eyes is used by the brain to give us perspective. This is critical when judging distances.

3D stereo industrial vision uses two cameras in an identical way. The software reads both images and can compare the differences between the two images. If the cameras are calibrated so that the relative position between each camera is known, then the vertical position (Z) of an object can be measured. In computing terms, this takes more processing time than an X and Y measurement. But with modern, multicore processors now ubiquitous, 3D industrial vision is no longer limited by processing time, meaning “real-time” systems can be improved with 3D industrial vision.

An obvious real-world benefit of this is 3D robot guidance and taking the initial example of known product dimensions on a conveyor belt. A 3D robot guidance system will deal with product variants even when the next product type and size is unknown. With a competent 3D vision system, the robot will not only receive X, Y, and Z data but also the corresponding roll, pitch, and yaw angle of each pixel in the combined image.

Automated Inspection

With the increasing demands of consumers, from a quality and quantity perspective, the use of automated inspection is now one of the key technology areas in the manufacturing industry. Traditionally a manufacturing process has a human element at the inspection stage where products are being visually inspected. This often involves handling the product. But due to manufacturing speeds, the inspection is on a subset of the entire production run. However, today's markets demand higher production throughput with guaranteed quality. Hence this is where the importance of automated inspection is realized.

With the use of industrial cameras, illumination, and optics, a high-quality image of the product can be captured. This can be from a few images a

second to hundreds of images per second depending on the demands of the manufacturing process. Once the image has been captured it is processed to include tasks such as product identification, surface/finish inspection, measurement/dimension inspection, presence/absence checks, barcode reading, character recognition, and so on. There is also an emerging area of vision that deals with 3D data analysis critical for processes that require robotic control such as pick and place.

Furthermore, combining systems can be developed, which not only inspect the product but also handle the product. They are based on industrial cameras for grabbing images and robust image processing software for the inspection. It is essential for manufacturing processes to automate to remain competitive and to manage with the demands of the worldwide consumer base. For these reasons alone, automated inspection is a key technology now and into the future.

Robotic Guidance

Industrial robots have been around for some fifty-plus years with the first officially recognized device being created in 1954. It wasn't until 1960, however, that the first robot company was formed. All robot guidance was by control systems with known coordinates. Their accuracy was highly critical with any deviation in either product variation or position errors causing major headaches for the end customer. This state of affairs continued for some time until at least 1983 when the first commercial vision systems became available.

Vision brings benefits to many sectors but robotics is one of the big winners for many reasons. One key point is the fact that it can be seen as an enabling technology in robot guidance. Robots are good at repetitive tasks but they cannot accommodate changing parameters, so when a location changes or a product is not where it is supposed to be, the robot system will fail. Industrial vision enables robots to “see” an object and calculate its X and Y position, with X in relation to the robot picking arm. But it also enables the robot to “see” the correct placing position. More recently, robot guidance has evolved with 3D industrial vision. So a third coordinate is also available, typically the height of an object. There is a steadily increasing range of smart cameras with the list of established image sensors and software packages, there are robot guidance systems for any application. With the advent of low-cost multicore processors more can be done.

3D Imaging

3D cinema films, 3D TV, and 3D gaming consoles are all familiar concepts in the world at large, but now 3D industrial vision imaging is increasingly having a major impact in a wide range of industries and applications from volumetric measurements, to inspection for packaging and robot vision.



FIGURE 2.2. 3D imaging.

The biggest challenge for 3D industrial vision imaging is time. Creating complex 3D images is computationally intensive and therefore time consuming. So it has been the emergence of processors capable of handling the computational overhead required at production line speeds that has been the key to establishing true 3D measurement techniques. 3D imaging is processor intensive. It is important to be able to assess whether an application really needs 3D measurements or whether conventional 2D imaging is more appropriate. Looking at the component in Figure 2.2, if one wants to measure the inner or outer diameter, 2D imaging is more than adequate. But if they want to be able to measure the defect in the surface, 3D imaging is needed. In the same way, using 3D robot vision to pick unordered parts enables manufacturers to save a lot of time and resources shifting or organizing parts in the manufacturing process. A number of different 3D imaging techniques have evolved with different capabilities, and there is plenty of choice of components and systems from different suppliers.

3D Imaging Methods

Laser profiling

Laser profiling using triangulation is one of the most popular 3D techniques and is shown in Figure 2.3. The object to be measured passes through a line of laser light and a camera mounted at a known angle to the laser records the resulting changing profile of the laser line. These

3D profiles deliver high measurement resolution with good measurement range. They produce a point cloud that when projected onto a designated plane creates a depth map that is conveniently analyzed using well-known 2D vision tools like blob analysis, pattern recognition, and optical character recognition. This technique relies on the object moving relative to the laser line so this configuration is particularly popular on production and packing lines where the product moves on a conveyor. The system can be configured using individual laser sources and cameras, or integrated systems where the source and camera are housed in a single enclosure. Care must be taken to avoid shadowing, where higher regions of the object block the view of the laser line so that data from the structures behind cannot be obtained. One solution is the use of several cameras that track the laser line from different angles and then merge the different data sets to a single height profile using sophisticated software tools.

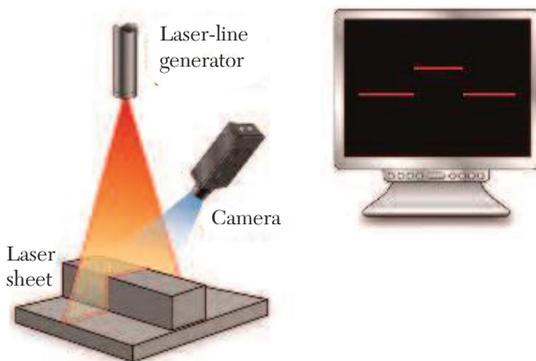


FIGURE 2.3. Laser profiling 3D imaging method.

Stereo imaging

Another common 3D method mimics nature by using a binocular stereo set-up where two cameras are used to record 2D images of an object. A 3D image can then be calculated using triangulation. This technology also allows for movement of the objects to be measured during recording. A random static illumination pattern can be used to add arbitrary texture to plain surfaces and objects that do not have the natural edges (texture) information which the stereo reconstruction algorithms require. This technology has proven very successful in applications such as volumetric measurements and robot bin-picking. Some systems are available that utilize line-scan cameras instead of area-scan cameras and are particularly useful for fast-moving objects or web applications. Photometric stereo uses a number of images to reconstruct the

object surface. Here a single camera and the object are fixed, while the scene is illuminated from different known orientations taken in consecutive images. This method gives only relative height measurements, making it an excellent choice for 3D surface inspection.

Fringe projection

Light stripe projection requires static objects. Here, the whole surface of the sample is acquired at once by projecting a stripe pattern to the surface, typically at an angle of 30° and recording the resulting image with a camera perpendicular to the surface as shown in Figure 2.4. The large number of points acquired simultaneously gives height resolution up to two orders of magnitude better than with laser profiling. The measuring area can be scaled from less than a millimeter up to more than one meter so suits small as well as large samples. *Time of flight cameras measure the time taken for a light pulse to reach the object and return for each image point.*

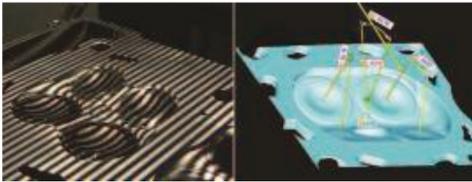


FIGURE 2.4. Fringe projection.

For more details refer Section 9.3, 3D imaging technologies.

3D Inspection

3D imaging has opened up a host of opportunities for manufacturing industry. The ability to make real-time measurements in the X, Y, and Z axis at production line speeds not only allows volumes of product to be calculated and defects to be detected. It also allows pass/fail decisions to be made on far more parameters than would be possible for traditional 2D measurements. 3D imaging provide automated inspection solutions for both simple and complex objects. It gives the user all of the traditional benefits associated with 2D inspection, such as reduced reliance on manpower for quality control; availability of live production data for monitoring systems; improved consistency of product; increased throughput and reduced wastage, as well as 100% inspection. Decisions can be made on product shape, proportions and even surface quality (indentations scratches, dents, etc.). The use of 3D matching tools enables 3D models to be compared to a known 3D or “golden” template for product verification.

3D inspection has applications in industries as diverse as food and beverage, pharmaceutical, automotive, packaging, electronics, transport, logistics, and many more. In the food industry, 3D measurements have been utilized for applications, such as portion control and potato sorting by shape. They have also been used extensively in the baking industry on products such as pizza, pies, bread, pastries, cakes, and cookies to check for shape, size, edge defects, and thickness. Assembly verification, especially of metal or plastic parts is critical in many industries and especially in the automotive industry. For example checking the height of components can be used to verify assemblies such as bearings, and 3D matching can confirm the surface integrity of parts. In final vehicle assembly, making gap and flush measurements on automotive panels allows correction and even rejection of the vehicle if the panels are badly misaligned.

In the pharmaceutical industry (Figure 2.5), 3D inspection can be used to detect shape defects in tablets including chips as well as reading embossed details on the tablet. Tablets can be checked in blister packs. 3D imaging can recognize grey products in aluminum foil as well as deformations and tablet fragments. Height measurement allows the detection of tablet capping and upright tablets. There are a host of applications for 3D inspection in general packaging applications. Some of them are:

- i. measuring the height of items in the packaging before it is closed to make sure that it is not too high,
- ii. checking that final packages are properly closed,
- iii. ensuring that there are flaps sticking out,
- iv. checking for dents in the packaging, and
- v. checking rims of foil lids on containers for defects in the surface that would affect seal integrity.

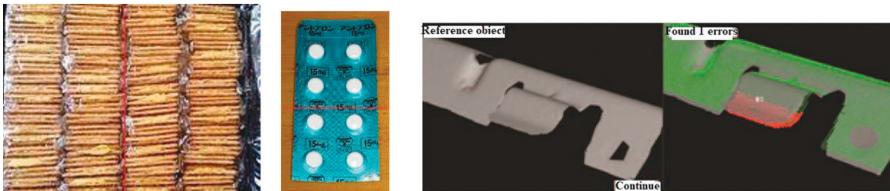


FIGURE 2.5. 3D inspection for volume of product and defects.

3D Processing

Making measurements

The major imaging toolkits that are available from a number of different vendors, offer a multitude of 3D measurement, image manipulation, and presentation capabilities. Vision tools are available for registration and triangulation of point clouds, calculation of features like shape and volume, segmentation of point clouds by cutting planes, and many more.

It is possible to make a 3D surface comparison between the expected and measured shape of a 3D object surface. “Golden template” matching is also possible in 3D with deviations between the template and test part



FIGURE 2.6. 3D processing for measurements.

identified in real time using real 3D point clouds and automatically adjusted for variations in orientation in all 3 axis. With 3D “smart” cameras, however, acquisition, measurement, decision, and control are all performed within the unit, although data can be output for further processing, if required. The 3D processing for measurement is shown in Figure 2.6.

Calibration

Many 3D applications can work reliably with noncalibrated images, while others do need calibrated images and measurement data. The easiest calibration setup comes using 3D smart cameras, where the laser, camera, and lens are located in a single housing. These systems are precision factory aligned to provide consistent, reliable measurements in real-world coordinates, and require only minor adjustments. For systems where the components are mounted independently, calibration involves moving a known object accurately through the field of view using a positioning stage. From this, the system can build a lookup table for converting xyz pixel values to real-world coordinates.

3D Robot Vision

Automation is a key factor in improving productivity and competitiveness in world markets. The use of 3D vision to guide robots’ pick and place is key in maximizing this competitiveness. In the automotive and pharmaceutical

industries 100% inspection is critical. But using 3D robot vision to pick unordered parts enables manufacturers to save a lot of time and resources shifting. It helps in organizing parts in the manufacturing process or in feeding robots and machines with parts. The challenge lies in acquiring images in 3D, building a mathematical model, and analyzing the position of something in 3D space and then transmitting 3D picking coordinates to a robot, all in just a few seconds to meet the cycle time of the robot and avoid it having to wait for the next set of coordinates. Fortunately, complex 3D images do not necessarily have to be created to achieve this feat. It is possible to do this using stereo vision imaging techniques, where features are extracted from 2D images that are calibrated in 3D.

As a rule of thumb, if there are a minimum of four recognizable features on an object, it is possible to create 3D measurements of the object and therefore generate the X,Y, and Z coordinates of any part of the object, with a level of accuracy that allows the robot to grip it without causing any damage. If, however, there are not enough features, or no features at all that can be used for the 3D calibration, features can be “created” using laser lines or dots to illuminate the area. A good example of this is 3D depalletizing of sacks (Figure 2.7), which could contain anything from concrete to grain or



FIGURE 2.7. 3D depalletizing

the lens and camera calibration that can achieve subpixel measurements.

As the sacks are rather featureless, the whole pallet is illuminated with lasers and the laser lines located in 2D images. The sacks are also recognized in the 2D images and all the information is combined to get 3D picking data all well within the cycle time of the robot. So most of the work is done in 2D, with far fewer pixels to process, yet a high level of accuracy is maintained due to

High-speed Imaging

Most of us have had some experience of high-speed imaging. Classic images such as the corona formed in a liquid when a droplet hits the surface or the rupturing of a balloon’s fabric as it bursts are well known. High-speed imaging is used extensively in filmmaking and in TV programs such as sports coverage.

However, high-speed imaging is a diverse technology that also has a host of industrial applications. Examples include sports biomechanics

analysis and performance evaluation; vehicle crash testing including impact analysis (Figure 2.8) and airbag deployment and ballistics analysis in the military and in mining. In the manufacturing industry inspection applications on high-speed production lines include product, packaging and label defect identification, label reading and verification for codes and human readable data, and web-based product inspection (Figure 2.9). Process and machinery diagnostics are another important application. Even minimal discrepancies in high-speed process machinery mechanisms can cause an entire production line to come to a standstill. Intermittent failures can be even more difficult to troubleshoot. High-speed diagnostic systems can record image sequences both before and after an event. Slow motion review allows causes of failures to be identified. It will make any necessary adjustments to avoid failure.

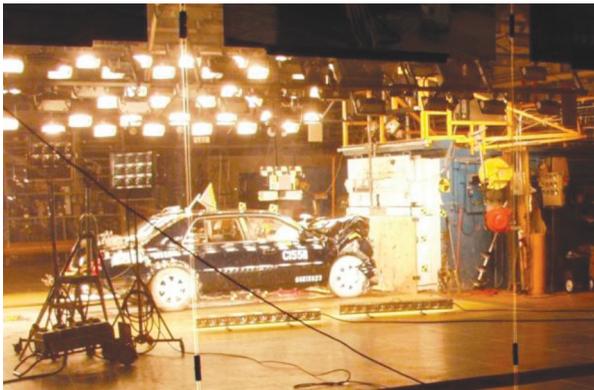


FIGURE 2.8. Vehicle crash testing for high-speed imaging.



FIGURE 2.9. Label defect identify by high-speed imaging.

High-speed Cameras

High-speed cameras are capable of short exposure times and fast frame rates. Short exposure times are needed when imaging a fast-moving object. Typically, the object moves less than one pixel during the exposure to avoid motion blur as shown in Figure 2.10. However when a series of objects are moving past the camera, high frame rates will also be required to ensure that each item is imaged on a successive frame for analysis. Many manufacturers would suggest that “high-speed” cameras operate at frame rates in excess of 200 frames/second, although frame rates of thousands of frames/second may be required for troubleshooting manufacturing process

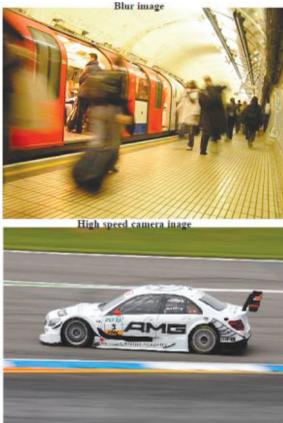


FIGURE 2.10. Blur image and high-speed camera image.

problems. However, effective high-speed imaging is a function of much more than just the frame rate and exposure time of the camera.

Exposure time is the amount of time that light is allowed to fall onto the camera sensor. Longer exposure times allow the sensor to gather more light, but this leads to more noise being generated on the sensor. Standard vision cameras mostly specify the maximum exposure time to avoid noise becoming an issue. Short exposure times are needed when imaging a fast-moving scene to avoid motion blur and typically the exposure time should be short enough so that the object moves by less than 1 pixel. Consider an object moving at 100mm/sec with an area of 100mm x 100mm to be imaged. Using a camera with a resolution of 1K x 1K pixels, each pixel will be imaging an area of 0.1mm. In 1 second, the object will have moved by 1000 pixels which will require a camera capable of exposure times of 1/1000th second or faster to avoid motion blur.

The frame rate, or number of complete images from an area scan camera that can be output in a particular time is important. For example, a production line where objects are passing by at a rate of 20 units per second will require a camera capable of capturing 20 discrete frames per second. Frame rates are quoted at the full resolution of the particular camera sensor, but many cameras offer the ability to partially scan the sensor or sample a discrete portion of the sensor, allowing much higher frame rates for that area. This can be useful if the full frame is not required for imaging. A technique known as “binning” can also increase frame rates. Binning combines the output of adjacent pixels on a sensor and this also results in increased sensitivity and S/N, but decreased spatial resolution.

Line Scan Imaging

Line scan cameras are used extensively in high-speed imaging. In general, a single line of pixels is scanned at high speed and the frame is built up by the motion of the object past the camera. The size of the object to be imaged and the speed of movement determines the line rate required in the camera. Line scan cameras have shorter exposure times and therefore require greater illumination levels.

Light emitting diodes (LEDs) are a popular form of illumination for machine vision applications, offering a good deal of control. They can be readily pulsed, or strobed to capture images of objects moving past the camera at high speeds. Strobing needs to be synchronized with the objects to be inspected so the camera is triggered at the same moment as the pulse of light. The short exposure times required for high-speed imaging mean that high light intensities are required. It is possible to dramatically increase the LED intensity over short exposure times by temporarily increasing the current beyond the rated maximum using lighting controllers. However the LED must be allowed to cool between pulses to avoid heat damage. Lighting controllers can provide fine adjustment of the pulse timing, which is often more flexible than the camera's timing. The camera can be then set for a longer exposure time and the light pulsed on for a short time to "freeze" the motion.

High-speed imaging requires that the exposure of the camera happens exactly when the object is in the correct position. Initiating the start of an exposure at a particular time is called triggering. If a camera is free running the position of the moving object could be anywhere in each captures frame or even completely absent from some frames. Triggering delivers image acquisition at a precise time. The frequency of a trigger should not exceed the maximum frame rate of the camera to avoid over triggering. This also means that the exposure time cannot be greater than the inverse value of the image sequence. The exposure is generally triggered by an external source such as a PLC with a simple optical sensor often used to detect when the object is in the correct position (Figure 2.11). Precise triggering is very important for high-speed imaging. In very high-speed applications great care must be taken to assess and reduce all of the factors that can influence any delays from initiating a signal in the sensor in order to ensure the required image is acquired. These factors could include opto-isolators in cameras as well as the latency and jitter within the imaging hardware.

Capture and Storage

High frame rates and high spatial resolution generate high volumes of data for processing. Image data are generally transferred directly to a PC's system memory or hard disk. This relies on an appropriate interface speed between



FIGURE 2.11. Camera in correct object position.

the camera and the computer. There are a number of vision image data transfer standards such as GigE Vision, Camera Link, Camera Link HS, USB 3 Vision, and CoaXPress that generally offer a trade-off between data transfer rates and allowable distance between camera and PC. One of these interfaces offer an acceptable data transfer rate for the application and long sequences are required. The alternative is to have the image recording memory within the camera itself. This increases data throughput significantly since images are held in the camera without any need for transmission while recording. However, the amount of onboard memory is significantly less than that of PC hard drive. Only relatively short sequences can be recorded in the camera.

High-Speed Inspection for Product Defects

Inspection of products for defects is an important aspect of high-speed inspection applications (Figure 2.12). The key areas are verification, measurement, and flaw detection. Verification ensures that a product, assembly or package has been correctly produced. Applications range from simple checks such as ensuring that all the caps are in place on a bottling line or components such as clips, screws, springs, and parts are in place. It also checks that seals and tamperproof bands are in the correct position. No product has been trapped in a packaging seal. Other verification examples include solder joints, molded parts, assembly, and blister packs. The accurate measurement of component dimensions are necessary to ensure that they are within predefined manufacturing tolerances. It is also extremely important from the point of quality control for the end user, and from the point of monitoring the manufacturing process to keep it within tolerance and therefore minimize waste. Products or components also need to be inspected for flaws such as contamination, scratches, cracks, discoloration, textural changes, and so on.



FIGURE 2.12. Product defect inspection.

Labels and Marking

Vision systems are also used extensively to check labels. Code reading tasks can include fully validated character recognition, character verification, robust 2D data matrix handling and grading. Code verification systems can help eliminate variables that affect the readability of a code. It confirms the printing is good from the start. Reading tasks could include simultaneously

detecting changing batch code orientation, differing quantities, and location of characters on a label. Vision is also closely involved in the implementation of safety features to verify the authenticity. It identifies individual packs of medication rather than just batches. Vision systems are also used for code reading of directly marked components for product identification and traceability, especially in safety critical industries such as aerospace and automotive. Tracking a component and all the processes it has gone through, from manufacturing, assembly, and right through to end user requirements for spare parts replacement helps compliance with industry guidelines and standards and quality assurance used in the manufacturing supply chain.

Web Inspection

High-speed line scan imaging is used extensively for industrial web inspection applications. It's used on fast-moving continuous product in a variety of industries including print manufacturing, paper processing, steel-plate manufacturing, glass tape, and textiles. Applications generally involve the identification and classification of faults and defects in these materials. In the food and packaging industry, high-speed inspection systems have been developed to inspect laser perforated holes in flexible modified atmosphere packaging films. The vision system can locate and measure hole sizes ranging from 30–120 μm diameter (approximately the diameter of a human hair) on a web running at a speed of 350m/min.

High-Speed Troubleshooting

High-speed vision systems can significantly improve the accuracy of diagnostic analysis and maintenance operations in industrial manufacturing applications. Users can record and review a high-speed sequence, either frame-by-frame or using slow-motion playback to allow perfect machine setup and system synchronization. Alternatively the system can be used as a “watchdog” by continuously monitoring a process and waiting for a predefined image trigger to occur. These troubleshooting systems are generally portable and can be used in a wide range of manufacturing applications. These include bottling lines, packaging manufacture, food production lines, plastic container manufacture, pharmaceutical packaging, component manufacturing, paper manufacture, and printing.

Troubleshooting applications can require short exposure times so high intensity illumination is required. Camera frame rates of thousands of frames/second generate a lot of data at high speed, especially if they have

high spatial resolution. Rather than try to transfer this data to a PC, in-built high-speed ring buffers may be used for image recording. Image sequences can be replayed in slow motion on self-contained image displays after the event has been recorded or transferred to the hard disk of a PC for later review. Image sequences can be recorded in standard video file format.

Specialist triggering is used for troubleshooting. Generally it is important to see what is happening both before and after the trigger event. The system continuously records into a ring buffer. Once it is completed, the system starts overwriting the first records. Once a trigger event has occurred, the system records until the ring buffer is filled up and stops. In this way both pre- and postevent recording information is acquired. Sequences can also be triggered by monitoring either changes in intensity or movement in the image. The camera triggers itself to send an image or sequence. It removes the need to generate a trigger using hardware. This is particularly useful for capturing intermittent or random events.

Line Scan Technology

Materials produced in continuous rolls (web) or sheet, such as paper, textiles, film, foil, plastics, metals, glass, or coatings are generally inspected using line scan technology to detect and identify defects in order to avoid defective material being sent to customers or to add value to downstream processes (as shown in Figure 2.13). Like so many areas of industrial vision camera technology, line scan imaging has seen some significant developments in recent years. It not only benefits web inspection but other line scan imaging applications as well.



FIGURE 2.13. Line scan imaging for paper.

Line scan technology involves building up an image, one line at a time, using a linear sensor. For web inspection and many other industrial vision applications the object passes under the sensor, typically on a conveyor belt. Although linear sensors have similar pixel sizes to the sensors used in area

scan cameras, the line lengths can be much greater. Instead of the 1-2K width typical in most megapixel area scan sensors, a line scan sensor can have up to 16K pixels. This means that for a given field of view, a line scan camera will provide a far higher resolution. The line scan technology makes it possible to capture images of wide objects at a single pass. High scan speeds for linear arrays means that the amount of light falling on individual pixels is often lower than in area scan applications. Hence, consideration must be given to overcome this.

Both CCD and CMOS linear sensors have been used in line scan cameras for many years. But developments in CMOS technology driven by the mobile phone market have also led to significant benefits in industrial imaging sensors. Recent developments have included the introduction of 16K pixel sensors, simultaneous RGB and NIR imaging, higher line speeds, larger pixel variants for enhanced sensitivity, lower cost systems, enhanced software, and the use of the newer data transmission standards such as CameraLink HS and CoaXPress. Another interesting development has been the emergence of contact image sensors (previously found in photocopiers and scanners) as a viable alternative to line scan cameras for industrial applications. Also, some area scan cameras offer a line trigger mode for use in some line scan applications.

CMOS sensor developments have allowed increases in pixel resolution to 16K and line speeds up to 140 kHz. More pixels and higher line speeds generate more data. A 16K sensor operating at 120 kHz line rate produces 2 GBytes/s of data which necessitates camera/frame grabber combinations utilizing the new generation of camera interfaces such as CameraLink HS and CoaXPress. Pixel size and number determine the length of the sensor. By making use of binning techniques and FGPA, resolution and effective pixel size can be adjusted in a single camera to optimize between resolution and sensitivity. For wide-web inspection applications multiple line scan cameras may be needed to cover the entire width.

Line scan cameras generally have short pixel exposure times. It require more illumination than area scan cameras because higher line rates bring even shorter pixel exposure times. Sensors frequently use dual line technology with two rows of pixels scanning each line on the sample, improving the S/N. Time delay integration sensors offer multiple integration stages, giving substantial S/N enhancement. Typically, line scan pixel sizes range from 3.5 to 14 μm square, but a new range of single line scan cameras features 20 μm square pixels, with a 2K CMOS sensor capable of operating

up to 80 kHz. The larger pixel size gives better signal to noise ratio for a given exposure level, and higher line speeds than smaller pixel systems at the same exposure level.

Three-sensor color imaging in line scan cameras allows collection of independent RGB information. Prism systems collect light from a single line and split it spectrally onto three sensors. Trilinear sensors collect the RGB components from three separate lines. These lines need to be physically separated to accommodate the necessary electronics structure. A cost-effective alternative is a bilinear detector with no line gap that uses color filters similar to the Bayer arrangement used in area scan cameras. In another recent development, quadlinear and prism-based four sensor cameras are now available to provide NIR outputs as well as RGB for multispectral imaging. This enhances imaging possibilities for a wide range of applications; including print, bank note inspection, electronics manufacturing, food-sorting, and material sorting.

Contact Image Sensors

Contact image sensors are an interesting alternative to line scan cameras for the inspection of flat products such as textiles, glass foils, wood, and other web-like materials for defects. Other applications include PCB, solder paste, and packaging inspections as well as print inspection and high-end document scanning. They offer high data rates as well as high sensitivity and simple set up. Contact image sensors use the same concept as used in fax machines and desktop scanners. They include a sensor and lens with pixels being mapped 1:1 to the object, with a working distance from a few mm up to around 12mm. This means the sensor has to be as big as the item being imaged, but has the advantage that distortion found in traditional lens/line scan camera combinations is removed. They are available with and without integral LED light sources.

The sensor head generally features a lens array using gradient index rod-lenses. Because these lenses are graded, they do not suffer from any variation in their refractive index. Each individual lens captures an image of a very small region of the target. A clear, sharp quasitelecentric image is produced along the narrow line of the sensor head, with remarkable image uniformity. This is particularly important in applications such as high value print inspections such as on banknotes, passports, and so on, which may contain holograms. These are particularly susceptible to the angle of light entering them, so the virtually telecentric structure of the contact image sensor is well suited to these applications.

Compact image sensors can be combined to offer extended lengths and provide similar features to line scan cameras in terms of dark current, peak response nonuniformity, and dynamic range, but without the trade-offs concerning spatial resolution and light efficiency. Contact image sensor heads can use CMOS or CCD sensors as detectors. There is a choice of pixel layouts from monochrome sensors to color versions using alternating colored pixels or trilinear sensors. Resolutions up to 600 dpi are available with scan speeds up to 1.8 m/s for monochrome sensors. Image data output is generally provided via standard industrial CameraLink interfaces.

The shorter pixel exposure times for line scan cameras compared to area scan cameras generally means that line scan applications require a greater level of illumination. Since line scan applications only require imaging of one line on the sample, line light illumination systems are usually used. High intensity LED line lights provide long lifetimes, and consistent, stable, light output along the entire length of the light. Line lights are available for both front and back lighting, with bright field and dark field illumination being the most popular choices for front illumination depending on the material being imaged. LEDs also offer a choice of wavelengths. The light unit can effectively be made of any length and any intensity. However the higher the intensity, the more heat and heat sinking are needed to dissipate this. The use of enhanced sensitivity sensors helps reduce the intensity of lighting required.

A line scan image is produced from the relative movement of the sample and the camera. Synchronization of the movement between the object and camera is required to ensure that there is no distortion in the image. This is usually achieved by setting up a line trigger signal from an encoder signal from the sample movement method (typically a conveyor belt), to ensure that the scanned lines are synchronous with the movement of the object. The camera will collect light between these trigger signals, so if the movement speed varies the image brightness will also vary. In order to ensure constant image brightness exposure control is needed. This can either be set up on the camera itself or by controlling the illumination intensity.

Lenses

The sensor length is a function of the number and size of the pixels it contains, the more pixels there are and the larger they are, the longer the sensor will be and this has a direct influence on the size of the camera

lens (Figure 2.14). For sensors with a line length of more than 20 mm, the use of traditional C-mount lenses becomes problematic, since there is a significantly different viewing angle at the ends of the sensor. The solution is to use F-mount lenses with a larger image circle diameter, but this adds to



FIGURE 2.14. Camera lens.

the cost of the optics. Alternatively, a sensor with smaller pixels, and hence a shorter line length could be used, but this may require increased illumination. A uniform viewing angle can be obtained using telecentric lenses, but these again add size and cost to

the installation. Thus careful thought must be given to the imaging and resolution requirements to get the optimum choice of sensor and lens.

Image Processing

The major image processing toolkits provide all of the tools necessary for inspecting continuous webs. These offer the facilities to find and classify defects such as cracks, tears, knots, holes, and to find color variations or perform critical dimensional measurements at the high speeds needed. Other capabilities include code reading, robot guidance for cutting, trimming, or shaping and communication with other third-party equipment such as PLCs, HMIs (Human Machine Interfaces), and remote storage.

Line Scan Inspection

Line scan imaging is used in continuous web inspection systems to perform 100% inspection to detect defects such as dirt, debris, pinholes, roll-marks, holes, cracks, tears, and scratches on materials such as paper, foils, films, nonwovens, metals, and so on. Web inspection (Figure 2.15) is possible for web materials with a uniform or textured, glossy or matte surface or transparent materials. It is generally carried out on wide rolls of material (for example some 8m wide) and at high speeds, so it is frequently



FIGURE 2.15. Web inspection.

necessary to use multiple line scan cameras to cover the entire width. Depending on the particular material, incident or transmitted illumination can be used. Defect classification is based on size, range, and contrast type. For example a contaminant may show up dark, and a hole bright in transmission. Typically defects down to around $50\mu\text{m}$ can be detected. Because the material is on a continuous roll, it is not possible to carry out instant rejection system when a defect is detected. Instead, a roll map is produced which

shows the defect type and location on the roll so that it can be identified and removed when the material is actually used.

It is possible for numerous defects to occur during the printing processes, such as ink spot marks, embossing defects, misregistered colors, and color variations. Applying 100% print inspection on materials as varied as banknotes, and pharmaceutical and food packaging is challenging, and frequently makes use of line scan cameras and software utilizing the “golden template” or an “intelligent template” model to compare the item under test with a standard image and measure the differences. Other applications include continuous verification and/or quality inspection of numbered print and inspection of symbols and labels on web, sheet, or single documents.

Line scan is an excellent way of imaging cylindrical components as shown in Figure 2.16. A line scan camera records the same position across the whole cylinder, and as the cylinder rotates an “unwrapped” image of the entire surface is generated without the distortion that would result from imaging a curved surface with an area scan camera. This technique allows labels to be unwrapped, allowing the reading of codes and human readable characters. It also allows the inspection of surface for defects such as pits, scratches, holes, and so forth.



FIGURE 2.16. Line scan for cylinder components.

Inspection systems equipped with line scan cameras can be used for sorting of a large variety of objects, such as food, waste, mining products, mail, parcels, and so on. Typically these objects are moving past a camera system on a conveyor. Line scan systems can also be used for sorting free-falling products such as rice, vegetables, postal, molten glass, steel, pharmaceutical products, and rocks, which cascade past one or more cameras like a curtain. Since the “length” of the image produced by a line scan camera is determined by the number of lines recorded for each image, very high resolution images can be produced. This is particularly useful for inspection of a host of products including flat panel displays, solar panels, PCBs, silicon wafers, and so on.

End of line inspection is one of the most important uses of vision in the manufacturing industry with applications on both manufacturing and packaging lines. It is shown in Figure 2.17. The combination of vision

technology developments and the emergence of specialist vision systems integrators make the use of vision much more practicable. In addition, pressure from major players in industry and legislative requirements in the pharmaceutical industry are making the use of vision an essential requirement.



FIGURE 2.17. End of line inspection.

Major players in a number of industries can impose stringent quality requirements on their suppliers. In the food industry, supermarkets use a significant amount of power over their suppliers. Margins are squeezed, and penalties can be imposed on suppliers whose products or packaging do not meet their demanding standards. The use of vision technology can help meet such demands. The automotive industry is one of the world's most cost-sensitive industries and also one of the most demanding in terms of product quality and dislike to component failures. Both manufacturers and component suppliers rely increasingly on leading-edge vision technology to validate complex assembly processes. The use of vision also helps the industry's changing approach to quality inspection that now concentrates more on differentiating between critical and noncritical defects than those that affect the functionality of the object.

Tracking and Traceability

Vision plays an important role in reading unique identifiers in the form of 1D or 2D codes, alpha-numeric, or even braille for tracking and tracing applications in industries as diverse as aerospace, automotive, food, healthcare, and pharmaceutical. Human readable on-pack data, such as batch, lot numbers, best before or expiration dates are also critical for products such as food, pharmaceutical, medical devices, and cosmetics. In the automotive industry, data management allows manufacturers to optimize processes and perform product traceability. Having the appropriate data

on vehicles and related parts can also help to reduce costs and makes it possible to accurately and promptly respond to quality assurance and recall problems. The pharmaceutical industry will require individual packs of medicines to carry a unique, machine readable identifier which will provide traceability from the point of sale back to manufacture as shown in Figure 2.18.



FIGURE 2.18. Tracing.

Serialization

EU regulators are introducing a new era of pharmaceutical manufacturing and distribution compliance. The pharmaceutical products of individual packs medicines will carry a unique, machine-readable identifier. This item-level serialization will provide traceability of a pack from the point of sale back to manufacturer so that its authenticity can be checked at any point in the supply chain. It is shown in Figure 2.19. Industrial vision will have a key role to play in this. Serialization means that packs must be labeled correctly and the labels verified by industrial vision, and all data passed upstream to the appropriate place, and all at production-line speeds. The process will generate huge quantities of data compared to present levels and product data will need to be uploaded to a national or international database against which product IDs can be verified. The challenges for vision systems used in serialization applications are primarily high-speed inspection of codes and transferring data and handshaking with control hardware on the shop floor.

Direct Part Marking

For some time, traceability and quality control of parts particularly in the automotive and aerospace industries has been carried out using 2-D Data Matrix direct part marked (DPM) codes as in Figure 2.20. These are normally laser-etched onto the component, providing an almost indestructible code to survive a life that a traditional barcoded label would



FIGURE 2.19. Serialization.



FIGURE 2.20. DPM codes.

not survive. Using vision systems to read these codes, key components such as differential gears, clutches, transmission case, housings, valve bodies, and so on, can be traced throughout the production process. In addition, engine components, such as pistons, cylinder heads, engine block, CAM shaft, and crank shaft can be traced throughout the manufacturing and distribution processes.

In spite of the obvious benefits of this “cradle to grave” tracking, factors such as shiny surfaces, curved surfaces, rough finishes, and dirt or oil contamination can lead to unreliability and low-read rates. However, recent enhancements in code-reading cameras and lighting, with economies of scale driving down pricing, means that direct-part marking and identification is now becoming a more cost-effective and robust technology. Since industrial vision can make simple or complex repetitive measurements accurately, at speed and objectively, it is used in a wide range of end of line inspections in a host of different industries.

Product Conformity

This is perhaps the most traditional application where the final product must be inspected as part of the quality control procedure. Typically, this involves checking parameters such as shape, size, volume, geometry, surface finish, color, and so on, to ensure that the final product meets the required specification. Products that fail the inspection will be rejected, and can possibly be reworked depending on the application. The speed and accuracy offered by the latest vision technology means that in many applications, 100% inspection can be carried out, and the quality of the final product can be controlled to demanding standards. Almost any product manufactured on a production line is a potential candidate for this type of inspection. Ensuring that packaging is “right” is of paramount importance, ranging



FIGURE 2.21. Product conformity.

from consistency in colors and logo positioning to the integrity of packaging enclosures for product purity and shelf-life as shown in Figure 2.21.

Typical applications include:

- Packaging defects, for example, rim damage on tins, straightness of bottle caps, presence of tamper-proof bands, correct application of foil seals
- Packaging contents checking, for example, fill levels, dimensional checking of end-of-line packaging to ensure inclusion of correct contents, positioning of product within packaging. Supermarkets, in particular, can impose exact requirements on suppliers with regards to package appearance.

Correct package labeling is critical for consumer safety, especially in the food, beverage, pharmaceutical, and medical industries. Ingredients must be listed accurately, together with nutritional information. Omitting a warning that a product could contain nuts could be catastrophic for a consumer allergic to nuts. Inserting the wrong patient information in a medicine packet could be equally catastrophic. Other applications include checking the presence or absence of labels, character recognition, and print verification. A huge array of products are tagged either by a stick-on label or by information printed directly onto the packaging, with information such as barcodes, lot details, and best before codes being the most common ones that need to be checked with total accuracy.

Vision-based code readers and optical character recognition systems are an essential requirement, given the potential variables in the codes themselves. These include:

- Regularity of code or label orientation on the product or package
- Regularity of the codes or characters themselves
- Possible damage to the code
- Contrast between the code or characters and the background

It not just enough to read these codes, but also to verify they have been printed with a quality that will allow for robust reading later in the supply chain.

Systems Integration Challenges

Projects involving the integration of cameras into existing production lines will generally require the combination of vision systems with ancillary equipment such as conveyors, product rejection mechanisms, pick and place robotics, as well as production control systems, or the provision of stand-alone inspection systems. This requires expertise in fields as diverse as mechanical design, mechanical handling and transport systems, software, electronics, robotics, control systems and factory networks and CAD. A dedicated user interface will also be required. In addition, it is important to have a good understanding of the specific requirements and standards required in different industries. This could range from environmental considerations such as hygiene and wash-down requirements in the food and beverage industries, to the need for part traceability and identification in safety critical industries such as aerospace industry, to the security and auditing requirements for validation in the healthcare and pharmaceutical industries.

The process begins by understanding the customer's unique requirements in order to develop proposals to meet the specific manufacturing needs in terms of performance, reliability, and adaptability. Typical factors to be considered could include the linear speed of the system, the number of parts per minute for inspection, the product spacing and orientation and whether they arrive singly or in an array. These latter factors are important for the reject process, which must be configured so that the correct item is rejected and that the system can be certain that the correct item has indeed been rejected. When these and all the other factors are assessed, a detailed project proposal can be prepared. Once this is accepted, there are a number of discrete stages for an integration project which would include proof of process.

The proof of process phase is crucial. This essentially allows a preliminary vision system to be designed, built, and tested on real-life samples under conditions as close to production line conditions as possible. This is where the key decisions about individual vision components and their compatibility to work in a system are evaluated. Timing is always a key issue, so the choice between an area scan or line scan camera may be influenced by the image processing and measurement time required, since image processing using a line scan camera cannot be completed until the object has passed by the sensor. Similarly, the use of a fast readout area scan camera may give the extra time needed for complex image processing.

Then there are the physical considerations mount a camera and lens. Once the proof of process is deemed satisfactory, the system can be scaled up to the production line environment. Proof of process is a relatively low-cost exercise, compared to the overall project. Ultimately the proof of process needs to prove the robustness of an inspection and be capable of identifying all necessary faults without producing false waste.

There are many ways in which vision technology can be used in end of line applications. These are frequently “smart” cameras that can be set up at the end of a production line by the customer’s production engineers. These are particularly appropriate for single inspection applications. Smart cameras combine image capture, processing, and measurement in a single housing and output the results from the analysis over industry standard connections. They can be used for high-volume component inspection, 1D and 2D code reading and verification, optical character recognition, and so on. For solely code reading applications, dedicated high-speed code readers also featuring integrated lighting, camera, processing, software, and communications are available. Where multiple inspections are required (for example, where the same object may need to be viewed from different directions), the use of multiple smart cameras may not be the most cost effective. Using multiple cameras controlled by a single PC may offer a better solution and these type of systems can generally be set up and installed with the help of the manufacturers or vision component distributors.

Challenging end of line inspection applications are generally handled by specialist vision system integrators. These are where the installation set up is complex, or a complete solution including product reconciliation, rejection and handling is required. Systems integrators will also provide the detailed documentation needed to support validation and auditing of equipment (essential in the healthcare and pharmaceutical industries), manuals, commissioning, training, and postinstallation support.

2.4 INDUSTRIAL VISION MEASUREMENT

Measurement is a main stay for automated inspection and has provided the platform for ever faster, more efficient, and more accurate quality control. In addition to preventing defective product reaching the customer, vision measurements can also be directly linked into statistical process control methods to improve product quality, reduce wastage, improve productivity, and streamline the process. Industrial vision does not examine

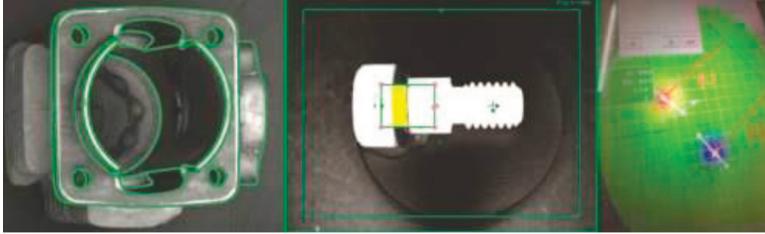


FIGURE 2.22. Vision measurements.

the object itself, measurements are made on the image of the object on the sensor as in Figure 2.22. All of the factors that contribute to the quality of that image must be optimized, so careful consideration must be given to every element of the industrial vision system, including lenses, illumination, camera type and resolution, image acquisition, measurement algorithms, as well as external factors such as vibrations, electromagnetic interference, and heat.

Measurements fall into three categories: 1D, 2D, and 3D. 1D measurements are typically used to obtain the positions, distances, or angles of edges that are measured along a line or an arc. 2D measurements provide length and width information and are used for a host of measurements including area, shape, perimeter, center of gravity, the quality of surface appearance, edge-based measurements, and the presence and location of features. Pattern matching of an object against a template is also an important part of the 2D armory. Reading and checking of characters and text, and decoding 1D or 2D codes is another key activity. The emergence of many affordable 3D measurement methods provide length, width, and height information, allowing the measurement of volume, shape, and surface

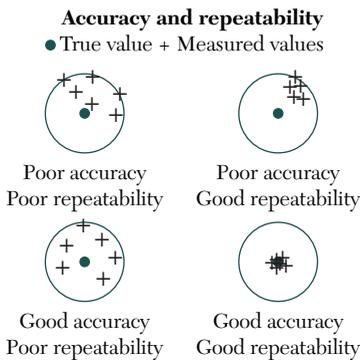


FIGURE 2.23. Accuracy and repeatability.

quality such as indentations, scratches, and dents as well as 3D shape matching.

Good accuracy and repeatability of vision-based measurements are of paramount importance. Accuracy is an indication of how close the actual measurement is to the true value. Repeatability shows the closeness of a number of repeated measurements. A group of measurements could have poor accuracy and repeatability, or good repeatability but poor accuracy, or good accuracy but poor

repeatability, as well as the desired combination of good accuracy and repeatability. Figure 2.23 shows the accuracy and repeatability.

Character Recognition, Code Reading, and Verification

Optical character recognition (OCR), verification (OCV), code reading, and verification are a major application area for industrial vision. Ensuring alphanumeric codes (e.g., lot details and best before information), 1D barcodes, 2D data matrix codes, and QR codes are correct can be critical for consumer safety and for product identification and traceability. Products can be tagged either by a stick-on label or by information printed directly onto the product or onto the packaging.

OCR tools use some pattern matching techniques such as pattern finding and contour detection since an alphanumeric character is simply a pattern that needs to be recognized (Figure 2.24). Contour-based tools work well on text with clear backgrounds and can accommodate variations in scale and rotation with no loss in speed. Contrast-based tools provide more robust detection in applications where the contrast is poor or changing. Industrial vision OCR algorithms need a fairly complete letter to decipher, especially if the text is structured. Once the character has been detected, OCR systems work by comparing a library of taught models to what is printed. The output from an OCR system is the alphanumeric string that has been read, for example, a use by date. Special consideration must be given for text that is written in a curved pattern.



FIGURE 2.24. Character recognition.

Pattern matching techniques are used to locate the code and extract the data. However, to improve reliability, barcodes, 2D data matrix and QR codes have a built-in redundancy so the code can still be successfully read even if it sustains a certain amount of damage in a production environment.

Verifying a barcode has been printed accurately is very different from simply reading the code itself. A good reading algorithm should be able to read poor quality codes, but a barcode verification algorithm should grade how well the code is printed. There are a number of verification standards that cover parameters such as symbol contrast, fixed pattern damage, and distortion. Each result is then graded accordingly. Similarly, OCV is a tool used to inspect the print quality and confirm its legibility. The output from an OCV system is usually a pass or fail signal based on whether the text string is correct, as well as the quality, contrast, and sharpness of the text.

Particularly for alpha numeric codes directly marked on a component, there can be challenges in acquiring a suitable image for reading and verification. This may be due to lack of contrast, a reflective surface, a curved surface, or some combinations of these. Even for codes written on packaging or labels there may be problems with contrast or reflections from shiny backgrounds. Therefore, just as in any other industrial vision application, correct illumination is of paramount importance.

Making Measurements

Industrial vision measurements are made in software. For vision systems utilizing a smart camera, the measurement software and measurement tools are built into the camera itself. For PC-based systems, there are essentially three main software categories that can be used with single or multiple cameras:

- Simple to use systems with graphical point and click interfaces often accessed via a web browser. These will feature a wide range of measurement tools that can be linked together to meet many measurement requirements.
- More sophisticated machine vision software that provides a much wider range of functionality and is frequently used by systems integrators to develop complete vision measurement solutions.
- Powerful programming libraries for development and implementation of vision solutions, primarily by vision experts. These feature a comprehensive collection of software tools and provide the ultimate flexibility for developing machine vision software applications from application feasibility, to prototyping, through to development and ultimately deployment.

In order to make actual measurements, pixel values must be converted into real-world values, which means that system calibration is required and the system must be set up to ensure that measurements can be made with the required accuracy, precision, and repeatability. For the best repeatability, all of the set up conditions for the vision system should be recorded. These include: exposure time, camera gain, white balance of the camera, light intensity settings (and strobe, if used), working distances and angles, and f-stop of the lens. A universal test chart can be used for quick and convenient system set-up and checking, for focus, system resolution, alignment, color balance. Geometric distortions from the lens can usually be corrected in software.

Special 3D calibration bodies with known reference surfaces and angles allow metric calibration in combination with special software packages. They can be used for the simultaneous calibration of one or more cameras. In addition to metric calibration a plane fit for alignment of 3D point clouds is possible. This is important for 3D matching and for easy processing of range map images.

It is important to check the accuracy and repeatability of a vision system. One way of doing this is to perform a series of 30 measurements of the same part with automatic or individual part feeding, and check the variation of the results measuring the same part compared to the allowed tolerance. If this is acceptable, it sets a benchmark for future measurements. Many machine vision systems offer extra statistical information, such as minimum, maximum, mean, standard deviation, C_p and C_{pK} (process capability) of measured values. The stability of the system can be checked by performing measurements with the same equipment at the same place and the same operator but at different times.

Pattern Matching

Pattern recognition is hugely important and most vision toolkits offer some type of pattern matching tool. Pattern matching can be used to localize a pattern in a large area, such as for pick and place applications where the camera identifies the object and tells the robot where it is. It can also be used for classification to decide which object is at a known location. Essentially pattern matching involves subtracting the image of a part under test from an image of a good part (“golden template”) to highlight any differences. Pattern matching techniques include normalized greyscale correlation, geometric-based search, binary chain-code-based search, decision-tree-based search, and neural-network-based search.

Important factors to consider are potential changes in lighting conditions and/or scale and rotation between the template and the parts under test. Some methods utilize a single image for the template, but decision tree and neural network methods produce the template from many images using a learning algorithm. More sophisticated approaches include those where complex appearance models of good examples can be created during the training phase in order to be able to allow for any natural variability in the parts.

3D Pattern Matching

Pattern matching in 3D imaging uses the same principle of comparing a 3D golden template image of a perfect sample with the 3D images of parts under test produced using real 3D point clouds. However, the alignment of both images is more complex since additional degrees of movement and rotation are possible. Automatic adjustment of position errors or tipping and tilts in all three axis (six variants) are possible in software so there is no need for accurate positioning and handling of the test sample. This allows deviations between the template and test sample to be identified in real time.

Preparing for Measurement

Image quality has a major influence on the resulting measurements. Image quality is dependent on resolution, contrast, and depth of field, perspective, and distortion. These, in turn, are determined by the choice of system components including cameras, lenses, and illumination. Cost is also an important consideration. The best components for the application should be used, but overspecifying them leads to unnecessary costs for no gain. Since all the components in the industrial vision system must be perfectly coordinated it is essential to make an initial evaluation of the application:

- What objects are to be measured?
- How large is the measurement area?
- What type of measurement is required?
- Are multiple views/measurements required?
- How fast are the parts moving?
- What measurement accuracy is needed?
- Is color identification needed?

These and other factors help to determine the specification of the vision components needed, but there are also environmental issues that should be taken into account. These include physical constraints on positioning of components and environmental conditions such as ambient light, and so on. The resulting system does not need to be a complicated set-up, it simply needs to be fit for purpose.

With 3D machine vision technology becoming much more widely available, a similar process should be adopted when specifying a system to make 3D measurements. Although 3D systems have become much more affordable in recent years, they are still generally more expensive than 2D systems and add more data and more complexity, so they should only be specified when the required measurement can't be made using 2D methods. With a variety of 3D measurement techniques available, it is also important to choose the method.

Industrial Control

Many of today's industrial controllers feature embedded i7 Intel processors, multiple I/O channels, display output capability, and support for dual SATA raid systems. While some offer the ability to add extra board-level peripherals, many simply offer Gigabit Ethernet or USB interfaces. For developers wishing to interface to higher-speed, deterministic interfaces such as Camera Link, industrial controller manufacturers and industrial vision vendors now offer embedded systems to support such interfaces. Indeed, because of the blurring differences between many of the functions of industrial controllers and vision processors, the choice of which camera and camera interface can rapidly narrow the choice of which to use.

In some cases, an embedded industrial controller or industrial vision system may not provide the correct type of interface required. In such cases, systems integrators can choose from a number of CPUs and add-in boards with which to develop an embedded system. Then, by using PMC, XMC, FMC mezzanine, or PCI Express Mini frame grabber cards, numerous analog and digital interfaces can be accommodated.

To accommodate other peripherals such as digital I/O, WiFi and communication protocols such as PROFIBUS (Process field bus), however, will require the use of multiple add-in cards, making the systems integration task more complex. Rather than take this approach, systems developers can choose from a number of industrial controllers that can be expanded to support such interfaces while already integrating much of the functionality required by industrial control systems.

Combining small dimensions with powerful processing, embedded vision systems are suited to an extremely diverse range of industrial applications. These include everything from self-driving vehicles and driver assistance systems through drones, retail, security, biometrics, medical imaging, and augmented reality to robots and networked objects. The list is endless.

Development Approaches and Environments

The development of a machine vision system begins with understanding the application's requirements and constraints and proceeds with selecting appropriate machine vision software and hardware (if necessary) to solve the task at hand. Older machine vision systems were built around low-level software, requiring full programming control. They were based on simple frame grabbers providing low-level interface capabilities with other system components. They were also characterized by low-level user interfaces, low-level image analysis capabilities and difficulties in system integration and maintenance. Eventually, machine vision inspection systems became more modular, providing more abstract capabilities for system development and maintenance and reaching higher level of robustness.

Today's applications need environments that are developed in short time and are adjusted to modifications of the manufacturing process. In addition, the system must be simple to operate and maintain. The key here is to select an appropriate development environment providing graphical user interfaces (GUIs) or other programming tools. Through GUIs and visual programming tools, even nonvision experts but authorized users, for example, manufacturing engineers, are allowed to interact with the application and specify sequences of operations from pull-down menus offering access to large pools of tested algorithms. Programming is easier in this case, since the algorithms are selected based on knowledge of what they do and not on how they do it. The use of GUIs shifts the effort of application development to the manufacturing engineer from the programmer expert, as in the earlier days of machine vision systems. This feature not only results in faster and cheaper application developments, but also allows addressing several applications with a single piece of reconfigurable software (i.e., the application development tool).

Industrial vision systems must be fast enough to meet the speed requirements of their application environment. Speed depends on the task to be accomplished and may range from milliseconds to seconds or

minutes. As the demands of processing increase, special purpose hardware is required to meet high-speed requirements. A cost-saving feature of industrial vision systems is their ability to meet the speed requirements of an application without the need of special purpose hardware. PCs and workstations are nowadays fast enough so that this can be achieved in many application domains, especially in those with less demanding run time requirements.

Advances in hardware technology in conjunction with the development of standard processing platforms have made the production and maintenance of industrial automation systems feasible at relatively low cost. Pentium PCs with Windows NT (Windows 2000, XP) or UNIX-based systems or Linux are considered the main alternatives. Windows being preferred to achieve labor saving application development with maximum portability based on ready-to-use software (e.g., commercially available software). Linux is becoming a standard especially in cases where customized or cost-saving solutions are preferred. Linux is sometimes offered as open-source freeware and appears to be the ideal solution in the case of dedicated applications where independency on vendor-specific software has to be achieved. However, the limited availability of application development tools (e.g., interfacing software) is a serious drawback of Linux.

Software implementations are often insufficient to meet the real-time requirements of many industrial vision applications. The ever-increasing computational demands of such applications call for hardware tools implementing image processing algorithms. In the following ASICs, DSPs, FPGAs and general-purpose processors are considered as possible alternatives in dealing with the problem of processing speed. The choice among them has to be made taking into account issues such as size of chip, power dissipation, and performance. However, issues such as flexibility of usage and programming environment are now becoming of great importance for the application developers.

Development Software Tools for Industrial Vision Systems

The selection of the appropriate software tools is of crucial importance. A software tool must have the following desirable features:

Multi-processing level support: The type of processing in an industrial vision system varies from low level (e.g., filtering, thresholding), to medium level (e.g., segmentation, feature computation) and high level

(e.g., object recognition, image classification, etc.). An image software package must support all levels of functionality. Otherwise, different software tools must be adopted and integrated into the same system.

Ease of manipulation: Graphical user interfaces, visual programming and code generation are typical features facilitating application development. Image functions must be categorized by type and scope so that even a nonexpert may choose the appropriate function based mostly on what it does rather than on how it is done.

Dynamic range and frame-rate support: New types of sensors (e.g., CMOS sensors) offer high dynamic range and faster image acquisition (e.g., 16 bits per pixel instead of 8 bits per pixel). Image software must support the processing of such high dynamic range images at variable frame rates.

Expandability: The software package must be able to accommodate new or better algorithms substituting old ones. In addition, the software package must easily adjustable to new requirements of the running application without major programming effort.

Dedicated hardware support: The software package must be able to work in collaboration with hardware (e.g., DSPs, ASICs, or FPGAs) to alleviate the problem of processing speed in the case of computationally intensive applications.

The following presents either the most commonly used or best-suited software tools for industrial vision applications. This list is not a complete one. This review includes image processing and analysis tools, as well as tools based on neural networks, fuzzy logic, and genetic algorithms.

Image Processing and Analysis Tools

Image processing is usually performed within rectangles, circles, or along lines and arcs. Image processing operators include filtering (e.g., smoothing, sharpening), edge detection, thresholding, morphological operations, and so on. Such operations can be used to improve image quality (e.g., noise removal, improve contrast) and to enhance or separate certain image features (e.g., regions, edges) from the background. Image processing operations transform an input image to another image having the desired characteristics.

Image analysis transforms images to measurements. In particular, image analysis is related to the extraction and measurement of certain

image features (e.g., lines, and corners) and transforms these image features to numbers, vectors, character strings, etc. For example, lines, regions, characters, holes, rips, and tears can be gauged or counted. Image analysis involves feature extraction operations (e.g., Hough transform for line and curve detection) in conjunction with operations that measure average light intensity, texture, and shape characteristics such as Fourier descriptors, moments, edge thinning, edge connectivity, and linking. The ultimate goal of image analysis is geared toward pattern recognition, that is, the extraction of features that can be used by classifiers to recognize or classify objects.

An image processing environment to be suitable for industrial inspection, must (at least) contain algorithms for edge and line detection, image enhancement, illumination correction, geometry transforms, region of interest (RoI) selection, object recognition, feature selection and classification. These tools offer adequate features and performance for several applications involved in the industrial sector. In terms of combined software and hardware software packages are IPL Lib, Sherlock32/MV Tools, WiT, PC Image Flow, MIL, and Rhapsody. Software packages like Hhoros, SCIL Image, and Image-Pro plus, OPTIMAS, VISION97, and AdOculus are also available for image processing and analysis.

Summary

- Industrial vision includes algorithms to measure and identify items in the image and pass data in a compatible format back to the controlling PLC.
- Industrial vision is used for faster manufacturing and inspection process.
- Dimensional quality such as dimensions, shape, positioning, orientation, alignment, roundness, and corners are detected features of the inspection products.
- Structural quality includes the assembling of holes, slots, rivets, screws, clamps, and removing foreign objects such as dust, bur, swarm.
- Pits, scratches, cracks, wear, finish, roughness, texture, seams-folds-lap, continuity are surface quality features of inspected products.
- Inspection of accurate or correct operation of inspected product is known as operational quality.
- 3D imaging has the ability to make real-time measurements in the X, Y, and Z axis at production line speedup allowing volumes of product as well as defects to be detected.

- Time of flight cameras measure the time it takes for a light pulse to reach the object and return for each image point.
- High-speed line scan imaging is used for web-inspection application on fast-moving continuous product for identifying, classifying of faults and defects.
- Exposure time is the amount of time that light is allowed to fall onto the camera sensor.
- Line scan cameras are used for sorting a large variety of objects such as food, waste, parcel, and so on.

References

ukiva.org

<http://www.eenewseurope.com/news/machine-learning-embedded-vision-applications>

<https://arxiv.org/pdf/1510.00149v5.pdf>

<https://www.alliedvision.com/en/embedded-vision/understanding-embedded-vision.html>

Learning Outcomes

- 2.1 What do you mean by industrial vision system?
- 2.2 Draw a typical industrial vision system and write about it.
- 2.3 What are the requirements of industrial cameras?
- 2.4 Write about a food factory industrial vision system.
- 2.5 Define smart camera.
- 2.6 Classify industrial vision applications based on inspection.
- 2.7 Write a short note on 3D imaging methods.
- 2.8 What is 3D inspection?
- 2.9 What is the need of high-speed cameras in industry?
- 2.10 Explain industrial vision measurement.

Further Reading

Intelligent Vision Systems for Industry by Bruce G. Batchelor and Paul F. Whelan

MEDICAL VISION

Overview

Vision technology in medicine allows doctors to see interior portions of the body for easy diagnosis, making it possible to perform minimally invasive surgery when surgery is necessary. Image processing for medical applications include endoscopy, CT, ultrasonic imaging system, MRI, X-ray imaging, MIS, PACS, Ophthalmology, ICG imaging, corneal image analyzer, fundus image analyzer, facial recognition to determine pain, patient activity detection, peripheral vein imaging, and stereoscopic microscope. CNN takes an input image of raw pixels and transforms them via convolution layers, rectified linear unit layers, and pooling layers. This feeds into fully connected layers that assign class scores, thus classifying the input into the affected area.

Learning Objectives

After reading this the reader will be know about

- different applications used in the medical field,
- the designing of telemedicine and picture archiving communication systems,
- modules to get information from medical images,
- the scheme of medical vision image processing,
- vision algorithms for biomedical,
- algorithms for image compression,
- CNN in medical image analysis, and
- machine learning for the medical field.

3.1 INTRODUCTION TO MEDICAL VISION

Medical vision imaging has been undergoing a revolution in the past decade with the advent of faster, more accurate, and less invasive devices. This has driven the need for corresponding software development, which in turn has provided a major driving force for new algorithms in signal and image processing. Many of these algorithms are based on partial differential equations and curvature driven flows. Mathematical models are the foundation of biomedical computing. Based on those models data is extracted from images continues to be a fundamental technique for achieving scientific progress in experimental, clinical, biomedical, and behavioral research. Today, medical images are acquired by a range of techniques across all biological scales, which go far beyond the visible light photographs and microscope images of the early 20th century.

In the future, medical devices will use increasing degrees of embedded vision technology to better diagnose, analyze, and treat patients. The technology has the potential to make healthcare safer, more effective, and more reliable than ever before. Subjective methods of identifying ailments and guess-and-check treatment methods are fading. Smarter medical devices with embedded vision are the future. The artificial intelligence models used in the study for diagnosing and treating patients resulted in a 30–35% increase in positive patient outcomes.

Vision enhanced medical devices will be able to capture high-resolution images, analyze them, and provide recommendations and guidance for treatment. Next generation devices will also be increasingly automated, performing pre-operation analysis, creating surgical plans from visual and other inputs, and in some cases even performing surgeries with safety and reliability that human hands cannot replicate. As embedded vision in medical devices evolves, more uses for it will emerge. The goal is to improve safety, remove human subjectivity and irregularity, and administer the correct treatment the first (and only) time. Embedded vision has the potential to enable a range of medical and other electronic products that will be more intelligent and responsive than before and thus more valuable to users. The technology can both add helpful features to existing products and open up brand new markets for equipment manufacturers.

Biomedical systems support the more effective delivery of many image guided procedures such as biopsy, minimally invasive surgery, and radiation therapy. In order to understand the extensive role of imaging in the

therapeutic process, and to appreciate the current usage of images before, during, and after treatment, the analysis on four main components of image guided therapy (IGT) and image guided surgery (IGS) are required. They are localization, targeting, monitoring, and control. Specifically, in medical imaging there are four key problems:

1. Segmentation—automated methods that create patient-specific models of relevant anatomy from images;
2. Registration—automated methods that align multiple data sets with each other;
3. Visualization—the technological environment in which image-guided procedures can be displayed;
4. Simulation—software that can be used to rehearse and plan procedures, evaluate access strategies, and simulate planned treatments.

Vision imaging technology in medicine allows doctors to see the interior portions of the body for easy diagnosis. It also helps doctors to make keyhole surgeries for reaching the interior parts without being too invasive. CT scanner, ultrasound, and magnetic resonance imaging took over X-ray imaging by allowing doctors to look at the body's elusive third dimension. With the CT scanner, the body's interior can be bared with ease and the diseased areas can be identified without causing either discomfort or pain to the patient. MRI picks up signals from the body's magnetic particles spinning to its magnetic tune and with the help of its powerful computer converts scanner data into revealing pictures of internal organs. Image processing techniques developed for analyzing remote sensing data may be modified to analyze the outputs of medical vision imaging systems for easy and optimal patient symptom analysis.

Advantages of Digital Processing for Medical Applications

- Digital data will not change when it is reproduced any number of times and retains the originality of the data.
- Digital image data offers a powerful tool to physicians by easing the search for representative images.
- Images are displayed immediately after acquisition.
- Enhancement of images makes them easier for the physician to interpret.

- Allows for quantifying changes over time.
- Provides a set of images for teaching to demonstrate examples of diseases or features in any image.
- Allows for quick comparison of images.

Digital Image Processing Requirements for Medical Applications

- Interfacing analog outputs of sensors such as microscopes, endoscopes, ultrasound, and so on to digitizers and in turn to digital image processing systems
- Image enhancements
- Changing density dynamic range of B/W images
- Color correction in color images
- Manipulation of colors within an image
- Contour detection
- Area calculations of the cells of a biomedical image
- Display of image line profile
- Restoration of images
- Smoothing of images
- Registration of multiple images
- Construction of 3-D images from 2-D images
- Generation of negative images
- Zooming of images
- Pseudo coloring
- Point-to-point measurements
- Getting relief effect
- Removal of artifacts from the image

Advanced Digital Image Processing Techniques in Medical Vision

- Neural network-based image processing
- Statistical approach for texture analysis

- Segmentation in color and B/W images
- Expert system-based image processing.
- Application of object-oriented programming techniques in image processing environments
- Shape in machine vision
- Multispectral classification techniques
- Auto-focus techniques for MRI images
- Threshold technique for finding contours of objects
- Sequential segmentation technique to detect thin vessels in medical images and hair-line cracks in nondestructive testing
- Fractal method for texture classification
- Data compression techniques using fractals and discrete cosine transformers
- Image restoration methods using point-spread functions and Wiener filter, and so on.

Image Processing Systems for Medical Applications

(a) Endoscopy

In each endoscope, there are two fiber bundles. One is used to illuminate the inner structure of object. The other is used to collect the reflected light from that area. The endoscope is a tubular optical instrument used to inspect or view the body cavities, which are not normally visible to the naked eye.

For a wider field of view and better image quality, a telescope system is added in the internal part of the endoscope. Gastrointestinal fiberoscopes and laparoscopes are important endoscopes used in hospitals for examination, treatment of diseases, and surgery. Technological advances in computers and semiconductor chips have brought about lots of changes in healthcare during the last decade. For digestive diseases, this advancement is represented by the incorporation of charge-coupled devices (convert optical image to electronic image) into gastrointestinal endoscopy. These *video endoscopes* use xenon arc lamps as light source. Color imaging is achieved by incorporating RGB filters between Xenon Lamp Supply and

the proximal end of the endoscope. The other approach to the generation of color image is to divide the available cells on the CCD among the three primary colors by means of filters. Three images one for each color are then produced simultaneously by the CCD. Endoscopic pictures are converted to digital images by using CCD cameras and associated image digitizer circuits into a PC /AT. The recorded images can be image processed for better quality.

Stereoscopic Endoscope

An endoscope is a medical instrument used to examine the interior of a hollow organ or a cavity of the body. Conventional endoscopes included just one camera module and therefore, and could only provide 2D images of the operative site on a monitor. When performing laparoscopic surgery, surgeons depend on the endoscopic image to perform very sophisticated and time-consuming surgical processes, cutting, suturing, and clipping. However, there are possibilities of accidents during surgical procedures since the depth information is missing in the 2D endoscopic image. If the endoscopic image is provided to the surgeon in three dimensions, the accuracy of the operation can be improved. Recently, a stereoscopic endoscope has been developed and applied to minimally invasive surgery to improve the surgeon facility and the patient safety. A robot system utilizes the stereoscopic endoscope to provide more realistic images of the operative site to the operator.

Two cameras are mounted on a single laparoscope. Images from the cameras are transmitted alternately to a video monitor. Few types of display techniques are used to realize *stereo images* from two-dimensional images recorded from the cameras. As the cameras transmit images at 60–120 cycles per second a three-dimensional, real-time image is perceived. As the images are transmitted at a high frequency, the effect is that of seeing different images simultaneously.

(b) Computer tomography (CT)

Computerized axial tomography or computer transmission tomography or computer tomography is a method of forming images from X-rays. Measurements are taken from X-rays transmitted through the body. These contain information on the constituents of the body in the path of the X-ray beam. By using multidirectional scanning of the object, multiple data is collected.

An image of a cross-section of the body is produced by first measuring the total attenuation along rows and columns of a matrix and then by computing the attenuation of the matrix elements at the intersections of the rows and columns. This produces the number of mathematical operations necessary to yield clinically applicable and accurate images. Hence a computer is essential to do them. The information obtained from these computations can be presented in a conventional raster form resulting in a two dimensional picture.

The timing, anode voltage, and beam current are controlled by a computer through a control bus. The high-voltage DC power supply drives an X-ray tube that can be mechanically rotated along the circumference of a gantry. The patient lies in a tube through the center of the gantry. The X-rays pass through the patient and are partially absorbed. The remaining X-ray photons impinge upon several radiation detectors fixed around the circumference of the gantry. The detector response is directly related to the number of photons impinging on it and hence to the tissue density. When they strike the detector, the X-ray photons are converted to scintillations. The computer senses the position of the X-ray tube and samples the output of the detector along a diameter line opposite to the X-ray tube. A calculation based on data is obtained from a complete scan made by the computer. The output unit then produces a visual image of a transverse plane cross-section of the patient on the cathode ray tube. These images are also stored into computer for image processing.

(c) Ultrasonic imaging system

Ultrasonography is a technique by which ultrasonic energy is used to detect the state of the internal body organs. Bursts of ultrasonic energy are transmitted from a piezo electric or magnetostrictive transducer through the skin and into the internal anatomy. When this energy strikes an interface between two tissues of different acoustical impedance, reflections (echoes) are returned to the transducer. The transducer converts these reflections to an electric signal proportional to the depth of the interface, which is amplified and displayed on an oscilloscope.

An image of the interior structure is constructed based on the total wave travelling time, the average sound speed, and the energy intensity of the reflected waves. The echoes from the patient body surface are collected by the receiver circuit. Proper depth gain compensation (DGC) is given by DGC circuit. The received signals are converted into digital signals and

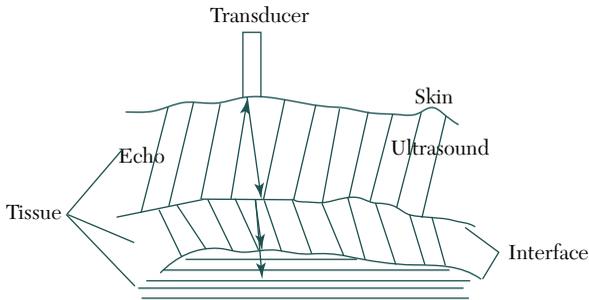


FIGURE 3.1. Ultrasound imaging system.

stored in memory. The scan converter control receives signals of transducer position and TV synchronous pulses. It generates X and Y address information and feeds to the digital memory. The stored digital image signals are processed and given to the digital analog converter.

They are then fed to the TV monitor. These signals are converted to digital form using a frame grabber and can be stored onto a PC/AT disk. Wherever the images lack in contrast and brightness, image processing techniques may be used to get full details from ultrasound images. Figure 3.1 shows an ultrasound imaging system.

(d) *Magnetic resonance imaging (MRI)*

Superconducting magnets are used in MRI systems to provide strong, uniform, steady magnetic fields. The superconducting magnetic coils are cooled to liquid helium temperature and can produce very high magnetic fields. Hence the signal to noise ratio of the received signals and image quality are better than the conventional magnets used in the MRI systems.

Different gradient coil systems produce time varying, controlled, spatial nonuniform magnetic fields in different directions. The patient is kept in this gradient field space. There are also transmitter and receiving RF coils surrounding the site on which the image is to be constructed. There is a superposition of a linear magnetic field gradient on to the uniform magnetic field applied to the patient. When this superposition takes place, the resonance frequencies of the processing nuclei will depend on the positions along the direction of the magnetic field gradient. This produces a one-dimensional projection of the structure of the three-dimensional object. By taking a series of these projections at different gradient orientations using X, Y, and Z gradient coils a two- or three-dimensional image can be obtained. The slice of the image depends upon the gradient magnetic field. The gradient magnetic field is controlled by computer and that field can be positioned in three time-invariant planes (X, Y, and Z). The transmitter provides the RF signal pulses. The received nuclear magnetic resonance

signal is picked up by the receiver coil and is fed into the receiver for signal processing. By two-dimensional Fourier transformation, the images are constructed by the computer and analyzed using image processing techniques.

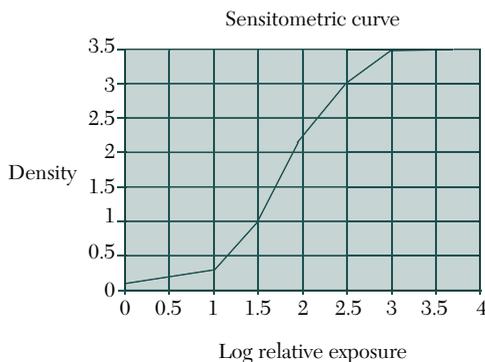
MRI data consists of multiple channels of independent but geometrically registered medically significant data, analogous to multispectral remote sensing data. Multispectral analysis of proton MR images may provide tissue characteristic information encoded therein. Using well-established methods for computer processing of multispectral images, tissue characterization signatures are sought using supervised or unsupervised classification methods.

The principal advantages of multispectral analysis of MRI include:

- It is a quantitative means of analyzing multidimensional image data.
- In other applications, multispectral methods have been useful in identifying qualities that would otherwise be overlooked.
- MR images are intrinsically multispectral. The data in a set of MR images is highly redundant, both geometrically and radiometrically.
- Multispectral methods are well developed and have been implemented on computers, for which software is readily available that can process MR image data efficiently, and can adapt to existing MR scanners.

(e) X-ray imaging

X-ray films have large dynamic range to accommodate maximum possible details of X-ray images. The sensitometric curve of the X-ray image is as shown in Figure 3.2.



X-ray images can be converted into digital form using X-ray fluoroscopy technique or by digitizing X-ray film using scanners. By applying image processing techniques, the digital images can be manipulated for easy interpretation. Using these techniques allows for a reduction of additional X-ray exposure to the patient.

FIGURE 3.2. Sensitometric curve of an X-ray image.

(f) Medical imaging system (MIS)

Most hospital imaging departments have to computerize information systems in which patient images and reports are to be stored. The stored information can be handled by two major types of medical applications, the *integrated report* and *review* applications. The former is performed by experts (e.g., radiologists) in four steps:

1. retrieving and viewing images;
2. processing, interpretation and annotation of the diagnosis;
3. composition of final diagnostic multimedia report; and
4. permanent storing in the database.

The latter allows many simultaneous users (authorized patient-care personnel) to view, read, and listen to the diagnostic report; these users do not alter the handled data. It opens hospitals the possibility to obtain applications enhanced with conferencing services, referred to as *consult* applications, to geographically distributed users. For example, physicians located in a rural healthcare center could consult one or more experts located in the regional hospital of an urban area and gain access to patient images and other information through parallel interfacing with the database of the hospital imaging department. The implementation of such applications requires an appropriate infrastructure with high-speed networking and image manipulation facilities. The consult application is achieved by accessing picture archiving and communication systems (PACS).

(g) Picture archiving and communication systems (PACS)

A picture archiving and communication system (PACS) is essentially a network system that allows digital or digitized images from any modality to be retrieved, viewed, and analyzed by a relevant expert, or by an appropriate expert system, at different workstations. These images may be held in archives, that is, they are to be stored “permanently” on DVD, and/or be transmitted to/from remote sites, “tele-radiology.” The digital imaging and communications in medicine (DICOM) format allows images, and cine-loop images, with associated patient information and reports, including voice notes, to be stored and exchanged readily over the network. PACS systems can be integrated into radiological/hospital information systems (RIS/HIS), with the inclusion of administrative information such as billing and inventory. This is shown in Figure 3.3.

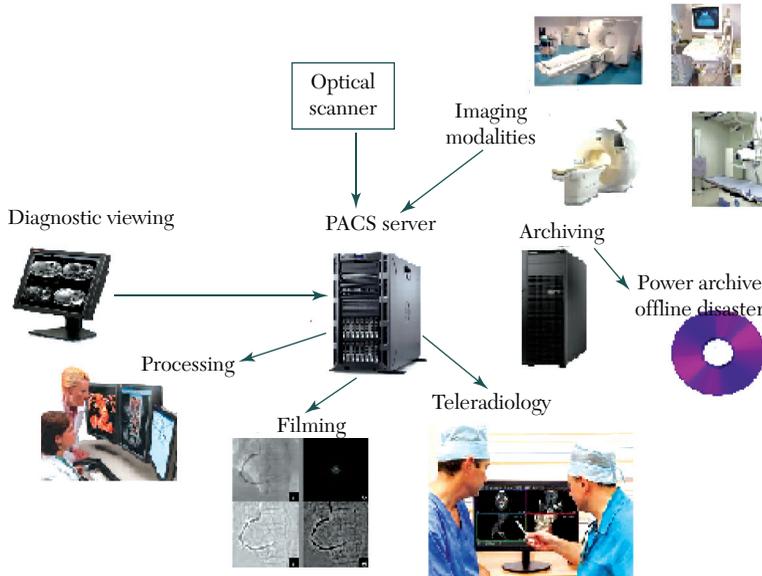


FIGURE 3.3. A PACS system.

Most modalities, including mammography, are now becoming digital. Increasingly, plain-film radiography is being replaced by computed radiography or by direct digital radiography. The advantages of PACS systems include:

1. Filmless radiology (no darkroom required, no chemical developers to purchase, no bulky storage rooms);
2. Easy access to images, including those from remote sites;
3. Easy image processing/enhancement;
4. Easy registration of images from different modalities;
5. Compression of images for quicker communication.

Major disadvantages are the large capital costs of setting it up and training personnel, as well as the inevitable difficulties of phasing it in. Nevertheless, it is the obvious way to proceed, and a large number of hospitals have implemented or are implementing PACS systems.

- In application domains such as remote sensing, astronomy, cartography, meteorology, and medical imaging, images comprise the vast majority of acquired, processed, and archived data.

- The medical imaging field in particular, has grown substantially in recent years, and has generated additional interest in methods and tools for the management, analysis, and communication of medical image data.
- PACS are currently used in many medical centers to manage the image data.
- Important considerations in the design and implementation of image database (IDB) systems are:
 - image feature extraction;
 - image content representation and organization of stored information;
 - search and retrieval strategies; and
 - user interface design.
- Image retrieval using color, shape, and texture.
- Content-based image retrieval for image databases.
- Region detection in medical images. For example, locating endocardium boundaries of the left and right ventricles from gradient-echo MR images.

Current research activities are interested in the implementation of smaller and more flexible systems than PACS, called mini-PACS.

(h) Digital Image Processing For Ophthalmology

- To analyze retina, optic nerve, pigment epithelium, and choroid in the ocular fundus.
- Color slides have a resolution of 4000 x 3000 pixels.
- Fluorescein Angiograms have a resolution of 1800 x 1350 pixels.
- Common standard digital cameras have resolution of 512 x 480, which may be sufficient for obtaining relevant information about blood vessels, for example; 2048 x 2048 element resolution cameras are also presently used.
- 8-bit resolution (indicative of contrast) is sufficient for most of the ophthalmology images.

(i) Indo cyanine green (ICG) imaging

In blood, about 20% to 40% of injected sodium fluorescein remains unbound to serum albumin. This unbound fluorescein leaks rapidly from the highly fenestrated choriocapillaries into the choroidal anatomy. Because of this, details of the choroidal vascular pattern are obscured. For these two reasons, fluorescein angiography cannot provide useful information on choroidal circulation. In ICG angiography, the maximum absorption and peak fluorescence of Indocyanine dye is in the spectrum at 805nm and 835nm respectively. This near infrared light can penetrate the retinal pigment epithelium much more effectively than visible light, allowing uninterrupted examination of the choroidal vascular network. In addition, since approximately 98% of ICG dye in blood is bound to serum proteins, it leaks very slowly from the choroidal capillaries.

- ICG angiography is basically similar to that of fluorescein angiography.
- Differences are: spectral characteristics and permeability from choroidal capillaries.
- Sodium fluorescein dye used in fluorescein angiography has a maximum absorption at 485nm and peak emission at 520nm.
 - The largest portion of excitation and emission energy of this visible light is absorbed by the retinal pigment epithelium and macular xanthophyll.
 - As a result, it is difficult to obtain sufficient fluorescence from the deeper layers of choroidal vessels.

(j) Corneal Image Analyzer

Corneal Image Analyzer (CIA) is software for analyzing endothelial images of cornea. The corneal image consists of a set of hexagonal shaped cells of different sizes. The aim of this software is to compute the statistics of the cornea endothelial such as the cell density, minimum, maximum, and mean cell sizes, their standard deviation, covariance, and so on. This data is useful for various studies such as dystrophy and degeneration, intraocular lens implantation, corneal transplantation, drug toxicity, glaucoma, and so on

The cornea of the patient is scanned by special devices and these images can be analyzed using computers at greater speed and precision. The

images of endothelium can be filmed with a 35mm still camera attached to the specular microscope. The images can be fed to the computers as input by scanning the films/prints or by capturing the images directly from the patient's eye through a high resolution and highly sensitive CCD camera whose video output is connected to a frame-grabber board on a PC/AT. The analysis is done on a high-resolution monitor. The quantitative measurements can be done on computers using image processing techniques. Prints of these images are used for diagnosis.

The software gives a printout of the cell density, minimum, maximum, and mean cell sizes, standard deviation and coefficient of variance of the cell sizes, histogram of cell sizes, and distribution of cell areas. Several reports of a patient at various dates are combined to give a combined report.

(k) Fundus image analyzer

Many diseases of the human visual system and of the whole body can have a dramatic impact on the 3-dimensional geometry of the ocular fundus. Glaucoma is probably the most important disease in this category. It increases the cupping of the optic nerve head at an early stage of the disease, in many cases before a reliable diagnosis can be made and visual-field losses occur. The early diagnosis of glaucoma is a major issue in general public healthcare. Quantitative assessment of fundus geometry is expected to be helpful for this purpose.

The ocular fundus consists of several layers of highly transparent tissue, each having individual physical properties, reflectivity, absorption, and scatter. 2-dimensional geometry normally specifies substructures such as the vessel pattern or the area of pallor delineated by contrast or color variations. It is less important how deep they are located within the fundus. Depth is commonly associated with the topography of the interior limiting surface of compact retina and optic disc tissue. A system for ocular fundus analysis consists of two parts: the image acquisition and the analysis software. The image is normally obtained using a telecentric fundus camera (e.g., Ziess-30 degree fundus camera). The image is captured onto a slide film. The film is scanned using a slide film scanner and transcribed to a personal computer. Alternatively the image can be directly acquired from the camera by interfacing it to the personal computer using a frame-grabber card. The present version operates on 2-D images only and does not support depth/volume calculation. Figure 3.4 shows the fundus image analysis system. Quantitative measurements of a fundus image are shown

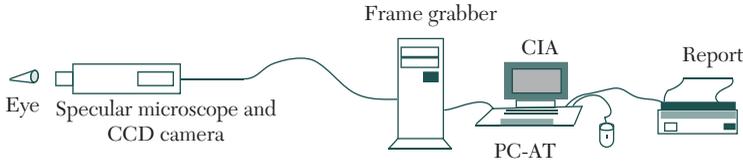


FIGURE 3.4. Image acquisition and processing of endothelial cells of the cornea.

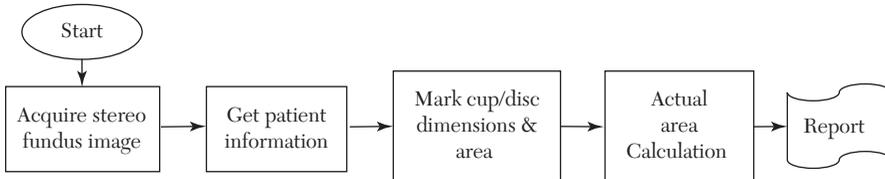


FIGURE 3.5. Measurement of fundus image.

in Figure 3.5. The following is the list of parameters calculated using this software:

Disc diameter: Specifies the horizontal/vertical diameter of the selected disc edge.

Cup diameter: Specifies the horizontal/vertical diameter of the selected cup edge.

Disc Area: Specifies the area within the selected disc edge.

Cup Area: Specifies the area within the selected cup edge.

Rim Area: Specifies the area between the selected disc edge and the calculated cup edge.

Cup-to-Disc Ratio: Specifies the ratio of the cup size to disc size.

This package is useful for quantitative measurements of the optic nerve. If the depth information is derived from the stereo pair, then volume calculations and profile generations can also be done.

(I) Automatic classification of cancerous cells from a digitized picture of a microscope section

Mathematical morphology is used to remove the background noise from the image and the nuclei of the cells are segmented. These nuclei are analyzed for shape and size. The texture of the nuclei is evaluated by using a neural network. Automatic classification of the image is done.

(m) Facial Recognition to Determine Pain Levels

High-fidelity robotic human patient simulators (HPS) have the ability to exhibit realistic clinically relevant facial expressions to access and treat patients. With a high rate of accuracy in identifying a wide range of facial expression, 3D facial expression databases operate with the ultimate goal of increasing the general understanding of facial behavior and 3D structure of facial expressions on a detailed level. The building blocks of these projects are cameras linked with data that can identify and quantify pain states.

(n) Automated Detection of Patient Activity

Traditionally, monitoring of patients has required some sort of physical contact with the monitoring device, such as EEG patches and fingertip oxygen sensors. For patients at risk for falls or self-injury, the solution has usually been “patient-sitters” who are stationed in the room. Video surveillance manager feeds high-definition video to an operations center where trained staff can notify appropriate personnel by two-way video, voice, text, paging, or integration with existing nurse call systems. A camera that recognizes when a patient sits up in bed, gets out of bed, or is tossing and turning in a restless effort to get to sleep will help patient caretakers.

(o) Measurement of changes in heart rate from head movement

Each time the human heart beats, the head moves slightly. Video of head movements into accurate data on heart activity based on imaging and analysis will improve diagnosing.

(p) Peripheral vein imaging

Finding a viable vein for an injection can be tricky, even for experienced phlebotomists. Multiple sticks can waste time during an emergency, in addition to causing the patient discomfort. Projected near infrared light is absorbed by blood and reflected by surrounding tissue. The information is captured, processed, and projected digitally in real time directly onto the surface of the skin. It provides a real-time accurate image of the patient’s blood pattern. Technology like this would be very welcome in hospitals, outpatient labs, chemotherapy suites, and blood drives.

(q) Stereoscopic microscope

A stereo microscope is an optical microscope that consists of two separate optical paths and two eyepieces to provide slightly different viewing angles to the left and right eyes, which produces a 3D visualization

of the object being examined. It eliminates the need to keep one eye closed while peering through a microscope lens and makes it easier to focus on the object. It can provide more realistic 3D morphology of the object and therefore is suitable for investigating the 3D structure of complex biological specimens.

3.2 FROM IMAGES TO INFORMATION IN MEDICAL VISION

Basic endoscopic technologies and their routine applications (Figure 3.6, bottom layers) are still purely data-oriented, as the complete image analysis and interpretation is performed only by the physician. If content of endoscopic imagery is analyzed automatically, several new application scenarios for diagnostics and intervention with increasing complexity can be identified (Figure 3.6, upper layers). As these new possibilities of endoscopy are inherently coupled with the use of computers, these new endoscopic methods and applications can be referred to as computer-integrated endoscopy. Information however referred from the highest of the five levels of semantics is shown in Figure 3.6.

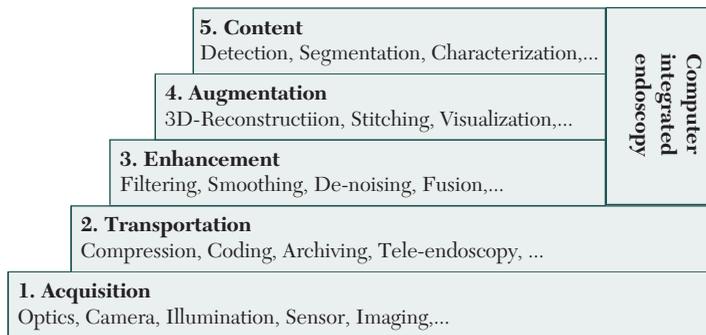


FIGURE 3.6. Modules to build computer integrated endoscopy that enables information gain from image data.

1. *Acquisition*: Advancements in diagnostic endoscopy were obtained by glass fibers for the transmission of electric light into and image information out of the body. It includes wireless broadcast available for gastroscopic video data captured from capsule endoscopes.
2. *Transportation*: Based on digital technologies, essential basic processes of endoscopic still image and image sequence capturing, storage, archiving, documentation, annotation, and transmission have been simplified. These developments have initially led to the possibilities for tele-di-

agnosis and tele-consultations in diagnostic endoscopy, where the image data is shared using local networks or the Internet.

3. *Enhancement*: Methods and applications for image enhancement include intelligent removal of honey-comb patterns in fiberscope recordings, temporal filtering for the reduction of ablation smoke, moving particles, and image rectification for gastroscopies. Additionally, besides having an increased complexity, they have to work in real time with a maximum delay of 60 milliseconds, to be acceptable for surgeons and physicians.
4. *Augmentation*: Image processing enhances endoscopic views with additional information. Examples of this type are artificial working horizon, key-hole views to endoscopic panorama-images, 3D surfaces computed from point clouds obtained by special endoscopic imaging devices such as stereo endoscopes, time-off light endoscopes, or shape-from polarization approaches. This level also includes the possibilities of visualization and image fusion of endoscopic views with preoperative acquired radiological imagery such as angiography or CT data for better intra-operative orientation and navigation, as well as image-based tracking and navigation through tubular structures.
5. *Content*: Methods of content-based image analysis consider the automated segmentation, characterization and classification of diagnostic image content. Such methods describe computer-assisted detection (CADe) of lesions (such as polyps) or computer-assisted diagnostics (CADx) where already detected and delineated regions are characterized and classified into, for instance, benign or malign tissue areas. Furthermore, such methods automatically identify and track surgical instruments, for example, supporting robotic surgery approaches.

On the technical side the semantics of the extracted image contents increases from the pure image recording up to the image content analysis level. This complexity also relates to the expected time axis needed to bring these methods from science to clinical applications.

From the clinical side, the most complex methods such as automated polyp detection (CADe) are considered as most important. However, it is expected that computer-integrated endoscopy systems will increasingly enter clinical applications and as such will contribute to the quality of the patient's healthcare.

Digital images are composed of individual pixels (this acronym is formed from the words “picture” and “element”), where discrete brightness or color values are assigned. They can be efficiently processed, objectively evaluated and made available on many places at the same time by means of appropriate communication networks and protocols, such as picture archiving and communication systems (PACS) and the digital imaging and communications in medicine (DICOM) protocol, respectively. Based on digital imaging techniques, the entire spectrum of digital image processing is now applicable to the study of medicine.

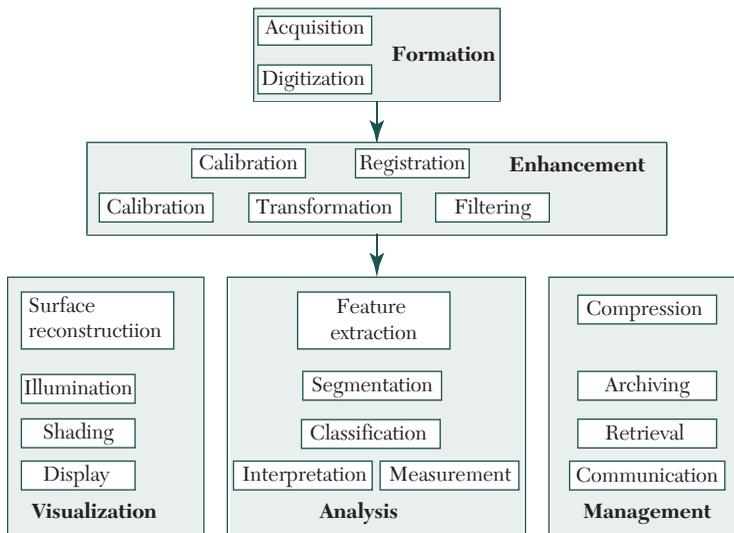


FIGURE 3.7. Scheme of medical vision image processing.

The commonly used term “medical image processing” means the provision of digital image processing for medicine. Medical image processing covers five major areas as shown in the Figure 3.7.

1. *Image formation* includes all the steps from capturing the image to forming a digital image matrix.
2. *Image visualization* refers to all types of manipulation of this matrix, resulting in an optimized output of the image.
3. *Image analysis* includes all the steps of processing used for quantitative measurements as well as abstract interpretations of medical images. These steps require a priori knowledge on the nature and content of the

images, which must be integrated into the algorithms on a high level of abstraction. Thus, the process of image analysis is very specific, and developed algorithms can rarely be transferred directly into other domains of applications.

4. *Image management* encompasses all the techniques that provide the efficient storage, communication, transmission, archiving, and access (retrieval) of image data. A simple greyscale radiograph in its original condition may require several megabytes of storage capacity and compression techniques are applied. The methods of telemedicine are also a part of image management.
5. *Image enhancement* is different from image analysis, and is also referred to as high-level image processing, low-level processing, or image enhancement; and denotes manual or automatic techniques, which can be realized without a priori knowledge on the specific content of images. This type of algorithm has similar effects regardless of what is shown in an image.

The complexity of an algorithm, the difficulty of its implementation, or the computing time required for image processing plays a secondary role in the distinction between low-level and high-level processing methods. Rather, the degree of abstraction of the a priori knowledge is important for determining the distinction (Figure 3.8):

- The *raw data level* records an image as a whole. Therefore, the totality of all data pixels is regarded on this level.
- The *pixel level* refers to discrete individual pixels or points.
- The *edge level* represents the one-dimensional (1D) structures, which are composed of at least two neighboring pixels.
- The *texture level* refers to two-dimensional (2D) structures. On this level however, the delineation of the area's contour may be unknown.
- The *region level* describes 2D structures with a well-defined boundary.
- The *object level* associates textures or regions with a certain meaning or name, that is, semantics is given on this level.
- The *scene level* considers the ensemble of image objects in spatial and/or temporal terms.

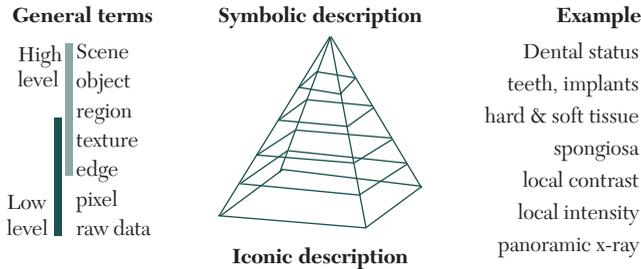


FIGURE 3.8. Levels of abstraction for image processing.

From an iconic (concrete) to a symbolic (abstract) description of images, information is gradually reduced. Methods of low-level image processing operate on the raw data as well as a pixel, edge, or texture levels, and thus at a minimal level of abstraction. Methods of high-level image processing include the texture, region, object, and scene levels. The required abstraction can be achieved by an increased modeling of a priori knowledge.

With these definitions, a particular problem in high-level processing of medical images is immediately apparent: resulting from its complex nature, it is difficult to formulate medical a priori knowledge such that it can be integrated directly and easily into automatic algorithms of image processing. This is referred to as the *semantic gap*, which means the discrepancy between the cognitive interpretation of a diagnostic image by the physician (high level) and the simple structure of discrete pixels, which is used in computer programs to represent an image (low level).

In the medical domain, there are four main factors that make bridging the semantic gap difficult:

1. *Low image quality*: Most imaging modalities that are used in diagnostics or therapy are harmful to the human body. Therefore, images are taken with low energy or dose, and the signal to noise ratio is rather bad.
2. *Heterogeneity of images*: Medical images display organs or body parts. Even if captured with the same modality and following a standardized acquisition protocol, the shape, size, and internal structures of these objects may vary remarkably not only from patient to patient (inter subject variation) but also among different views of a certain patient and similar views of the same patients at different times (intra subject variation). In other words, biological structures are subject to both inter and intra in-

dividual alterability. Thus, universal formulation of a priori knowledge is impossible.

3. *Unknown delineation of objects*: Frequently, biological structures cannot be separated from the background because the diagnostically or therapeutically relevant object is represented by the entire image. Even if definable objects are observed in medical images, their segmentation is problematic because the shape or borderline itself is fuzzy or only partly represented. Hence, medically related items often can be abstracted at most on the texture level.
4. *Robustness of algorithms*: In addition to these inherent properties of medical images, which complicate their high-level processing, special requirements for reliability and robustness of medical procedures and, when applied in routine, image processing algorithms are also demanded in the medical area. As a rule, automatic analysis of images in medicine shouldn't provide wrong measurements. This means that images which cannot be processed correctly must automatically be rejected. Consequently, all images that are not rejected must be evaluated correctly. Furthermore, the number of rejected images must be quite small, since most medical imaging procedures are harmful and cannot be repeated just because of image processing errors.

Historically, endoscope systems only displayed unenhanced, low-resolution images. Physicians had to interpret the images they saw based solely on knowledge and experience. The low-quality images and subjective nature of the diagnoses inevitably resulted in overlooked abnormalities and incorrect treatments. Today, endoscope systems are armed with high-resolution optics, image sensors, and embedded vision processing capabilities. They can distinguish among tissues, enhance edges, and other image attributes; perform basic dimensional analytics (e.g., length, angle); and overlay this data on top of the video image in real time. Advanced designs can even identify and highlight unusual image features, so physicians are unlikely to overlook them.

Figure 3.9 shows leading-edge endoscope systems not only output high-resolution images, but also enhance and information-augment them to assist in physician analysis and diagnosis. In ophthalmology, doctors historically relied on elementary cameras that only took pictures of the inner portions of the eye. Subsequent analysis was left to the healthcare professional. Today, ophthalmologists use medical devices armed with



FIGURE 3.9. Endoscope.



FIGURE 3.10. Robotic surgery.

embedded vision capabilities to create detailed 2-D and 3-D models of the eye in real time as well as overlay analytics such as volume metrics and the dimensions of critical ocular components. With ophthalmological devices used for cataract or lens correction surgery preparation, for example, embedded vision processing helps differentiate the cornea from the rest of the eye. It then calculates an overlay, complete with surgical cut lines, based on various dimensions it has ascertained. Physicians now have a customized operation blueprint, which dramatically reduces the likelihood of mistakes. Such data can even be used to guide human-assisted or fully automated surgical robots with high precision. Figure 3.10 shows robust vision capabilities are essential to robotic surgery systems, whether they be human-assisted (either local or remote) or fully automated.

Another example of how embedded vision can enhance medical devices involves advancements in clinical testing. In the past, it took days, if not weeks, to receive the results of blood and genetic tests. Today, more accurate results are often delivered in a matter of hours. DNA sequencers use embedded vision to accelerate analysis by focusing in on particular areas of a sample. After DNA molecules and primers are added to a slide, the sample begins to group into clusters. A high-resolution camera then scans the sample, creating numerous magnified images. After stitching these images together, the embedded vision enhanced system identifies the clusters, along with their density parameters. These regions are then subjected to additional chemical analysis to unlock their DNA attributes. This method of visually identifying clusters before continuing the process drastically reduces procedure times and allows for more precise results. Faster blood or genetic test results enables quicker treatment and improves healthcare.

Magnifying Minute Variations

Electronic systems are also adept at detecting and accentuating minute image-to-image variations that the human visual system is unable to perceive, whether due to insufficient sensitivity or inadequate attention. It is possible to accurately measure pulse rate simply by placing a camera in front of a person and logging the minute facial-color change cycles that are reflective of capillary blood flow. Figure 3.11 shows embedded vision systems are capable of discerning (and amplifying) the scant frame-to-frame color changes in a subject's face,



FIGURE 3.11. Blood flow analysis from image.

suggestive of blood flow. Similarly, embedded vision systems can precisely assess respiration rate by measuring the periodic rise and fall of the subject's chest.

This same embedded vision can be used to provide early indication of neurological disorders such as amyotrophic lateral sclerosis (also known as Lou Gehrig's disease) and

Parkinson's disease. Early warning signs such as minute trembling or aberrations in gait so slight that they may not yet even be perceptible to the patient are less likely to escape the perceptive gaze of an embedded-vision enabled medical device.

Microsoft Kinect game console peripheral (initially harnessed for image-processing functions such as a gesture interface and facial detection and recognition) is perhaps the best-known embedded vision product. Microsoft began shipping an upgraded Kinect for Windows that includes Fusion, a feature that transforms Kinect-generated images into 3-D models. Microsoft demonstrated the Fusion algorithm by transforming brain scans into 3-D replicas of subjects' brains, which were superimposed onto a mannequin's head and displayed on a tablet computer's LCD screen.

Gesture and Security Enhancements

The Kinect and its 2-D- and 3-D-sensor counterparts have other notable medical uses. Microsoft released "The Kinect Effect," a video showcasing a number of Kinect for Windows applications. One showed a surgeon in the operating room flipping through LCD-displayed X-ray images simply by gesturing with his hand in the air. A gesture interface is desirable in at least two scenarios: when the equipment to be manipulated is out of

arm's reach (thereby making conventional buttons, switches, and touch screens infeasible) and when sanitary concerns prevent tactile control of the gear. Figure 3.12 shows gesture interfaces are useful in situations where the equipment to be controlled is out of arm's reach and when sanitary or other concerns preclude tactile manipulation of it.



FIGURE 3.12 Gesture interface.

Embedded vision intelligence can also ensure medical facility employees follow adequate sanitary practices. Conventional video capture equipment, such as that found in consumer and commercial settings, records constantly while it is operational, creating an abundance of wasted content that must be viewed in its entirety in order to pick out footage of note. An intelligent video monitoring system is a preferable alternative; it can employ motion detection to discern when an object has entered the frame and use facial detection to confirm the object is a person. Recording will only continue until the person has exited the scene. Subsequent review either by a human operator or, increasingly commonly, by the computer itself will assess whether adequate sanitary practices have been followed in each case.

3.3 MATHEMATICS, ALGORITHMS IN MEDICAL IMAGING

Medical imaging needs highly trained technicians and clinicians to determine the details of image acquisition such as choice of modality, patient position, optional contrast agent, and so forth. Imaging also analyzes the results. Artificial systems must be designed to analyze medical datasets either in a partially or even a fully automatic manner. This is a challenging application of the field known as artificial vision. Such algorithms are based on mathematical models. In medical image analysis, as in many practical mathematical applications, numerical simulations should be regarded as the end product. The purpose of the mathematical analysis is to guarantee that the constructed algorithms will behave as desired.

Artificial Intelligence (AI)

Artificial Intelligence (AI) was initiated as a field in the 1950s with the ambitious goal of creating artificial systems with human-like intelligence.

Classical AI had been mostly concerned with symbolic representation and reasoning. In particular, artificial vision emerged in the 1970s with the more limited goal to mimic human vision with man-made systems. A chess-playing program is not directly modeled after a human player, many mathematical techniques are employed in artificial vision that do not pretend to simulate biological vision. Artificial vision systems will therefore not be set within the natural limits of human perception. For example, human vision is inherently two dimensional. To accommodate this limitation, radiologists must resort to visualizing only 2D planar slices of 3D medical images. An artificial system is free of that limitation and can “see” the image in its entirety. Other advantages are that artificial systems can work on very large image datasets, are fast, do not suffer from fatigue, and produce repeatable results.

Many mathematical approaches have been investigated for applications in artificial vision (e.g., fractals and self-similarity, wavelets, pattern theory, stochastic point process, random graph theory). In particular, methods based on partial differential equations (PDEs) have been extremely popular in the past few years.

Medical images typically suffer from one or more of the following imperfections:

- low resolution (in the spatial and spectral domains);
- high level of noise;
- low contrast;
- geometric deformations; and
- the presence of imaging artifacts.

These imperfections can be inherent to the imaging modality. For example, X-rays offer low contrast for soft tissues, ultrasound produces very noisy images, and metallic implants will cause imaging artifacts in MRI. Finer spatial sampling may be obtained through a longer acquisition time. However that would also increase the probability of patient movement and thus blurring.

Several tasks can be performed (semi)-automatically to support the eye-brain system of medical practitioners. Smoothing is the problem of simplifying the image while retaining important information. Registration is the problem of fusing images of the same region acquired from

different modalities or putting in correspondence images of one patient at different times or of different patients. Finally, segmentation is the problem of isolating anatomical structures for quantitative shape analysis or visualization. The ideal clinical application should be fast, robust with regards to image imperfections, simple to use, and as automatic as possible. The ultimate goal of artificial vision is to imitate human vision, which is intrinsically subjective.

Image smoothing is the action of simplifying an image while preserving important information. The goal is to reduce noise or useless details without introducing too much distortion so as to simplify subsequent analysis.

Image registration is the process of bringing two or more images into spatial correspondence, in other words aligning them. In the context of medical imaging, image registration allows for the concurrent use of images taken with different modalities (e.g., MRI and CT), at different times or with different patient positions. In surgery, for example, images are acquired before (pre-operative), as well as during (intra-operative) surgery. Because of time constraints, the real-time intra-operative images have a lower resolution than the pre-operative images obtained before surgery. Moreover, deformations which occur naturally during surgery make it difficult to relate the high-resolution pre-operative image to the lower-resolution intra-operative anatomy of the patient. Image registration attempts to help the surgeon relate the two sets of images.

Registration typically proceeds in several steps. First, one decides how to measure similarity between images. One may include the similarity among pixel intensity values, as well as the proximity of predefined image features such as implanted fiducials or anatomical landmarks. Next, one looks for a transformation which maximizes similarity when applied to one of the images.

When looking at an image, a human observer cannot help seeing structures that often may be identified with objects. However, digital images as the raw retinal input of local intensities are not structured. Image segmentation is the process of creating a structured visual representation from an unstructured one. In its modern formulation, image segmentation is the problem of partitioning an image into homogeneous regions that are semantically meaningful. That is, it corresponds to objects one can identify. Segmentation is not concerned with actually determining what the partitions are. In that sense, it is a lower-level problem than object recognition.

In the context of medical imaging, these regions have to be anatomically meaningful. A typical example is partitioning an MRI image of the brain into the white and grey matter. Since it replaces continuous intensities with discrete labels, segmentation can be seen as an extreme form of smoothing/information reduction. Segmentation is also related to registration in the sense that if an atlas can be perfectly registered to a dataset at hand, then the registered atlas labels are the segmentation. Segmentation is useful for visualization, it allows for quantitative shape analysis, and provides an indispensable anatomical framework for virtually any subsequent automatic analysis. There are basically two dual approaches. In the first, one can start by considering the whole image to be the object of interest, and then refine this initial guess. These “split and merge” techniques can be thought of as somewhat analogous to the top-down processes of human vision. In the other approach, one starts from one point assumed to be inside the object, and adds other points until the region encompasses the object. Those are the “region growing” techniques and bear some resemblance to the bottom-up processes of biological vision.

The dual problem with segmentation is that of determining the boundaries of the segmented homogeneous regions. This approach has been popular for some time since it allows one to build upon the well-investigated problem of edge detection. Difficulties arise with this approach because noise can be responsible for spurious edges. Another major difficulty is that local edges need to be connected into topologically correct region boundaries. To address these issues, it has been proposed to set the topology of the boundary to that of a sphere and then deform the geometry in a variational framework to match the edges.

Computer-Aided Diagnostic Processing

Medical image processing deals with the development of problem-specific approaches to the enhancement of raw medical image data for the purposes of selective visualization as well as further analysis. Image segmentation is defined as a partitioning of an image into regions that are meaningful for a specific task; it is a labeling problem. This may, for instance, involve the detection of a brain tumor from MR or CT images. Segmentation is one of the first steps leading to image analysis and interpretation. The goal is easy to state, but difficult to achieve accurately.

A 3D image of brain tissue, major blood vessels, and a tumor from an MRI is shown in Figure 3.13. This allows surgeons to visualize the actual lo-

cation and to plan and simulate specific procedures. Image segmentation approaches can be classified according to both the features and the type of techniques used. Features include pixel intensities, edge information, and texture. Techniques based on these features can be broadly classified into structural and statistical methods.

Structural methods are based on the spatial properties of the image, such as edges and regions. Various edge detection algorithms have been applied to extract boundaries between different brain tissues. However such algorithms are sensitive to artifacts and noise. Region growing is another popular structural technique. In this approach, one begins by dividing an image into small regions, which can be considered as “seeds.” Then, all boundaries between adjacent regions are examined. Strong boundaries (in terms of certain specific properties) are kept, while weak boundaries are rejected and the adjacent regions merged. The process is carried out iteratively until no boundaries are weak enough to be rejected. However, the performance of the method depends on seed selection and whether the regions are well defined, and therefore is also not considered robust.

Starting from a totally different viewpoint, statistical methods label pixels according to probability values, which are determined based on the intensity distribution of the image. Grey-level threshold is the simplest, yet often effective, segmentation method. In this approach, structures in the image are assigned a label by comparing their grey-level value to one or more intensity thresholds. A single threshold serves to segment the image into only two regions: a background and a foreground. Sometimes the task of selecting a threshold is quite easy, when there is a clear difference between the grey-levels of the objects we wish to segment.

However, things are not normally so simple. Inhomogeneity in the imaging equipment and the partial-volume effect (multiple tissue class occupation within a voxel) give rise to a smoothly varying, nonlinear gain field. While the human visual system easily compensates for this field, the gain can perturb the histogram distributions, causing significant overlaps

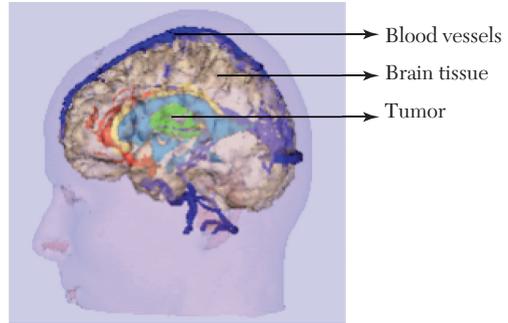


FIGURE 3.13. A 3D rendering of a segmented image from an MRI.

between intensity peaks and thus leading to substantial misclassification in traditional intensity-based classification methods.

Hence, there are more sophisticated statistical approaches relying on certain assumptions or models of the probability distribution function of the image intensities and its associated class labels, which can both be considered random variables. Let C and Y be two random variables for the class label and the pixel intensity, respectively, and c and y be typical instances. The class-conditional density function is $p(y|c)$. Statistical approaches attempt to solve the problem of estimating the associated class label c , given only the intensity y for each pixel. Such an estimation problem is necessarily formulated from an established criterion. Maximum a posteriori (MAP) or maximum likelihood (ML) principles are two such examples. Many statistical segmentation methods differ in terms of models of $p(y)$. Depending on whether a specific functional form for the density model is assumed, a statistical approach can either be *parametric* or *nonparametric*. Both have been widely used in segmentation of brain MR images.

In nonparametric methods, the density model $p(y)$ relies entirely on the data itself, that is, no prior assumption is made about the functional form of the distribution but a large number of correctly labeled training points are required in advance.

One of the most widely used nonparametric methods is the K -nearest-neighbors (KNN) rule classification. First fix K , the number of nearest neighbors to find in the neighborhood of any unlabeled intensity y . The KNN involves finding a neighborhood around the point y that contains K points independent of their class, and then assigning y to the class having the largest number of representatives inside the neighborhood. When $K = 1$, it becomes the nearest-neighbor rule, which simply assigns a point y to the same class as that of the nearest point from the training set.

Nonparametric methods are adaptive, but suffer from the difficulty of obtaining a large number of training points, which can be tedious and a heavy burden even for experienced people. Clearly, such methods are not fully automatic. Unlike nonparametric approaches, parametric approaches rely on an explicit functional form of the intensity density function. For instance, the intensity density function can be modeled by a sum of Gaussian distributions, each of which models an intensity distribution of each class. Here, means and variances of Gaussian distributions becomes parameters

of the model to be determined. Maximum likelihood (ML) methods aim at estimating a set of parameters that maximize the probability of observing the grey-level distribution. A good method to estimate these parameters is to use an expectation-maximization (EM) approach, an iterative technique.

Vision Algorithms for Biomedical

Vision algorithms typically require high compute performance, and, of course, embedded systems of all kinds are usually required to fit into tight cost and power consumption envelopes. In other digital signal processing application domains, such as wireless communications and compression-centric consumer video equipment, chip designers achieve this challenging combination of high performance, low cost, and low power by using specialized coprocessors and accelerators to implement the most demanding processing tasks in the application. These coprocessors and accelerators are typically not programmable by the chip user, however.

This trade-off is often acceptable in applications where software algorithms are standardized. In vision applications, however, there are no standards constraining the choice of algorithms. On the contrary, there are often many approaches to choose from to solve a particular vision problem. Therefore, vision algorithms are very diverse, and change rapidly over time. As a result, the use of nonprogrammable accelerators and coprocessors is less attractive.

Achieving the combination of high performance, low cost, low power, and programmability is challenging. Although there are few chips dedicated to embedded vision applications today, these applications are adopting high-performance, cost-effective processing chips developed for other applications, including DSPs, CPUs, FPGAs, and GPUs. Particularly demanding embedded vision applications often use a combination of processing elements. As these chips and cores continue to deliver more programmable performance per dollar and per watt, they will enable the creation of more high-volume embedded vision products. Those high-volume applications, in turn, will attract more attention from silicon providers, who will deliver even better performance, efficiency, and programmability.

Real-Time Radiography

A typical system of real-time radiography (RTR) is shown in Figure 3.14. The main components of a real-time radiograph system are

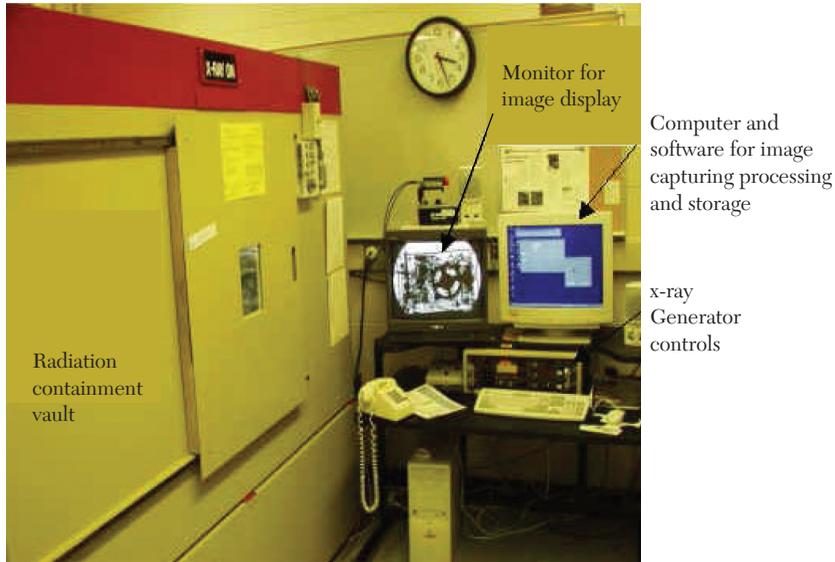


FIGURE 3.14. Photograph of an example of a real-time radiograph system.

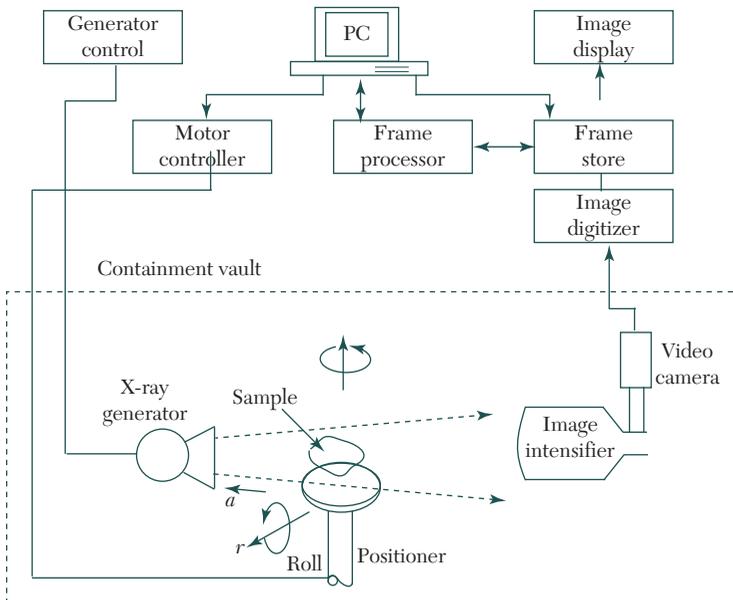


FIGURE 3.15. Schematic showing the various components of a real-time system.

shown in Figure 3.15. The usual source of radiation for RTR systems is the X-ray generator. The main reason that an X-ray generator is used is because image intensifiers are relatively inefficient at converting radiation to light, and thus more flux is required than an isotope can offer. Additionally, X-ray energy and current must be adjustable to permit the correct exposure. (Time cannot be adjusted in this instance because the image is being viewed in real time.) With radioactive isotope sources there is also the disadvantage that few wavelengths of different energies exist, and therefore the beam tends to be monochromatic compared with the beam from an X-ray machine.

As has just been established, a number of factors can adversely affect RTR image quality. With the use of image enhancement techniques, the difference in sensitivity between film and RTR can be decreased. A number of image processing techniques, in addition to enhancement techniques, can be applied to improve the data usefulness. Techniques include convolution edge detection, mathematics, filters, trend removal, and image analysis. The following are the computer software programs,

Enhancement programs make information more visible.

- Histogram equalization—Redistributes the intensities of the image of the entire range of possible intensities (usually 256 greyscale levels).
- Unsharp masking—Subtracts smoothed image from the original image to emphasize intensity changes.

Convolution programs are 3-by-3 masks operating on pixel neighborhoods.

- High-pass filter—Emphasizes regions with rapid intensity changes.
- Low-pass filter—Smooth images, blurs regions with rapid changes.

Math processes programs perform a variety of functions.

- Add images—Adds two images together, pixel-by-pixel.
- Subtract images—Subtracts second image from first image, pixel by pixel.
- Exponential or logarithm—Raises e to power of pixel intensity or takes log of pixel intensity. Nonlinearly accentuates or diminishes intensity variation over the image.

- Add, subtract, multiply, or divide—Applies the same constant values as specified by the user to all pixels, one at a time. Scales pixel intensities uniformly or nonuniformly
- Dilation—Morphological operation expanding bright regions of image.
- Erosion—Morphological operation shrinking bright regions of image.

Noise filters decrease noise by diminishing statistical deviations.

- Adaptive smoothing filter—Sets pixel intensity to a value somewhere between original value and mean value corrected by degree of noisiness. Good for decreasing statistical, especially single-dependent noise.
- Median filter—Sets pixel intensity equal to median intensity of pixels in neighborhood. An excellent filter for eliminating intensity spikes.
- Sigma filter—Sets pixel intensity equal to mean of intensities in neighborhood within two of the mean. Good filter for signal-independent noise.

Trend removal programs remove intensity trends varying slowly over the image.

- Row column fit—Fits image intensity along a row or column by a polynomial and subtract fit from data. Chooses row or column according to direction that has the least abrupt changes.

Edge detection programs sharpen intensity-transition regions.

- First difference—Subtracts intensities of adjacent pixels. Emphasizes noise as well as desired changes.
- Sobel operator—3-by-3 mask weighs inner pixels twice as heavily as corner values. Calculates intensity differences.
- Morphological edge detection—Finds the difference between dilated (expanded) and eroded (shrunk) version of image.

Image analysis programs extract information from an image.

- Grayscale mapping—Alters mapping of intensity of pixels in file to intensity displayed on a computer screen.
- Slice—Plots intensity versus position for horizontal, vertical, or arbitrary direction. Lists intensity versus pixel location from any point along the slice.

- Image extraction—Extracts a portion or all of an image and creates a new image with the selected area.
- Images statistics—Calculates the maximum, minimum, average, standard deviation, variance, median, and mean-square intensities of the image data.

Image Compression Technique for Telemedicine

There are many areas where algorithms are implemented in medical embedded vision. Here some of them are discussed.

Many classes of images contain spatial regions that are more important than other regions. Compression methods capable of delivering higher reconstruction quality for important parts are attractive in this situation. For medical images, only a small portion of the image might be diagnostically useful, but the cost of a wrong interpretation is high. Hence, region-based coding (RBC) technique is significant for medical image compression and transmission. Lossless compression schemes with secure transmission play a key role in telemedicine applications that help in accurate diagnosis and research. The compressed image can be accessed and sent over telemedicine network using personal digital assistance (PDA) like mobile. A large amount of image data is produced in the field of medical imaging in the form of computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound images, which can be stored in picture archiving and communication system (PACS) or hospital information system.

A medium-scale hospital with previously mentioned facilities produces on an average 5 GB to 15 GB of data. Therefore, it's really difficult for hospitals to manage storing facilities of that size. Moreover, such high data demand a high-end network, especially for transmitting the images over the network such as in telemedicine. This is significant for a telemedicine scenario due to limitations of transmission medium in information and communication technology (ICT), especially in rural areas. Image compression is useful in reducing the storage and transmission bandwidth requirements of medical images. For example, an 8-bit grey-scale image with 512×512 pixels requires more than 0.2 MB of storage. If the image is compressed by 8:1 compression without any perceptual distortion, the capacity of storage increases 8 times. Compression methods are classified into lossless and lossy methods. In the medical imaging scenario, lossy compression schemes are not generally used. This is due to possible loss of useful clinical information that may influence diagnosis. In addition

to these reasons, there can be legal issues. Storage of medical images is generally problematic because of the requirement to preserve the best possible image quality, which is usually interpreted as a need for lossless compression.

A 3D MRI contains multiple slices representing all information required about a body part. Some of the most desirable properties of any compression method for 3D medical images include: (a) high lossless compression ratios, (b) resolution scalability, which refers to the ability to decode the compressed image data at various resolutions, and (c) quality scalability, which refers to the ability to decode the compressed image at various qualities or signal-to-noise ratios (SNR) up to lossless reconstruction. DICOM is the most comprehensive and accepted version of an imaging communications standard. DICOM format has a header which contains information about the image, imaging modality and information about the patient. The header also contains information about the type of media (CT, MRI, audio recording, etc.) and the image dimensions. Body of DICOM standard contains information objects such as medical reports, audio recordings, and images. The coding–decoding algorithm must take care of other information in the DICOM file. Also, the algorithms should accept the input image in DICOM format at encoder end and produce DICOM file at decoder end.

Region of Interest

Basic concept of region of interest (ROI) is introduced due to limitations of lossy and lossless compression techniques. For well-known lossless compression technique the compression ratio is approximately 25% of original size, whereas for lossy encoders the compression ratio is much higher (up to 1% also), but there is loss in the data. Now this loss may hamper some diagnostically important part of the image. Hence, there is a need of some hybrid technique which will take care of diagnostically important part (ROI) as well as will provide high compression ratio. The functionality of ROI is important in medical applications where certain parts of the image are of higher diagnostic importance than others. For most medical images, the diagnostically significant information is localized over relatively small regions, about 5%–10% of total area. In such cases, these regions need to be encoded at higher quality than the background. During image transmission for telemedicine purposes, these regions are required to be transmitted first or at a higher priority.

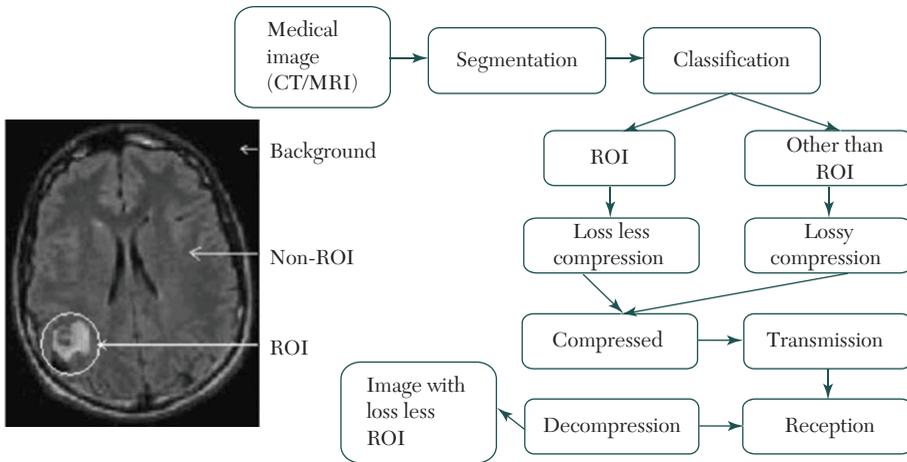


FIGURE 3.16. Different parts of the medical image and Block diagram of the image compression method.

A CT or MRI image contains three parts: ROI (the diagnostically important part), non-ROI image part, and the background (part other than image contents) as shown in Figure 3.16. The ROI is selected by expert radiologists. Depending on the selected part, ROI-mask is generated in such a way that the foreground is totally included and the pixel values in the background are made zero. Although the background regions appear to be black, they do not have zero grey-level values.

The two separated parts can be processed separately as per the requirement, that is, ROI part will be processed by lossless technique, while Non-ROI will be compressed with accepted lossy compression methods, as shown in Figure 3.16.

Structure Sensitive Adaptive Contrast Enhancement Methods

- For medical images, the overall goal of display is the detection, localization, and qualitative characterization of anatomical objects represented by the intensity variations in the recorded data.
- Global histogram equalization is justified by the argument that for noise-free images, it maximally transmits information as to scene intensity values.
- Adoptive histogram equalization has demonstrated its effectiveness in the display of images from a wide range of imaging modalities, including CT, MRI, and radiotherapy portal films.

- Gordon's technique is another method for contrast-based enhancement for the detection of edges within the contextual region.

LSPIHT Algorithm for ECG Data Compression and Transmission

Layered set partitioning in hierarchical trees (LSPIHT) algorithm is used for medical ECG data compression and transmission. In the LSPIHT, the encoded bit streams are divided into a number of layers for transmission and reconstruction. Starting from the base layer, by accumulating bit streams up to different enhancement layers, medical data can be reconstructed with various signal-to-noise ratios (SNRs) and resolutions. Receivers with distinct specifications can then share the same source encoder to reduce the complexity of telecommunication networks for telemedicine applications. The algorithm allows the compression ratio (CR) at each layer to be prespecified. Therefore, the SNR scalability can be attained if the bit stream is delivered in layer resolution component position order.

In the LSPIHT, the CR and resolution associated with each layer can be prespecified before encoding. The transmitted images are reconstructed with CR and resolution identical to those of the highest layer accumulated by the decoder. Both the SNR and resolution scalabilities therefore can be achieved. To satisfy both CR and resolution constraints at each layer, starting from the base layer, the LSPIHT encodes one layer at a time until the design of the top layer is completed. The encoding of each layer is based on SPIHT which only covers the sub-bands with resolution lower than the resolution constraint of that layer. To enhance the performance of SPIHT at each layer, the encoding results of the previous layers are used. The LSPIHT is applied to images of recorded ECGs for scalable transmission, where different layers are associated with distinct CRs and resolutions. The LSPIHT algorithm attains better rate distortion performance for the encoding of each layer while demanding fewer resources and lower costs. The wavelet-based embedded coding algorithms can be employed for realizing scalable systems. Some of these algorithms, such as JPEG2000 6 and the set partitioning in hierarchical trees (SPIHT) algorithm have been found to outperform many existing methods for image compression.

Retrieval of Medical Images in a PACS

PACS (picture archiving and communication system) has the ability to integrate all the information of patients such as textual descriptions, images, graphs, temporal data, and so on into one system. As PACS organize and allow the distribution and communication of the images from exams of the

patients, it can be a valuable tool to aid diagnosis. This is even stronger, if the physicians can search and retrieve images based only on the images themselves, not depending on a textual annotation or description of the images. This is due to the fact that descriptions can be biased, as the specialist can be interested, at the moment, in a particular aspect of the image and other details may be overlooked. The amount of image data has increased tremendously over the past years, with a tendency to grow even faster, because of the diminishing computational costs of the devices. Therefore, image search and retrieval systems should depend more and more on automatic processing. The area of content-based image retrieval (CBIR) has also grown. The current development of PACS intends to incorporate such facilities, that is, retrieval of images by content and similarity search also over the image content.

An approach for extraction of characteristics based on color intensity, called metric histogram (MH) is available. The MH keeps the original behavior of the image brightness distribution, given by traditional histograms, without loss of the information brought by histograms, but with a reduced number of bins, and lower computational cost for search processing. The invariance about brightness transformations is rather important, because images from the same patient acquired in different situations and device settings tend to vary in this regard, what makes harder to automatically retrieve such images. Thus, direct searching of images through traditional histograms would not recover similar images, if they have different quantization ranges. Many hybrid or combination techniques are used for image retrieval.

Digital Signature Realization Process of DICOM Medical Images

In order to ensure the integrity of the patient information, authenticity, and reliability, the safety of telemedicine treatment and the regional PACS (picture archiving and communication systems) transmission, digital signature of ECDSA (elliptic curve digital signature algorithm) in DICOM (digital imaging and communications in medicine) medical images during the rapid development of digital and information in telemedicine and medical care is required. During the rapid development of digital and information in telemedicine and medical care, hospital information systems have become more and more popular and have also been connected with other hospital's information systems. Security of medical information will become more important and prominent, as privacy data of patients is gradually exposed to open environment. During diagnosis and treatment,

a series of health records and medical images are generated, stored, transmitted, and withdrawn, so the security of medical information research has become necessary.

ECDSA (elliptic curve digital signature algorithm) is the discrete logarithm problem based on the point group of elliptic curves in finite fields. Compared to RSA algorithm, it has advantages of small key size, saving bandwidth and storage space. Some characteristics such as data confidentiality, data authentication, data integrity, and source key management are satisfied in the algorithm. This model contains the following information: the roles of the signer, signature property list, the mechanism of producing and validating the signature, and how to identify the signer, other relationships with digital signature, and other factors used to create, calibrate, and explain the signature. In the method of producing and validating the signature, it also contains creating MAC or disorder yards of algorithm and related parameters, the encryption algorithm and parameters, the certificate type or release mechanism, and so on.

Digital signature scheme is an electronic form of storage news signature. A complete digital signature scheme should be made by two parts: the signature algorithm and validated algorithms. Generally speaking, any public key cryptosystems can be used as a digital signature scheme separately. We can sign all tags by using elliptic curve encryption algorithm for digital signature of DICOM files and it can ensure the integrity of the information, but when the message is long, the efficiency of signature is also low.

Computer Neural Networks (CNNs) in Medical Image Analysis

CNNs have been put to task for classification, localization, detection, segmentation, and registration in image analysis. Machine learning research draws a distinction between localization (draw a bounding box around a single object in the image), and detection (draw bounding boxes around multiple objects, which may be from different classes). Segmentation draws outlines around the edges of target objects, and labels them (semantic segmentation). Registration refers to fitting one image (which may be 2- or 3-dimensional) onto another. Localization implies the identification of normal anatomy, for example, where is the kidney in this ultrasound image? This is in contrast to detection, which implies an abnormal, pathological state, for example, where are all the lung tumors in this CT scan of the lung? Segmenting the outline of a lung tumor helps the clinician determine its

distance from major anatomical structures, and helps to answer a question such as, should this patient be operated on, and if so, what should be the extent of resection? Classification is sometimes also known as computer-aided diagnosis (CADx). For example, a CNN to detect lung nodules on chest X-rays used 55 chest X-rays and a CNN with 2 hidden layers to output whether or not a region had a lung nodule (growth of abnormal tissue).

Localization of normal anatomy is less likely to interest the practicing clinician although applications may arise in anatomy education. Alternatively, localization may find use in fully automated end-to-end applications, whereby the radiological image is autonomously analyzed and reported without any human intervention. Detection, sometimes known as computer-aided detection (CADe) is a keen area of study as missing a lesion on a scan can have drastic consequences for both the patient and the clinician. The task involved the detection of cancerous lung nodules on CT lung scans. CT and MRI image segmentation research covers a variety of organs such as liver, prostate, and knee cartilage, but a large amount of work has focused on brain segmentation, including tumor segmentation. The latter is especially important in surgical planning to determine the exact boundaries of the tumor in order to direct surgical resection. Sacrificing too much of eloquent brain areas during surgery would cause neurological deficits such as limb weakness, numbness, and cognitive impairment.

Although the registration of medical images has many potential applications. Their actual clinical use is encountered in niche areas. Image registration is employed in neurosurgery or spinal surgery, to localize a tumor or spinal bony landmark, in order to facilitate surgical tumor removal or spinal screw implant placement. A reference image is aligned to a second image, called a sense image and various similarity measures and reference points are calculated to align the images, which can be 2- or 3-dimensional. The reference image may be a pre-operative MRI brain scan and the sense image may be an intra-operative MRI brain scan done after a first-pass resection, to determine if there is remnant tumor and if further resection is required.

Deep Learning and Big Data

Big data is going to play an essential role in future medicine. The availability of medical devices and their resolution are growing rapidly, resulting in a fast increase of the size of medical data, which requests innovative solutions to process these data in reasonable time-scales. One

of these solutions could be new-generation PACS using distributed data storage and processing. The map reduce programming model could be especially useful for medical imaging, since this paradigm implies moving code to data instead of vice versa, and thus eliminates the necessity to transfer large data volumes through the storage system. Besides, it was designed for being deployed on inexpensive clusters using commodity hardware. The practical implementation of distributed PACS based on the Map Reduce model could be done using recent developments in open-source software, namely, the Hadoop project and the associated frame works. Such systems could provide means for storing terabytes and petabytes of medical imaging data and for parallel fault-tolerant processing of these data. These tools will form the core of future medical informatics software as the quantity and quality of medical data are continuously increasing.

The original concept of precision medicine involves the prevention and treatment strategies that consider individual variability by assessing large sets of data, including patient information, medical imaging, and genomic sequences. The success of precision medicine is largely dependent on robust quantitative biomarkers. In general, deep learning can be used to explore and create quantitative biomarkers from medical big data obtained through Internet of things, genetics and genomics, medicinal imaging, and mobile nitoring sources as shown in Figure 3.17. In particular, imaging is

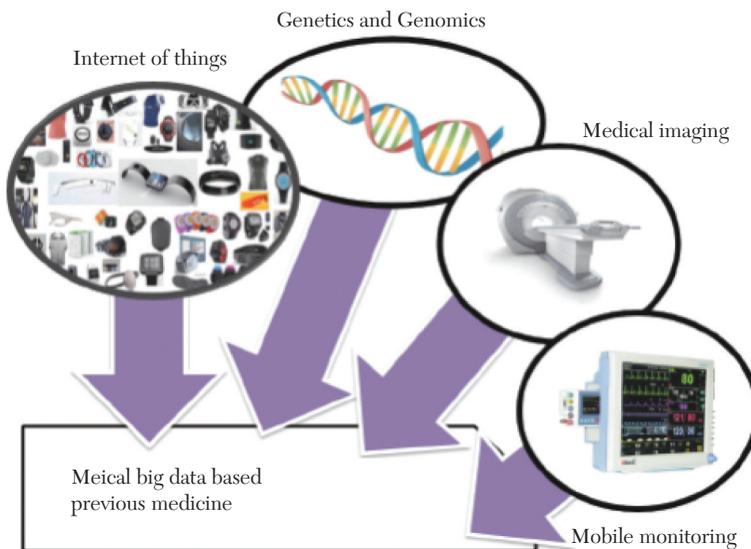


FIGURE 3.17. Precision medicine based on medical big data, including Internet of things, genetics and genomics, medicinal imaging, and mobile monitoring.

noninvasively and routinely performed for clinical practice, and can be used to compute quantitative imaging biomarkers. Deep learning techniques can be used to generate more reliable imaging biomarkers for precision medicine.

3.4 MACHINE LEARNING IN MEDICAL IMAGE ANALYSIS

Machine learning algorithms have the potential to be invested deeply in all fields of medicine, from drug discovery to clinical decision making, significantly altering the way medicine is practiced. The success of machine learning algorithms at computer vision tasks in the years comes at an opportune time when medical records are increasingly digitalized. There is a myriad of imaging modalities, and the frequency of their use is increasing. One can see that CT, MRI, and PET usage increased 7.8%, 10% and 57% respectively. Modalities of digital medical images include ultrasound (US), X-ray, computed tomography (CT) scans, magnetic resonance imaging (MRI) scans, positron emission tomography (PET) scans, retinal photography, histology slides, and dermoscopy images.

Convolutional Neural Networks

Both the 2-dimensional and 3-dimensional structures of an organ being studied are crucial in order to identify what is normal versus abnormal. By maintaining these local spatial relationships, CNNs are well-suited to perform image recognition tasks. CNNs have been put to work in many ways, including image classification, localization, detection, segmentation, and registration. CNNs are the most popular machine learning algorithm in image recognition and visual learning tasks, due to its unique characteristic of preserving local image relations, while performing dimensionality reduction. This captures important feature relationships in an image (such as how pixels on an edge join to form a line), and reduces the number of parameters the algorithm has to compute, increasing computational efficiency. CNNs are able to take as inputs and process both 2-dimensional images, as well as 3-dimensional images with minor modifications. This is a useful advantage in designing a system for hospital use, as some modalities like X-rays are 2-dimensional while others like CT or MRI scans are 3-dimensional volumes. CNNs and recurrent neural networks (RNNs) are examples of supervised machine learning algorithms, which require significant amounts of training data. Unsupervised learning algorithms have

also been studied for use in medical image analysis. These include auto encoders, restricted Boltzmann machines (RBMs), deep belief networks (DBNs), and generative adversarial networks (GANs).

Currently, CNNs are the most researched machine learning algorithms in medical image analysis. The reason for this is that CNNs preserve spatial relationships when filtering input images. As mentioned, spatial relationships are of crucial importance in radiology, for example, in how the edge of a bone joins with muscle, or where normal lung tissue interfaces with cancerous tissue. As shown in Figure 3.18, a CNN takes an input image of raw pixels, and transforms it via convolutional layers, rectified linear unit (RELU) layers, and pooling layers. This feeds into a final fully connected layer that assigns class scores or probabilities, thus classifying the input into the class with the highest probability.

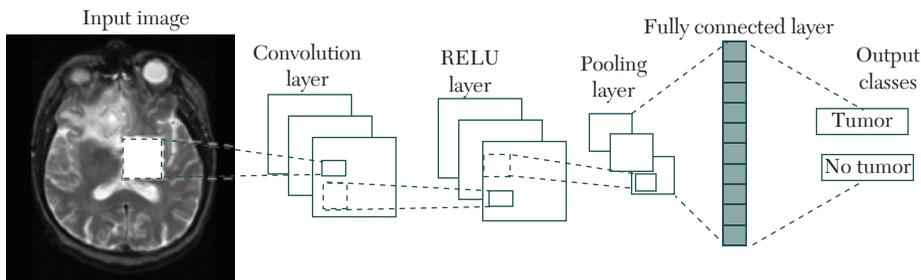


FIGURE 3.18. In this example disease classification task, an input image of an abnormal axial slice of a T2-weighted MRI brain is run through a schematic depiction of a CNN. Feature extraction of the input image is performed via the convolution, RELU and pooling layers, before classification by the fully connected layer.

Convolution Layer

A convolution is defined as an operation on two functions. In image analysis, one function consists of input values (e.g., pixel values) at a position in the image, and the second function is a filter (or kernel); each can be represented as array of numbers. Computing the dot product between the two functions gives an output. The filter is then shifted to the next position in the image as defined by the stride length. The computation is repeated until the entire image is covered, producing a feature (or activation) map. This is a map of where the filter is strongly activated and “sees” a feature such as a straight line, a dot, or a curved edge. If a photograph of a face was fed into a CNN, initially low-level features such as lines and edges are discovered by the filters. These build up to progressively higher features in subsequent layers, such as a nose, eye, or ear, as the feature maps become inputs for the next layer in the CNN architecture.

Convolution exploits three ideas intrinsic to perform computationally efficient machine learning: sparse connections, parameter sharing (or weights sharing), and equivariant (or invariant) representation. Unlike some neural networks where every input neuron is connected to every output neuron in the subsequent layer, CNN neurons have sparse connections, meaning that only some inputs are connected to the next layer. By having a small, local receptive field (i.e., the area covered by the filter per stride), meaningful features can be gradually learnt, and the number of weights to be calculated can be drastically reduced, increasing the algorithm's efficiency. In using each filter with its fixed weights across different positions of the entire image, CNNs reduce memory storage requirements. This is known as parameter sharing. This is in contrast to a fully connected neural network where the weights between layers are more numerous, used once and then discarded. Parameter sharing results in the quality of equivariant representation to arise. This means that input translations result in a corresponding feature map translation.

Rectified Linear Unit (RELU) Layer

The RELU layer is an activation function that sets negative input values to zero. This simplifies and accelerates calculations and training, and helps to avoid the vanishing gradient problem. Other activation functions include the sigmoid, tanh, leaky RELUs, randomized RELUs, and parametric RELUs.

Pooling Layer

The pooling layer is inserted between the convolution and RELU layers to reduce the number of parameters to be calculated, as well as the size of the image (width and height, but not depth). Max-pooling is most commonly used; other pooling layers include average pooling and L2-normalization pooling. Max-pooling simply takes the largest input value within a filter and discards the other values; effectively it summarizes the strongest activations over a neighborhood. The rationale is that the relative location of a strongly activated feature to another is more important than its exact location.

Fully Connected Layer

The final layer in a CNN is the fully connected layer, meaning that every neuron in the preceding layer is connected to every neuron in the fully connected layer. Like the convolution, RELU, and pooling layers, there can be one or more fully connected layers depending on the level of feature

abstraction desired. This layer takes the output from the preceding layer (convolutional, RELU or pooling) as its input, and computes a probability score for classification into the different available classes. In essence, this layer looks at the combination of the most strongly activated features that would indicate the image belongs to a particular class.

For example, on histology glass slides, cancer cells have a high DNA to cytoplasm ratio compared to normal cells. If features of DNA were strongly detected from the preceding layer, the CNN would be more likely to predict the presence of cancer cells. Standard neural network training methods with back propagation and stochastic gradient descent help the CNN learn important associations from training images.

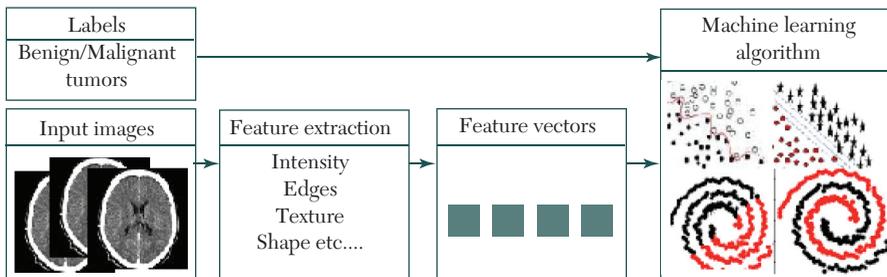
Machine learning is a technique for recognizing patterns that can be applied to medical images. Although it is a powerful tool that can help in rendering medical diagnoses, it can be misapplied. Machine learning typically begins with the machine learning algorithm system computing the image features that are believed to be of importance in making the prediction or diagnosis of interest. The machine learning algorithm system then identifies the best combination of these image features for classifying the image or computing some metric for the given image region. There are several methods that can be used, each with different strengths and weaknesses. There are open-source versions of most of these machine learning methods that make them easy to try and apply to images. Several metrics for measuring the performance of an algorithm exist; however, one must be aware of the possible associated pitfalls that can result in misleading metrics. More recently, deep learning has started to be used; this method has the benefit that it does not require image feature identification and calculation as a first step; rather, features are identified as part of the learning process.

Machine learning is considered a branch of artificial intelligence because it enables the extraction of meaningful patterns from examples, which is a component of human intelligence. The appeal of having a computer that performs repetitive and well-defined tasks is clear: computers will perform a given task consistently and tirelessly; however, this is less true for humans. More recently, machines have demonstrated the capability to learn and even master tasks that were thought to be too complex for machines, showing that machine learning algorithms are potentially useful components of computer-aided diagnosis and decision support systems. Even more exciting is the finding that in some cases, computers seem to be able to “see” patterns that

are beyond human perception. This discovery has led to substantial and increased interest in the field of machine learning specifically, how it might be applied to medical images. Computer-aided detection and diagnosis performed by using machine learning algorithms can help physicians interpret medical imaging findings and reduce interpretation times. These algorithms have been used for several challenging tasks, such as pulmonary embolism segmentation with computed tomography (CT) angiography, polyp detection with virtual colonoscopy or CT in the setting of colon cancer, breast cancer detection and diagnosis with mammography, brain tumor segmentation with magnetic resonance (MR) imaging, and detection of the cognitive state of the brain with functional MR imaging to diagnose neurologic disease (e.g., Alzheimer’s disease).

If a machine learning algorithm is applied to a set of data (in our example, tumor images) and to some knowledge about these data (in our example, benign or malignant tumors), then the algorithm system can learn from the training data and apply what it has learned to make a prediction (in our example, whether a different image is depicting benign or malignant tumor tissue). Figure 3.19 shows that if the algorithm system optimizes its parameters such that its performance improves that is, more test cases are

(a) Training: Iteratively learning until finding the best model to classify benign/malignant tumors



(a) Predicting: Applying the best model to predict a new image

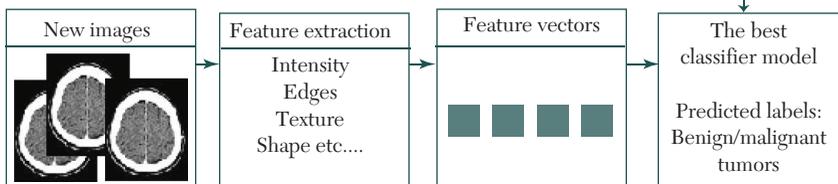


FIGURE 3.19. Machine learning model for medical image.

diagnosed correctly then it is considered to be learning that task. The key terms used in machine learning are given below:

Classification: The assigning of a class or label to a group of pixels, such as those labeled as tumor with use of a segmentation algorithm. For instance, if segmentation has been used to mark some part of an image as “abnormal brain,” the classifier might then try to determine whether the marked part represents benign or malignant tissue.

Model: The set of weights or decision points learned by a machine learning system. Once learned, the model can be assigned to an unknown example to predict which class that example belongs to.

Algorithm: The series of steps taken to create the model that will be used to most accurately predict classes from the features of the training examples.

Labeled data: The set of examples (e.g., images), each with the correct “answer.” For some tasks, this answer might be the correct boundary of a tumor, and in other cases, it might be whether cancer is present or the type of cancer the lesion represents.

Training: The phase during which the machine learning algorithm system is given labeled example data with the answers (i.e., labels) for example, the tumor type or correct boundary of a lesion. The set of weights or decision points for the model is updated until no substantial improvement in performance is achieved.

Validation set: The set of examples used during training. This is also referred to as the training set.

Testing: In some cases, a third set of examples is used for “real-world” testing. Because the algorithm system iterates to improve performance with the validation set, it may learn unique features of the training set. Good performance with an “unseen” test set can increase confidence that the algorithm will yield correct answers in the real world. Note that different groups sometimes use validation for testing and vice versa. This tends to reflect the engineering versus statistical background. Therefore, it is important to clarify how these terms are used.

Node: A part of a neural network that involves two or more inputs and an activation function. The activation function typically sums the inputs and then uses some type of function and threshold to produce an output.

Layer: A collection of nodes that computes outputs (the next layer unless this is the output layer) from one or more inputs (the previous layer unless this is the input layer).

Weights: Each input feature is multiplied by some value, or weight; this is referred to as weighting the input feature. During training, the weights are updated until the best model is found. Machine learning algorithms can be classified on the basis of training styles: supervised, unsupervised, and reinforcement learning. To explain these training styles, consider the task of separating the regions on a brain image into tumor (malignant or benign) versus normal (nondiseased) tissue.

Supervised learning involves gaining experience by using images of brain tumor examples that contain important information specifically, “benign” and “malignant” labels and applying the gained expertise to predict benign and malignant neoplasia on unseen new brain tumor images (test data). In this case, the algorithm system would be given several brain tumor images on which the tumors were labeled as benign or malignant. Later, the system would be tested by having it try to assign benign and malignant labels to findings on the new images, which would be the test dataset. Examples of supervised learning algorithms include support vector machine, decision tree, linear regression, logistic regression, naive Bayes, k -nearest neighbor, random forest, AdaBoost, and neural network methods.

With unsupervised learning, data (e.g., brain tumor images) are processed with a goal of separating the images into groups for example, those depicting benign tumors and those depicting malignant tumors. The key difference is that this is done without the algorithm system being provided with information regarding what the groups are. The algorithm system determines how many groups there are and how to separate them.

Examples of unsupervised learning algorithm systems include K-means, mean shift, affinity propagation, hierarchical clustering, DBSCAN (density-based spatial clustering of applications with noise, Gaussian mixture modeling, Markov random fields, ISODATA (iterative self-organizing data), and fuzzy C-means systems.

Like supervised learning, reinforcement learning begins with a classifier that was built by using labeled data. However, the system is then given unlabeled data, and it tries to further improve the classification by better characterizing these data similar to how it behaves with unsupervised learning. Examples of reinforcement learning algorithm systems is teaching box systems.

Segmentation: The splitting of the image into parts. For instance, with tumor segmentation, this is the process of defining where the tumor starts and stops. However, this does not necessarily include deciding that

what is included in the tumor. The goal in this step is to determine where something starts and stops. This technique is usually used with a classifier that determines that a segment of an image is depicting enhancing tumor and another segment is depicting nonenhancing tumor.

Overfitting: When a classifier that is too specific to the training set is not useful because it is familiar with only those examples, this is known as overfitting (Figure 3.20). In general, the training set needs to contain many more examples above the number of coefficients or variables used by the machine learning algorithm.

Features: The numeric values that represent the example. In the case of medical images, features can be the actual pixel values, edge strengths, variation in pixel values in a region, or other values. One can also use nonimage features such as the age of the patient and whether a laboratory test has positive or negative results. When all of these features are combined for an example, this is referred to as a feature vector, or input vector.

Figure 3.19 (a) shows machine learning model development and application model for medical image classification tasks. For training, the machine learning algorithm system uses a set of input images to identify the image properties that, when used, will result in the correct classification of the image that is, depicting benign or malignant tumor as compared with the supplied labels for these input images. Figure 3.19 (b) for predicting, once the system has learned how to classify images, the learned model is applied to new images to assist radiologists in identifying the tumor type.

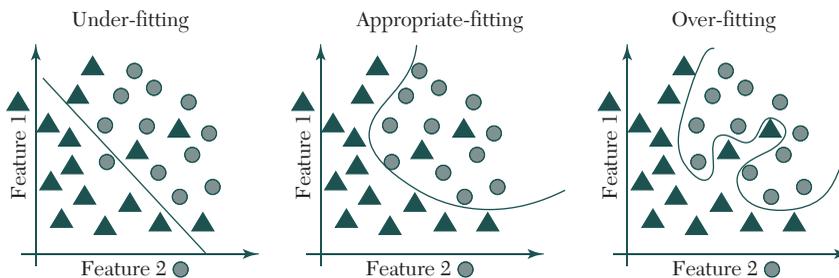


FIGURE 3.20. Diagrams illustrate under- and overfitting. Underfitting occurs when the fit is too simple to explain the variance in the data and does not capture the pattern. An appropriate fit captures the pattern but is not too inflexible or flexible to fit data. Overfitting occurs when the fit is too good to be true and there is possibly fitting to the noise in the data. The axes are generically labeled *feature 1* and *feature 2* to reflect the first two elements of the feature vector.

Feature Computation

The first step in machine learning is to extract the features that contain the information that is used to make decisions. Humans learn important features visually, such as during radiology residencies; however, it can be challenging to compute or represent a feature to assign a numeric value to ground glass texture, for example. Image features should be robust against variations in noise, intensity, and rotation angles, as these are some of the most common variations observed when working with medical imaging data.

Feature Selection

Although it is possible to compute many features from an image, having too many features can lead to overfitting rather than learning the true basis of a decision. The process of selecting the subset of features that should be used to make the best predictions is known as feature selection. One feature selection technique is to look for correlations between features: having large numbers of correlated features probably means that some features and the number of features can be reduced without information being lost. However, in some cases, a more complex relationship exists and evaluating a feature in isolation is dangerous. Suppose, for instance, that you are given a list of weights with binary classifications of whether each weight indicates or does not indicate obesity. One could make some guesses, but adding heights would improve the accuracy: a rather high weight value in conjunction with a low height value is more likely to reflect obesity than is a high weight value in conjunction with a high height value.

Training and Testing: The Learning Process

Supervised machine learning is so named because examples of each type of thing to be learned are required. An important question to ask is “How many examples of each class of the thing do I need to learn it well?” It is easy to see that having too few examples will prevent a computer or a person, for that matter from recognizing those features of an object that allow one to distinguish between the different classes of that object. The exact number of examples in each class that is required depends heavily on how distinctive the classes are.

For instance, if you wish to create an algorithm to separate cars and trucks and you provide a learning algorithm system with an image of a red car labeled “class A” and an image of a black truck labeled “class B,” then using an image of a red truck to test the learning algorithm system may or

may not be successful. If you provide examples of “class A” that include red, green, and black trucks, as well as examples of “class B” that include red, yellow, green, and black cars, then the algorithm system is more likely to separate trucks from cars because the shape features override the color features. Of course, if the person who computed the features used in training did not provide color as an input, then color would not be mistaken as a feature for separating trucks and cars. One popular way to estimate the accuracy of a machine learning system when there is a limited dataset is to use the cross validation technique. With cross validation, one first selects a subset of examples for training and designates the remaining examples to be used for testing. Training proceeds, and the learned state is tested. This process is then repeated, but with a different set of training and testing examples selected from the full set of training examples. In the extreme case, one may remove just one example for testing and use all of the others for each round of training; this technique is referred to as leave one out cross validation. Although cross validation is a good method for estimating accuracy, an important limitation is that each set of training and testing iterations results in a different model, so there is no single model that can be used at the end.

Example of Machine Learning With Use of Cross Validation

Imagine that we wish to separate brain tumor from normal brain tissue and that we have CT images that were obtained without and those that were obtained with contrast material. We have 10 subjects, and 10 regions of interest (ROIs) in normal white matter and 10 ROIs in tumor tissue have been drawn on the CT images obtained in each of these subjects. This means that we have 100 input vectors from white matter and 100 input vectors from tumor, and we will sequence the vectors such that the first value is the mean CT attenuation of the ROI on the noncontrast material enhanced image, and the second value is the mean attenuation of the ROI on the contrast material enhanced image.

With CT of brain tumors, the attenuation values on the nonenhanced images will be similar, though perhaps lower on average for normal brain tissue than for tumors. Enhancing tumor will have higher attenuation on the contrast enhanced images. However, other tissues in the brain, such as vessels, also will enhance. It is also possible that parts of the tumor will not enhance. In addition, although much of the tumor may be darker on the nonenhanced images, areas of hemorrhage or calcification can make the lesion brighter. On the basis of the latter observation, we will also calculate

the variance in attenuation and use this value as the third feature in the vector. To help eliminate vessels, we will calculate the tubularity of the voxels with an attenuation higher than 300 HU (Hounsfield Unit) and store this value as the fourth feature. One can imagine many more values, such as location of the tumor in the head, that might be useful for some tasks, but we will stick with these four features.

We will take 70 of the normal brain tissue ROIs and 70 tumor ROIs and send them to the machine learning algorithm system. The algorithm system will start with random weights for each of the four features and in this simple model add the four products. If the sum is greater than zero, the algorithm system will designate the ROI as tumor; otherwise, the ROI will be designated as normal brain tissue. The algorithm system will do this for all 140 examples. It will then try to adjust one of the weights to see whether this reduces the number of wrong interpretations. The system will keep adjusting weights until no more improvement in accuracy is seen. It will then take the remaining 30 examples of each normal brain tissue ROI and each tumor ROI and evaluate the prediction accuracy; in this example case, let us say that it will designate 50 of these 60 ROIs correctly.

Summary

- Vision imaging technology in medicine made the doctors to see the interior portions of the body for easy diagnosis.
- Medical imaging has four key problems as segmentation, registration, visualization, and simulation.
- An endoscope is a medical instrument used to examine the interior of a hollow organ or a cavity of the body using camera.
- A picture archiving and communication system (PACS) is essentially a network system that allows digital or digitized images from any modality to be retrieved, viewed and analyzed by a relevant expert, or by an appropriate expert system, at different workstations.
- The digital imaging and communications in medicine (DICOM) format allows images and cine-loop images, with associated patient information and reports, including voice notes, to be stored and exchanged readily over the network.
- Ophthalmology is used for analyzing the retina, the optic nerve, and so on. Medical image processing covers five major areas *image formation*,

image visualization, image analysis, image management, and image enhancement.

- Low resolution, high noise, low contrast, geometric deformations, and imaging artifacts presence are the medical image imperfections.
- Image smoothing is the action of simplifying an image while preserving important information.
- Image registration is the process of bringing two or more images into spatial correspondence, that is, aligning them.
- Layered set partitioning in hierarchical trees algorithm is used for medical ECG data compression and transmission.

References

<https://www.embedded-vision.com/embedded-vision-alliance/embedded-vision-revolution>

<https://www.forbes.com/4-ways-cameras-are-changing-healthcare>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3483472/>

<http://spie.org/publications/deserno-medical-image-processing?SSO=1>

Learning Outcomes

- 3.1 Define vision imaging in medicine.
- 3.2 What are the four key problems in medical imaging?
- 3.3 List some advantages of digital processing medical applications.
- 3.4 What are the digital image processing requirements for medical applications?
- 3.5 Write a short note on image processing systems for medical applications.
- 3.6 How are images converted into information in medical vision of endoscopy?
- 3.7 What are the major areas of medical vision image processing?
- 3.8 Write about vision algorithms for bio-medical.
- 3.9 What is the role of mathematics and algorithms in medical imaging?
- 3.10 Write about the algorithm used for ECG data compression.

- 3.11 What is the importance of digital signature in DICOM medical images?
- 3.12 Explain machine learning in medical image analysis.
- 3.13 What are the computer software algorithms implemented in real-time radiography?

Further Reading

1. *Fundamentals of Medical Imaging* by Paul Suetens
2. *The Essential Physics of Medical Imaging* by Jerrold T. Bushberg
3. *Medical Imaging: Principles, Detectors and Electronics* by Krzysztof Iniewski

VIDEO ANALYTICS

Overview

Video analytics is automatic computerized video footage analysis which uses algorithms to differentiate between object types and identify certain action in real time to alert users. The algorithms have sequences such as image acquisition, image preprocessing, segmentation, object analysis, and expert system. Machine learning enables computers to learn pattern found in the object surrounding us. Steps in vision learning are data collection, features designing, training, and model testing.

Learning Objectives

After reading this the reader will know about the

- application of video analytics
- defining metadata search
- machine learning and its types for embedded vision system
- automatic number plate recognition system design
- convolutional neural networks for autonomous cars design requirements
- smart fashion AI architecture explanation
- computers to recognize cats

4.1 DEFINITION OF VIDEO ANALYTICS

The old idea of a video surveillance system is of a security guard sitting in a booth watching the security camera feed live, hoping to catch suspicious activity. This model relies on having a live person watching and reviewing

all the video, however, which is not practical or efficient. Different security guards may have differing levels of focus or different ideas of suspicious activity. Video management software changes this system by using software to monitor the video feed around the clock, alerting person to activity so person only need to watch the cameras when something happens. This will help best utilize the surveillance system, saving time and effort.

Video analytics can be used for motion detection, facial recognition, license plate reading, people counting, dwell-time monitoring for retail stores, recognizing long lines at checkouts, and more. Video analytics can transform standard CCTV systems into intelligent and effective detection and alert systems. CCTV technology is now capable of recognizing faces of people, vehicles, animals, and bags automatically.

Integrated with CCTV video analytics, both facial and general recognition software systems are capable of counting, measuring speed, and monitoring direction. For example, recognition systems can monitor the duration of time that people are present in a specific area or how long a bag has been left unattended. Some video analytics solutions are even capable of understanding color. Intelligent video analytics software can recognize different behaviors and create an alarm on a user-defined rule, such as “person moving from vehicle to vehicle in a car park” or “large white vehicle parked in controlled zone.” Intelligent video can also improve the efficiency of control room operators, automate alarms for nonmonitored systems, enable PTZ cameras to zoom into objects creating alarms, and save time when searching CCTV recordings.

Video analytics is computerized video footage analysis that uses algorithms to differentiate between object types and identify certain behavior or action in real time, providing alerts and insights to users. Video Analytics is also referred to as video content analysis (VCA). It is the capability of automatically analyzing video to detect and determine temporal events not based on a single image. Many different functionalities can be implemented in VCA. Video motion detection is one of the simpler forms where motion is detected with regard to a fixed background scene. In other words video analytics is the practice of using computers to automatically identify things of interest without an operator having to view the video.

Applications of Video Analytics

Some applications of video analytics are:

- Banks—suspicious person detection
- Perimeter intrusion for critical infrastructures
- Stadiums—suspicious person detection, abandoned object detection
- Airports—suspicious person detection, abandoned object detection
- Railway/Metro Stations—Suspicious person detection, abandoned object detection
- Parking Management
- Vehicle Monitoring on roads

Metadata Search

A metadata search system enables fast and accurate searching for persons in huge surveillance video archives. The system includes automatic searching in response to real-time detection of suspicious individuals as well as offline searching defined by user input images or descriptions. There are offline metadata search and real-time meta analysis.

Offline Metadata Search

- Searches video archive for person through text or image
- Searches by face, clothes, age, gender, or combination
- Recognizes key words and natural sentence descriptions

Real-Time Meta-Analysis

- Extracts facial features, age, gender, presence info, and clothes in real time
- Activates an alert if behavior is suspicious or person is blacklisted
- Automatically searches video archive for footage of person following alert

Automatic Number Plate Recognition

Plate Analyzer is one of the world's leading automatic number plate recognition systems (ANPR), which uses high-speed and highly accurate GLVQ (generalized learning vector quantization) algorithm. It extracts

tilted/shifted/rotated license plate and deals with noisy or less contrast images. Once an image is captured, the image is enhanced, the license plate is detected and divided into sections, and characters on the license plate are detected and read using optical character recognition technology as shown in Figure 4.1.

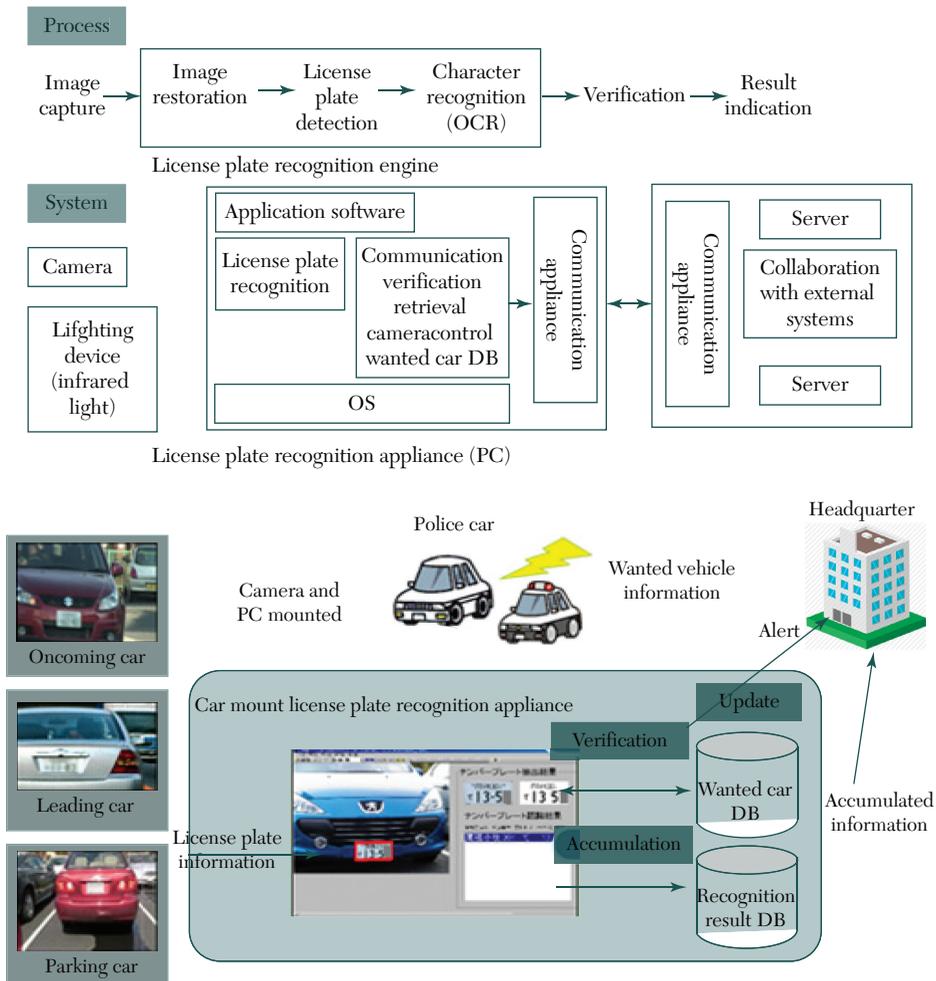


FIGURE 4.1. License plate recognition system.

License Plate Analyzer, combined with a car-mounted camera and processor, enables traveling law enforcement officials to swiftly recognize the license plates of parked and moving vehicles even when traveling at

high speeds. It can accurately recognize the license plates of oncoming vehicles even when the total combined speed of both vehicles is 100km/hr. Moreover, the license plates of parked cars can be recognized from all sides of the vehicle. Applications include car-park management, automated real-time alerts to unauthorized vehicles, toll booths, traffic and parking flow surveys, and site access control.

Features

- Car-mounted appliance recognizes in real time
- Verifying recognition results and wanted car DB in real time
- Transmitting alert information via wireless network
- Updating wanted car DB through wireless network
- Accumulating recognition result in DB
- Register recognition result to HQ server upon return

Image Analysis Software

Video monitoring systems basically collect and record video content recorded by surveillance cameras. When combined with image analysis technology, however, these systems can be transformed into tools that improve the efficiency and functionality of safety and security services. This technology is capable of efficiently extracting desired information from large amounts of video content collected by surveillance cameras.

Super resolution

Super-resolution technologies enable fine magnification of surveillance camera images for purposes such as face and license plate recognition. Conventional technology requires numerous extracted still images to improve the resolution of subjects in video content and enable clear subject magnification. When magnified by more than 2 or 3 times (4 to 9 times the number of pixels), however, these images become blurred. Therefore, there has been significant demand for technologies that could further improve resolution and enable greater image clarity at higher magnification. New technology creates a super-resolution image from a single extracted image (of a person's face, license plate, etc.) by utilizing a database (library) of categorized images. These images maintain fine details

even when magnified by more than 4 times (more than 16 times the number of pixels), making it possible to distinguish small and distant subjects much more easily than with conventional technologies as shown in Figure 4.2. Key features of these technologies are as follows:

1. *It creates images with fine details when highly magnified.* These technologies utilize images of subjects from a large library of images stored at various resolutions to create super-resolution images. The best images, at the most appropriate resolution, are automatically selected for use.
2. *It creates customized image libraries.* These technologies can efficiently extract small, optimized image libraries from huge libraries of images for specific purposes. Redundant images are eliminated to make the library as small and efficient as possible without compromising image quality.

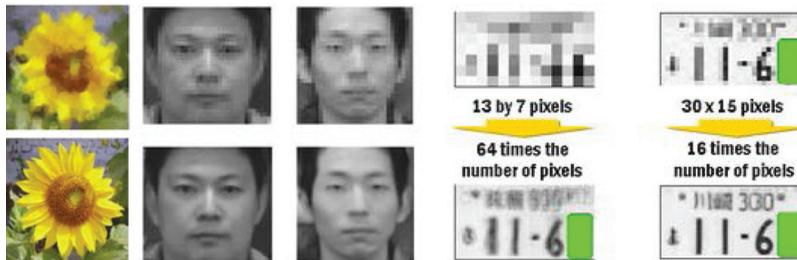


FIGURE 4.2. Top row with unprocessed images and bottom row with processed images (16 times the number of pixels)

Image restoration

The challenges of motion blur, defocusing blur, poor lighting, and severe angles are overcome by advanced image restoration technology that removes noise and blur while improving contrast. Blur and noise caused by motion are automatically detected and removed. Figure 4.3 shows original images and blur and noise removed images.

Contrast enhancement

Luminance information is analyzed and real-time contrast enhancement (image compensation processing) is performed to effectively extract target objects even from dark images of moving subjects that are otherwise difficult to confirm visually as shown in Figure 4.4.

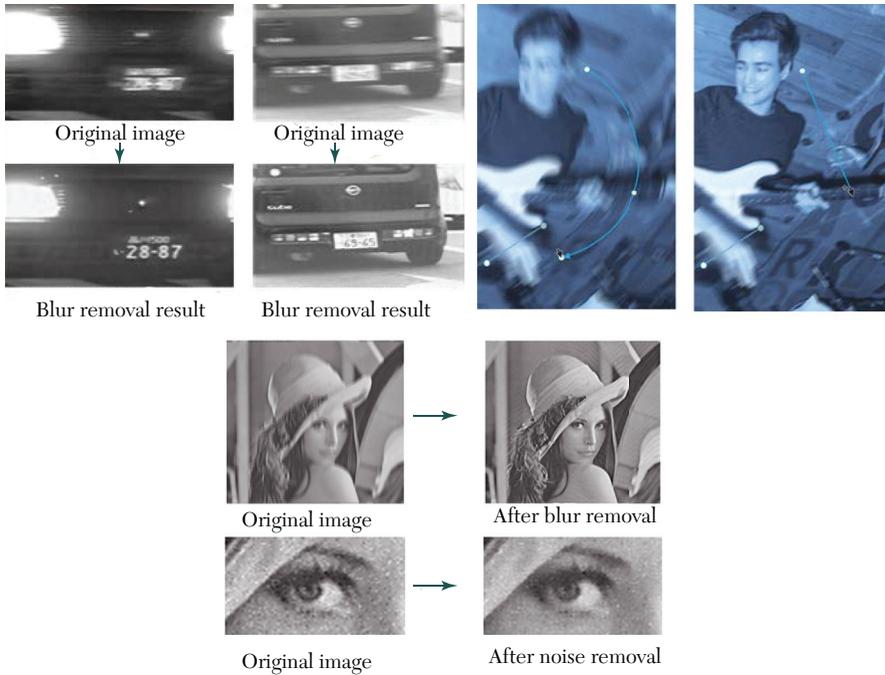


FIGURE 4.3. Original and image restored after removing blur and noise.

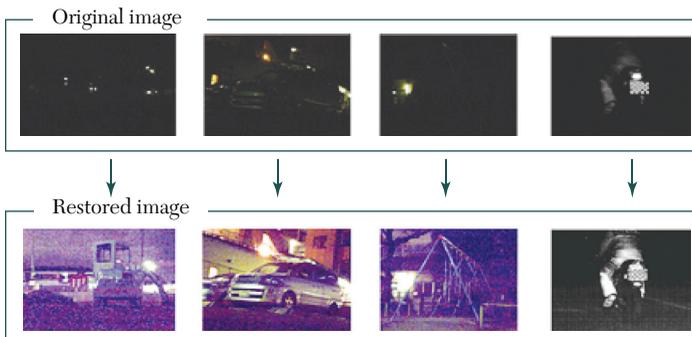


FIGURE 4.4. Contrast enhancement.

Moving object retrieval

This technology helps investigators search for specific persons and/or vehicles in archived video content recorded by surveillance cameras as listed here.

- Automatically extracts people, cars (standard, large, bus), and bikes from video
- Automatically classifies extracted cars as standard, large, or bus
- Acquires direction data from people, cars, and bikes appearing in video
- Automatically determines the colors of cars and clothes
- Acquires the time and date (timestamp) of extracted data
- Subject extraction via background sequential learning and action prediction Technology is used to extract moving object as shown in Figure 4.5.

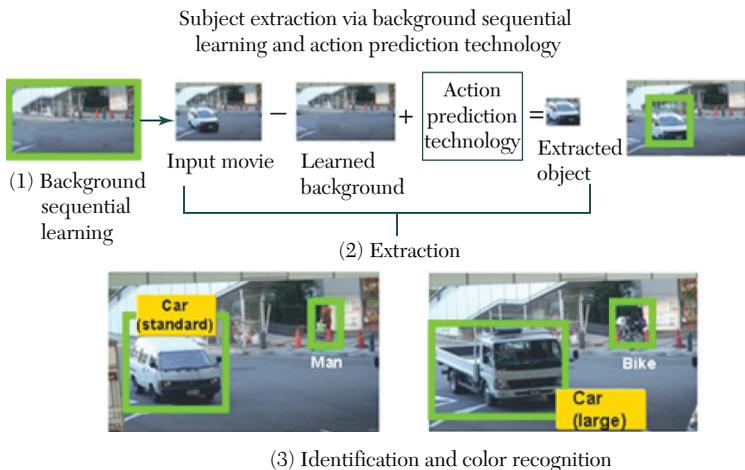


FIGURE 4.5. Moving object retrieval.

Security Center Integration

Video analytics is ideal for city-wide surveillance, traffic surveillance, building surveillance, and business intelligence applications. It is a server-based solution that offers a list of basic and advanced features that help security operators quickly and easily identify the following:

- **Intrusion Detection/Incidents:** Trespassing, line crossing (tripwire), auto-steering of PTZ (automatically zooms into or follows a violator), tailgating, and camera tampering
- **Suspicious Incidences (Objects):** Left-behind object detection and missing object detection

- Advanced Suspicious Incidents (People): Loitering, crowding, and tailgating
- Retail Applications (Counting/Flow): People counting, reporting, automatic emailing, face capturing, and queue management
- Traffic and Parking Management: Vehicle wrong-way detection, illegal parking detection, speeding detection, congestion detection, and vehicle counting
- Enhanced Monitoring: Video stitching and video smoke detection

Video Analytics for Perimeter Detection

Analytics can create a virtual fence that generates an alarm if a vehicle, person, or boat crosses a virtual tripwire. The system is capable of categorizing objects and can determine their direction. It detects breaches to secure zones, unauthorized activity and movement in specific areas, perimeter intrusion, loitering in sensitive areas, person or vehicle tailgating.

Video Analytics for People Counting

Retailers and visitor attractions require accurate visitor information as a key requirement for their business. The people counter accurately counts the number of people moving through shops, schools, banks, sports and transport facilities, prisons, museums, visitor attractions, airports, and car parks. This data is invaluable to organizations requiring key performance indicators such as footfall, customer to sales conversion ratios, marketing cost per thousand, and shoppers per square meter (SSM) and can be measured in real-time. Analysis of these metrics can provide critical profit and efficiency information to the retailer. The system uses standard CCTV cameras which can operate as stand-alone or be networked together to produce a map of visitor numbers across a shopping center for a whole retail chain. People counting CCTV cameras connect to a standard central PC over an existing IP network. The system accurately detects and records entries, exits, queues, and waiting times allowing the person watching to detect how long people spend in one place and how often that place was visited. This tests the effectiveness of new displays, compares one display with another, and evaluates new-store layouts, monitors queuing times, and tracks people around a store. To verify system accuracy, the person observing can see marker lines flash when the system detects someone crossing the counting zone.

Traffic Monitoring

To detect stopped vehicles and traffic obstacles, vehicle counting for traffic analysis, tailbacks, obstructions to traffic paths, vehicles traveling in the wrong direction, vehicles stopped in restricted zones, and vehicle speed monitoring are the example applications.

Auto Tracking Cameras for Facial Recognition

Although CCTV cameras have the ability to record clear details of an incident, without an operator, good facial recognition shots and clear vehicle registrations are often not recorded since the camera is facing in the wrong direction or giving too general a view. Auto-tracking cameras driven by video analytics software ensure that cameras are always recording activity of interest. The cameras are programmed to zoom into and follow user-defined behavior. Typically, behavior rules are defined relating to people, vehicles, and bags removed or deposited. Biometric facial recognition systems compare images of individuals from incoming CCTV video against specific databases and send alerts when a positive match occurs. The key steps in facial recognition are face detection, recording detected faces, and match recorded faces with those stored in a database and automatic process to find the closest match. Applications include:

- VIP Lists—make staff aware of important individuals (VIPs) and respond in an appropriate manner.
- Black Lists—identify known offenders or to register suspects to aid public safety.
- Banking Transactions—verify the person attempting a financial transaction.
- Access Control Verification—confirm identity visually, manually, or automatically.
- Mustering—Keep a tally of who is in and who is out.

Left Object Detection

Left object detection is used as an alarm if an object (bag) is located in a public area for a prolonged period without its owner being present. Applications include detection of unattended bags in airports or train stations. CCTV technology is now capable of recognizing colors, as well as recognizing faces of people, vehicles, animals, and bags automatically.

Next are some of the many reasons why analytics is a popular subject and is likely to remain a major industry focus.

Video analytics for everyone. Video analytics or CCTV software has come a long way over the past few years in terms of capabilities and accessibility. In previous years, analytics were primarily needed and available to large, corporate or government systems, requiring powerful servers to run each application along with high-end infrastructure. Now, due to maturing analytic engines and the exponential increase in camera and server processing power, analytics can be used by many different kinds of users and in a variety of environments. Analytics can run on the camera (edge) or on a server running multiple video streams or multiple applications.

Choice on the edge. The growing prevalence of analytics on the edge offers system flexibility and can significantly reduce the cost of the overall solution, as fewer servers are required to run the analytics. Edge-based analytics can also lessen system bandwidth demands, as video can be transmitted from the camera only after being prioritized by the analytics. Due to the onboard processing power of modern IP cameras, an edge-analytics-based approach can also offer device-specific selection of applications. Consider smartphone and the apps ecosystem in which it functions. With IP cameras, analytics present the same types of possibilities, adding in optional applications that build on the camera out of the box capabilities. Many premium cameras already possess the processing horsepower and available memory to do this today. It is simply a matter of time for an ecosystem of add-in functionality to become available.

Business intelligence. In the analog age, surveillance devices themselves were used purely as security solutions. Now that IP network cameras have become so popular and especially because of the edge analytics they offer, the camera has become a business intelligence tool. By deploying video analytics, end users can leverage specific analytics data and leverage it into actionable intelligence for functions such as marketing, health, and safety and personnel management. Analytics can offer such tools as heat mapping and queue monitoring for retail and hospitality, as well as people counting and foot traffic.

Processing power. The processing power of cameras and servers continues to grow exponentially while prices are steadily declining. As time passes and the camera's processing power increases, its capability for hosting an ever-increasing array of analytic functions also increases. A person can foresee

a time when the apps ecosystem makes available edge-based analytics that monitor air-conditioning applications, lighting apps, and access control apps, working together to make the “Smart Home” and “Smart Building” ideas a reality.

Better decision making. Even with the great strides in improved effectiveness, increased accessibility and cost, the primary function of analytics is still to complement the role of system operators, not to eliminate them. Ultimately, analytics assist operators in making informed decisions by illuminating the unusual from the mundane, as the number of cameras being deployed and monitored continues to increase. Better analytics can’t help but give the operator more reliable information, which in turn improves response time and effectiveness.

4.2 VIDEO ANALYTICS ALGORITHMS

Algorithms are the essence of embedded vision. Through algorithms, visual input in the form of raw video or images is transformed into meaningful information that can be acted upon. The algorithms that are applicable depend on the nature of the vision processing being performed. Vision applications are generally constructed from a pipelined sequence of algorithms, as shown in Figure 4.6. Typically, the initial stages are concerned with improving the quality of the image. For example, this may include correcting geometric distortion created by imperfect lenses, enhancing contrast, and stabilizing images to compensate for undesired movement of the camera.

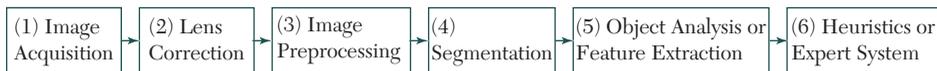


FIGURE 4.6. A typical embedded vision algorithm pipeline.

The second set of stages in a typical embedded vision algorithm pipeline are concerned with converting raw images (i.e., collections of pixels) into information about objects. A wide variety of techniques can be used, identifying objects based on edges, motion, color, size, or other attributes. The final set of stages in a typical embedded vision algorithm pipeline are concerned with making inferences about objects. For example, in an automotive safety application, these algorithms would attempt to distinguish between vehicles, pedestrians, road signs, and other features of

the scene. Generally speaking, vision algorithms are very computationally demanding, since they involve applying complex computations to large amounts of video or image data in real time. There is typically a trade-off between the robustness of the algorithm and the amount of computation required.

Algorithm Example: Lens Distortion Correction

Lenses, especially inexpensive ones, tend to introduce geometric distortion into images. This distortion is typically characterized as “barrel” distortion or “pincushion” distortion, as illustrated in Figure 4.7.

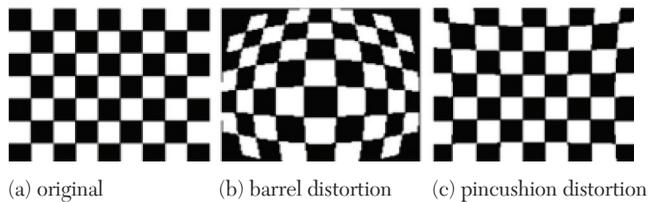


FIGURE 4.7. Examples of different types of lens distortion.

As shown in Figure 4.7, this kind of distortion causes lines that are in fact straight to appear curved, and vice-versa. This can prevent vision algorithms. Hence, it is common to apply an algorithm to reverse this distortion. The usual technique is to use a known test pattern to characterize the distortion. From this characterization data, a set of image warping coefficients is generated, which is subsequently used to “undistort” each frame. In other words, the warping coefficients are computed once, and then applied to each frame. This is illustrated in Figure 4.8.

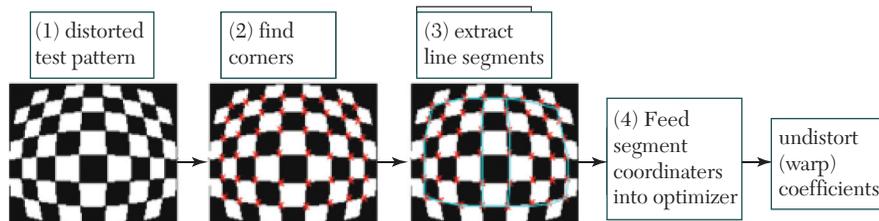


FIGURE 4.8. Camera calibration procedure.

One complication that arises with lens distortion correction is that the warping operation will use input data corresponding to pixel locations that do not precisely align with the actual pixel locations in the input frame. To

enable this to work, interpolation is used between pixels in the input frame. The more demanding the algorithm, the more precise the interpolation must be and the more computationally demanding. For color imaging, the interpolation and warping operations must be performed separately on each color component. For example, a 720p video frame comprises 921,600 pixels, or approximately 2.8 million color components. At 60 frames per second, this corresponds to about 166 million color components per second. If the interpolation and warping operations require 10 processing operations per pixel, the distortion correction algorithm will consume 1.66 billion operations per second.

Dense Optical Flow Algorithm

“Optical flow” is a family of techniques used to estimate the pattern of apparent motion of objects, surfaces, and edges in a video sequence. In vision applications, optical flow is often used to estimate observer and object positions and motion in 3-D space, or to estimate image registration for super resolution and noise reduction algorithms. Optical flow algorithms typically generate a motion vector for each pixel a video frame. Optical flow requires making some assumptions about the video content (this is known as the “aperture problem”). In Figure 4.9, prototype implementation of a stationary-camera pedestrian detection system implemented using a combination of a CPU and an FPGA.

In Figure 4.9, the preprocessing block comprises operations such as scaling and noise reduction, intended to improve the quality of the image. The image analysis block incorporates motion detection, pixel statistics such as averages, color information, edge information, and so on. At this stage of processing, the image is divided into small blocks. The object segmentation step groups blocks having similar statistics and thus creates an object. The statistics used for this purpose are based on user defined features specified in the hardware configuration file.

The identification and metadata generation block generates analysis results from the identified objects such as location, size, color information, and statistical information. It puts the analysis results into a structured data format and transmits them to the CPU. Finally, the on-screen display (OSD) block receives command information from the host and superimposes graphics on the video image for display. Most computer vision algorithms were developed on general-purpose computer systems with software written in a high-level language. Some of the pixel-processing operations have changed very little in the decades since they were first implemented

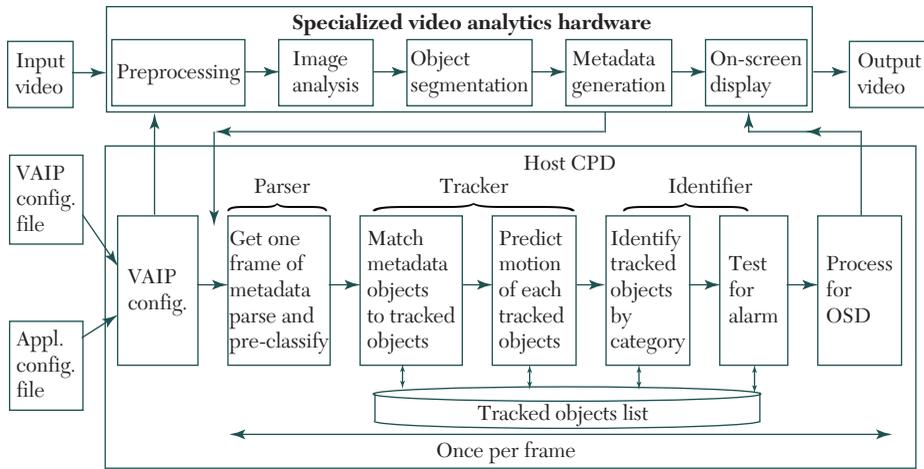


FIGURE 4.9. Block diagram of proof-of-concept pedestrian detection application using an FPGA and a CPU.

on mainframes. With today's broader embedded vision implementations, existing high-level algorithms may not fit within the system constraints, requiring new innovation to achieve the desired results. Some of this innovation may involve replacing a general-purpose algorithm with a hardware-optimized equivalent. With such a broad range of processors for embedded vision, algorithm analysis will likely focus on ways to maximize pixel-level processing within system constraints.

Camera Performance Affecting Video Analytics

Camera designers have decades of experience in creating image-processing pipelines that produce attractive and/or visually accurate images, but image-processing concepts that produce video that is well-purposed for subsequent video analytics. It seems reasonable to begin by considering a conventional ISP (image signal processor). After all, the human eye-brain system produces what we consider aesthetically pleasing imagery for a purpose: to maximize our decision-making abilities. But which elements of such an ISP are most important to get right for good video analytics, and how do they impact the performance of the algorithms which run on them? Figure 4.10 shows a simplified block schematic of a conventional ISP.

The input is sensor data in a raw format (one color per pixel), and the output is interpolated RGB or YCbCr data (three colors per pixel). Table 4.1 briefly summarizes the function of each block. Hence an ISP design team will frequently also implement these modules.

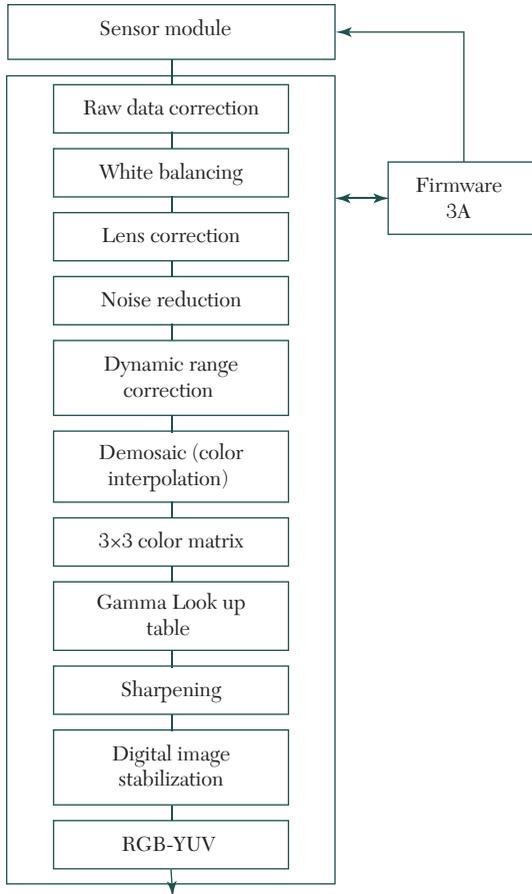


FIGURE 4.10. This block diagram provides a simplified view inside a conventional ISP (image signal processor).

TABLE 4.1. Functions of Main ISP NModules

Module	Function
Raw data correction	Set black point, remove defective pixels.
Lens correction	Correct for geometric and luminance/color distortions.
Noise reduction	Apply temporal and/or spatial averaging to increase SNR (signal to noise ratio).
Dynamic range compression	Reduce dynamic range from sensor to standard output without loss of information.
Demosaic	Reconstruct three colors per pixel via interpolation with pixel neighbors.

Module	Function
3A	Calculate correct exposure, white balance, and focal position.
Color correction	Obtain correct colors in different lighting conditions.
Gamma	Encode video for standard output.
Sharpen	Edge enhancement.
Digital image stabilization	Remove global motion due to camera shake/vibration.
Color space conversion	RGB to YCbCr.

Video analytics algorithms may operate directly on the raw data, on the output data, or on data which has subsequently passed through a video compression codec. The data at these three stages often has very different characteristics and quality, which are relevant to the performance of video analytics algorithms.

Let us now review the stages of the ISP shown in Figure 4.10 and Table 4.1. The better the sensor and optics, the better the quality of data on which to base decisions. But “better” is not a matter simply of resolution, frame rate or SNR (signal-to-noise ratio). Dynamic range is also a key characteristic. Dynamic range is essentially the relative difference in brightness between the brightest and darkest details that the sensor can record within a single scene, normally expressed in dB.

Common CMOS and CCD sensors have a dynamic range of between 60 and 70 dB, which is sufficient to capture all details in scenes which are fairly uniformly illuminated. However, special sensors are required to capture the full range of illumination in high-contrast environments. Around 90dB of dynamic range is needed to simultaneously record information in deep shadows and bright highlights on a sunny day; this requirement rises further if extreme lighting conditions occur (the human eye has a dynamic range of around 120dB). If the sensor can't capture such a range, objects which move across the scene will disappear into blown-out highlights, or into deep shadows below the sensor black level. High (e.g., wide) dynamic range sensors are certainly helpful in improving video analytics in uncontrolled lighting environments.

The next most important element is noise reduction, which is important for a number of reasons. In low light, noise reduction is frequently necessary to raise objects above the noise background, subsequently aiding in accurate segmentation. Also, high levels of temporal noise can easily confuse tracking algorithms based on pixel motion even though such noise is largely

uncorrelated, both spatially and temporally. If the video goes through a lossy compression algorithm prior to video analytics postprocessing, one should also consider the effect of noise reduction on compression efficiency. The bandwidth required to compress noisy sources is much higher than with “clean” sources. If transmission or storage is bandwidth-limited, the presence of noise reduces the overall compression quality and may lead to increased amplitude of quantization blocks, which easily confuses video analytics algorithms.

Effective noise reduction can readily increase compression efficiency by 70% or more in moderate noise environments, even when the increase in SNR is visually not very noticeable. However, noise reduction algorithms may themselves introduce artifacts. Temporal processing works well because it increases the SNR by averaging the processing over multiple frames. Both global and local motion compensation may be necessary to eliminate false motion trails in environments with fast movement. Spatial noise reduction aims to blur noise while retaining texture and edges and risks suppressing important details. Hence one must therefore strike a careful balance between SNR increase and image quality degradation.

The correction of lens geometric distortions, chromatic aberrations, and lens shading (vignetting) is of inconsistent significance, depending on the optics and application. For conventional cameras, uncorrected data may be perfectly suitable for postprocessing. In digital PTZ (point, tilt and zoom) cameras, however, correction is a fundamental component of the system. A set of “3A” algorithms control camera exposure, color, and focus, based on statistical analysis of the sensor data. Their function and impact on analytics is shown in Table 4.2.

Finally, DRC (dynamic range compression) is a method of nonlinear image adjustment which reduces dynamic range, that is, global contrast. It has two primary functions: detail preservation and luminance normalization. For some embedded vision applications, it may be no problem to work directly with the high-bit-depth raw sensor data. But if the video analytics is run in-camera on RGB or YCbCr data, or as postprocessing based on already lossy-compressed data, the dynamic range of such data is typically limited by the 8-bit standard format, which corresponds to 60 dB. This means that, unless dynamic range compression occurs in some way prior to encoding, the additional scene information will be lost. While techniques for DRC are well established (gamma correction is one form, for example), many of these techniques decrease image quality in the process, by degrading local contrast and color information, or by introducing spatial artifacts.

TABLE 4. 2. The Impact of “3A” Algorithms

Algorithm	Function	Impact
Auto exposure	Adjust exposure to maximize the amount of scene captured. Avoid flicker in artificial lighting.	A poor algorithm may blow out highlights or clip dark areas, losing information. Temporal instabilities may confuse motion-based analytics.
Auto white balance	Obtain correct colors in all lighting conditions.	If color information is used by video analytics, it needs to be accurate. It is challenging to achieve accurate colors in all lighting conditions.
Auto focus	Focus the camera.	Which regions of the image should receive focus attention? How should the algorithm balance temporal stability versus rapid refocusing in a scene change?

Another application of DRC is in image normalization. Advanced video analytics algorithms, such as those employed in facial recognition, are susceptible to changing and nonuniform lighting environments. For example, an algorithm may recognize the same face differently depending on whether the face is uniformly illuminated or lit by a point source to one side, in the latter case casting a shadow on the other side. Good DRC processing can be effective in normalizing imagery from highly variable illumination conditions to simulated constant, uniform lighting.

In general, the requirements of an ISP is to produce natural, visually-accurate imagery and to produce well-purposed imagery for video analytics are closely matched. However, as is well known, it is challenging to maintain accuracy in uncontrolled environments.

Video Imaging Techniques

There are three techniques for imaging. They are:

1. Normal Imaging—capturing with a digital camera. Images stored in the form matrices of numbers.
2. IR Imaging—Infrared imaging is used extensively for both military and civilian purposes. It involves tracking of the heat signature.

3. Electric Field Imaging—imaging by tracking changes in the electrostatic field generated by the system. Used in nature by small electric fish for navigation. Electric-field imaging starts with the electric field are generated by a voltage potential between two conductors. It works by measuring changes in electric field in proximity of an object. The Motorola MC33794, together with a microcontroller, simplifies electric-field imaging. The chip supports up to nine electrodes and has built-in watchdog and power-on-reset timers as shown in Figure 4.11.



FIGURE 4.11. Electric static-field imaging by MC33794.

4.3 MACHINE LEARNING IN EMBEDDED VISION APPLICATIONS

One of the hottest topics within the embedded vision space at the current time is machine learning. Machine learning spans several industry megatrends, playing a very prominent role within not only embedded vision (EV), but also Industrial Internet of Things (IoT) and cloud computing. For those unfamiliar with machine learning it is most often implemented by the creation and training of a neural network. The term neural network is very generic and includes a significant number of distinct subcategories whose names are normally used to identify the exact type of network being implemented. These neural networks are modeled upon the cerebral cortex in that each neuron receives an input, processes it and communicates it on to another neuron. Neural Networks therefore typically consist of an input layer, several hidden internal layers, and an output layer as shown in Figure 4.12.

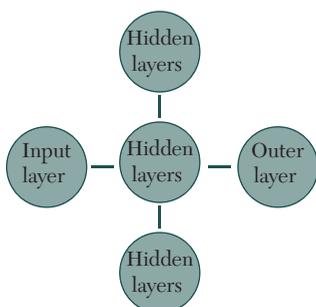


FIGURE 4.12. Machine learning layer for neural networks (NN).

At the simplest level the neuron takes its input and applies a weight to it before performing a transfer function upon the sum of the weighted inputs. This result is then passed onto either another layer within the hidden layers or to the output layer. Neural networks which pass the output of one stage to another without forming a cycle are called feed-forward neural networks (FNN), while those which contain directed cycles where there is feedback, and are called recurrent neural networks (RNN).

One very commonly used term in many machine-learning applications is deep neural networks (DNN). These are neural networks that have several hidden layers enabling a more complex machine-learning task to be implemented. Neural networks are required to be trained to determine the value of the weights and biases used within each layer. During training, the network has a number of both correct and incorrect inputs applied with the error function being used to teach the network the desired performance. Training a DNN may require a very significant data set to correctly train the required performance. One of the most important uses of machine learning is within the embedded vision sphere where systems are evolving into vision-guided autonomous systems from vision enabled systems. What separates embedded vision applications from other simpler machine-learning applications is that they have a two-dimensional input format. As such, in machine-learning implementations, a network structure called convolutional neural networks (CNN) are used as they have the ability to process two-dimensional inputs.

CNN are a type of feed-forward network that contains several convolutional and subsampling layers along with a separate fully connected network to perform the final classification. Due to the complexity of CNNs, they also fall within the deep learning classification. Within the convolution layer the input image will be broken down into a number of overlapping smaller tiles. The results from this convolution is used to create an activation map by the use of an activation layer, before being subject to further subsampling and additional stages, prior to being applied to the final fully connected network. The exact definition of the CNN network will vary depending upon the network architecture implemented, however it will typically contain at least the following elements:

- Convolution—used to identify features within the image
- Rectified Linear Unit (ReLU)—activation layer used to create an activation map following the convolution
- Max Pooling—performs subsampling between layers
- Fully Connected—performs the final classification

The weights for each of these elements is determined via training, and one of the advantages of the CNN is the relative ease of training the network. Training to generate the weights requires large image sets of both the object which is wished to be detected, and the false images.

This enables us to create the weights required for the CNN. Due to the processing requirements involved in the training process, it is often run on cloud-based processors that offer high-performance computing.

Machine learning is a complex subject, especially if one had to start from the beginning each time and define the network, its architecture, and generate the training algorithms. To help engineers both implement the networks and train the network, there are a number of industry standard frameworks such as Caffe and Tensor Flow. The Caffe framework provides machine-learning developers with a range of libraries, models, and pretrained weights within a C++ library, along with Python and MATLAB bindings. This framework enables the user to create networks and train them to perform the operations desired without the need to start from scratch. To aid reuse, Caffe users can share their models via the model zoo, which provides several network models that can be implemented and updated for a specialized task if desired. These networks and weights are defined within a prototxt file. When deployed in the machine-learning environment, this file is used to define the inference engine.

Types of Machine-Learning Algorithms

There are many algorithms for selecting the best weights for features. These algorithms are based on different methods for adjusting the feature weights and assumptions about the data. Some of the common techniques specifically, those involving neural networks, k -nearest neighbors, support vector machines, decision trees, the naive Bayes algorithm, and deep learning are described in the following sections.

Neural networks

Learning with *neural networks* is the archetypal machine-learning method. The following three functions are parts of the learning schema for this method (Figure 4.13): (a) the error function measures how good or bad an output is for a given set of inputs, (b) the search function defines the direction and magnitude of change required to reduce the error function, and (c) the update function defines how the weights of the network are updated on the basis of the search function values.

The example provided in Figure 4.13 would be a neural network with several input nodes (referred to as x_1 to x_n), two hidden layers, and an output layer with several output nodes. The output nodes are summed and compared with the desired output by the error (loss) function, which then

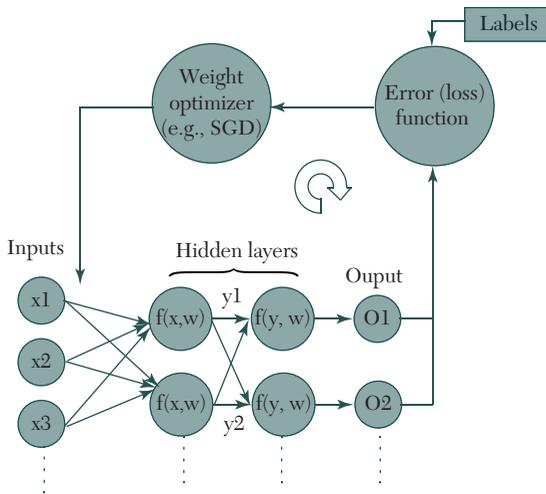


FIGURE 4.13. Example of a neural network. In this case, the input values (x_1 , x_2 , x_3) are multiplied by a weight (w) and passed to the next layer of nodes. Although we show just a single weight, each such connection weight has a different numeric value, and it is these values that are updated as part of the learning process. Each node has an activation function (f) that computes its output (y) by using x and w as inputs. The last layer is the output layer. Those outputs are compared with the expected values (the training sample labels), and an error is calculated. The weight optimizer determines how to adjust the various weights in the network in order to achieve a lower error in the next iteration. Stochastic Gradient Descent (*SGD*) is one common way of updating the weights of the network. The network is considered to have completed learning when there is no substantial improvement in the error over prior iterations.

uses the weight optimizer to update the weights in the neural network. During the training phase, examples are presented to the neural network system, the error for each example is computed, and the total error is computed. On the basis of the error, the search function determines the overall direction to change, and the update function then uses this change metric to adjust the weights. This is an iterative process, and one typically continues to adjust the weights until there is little improvement in the error. Real world examples typically have one or more hidden layers and more complex functions at each node.

***k* nearest neighbors**

With *k nearest neighbors*, one classifies an input vector that is, a collection of features for one unknown example object by assigning the object to the most similar class or classes (Figure 4.14). The number of neighbors, or known objects that are closest to the example object, which “vote” on the classes that the example object may belong to is k . If k is equal to 1, then the unknown object is simply assigned to the class of that single

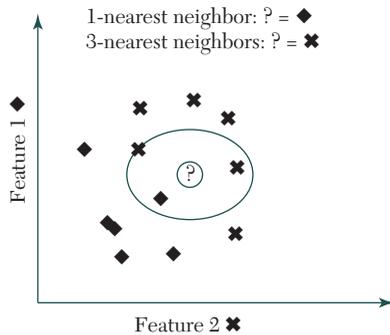


FIGURE 4.14. Example of the k -nearest neighbors algorithm. The unknown object (?) would be assigned to the ◆ class on the basis of the nearest neighbor ($k = 1$), but it would be assigned to the ✕ class if k were equal to 3, because two of the three closest neighbors are ✕ class objects. Values plotted on the x and y axes are those for the two element feature vector describing the example objects.

nearest neighbor. The similarity function, which determines how close one example object is to another, can be the Euclidean distance between the values of the input vector versus the values of the vector for the other examples. However, it is critical that the normalization of the values in the feature vectors be performed correctly.

Support vector machines

Support vector machines are so named because they transform input data in a way that produces the widest plane, or support vector, of separation between the two classes. Support vector machines allow flexible selection of the degree to which one wishes to have a wide plane of separation versus the number of points that are wrong owing to the wide plane. These learning machines were invented some time ago, and the reason for their recent greater popularity is the addition of basic functions that can map points to other dimensions by using nonlinear relationships and thus classify examples that are not linearly separable. This capability gives support vector machine algorithms a big advantage over many other machine-learning methods. A simple example of how a nonlinear function can be used to map data from an original space (the way the feature was collected e.g., the CT attenuation) to a hyperspace (the new way the feature is represented e.g., the cosine of the CT attenuation) where a hyper-plane (a plane that exists in that hyperspace, the idea being to have the plane positioned to optimally separate the classes) can separate the classes is illustrated in Figure 4.15.

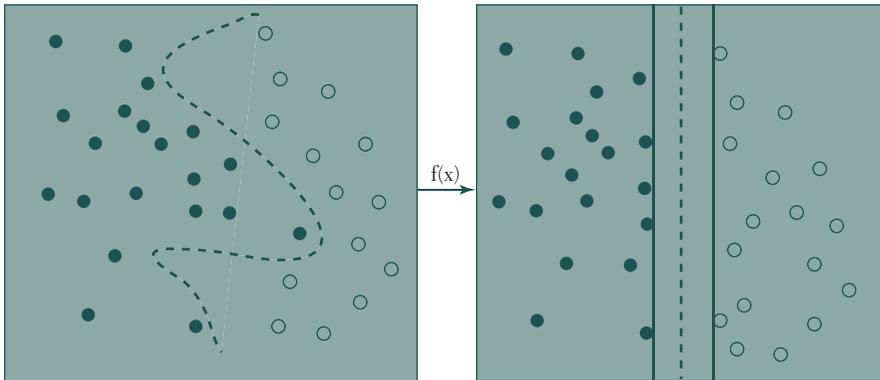


FIGURE 4.15. Example shows two classes (●,○) that cannot be separated by using a linear function (left diagram). However, by applying a nonlinear function $f(x)$, one can map the classes to a space where a plane can separate them (right diagram). This example is two dimensional, but support vector machines can have any dimensionality required. These machines generally are “well behaved,” meaning that for new examples that are similar, the classifier usually yields reasonable results. When the machine-learning algorithm is successful, the two classes will be perfectly separated by the plane. In the real world, perfect separation is not possible, but the optimal plane that minimizes misclassifications can be found.

Decision trees

All of the machine-learning methods described up to this point have one important disadvantage: the values used in the weights and the activation functions usually cannot be extracted to gain some form of information that can be interpreted by humans. *Decision trees* offer the substantial advantage that they produce human-readable rules regarding how to classify a given example. Decision trees are familiar to most people and typically take the form of yes or no questions for example, whether a numeric value is higher than a certain value.

The aspect of decision trees that applies to machine learning is the rapid search for the many possible combinations of decision points to find the points that, when used, will result in the simplest tree with the most accurate results. When the algorithm is run, one sets the maximal depth (i.e., maximal number of decision points) and the maximal breadth that is to be searched and establishes how important it is to have correct results versus more decision points.

In some cases, one can improve accuracy by using an ensemble method whereby more than one decision tree is constructed. Two commonly used ensemble methods are bagging and random forest techniques. By boosting with aggregation, or bagging, one builds multiple decision trees by repeatedly resampling the training data by means of replacement, and voting on the

trees to reach a consensus prediction. Although a random forest classifier uses a number of decision trees to improve the classification rate and is often high performing, it does not resample the data. In addition, with random forests, only a subset of the total number of features is randomly selected and the best split feature from the subset is used to split each node in a tree unlike with bagging, whereby all features are considered for splitting a node.

Naive Bayes algorithm

According to the Bayes theorem, one of the oldest machine-learning methods, the probability of an event is a function of related events. The Bayes theorem formula is $P(y|x) = [P(y) \times P(x|y)]/P(x)$: the probability (P) of y given x equals the probability of y times the probability of x given y , divided by the probability of x . In machine learning, where there are multiple input features, one must chain the probabilities of each feature together to compute the final probability of a class, given the array of input features that is provided. The *naive Bayes algorithm* is different from most machine-learning algorithms in that one calculation is used to define the relationship between an input feature set and the output. As such, this method does not involve the same iterative training process that most other machine-learning methods involve. It does require training and testing data, so the issues related to training and testing data still apply.

This algorithm is referred to as the naive Bayes algorithm rather than simply the Bayes algorithm to emphasize the point that all features are assumed to be independent of each other. Because this is usually not the case in real life, using this approach can lead to misleading results. However, this method can be used to acquire useful estimates of performance, even when this assumption is violated. In addition, the use of this approach often leads to more robust results when there are fewer examples and when the examples do not include all possibilities.

These considerations also raise the important issue of pretest probabilities and accuracy: if the prevalence of a positive finding were 1%, then one could simply designate all cases as those of negative findings and achieve 99% accuracy. In many cases, 99% accuracy would be good, and this algorithm would also have 100% specificity; however, it would have 0% sensitivity. From this perspective, it is important to recognize that accuracy alone is not sufficient and prior probability is an important piece of information that will affect performance measures.

Deep learning

Deep learning, also known as deep neural-network learning, is a new and popular area of research that is yielding impressive results and growing fast. Early neural networks were typically only a few (<5) layers deep, largely because the computing power was not sufficient for more layers and owing to challenges in updating the weights properly. Deep learning refers to the use of neural networks with many layers typically more than 20. This has been enabled by tools that leverage the massively parallel computing power of graphics processing units that were created for computer gaming, such as those built by NVIDIA Corporation (Santa Clara, CA). Several types of deep-learning networks have been devised for various purposes, such as automatic object detection and segmentation on images, automatic speech recognition, and genotypic and phenotypic detection and classification of diseases in bioinformatics. Some deep learning algorithm tools are deep neural networks, stacked auto encoders, deep Boltzmann machines, and convolutional neural networks (CNNs).

CNNs are similar to regular neural networks. The difference is that CNNs assume that the inputs have a geometric relationship like the rows and columns of images. The input layer of a CNN has neurons arranged to produce a convolution of a small image (i.e., kernel) with the image. This kernel is then moved across the image, and its output at each location as it moves across the input image creates an output value. Although CNNs are so named because of the convolution kernels, there are other important layer types that they share with other deep neural networks. Kernels that detect important features (e.g., edges and arcs) will have large outputs that contribute to the final object to be detected. In deep networks, specialized layers are now used to help amplify the important features of convolutional layers. The layer typically found after a convolution layer is an activation layer. In the past, activation functions were designed to simulate the sigmoidal activation function of a neuron, but current activation layers often have a much simpler function. A common example is the rectified linear unit, or ReLU, which has an output of 0 for any negative value and an output equal to the input value for any positive value.

The pooling layer is another type of layer that is important to CNNs. A pooling layer will take the output of something like a convolution kernel and find the maximal value; this is the so-called max-pool function. By taking the maximal value of the convolution, the pooling layer is rewarding the convolution function that best extracts the important features of an image.

An important step in training deep networks is regularization, and one popular form of regularization is dropout. Regularization refers to rescaling the weights connecting a pair of layers to a more effective range. Somewhat counterintuitively, randomly setting the weights between nodes of layers to 0 has been shown to substantially improve performance because it reduces overfitting. Dropout regularization is typically implemented by having weights (often 50% or more between two layers) set to 0. The specific connections that are set to 0 at a given layer are random and vary with each round of learning. One can imagine that if random connection weights are set to 0 and a group of examples is tested, then those weights that are really important will affect performance, but those weights that are not so important and perhaps reflective of a few specific examples will have a much smaller influence on performance. With enough iterations, only the really important connections will be kept. There are many possible combinations of layers and layer sizes. At present, there is no formula to define the correct number and type of layer for a given problem. Selecting the best architecture for a given problem is still a trial and error process. It is interesting that some different neural-network architectures have been successful in machine-learning competitions such as the ImageNet Challenge. Some of these architectures are LeNet, GoogleNet, AlexNet, VGGNet, and ResNet.

An important benefit of CNN deep learning algorithms, as compared with traditional machine-learning methods, is that there is no need to compute features as a first step. The CNN effectively finds the important features as a part of its search process. As a result, the bias of testing only those features that a human believes to be important is eliminated. The task of computing many features and then selecting those that seem to be the most important also is eliminated.

Open source tools

A wide variety of *open source tools* for developing and implementing machine learning are available. These tools are compatible with the majority of modern programming languages, including Python, C++, Octave MATLAB, R, and Lua. Furthermore, tools such as Apache Storm, Spark, and H2O libraries have been developed for machine learning tasks and large datasets. Most deep-learning tool kits can now leverage graphics processing unit power to accelerate the computations of a deep network. Python libraries tend to be the most popular and can be used to implement the most recently available algorithms; however, there are many ways to

access the algorithms implemented in one language from another language. In fact, many Python libraries are implemented in C++. Furthermore, some libraries are built on other libraries for example, the Keras library runs on top of either Theano or TensorFlow.

Implementing Embedded Vision and Machine Learning

Programmable logic-based solutions such as the heterogeneous Xilinx All Programmable Zynq, 7000 SoC (System on Chip) and Multi-Processor System on Chip, (MPSoC) like Zynq UltraScale MPSoC shown in Figure 4.16, are increasingly used in embedded vision applications.

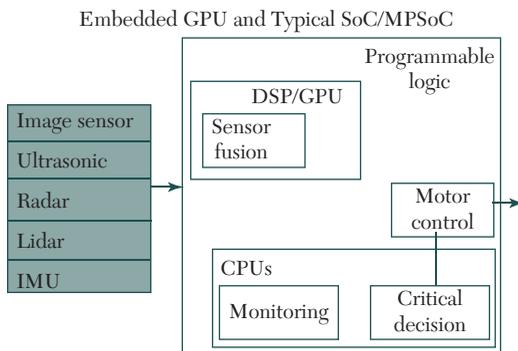


FIGURE 4.16. Embedded GPU, SoC/ MPSoC.

These devices combine programmable logic (PL) fabric with a high-performance ARM core in the processing system (PS). This combination allows the creation of a system that has an increased response time, is very flexible to future modification, and offers a power efficient solution. A low-latency decision and response loop is of critical importance for many applications, such as vision-guided autonomous robots, where the response time is critical to avoid injury or damage to people and its environment. This increased response time is enabled by the use of programmable logic to implement the vision processing pipeline, and machine-learning inference engine to implement the machine learning.

Implementing both the image-processing algorithm and the machine-learning network within a heterogeneous SoC can be achieved with ease using the reVISION stack from Xilinx as shown in Figure 4.17. It provides support for both traditional image-processing applications and machine learning applications based around the SDSoC tool. Within reVISION support is provided for both the OpenVX and Caffe Frameworks. To support

OpenVX	Caffe	Application development
OpenCV	SDSoC	Algorithm development
		Platform development

FIGURE 4.17. reVISION stack.

the OpenVX framework, the core image-processing functions can be accelerated into the programmable logic to create the image-processing pipeline. While the machine-learning inference environment provides support for hardware-optimized libraries in the programmable logic to implement the inference engine which performs the machine-learning implementation.

The revision stack provides integration with Caffe as such implementing machine-learning inference engines is as easy as providing a prototxt file and the trained weights, and the framework handles the rest. This prototxt file is then used to configure the C/C++ scheduler running on the processing system to accelerate the neural-network inference on the hardware optimized libraries within the programmable logic. The programmable logic is used to implement the inference engine and contains such functions as convolution, reLu, and pooling.

The number representation systems used within machine-learning inference engine implementations also play a significant role in its performance. Machine-learning applications are increasingly using more efficient reduced precision fixed-point number systems, such as INT8 representation. The use of fixed-point reduced precision number systems comes without a significant loss in accuracy when compared to a traditional floating point 32 (FP32) approach. As fixed-point mathematics are also considerably easier to implement than floating point this move to INT8 provides for more efficient faster solutions in some implementations.

This use of fixed-point number systems is ideal for implementation within a programmable logic solution, reVISION provides the ability to work with INT8 representations in the PL. These INT8 representations enable the use of dedicated DSP blocks within the PL. The architecture of these DSP blocks enables a maximum of two concurrent INT8 multiply accumulate operations to be performed when using the same kernel weights. This provides not only a high-performance implementation but also one which provides a reduced power dissipation. The flexible nature of programmable logic also enables easy implementation of further reduced precision fixed-point number representation systems as they are adopted.

Within the real world, the reVISION stack can provide a significant benefit. One example of an application which deploys machine learning within an embedded vision application would be a vehicle collision avoidance system. Targeting a Xilinx UltraScale+ MPSoC and developing the application within reVISION using SDSoC to accelerate functions to the programmable logic as required to optimize performance, provides a significant increase in responsiveness.

The reVISION design can identify a potential collision event and engage the vehicle brakes within 2.7ms (with batch size of 1) while the GPU-based approach takes between 49ms and 320ms (with large batch size) depending upon its implementation. The large batch size is needed for GPU architecture to get to reasonable throughput with significant sacrifice in response time while Zynq can achieve high performance even at batch size of 1 with lowest latency. This difference in reaction time can be the difference between avoiding a collision or not.

Machine learning will continue to be a significant driving factor in many applications, especially vision guided robotics or “cobots” as they are increasingly called. Heterogeneous SoC which combine processor cores with programmable logic enable the creation of very efficient, responsive and reconfigurable solutions. The provision of stacks like reVISION, open up for the first time the benefits of programmable logic to a wider developer community and reduce the development time of the solution.

Embedded Computers Make Inroads to Vision Applications

Embedded vision systems are replacing industrial PCs in machine vision and image-processing applications. Whether running general purpose operating systems (GPOS) such as Windows or real-time operating systems (RTOS) such as VxWorks, PCs form the heart of many of today’s machine vision systems. However, unlike PCs developed for the consumer market, the PCs used in industrial applications must be capable of withstanding shock and vibration in harsh environments, incorporate industrial networking standards, high-performance CPUs, and commonly used camera-to-computer interfaces.

In the past, it was the task of the systems integrator to configure these PCs to meet the needs of their particular application using off-the-shelf motherboards, I/O controllers and frame grabbers. Now, however, a number of companies offer embedded PC systems specifically tailored for industrial and machine vision applications. By incorporating standard

camera and industrial interfaces and I/O, these PCs alleviate the need for systems developers to configure such systems, allowing them to concentrate more on the vision problem needed to be solved.

Because of the increased use of machine vision in industrial applications, a number of traditional industrial controller manufacturers are now targeting the embedded vision market. While such controllers have traditionally used interfaces such as Gigabit Ethernet for communications, the increased use of such standards allows them to be targeted for machine vision applications. Indeed, as such embedded controllers emerge, the distinguishing features between them and traditional systems offered by established machine vision companies is slowly blurring.

4.4 EXAMPLES FOR MACHINE LEARNING

1. Convolutional Neural Networks for Autonomous Cars

Convolutional neural networks (CNNs or ConvNets) used in machine learning are similar to neural networks. CNNs have revolutionized the pattern recognition computation process. Prior to the widespread adoption of CNNs, most pattern-recognition tasks involved hand-crafted features extraction followed by classifications. With CNNs, features are now learned automatically from training examples. The CNN approach is especially powerful when applied to image recognition because convolution operation captures the 2D nature of the images. By using the convolution kernel to scan an entire image, relatively few parameters need to be learned compared to the total number of operations. CNNs transform the original image layer by layer from the original pixel values to the final class scores.

Architecture for vehicle control

In end-to-end learning system for self-driving cars, weights of the network are trained to minimize the mean-squared error between the steering command output by the network and the command of either the human driver or the adjusted steering command for off-center and rotated images (Figure 4.18). The input image is split into YUV planes and passed to the network. The network has about 27 million connections and 250,000 parameters. The first layer of the network performs image normalization; the normalizer is hard-coded and not adjusted in the learning process and accelerated via GPU processing. The next layer is many number of



FIGURE 4.18. Car cruise-control system.

convolutional layer. They are designed to perform feature extraction, and are chosen empirically through a series of experiments that vary convolutional layer configurations. The many convolutional layers are followed with few fully connected layers, leading to a final output value. The fully connected layers are designed to function as a controller for steering, but by training the system end-to-end, it is not possible to make a clean break between which parts of the network function primarily as feature extractor, and which serve as controller.

Training and simulator

The first step in training a neural network is to select the frames for use. The collected data is labeled with road type, weather condition and the driver's activity (staying in a lane, switching lanes, turning, and so forth). To train a CNN to do lane following, data is selected where the driver is staying in a lane while the rest is discarded. Then the video is sampled at a rate of 10 frames per second because a higher sampling rate would include images that are highly similar, and thus not provide much additional useful information. To remove a bias toward driving straight, the training data includes a higher proportion of frames that represent road curves. After selecting the final set of frames, the data is augmented by adding artificial shifts and rotations to teach the network how to recover from a poor position or orientation. The magnitude of these perturbations is chosen randomly from a normal distribution. The distribution has zero

mean, and the standard deviation is twice the standard deviation that is measured with human drivers. Artificially augmenting the data does add undesirable artifacts as the magnitude increases.

The simulator takes prerecorded videos from a forward-facing on-board camera connected to a human-driven data-collection vehicle, and generates images that approximate what would appear if the CNN were instead steering the vehicle. These test videos are time-synchronized with the recorded steering commands generated by the human driver. Since human drivers don't drive in the center of the lane all the time, there is a need to manually calibrate the lane's center as it is associated with each frame in the video used by the simulator. The simulator transforms the original images to account for departures from the ground truth. Note that this transformation also includes any discrepancy between the human driven path and the ground truth. The transformation is accomplished by the same methods. The simulator accesses the recorded test video along with the synchronized steering commands that occurred when the video was captured. The simulator sends the first frame of the chosen test video, adjusted for any departures from the ground truth, to the input of the trained CNN, which then returns a steering command for that frame.

The CNN steering commands as well as the recorded human-driver commands are fed into the dynamic model of the vehicle to update the position and orientation of the simulated vehicle. The simulator then modifies the next frame in the test video so that the image appears as if the vehicle was at the position that resulted by following steering commands from the CNN. This new image is then fed to the CNN and the process repeats. The simulator records the off-center distance (distance from the car to the lane center), the yaw, and the distance travelled by the virtual car. When the off-center distance exceeds one meter, a virtual-human intervention is triggered, and the virtual-vehicle position and orientation is reset to match the ground truth of the corresponding frame of the original test video. CNNs are able to learn the entire task of lane and road following without manual decomposition into road or lane marking detection, semantic abstraction, path planning and control. A small amount of training data from less than 100 hours of driving is sufficient to train the car to operate in diverse conditions, on highways, local, and residential roads in sunny, cloudy, and rainy conditions. The drive simulator is shown in Figure 4.19.

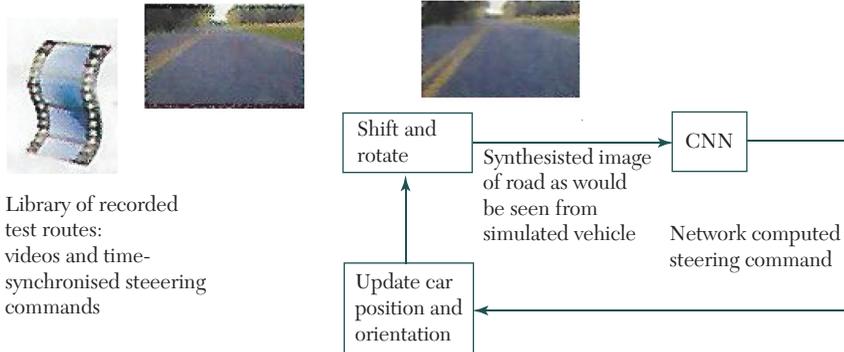


FIGURE 4.19. Drive simulator.

Image identification in autonomous vehicles

To be able to identify images, autonomous vehicles need to process a full 360-degree dynamic environment. This creates the need for dual-frame processing because collected frames must be combined and considered in context with each other. A vehicle can be equipped with a rotating camera to collect all relevant driving data. The machine must be able to recognize metric, symbolic and conceptual knowledge. Metric knowledge is the identification of the geometry of static and dynamic objects, which is required to keep the vehicle in its lane and at a safe distance from other vehicles. Symbolic knowledge allows the vehicle to classify lanes and conform to basic rules of the road. Conceptual knowledge allows the vehicle to understand relationships between traffic participants and anticipate the evolution of the driving scene. Conceptual knowledge is the most important aspect for being able to detect specific objects and avoid collisions.

One current method of obstacle detection in autonomous vehicle is the use of detectors and sets of appearance-based parameters. The first step in this method is the selection of areas of interest. This process narrows down areas of the field of vision that contain potential obstacles. Appearance cues are used by the detectors to find areas of interest. These appearance cues analyze two-dimensional data and may be sensitive to symmetry, shadows, or local texture and color gradients. Three-dimensional analysis of scene geometry provides greater classification of areas of interest. These additional cues include disparity, optical flow and clustering techniques. Disparity is the pixel difference for an object from frame-to-frame. If you look at an object alternately closing one eye after the other, the “jumping” you see in the object is the disparity. It can be used to detect and reconstruct arbitrarily

shaped objects in the field. Optical flow combines scene geometry and motion. It samples the environment and analyses images to determine the motion of objects. Finally, clustering techniques group image regions with similar motion vectors as these areas are likely to contain the same object. A combination of these cues is used to locate all areas of interest.

While any combination of cues is attainable, it is necessary to include both appearance cues and three-dimensional cues as the accuracy of three-dimensional cues decreases quadratically with increasing distance. In addition, only persistent detections are flagged as obstacles so as to lower the rate of false alarms. After areas of interest have been identified, these must be classified by passing them through many filters that search for characteristic features of on-road objects. This method takes a large amount of computation and time. The use of CNNs can increase the efficiency of this detection process. A CNN-based detection system can classify areas that contain any type of obstacle. Motion-based methods such as optical flow heavily rely on the identification of feature points, which are often misclassified or not present in the image.

All the knowledge-based methods are for special obstacles (pedestrians, cars, etc.) or in special environments (flat road, obstacles differing in appearance from ground). Convolutional neural networks are the most promising for classifying complex scenes because these closely mimic the structure and classification abilities of the human brain. Obstacle detection is only one important part of avoiding a collision. It is also vital for the vehicle to recognize how far away the obstacles are located in relation to its own physical boundaries.

Depth estimation

Depth estimation is an important consideration in autonomous driving as it ensures the safety of passengers as well as other vehicles. Estimating the distance between an obstacle and the vehicle is an important safety concern. A CNN may be used for this task as CNNs are a viable method to estimate depth in an image. The researchers trained their network on a large dataset of object scans, which is a public database of over ten thousand scans of everyday 3D objects, focused on images of chairs and used two different loss functions for training. They found that bi-weight trained network was more accurate at finding depth than the L2 norm. With images of varying size and resolution, it has an accuracy between 0.8283 and 0.9720 with a perfect accuracy being 1.0.

While estimating depth on single-frame stationary objects is simpler than on moving objects seen by vehicles, researchers found that CNNs can also be used for depth estimation in driving scenes. They fed detected obstacle blocks to a second CNN programmed to find depth. The blocks were split into strips parallel to the lower image boundary. These strips were weighted with depth codes from bottom to top with the notion that closer objects would normally appear closer to the lower bound of the image. The depth codes went from “1” to “6” with “1” representing the shallow areas and “6” representing the deepest areas. The obstacle blocks were assigned the depth code for the strip they appeared in.

The CNN then used feature extraction in each block area to determine whether vertically adjacent blocks belonged to the same obstacle. If the blocks were determined to be the same obstacle, they were assigned the lower depth code to alert the vehicle of the closest part of the obstacle. CNN was trained on image block pairs to develop a base for detecting depth and then tested on street images as in the obstacle detection method. The CNN had an accuracy of 91.94% in two block identification.

Development of a sustainable driving environment

A mechanism for deciding the best route can be included in the neural network of each vehicle. This route optimization and resulting decrease in traffic congestion is predicted to reduce fuel consumption up to 4%, thus reducing the amount of ozone and environmentally harmful emissions. Each vehicle will have a set of commands for different driving situations. When efficiency and part preservation are priority, commands will be executed to minimize wear on the vehicle and reduce energy consumption by 25%. For example, a human driver struck in traffic might hit the accelerator and then the brake excessively to move every time the traffic inches forward. This causes excessive wear on the engine and brakes of the vehicle. However, an autonomous vehicle system would be optimized to either roll forward at a slow enough rate so that it does not collide with the vehicle ahead, or not move until there is enough free road available. This will decrease wear on the vehicle’s brakes and engine while optimizing fuel efficiency. As a result, the lifespan of each vehicle will extend, thus decreasing demand for new vehicles and vehicle parts. Fewer vehicles manufacturing means conservation of resources such as fuels burnt in factories and metals used in production.

2. CNN Technology Enablers

OPENVX

OPENVX is an open, royalty-free standard for cross-platform acceleration of computer-vision applications. It enables performance and power-optimized computer-vision processing, which is especially important in embedded and real-time-use cases such as face, body, and gesture tracking, smart-video surveillance, advanced driver-assistance systems (ADAS), object and scene reconstruction, augmented reality, visual inspection, and robotics. *OPENVX* is a low-level programming frame-work domain that enables software developers to efficiently access computer-vision hardware acceleration with both functional and performance portability. It has been designed to support modern hardware architectures, such as mobile and embedded SoCs as well as desktop systems. Many of these systems are parallel and heterogeneous, containing multiple processor types including multi-core CPUs, DSP subsystems, GPUs, dedicated vision computing fabrics, as well as hardwired functionality. Additionally, vision-system memory hierarchies can often be complex, distributed, and not fully coherent.

Developers need a set of high-level tools and standard libraries like Open CV and Open VX that work in conjunction with and complement the underlying C/C++ tool chain. OpenCV is an open source computer-vision software library that contains 2500 functions which, when used by a high-level application, can facilitate tasks like object detection and tracking, image stitching, 3D reconstruction, and machine learning.

OpenVX contains a library of predefined and customisable vision functions, a graph-based execution model with task and data independent execution, and a set of memory objects that abstract the physical memory. It defines a “C” application programming interface (API) for building, verifying, and coordinating graph execution, as well as accessing memory objects.

Graph abstraction enables OpenVX implementers to optimize graph execution for the underlying acceleration architecture. It also defines the vxu utility library, which exposes each OpenVX predefined function as a directly callable “C” function, without the need to first create a graph. Applications built using the vxu library do not benefit from optimisation enabled by graphs; however the vxu library can be useful as the simplest way to use OpenVX and as the first step in porting existing vision applications.

As computer-vision domain is still rapidly evolving, OpenVX provides an extensibility mechanism for adding developer-defined functions to the application graph.

Conventional ADAS technology can detect some objects, do basic classification, alert the driver of hazardous road conditions and, in some cases, slow or stop the vehicle. This level of ADAS is great for applications like blind-spot monitoring, lane-change assistance, and forward-collision warnings. NVIDIA DRIVE PX 2 AI car computers take driver assistance to the next level. These take advantage of deep learning and include a software development kit (SDK) for autonomous driving called Drive Works. This SDK gives developers a powerful foundation for building applications that leverage computation-intensive algorithms for object detection, map localisation and path planning. With NVIDIA self-driving car solutions, a vehicle's ADAS can discern a police car from a taxi, an ambulance from a delivery truck, a parked car from a taxi, or a parked car from one that is about to pull out into traffic. It can even extend this capability to identity everything from cyclists on the sidewalk to absentminded pedestrians.

OpenPilot

OpenPilot consists of two component parts: on-board firmware and ground control station (GCS). The firmware is written in “C”, while ground control-station is written in “C++” utilizing Qt. This platform is meant for development only. *OpenPilot* is basically a behavior model based on Comma. The car feeds images from a camera into the network, and out from the network come commands to adjust the steering and speed to keep a car in its lane. As such, there is very little traditional code in the system, just the neural network and a bit of control logic.

The network is built by training it. As a car is driven around, it learns from the human driving what to do when it sees things in the field of view. Light detection and ranging (LIDAR) gives the car an accurate 3D scan of the environment to more absolutely detect the presence of cars and other users of the road. By getting to know during training that there is really something there at these coordinates, the network can learn how to tell the same thing from just the camera images. When it is time to drive, the network does not get the LIDAR data, however it does produce outputs of where it thinks the other cars are, allowing developers to test how well it is seeing things. This allows development of a credible autopilot, but, at the same time, developers have minimal information about how it works,

and can never truly understand why it is making the decisions it does. If it makes an error, they will generally not know why it made the error, though they can give it more training data until it no longer makes the error.

Developing an autonomous car requires a lot of training data. To drive using vision, developers need to segment 2D images from cameras into independent objects to create a 3D map of the scene. Other than using parallax, relative motion, and two cameras to do that, autonomous cars also need to see patterns in the shapes to classify them when they see them from every distance and angle, and in every lighting condition. That's been one of the big challenges for vision—one must understand things in sunlight, in diffuse light, in night time (when illuminated by headlights, for example), when the sun is pointed directly into the camera, when complex objects like trees are casting moving shadows on the desired objects, and so on. One must be able to identify pavement marking, lane marking, shoulders, debris, other road users, and so on in all these situations and must do it so well that one make a mistake perhaps only once every million miles. A million miles is around 2.7 billion frames of video.

3. Smart Fashion AI Architecture

AI stylist programs consider three major components:

1. Visual Garment Representation—a garment database that stores garment items by categories.
2. Computational Styling—processes styling rules and assigns popularity to individual items and completed looks.
3. Fashion-Trend Tracking—a learning component that reads users' feedback and adjusts weight in the style-engine based on the popularity rank of a completed look and user opinions on styling looks.

Computer-vision techniques can extract attribute information from an image and AI techniques such as semantic mapping enable the program to deal with tricky attributes such as fabric. AI programs have the ability to execute a fashion-styling task with a simplified model. An AI-based program is capable of handling more attributes such as event, clothes, shoes, accessories, makeup, and hair styling. Complete fashion design usually includes details such as a top with a bottom or dress, shoes, accessories, bags, hair styles, and makeup. Support vector machine classifiers are applied to all attributes such as color, shape, print, and fabric to determine how useful these attributes would be in prediction. Then, the system makes

predictions based on the inference of different attributes and mutual dependency relations.

Each individual has a unique style preference. An intelligent style system can instantaneously adapt from a standard color scheme to suit a user's personal preference. Linguistic labels (neutral, a little, slightly, fairly, very, and extremely) are added to the color scheme and presented as fuzzy sets, respectively. The system adapts to a user's preference during its interaction with the user. AI methods include:

1. Fuzzy logic. It utilizes uncertainty and approximate reasoning and works closer to human brain, providing outputs as straightforward like or dislike.
2. Artificial neural networks (ANNs). This is a learning method, resembling animal nervous systems, mapping input to a target output by adjusting weights. It is suitable for modeling complex styling tasks with multiple features.
3. Decision trees. This method is used in human decision-making models and includes tree-structured graphs that represent attributes as internal nodes and outcome as branches.
4. Knowledge-based systems. These are programs that represent knowledge and solve complex problems by reasoning out how knowledge artifacts are related or not related. These are used to show relationships between features in fashion styling.
5. Genetic algorithms. These are search techniques that look for approximate or exact solutions to optimization problems. They are guided by a fitness function.

AI computer modelling program usually includes the following functions: Representation of the garments computationally (focuses on fashion object), detection, tracking and forecasting fashion trends and modelling human stylist behavior. Smart fashion is a machine-learning application that recommends fashion looks by learning user preferences through Multilayer Perception model. The system scores user customized style looks based on fashion trends and users personal style history.

Smart-fashion system comprises five major components for fashion-styling tasks: initialization program, database, smart engine, learning components, and user interface. It uses artificial neural networks to learn

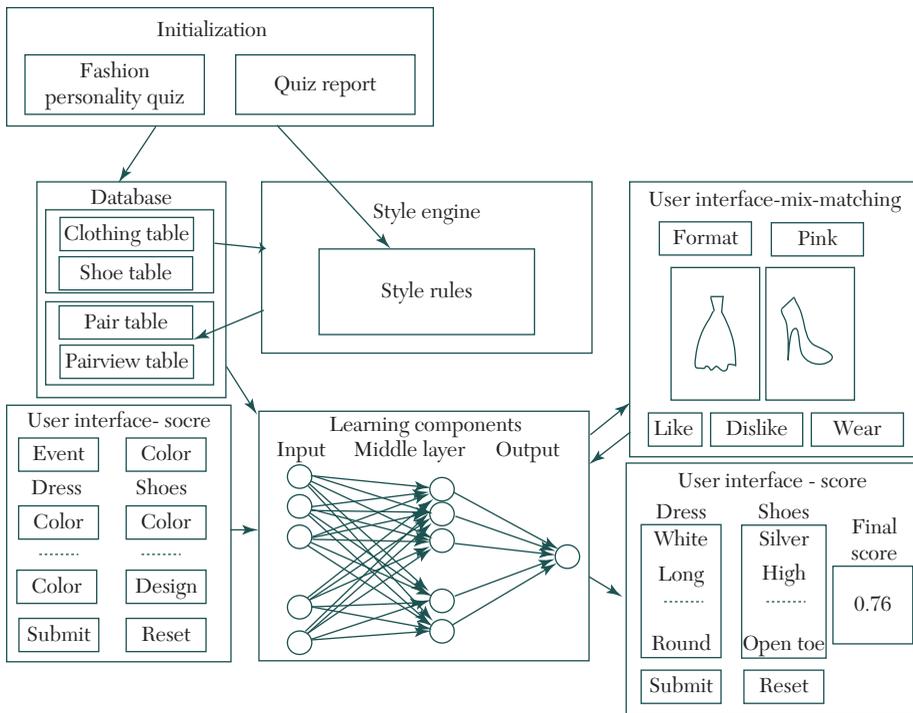


FIGURE 4.20. Schematic overview of smart fashion architecture

user preferences. It creates a database containing, for example, 32 dresses and 20 shoes for four different events, encodes a standard style rules engine, generates more than 500 looks and ranks them by a final score in descending order. The score indicates how fashionable each look is based on user feedback. The learning component trains artificial neural networks to learn user personal preferences and adjusts the final score. The system allows users to customize a fashion look and then evaluate it. This feature provides a shopping guide to inform user's purchase plans. Figure 4.20 shows the smart-fashion architecture.

4. Teaching Computers to Recognize Cats

A simple example can be used showing how a computer can recognize a cat from other animals. Machine learning is a rapidly growing field that enables computers to learn patterns found in the objects surrounding us. This may include tasks such as detection of an object from an image, speech recognition, autonomous driving, and much more. The steps in machine learning are as follows:

Data collection

Like human beings, computers must be trained to distinguish between two types of animals such as dogs and cats by presenting these with a series of images. This is referred to as a training set of data. Figure 4.21 shows one such training set, which contains a few images of cats and dogs. The larger and more diverse the training set, the better a computer can perform a learning task.

Features designing

Think about how human beings are able to tell the difference between images of cats and dogs. What do they look for? Human beings use color, size, shape of ears or nose, and/or a combination of all to distinguish between the two animals. Humans do not look at an image as only a collection of pixels. Instead, humans pick out features from the image to identify either animal. This is true for computers as well. To successfully train a computer to perform the task of classification, programmers need to provide it with properly designed features. This is not an easy task, because designing good quality features is application dependent. For example, a feature like number of legs would be unhelpful in distinguishing between cats and dogs, as both have four legs. But, it would be useful in discriminating between cats and snakes. Extracting features from an image can be challenging at times. For instance, if an image is blurred, features will not be properly extracted. For example, humans can easily extract two features from each image of the training set, that is, size of the nose (relative to size of the head—from small to big) and shape of the ears (from round to pointed). By examining the images in the training set (Figure 4.21), human beings can observe that all cats have a small nose and pointed ears, while dogs have a long nose and round ears.

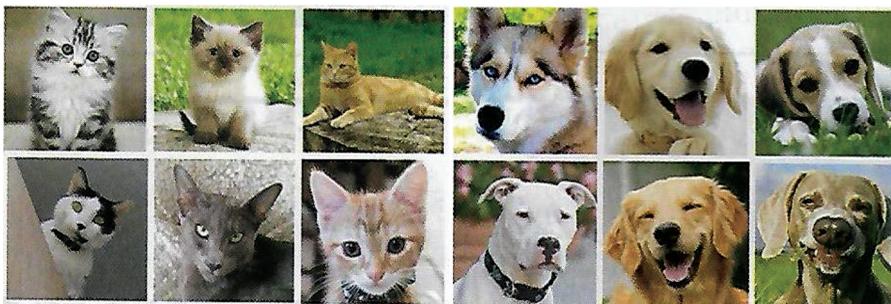


FIGURE 4.21. Training set comprising cats and dogs.

With the current choice of features, each image can be represented using a number expressing relative nose size and another number capturing pointed or round ears. Hence, the programmer can represent training set in a two-dimensional feature space where the feature nose size and ears' shape are on horizontal and vertical axes, respectively, as shown in Figure 4.22.

Training

After having a good feature representation of the images in the training set, the final task is to teach a computer how to distinguish between cats and dogs. It is a simple geometric problem where the computer must find

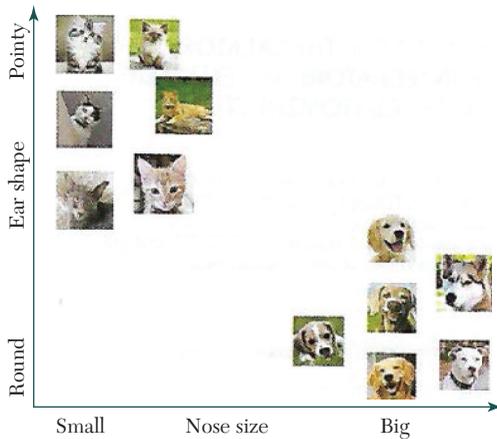


FIGURE 4.22. Representation of training set in terms of features.

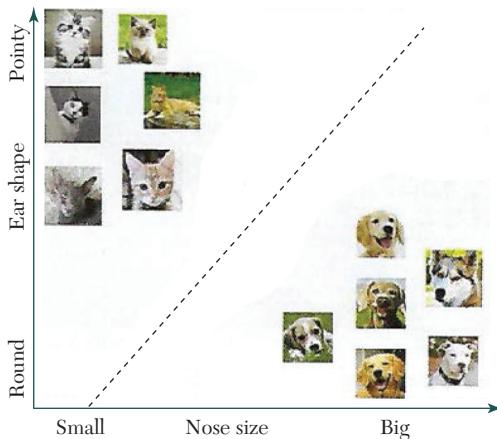


FIGURE 4.23. Trained linear model (dotted lines) separating two classes of animals.

a line that separates the two images in the designed feature space. Since the line is parameterized by a slope and intercept, finding the right values for both is important. Figure 4.23 shows the trained line that divides the feature space into cats and dogs, respectively. Once the line is obtained, any image added later, whose features lie above the line, is considered a cat by the computer. Any image that falls below the line is considered a dog.

Model testing

To test the computer on how well it has learned to distinguish between cats and dogs, provide it with a series of images not seen earlier and see how well it can identify the animal in each image. In Figure 4.24, a sample of images (of new cats and dogs), not seen by the computer during training phase, is shown. What the computer does is, it takes the features into account and checks which side of the line these images fall. For instance, it is observed in Figure 4.25 that all new cats and dogs (except one) from the testing set were identified accurately. Misidentification of one dog (shown in second image at top of Figure) is due to the choice of features that were



FIGURE 4.24. Testing set of cat and dog images not included in training set.

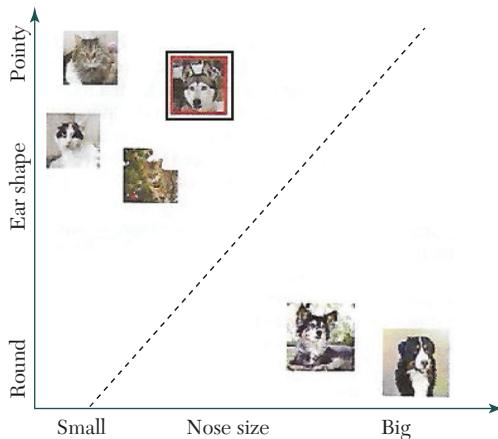


FIGURE 4.25. Classification of test images using trained linear model.

designed using the training set, as shown in Figure 4.24. This dog was identified as a cat because the size of its node and shape of its ears match that of cats in the training set. To avoid this issue, train the computer with more data and use more discriminating features that will help distinguish cats and dogs better.

Summary

- Video analytics is used for motion detection, facial recognition, license plate reading, people counting, and many more.
- Optical flow is a technique used to estimate the pattern of apparent motion of objects, surfaces, and edges in a video sequence.
- Neural networks consist of an input layer, several hidden internal layers, and output layers in machine learning.
- Convolution, rectified linear unit, pooling, and fully connected are elements in CNN.
- Types of machine learning-algorithms are NN, K nearest neighbor, SVM, Decision tree, Naïve Bayes algorithm, and deep learning.
- AI for smart-fashion has database, computational styling, and learning component for fashion-trend tracking.

References

<https://www.videosurveillance.com/tech/video-analytics.asp>

<http://www.clearview-communications.com/cctv/facial-recognition-video-analytics>

http://in.nec.com/en_IN/products/public-safety-security/technology/video-analytics.html

<https://www.qognify.com/video-analytics/>

<https://www.ifsecglobal.com/5-reasons-why-video-analytics-is-now-the-dominant-tech-in-cctv-surveillance/>

<https://www.embedded-vision.com/camera-performance-analytics>

Learning Outcomes

- 4.1 Define video analytics.
- 4.2 What are the steps involved in a license-plate recognition system?
- 4.3 List some applications of video analytics.
- 4.4 What are the algorithm pipeline for embedded video analytics?
- 4.5 What is optical flow?
- 4.6 What are the layers for neural network in machine learning?
- 4.7 Write about convolutional neural networks for autonomous cars.
- 4.8 Explain smart-fashion AI architecture.
- 4.9 What are the types of machine learning for vision system?
- 4.10 Write the steps in machine learning for teaching computers to recognize cats.

Further Reading

1. *Video Analytics for Business Intelligence* by Shan, C., Porikli, F., Xiang, T., Gong, S.
2. *Pattern Recognition and Machine Learning* by Bishop and Christopher.

DIGITAL IMAGE PROCESSING

Overview

An image is two-dimensional input for embedded vision system. This continuous time varying signal is processed for machine vision, robot vision, color processing, pattern recognition, medical field and remote sensing. Image sharpening, restoration, histogram, transformation, convolution, edge detection, frequency domain, color spaces, and JPEG compression are the few processing techniques of digital image. Smoothing is to reduce and improve the visual quality of the image. Linear and nonlinear algorithms are used for filtering the images. Pattern matching is a technique used to locate specified patterns within an image. Template is a subpart of an object that is to be matched among entirely different objects.

Learning Objectives

After reading this the reader will be able to understand:

- the differences between image, image signal, digital, and analog image signal
- functions such as zoom, blur, sharp, brightness, histogram, and edges in the image
- different transformation such as linear, logarithmic, and power law for image enhancement
- blurring masks such as mean, weighted average, and Gaussian filter
- different types of edge detection with different masks
- filtering methods in frequency domain and compression technique
- pattern matching and template matching

5.1 IMAGE PROCESSING CONCEPTS FOR VISION SYSTEMS

Embedded vision system dealt with videos and images for extracting information. Main important input in vision system is the image and processing image to get meaningful data. Signal processing is a discipline in electrical engineering and in mathematics that deals with analysis and processing of analog and digital signals, and also deals with storing, filtering, and other operations on signals. These signals include transmission signals, sound or voice signals, image signals, and other signals, etc. Out of all these signals, the field that deals with the type of signals for which the input is an image and the output is also an image is done in image processing. As its name suggests, it deals with the processing of images. It can be further divided into analog image processing and digital image processing. Analog image processing is done on analog signals. In this type of processing, the images are manipulated by electrical means by varying the electrical signal. Digital image processing has dominated over analog image processing with the passage of time due its wider range of applications. The digital image processing deals with developing a digital system that performs operations on a digital image.

Image

An image is nothing more than a two-dimensional signal. It is defined by the mathematical function $f(x,y)$ where x and y are the two coordinates horizontally and vertically. The value of $f(x,y)$ at any point gives the pixel value at that point of an image. Figure 5.1 is an example of digital image. But actually, this image is nothing but a two-dimensional array of numbers ranging between 0 and 255 as shown in matrix Table 5.1.

Each number represents the value of the function $f(x,y)$ at any point. In this case the values are 128 , 232 , 123, and so on. Each represents an individual pixel value. The dimensions of the picture are actually the dimensions of this two-dimensional array.



FIGURE 5.1. Digital image.

128	30	123
232	123	321
123	77	89
80	255	255

TABLE 5.1. Two-dimensional array of digital image.

Signal

In physical world, any quantity measurable through time over space or any higher dimension can be taken as a signal. A signal can be one-dimensional, two-dimensional, or higher-dimensional. A one-dimensional signal is a signal that is measured over time. The common example is a voice signal. The two-dimensional signals are those that are measured over some other physical quantities. An example of a two-dimensional signal is a digital image. Anything that conveys information or broadcasts a message in the physical world between two observers is a signal. This includes speech or (human voice) or an image as a signal.

A signal can be an analog quantity; that means it is defined with respect to the time. It is a continuous signal. They are difficult to analyze, as they carry a huge number of values. They are very much accurate due to a large sample of values. In order to store these signals, you require an infinite memory because it can achieve infinite values on a real line. As compared to analog signals, digital signals are very easy to analyze. They are discontinuous signals. They are the appropriation of analog signals. The word digital stands for discrete values and hence it means that they use specific values to represent any information. With digital signals, only two values are used to represent something, that is, 1 and 0 (binary values). Digital signals are less accurate than analog signals because they are the discrete samples of an analog signal taken over some period of time. However, digital signals are not subject to noise. So they last long and are easy to interpret. Table 5.2 shows the difference between analog and digital signals.

TABLE 5.2. Differences Between Analog and Digital Signals

Comparison element	Analog signal	Digital signal
Analysis	Difficult	Discontinuous
Accuracy	More accurate	Less accurate
Storage	Infinite memory	Easily
Subject to noise	Yes	No
Recording technique	Original signal is preserved	Samples of the signal are taken and preserved
Examples	Human voice, thermometer, analog phones, etc.	Computers, digital phones, digital pens, etc.

Systems

A system is a defined by the type of input and output it deals with. Types of system can be a mathematical model, a piece of code/software,

or a physical device, or a black box whose input is a signal and it performs some processing on that signal, and the output is a signal. The input is known as excitation and the output is known as response.

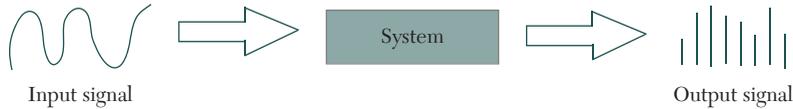


FIGURE 5.2. A system.

In Figure 5.2, a system has been shown whose input and output both are signals but the input is an analog signal. And the output is a digital signal. It means our system is actually a conversion system that converts analog signals to digital signals. There are lot of concepts related to this analog to digital conversion and vice-versa. There are two main concepts involved in the conversion: sampling and quantization. Sampling, as its name suggests, can be defined as taking samples, such as taking samples of a digital signal over x axis. Sampling is done on an independent variable.

Sampling is done on the x variable as shown in Figure 5.3(a). The conversion of x -axis (infinite values) to digital is done under sampling. Quantization, as its name suggests, can be defined as dividing into quanta (partitions). Quantization is done on dependent variable. It is opposite to sampling as in Figure 5.3(b). It is done on the y axis. The conversion of y -axis infinite values to 1, 0, -1 (or any other level) is known as *quantization*. These are the two basic steps that are involved while converting an analog signal to a digital signal. Whenever the image is captured, it is converted into digital format and then it is processed. The important reason is, that in order to perform operations on an analog signal with a digital computer,

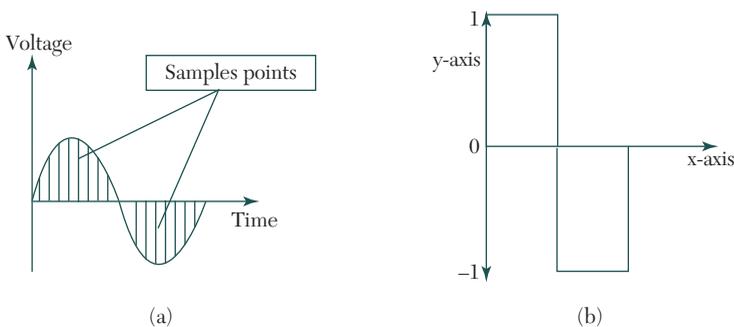


FIGURE 5.3. (a) Sampling. (b) Quantization

one has to store the analog signal in the computer. And in order to store an analog signal, infinite memory is required. And since that's not possible, one should first convert that signal into digital format, then store it in a digital computer, and then perform operations on it.

Continuous systems versus discrete systems

The type of systems whose input and output both are continuous signals or analog signals are called continuous systems as shown in Figure 5.4(a). The type of systems whose input and output are both discrete signals or digital signals are called digital systems as shown in Figure 5.4(b).

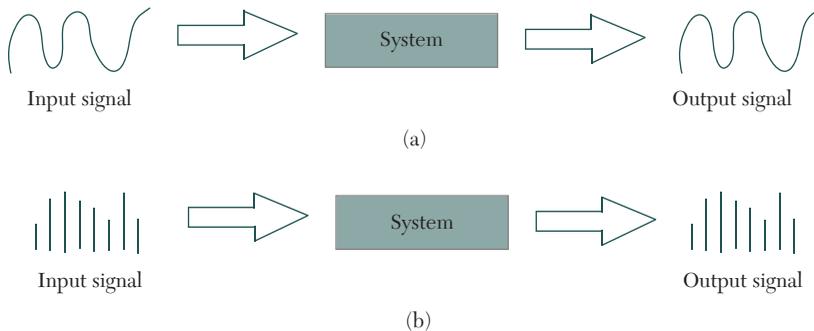


FIGURE 5.4. (a) Continuous system. (b) Discrete systems.

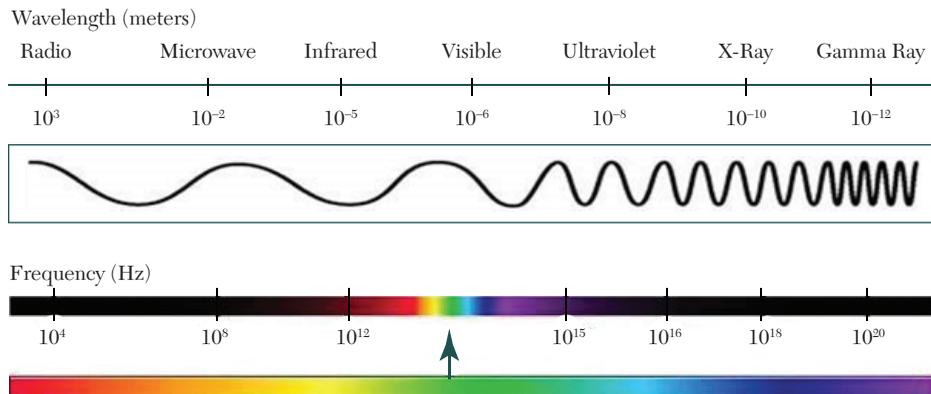


FIGURE 5.5. Electromagnetic spectrum.

Electromagnetic waves can be thought of as stream of particles, where each particle is moving with the speed of light. Each particle contains a bundle of energy. This bundle of energy is called a photon. The electromagnetic spectrum according to the energy of photons is shown in

Figure 5.5. In this electromagnetic spectrum, we are only able to see the visible spectrum. Visible spectrum mainly includes seven different colors that are commonly termed as (VIBGOYR). VIBGOYR stands for violet, indigo, blue, green, orange, yellow, and red. Our human eye can only see the visible portion, in which we see all the objects. But a camera can see the other things that a naked eye is unable to see. For example: X-rays, gamma rays, and so on. Thus, the analysis of all these subjects is done in digital image processing. X-ray has been widely used in the medical field. The analysis of gamma rays is necessary because they are widely used in nuclear medicine and astronomical observation.

Some of the major fields in which digital image processing is widely used are image sharpening and restoration, in the medical field, remote sensing, transmission and encoding, machine/robot vision, color processing, pattern recognition, video processing, microscopic imaging, and others. The common applications of DIP in the field of medical is gamma ray imaging, PET scan, X-ray imaging, medical CT, and UV imaging. In the field of remote sensing, the area of the earth is scanned by a satellite or from a very high ground and then it is analyzed to obtain information about it. One particular application of digital image processing in the field of remote sensing is to detect infrastructure damages caused by an earthquake as shown in Figure 5.6(a). Because the area affected by an earthquake is sometimes so wide, it is not possible to examine it with the human eye in order to estimate damages. So a solution to this can be found in digital image processing. An image of the affected area is captured from the above ground and then it is analyzed to detect the various types of damage done by the earthquake.

Hurdle detection enables robots see things, identify them, identify the hurdles, and so on. Hurdle detection is one of the common tasks that is done through image processing, by identifying different types of objects in the



(a)



(b)



(c)

FIGURE 5.6. (a) Remote sensing (b) Hurdle detection (c) Line follower.

image and then calculating the distance between the robot and the hurdles as shown in Figure 5.6 (b). Most of the robots today work by following the line, and thus are called line-follower robots. This helps a robot to move on its path and perform some tasks. This has also been achieved through image processing as shown in Figure 5.6 (c).

5.2 IMAGE MANIPULATIONS

Image sharpening, restoration, histograms, transformation, convolution, and edge detection are few of operations done on image.

Image Sharpening and Restoration

Image sharpening and restoration refers here to process images that have been captured from a modern camera to make them a better image or to manipulate those images in way to achieve the desired result. It refers to what Photoshop usually does. This includes zooming, blurring, sharpening, greyscale to color conversion, detecting edges, and vice versa, image retrieval and image recognition. The common examples are shown in Figure 5.7.



FIGURE 5.7 Image sharpening and restoration.

Histograms

A *histogram* is a graph that shows the frequency of anything. Usually histograms have bars that represent frequency of occurring of data in the

whole data set. A histogram has two axis. The x axis and y axis. The x axis contains an event whose frequency you have to count. The y axis contains frequency. The different heights of the bars show different frequencies of occurrence of data as shown in Figure 5.8(a). An image histogram, shows the frequency of pixels' intensity values. In an image histogram, the x axis shows the grey-level intensities and the y axis shows the frequency of these intensities. For example: The histogram of the picture of the APJ Abdulkalam would be something like in Figure 5.8(b).

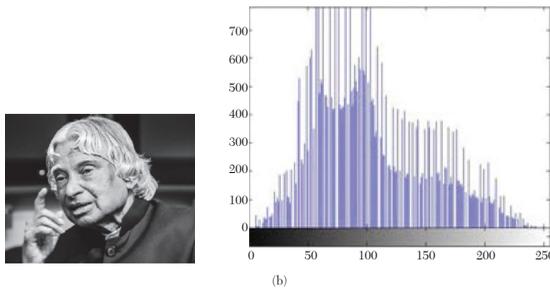
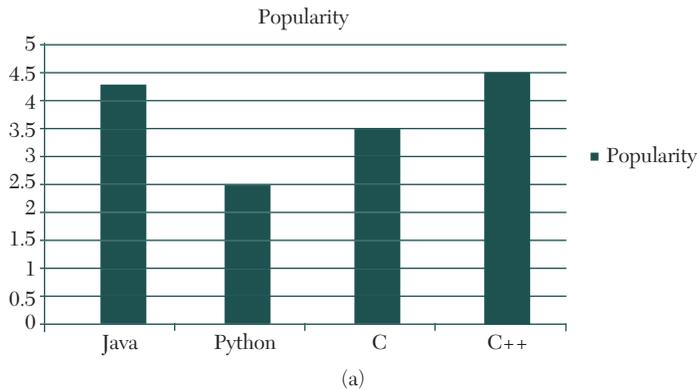


FIGURE 5.8. (a) Example. (b) APJ Abdulkalam picture and its histogram.

The x axis of the histogram shows the range of pixel values. Since it's an 8 bpp image, which means it has 256 levels of grey or shades of grey in it. That's why the range of x axis starts from 0 and end at 255 with a gap of 50. Whereas on the y axis, is the count of these intensities.

Applications of histograms

Histograms have many uses in image processing. It's like looking an X-ray of a bone in a body. Histograms can also be used brightness purposes.

Histograms have a wide application in image brightness. Histograms are also used in adjusting contrast of an image. Another important use of the histogram is to equalize an image. Histograms have wide uses in thresholding. This is mostly used in computer vision. Since brightness is a relative term, brightness can be defined as the amount of energy output by a source of light relative to the source to which it is being compared. In some cases we can easily say that the image is bright, and in some cases, it's not easy to perceive.

For example, in Figure 5.9 we can easily see that the image on the right side is brighter as compared to the image on the left. But if the image on the right is made darker than the first one, then we can say that the image on the left is brighter than the right. Brightness can be easily increased or decreased by simple addition or subtraction to the image matrix. Contrast can be simply explained as the difference between maximum and minimum pixel intensity in an image. For example, consider the image in Figure 5.10:



FIGURE 5.9. Sample picture with different brightness.



100	100	100
100	100	100
100	100	100

FIGURE 5.10. Contrast analysis and matrix of this image.

The maximum value in this matrix is 100. The minimum value in this matrix is 100. Contrast = maximum pixel intensity (subtracted by) minimum pixel intensity = 100 (subtracted by) 100 = 0; 0 means that this image has 0 contrast.

Transformation

Transformation is a function that maps one set to another set after performing some operations. A digital image processing system would

perform some processing on the input image and give its output as a processed image, as shown in Figure 5.11. Function applied inside this digital system that process an image and converts it into output is called transformation function. Image 5.11 shows how the image on the left is converted into the image on the right.

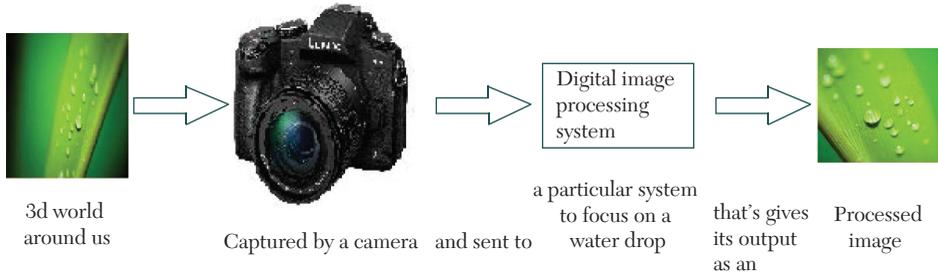


FIGURE 5.11. Transformation.

Image transformation

Consider this equation: $G(x,y) = T\{ f(x,y) \}$. In this equation, $F(x,y)$ = input image on which transformation function has to be applied. $G(x,y)$ = the output image or processed image. T is the transformation function. This relation between input image and the processed output image can also be represented as, $s = T(r)$; where r is actually the pixel value or grey-level intensity of $f(x,y)$ at any point. And s is the pixel value or grey-level intensity of $g(x,y)$ at any point.

Image enhancement

Enhancing an image provides better contrast and a more detailed image as compared to a nonenhanced image. Image enhancement has many applications. It is used to enhance medical images, images captured in remote sensing, images from a satellite, and so on. The transformation function has been given below $s = T(r)$. where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of r to each value of s . Image enhancement can be done through grey-level transformations. There are three basic grey-level transformations. They are linear, logarithmic, and power law.

Linear transformation

Linear transformation includes simple identity and negative transformation. Identity transition is shown by a straight line. In this transition,

each value of the input image is directly mapped to each other value of the output image. That results in the same input image and output image. And hence is called identity transformation, as shown in Figure 5.12 (a).

The second linear transformation is negative transformation, which is an invert of identity transformation. In negative transformation, each value of the input image is subtracted from the $L-1$ and mapped onto the output image. In this case the following transition has been done: $s = (L - 1) - r$; since the input image of APJ Abdulkakam is an 8 bpp image, so the number of levels in this image are 256. Putting 256 in the equation, we get this, $s = 255 - r$; therefore each value is subtracted by 255 and the resulting image is shown in Figure 5.12 (b). So the lighter pixels become dark and the darker picture becomes light and it results in image negative. It has been shown in Figure 5.12.

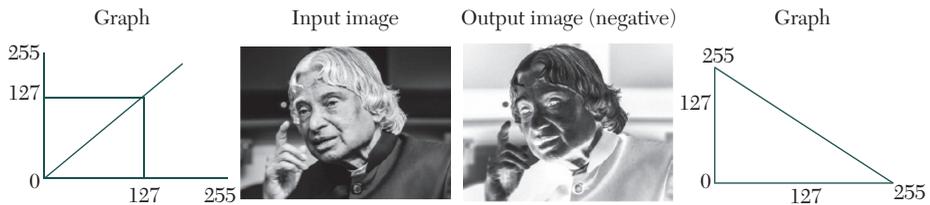


FIGURE 5.12. (a) Identity transformation graph; (b) Negative transformation input, output, and graph.

Logarithmic transformations

Logarithmic transformation further contains two types of transformation. Log transformation and inverse log transformation. The log transformations can be defined by this formula: $s = c \log(r + 1)$, where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel values of the input image because if there is a pixel intensity of 0 in the image, then $\log(0)$ is equal to infinity. So 1 is added, to make the minimum value at least 1. During log transformation, the dark pixels in an image are expanded as compared to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This results in following image enhancement as shown in Figure 5.13. The inverse log transform is opposite to log transform.



FIGURE 5.13. Input image and log transformation image.

Power law transformations

Power law transformations include n th power and n th root transformation. These transformations can be expressed as follows: $s = cr^\gamma$; this symbol γ is called gamma, due to which this transformation is also known as gamma transformation. Variation in the value of γ varies the enhancement of the images. This type of transformation is used for enhancing images for different types of display devices. The gamma of different display devices is different. For example the gamma of CRT lies in between 1.8 and

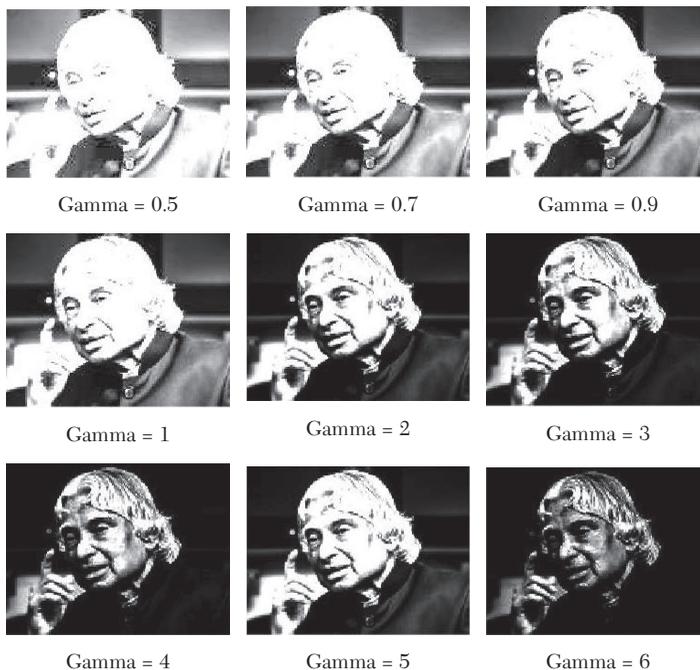


FIGURE 5.14. Different gamma values for the APJ Abdul Kalam image.

2.5, which means the image displayed on CRT is dark. The same image but with different gamma values has been shown in Figure 5.14.

Convolution concept

Convolution can be mathematically represented in two ways: $g(x,y) = h(x,y) * f(x,y)$, which can be explained as the “mask convolved with an image.” Or it can be represented as $g(x,y) = f(x,y) * h(x,y)$, which can be explained as “image convolved with mask” as in Figure 5.15. There are two ways to represent this because the convolution operator(*) is commutative. The $h(x,y)$ is the mask or filter. Mask is also a signal that can be represented by a two-dimensional matrix. The mask is usually of the order of 1x1, 3x3, 5x5, 7x7. A mask should always be an odd number, because otherwise you cannot find the mid of the mask. In order to perform convolution on an image, following steps should be taken.



FIGURE 5.15. Convolution system.

- Flip the mask (horizontally and vertically) only once.
- Slide the mask onto the image.
- Multiply the corresponding elements and then add them.
- Repeat this procedure until all values of the image have been calculated.

Let’s take mask to be this

1	2	3
4	5	6
7	8	9

Flipping the mask horizontally

3	2	1
6	5	4
9	8	7

Flipping the mask vertically

9	8	7
6	5	4
3	2	1

Image: Let's consider an image to be like this

```

2  4  6
8  10 12
14 16 18

```

When convolving mask over the image place the center of the mask at each element of the image. Multiply the corresponding elements, add them, and paste the result onto the element of the image on which you place the center of mask as in Figure 5.16.

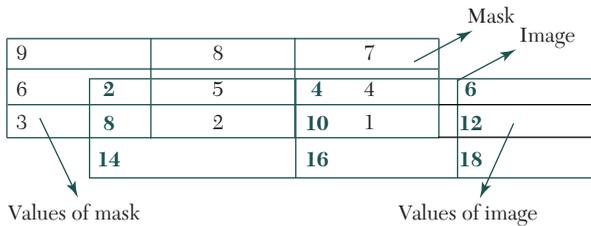


FIGURE 5.16. Convolution mask.

For the first pixel of the image, the value will be calculated as first pixel = $(5*2) + (4*4) + (2*8) + (1*10) = 10 + 16 + 16 + 10 = 52$. Place 52 in the original image at the first index and repeat this procedure for each pixel of the image. Convolution can achieve something that includes the blurring, sharpening, edge detection, noise reduction, and so on. The process of filtering is also known as convolving a mask with an image. This process is the same as convolution, so filter masks are also known as convolution masks. The general process of filtering and applying masks consists of moving the filter mask from point to point in an image. At each point (x,y) of the original image, the response of a filter is calculated by a predefined relationship. Generally there are two types of filters. One is called a linear filter or smoothing filter, and the other is called a frequency domain filter. Filters are applied on an image for multiple purposes. The two most common uses are as follows:

- Filters are used for blurring and noise reduction.
- Filters are used for edge detection and sharpness.

Blurring and noise reduction

Filters are most commonly used for blurring and noise reduction. Blurring is used in preprocessing steps, such as removal of small details

from an image prior to large object extraction. The common masks for blurring are:

- mean filter
- weighted average filter
- Gaussian filter

In the process of blurring, reduces the edge content in an image and try to make the transitions between different pixel intensities as smooth as possible. Noise reduction is also possible with the help of blurring. An image looks sharper or more detailed if we are able to perceive all the objects and their shapes correctly in it. For example, an image with a face, looks clear when we are able to identify eyes, ears, nose, lips, forehead, etc very clear. This shape of an object is due to its edges. So in blurring, simply reduces the edge content and makes the transition from one color to the other very smooth. When you zoom an image using pixel replication, and zooming factor is increased, you saw a blurred image. This image also has less details, but it is not true blurring. Because in zooming, you add new pixels to an image that increase the overall number of pixels in an image, whereas in blurring, the number of pixels of a normal image and a blurred image remains the same. Common example of a blurred image is shown in Figure 5.7(c).

Mean filter

Mean filter is also known as box filter and average filter. The properties of a mean filter are that it must be odd ordered, the sum of all the elements should be 1, and all of the elements should be the same. If we follow this rule, then for a mask of 3×3 we get the following result:

$$\begin{array}{ccc} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{array}$$

3×3 mask has nine cells. The condition that the sum of all elements should be equal to one can be achieved by dividing each value by nine: $1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 = 9/9 = 1$. The blurring can be increased by increasing the size of the mask. The bigger the size of the mask, the more blurring, as shown in Figure 5.17. The greater the mask, the greater number of pixels are catered and one smooth transition is defined.

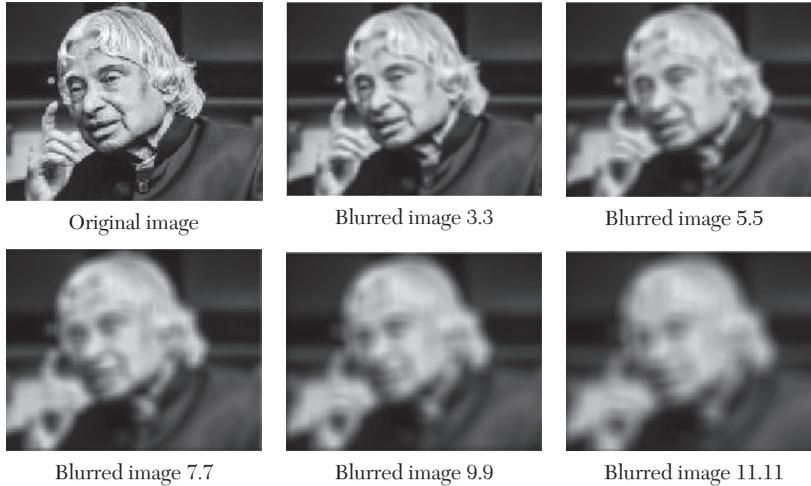


FIGURE 5.17. Different blurring images based on the size of the mask.

Weighted average filter

In a weighted average filter, we gave more weight to the center value, because the contribution of the center becomes more than the rest of the values. Due to weighted average filtering, we can actually control the blurring. Properties of the weighted average filter are:

1. It must be odd ordered.
2. The sum of all the elements should equal 1.
3. The weight of the center element should be more than all of the other elements.

Filter 1

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{array}$$

The two properties are satisfied which are (1 and 3). But the property 2 is not satisfied. So in order to satisfy that we will simply divide the whole filter by 10, or multiply it by 1/10.

Filter 2

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 1 \end{array}$$

Dividing factor = 18.

Sharpening

Sharpening is opposite of blurring. In blurring, we reduce the edge content, and in sharpening we increase the edge content. So in order to increase the edge content in an image, we have to find edges first. Edges can be found by using any of the previously described methods using any operator. After finding edges, we will add those edges to an image and thus the image would have more edges and would look sharpened. This is one way of sharpening an image. The sharpened image is shown in Figure 5.7(d).

Edge Detection

Masks or filters can also be used for edge detection and to increase sharpness of an image. The sudden changes of discontinuities in an image are called edges. Significant transitions in an image are called edges. A picture with edges is shown in Figure 5.7 (e). Generally there are three types of edges:

- Horizontal edges
- Vertical Edges
- Diagonal Edges

Most of the shape information of an image is enclosed in edges. Here are some of the masks for edge detection.

- Prewitt operator
- Sobel operator
- Robinson compass masks
- Krisch compass masks
- Laplacian operator

All of the preceding filters are linear filters or smoothing filters. Prewitt operator is used for detecting edges horizontally and vertically. The Sobel operator is very similar to the Prewitt operator. It is also a derivate mask and is used for edge detection. Plus it calculates edges in both horizontal and vertical directions. Robinson compass masks operator is also known as direction mask. In this operator, one mask is rotated in all eight major compass directions to calculate the edges of each direction. Kirsch compass

mask is also a derivative mask used for finding edges. The Kirsch mask is also used for calculating edges in all the directions. Laplacian operator is also a derivative operator which is used to find edges in an image. Laplacian is a second order derivative mask. It can be further divided into positive Laplacian and negative Laplacian. All these masks find edges. Some find edges horizontally and vertically, some find only one direction, and some find all directions.

Prewitt operator

Edges are calculated by using the difference between corresponding pixel intensities of an image. All the masks that are used for edge detection are also known as derivative masks or derivative operators. All the derivative masks should have the following properties:

- The opposite sign should be present in the mask.
- The sum of a mask should be equal to zero.
- More weight means more edge detection.

The Prewitt operator provides us two masks, that is, one for detecting edges in a horizontal direction and another for detecting edges in a vertical direction.

Vertical Direction

$$\begin{array}{ccc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array}$$

The above masks will find edges in a vertical direction because of the zeros column in the vertical direction. When this mask is convolved on an image, it will give the vertical edges in an image. It simply works like a first order derivative and calculates the difference of pixel intensities in an edge region. The center column is zero so it does not include the original values of an image, but rather it calculates the difference of right and left pixel values around that edge. This increases the edge intensity and it becomes enhanced comparatively to the original image.

Horizontal Direction

$$\begin{array}{ccc} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{array}$$

The above masks will find edges in a horizontal direction because the zeros column is in a horizontal direction. When this mask is convolved onto an image, horizontal edges in the image are prominent. As the center row of mask consists of zeros, it does not include the original values of edges in the image, but rather it calculates the difference of above and below pixel intensities of the particular edge, thus increasing the sudden change of intensities and making the edge more visible. Both the masks follow the principle of derivate masks, have opposite signs in them, and the sum of both masks equals zero. The third condition will not be applicable in this operator as both the above masks are standardized and we can't change the value in them. After applying a vertical mask onto the APJ Abdulkalam image, the image contains vertical edges. After applying a horizontal mask onto the same image, the image obtained is shown in Figure 5.18.

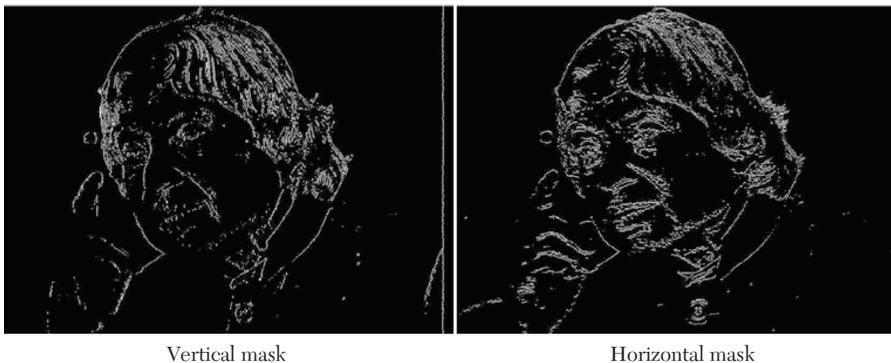


FIGURE 5.18. Prewitt edge detection.

In the first picture on which the vertical mask is applied, all the vertical edges are more visible than the original image. Similarly, in the second picture the horizontal mask is applied and all the horizontal edges are visible. So it is possible to detect both horizontal and vertical edges from an image.

Sobel Operator

The Sobel operator is very similar to Prewitt operator. It is also a derivate mask and is used for edge detection. Like the Prewitt operator, the Sobel operator is also used to detect two kinds of edges in an image: They are vertical direction and horizontal direction. The major difference is that with the Sobel operator the coefficients of masks are not fixed, and they can be adjusted according to requirement unless they do not violate

any property of derivative masks. Following is the vertical mask of the Sobel operator:

$$\begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array}$$

This mask works exactly the same as the Prewitt operator vertical mask. There is only one difference: it has “2” and “-2” values in center of the first and third columns. When applied on an image this mask will highlight the vertical edges. When this mask is applied on the image it vertical edges are prominent. It simply works like a first-order derivate and calculates the difference of pixel intensities in an edge region. The center column is zero, and it does not include the original values of an image, but rather it calculates the difference of right and left pixel values around that edge. Also the center value of both the first and third column is 2 and -2 respectively. This gives more weight to the pixel values around the edge region. This increases the edge intensity and it becomes enhanced comparatively to the original image. Following is the horizontal mask of Sobel operator:

$$\begin{array}{ccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array}$$

In this mask you will find edges in a horizontal direction because that zeros column is in a horizontal direction. When you convolve this mask onto an image horizontal edges will be prominent. The only difference is that it has 2 and -2 as center elements of the first and third rows. It also works on the principle of the above mask and calculates the difference among the pixel intensities of a particular edge. Because the center row of the mask consist of zeros, it does not include the original values of edge in the image, but rather it calculates the difference of above and below pixel intensities of the particular edge. This increases the sudden change of intensities and makes the edges more visible. Following is a Figure 5.19, on which we will apply the two masks shown above one at time.

In the image on the left, when a vertical mask is applied, all the vertical edges are more visible than the original image. Similarly in image on the right we have applied the horizontal mask and in result all the horizontal edges are visible. We can detect both horizontal and vertical edges from an image. Also if the result of the Sobel operator is compared to the Prewitt operator, the Sobel operator finds more edges or make edges more visible.

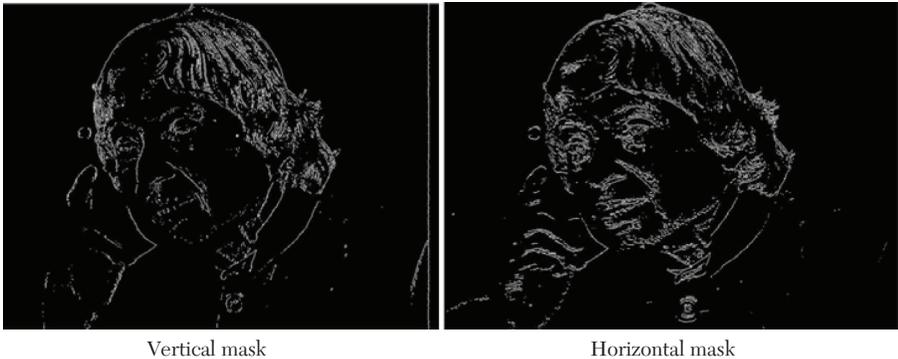


FIGURE 5.19. Sobel edge detection.

This is because The Sobel operator is allotted more weight to the pixel intensities around the edges. Now we can also see that if we apply more weight to the mask, the more edges it will get for us. Also there is no fixed coefficients in the Sobel operator, so here is another weighted operator

$$\begin{matrix} -1 & 0 & 1 \\ -5 & 0 & 5 \\ -1 & 0 & 1 \end{matrix}$$

If it is compared the result of this mask with of the Prewitt vertical mask, it is clear that this mask will give out more edges just because we have allotted more weight in the mask.

Robinson Compass Mask

Robinson compass masks are another type of derivate mask which is used for edge detection. This operator is also known as a direction mask. In this operator, take one mask and rotate it in all eight major compass directions: North, North West, West, South West, South, South East, East, and North East. There is no fixed mask. It can take any mask and rotate it to find edges in all the previously mentioned directions. All the masks are rotated on the bases of direction of zero columns. For example, a North direction mask is taken and then rotated to make all the direction masks.

North Direction Mask

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

North West Direction Mask

$$\begin{array}{ccc} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{array}$$

West Direction Mask

$$\begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array}$$

South West Direction Mask

$$\begin{array}{ccc} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{array}$$

South Direction Mask

$$\begin{array}{ccc} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{array}$$

South East Direction Mask

$$\begin{array}{ccc} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{array}$$

East Direction Mask

$$\begin{array}{ccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array}$$

North East Direction Mask

$$\begin{array}{ccc} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{array}$$

All the directions are covered on the basis of zeros direction. Each mask give the edges on its direction. Suppose we have a sample (Figure 5.20) from which we have to find all the edges. By applying all the above masks,

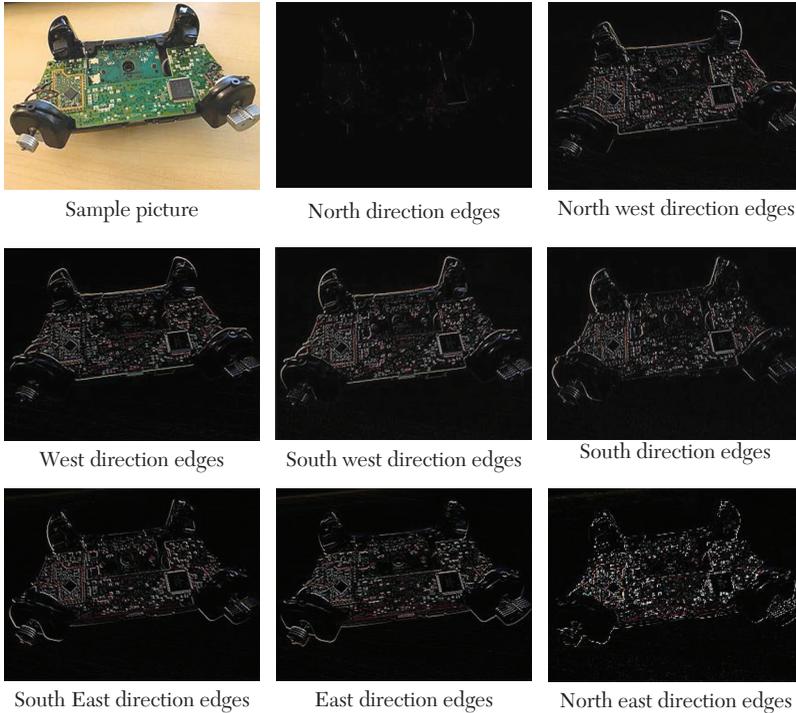


FIGURE 5.20. Robinson compass mask.

it gives edges in all the direction. Suppose there is an image that does not have any North East direction edges; that mask will be ineffective.

Kirsch Compass Mask

Kirsch compass mask is also a derivative mask used for finding edges. Similar to the Robinson compass it will find edges in all eight directions of a compass. Kirsch has a standard mask and will change the mask according to its own requirements. The difference between Robinson and Kirsch are that with the help of Kirsch compass masks edges can be found in the following eight directions: North, North West, West, South West, South, South East, East, and North East. Take a standard mask that follows all the properties of a derivative mask and then rotate it to find the edges. For example, let's look at the following mask, which is in North direction, and then rotate it to make all the direction masks.

North Direction Mask

$$\begin{array}{ccc} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{array}$$

North West Direction Mask

$$\begin{array}{ccc} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{array}$$

West Direction Mask

$$\begin{array}{ccc} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{array}$$

South West Direction Mask

$$\begin{array}{ccc} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{array}$$

South Direction Mask

$$\begin{array}{ccc} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{array}$$

South East Direction Mask

$$\begin{array}{ccc} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{array}$$

East Direction Mask

$$\begin{array}{ccc} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{array}$$

North East Direction Mask

$$\begin{matrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{matrix}$$

All the directions are covered and each mask will give the edges of its own direction. Let's find all the edges in Figure 5.21. All the above masks will give edges in all the direction. Suppose there is an image that does not have any North East direction edges; that mask will be ineffective.

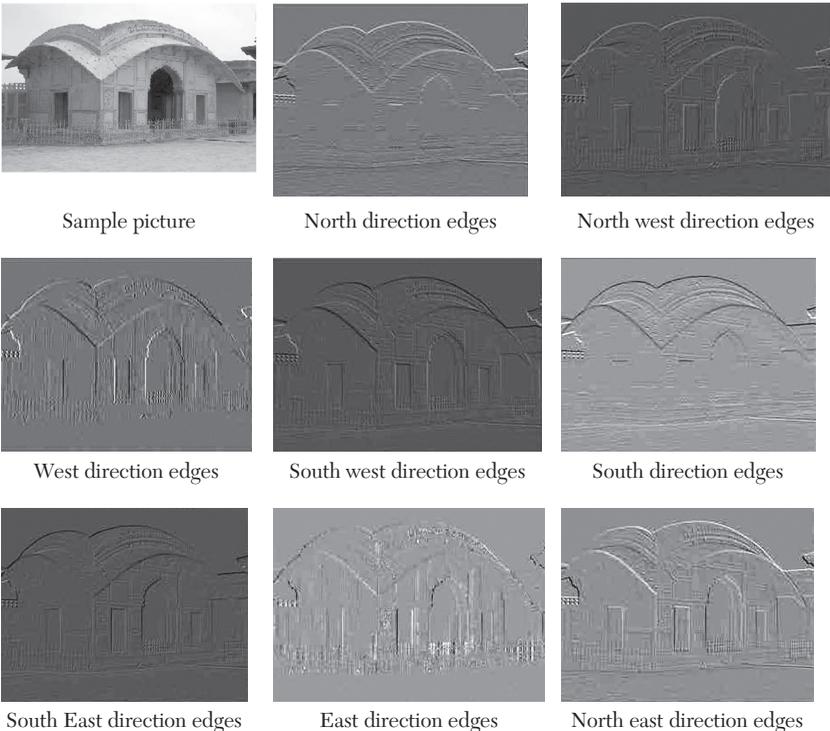


FIGURE 5.21. Kirsch compass mask.

Laplacian Operator

The Laplacian operator is also a derivative operator used to find edges in an image. The major difference between Laplacian and other operators like Prewitt, Sobel, Robinson, and Kirsch is that these are all first-order derivative masks, but the Laplacian is a second-order derivative mask. It has two further classifications: positive Laplacian operator and negative

Laplacian operator. Another difference between the Laplacian and other operators is that the Laplacian didn't take out edges in any particular direction, but rather it takes out edges in following classifications: inward edges and outward edges.

Positive Laplacian Operator

The positive Laplacian has a standard mask, in which the center element of the mask should be negative and corner elements of the mask should be zero. The positive Laplacian operator is used to take out outward edges in an image.

$$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array}$$

Negative Laplacian Operator

The negative Laplacian operator has a standard mask, in which the center element should be positive. All the elements in the corner should be zero, and the rest of all the elements in the mask should be -1. A negative Laplacian operator is used to take out inward edges in an image.

$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array}$$

Laplacian is a derivative operator that uses highlight grey-level discontinuities in an image and tries to deemphasize regions with slowly varying grey levels. This produces inward and outward edges in an image. It can't be applied to both the positive and negative Laplacian operators on the same image. If a positive Laplacian operator is applied on the image then subtract the resultant image from the original image to get the sharpened image. Similarly for the negative Laplacian operator applied on an image, we have to add the resultant image onto the original image to get the sharpened image. Let's apply these filters onto an ABJ Abdulkalam image (Figure 5.22) to give us inward and outward edges of an image.

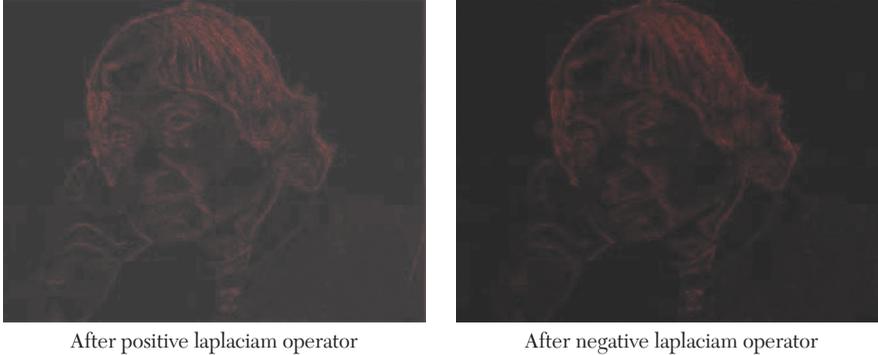


FIGURE 5.22. Laplacian operator.

5.3 ANALYZING AN IMAGE

Processing of signals in frequency domain, color models, and JPEG compress are analyzed.

Frequency domain

We are processing signals (images) in frequency domain. In spatial domain, we deal with images as in Figure 5.23. The value of the pixels of the image change with respect to scene. Whereas in frequency domain, we deal with the rate at which the pixel values are changing in spatial domain. In simple spatial domain, it directly deal with the image matrix.



FIGURE 5.23. Spatial domain.

We first transform the image to its frequency distribution. Then the black-box system performs whatever processing is necessary, and the output of the black box in this case is not an image, but a transformation. After performing inverse transformation, it is converted into an image which is then viewed in spatial domain. It can be pictorially viewed as in Figure 5.24.

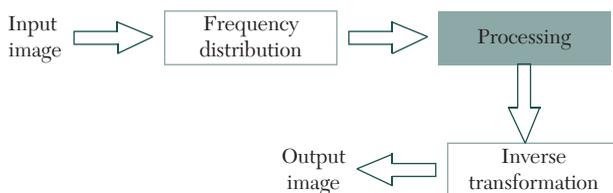


FIGURE 5.24. Frequency domain.

Transformation

A signal can be converted from time domain into frequency domain using mathematical operators called transforms. There are many kinds of transformation. Some of them are given here:

- Fourier series
- Fourier transformation
- Laplace transform
- Z transform

Any image in spatial domain can be represented in a frequency domain. We will divide frequency components into two major components. High-frequency components correspond to edges in an image and low frequency components in an image correspond to smooth regions.

Fourier series

Fourier was a mathematician in 1822. He developed Fourier series and the Fourier transform. They are used to convert a signal into frequency domain. A Fourier series simply states that periodic signals can be represented by the sum of sines and cosines when multiplied with a certain weight. It further states that periodic signals can be broken down into further signals with the following properties: the signals are sines and cosines and

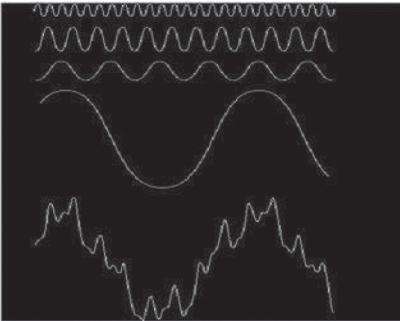


FIGURE 5.25. Fourier series.

the signals are harmonics of each other (See Figure 5.25.) In the above signal, the last signal is actually the sum of all the above signals. In order to process an image in frequency domain, we need to first convert it using into frequency domain and we have to take the inverse of the output to convert it back into spatial domain. That's why both Fourier series and Fourier transforms have two formulas. One for conversion and the other for converting it back to the spatial domain.

The Fourier series can be denoted by this formula:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

The inverse can be calculated by this formula:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$

Fourier Transform

The Fourier transform simply states that the nonperiodic signals whose area under the curve is finite can also be represented into integrals of the sine and cosines after being multiplied by a certain weight. The Fourier transform has many wide applications that include image compression (e.g., JPEG compression), filtering, and image analysis. Although both Fourier series and Fourier transforms were developed by Fourier, they differ in that a Fourier series is applied to periodic signals and a Fourier transform is applied for non-periodic signals. The Fourier term of a sinusoidal includes three things.

- Spatial frequency
- Magnitude
- Phase

The spatial frequency directly relates with the brightness of the image. The magnitude of the sinusoid directly relates with the contrast. Contrast is the difference between maximum and minimum pixel intensity. Phase contains the color information. The formula for 2-dimensional discrete Fourier transform is given below.

$$F(u, v) = \frac{1}{MN} \sum_{X=0}^{M-1} \sum_{Y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}$$

The discrete Fourier transform is actually the sampled Fourier transform, so it contains some samples that denote an image. In the above formula $f(x,y)$ denotes the image, and $F(u,v)$ denotes the discrete Fourier transform. The formula for 2-dimensional inverse discrete Fourier transform is given below.

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)}$$

The inverse discrete Fourier transform converts the Fourier transform back to the image. Now we will see an image of APJ Abdulkalam, and we will calculate FFT magnitude spectrum and then shifted FFT magnitude spectrum as shown in Figure 5.26.

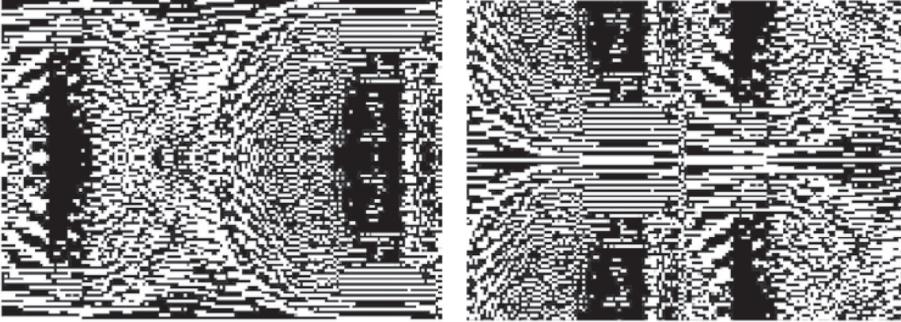


FIGURE 5.26. Fourier transform.

The relationship between the spatial domain and the frequency domain can be established by convolution theorem. The convolution theorem can be represented as.

$$f(x,y) * h(x,y) \longleftrightarrow F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \longleftrightarrow F(u,v) * H(u,v) \cdot$$

$$h(x,y) \longleftrightarrow H(u,v)$$

It can be stated the convolution in spatial domain is equal to filtering in frequency domain and vice versa. The filtering in frequency domain is represented in Figure 5.27.



FIGURE 5.27. Filtering frequency domain.

The steps in filtering are given below.

- In the first step we have to do some preprocessing to an image in spatial domain, means increase its contrast or brightness.
- Then we will take discrete Fourier transform of the image.
- Then we will center the discrete Fourier transform, as we will bring the discrete Fourier transform in center from corners.
- Then we will apply filtering, means we will multiply the Fourier transform by a filter function.
- Then we will again shift the DFT from center to the corners.
- The last step would be taken to inverse discrete Fourier transform, to bring the result back from frequency domain to spatial domain.

- And this step of postprocessing is optional, just like preprocessing, in which we just increase the appearance of image.

Filters

The concept of filter in frequency domain is the same as the concept of a mask in convolution. After converting an image to frequency domain, some filters are applied in filtering process to perform different kind of processing on an image. The processing includes blurring an image, sharpening an image, and so on. The common type of filters for these purposes are:

- Ideal high-pass filter
- Ideal low-pass filter
- Gaussian high-pass filter
- Gaussian low-pass filter

Blurring masks:

A blurring mask has the following properties:

- All the values in blurring masks are positive.
- The sum of all the values is equal to 1.
- The edge content is reduced by using a blurring mask.
- As the size of the mask grows, more smoothing effect will take place.

Derivative masks:

A derivative mask has the following properties:

- A derivative mask have positive, as well as negative values.
- The sum of all the values in a derivative mask is equal to zero.
- The edge content is increased by a derivative mask.
- As the size of the mask grows, more edge content is increased.

The relationship between a blurring mask and a derivative mask with a high-pass filter and low-pass filter can be defined simply as:

- Blurring masks are also called as low-pass filter.
- Derivative masks are also called as high-pass filter.

The high-pass frequency components denotes edges whereas the low pass frequency components denotes smooth regions. The common example of low pass filter and its graphical representation is as shown in Figure 5.28.

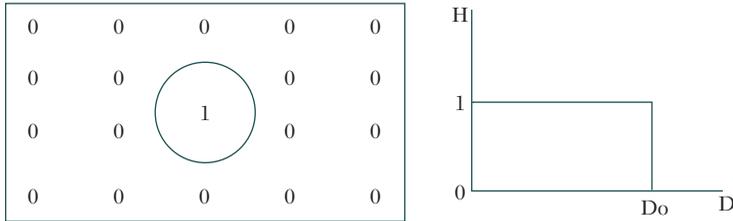


FIGURE 5.28. (a) Low pass filter and (b) graphical representation.

When 1 is placed inside and the 0 is placed outside, we get a blurred image. Now as we increase the size of 1, blurring would be increased and the edge content would be reduced. This is a common example of high-pass filter (Figure 5.29). When 0 is placed inside, we get edges, which gives us a sketched image.

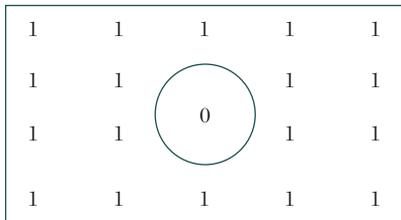


FIGURE 5.29. High-pass filter.

Now let's apply this filter to an actual image and results are shown in Figure 5.30.

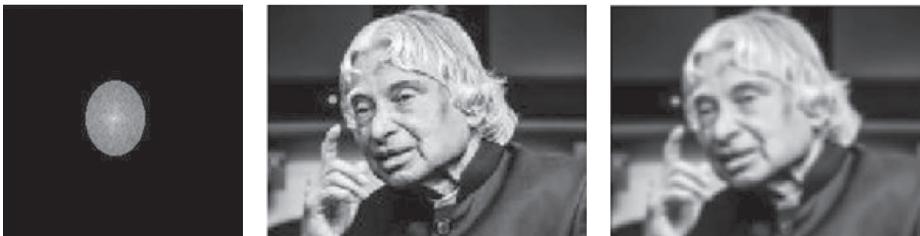


FIGURE 5.30. Filter over the image, input, and output images

With the same way, an ideal high-pass filter can be applied on an image. But obviously the results would be different as, the low pass reduces the edged content and the high pass increases it.

Gaussian low-pass and Gaussian high-pass filter

Gaussian low-pass and Gaussian high-pass filter minimizes the problems that occur in ideal low-pass and high-pass filter. This problem is known as ringing effect. This is because at some points transition between one color to the other cannot be defined precisely, and the ringing effect appears at that point. Have a look at the graph 5.30(b) which is the representation of ideal low-pass filter. Now at the exact point of D_0 , you cannot tell that the value would be 0 or 1. The ringing effect appears at that point. So in order to reduce the effect that appears as the ideal low pass and the ideal high-pass filters, the following Gaussian low pass filter and Gaussian high-pass filter is introduced. The concept of filtering and low pass remains the same, but only the transition becomes different and become smoother. The Gaussian low pass filter can be represented as in Figure 5.31. Note the smooth curve transition, due to which at each point, the value of D_0 , can be exactly defined.

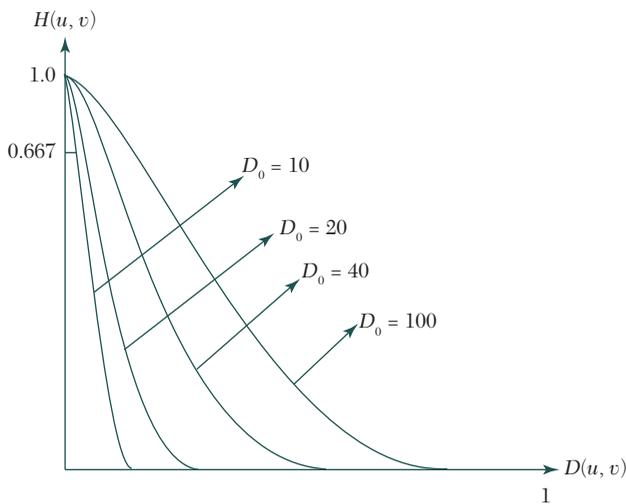


FIGURE 5.31 Gaussian low-pass filter.

Gaussian high-pass filter has the same concept as ideal high-pass filter, but again the transition is smoother as compared to the ideal one.

The purpose of smoothing is to reduce noise and improve the visual quality of the image. A variety of algorithms, that is, linear and nonlinear algorithms are used for filtering the images. *A filter can be applied to reduce the amount of unwanted noise in a particular image.* Another type of filter can be used to reverse the effects of blurring on a particular picture. Nonlinear filters display quite different behavior compared to linear filters. For nonlinear filters, the filter output or response of the filter does not obey the principles outlined earlier, particularly scaling and shift invariance. Moreover, a nonlinear filter can produce results that vary in a nonintuitive manner.

Linear smoothing

The most common, simplest, and fastest kind of filtering is achieved by linear filters. The linear filter replaces each pixel with a linear combination of its neighbors, and convolution kernel is used in prescription for the linear combination. Linear filtering of a signal can be expressed as the convolution.

1. Box blur

A box blur, also known as “moving average,” is a simple linear filter with a square kernel and it contains all the kernel coefficients equal. It is the quickest blur algorithm, but it has a drawback, that is, it lacks smoothness of a Gaussian blur. A box blur can be with a complexity independent of a filter radius. The algorithm is based on a fact that sum of elements in the rectangular window can be decomposed into sums of columns of this window. Then multiply complex spectra and do the inverse transform.

2. Hann window

Hann window is a smooth function and is based on modulation of the input signal with a complex exponent.

3. Gaussian Blur

Gaussian blur is considered a “perfect” blur for many applications, provided that kernel support is large enough to fit the essential part of the Gaussian.

Nonlinear Smoothing

1. Median filtering

In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. *The median filter is a nonlinear digital filtering technique, often used to remove noise.* Such noise reduction

is a typical preprocessing step to improve the results of later processing (for example, edge on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges of the images while removing noise. Median is a nonlinear local filter whose output value is the middle element of a sorted array of pixel values from the filter window. Since median value is robust to outliers, the filter is used for reducing the impulse noise. Since large computational time and effort is spent on calculating the median of any window. Because filter consider every entry in the signal and then median of that values is calculated. Some types of signal contain the whole number representation. In that case the images can be easily described by histograms and median can be easily calculated in that case.

2. Binary morphological operations

A basic morphological operation is dilation. When a structuring element is defined inside a square window with a radius r , the dilation operation sets to 1 all the pixels from which the structuring element overlaps at least one nonzero pixel of the source image. When a structuring element window shifts one pixel to the right, some image pixels that can become overlapped are shifting in from the right border of a structuring element, and some image pixels can be shifting out of overlapping area through the left border of a structuring element. So, instead of counting a total number of overlapping pixels, we can increment the previous count by a number of pixels covered by the right border of the structuring element and decrement by the number of pixels that are lying to the left of the left border of a structuring element.

3. Min/Max filters

A max filter outputs a maximal pixel value from its rectangular window. In the case of small data bit depth, a histogram approach can be used. But when the bit depth is large, another approach based on a 1D running max filter appears more practical. A simple and fast algorithm called MAXLINE is using a circular buffer of delayed input elements. The anchor points to the current maximal value. When the window is shifted, a new element is added to the delay line and compares against the anchor element. If the new element is smaller, the maximum stays at the anchor. Otherwise anchor moves to a new element. When the anchor shifts out of the delay line, the whole delay line is scanned for a new anchor. This algorithm works very fast on IID (independent identically distributed) data, but has a worst-case complexity for a monotonically decreasing data.

4. Greyscale morphological operations:

Greyscale morphology is simply a generalization from 1 bpp (bits per pixel) images to images with multiple bits/pixel, where the max and min operations are used in place of the OR AND operations, respectively, of binary morphology. Greyscale morphological operations are based on min/max filters. When structuring element is rectangular, they can be optimized by using min/max filter.

There are two types of filters that have been found useful in nuclear medicine. They are spatial filter and temporal filter.

Spatial filters are applied to both static and dynamic images, whereas temporal filters are applied only to dynamic images. Spatial filter techniques are useful when analyzing regional unemployment data, particularly, when the final aim is to develop forecasting models for some regional scale. Among conventional spatial econometric methods, spatial auto regression is a powerful method commonly employed. Spatial autoregressive techniques take into account spatial effects by means of geographic weights matrices that provide measures of the spatial linkages (dependence) between values of geo referenced variables.

Temporal filtering allows reducing signals that are not correlated from frame to frame. It can very effectively reduce noise when combined with motion compensation, as motion compensation correlates the image content from frame to frame. This makes this processing suitable to improve the efficiency of subsequent encoders. It is implemented using a recursive filter since it provides a better selectivity at lower costs. The overall goal of temporal filtering is to increase the signal-to-noise ratio. Due to the relatively poor temporal resolution of MRI (Functional magnetic resonance imaging), time series data contain little high-frequency noise. They do, however, often contain very slow frequency fluctuations that may be unrelated to the signal of interest. Slow changes in magnetic field strength may be responsible for part of the low-frequency signal observed in fMRI time series.

Color Spaces

Color spaces are different types of color modes, used in image processing and signals and system for various purposes. Some of the common color spaces are: RGB, CMYK, YUV, YIQ, YCbCr, and HSV.

RGB

RGB is the most widely used color space. RGB stands for red, green, and blue. The RGB model states that each color image is actually formed of three different images. Red image, blue image, and black image. A normal greyscale image can be defined by only one matrix, but a color image is actually composed of three different matrices. One color image matrix = red matrix + blue matrix + green matrix, as shown in Figure 5.32.

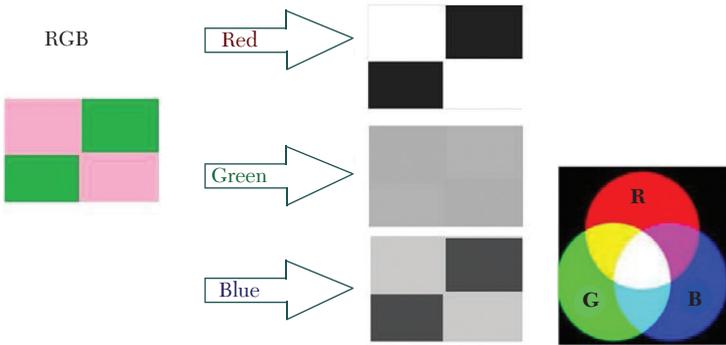


FIGURE 5.32. RGB.

CMYK

The conversion from RGB to CMY is done using this method.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Consider a color image means that it have three different arrays of RED, GREEN, and BLUE. Now to convert it into CMY, subtract it by the maximum number of levels – 1. Each matrix is subtracted and its respective CMY matrix is filled with result.

Y'UV

Y'UV defines a color space in terms of one luma (Y') and two chrominance (UV) components as in Figure 5.33 (a).

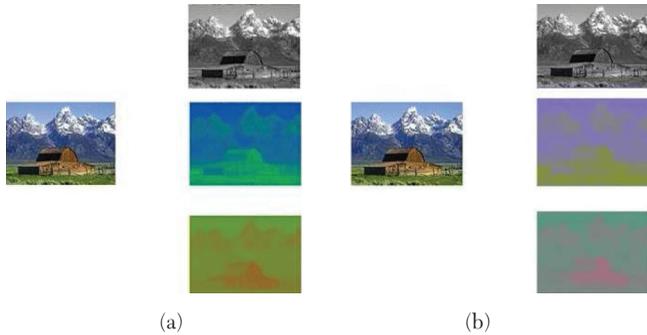


FIGURE 5.33. (a) $Y'UV$ (b) $Y'CbCr$

$Y'CbCr$

$Y'CbCr$ color model contains Y' , the luma component and cb and cr are the blue-difference and red difference chroma components as shown in Figure 5.33 (b). It is not an absolute color space. It is mainly used for digital systems. Its common applications include JPEG and MPEG compression. $Y'UV$ is often used as the term for $Y'CbCr$, however they are totally different formats. The main difference between these two is that the former is analog, while the latter is digital.

JPEG Compression

Image compression is the method of data compression on digital images. The main objective in the image compression is to store data in an efficient form and transmit data in an efficient form. Image compression can be lossy or lossless. JPEG stands for joint photographic experts group.

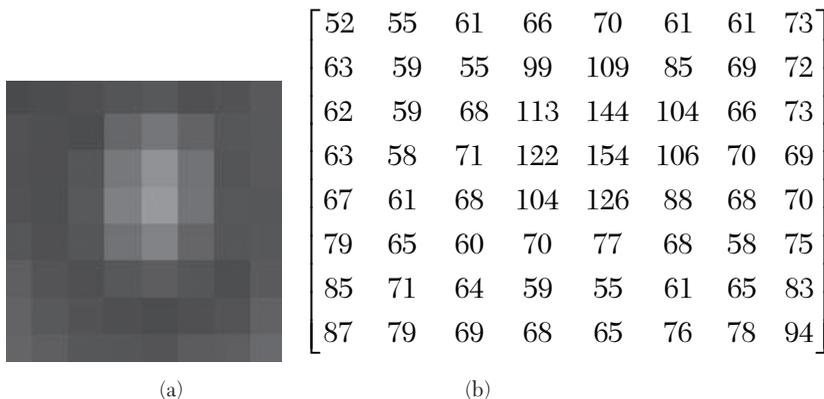


FIGURE 5.34. (a) Image into 8×8 blocks and (b) values of 8×8 image.

It is the first international standard in image compression. The first step is to divide an image into blocks with each having dimensions of 8 x 8 as in Figure 5.34(a). This 8 x 8 image contains the following values in Figure 5.34(b) as assumed.

The range of the pixels intensities now are from 0 to 255. Change the range from -128 to 127. Subtracting 128 from each pixel value yields pixel value from -128 to 127. After subtracting 128 from each of the pixel value, you get the following result.

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -61 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -19 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -13 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Now compute using the below formula.

$$G_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{\pi}{8} \left(x + \frac{1}{2} \right) u \right] \cos \left[\frac{\pi}{8} \left(y + \frac{1}{2} \right) v \right]$$

$$\alpha_p(n) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{if } n = 0 \\ \sqrt{\frac{2}{8}}, & \text{otherwise} \end{cases}$$

The result comes from this is stored in let's say A(j,k) matrix. There is a standard matrix that is used for computing JPEG compression, which is given by a matrix called as Luminance matrix. This matrix is given below.

$$Q_{j,k} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 20 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Applying the following formula:

$$B_{j,k} = \text{round} \left(\frac{A_{j,k}}{Q_{j,k}} \right)$$

The result after applying is given below:

$$B_{j,k} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now perform zig-zag until you find all zeroes ahead. Hence the image is now compressed. The steps in the JPEG compression are given as follows: The first step is to convert an image to YCbCr and just pick the Y channel and break into 8 x 8 blocks. Then starting from the first block, map the range from -128 to 127. After that you have to find the discrete Fourier transform of the matrix. The result of this should be quantized. The last step is to apply encoding in the zig-zag manner and do it till you find all zero. Save this one-dimensional array and it is a JPEG compression of image.

Pattern Matching

Pattern matching is a technique used to locate specified patterns within an image. It uses computer vision as a service at the backend. It can be used to determine the existence of specified characteristics within a captured image, for example, the expected label on a defective product in a factory line or the specified dimensions of a component. It is different from the “pattern recognition” that is recognized on the basis of general patterns and larger collections of related samples. It specifically dictates what we are looking for, then tells us whether the expected pattern exists or not.

Pattern matching quickly locates regions of greyscale images that match a known reference pattern, also referred to as a model or a template. This could be implemented in a total of two stages. Learning becomes the first stage while matching becomes the second. Greyscale value is extracted from the template image provided by the user. This is done in the learning

stage. The algorithm organizes and stores the information in a manner that facilitates faster searching in the inspection image.

The image to be inspected is processed in the matching stage of pattern matching. Now, the algorithm extracts the grey value from the inspection image. This corresponds to the information learned from the template. Then, the algorithm finds matches by locating regions in the inspection image where the highest cross-correlation is observed.

There are various algorithms implemented to perform pattern matching. Some of them include normalized cross correlation, pyramidal matching, grey-value method, gradient method. The images where pattern matching shows erroneous results prove to be defective (pattern-wise alone).

Pattern recognition primarily cares about the representation. It faces the challenge of dealing with images of different sizes, orientations, and illumination conditions, or with time signals of arbitrary length and varying offset. Pattern recognition includes preprocessing procedures to normalize observations, to deal with invariants, and to define proper features and distance measures. Once a proper representation is found, learning procedures become of interest and the results of machine learning can be applied.

In many machine vision systems, it is necessary to locate objects or features of objects as rapidly as possible so that further image-processing algorithms can extract additional features. For example, finding the correct orientation of a part within 2D or 3D space can speed up robotic-based pick-and-place applications. In food and beverage applications, pattern-matching techniques allow for the reading and examination of specific characters or patterns, reducing the processing power needed to extract further data from an image.

There are two main approaches to pattern matching: those based on correlation, and geometric pattern matching. While they are technically dissimilar, both approaches rely on first locating a region or regions of a template image to provide reference data. Once extracted, this data is compared with a newly captured unknown image to find matching characteristics. Establishing a correspondence between the reference template image and the newly captured image allows for the location of objects in the newly captured image.

Cross-correlation

Historically, cross-correlation (CC) was one of the first statistical approaches used for pattern matching. In this rather brute-force approach, a simple sum of pairwise multiplications of corresponding pixel values of the template image and regions of the same size in the captured image is computed, yielding a similarity value between the images. However, because this value is subject to changes in reflectivity or illumination in the captured image, the approach has been replaced by normalized grey-scale correlation (NGC), in which the correlation value is invariant to global brightness changes.

While traditional CC and NGC are not invariant to large degrees of rotation, translation, and scale, such methods can be improved by rotating, translating, and scaling the template image and then using the template image to perform pattern matching.

To reduce this computational overhead, a pyramid-based approach can be used. In this method, both the template and the captured image are subsampled a number of times, in effect building two pyramids, with increasingly lower resolutions as levels increase. Correlation is performed at the top level of the pyramid and used as an initial estimate for a possible correlation match at the next level. This process is repeated at different levels and areas of successively increasing resolution of the pyramid until a suitable correlation coefficient is determined. This method can be used with both correlation-based and geometric-based pattern matching methods (Figure 5.35).

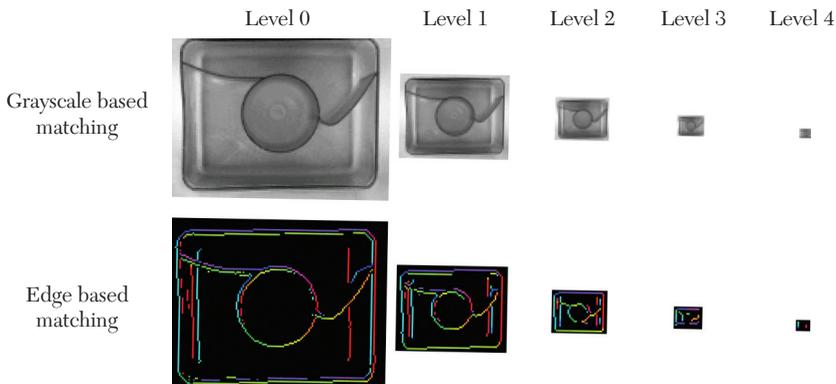


FIGURE 5.35. To reduce the computational overhead in correlation and geometric pattern matching, a pyramid-based approach can be used. In this method, both the template and the captured image are subsampled a number of times, in effect building two pyramids, with increasingly lower resolutions at higher levels.

Geometric pattern matching

While standard correlation methods have limitations in terms of rotational, translational, and scale invariance, they are also limited if the part being inspected is somewhat occluded. To overcome this problem, geometric pattern matching techniques can be used to extract geometric features, such as shapes, dimensions, angles, and arcs, within a template image. Then their spatial relationships are used to find correspondences within a captured image.

Geometric matching locates regions in a greyscale image that match a model, or template, of a reference pattern. Geometric matching is specialized to locate templates that are characterized by distinct geometric or shape information. Region of interest is an area of image in which you want to focus your image analysis. Geometric matching quickly locates known reference or fiducial patterns in an image using edge information. The presence or absence, number of and location of a specified template model could be identified in the image to be inspected using the geometric pattern matching.

Template Matching

Template is primarily a subpart of an object that is to be matched among entirely different objects. The techniques of template matching are flexible and generally easy to make use of, that makes it one among the most famous strategies of object localization. Template matching may be a high-level machine vision method which determines the components of a Figure which matches a predefined template. This technique is repeated for the whole image, and the point which leads to a best match, the utmost count, is defined to be the point wherever the shape (given by the template) lies inside the image. Templates are usually employed to print characters, identify numbers, and other little, simple objects. It can be used for detection of edges in Figures, in manufacturing as a part of quality control and a means to navigate a mobile robot. *Detecting eye region in human face is example for template matching.*

Template matching is a strategy for discovering zones of an image that match (are indistinguishable) a template image (patch). We require two crucial segments. Source image (I): The picture inside which we are hoping to find out a match to the template image. Template image (T): The patch image that can be compared to the template image, and our objective is to discover the most effective technique for the best matching region.

Template Matching Approaches

General categorizations of template or image matching approaches are featured-based approach and template- or area-based approach.

Featured-based approach

The featured-based method is appropriate while both reference and template images have more connection with regards to features and control points. The features comprises of points, a surface model that needs to be matched, and curves. At this point, the goal is to position the pairwise association among reference and layout picture using their spatial relations or descriptors of components.

Area-based approach

The area-based methods are typically referred to as correlation like methods or template-matching methods, that is the blend of feature matching, feature detection, motion tracking, occlusion handling, and so on. Area-based methods merge the matching part with the feature detection step. These techniques manage the pictures without attempting to identify the remarkable article. Windows of predefined size are used for the estimation of correspondence.

Template-based approach

Template-based template-matching approach could probably require sampling of a huge quantity of points, it is possible to cut back the amount of sampling points via diminishing the resolution of the search and template images via the same factor and performs operation on resulting downsized images (multiresolution, or pyramid (image processing), providing a search window of data points inside the search image so that the template doesn't have to be compelled to look for each possible data point and the mixture of the two.

Motion Tracking and Occlusion Handling

For the templates that can't provide and may not provide an instantaneous match, Eigen spaces may be used to provide the details of matching the image beneath numerous conditions, appropriate matching poses, or color contrast. For instance, if the person turned into searching out a specimen, the Eigen spaces may include templates of a specimen totally different in numerous positions such as a camera with different lighting conditions or expressions. There is feasibility for the matching figure

to be occluded via an associated item or issues involved in movement turn out to be ambiguous. One probable answer for that can be to separate the template into more than one subimage and carry out matching on them. The steps of template matching are described as follows.

1. First we select the original image. The image will be in file formats such as JPG/JPEG, PNG, etc.
2. Convert to binary image. The process to convert the color image into white and black image is called binary image. This method is based totally on numerous color transforms. It analyzes the values of greyscale and also achieves the grey image according to the R, G, B value within the image. The technique of template matching can be easily carried out on edge figures or grey figures.
3. Using the template image. Template image may be a small portion of an input image and is used to find the template within the given search image.
4. Apply template matching techniques like normalized cross-correlation, cross-correlation, sum of squared difference.
5. Then match the images with the original image.
6. Display the result.

Template-Matching Techniques

The template matching techniques are described as follows:

1. Naive Template Matching

Assume that you are provided a picture of a plug and our goal is to search out its pins. We are supplied with the pattern image that represents the reference object we are looking for. Therefore the input image to be used for that position is the pattern above the image at each attainable location. Every instant we tend to calculate some numeric measure of similarity amid the pattern and therefore the image section it presently overlaps. At last we tend to determine the locations that give the most effective similarity measures as the feasible pattern positions.

2. Image Correlation Matching

The problem that occurs in naive template matching is in computing the similarity measure of the aligned pattern image, and therefore the

overlapped section of the input image, that's equal to computing a similarity measure of two figures of same dimensions. So the numeric measure of image similarity is known as image-correlation

Cross-Correlation

$$\text{Cross-correlation (Image1, Image2)} = \sum_{u,v} \text{Image1}(u,v) \times \text{Image2}(u,v)$$

For example, take two images, Image1 and Image2 and their pixel coordinates u and v . The fundamental strategy of computing the image correlation is so referred to as cross-correlation, basically a simple sum of pair wise-multiplications of corresponding pixel values of the images.

Normalized cross-correlation

Normalized cross-correlation is an improved model of the traditional cross-correlation methodology which bring in two improvements over the primary one:

- The outcomes are constant to the global brightness changes that is darkening of whichever figure or consistent brightening have no impact on end result. (This can be achieved via subtracting the mean image brightness from every pixel value.)

The final correlation value is scaled to $[-1, 1]$ range, in order that NCC of two alike figures is equal to 1.0, though NCC of a figure plus its negation is equal to -1.0. Normalized cross-correlation can be generally applied as an efficient resemblance measure meant for matching applications. However, conventional correlation-based image-matching techniques may not succeed while there are significant scale changes or large rotations among the two figures. That's the reason why normalized cross-correlation is sensitive to revolution in addition to scale changes.

NCC technique is used in face-recognition. Normalized cross-correlation (NCC) is the technique that is employed in image registration for matching the template with an image. However, NCC was influenced via factors such as illumination and clutter background issues. In case of NCC, there's a big increase within the inaccuracy rate because of the shaded input figure. The values of the pixels of the shaded portions are less than the portions of an image and this encompasses a high proportion of NCC among template image and input image that can conclude an incorrect location.

3. *Sum of Absolute Difference*

Sum of absolute difference (SAD) can be computed via obtaining the absolute difference among every pixel within the source image (original block) and therefore the subsequent pixel within the block that's meant for the aim of comparison. It could be used as a measure to determine the similarity among image blocks. It is used in many fields such as object recognition, estimation of movement for video compression and generates dissimilarity maps designed for stereo imagery.

The SAD is used to measure the similarities among template images T and subimages within the source image S . It works via computing the absolute difference among every pixel in template image T and as well as subsequent pixel within the subimages which is intended for comparison in the source image S . Then we find the summation of all the differences obtained to produce a straightforward metric of similarity.

However, due to the illumination and clutter background issues, NCC is affected for the reason that at times there are non-face blocks that had nearly identical value as that of average face template matrix. This drawback is resolved by the use of SAD algorithm for image compressing and object tracking. However, for giving more accurate positions of face within the input image SAD needs more optimization. Moreover, SAD will provide high localization rate for facial image wherever the image is with high-illumination variation, however, variation in background may affect it.

4. *Sum of squared differences*

Sum of squared differences (SSD) is procedure that is employed in image registration for matching the template with an image. Moreover, it also tests the effect of template image on output image when there is noise and rotation. The measure of variation or deviation from the mean is represented by sum of squares. It is calculated as a summation of the squares of the variations from the mean. The performance of this method is done by making the comparison based on the value of correlation coefficient and that produced from different template images.

Advanced Methods

- There are greyscale-based matching and edge-based matching of two advanced template-matching methods.

Greyscale-based matching

Greyscale-based matching is an enhanced template-matching method, which is an extension of correlation-based pattern detection. It enhances its efficiency and allows us to search for pattern occurrences not considering its orientation. It is done by not computing the pattern image pyramid for every attainable rotation of the pattern. There are many applications of greyscale matching such as computer animation, human computer interaction, and virtual reality to human-motion analysis. This can be done via calculating not just single pattern image pyramid. This algorithm makes out the pairs (pattern position, pattern orientation) throughout the pyramid search of the input image instead of sole pattern position. The approach of pyramid matching used together by means of multi-angle search is called greyscale-based t-matching method.

Edge-based matching

Edge-based matching improves this methodology by off-putting the computation to the object edge-areas. Edge-based matching upgrades the formerly explained greyscale-based matching via the use of vital statement: that the form of any object is outlined only by the form of its edges. Thus rather than matching the entire pattern, we tend to take out its edges and then match solely the close by pixels, therefore keeping away from some superfluous computations. Normally in these applications the acquired speed-up is typically important.

Application areas of template matching

Following are the application areas of template matching:

1. Object Recognition using Template Matching

Object recognition is job of discovering a known item inside an image or video sequence. It's (object recognition) used to properly determine objects in a scene and estimate their cause (location and orientation). The purpose is to understand the capability of existing object recognition methods to search out alike objects once input is completely of image type. We would like to rearrange these objects that are visible to us. These objects are totally visible or partly hidden behind another object. Similar objects might also be available in the various pose. The identification of those objects is easy for human being as he can easily identify any object-based on his knowledge or expertise, yet it is much harder to distinguish a specific item for a machine. The machine has to learn how to recognize any object. For this issue, certain algorithms are proposed. With the assistance

of those algorithms, a machine will understand objects present in the various pose, lightning conditions, camera parameters, appearance, and so on. For instance, the writing style of various individuals is totally different. Two person can compose one letter with varied designs.

2. Biological area

Biological area is used in biological science such as nuclear agriculture and molecular biology. It involves applications that involve the use of camera-based hardware systems or colored scanners for inputting pictures. The software package that has been designed for such purpose is the BIAS software that supports DOS and Windows-friendly Color-Pro software which is developed in Electronics Systems Division and Comprehensive Image method. It has the following features like color-image analysis for evaluation of leaf, chlorophyll, and defected leaf area. For plant-breeding, estimation of leaves area was extremely necessary. Years ago, leaf-area meters were used for this function. However, nowadays image analysis is used for measurement of leaf area. The image of leaf is initially taken via camera or a scanner and so analyzed via the Color Pro software package. A range of color plates, as well as chlorophyll meters was earlier employed to examine chlorophyll substance of leaf inside situ.

3. Eye detection in a facial image

- In this technique, we are provided with an eye template and a face image. Then we find the correlation of an eye template through the overlapping areas of the face image, the section that offers the maximum correlation coefficient with the eye template is referred to as eye region, this is how eye image is found.

The Algorithm for eye detection in face image

The methodology of template matching is explained with the help an algorithm that is straightforward and simple to execute. The rules of an algorithm are described as follows:

- Suppose we take an eye template that has size $a \times b$.
- The normalized two-dimensional auto-correlation of an eye template is determined.
- The normalized two-dimensional cross-correlation of an eye template with numerous overlapping sections of a face image are found out.

- The meansquared error (MSE) of auto correlation as well as cross-correlation of various areas are found out. The value that gives the minimum MSE is determined as well as stored.
- The sections of a face that has lowest MSE signify eye region. Matching method not solely obtains similarity measure, it also computes the inaccuracy among images reckoning on its difference by means of mean-squared error.

4. Remote Sensing

- Remote sensing may be applied at precise wavelengths at the same time giving thousands of digital images. Its knowledge can be gathered from hyperspectral devices that contain not solely the visible spectrum, but also ultraviolet and infrared ranges. It's general to list the hyperspectral data in a 3-D array or "cube," with the first 2-D matching to geographical dimensions and therefore the third one similar to the spectrum. During hyperspectral categorization and particularly target detection, the most important purpose was to seek out spatial pixels in 3-D hyperspectral cube data for a few best known spectral signals of interest. Though, it becomes complicated, since there is variability and uncertainty of every material's spectral signature. These difficulties comprise of noise from atmospherical conditions, illumination, location, and sensor control, all of which rely on when and where the image was taken.

Limitation

Following are the limitations of template matching:

- Templates are not rotation or scale invariant.
- Slight change in size or orientation variations can cause problems.
- It often uses several templates to represent one object.
- Templates may be of different sizes.
- Rotations of the same template.
- If you search the entire image or use several templates, template matching is a very expensive operation.
- Template matching is easily "parallelized."
- Template matching requires high computational power because the detection of large patterns of an image is very time-consuming.

Advantage

Estimates are quite good with enough data. Template matching is the most efficient technique to be used in pattern recognition machines that read numbers and letters and are available in standardized, constrained contexts (i.e., scanners that read your financial credit number from machines or checks that read postal zip codes from envelopes).

Enhancing the Accuracy of Template Matching

Advancements may be made to the matching methodology via the use of multiple templates. The additional templates that we take may have rotations and dissimilar scales. It's also achievable to boost the accuracy of the matching technique via combining feature-based with template-based methodology. It would be necessary for the search and template images to have characteristics understandable enough to support feature matching.

5.4 IMAGE-PROCESSING STEPS FOR VISION SYSTEM

Embedded vision is one of the most advanced areas of computer technology, electronics, and software development. Image processing is used for object recognition from the captured image for processing video or photos.

According to the complexity, the image processing can be divided into two levels according to the level of processing. The lower level of image processing is not used in the semantics of objects, that is, images are not interpreted. It uses the method for signal processing, for example, 2D Fourier transform. The aim of lower level is to analyze two-dimensional data input numeric character, remove noise from an image, recognize simple objects in an image, and find the necessary information for a higher level. Higher level is already perceived as an understanding of the image content. It is much more complicated and is based on knowledge-based systems and artificial intelligence techniques. Basic steps for image processing for vision system can be characterized in Figure 5.36.

Scanning and Image Digitalization

The basis for image processing is obtaining an image of the real world. The image is converted into digital format suitable for storage. The image is further processed by a computer or other computing system. Scanning of image is the transfer of optical quantities into electric signal which is

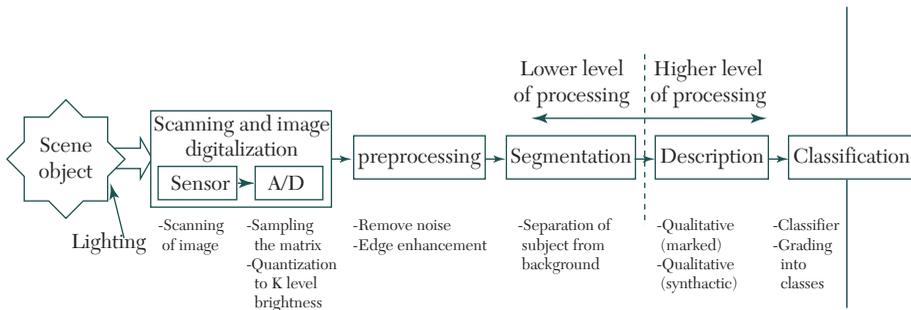


FIGURE 5.36. Basic steps for image processing.

continuous in time. The image device or sensor is used to obtain image. Image device is based on CCD sensors (charge coupled device) or CMOS (complementary metal oxide semiconductor). The images are two-dimensional, three-dimensional, or a sequence of images (video) according to sensor type and number of sensors. At first photons of scattered light are captured from the lens to the image sensor and transferred according to their intensity (i.e., their wavelength) on an electric charge. The intensity of electric current is manifested like brighter and darker places in the image. For further image processing the analog signal is converted to a digital signal in the form of 0, 1, and stored into memory. The sensors are usually nightblind and therefore for image storage RGB color model with 8 bits per color is used. Using the RGB model, great range of colors (256 colors) is obtained.

Image Preprocessing

The aim of preprocessing is to suppress the noise generated during digitization and image transmission. Preprocessing is done to remove distortions that arise while shooting. The basic methods of image preprocessing are greyscale conversion, the brightness and contrast adjustment, filtering, suppressing the influence of light, sharpening, and so on.

Image Segmentation on Object

Image segmentation refers to the process of dividing a digital image into multiple segments. The aim is to simplify the information from each pixel into something that will be more meaningful for analysis and further processing. Segmentation is usually used to find objects in the image and their boundaries (lines, curves). This means that merge groups of pixels into super pixels, and gives us a closer information about the pixels in

groups. The segments are created from information as color, intensity, or texture. The result of segmentation is a set of segments that together cover the entire image. Segmentation may use several methods. They are (1) segmentation by thresholding (color, brightness), (2) segmentation based on edge detection, (3) segmentation by accretion of areas (merge areas, cleavage of areas), and (4) segmentation by comparing with the pattern.

Description of Objects

Description of segmented objects from image preprocessing is the ultimate link in a chain of image processing. It is used to obtain marks from segmented data. Marks serve to classify objects and must therefore depict exactly the characteristics of objects. There are two methods of describing. One is called quantitative approach or marked. It means that the description of objects done by using a numerical characteristics. For example, these can be size of the object, the compactness, and so on. The second method is a qualitative approach or syntactic. In this method segmented data describe the relationship between the objects and their shape properties. Based on requirements of the projects, description method is selected. In most cases description is the input information for recognition or classification of objects. Choosing description depends on recognition algorithm used.

Classification of Objects

The final step in image processing is the classification of objects. That means recognition and image understanding. The aim of classification is to understand the semantics of the image. It is based on common marks of individual objects. The process consist of sorting objects into predetermined classes of objects. The class is understood as subset elements whose attributes have classification under the common features. Classification allows classifier, which decides sorting of objects into the class.

Summary

- Image sharpening and restoration refers to process image to achieve desired result.
- Histogram is a graph of frequency of anything used for brightness and contrast analysis.
- Image transformation maps one set to another set such as linear, logarithmic, and power law.

- Convolution is masking of image for blurring, sharpening, edge detection, and noise reduction.
- Mean, weighted average and Gaussian filter are used for blur.
- Box blur, Hann window, and Gaussian blur are linear smoothing.
- Median filtering, binary morphological operations, min/max filters, grayscale morphological operations are nonlinear smoothing.
- Spatial filter and temporal filter are useful in nuclear medicine.
- HSV, RGB, CMY are a few color models.
- Pattern matching locates regions of images that match a known reference pattern, also referred to as a model quickly through learning stage and matching stage.
- Normalized cross-correlation, pyramidal matching, grey-value method, and gradient method are examples of pattern-matching algorithms.
- Template matching is a high-level machine vision method which determines the components of a figure which matches a predefined template.
- Scanning and digitalization, preprocessing, segmentation, description, and classification are the steps in image processing.

References

<https://www.tutorialspoint.com/>

<https://www.visiononline.org/Pattern-Matching-Object-Location-Reduces-Image-Processing>

Learning Outcomes

- 5.1 Define image.
- 5.2 Differentiate analog and digital image signals.
- 5.3 List some applications of image processing.
- 5.4 What is histogram and its application?
- 5.5 What is transformation and write their types?
- 5.6 What is convolution and what are its uses in image processing?

- 5.7 Write about edge detection.
- 5.8 Explain frequency domain and filtering.
- 5.9 Write about linear smoothing and nonlinear smoothing.
- 5.10 What are color spaces?
- 5.11 Write the steps in JPEG compression and image processing.
- 5.12 What is pattern matching?
- 5.13 List application areas of template matching.
- 5.14 List limitations and advantages of template matching.

Further Reading

Digital Image Processing and Computer Vision by Robert J. Schalkoff

CAMERA—IMAGE SENSOR

Overview

A digital camera is an image sensor that senses the intensity of photon values and converts them into an electrical signal. Pixel is the smallest element of image, and it corresponds to any one value of 0 to 255 in the case of an 8-bit grayscale image. Two categories of vision camera sensors are CCD and CMOS. Digital camera settings are gain, gamma, area of interest, binning/sub-sampling, pixel clock, offset, and triggering. Pixel count and size, TV Lines, Camera MTF, Nyquist limit, pixel depth / grayscale, dynamic range, and signal to noise ratio (SNR) of a camera contribute to the quality of image. Camera interfaces are Capture board, FireWire, Camera Link GigE and USB. Many parameters determine selection of camera for the vision project application. Thermal-imaging cameras record the intensity IR wave radiation into visible images to find hot spots before failure.

Learning Objectives

After reading this one will be able to

- differentiate analog and digital camera,
- perform functions such as image formation, shutter, size of image, resolution, zoom,
- compare area-scan cameras and line-scan cameras,
- Understand average versus weighted grayscale imaging; CCD versus CMOS,
- adjust camera settings and resolution for improved image,
- be familiar with different zooming methods, interface boards,

- determine camera and lens selection for vision project, and
- understand the operation of thermal imaging camera.

6.1 HISTORY OF PHOTOGRAPHY

The concepts of the camera were introduced long before the concept of photography. The history of the camera begins in Asia and is described in short here. The principles of the camera were first introduced by the Chinese philosopher Mozi in his writings, and it is known as camera obscura. The cameras evolved from this principle. The term camera obscura evolved from two different words: the meaning of the word camera is a room or some kind of vault, and obscura stands for dark. The concept, introduced by the Chinese philosopher, consists of a device that projects an image of its surroundings on a wall. However, the camera was not built by the Chinese.



FIGURE 6.1. Camera obscura.

The concept of the camera was brought to reality by a scientist named Ibn al-Haitham. He built the first camera obscura. His camera follows the principles of the pinhole camera. He built this device sometime around the year AD1000, but its first actual use was described in the 13th century by English philosopher Roger Bacon. Bacon suggested the use of camera for the observation of solar eclipses. Although much improvement was made before the 15th century, the improvements and the findings done by Leonardo daVinci were remarkable (Figure 6.1). Da Vinci was credited for many inventions. Da Vinci not only built a camera obscura following the principle of a pinhole camera, but also used it as drawing aid for his art work. In his work, which was described in *Codex Atlanticus*, many principles of camera obscura were defined.

Camera obscura follows the principle of a pinhole camera which can be described as when images of illuminated objects penetrate through a small

hole into a very dark room, you will see on the opposite wall these objects in their proper form and color, reduced in size in a reversed position, owing to the intersection of rays.

In 1685, the first portable camera was built by Johann Zahn. Before the advent of this device, cameras were the size of room and not portable. Although a transportable device was made by Irish scientists Robert Boyle and Robert Hooke, that device was too huge to carry from one place to another. The first photograph was taken in 1814 by French inventor Joseph Nicéphore Niepce. He captured the first photograph (Figure 6.2a) of a view from the window at Le Gras, by coating the pewter plate with bitumen and then exposing that plate to light. The first underwater photograph (Figure 6.2b) was taken by English mathematician William Thomson using a water-tight box. This was done in 1856.

The origin of film (Figure 6.2c) was introduced by American inventor and philanthropist George Eastman, who is considered the pioneer of photography. He founded the Eastman Kodak company which is famous for developing photographic film. The company started manufacturing paper film in 1885. Eastman first created the Kodak camera and then later the Brownie camera. The Brownie was a box camera and gained its popularity due to its Snapshot feature.

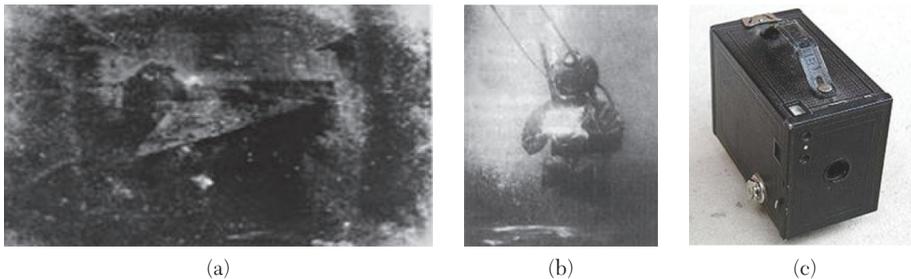


FIGURE 6.2. (a) First photograph. (b) First underwater photograph. (c) Film.

After the advent of film, the camera industry once again got a boom and one invention led to another. Leica and Argus are the two analog cameras developed in 1925 and in 1939 respectively. The camera Leica was built using 35mm cine film. Argus was another analog camera that used the 35mm format and was rather inexpensive as compared to Leica and became very popular. In 1942, German engineer Walter Bruch developed and installed the very first system of the analog CCTV camera. He was also credited for the invention of color television in the 1960.

The first disposable camera (Figure 6.3a) was introduced in 1949 by Photo Pac. The camera was only a one-time use camera with a roll of film already in it. The later versions of Photo Pac were waterproof and even had the flash. Mavica, the magnetic video camera was launched by Sony in 1981. It was the first game changer in the digital camera world (Figure 6.3b). The images were recorded on floppy disks and could be viewed later on any monitor screen. It was not a pure digital camera, but an analog camera. However, it got its popularity due to its storing capacity of images on floppy disks. That meant that you could now store images for a long period of time, and could save a huge number of pictures on the floppy which were replaced by the new blank disc when they got full. Mavica has the capacity of storing 25 images on a disk. One more important thing that Mavica introduced was its 0.3 mega pixel capacity of capturing photos. Fuji DS-1P camera by Fuji films 1988 was the first true digital camera. Nikon D1 was a 2.74 mega pixel camera and the first commercial digital SLR camera (Figure 6.3c) developed by Nikon. Nikon was affordable to professionals. Today digital cameras are included in mobile phones and have very high resolution and quality.



(a)



(b)



(c)

FIGURE 6.3. (a) Photo Pac. (b) Digital camera (Mavica). (c) Nikon

Image Formation on Cameras

The first discussion is about the image formation on the human eye because the basic principle followed by the camera has been taken from the way the human eye works. When light falls upon a particular object, it is reflected back after striking through the object. The rays of light when passed through the lens of eye, form a particular angle, and the image is formed on the retina which is the back side of the wall. The image that is formed is inverted. This image is then interpreted by the brain and that makes us able to understand things. Due to angle formation, we are able to perceive the height and depth of the object we are seeing.

As you can see in the Figure 6.4, when sun light falls on the object (in this case the object is a face), it is reflected back and different rays form different angles when they are passed through the lens and an invert image of the object has been formed on the back wall. The last portion of the figure denotes that the object has been interpreted by the brain and reinverted.

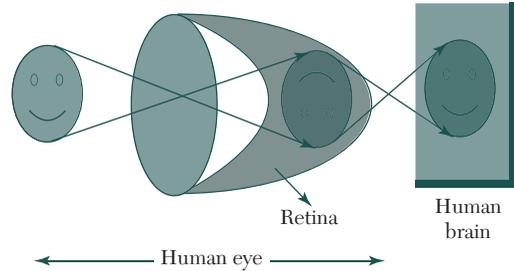


FIGURE 6.4. Image formation on human eye.

Image Formation on Analog Cameras

In analog cameras the image formation is due to the chemical reaction that takes place on the strip used for image formation. A 35mm strip is used in analog cameras. It is denoted in Figure 6.5 by a 35mm film cartridge. This strip is coated with silver halide (a chemical substance). The only light is as the small photon particles. When these photon particles are passed through the camera, it reacts with the silver halide particles on the strip and result in the silver color, which is the negative of the image. Image formation also involves many other concepts regarding the passing of light inside, as well as the concepts of shutter, shutter speed, aperture, and its opening.

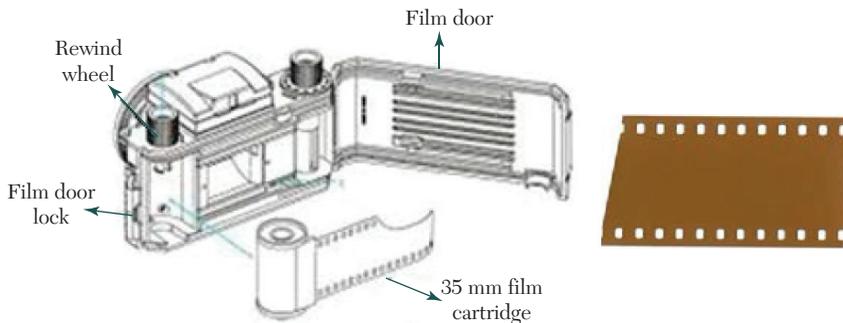


FIGURE 6.5. Analog camera and film.

Image Formation on Digital Cameras

In digital cameras, the image formation is not due to the chemical reaction that takes place, rather it is a bit more complex. In the digital camera, a charge coupled device (CCD) array of sensors is used for image

formation. It is an image sensor, and like other sensors it senses the values and converts them into an electric signal. This CCD is actually in the shape of array or a rectangular grid as shown in Figure 6.6. It is like a matrix, with each cell in the matrix containing a sensor that senses the intensity of photon.

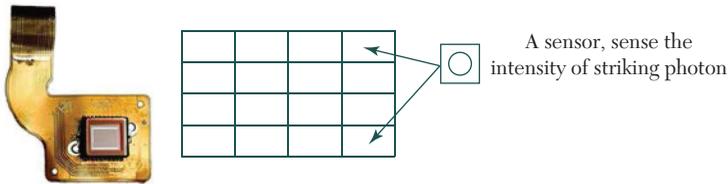


FIGURE 6.6. CCD array of sensor in a digital camera.

Like analog cameras, in the case of the digital, when light falls on the object, the light reflects back after striking the object, and is then allowed to enter inside the camera. Each sensor of the CCD array itself is an analog sensor. When photons of light strike on the chip, it is held as a small electrical charge in each photo sensor. The response of each sensor is directly equal to the amount of light or (photon) energy strike on the surface of the sensor. Since an image is defined as a two-dimensional signal and due to the two-dimensional formation of the CCD array, a complete image can be achieved from this CCD array. It has a limited number of sensors, which means limited detail can be captured by it. Also, each sensor can have only one value against each photon particle that strikes on it. So the number of photons striking (current) are counted and stored. In order to measure these accurately, external CMOS sensors are also attached with CCD array.

The value of each sensor of the CCD array refers to the value of each individual pixel. The number of sensors is equal to the number of pixels. It also means that each sensor could have only one value. The charges stored by the CCD array are converted to voltage one pixel at a time. With the help of additional circuits, this voltage is converted into digital information and then stored. Each company that manufactures digital cameras make their own CCD sensors. They include Sony, Mitsubishi, Nikon, Samsung, Toshiba, Fujifilm, and Canon. Apart from the other factors, the quality of the image captured also depends on the type and quality of the CCD array that has been used.

Camera Types and Their Advantages: Analog versus Digital Cameras

As imaging technology advances, the types of cameras and their interfaces continually evolve to meet the needs of a host of applications. For embedded vision applications in the semiconductor, electronics, biotechnology, assembly, and manufacturing industries where inspection and analysis are key, using the best camera system for the task at hand is crucial to achieving the best image quality. From analog and digital cameras, to progressive scan and interlaced scan formats, to FireWire and GigE interfaces, understanding parameters such as camera types, digital interfaces, power, and software provides a great opportunity to move from imaging novice to imaging expert.

On the most general level, cameras can be divided into two types: analog and digital. Analog cameras transmit a continuously variable electronic signal in real time. The frequency and amplitude of this signal is then interpreted by an analog output device as video information. Both the quality of the analog video signal and the way in which it is interpreted affect the resulting video images. Also, this method of data transmission has both pros and cons. Typically, analog cameras are less expensive and less complicated than their digital counterparts, making them cost-effective and simple solutions for common video applications. However, analog cameras have upper limits on both resolution (number of TV lines) and frame rate. For example, NTSC, standard video format is limited to about 800 TV lines (typically 525) and 30 frames per second. The PAL standard uses 625 TV lines and a frame rate of 25 frames per second. Analog cameras are also very susceptible to electronic noise, which depends on commonly overlooked factors such as cable length and connector type.

Digital cameras, the newest introduction and steadily becoming the most popular, transmit binary data (a stream of ones and zeroes) in the form of an electronic signal. Although the voltage corresponding to the light intensity for a given pixel is continuous, the analog to digital conversion process discretizes this and assigns a grayscale value between 0 (black) and 2^{N-1} , where N is the number of bits of the encoding. An output device then converts the binary data into video information. There are two key differences unique in digital cameras and not in analog. They are first, the digital video signal is exactly the same when it leaves the camera as when it reaches an output device, and second, the video signal can only be interpreted in one way.

These differences eliminate errors in both transmission of the signal and interpretation by an output device due to the display. Compared to analog counterparts, digital cameras typically offer higher resolution, higher frame rates, less noise, and more features. Unfortunately these advantages come with costs: digital cameras are generally more expensive than analog. Furthermore, feature-packed cameras may involve more complicated setup, even for video systems that require only basic capabilities. Digital cameras are also limited to shorter cable lengths in most cases. Table 6.1 provides a brief comparison of analog and digital camera types.

TABLE 6.1. Comparison of Analog Camera and Digital Camera Types

Analog Cameras	Digital Cameras
Vertical resolution is limited by the bandwidth of the analog signal	Vertical resolution is not limited; offer high resolution in both horizontal and vertical directions
Standard-sized sensors	With no bandwidth limit, offer large numbers of pixels and sensors, resulting in high resolution
Computers and capture boards can be used for digitizing, but are not necessary for display	Computer and capture board (in some cases) required to display signal
Analog printing and recording easily incorporated into system	Signal can be compressed so user can transmit in low bandwidth
Signal is susceptible to noise and interference, which causes loss in quality	Output signal is digital; little signal loss occurs during signal processing
Limited frame rates	High frame rates and fast shutters

Interlaced versus Progressive Scan Cameras

Camera formats can be divided into interlaced, progressive, area, and line scan. To easily compare, it is best to group them into interlaced versus progressive, and area versus line. Conventional CCD cameras use interlaced scanning across the sensor. The sensor is divided into two fields: the odd field (rows 1, 3, 5 ..., etc.) and the even field (rows 2, 4, 6 ..., etc.). These fields are then integrated to produce a full frame. For example, with a frame rate of 30 frames per second (fps), each field takes 1/60 of a second to read. For most applications, interlaced scanning does not cause a problem. However, some trouble can develop in high-speed applications because by the time the second field is scanned, the object has already moved. This causes ghosting or blurring effects in the resulting image (Figures 6.7a and



FIGURE 6.7. (a) Ghosting and blurring of TECHSPEC-man’s high-speed movement using an interlaced scanning sensor. (b) TECHSPEC-man’s high-speed movement using a progressive scanning sensor.

6.7b). In Figure 6.7a, notice how TECHSPEC Man appears skewed when taking his picture with an interlaced scanning sensor.

In contrast, progressive scanning solves the high speed issue by scanning the lines sequentially (rows 1, 2, 3, 4 ..., etc.). Unfortunately, the output for progressive scanning has not been standardized, so care should be taken when choosing hardware. Some progressive scan cameras offer an analog output signal, but few monitors are able to display the image. For this reason, capture boards are recommended to digitize the analog image for display.

Area Scan versus Line Scan Cameras

In area scan cameras, an imaging lens focuses the object to be imaged onto the sensor array, and the image is sampled at the pixel level for reconstruction (Figure 6.8). This is convenient if the image is not moving quickly or if the object is not extremely large. Familiar digital point and shoot cameras are examples of area scan devices. With line scan cameras, the pixels are arranged in a linear fashion, which allows for very long arrays. Long arrays are ideal because the amount of information to be read out per exposure decreases substantially and the speed of the readout increases by the absence of column shift registers or multiplexers; in other words, as the object moves past the camera, the image is taken line by line and reconstructed with software. Table 6.2 compares area scan and line scan cameras.

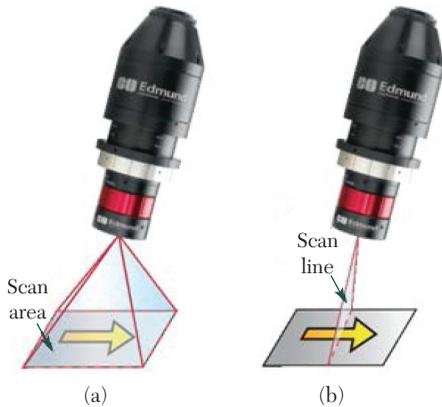


FIGURE 6.8. Illustration of area-scanning technique (left); illustration of line-scanning technique (right).

TABLE 6.2. Comparison of Area Scan Cameras and Line Scan Cameras

Area Scan Cameras	Line Scan Cameras
4:3 (H:V) ratio (Typical)	Linear sensor
Large sensors	Larger sensors
High-speed applications	High-speed applications
Fast shutter times	Constructs image one line at a time
Lower cost than line scan	Object passes in motion under sensor
Wider range of applications than line scan	Ideal for capturing wide objects
Easy setup	Special alignment and timing required; complex integration but simple illumination

Time Delay and Integration (TDI) versus Traditional Line Scan Cameras

In traditional line scan cameras, the object moves past the sensor and an image is made line by line. Since each line of the reconstructed image is from a single, short exposure of the linear array, very little light is collected. As a result, this requires substantial illumination (think of a copy machine or document scanner). The alternative is Time Delay and Integration (TDI) line scan cameras. In these arrangements, multiple linear arrays are placed side by side. After the first array is exposed, the charge is transferred to the neighboring line. When the object moves the distance of the separation between lines, a second exposure is taken on top of the first, and so on. Thus, each line of the object is imaged repeatedly, and the exposures are

added to each other. This reduces noise, thereby increasing signal. Also, it demonstrates the concept of triggering, wherein the exposure of a pixel array is synchronized with the motion of the object and the flash of the lighting.

Camera Mechanism

Aperture is a small opening that allows the light to travel into the camera. There are small blades inside the aperture. These blades create an octagonal shape that can be opened and closed. And thus it make sense that the more blades open, the hole from which the light would have to pass would be bigger. The bigger the hole, the more light is allowed to enter.

The effect of the aperture directly corresponds to brightness and darkness of an image. If the aperture opening is wide, it would allow more light to pass into the camera. More light would result in more photons, which ultimately result in a brighter image. The image on right side of Figure 6.9 (a) looks brighter, which means that when it was captured by the camera, the aperture was wide open. As compared to the other picture on the left side, which is very dark and shows that when that image was captured, its aperture was not wide open. The size of the aperture is denoted by f value. And it is inversely proportional to the opening of aperture. Large aperture size = Small f value; Small aperture size = Greater f value; pictorially it is shown in Figure 6.9 (b).

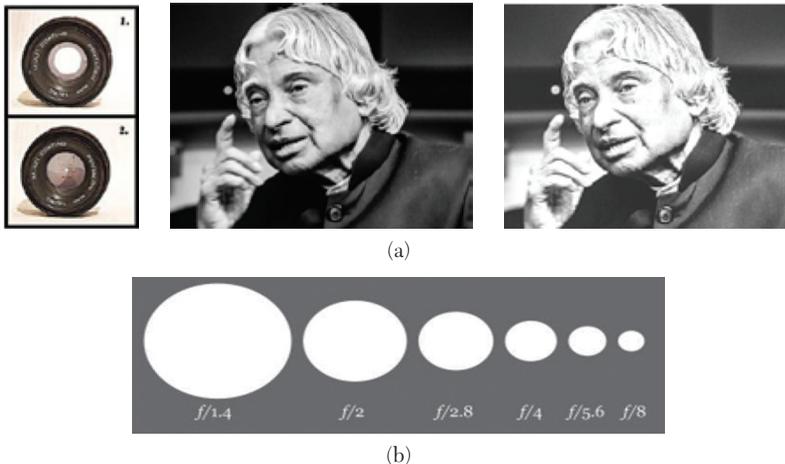


FIGURE 6.9 (a) Aperture and its effect (b) Aperture size

Perspective Transformation

When human eyes see things that are near, they look bigger as compared to when things are far away. This is called perspective. Whereas transformation is the transfer of an object from one state to another. So overall, the perspective transformation deals with the conversion of 3d world into 2-D image. This is the same principle on which human vision and the camera works. Frame of reference is basically a set of values in relation to which we measure something. In order to analyze a 3-D world/image/scene, five different frame of references are required. They are (1) object, (2) world, (3) camera, (4) image, and (5) pixel.

Object coordinate frame is used for modeling objects. For example, this involves checking if a particular object is in a proper place with respect to the other object. It is a 3-D-coordinate system. World coordinate frame is used for correlating objects in a 3-dimensional world. It is a 3-D coordinate system. Camera coordinate frame is used to relate objects with respect of the camera. It is a 3-D coordinate system. Image coordinate frame is not a 3-D-coordinate system, rather it is a 2-D system. It is used to describe how 3-D points are mapped in a 2-D-image plane. Pixel coordinate frame is also a 2-D-coordinate system. Each pixel has a value of pixel coordinates. That's how a 3-D-scene is transformed into 2-D, with the image of pixels as in Figure 6.10. This concept is mathematically written as $\tan\theta = -\frac{y}{f}$, where minus denotes that the image is inverted. The second angle formed is: $\tan\theta = \frac{Y}{Z}$. Comparing these two equations we get $y = -f\frac{Y}{Z}$; from this equation, when the rays of light reflect back after striking from the object, passed from the camera, an invert image is formed. For calculating the size of image formed, suppose an image has been taken of a person 5m tall, standing at a distance of 50m from the camera, with a camera of focal length of 50mm:

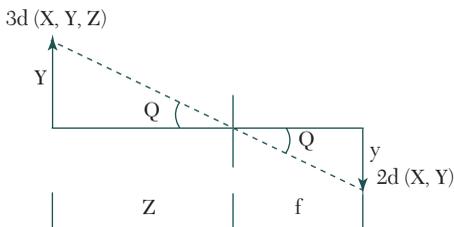


FIGURE 6.10. Image coordinate system.

Solution: Since the focal length is in millimeter, convert everything in millimeter in order to calculate it. So, $Y = 5000$ mm. $f = 50$ mm. $Z = 50000$ mm. Putting these values in the formula,

$$y = -f \frac{Y}{Z} = -50 \times \frac{5000}{50000} = -5 \text{ mm}$$

The minus sign indicates that the image is inverted.

Pixel

Pixel is the smallest element of an image. Each pixel corresponds to any one value. In an 8-bit grayscale image, the value of the pixel is between 0 and 255. The value of a pixel at any point corresponds to the intensity of the light photons striking at that point. Each pixel stores a value proportional to the light intensity at that particular location. A pixel is also known as PEL. In the picture, there may be thousands of pixels that together make up this image. We will zoom that image to the extent that we are able to see some pixel division, as shown in Figure 6.11. The smallest division of the CCD array is also known as pixel. Each division of CCD array contains the value against the intensity of the photon striking to it. This value can also be called a pixel.

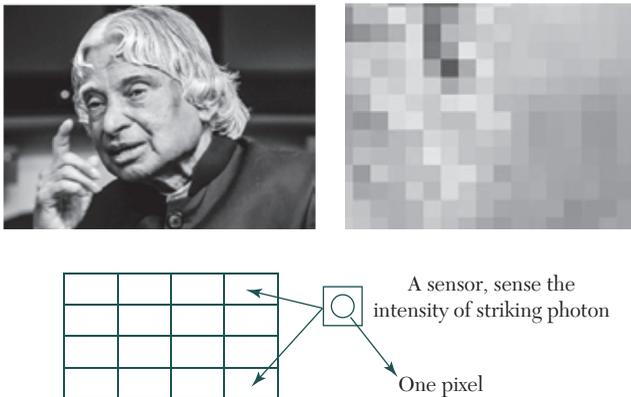


FIGURE 6.11. Pixel concept.

Calculation of total number of pixels

An image is as a two dimensional signal or matrix. Then in that case the number of PEL would be equal to the number of rows multiply with number of columns. This can be mathematically represented as below: Total number of pixels = number of rows X number of columns. The value

of the pixel at any point denotes the intensity of image at that location, and that is also known as gray level. Each pixel can have only one value and each value denotes the intensity of light at that point of the image. The value 0 means absence of light. It means that 0 denotes dark, and it further means that whenever a pixel has a value of 0, it means at that point, black color would be formed.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Now this image matrix has all filled up with 0. All the pixels have a value of 0. Total no of pixels = total no. of rows X total no. of columns = $3 \times 3 = 9$. It means that an image would be formed with 9 pixels, and that image would have a dimension of 3 rows and 3 column and most importantly that image would be black. The resulting image that would be made looks like this . The total number of combinations that can be made from bit, would be $(2)^{\text{bpp}}$, where bpp denotes bits per pixel. It grows exponentially. The number of different colors depend on the number of bits per pixel. The Table 6.3 shows bits per pixel and number of colors.

TABLE 6.3 Bits per pixel and number of colors

Bits per pixel	Number of colors
1 bpp	2 colors
2 bpp	4 colors
3 bpp	8 colors
4 bpp	16 colors
5 bpp	32 colors
6 bpp	64 colors
7 bpp	128 colors
8 bpp	256 colors
10 bpp	1024 colors
16 bpp	65536 colors
24 bpp	16777216 colors (16.7 million colors)
32 bpp	4294967296 colors (4294 million colors)

The famous grayscale image is of 8 bpp, means it has 256 different colors in it or 256 shades. Shades can be represented as: Shades = number of colors = $(2)^{\text{bpp}}$. Color images are usually of the 24 bpp format or 16 bpp.

0 pixel value denotes black color. But there is no fixed value that denotes white color. The value that denotes white color can be calculated as: White color = $(2)^{\text{bpp}} - 1$. In the case of 1 bpp, 0 denotes black, and 1 denotes white. In the case of 8 bpp, 0 denotes black, and 255 denotes white. Gray color is actually the midpoint of black and white. In the case of 8 bpp, the pixel value that denotes gray color is 127 or 128 bpp (if you count from 1, not from 0).

Image storage requirements

The size of an image depends upon three things. They are (1) the number of rows, (2) the number of columns, and (3) the number of bits per pixel. The formula for calculating the size is given as,

$$\text{Size of an image} = \text{rows} * \text{cols} * \text{bpp};$$

Assuming an image has 1024 rows and it has 1024 columns, and it is a grayscale image, it has 256 different shades of gray or it has 8 bits per pixel. Then putting these values in the formula,

$$\text{Size of an image} = \text{rows} * \text{cols} * \text{bpp} = 1024 * 1024 * 8 = 8388608 \text{ bits}$$

$$\text{Converting it into bytes} = 8388608 / 8 = 1048576 \text{ bytes}$$

$$\text{Converting into kilo bytes} = 1048576 / 1024 = 1024 \text{ kb}$$

$$\text{Converting into Mega bytes} = 1024 / 1024 = 1 \text{ Mb}$$

There are many types of images, and the color distribution in them. The binary image as its name states, contains only two pixel values: 0 and 1. Here 0 refers to black color and 1 refers to white color. It is also known as monochrome. Black and white image is an image which has only black and white color and thus can also be called as black and white image (Figure 6.12).

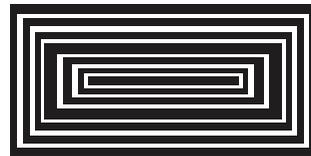


FIGURE 6.12. Black and white image.

One of the interesting things about this binary image is that there is no gray level in it—only two colors, black and white, are found in it. Binary images have a format of portable bit map (PBM) 2, 3, 4, 5, 6 bit color format. The images with a color format of 2, 3, 4, 5 and 6 bit are not widely used today. They were used for old TV or monitor displays. But each of these colors has more than two gray levels, and hence has gray color unlike the binary image. In a 2 bit 4, in a 3 bit 8, in a 4 bit 16, in a 5 bit 32, in a 6 bit 64 different colors are present.

8 bit color format

One of the most famous image formats is 8 bit color format. It has 256 different shades of colors in it. It is commonly known as a grayscale image. The range of colors in 8 bit vary from 0–255, where 0 stands for black, 255 stands for white, and 127 stands for gray color. This format was used initially by early models of the operating systems UNIX and the early color Macintoshes. The format of these images are portable gray map (PGM). This format is not supported by default from Windows. Grayscale image is nothing but a two-dimensional function, and can be represented by a two-dimensional array or matrix.

16 bit color format

Another color image format is 16 bit. It has 65,536 different colors in it. It is also known as high color format. It has been used by Microsoft in their systems that support more than 8 bit color format. The 16 bit format and 24 bit format are both color format. A 16 bit format is actually divided into three further formats: red, green, and blue (RGB) format. The distribution of 16 bit is 5 bits for R, 6 bits for G, and 5 bits for B. The additional one bit is added into the green bit because green is the color most soothing to the eyes in all of these three colors. Note this distribution is not followed by all the systems. Some have introduced an alpha channel in the 16 bit. Another distribution of 16 bit format is as follows: 4 bits for R, 4 bits for G, 4 bits for B, 4 bits for alpha channel. Or some distribute it like this: 5 bits for R, 5 bits for G, 5 bits for B, 1 bits for alpha channel.

24 bit color format

24 bit color format also known as true color format. Like 16 bit color format, in a 24 bit color format, the 24 bits are again distributed in three different formats of Red, Green, and Blue. Since 24 is equally divided on 8, so it has been distributed equally between three different color channels. Their distribution is 8 bits for R, 8 bits for G, 8 bits for B. Its format is portable pixmap (PPM) which is supported by Linux operating system. Windows has its own format for it: bitmap (BMP). 0 refers to black. So, to make a pure black color, all three portions of R, G, B, to 0. Each portion of R, G, B is an 8 bit portion. So in an 8-bit, the white color is formed by 255. So in order to make a white color, set each portion RGB to 255.

RGB color model

Red image  decimal code value is (255, 0, 0). For the color red, set green and blue portions to zero, and set the red portion to its

maximum of 255. The green image is  decimal code is (0, 255, 0). For the color green set red and blue portions to zero, and set the green portion to its maximum of 255. The blue image  decimal code is (0, 0, 255). For the color blue set red and green portions to zero, and set the blue portion to its maximum of 255. The color gray image  decimal code is (128, 128, 128). The color gray is actually the midpoint. In an 8-bit format, the midpoint is 128 or 127. In this case we have chosen 128. So set each portion to its midpoint of 128, and that results in overall mid value and produces the color gray.

CMYK color model

CMYK is another color model where c stands for cyan, m stands for magenta, y stands for yellow, and k for black. The CMYK model is commonly used in color printers in which there are two carters of color used. One consists of CMY and the other consists of black. The colors of CMY can also made from changing the quantity or portion of red, green, and blue. The cyan image  decimal code is (0,255,255). Cyan is formed from the combination of two different colors: green and blue. Set those two to maximum and zero out the portion of red and you get the color cyan. The magenta image  decimal code is (255, 0, 255). Magenta is formed from the combination of red and blue. Set those two colors to maximum and zero out the portion of green and you get magenta. The yellow image  decimal code is (255, 255, 0). The color yellow is formed from the combination of red and green. Set those two colors to maximum and zero out the portion of blue to get yellow. Common colors and their hex code are shown in Table 6.4.

TABLE 6.4. Colors and Their Hex Codes

Color	Hex Code
Black	#000000
White	#FFFFFF
Gray	#808080
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Cyan	#00FFFF
Magenta	#FF00FF
Yellow	#FFFF00

Grayscale to RGB conversion

To convert a color image into a grayscale image, there are two methods that can be used. Both have their own merits and demerits. The methods are average method and weighted method or luminosity method. Average method is the simplest one. It averages the intensity of three colors. Since it's an RGB image, add r with g with b and then divide it by 3 to get the desired grayscale image. $\text{Grayscale} = (R + G + B / 3)$. The average method has demerit. If both images are compared, the average method results were not as expected. This method turned out to be a rather black image instead of grayscale image as shown in Figure 6.13(b).

This problem arises due to the fact that this method takes the average of the three colors. Since the three different colors have three different wavelengths and have their own contribution in the formation of the image, one must take the average according to their contribution, not by using the average method. That is 33% of red, 33% of green, and 33% of blue are used for averaging. That means it takes 33% of each, and each portion has the same contribution in the image. But in reality that's not the case. The solution to this has been given by the luminosity method.

Weighted method or luminosity method has a solution to the problem of average method. Since red has more wavelengths of all three colors, and green is the color that not only has less wavelengths than red, but also is the color that gives a more soothing effect to the eyes. It means that one has to decrease the contribution of red and increase the contribution of green, and put the contribution of blue in between these two. So the new equation is given by a new grayscale image = $((0.3 * R) + (0.59 * G) + (0.11 * B))$. According to this equation, red has contributed 30%, green has contributed 59%, and blue has contributed 11%. Applying this equation to the image, original image, and its grayscale image are shown in Figure 6.14(c). As compared to the result of average method, this image is brighter.

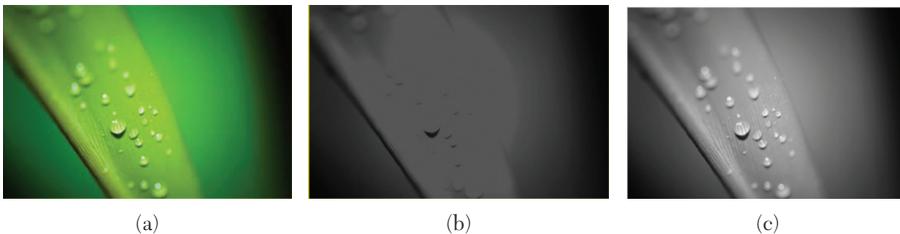


FIGURE 6.13 (a) RGB color image; (b) average method grayscale image; (c) weighted grayscale image.

Resolution

A pixel can store a value proportional to light intensity at a particular location. The resolution can be defined as pixel resolution, spatial resolution, temporal resolution, and spectral resolution. The resolution is monitored as 800×600 , 640×480 , etc. In pixel resolution, the term resolution refers to the total number of pixels in a digital image. For example, if an image has M rows and N columns, then its resolution can be defined as $M \times N$. If resolution is defined as the total number of pixels, then pixel resolution can be defined with set of two numbers. The first number is the width of the picture, or the pixels across columns; and the second number is the height of the picture, or the pixels across its width. The higher the pixel resolution, the higher is the quality of the image. It is defined pixel resolution of an image as 4500×5500 . One can calculate mega pixels of a camera using pixel resolution. Column pixels (width) \times row pixels (height) / 1 million. The size of an image can be defined by its pixel resolution. Size = pixel resolution \times bpp (bits per pixel). Consider an image of dimension: 2500×3192 . Its pixel resolution = $2500 * 3192 = 7982350$ bytes. Dividing it by 1 million = $7.9 = 8$ mega pixel (approximately).

Aspect ratio

Another important concept with pixel resolution is aspect ratio. Aspect ratio is the ratio between width of an image and the height of an image. It is commonly explained as two numbers separated by a colon (8:9). This ratio differs in different images, and in different screens. The common aspect ratios are: 1.33:1, 1.37:1, 1.43:1, 1.50:1, 1.56:1, 1.66:1, 1.75:1, 1.78:1, 1.85:1, 2.00:1, etc. Aspect ratio maintains a balance in the appearance of an image on the screen, in other words it means it maintains a ratio between horizontal and vertical pixels. It does not let the image get distorted when the aspect ratio is increased. With the aspect ratio, one can calculate the dimensions of the image along with the size of the image. If you are given an image with aspect ratio of 6:2 of an image of pixel resolution of 480000 pixels given the image is a grayscale image.

6.2 CAMERA SENSOR FOR EMBEDDED VISION APPLICATIONS

Imaging electronics, in addition to imaging optics, play a significant role in the performance of an imaging system. Proper integration of all components, including camera, capture board, software, and cables results in optimal system performance. It is important to understand the camera sensor and key concepts and terminology associated with it.

The heart of any camera is the sensor; modern sensors are solid-state electronic devices containing up to millions of discrete photo detector sites called pixels. Although there are many camera manufacturers, the majority of sensors are produced by only a handful of companies. Still, two cameras with the same sensor can have very different performance and properties due to the design of the interface electronics. In the past, cameras used phototubes such as vidicons and plumbicons as image sensors. Though they are no longer used, their mark on nomenclature associated with sensor size and format remains to this day. Today, almost all sensors in embedded vision fall into one of two categories: charge coupled device (CCD) and complementary metal oxide semiconductor (CMOS) imagers.

Charge Coupled Device (CCD) Sensor Construction

The charge coupled device (CCD) was invented in 1969 by scientists at Bell Labs in New Jersey, United States. For years, it was the prevalent technology for capturing images, from digital astrophotography to embedded vision inspection. The CCD sensor is a silicon chip that contains an array of photosensitive sites. It is shown in Figure 6.14.

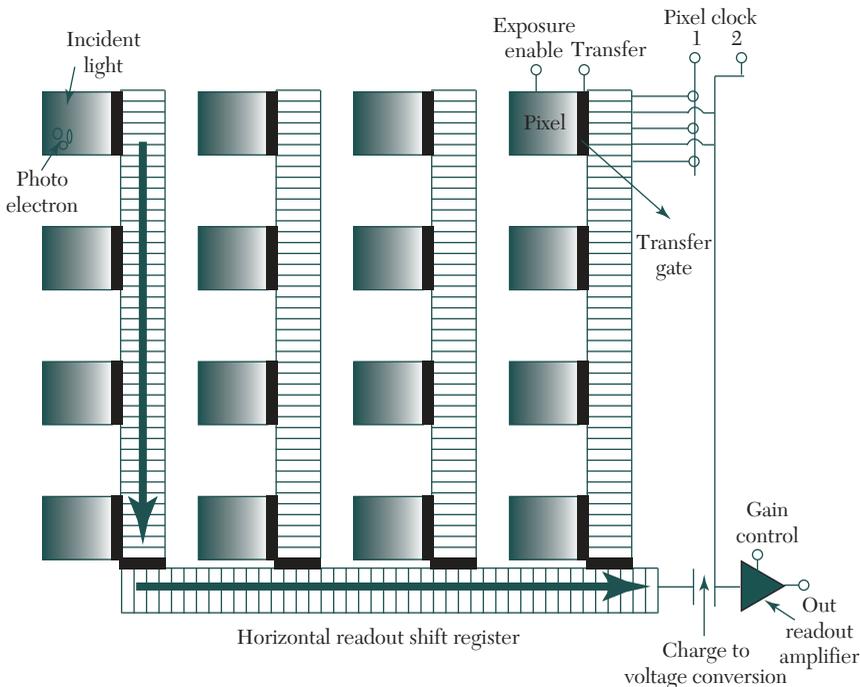


FIGURE 6.14. Block diagram of a charge coupled device (CCD).

The term charge coupled device actually refers to the method by which charge packets are moved around on the chip from the photosites to readout. Clock pulses create potential wells to move charge packets around on the chip, before being converted to a voltage by a capacitor. The CCD sensor is itself an analog device, but the output is immediately converted to a digital signal by means of an analog to digital converter (ADC) in digital cameras, either on or off chip. In analog cameras, the voltage from each site is read out in a particular sequence, with synchronization pulses added at some point in the signal chain for reconstruction of the image.

The charge packets are limited to the speed at which they can be transferred, so the charge transfer is responsible for the main CCD drawback of speed, but also leads to the high sensitivity and pixel to pixel consistency of the CCD. Since each charge packet sees the same voltage conversion, the CCD is very uniform across its photosensitive sites. The charge transfer also leads to the phenomenon of blooming, wherein charge from one photosensitive site spills over to neighboring sites due to a finite well depth or charge capacity, placing an upper limit on the useful dynamic range of the sensor. This phenomenon manifests itself as the smearing out of bright spots in images from CCD cameras.

To compensate for the low well depth in the CCD, micro lenses are used to increase the fill factor, or effective photosensitive area, to compensate for the space on the chip taken up by the charge coupled shift registers. This improves the efficiency of the pixels, but increases the angular sensitivity for incoming light rays, requiring that they hit the sensor near normal incidence for efficient collection.

Complementary Metal Oxide Semiconductor (CMOS) Sensor Construction

The complementary metal-oxide semiconductor (CMOS) was invented in 1963 by Frank Wanlass. However, he did not receive a patent for it until 1967, and it did not become widely used for imaging applications until the 1990s.

In a CMOS sensor, the charge from the photosensitive pixel is converted to a voltage at the pixel site and the signal is multiplexed by row and column to multiple on chip digital to analog converters (DACs). Inherent to its design, CMOS is a digital device. Each site is essentially a photodiode and three transistors, performing the functions of resetting or activating the pixel, amplification and charge conversion, and selection or multiplexing.

It is shown in Figure 6.15. This leads to the high speed of CMOS sensors, but also low sensitivity as well as high fixed pattern noise due to fabrication inconsistencies in the multiple charge to voltage conversion circuits.

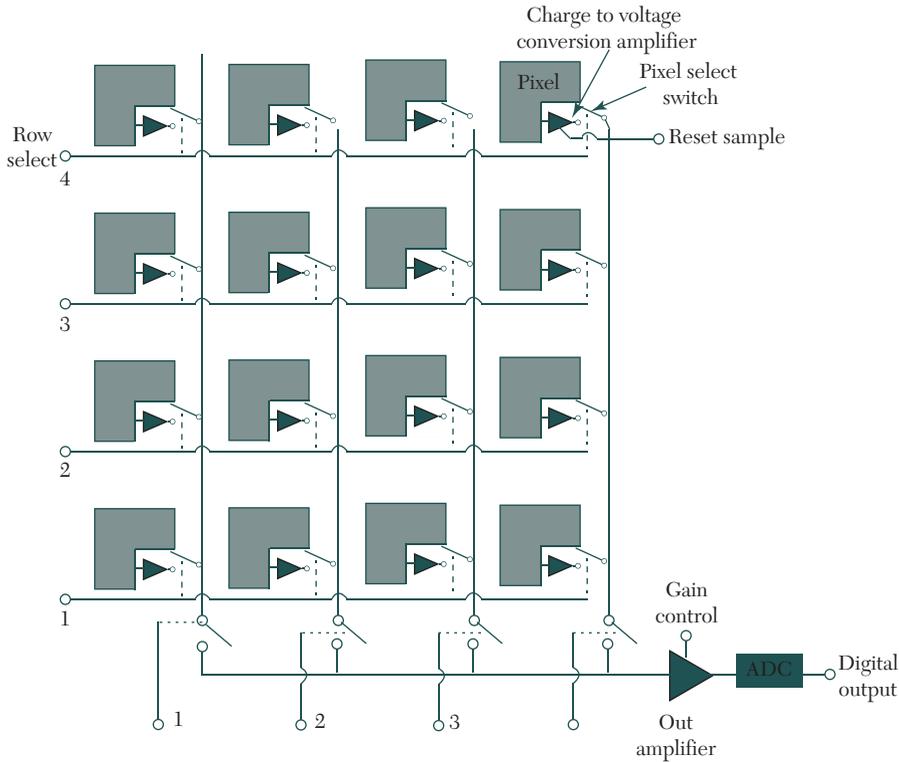


FIGURE 6.15. Block diagram of a complementary metal oxide semiconductor (CMOS).

The multiplexing configuration of a CMOS sensor is often coupled with an electronic rolling shutter; although with additional transistors at the pixel site, a global shutter can be accomplished wherein all pixels are exposed simultaneously and then readout sequentially. An additional advantage of a CMOS sensor is its low power consumption and dissipation compared to an equivalent CCD sensor, due to less flow of charge, or current. Also, the CMOS sensor's ability to handle high light levels without blooming allows for its use in special high dynamic range cameras, even capable of imaging welding seams or light filaments. CMOS cameras also tend to be smaller than their digital CCD counterparts, as digital CCD cameras require additional off chip ADC circuitry.

The multilayer MOS fabrication process of a CMOS sensor does not allow for the use of micro lenses on the chip, thereby decreasing the effective collection efficiency or fill factor of the sensor in comparison with a CCD equivalent. This low efficiency combined with pixel to pixel inconsistency contributes to a lower signal to noise ratio and lower overall image quality than CCD sensors. Table 6.5 gives general comparison of CCD and CMOS sensors.

TABLE 6.5. Comparison of (CCD) and (CMOS) Sensors

Sensor	CCD	CMOS
Pixel Signal	Electron Packet	Voltage
Chip Signal	Analog	Digital
Fill Factor	High	Moderate
Responsivity	Moderate	Moderate–High
Noise Level	Low	Moderate–High
Dynamic Range	High	Moderate
Uniformity	High	Low
Resolution	Low–High	Low–High
Speed	Moderate–High	High
Power Consumption	Moderate–High	Low
Complexity	Low	Moderate
Cost	Moderate	Moderate

Alternative Sensor Materials

Short wave infrared (SWIR) is an emerging technology in imaging. It is typically defined as light in the $0.9\text{--}1.7\mu\text{m}$ wavelength range, but can also be classified from $0.7\text{--}2.5\mu\text{m}$. Using SWIR wavelengths allows for the imaging of density variations, as well as through obstructions such as fog. However, a normal CCD and CMOS image is not sensitive enough in the infrared to be useful. As such, special indium gallium arsenide (InGaAs) sensors are used. The InGaAs material has a band gap, or energy gap, that makes it useful for generating a photocurrent from infrared energy. These sensors use an array of InGaAs photodiodes, generally in the CMOS sensor architecture.

At even longer wavelengths than SWIR, thermal imaging becomes dominant. For this, a micro bolometer array is used for its sensitivity in the $7\text{--}14\mu\text{m}$ wavelength range. In a micro bolometer array, each pixel has

a bolometer which has a resistance that changes with temperature. This resistance change is readout by conversion to a voltage by electronics in the substrate. It is shown in Figure 6.16. These sensors do not require active cooling, unlike many infrared imagers, making them quite useful.

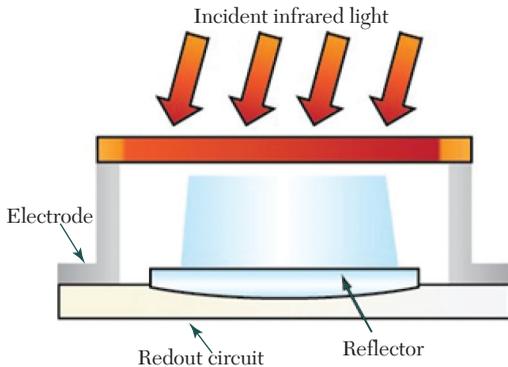


FIGURE 6.16. Illustration of cross-section of micro bolometer sensor array.

Sensor Features

When light from an image falls on a camera sensor, it is collected by a matrix of small potential wells called pixels. The image is divided into small discrete pixels. The information from these photosites is collected, organized, and transferred to a monitor to be displayed. The pixels may be photodiodes or photo capacitors, for example, which generate a charge proportional to the amount of light incident on that discrete place of the sensor, spatially restricting and storing it. The ability of a pixel to convert an incident photon to charge is specified by its quantum efficiency. For example, if for ten incident photons, four photo-electrons are produced, then the quantum efficiency is 40%. Typical values of quantum efficiency for solid-state imagers are in the range of 30–60%. The quantum efficiency depends on wavelength and is not necessarily uniform over the response to light intensity. Spectral response curves often specify the quantum efficiency as a function of wavelength.

In digital cameras, pixels are typically square. Common pixel sizes are between 3–10 μm . Although sensors are often specified simply by the number of pixels, the size is very important to imaging optics. Large pixels have, in general, high charge saturation capacities and high signal to noise ratios (SNRs). With small pixels, it becomes fairly easy to achieve high

resolution for a fixed sensor size and magnification, although issues such as blooming become more severe and pixel crosstalk lowers the contrast at high spatial frequencies. A simple measure of sensor resolution is the number of pixels per millimeter.

Analog CCD cameras have rectangular pixels (larger in the vertical dimension). This is a result of a limited number of scanning lines in the signal standards (525 lines for NTSC, 625 lines for PAL) due to bandwidth limitations. Asymmetrical pixels yield higher horizontal resolution than vertical. Analog CCD cameras (with the same signal standard) usually have the same vertical resolution. For this reason, the imaging industry standard is to specify resolution in terms of horizontal resolution. Figure 6.17 illustrates camera sensor pixels with RGB color and infrared blocking filters.

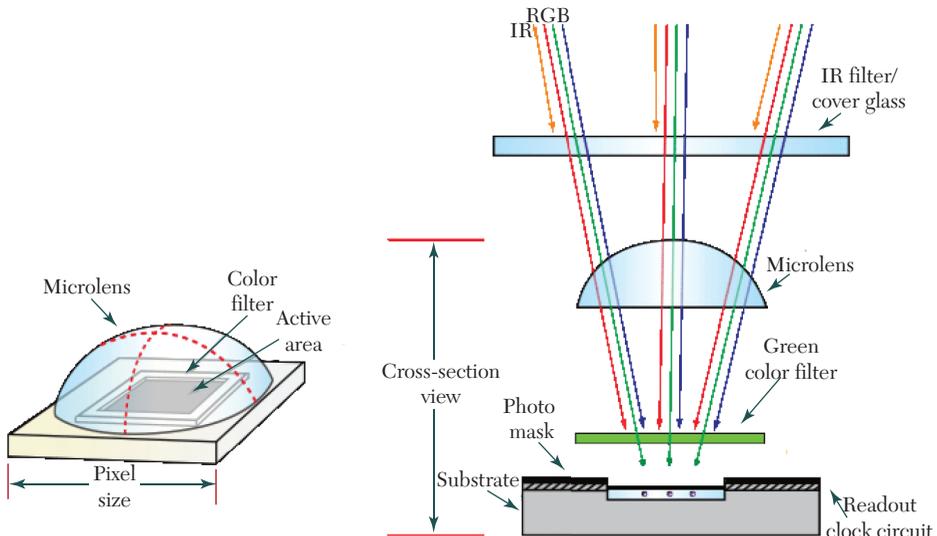


FIGURE 6.17. Illustration of camera sensor pixels with RGB color and infrared blocking filters.

Sensor Size

The size of a camera sensor's active area is important in determining the system's field of view (FOV). Given a fixed primary magnification (determined by the imaging lens), larger sensors yield greater FOVs. There are several standard area scan sensor sizes: $\frac{1}{4}$ ", $\frac{1}{3}$ ", $\frac{1}{2}$ ", $\frac{1}{1.8}$ ", $\frac{2}{3}$ ", 1 " and 1.2 ", with larger available. The nomenclature of these standards dates back to the vidicon vacuum tubes used for television broadcast imagers. There is no direct connection between the sensor size and its dimensions. However,

most of these standards maintain a 4:3 (horizontal: vertical) dimensional aspect ratio.

One issue that often arises in imaging applications is the ability of an imaging lens to support certain sensor sizes. If the sensor is too large for the lens design, the resulting image may appear to fade away and degrade toward the edges because of vignetting (extinction of rays which pass through the outer edges of the imaging lens). This is commonly referred to as the tunnel effect, since the edges of the field become dark. Smaller sensor sizes do not yield this vignetting issue.

Frame Rate and Shutter Speed

The frame rate refers to the number of full frames (which may consist of two fields) composed in a second. For example, an analog camera with a frame rate of 30 frames/second contains two 1/60 second fields. In high-speed applications, it is beneficial to choose a faster frame rate to acquire more images of the object as it moves through the FOV.

The shutter speed corresponds to the exposure time of the sensor. The exposure time controls the amount of incident light. Camera blooming (caused by over exposure) can be controlled by decreasing illumination, or by increasing the shutter speed. Increasing the shutter speed can help in creating snap shots of a dynamic object which may only be sampled 30 times per second.

Unlike analog cameras where, in most cases, the frame rate is dictated by the display, digital cameras allow for adjustable frame rates. The maximum frame rate for a system depends on the sensor readout speed, the data transfer rate of the interface including cabling, and the number of pixels (amount of data transferred per frame). In some cases, a camera may be run at a higher frame rate by reducing the resolution by binning pixels together or restricting the area of interest. This reduces the amount of data per frame, allowing for more frames to be transferred for a fixed transfer rate. To a good approximation, the exposure time is the inverse of the frame rate. However, there is a finite minimum time between exposures (on the order of hundreds of microseconds) due to the process of resetting pixels and reading out, although many cameras have the ability to readout a frame while exposing the next time (pipelining); this minimum time can often be found on the camera datasheet.

CMOS cameras have the potential for higher frame rates, as the process of reading out each pixel can be done more quickly than with the charge

transfer in a CCD sensor's shift register. For digital cameras, exposures can be made from tens of seconds to minutes, although the longest exposures are only possible with CCD cameras, which have lower dark currents and noise compared to CMOS. The noise intrinsic to CMOS imagers restricts their useful exposure to only seconds.

Electronic Shutter

Until a few years ago, CCD cameras used electronic or global shutters, and all CMOS cameras were restricted to rolling shutters. A global shutter is analogous to a mechanical shutter, in that all pixels are exposed and sampled simultaneously, with the readout then occurring sequentially; the photon acquisition starts and stops at the same time for all pixels. On the other hand, a rolling shutter exposes, samples, and reads out sequentially; it implies that each line of the image is sampled at a slightly different time.

Intuitively, images of moving objects are distorted by a rolling shutter; this effect can be minimized with a triggered strobe placed at the point in time where the integration period of the lines overlaps. Note that this is not an issue at low speeds. Implementing global shutter for CMOS requires a more complicated architecture than the standard rolling shutter model, with an additional transistor and storage capacitor, which also allows for pipelining, or beginning exposure of the next frame during the readout of the previous frame. Since the availability of CMOS sensors with global shutters is steadily growing, both CCD and CMOS cameras are useful in high speed motion applications.

In contrast to global and rolling shutters, an asynchronous shutter refers to the triggered exposure of the pixels. That is, the camera is ready to acquire an image, but it does not enable the pixels until after receiving an external triggering signal. This is opposed to a normal constant frame rate, which can be thought of as internal triggering of the shutter. Figure 6.18a compares motion blur for sensor chip on a fast-moving conveyer with triggered global shutter (left) and continuous global shutter (right) and Figure 6.18b compares motion blur in global and rolling shutters for sensor chip on a slow moving conveyer with global shutter (left) and rolling shutter (right).

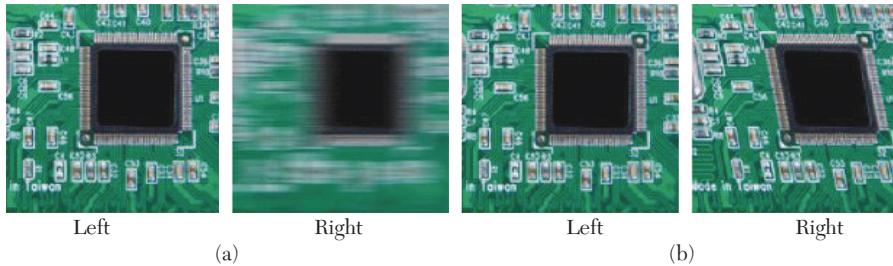


FIGURE 6.18 (a) Comparison of motion blur. Sensor chip on a fast-moving conveyor with triggered global shutter (left) and continuous global shutter (right). (b) Comparison of motion blur in global and rolling shutters. Sensor chip on a slow-moving conveyor with global shutter (left) and rolling shutter (right).

When allowed to pass from the aperture, the light falls directly on to the shutter. The shutter is actually a cover, a closed window, or can be thought of as a curtain. The shutter is the sensor. So the shutter is the only thing between the image formation and the light, when it is passed from aperture. As soon as the shutter is open, light falls on the image sensor, and the image is formed on the array. If the shutter allows light to pass a bit longer, the image would be brighter. Similarly a darker picture is produced, when a shutter is allowed to move very quickly and hence, the light that is allowed to pass has very less photons, and the image that is formed on the CCD array sensor is very dark.

The shutter has further two main concepts such as shutter speed and shutter time. The shutter speed can be referred as the number of times the shutter get open or close. It is not about for how long the shutter get open or close. When the shutter is open, the amount of time to wait it take till it is closed is called shutter time. In this case we are not talking about how many times the shutter opened or closed, but rather how much time it remains wide open. For example, let's say that a shutter opens 15 times and then closed, and then for each time it opens for 1 second and then get closed. In this example, 15 is the shutter speed and 1 second is the shutter time. The relationship between shutter speed and shutter time is that they are both inversely proportional to each other. This relationship can be defined in the equation as, more shutter speed = less shutter time; less shutter speed = more shutter time. These two concepts together make a variety of applications.

Imagine if you were to capture the image of a fast-moving object, such as a car. The adjustment of the shutter speed and its time would effect a lot. So, in order to capture an image like this, we will make two amendments: increase shutter speed and decrease shutter time. When we increase shutter speed more number of times, the shutter would open or close. It means

different samples of light would be allowed to pass in. When we decrease shutter time, we will immediately capture the scene and close the shutter gate to get a crisp image of a fast-moving object. With shutter speed of 1 second, 1/3 second and 1/200 the captured image of fast-moving waterfall photos are shown in Figure 6.19. In the last picture, we've increased shutter speed to very fast, which means the shutter opened or closed in 200th of 1 second and that enabled us to get a crisp image.



FIGURE 6.19. Shutter speed 1s, 1/3s, and 1/200s captured photo.

Sensor Taps

One way to increase the readout speed of a camera sensor is to use multiple taps on the sensor. This means that instead of all pixels being read out sequentially through a single output amplifier and ADC, the field is split and read to multiple outputs. This is commonly seen as a dual tap where the left and right halves of the field are readout separately. This effectively doubles the frame rate, and allows the image to be reconstructed easily by software. It is important to note that if the gain is not the same between the sensor taps, or if the ADCs have slightly different performance, as is usually the case, then a division occurs in the reconstructed image. The good news is that this can be calibrated out. Many large sensors which have more than a few million pixels use multiple sensor taps. This, for the most part, only applies to progressive scan digital cameras; otherwise, there will be display difficulties. The performance of a multiple tap sensor depends largely on the implementation of the internal camera hardware.

Spectral Properties of Monochrome and Color Cameras

CCD and CMOS sensors are sensitive to wavelengths from approximately 350–1050nm, although the range is usually given from 400–1000nm. This sensitivity is indicated by the sensor's spectral response curve, as shown in Figure 6.20. Most high-quality cameras provide an infrared (IR) cut-off filter for imaging, specifically in the visible spectrum. These filters are sometimes removable for near IR imaging. CMOS sensors are, in general, more sensitive to IR wavelengths than CCD sensors. This results from their

increased active area depth. The penetration depth of a photon depends on its frequency, so deeper depths for a given active area thickness produces less photoelectrons and decreases quantum efficiency.

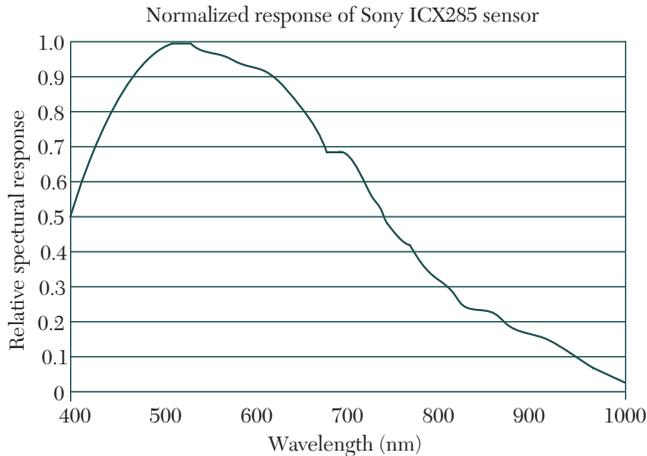


FIGURE 6.20. Normalized spectral response of a typical monochrome CCD.

The solid-state sensor is based on a photoelectric effect and, as a result, cannot distinguish between colors. There are two types of color CCD cameras: single chip and three chip. Single-chip color CCD cameras offer a common, low-cost imaging solution and use a mosaic (e.g., Bayer) optical filter to separate incoming light into a series of colors. Each color is, then, directed to a different set of pixels (Figure 6.21a). The precise layout of the mosaic pattern varies between manufacturers. Since more pixels are required to recognize color, single chip color cameras inherently have lower resolution than their monochrome counterparts; the extent of this issue is dependent upon the manufacturer specific color interpolation algorithm.

Three-chip color CCD cameras are designed to solve this resolution problem by using a prism to direct each section of the incident spectrum to a different chip (Figure 6.21b). More accurate color reproduction is possible, as each point in space of the object has separate RGB intensity values, rather than using an algorithm to determine the color. Three-chip cameras offer extremely high resolutions but have lower light sensitivities and can be costly. In general, special 3CCD lenses are required that are well corrected for color and compensate for the altered optical path and, in the case of C mount, reduced clearance for the rear lens protrusion. In the end, the choice of single chip or three chip comes down to application requirements.

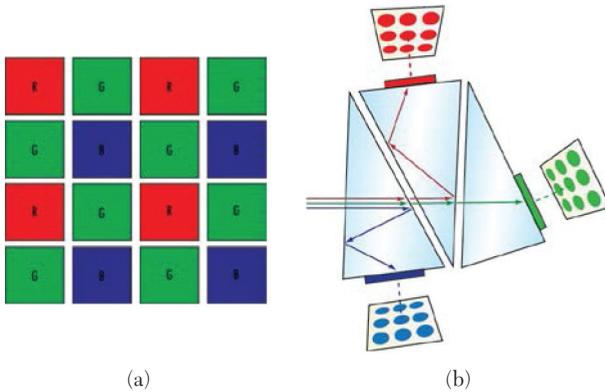


FIGURE 6.21 (a) Single-chip color CCD camera sensor using mosaic filter to filter colors. (b) Three-chip color CCD camera sensor using prism to disperse colors.

The most basic component of a camera system is the sensor. The type of technology and features greatly contributes to the overall image quality, therefore knowing how to interpret camera sensor specifications will ultimately lead to choosing the best imaging optics to pair with it.

Basics of digital camera settings for improved image results

Digital cameras, compared to their analog counterparts, offer greater flexibility in allowing the user to adjust camera settings through acquisition software. In some cases, the settings in analog cameras can be adjusted through hardware such as dual in line package (DIP) switches or RS232 connections. Nevertheless, the flexibility of modifying settings through the software greatly adds to increased image quality, speed, and contrast factors that could mean the difference between observing a defect and missing it altogether. Many digital cameras have on board field programmable gate arrays (FPGAs) for digital signal processing and camera functions. FPGAs perform the calculations behind many digital camera functions, as well as additional ones such as color interpolation for mosaic filters and simple image processing (in the case of smart cameras). Camera firmware encompasses the FPGA and on board memory; firmware updates are occasionally available for cameras, adding and improving features. The on board memory in digital cameras allows for storage of settings, look up tables, buffering for high transfer rates, and multi-camera networking with Ethernet switches.

Some of the most common digital camera settings are gain, gamma, area of interest, binning/sub-sampling, pixel clock, offset, and triggering. Understanding these basic settings will help to achieve the best results for a range of applications.

1. Gain

Gain is a digital camera setting that controls the amplification of the signal from the camera sensor. It should be noted that this amplifies the whole signal, including any associated background noise. Most cameras have automatic gain, or auto-gain, which is abbreviated as AGC. Some allow the user to turn it off or set it manually. Gain can be before or after the analog to digital converter (ADC). However, it is important to note that gain after the ADC is not true gain, but rather digital gain. Digital gain uses a look up table to map the digital values to other values, losing some information in the process. Gain before the ADC can be useful for taking full advantage of the bit depth of the camera in low light conditions, although it is almost always the case that careful lighting is more desirable. Gain can also be used to ensure that the taps of multi tap sensors are well matched. In general, gain should be used only after optimizing the exposure setting, and then only after exposure time is set to its maximum for a given frame rate. To visually see the improvement gain can make in an image, compare Figures 6.22a, 6.22b, 6.23a, and 6.23b.

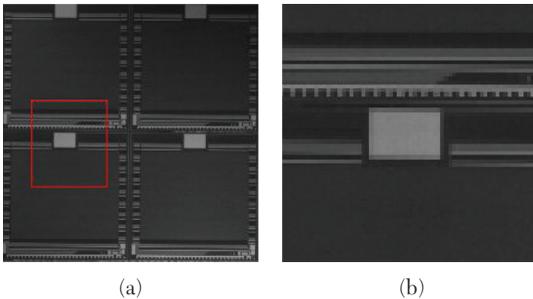


FIGURE 6.22. (a) Real-world image without gain AGC = 0, gamma = 1, 8MHz pixel clock, and 0.2ms exposure; (b) Close-up of image with AGC = 0, gamma = 1, 8Hz pixel clock, and 0.2ms exposure.

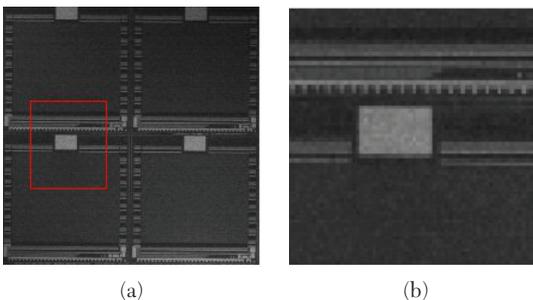


FIGURE 6.23. (a) Real-world image with high gain AGC = 100, gamma = 1, 8MHz pixel clock, and 3.4ms exposure; (b) Close-up of image with AGC = 100, gamma = 1, 8MHz pixel clock, and 3.4ms exposure.

2. Gamma

Gamma is a digital camera setting that controls the grayscale reproduced on the image. An image gamma of unity (Figures 6.24a and 6.24b) indicates that the camera sensor is precisely reproducing the object grayscale (linear response). A gamma setting much greater than unity results in a silhouetted image in black and white (Figures 6.25a and 6.25b). In Figure 6.25b, notice the decreased contrast compared to Figure 6.24b. Gamma can be thought of as the ability to stretch one side (either black or white) of the dynamic range of the pixel. This control is often used in signal processing to raise the signal to noise ratio (SNR).

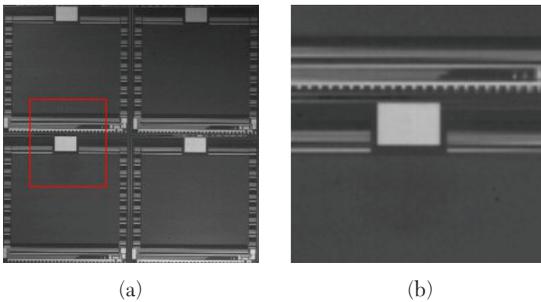


FIGURE 6.24. (a) Real-world image with gamma equal to unity (gamma = 1), 10MHz pixel clock, and 5ms exposure. (b) Close-up of image with gamma = 1, 10MHz pixel clock, and 5ms exposure.

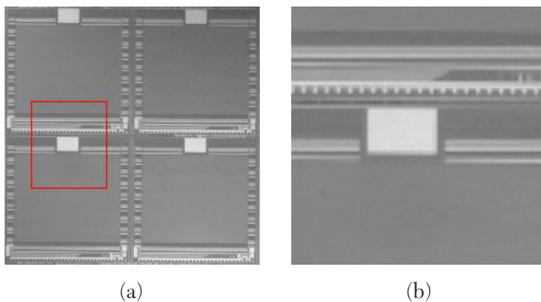


FIGURE 6.25. (a) Real-world image with gamma greater than unity (gamma = 2), 10MHz pixel clock, and 5ms exposure. (b) Close-up of image with gamma = 2, 10MH pixel clock, and 5ms exposure.

3. Area of Interest

Area of interest is a digital camera setting, either through software or on a board, which allows for a subset of the camera sensor array to be read out for each field. This is useful for reducing the field of view (FOV) or resolution to the lowest required rate in order to decrease the amount of data transferred, thereby increasing the possible frame rate. The full

resolution, in terms of Nyquist frequency or spatial sampling frequency, can be retained for this subset of the overall field. For example, a square field of 494×494 may contain all of the useful information for a given frame and can be used so as to not waste bandwidth.

4. Binning/Subsampling

With binning or subsampling, the entire FOV is desired, but the full camera resolution may not be required. In this case, the gray value of adjacent pixels can be averaged together to form larger effective pixels, or only every other pixel read out. It is shown in Figure 6.26. Binning or subsampling increases speed by decreasing the amount of data transferred. Binning is specific to CCD sensors, where the charge from adjacent pixels are physically added together, increasing the effective exposure and sensitivity. Subsampling generally refers to CMOS sensors, where binning is not strictly possible; subsampling offers no increase in exposure or sensitivity. Subsampling can also be used with CCD sensors in lieu of binning when low resolution and high transfer rates are desired without the desire for the original exposure.

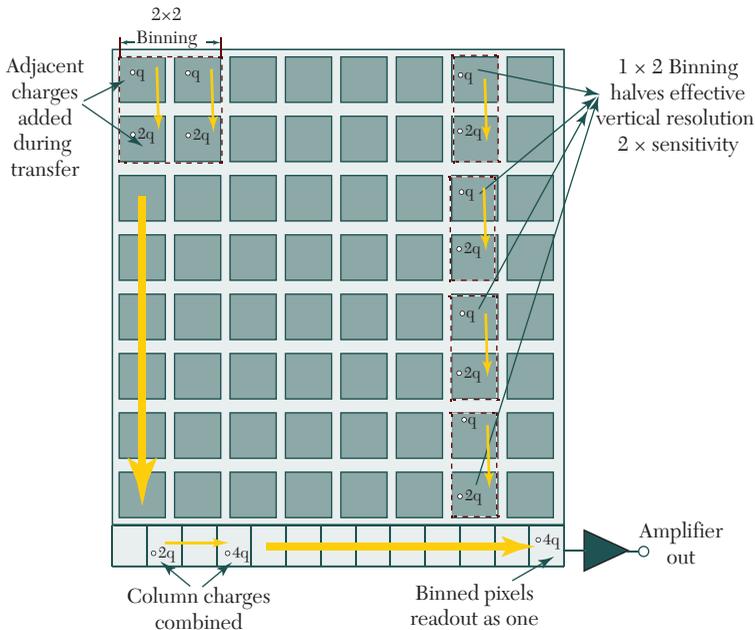


FIGURE 6.26. Illustration of camera pixel binning or subsampling.

5. Pixel Clock

In a CCD camera sensor, the pixel clock describes the speed of the complementary signals which are used to move the charge packets through the shift registers toward the read-out amplifiers. This determines how long it takes to read out the entire sensor, but it is also limited by noise and spillover issues which occur when the packets are transferred too quickly. For example, two cameras with identical sensors may use different pixel clock rates, leading to different performances in saturation capacity (linear range) and frame rate. This setting is not readily user adjustable, as it is generally set to an optimal value specific to the sensor and FPGA capabilities. Over-clocking a sensor by increasing the pixel clock can also lead to thermal issues.

6. Offset

Offset refers to the DC component of a video or image signal, and effectively sets the black level of the image. The black level is the pixel level (in electrons, or volts) which corresponds to a pixel value of zero. This is often used with a histogram to ensure the full use of the camera bit depth, effectively raising the signal to noise. Pushing nonblack pixels to zero lightens the image, although it gives no improvement in the data. By increasing the black level, offset is used as a simple embedded vision image processing technique for brightening and effectively creating a threshold (setting all pixels below a certain value to zero to highlight features) for blob detection.

7. Triggering

Depending upon the application, it can be useful to expose or activate pixels only when an event of interest occurs. In this case, the user can use the digital camera setting of trigger to make the camera acquire images only when a command is given. This can be used to synchronize image capture with a strobe light source, or take an image when an object passes a certain point or activates a proximity switch, the latter being useful in situations where images are being stored for review at a later time. Trigger can also be used on occasions when a user needs to take a sequence of images in a non-periodic fashion, such as with a constant frame rate.

Triggering can be done through hardware or software. Hardware triggers are ideal for high-precision applications, where the latency intrinsic to a software trigger is unacceptable (which can be many milliseconds).

Software triggers are often easier to implement because they take the form of a computer command sent through the normal communication path. An example of a software trigger is the snap function in image viewing software.

Though a host of additional digital camera settings exist, it is important to understand the basics of gain, gamma, and area of interest, binning/subsampling, pixel clock, offset, and trigger. These functions lay the groundwork for advanced image processing techniques that require knowledge of the aforementioned basic settings.

Camera Resolution for Improved Imaging System Performance

Camera resolution and contrast play an integral role in both the optics and electronics of an imaging system. Though camera resolution and contrast may seem like optical parameters, pixel count and size, TV lines, camera modulation transfer function (MTF), Nyquist limit, pixel depth/grayscale, dynamic range, and SNR contribute to the quality of what a user is trying to image. With tech tips for each important parameter, imaging users from novice to expert can learn about camera resolution as it pertains to the imaging electronics of a system.

1. Pixel count and pixel size

The more pixels within a field of view (FOV), the better the resolution. However, a large number of pixels requires either a larger sensor or smaller sized individual pixels. *Using a larger sensor to achieve more pixels means the imaging lens magnification and/or or field of view will change.* Conversely, if smaller pixels are used, the imaging lens may not be able to hold the resolution of the system due to the finite spatial frequency response of optics, primarily caused by design issues or the diffraction limit of the aperture. The number of pixels also affects the frame rate of the camera. For example, each pixel has 8 bits of information that must be transferred in the reconstruction of the image. *The more pixels on a sensor, the higher the camera resolution but the lower the frame rate.* If both high frame rates and high resolution (e.g., many pixels) are required, then the system price and set-up complexity quickly increases, often at a rate not necessarily proportional to the number of pixels.

2. TV lines

In analog CCD cameras, the TV line (TVL) specification is often used to evaluate resolution. The TVL specification is a unit of resolution based

on a bar target with equally spaced lines. If the target is extended so that it covers the FOV, the TVL number is calculated by summing all of the resulting lines and spaces. Equations 1 and 2 provide simple calculations for determining horizontal (H) and vertical (V) TVL. Included in Equation 1 is a normalization factor necessary to account for a sensor's 4:3 aspect ratio.

$$HTVL = \frac{2[H \text{ Resolution (lines per mm)}][H \text{ Sensing Distance (mm)}]}{1.333} \quad (1)$$

$$VTVL = 2[V \text{ Resolution (lines per mm)}][V \text{ Sensing distance (mm)}] \quad (2)$$

3. Modulation transfer function (MTF)

The most effective means of specifying the resolution of a camera is its modulation transfer function (MTF). The MTF is a way of incorporating contrast and resolution to determine the total performance of a sensor. A useful property of the MTF is the multiplicative property of transfer functions; the MTF of each component (imaging lens, camera sensor, display, etc.) can be multiplied to get the overall system response. The MTF takes into account not only the spatial resolution of the number of pixels/mm, but also the roll off that occurs at high spatial frequencies due to pixel cross talk and finite fill factors. *It is not the case that a sensor will offer 100% contrast at a spatial frequency equal to the inverse of its pixel size.*

4. Nyquist limit

The absolute limiting resolution of a sensor is determined by its Nyquist limit. For example, the Sony ICX285 is a monochrome CCD sensor with a horizontal active area of 9mm containing 1392 horizontal pixels each $6.45\mu\text{m}$ in size. This represents a horizontal sampling frequency of 155 pixels/mm ($1392 \text{ pixels} / 9\text{mm} = 1\text{mm} / 0.00645 \text{ mm/pixel} = 155$). The Nyquist limit of this calculates to 77.5 lp/mm. At the Nyquist limit, contrast is phase dependent for a constant incident square wave (imagine one pixel on, one pixel off, or each pixel with half a cycle). It is, therefore, common to include the Kell factor (~ 0.7), which reflects the deviation of the actual frequency response from the Nyquist frequency. Most importantly, the Kell factor compensates for the space between pixels. *Sampling at spatial frequencies above the system's Nyquist limit can create spurious signals and aliasing effects that are undesirable and unavoidable. Nyquist Limit given by following equation.*

$$\text{Nyquist limit(lp per mm)} = \frac{1}{2} [\text{Kell factor}] [\text{sampling frequency(pixels per mm)}]$$

5. Pixel depth/ Grayscale

Often referred to as grayscale is the dynamic range of a CCD camera; pixel depth represents the number of steps of gray in the image. Pixel depth is closely related to the minimum amount of contrast detectable by a sensor. In analog cameras, the signal is a time varying voltage proportional to the intensity of the light incident on the sensor, specified below the saturation point. After digitizing, this continuous voltage is effectively divided into discrete levels, each of which corresponds to a numerical value. At unity gain, light that has 100% saturation of the pixel will be given a value of $2^N - 1$, where N is the number of bits, and the absence of light is given a value of 0. *The more bits in a camera, the smoother the digitization process.* Also, more bits means higher accuracy and more information. With enough bits, the human eye can no longer determine the difference between a continuous grayscale and its digital representation. The number of bits used in digitization is called the bit depth or pixel depth.

For an example of pixel depth, consider the Sony XC series of cameras, which offer 256 shades of gray, and the Edmund Optics USB 2.0 CMOS series of cameras, which are available in 8 bit (256 grayscale) and 10 bit (1024 grayscales) models. Generally, 12 bit and 14 bit cameras have the option of running in a lower pixel depth mode. Although pixel depths above 8 bits are useful for signal processing, computer displays only offer 8-bit resolution. Thus, if the images from the camera will be viewed only on a monitor, the additional data does nothing but reduce frame rate. Figure 6.27 illustrates different pixel depths. Notice the smooth progression from gray to white as bit depth increases.



FIGURE 6.27. Illustration of 2-bit (top), 4-bit (middle), and 8-bit (bottom) grayscales.

6. Dynamic range

Dynamic range is the difference between the lowest detectable light level and the highest detectable light level. Physically, this is determined by the saturation capacity of each pixel, the dark current or dark noise, the ADC circuits, and gain settings. *For high-dynamic ranges, more bits are required to describe the grayscale in a meaningful fashion.* However, it is important to note that, with consideration of the signal to noise ratio, using 14 bits to describe a 50dB dynamic range gives redundant bits and no additional information.

7. Signal to noise ratio (SNR)

The signal to noise ratio (SNR) is closely linked to the dynamic range of a camera. *A higher SNR yields a higher possible number of steps in the grayscale (higher contrast) produced by a camera.* The SNR is expressed in terms of decibels (dB) in analog systems and bits in digital systems. In general, 6dB of analog SNR converts to 1 bit when digitized. For digital or analog cameras, X bits (or the equivalent in analog systems) correspond to 2^x grayscales (i.e. 8 bit cameras have 2^8 or 256 gray levels). There are two primary sources for the noise in camera sensors. The first is imperfections in the chip, which result in non-uniform dark current and crosstalk. The second is thermal noise and other electronic variations. Chip imperfections and electronic variations reduce camera resolution and should be monitored to determine how to best compensate for them within the imaging system.

The basics of camera resolution can be divided into parameters of pixel count and size, TV lines, camera MTF, Nyquist limit, pixel depth/grayscale, dynamic range, and SNR. Understanding these basic terms allows a user to move from being a novice to an imaging expert.

6.3 ZOOMING, CAMERA INTERFACE, AND SELECTION

Zooming refers to increase the quantity of pixels, so that when you zoom an image, you will see more detail. The increase in the quantity of pixels is done through oversampling. The one way to zoom is, or to increase samples, is to zoom optically, through the motor movement of the lens and then capture the image. But we have to do it, once the image has been captured. There is a difference between zooming and sampling. The concept is same, which is, to increase samples. But the key difference is that while sampling is done on the signals, zooming is done on the digital

image. Zooming simply means enlarging a picture in a sense that the details in the image became more visible and clear.

Zooming an image has many wide applications ranging from zooming through a camera lens, to zoom an image on Internet, and so on. One can zoom something at two different steps. The first step includes zooming before taking a particular image. This is known as preprocessing zoom. This zoom involves hardware and mechanical movement. The second step is to zoom once an image has been captured. It is done through many different algorithms in which we manipulate pixels to zoom in the required portion. Optical zoom and digital zoom are two types of zoom supported by the cameras.

Optical Zoom

The optical zoom is achieved using the movement of the lens of the camera. An optical zoom is actually a true zoom. The result of the optical zoom is far better than that of digital zoom. In optical zoom, an image is magnified by the lens in such a way that the objects in the image appear to be closer to the camera. In optical zoom, the lens is physically extend to zoom or magnify an object.

Digital Zoom

Digital zoom is basically image processing within a camera. During a digital zoom, the center of the image is magnified and the edges of the picture got crop out. Due to magnified center, it looks like that the object is closer to you. During a digital zoom, the pixels expand, and the quality of the image is compromised. The same effect of digital zoom can be seen after the image is taken through your computer by using an image processing toolbox/software, such as Photoshop. The methods of zooming are (1) pixel replication or (nearest neighbor interpolation), (2) zero-order hold method, and (3) zooming K times.

1. Pixel replication zooming method

Pixel replication zooming method is also known as nearest neighbor interpolation. As its name suggests, in this method it just replicates the neighboring pixels. Zooming is nothing but an increased amount of sample or pixels. In this method it creates new pixels form the already given pixels. Each pixel is replicated in this method n times row-wise and column-wise and gives a zoomed image. It's as simple as that. If an image of 2 rows and 2 columns needed to zoom twice using pixel replication, here's how it can

be done. For a better understanding, the image has been taken in the form of a matrix with the pixel values of the image. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ The image has two rows and two columns, first zoom it row-wise.

Row-wise zooming when zoomed row-wise, simply copy the row's pixels to its adjacent new cell. Here's how it would be done: $\begin{bmatrix} 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \end{bmatrix}$. In the matrix, each pixel is replicated twice in the rows.

Column-wise zooming

The next step is to replicate each of the pixels column-wise, so it will simply copy the column pixel to its adjacent new column or simply below it. Here's how it would be done:

$$\begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{bmatrix}$$

Now image size

As it can be seen from the above example, an original image of 2 rows and 2 columns has been converted into 4 rows and 4 columns after zooming. That means the new image has dimensions of (original image rows * zooming factor, original image cols * zooming factor). One of the advantages of this zooming technique is that it is very simple. Just copy the pixels and nothing else. The disadvantage of this technique is that image got zoomed, but the output is very blurry. And as the zooming factor increased, the image got more and more blurred. That would eventually result in fully blurred image.

2. Zero-order hold zooming method

Zero-order hold method is another method of zooming. It is also known as zoom twice because it can only zoom twice. In zero-order hold method, two adjacent elements are chosen from the rows respectively, then added, and the result divided by two. The result is placed in between those two elements. First do this row-wise and then do it column-wise. For example, take an image of the dimensions of 2 rows and 2 columns and zoom twice using zero-order hold. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. First zoom row-wise and then column-wise.

Row-wise zooming is $\begin{bmatrix} 1 & 1 & 2 \\ 3 & 3 & 4 \end{bmatrix}$. As we take the first two numbers: $(2 + 1) = 3$ and then divide it by 2, we get 1.5 which is approximated to 1. The same method is applied in the row 2.

Column-wise zooming is $\begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 3 & 3 & 4 \end{bmatrix}$. Take two adjacent column pixel values which are 1 and 3. Add them to get 4. 4 is then divided by 2, and the answer is 2 which is placed in between them. The same method is applied in all the columns.

New image size

The dimensions of the new image are 3×3 , whereas the original image dimensions are 2×2 . So it means that the dimensions of the new image are based on the following formula $(2(\text{number of rows}) - 1) \times (2(\text{number of columns}) - 1)$. One of the advantages of this zooming technique is that it does not create as blurry a picture as compared to the nearest neighbor interpolation method. But it also has a disadvantage of that it can only run on the power of 2.

Reason behind twice zooming

Consider the above image of 2 rows and 2 columns. If we have to zoom it 6 times, using zero-order hold method, we cannot do it as the formula shows us this. You can only zoom in the power of 2 such as 2, 4, 8, 16, 32, and so on. Even if you try to zoom, you cannot. Because when you first zoom it two times the result would be same as shown in the column-wise zooming with dimensions equal to 3×3 . If zoomed again, you will get dimensions equal to 5×5 . Now if you do it again, you will get dimensions equal to 9×9 , whereas according to the formula the answer should be 11×11 , as $(6(2) - 1) \times (6(2) - 1)$ gives 11×11 .

3. K-times zooming method

K times is the third zooming method. It is one of the most perfect zooming algorithms. It caters to the challenges of both twice zooming and pixel replication. K in this zooming algorithm stands for zooming factor. First take two adjacent pixels as you did in the zooming twice. Then subtract the smaller from the greater one. Call this output (OP). Divide the output (OP) with the zooming factor (K). Now add the result to the smaller value

and put the result in between those two values. Add the value OP again to the value you just put and place it again next to the previous putted value. You have to do it until you place $k-1$ values in it. Repeat the same step for all the rows and the columns, and you get zoomed images. For example: Suppose you have an image of 2 rows and 3 columns, as shown below. And you have to zoom it thrice or three times. $\begin{bmatrix} 15 & 30 & 15 \\ 30 & 15 & 30 \end{bmatrix}$ K in this case is 3.

$K = 3$. The number of values that should be inserted is $k - 1 = 3 - 1 = 2$.

Row-wise zooming

Take the first two adjacent pixels, which are 15 and 30. Subtract 15 from 30. $30 - 15 = 15$. Divide 15 by k . $15/k = 15/3 = 5$. Call it OP, where OP is just a name. Add OP to lower number. $15 + OP = 15 + 5 = 20$. Add OP to 20 again. $20 + OP = 20 + 5 = 25$. Do this two times because we have to insert $k - 1$ values. Now repeat this step for the next two adjacent pixels. It is shown in the first table. After inserting the values, you have to sort the inserted values in ascending order, so there remains a symmetry between them. It is shown in the second table.

$$\text{Table 1. } \begin{bmatrix} 15 & 20 & 25 & 30 & 20 & 25 & 15 \\ 30 & 20 & 25 & 15 & 20 & 25 & 30 \end{bmatrix}$$

$$\text{Table 2. } \begin{bmatrix} 15 & 20 & 25 & 30 & 25 & 20 & 15 \\ 30 & 25 & 20 & 15 & 20 & 25 & 30 \end{bmatrix}$$

Column-wise zooming

The same procedure has to be performed column-wise. The procedure include taking the two adjacent pixel values, and then subtracting the smaller from the bigger one. Then after that, divide it by k . Store the result as OP. Add OP to smaller one, and then again add OP to the value that comes in first addition of OP. Insert the new values. Here is the result,

$$\begin{bmatrix} 15 & 20 & 25 & 30 & 25 & 20 & 15 \\ 20 & 21 & 21 & 25 & 21 & 21 & 20 \end{bmatrix}$$

New image size

The best way to calculate the formula for the dimensions of a new image is to compare the dimensions of the original image and the final image. The dimensions of the original image were 2×3 . And the dimensions of the new image are 4×7 . The formula is: $(K (\text{number of rows minus } 1) + 1) \times (K (\text{number of cols minus } 1) + 1)$. The one of the clear advantage that k time

zooming algorithm has that it is able to compute zoom of any factor which was the power of pixel replication algorithm, also it gives improved result (less blurry) which was the power of zero order hold method. So hence, it comprises the power of the two algorithms. The only difficulty in this algorithm is that it has to be sort in the end, which is an additional step, and thus increases the cost of computation.

Spatial Resolution

Spatial resolution states that the clarity of an image cannot be determined by the pixel resolution. The number of pixels in an image does not matter. Spatial resolution can be defined as the smallest discernible detail in an image. Another way we can define spatial resolution is as the number of independent pixels values per inch. In short what spatial resolution refers to is that we cannot compare two different types of images to see that which one is clear or which one is not? If we have to compare the two images to see which one is clearer or which has more spatial resolution, we have to make the comparison using two images of the same size. For example, we cannot compare the two images in Figure 6.28 see the clarity of the image.



FIGURE 6.28. Image and zoomed image.

The picture on the left is zoomed out with dimensions of 227×222 . Whereas the picture on the right is also a zoomed image and has the dimensions of 980×749 . Remember the factor of zoom does not matter in this condition, the only thing that matters is that these two pictures are not equal. In order to measure spatial resolution, the pictures below in Figure 6.29 would serve the purpose.

Compare these two pictures. Both of the pictures have the same dimensions of 227×222 . When you compare them, you will see that the picture on the left has more spatial resolution or is clearer than the picture on the right. That is because the picture on the right is a blurred image. Since the spatial resolution refers to clarity, for different devices different



FIGURE 6.29. Spatial resolution comparison.

units of measurement have been made to measure it. For example: Dots per inch, Lines per inch and Pixels per inch. Dots per inch or DPI is usually used in monitors. Lines per inch or LPI is usually used in laser printers. Pixel per inch or PPI is measure for different devices such as tablets, mobile phones, etc.

Pixel density or pixels per inch is a measure of spatial resolution for different devices that includes tablets and mobile phones. The higher the PPI, the higher the quality. The Samsung Galaxy S4 has PPI or pixel density of 441. First, we will use the Pythagoras theorem to calculate the diagonal resolution in pixels. It can be given as: $c = \sqrt{a^2 + b^2}$. Where a and b are the height and width resolutions in pixel and c is the diagonal resolution in pixels. For the Samsung Galaxy S4, it is 1080 x 1920 pixels. So putting those values in the equation gives the result of $c = 2202.90717$. Now we will calculate PPI:- $PPI = c / \text{diagonal size in inches}$; The diagonal size in inches of Samsung Galaxy S4 is 5.0 inches. $PPI = 2202.90717 / 5.0$; $PPI = 440.58$; $PPI = 441$ (approx); that means that the pixel density of Samsung Galaxy S4 is 441 PPI.

The dpi or dots per inch is often relate to PPI, whereas there is a difference between the two. DPI or dots per inch is a measure of spatial resolution of printers. In the case of printers, dpi means how many dots of ink are printed per inch when an image is printed out from the printer. Remember, it is not necessary that each pixel per inch be printed by one dot per inch. There may be many dots per inch used for printing one pixel. The reason behind this is that most of the color printers use the CMYK model. The colors are limited. The printer has to choose from these colors to make the color of the pixel, whereas within a PC, there are hundreds of thousands of colors. The higher the dpi of the printer, the higher the quality of the printed document or image on paper. Usually some laser printers have dpi of 300 and some have 600 or more.

When dpi refers to dots per inch, liner per inch refers to lines of dots per inch. The resolution of a halftone screen is measured in lines per inch.

Gray-level Resolution

Gray-level resolution refers to the predictable or deterministic change in the shades or levels of gray in an image. In short gray-level resolution is equal to the number of bits per pixel. The number of different colors in an image is depends on the depth of color or bits per pixel. The mathematical relation that can be established between gray-level resolution and bits per pixel can be given as, $L = 2^k$. In this equation, L refers to number of gray levels. It can also be defined as the shades of gray. And k refers to bpp or bits per pixel. So the 2 raised to the power of bits per pixel is equal to the gray level resolution. For example: An image with 8 bits per pixel or 8 bpp have the gray-level resolution as follows, $L = 2^k$, Where $k = 8$; $L = 2^8 = 256$. Gray-level resolution of the image is 256. This image has 256 different shades of gray.

The more bits per pixel of an image, the more its gray-level resolution. Gray-level resolution should only be defined in terms of levels. It is also defined in terms of bits per pixel. If we were to find the bits per pixel, or in this case K, $K = \log \text{base } 2(L)$. For example: If you are given an image of 256 levels. The bits per pixel required for it is, $K = \log \text{base } 2(256)$; $K = 8$. So the answer is 8 bits per pixel. The image has been distorted badly by reducing the gray levels to 2 gray levels (as shown in Figure 6.30).

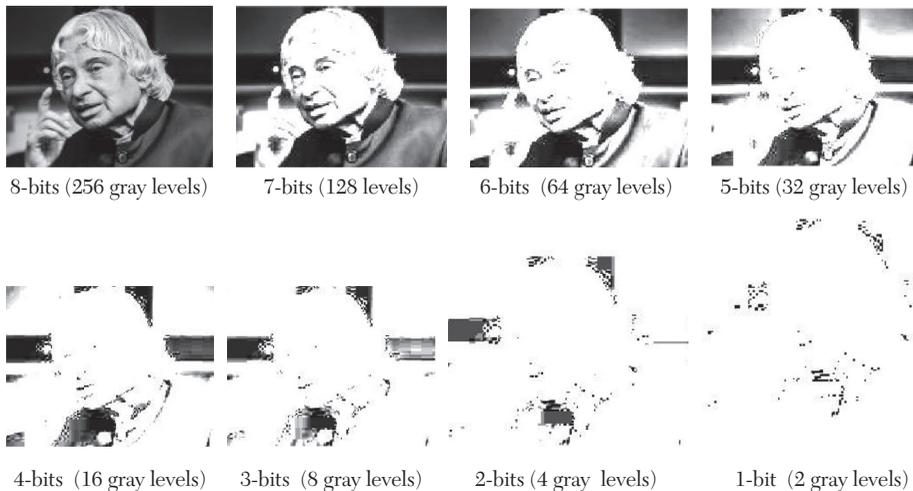


FIGURE 6.30. Gray-level resolution.

Contouring

As we reduce the number of gray levels, there is a special type of effect start appearing in the image, which can be seen clear in 32 gray-level picture. This effect is known as contouring. As we decrease the number of gray levels in an image, some false colors or edges start appearing. Consider an image of 8 bpp (a grayscale image) with 256 different shades of gray or gray levels. Figure 6.30 has 256 different shades of gray. Now when we reduce it to 128 and further reduce it 64, the image is more or less the same. But when we reduce it further to 32 different levels, you will find that the contour effects start appearing on the image. These effects are more visible when we reduce it further to 16 levels. The lines that start appearing on this image are known as contouring and are very visible. The effect of contouring increases as we reduce the number of gray levels, and the effect decreases as we increase the number of gray levels. They are both vice versa. That means more quantization will result in more contouring and vice versa.

Concept of dithering

Reducing the gray level of an image reduces the number of colors required to denote an image. If the gray levels are reduced to 2, the image that appears does not have much spatial resolution or is not very much appealing. Dithering is the process by which we create illusions of the color that are not present as shown in Figure 6.31(b). It is done by the random arrangement of pixels. When we perform quantization, to the last level, we see that the image that comes in the last level (level 2) is not very clear. Now if we were to change this image into some image that gives more detail than this, we have to perform dithering. First of all, dithering is usually working to improve thresholding. During thresholding, the sharp edges appear where gradients are smooth in an image. In thresholding, we simply choose a constant value. All the pixels above that value are considered as 1 and all the values below it are considered as 0.

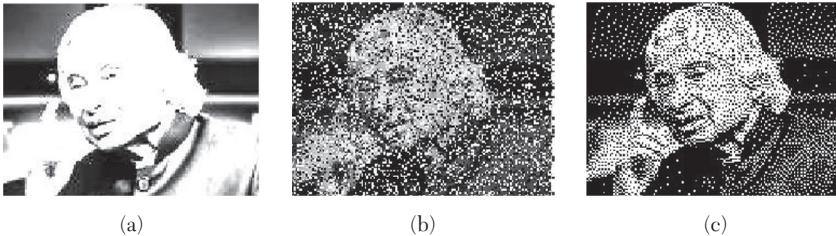


FIGURE 6.31. (a) Gray image (b) Dithering (c) Dithering and thresholding.

Since there is not much change in image (b), as the values are already 0 and 1 or black and white in this image. Now we perform some random dithering to it by randomly arranging the pixels. We now have an image that shows slightly more details, but its contrast is very low. So we do some more dithering to increase the contrast. Now we mix the concepts of random dithering along with threshold. We got all these images by just rearranging the pixels of an image. This rearranging could be random or could be deliberate to get the clear image as shown in Figure 6.31(c).

Digital camera interfaces

Computer-based interfaces such as USB 3 and Gigabit Ethernet PCI-based expansion slots can be used to support numerous types of analog and digital camera interfaces. The embedded controllers, for example, the EOS-4000 and the EOS-1200 support camera link interfaces. Both based on the Intel i7 processor. The EOS-4000 supports two independent camera link base PoCL ports and pixel clock rates to 85 MHz and two independent RS-232 serial communication ports, 64 isolated digital I/O connectors, and an internal USB ports (Figure 6.32a). Four independent PoE gigabit Ethernet ports are supported on the EOS-1200 and IEEE 1588 for synchronizing multiple cameras.

Digital cameras have gained in popularity over the past decade because transmission noise, distortion, or other signal degradations do not affect the information being transmitted. Since the output signal is digital, there is little information lost in the transmission process. As more and more users turn to digital cameras, imaging technology has also advanced to include a multitude of digital interfaces. The imaging landscape will be very different in another decade, but the most common interfaces available today are capture boards, FireWire, Camera Link, GigE, and USB. Table 6.6 compares all interfaces available today.

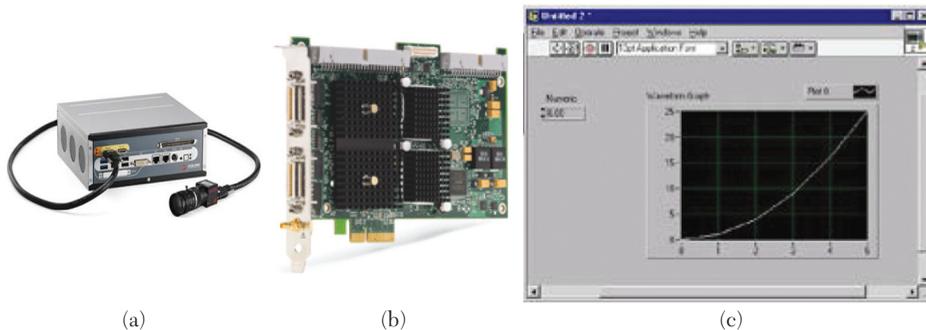
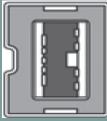


FIGURE 6.32. (a) Embedded controller for camera interface (b) capture board (c) NI LabView software.

TABLE 6.6. Comparison of Popular Digital Camera Interfaces

Digital Signal Options	FireWire 1394.b 	Camera Link 	USB 2.0 	USB 3.0 	GigE 
Data Transfer Rate:	800 Mb/s	3.6 Gb/s (full configuration)	480 Mb/s	5Gb/s	1000 Mb/s
Max Cable Length:	100m (with GOF cable)	10m	5m	3m (recommended)	100m
# Devices:	Up to 63	1	Up to 127	Up to 127	Unlimited
Connector:	9pin-9pin	26pin	USB	USB	RJ45/Cat5e or 6
Capture Board:	Optional	Required	Optional	Optional	Not Required
Power:	Optional	Required	Optional	Optional	Required (Optional with PoE)

As with many of the criteria for camera selection, there is no single best option interface, but rather one must select the most appropriate device for the application at hand. Asynchronous or deterministic transmission allows for data transfer receipts, guaranteeing signal integrity, placing delivery over timing due to the two-way communication. In isochronous transmission, scheduled packet transfers occur (e.g., every $125\mu\text{s}$), guaranteeing timing but allowing for the possibility of dropping packets at high transfer rates.

Capture Boards

Image processing typically involves the use of computers. Capture boards allow users to output analog camera signals into a computer for analysis; or an analog signal (NTSC, YC, PAL, CCIR), the capture board contains an analog to digital converter (ADC) to digitize the signal for image processing. Others enable real-time viewing of the signal. Users can then capture images and save them for future manipulation and printing. Basic capturing software is included with capture boards, allowing users to save, open, and view images. The term capture board also refers to PCI cards that are necessary to acquire and interpret the data from digital camera interfaces, but are not based on standard computer connectors.

FireWire IEEE 1394/IIDC DCAM standard

FireWire, IEEE 1394, is a popular serial, isochronous camera interface due to the widespread availability of FireWire ports on computers. However, FireWire.a is one of the slower transfer rate interfaces. Both FireWire.a and FireWire.b allow for the connection of multiple cameras and provide power through the FireWire cable. Hot swap/hot plugging is not recommended, as the connector's design may cause power pin shorting to signal pins, potentially damaging the port or the device.

Camera Link

Camera Link is a high-speed serial interface standard developed explicitly for embedded vision applications, most notably those that involve automated inspection and process control. A camera link capture card is required for usage, and power must be supplied separately to the camera. Special cabling is required because, in addition to low voltage differential pair LVDP signal lines, separate asynchronous serial communication channels are provided to retain full bandwidth for data transmission. The single cable base configuration allows 255 MB/s transfer dedicated for video. Dual outputs (full configuration) allow for separate camera parameter send/

receive lines to free up more data transfer space (680 MB/s) in extreme high speed applications. Camera Link HS (High Speed) is an extension to the Camera Link interface that allows for much higher speed (up to 2100MB/s at 15m) by using more cables. Additionally, Camera Link HS incorporates support for fiber optic cables with lengths of up to approximately 300m.

GigE Vision Standard

GigE is based on the gigabit Ethernet Internet protocol and uses standard Cat-5 and Cat-6 cables for a high speed camera interface. Standard Ethernet hardware such as switches, hubs, and repeaters can be used for multiple cameras, although overall bandwidth must be taken into consideration whenever non peer to peer (direct camera to card) connections are used. In GigE Vision, camera control registers are based on the EMVA GenICam standard. Optional on some cameras, Link Aggregation (LAG, IEEE 802.3ad) uses multiple Ethernet ports in parallel to increase data transfer rates, and multicasting to distribute processor load. Supported by some cameras, the network precision time protocol (PTP) can be used to synchronize the clocks of multiple cameras connected on the same network, allowing for a fixed-delay relationship between their associated exposures. Devices are hot swappable.

USB—Universal Serial Bus

USB 2.0 is a popular interface due to its ubiquity among computers. It is not high speed, but it is convenient; maximum attainable speed depends upon the number of USB peripheral components, as the transfer rate of the bus is fixed at 480Mb/s total. Cables are readily available in any computer store. In some cases, as with laptop computers, it may be necessary to apply power to the camera separately. USB 3.0 features the plug and play benefits of USB 2.0, while allowing for much higher data transmission rates.

CoaXPress

CoaXPress is a single-cable high bandwidth serial interface that allows for up to 6.25Gb/s transmission rates with cable lengths up to 100m. Multiple cables can be used for speeds of up to 25Gb/s. Much like PoE, power-over-coax is an available option, as well.

Many camera interfaces allow for power to be supplied to the camera remotely over the signal cable. When this is not the case, power is commonly supplied either through a Hirose connector (which also allows for trigger wiring and I/O), or a standard AC/DC adapter-type connection.

Even in cases where the camera can be powered by card or port, using the optional power connection may be advantageous. For example, daisy chaining FireWire cameras or running a system from a laptop are ideal cases for additional power. Also, cameras that have large, high-speed sensors and on board FPGAs require more power than can be sourced through the signal cable.

Currently, power injectors are available that allow, with particular cameras, the ability to deliver power to the camera over the GigE cable. This can be important when space restrictions do not allow for the camera to have its own power supply, as in factory floor installations or outdoor applications. In this case, the injector is added somewhere along the cable line with standard cables running to the camera and computer. However, not all GigE cameras are power over Ethernet (PoE) compatible. As with other interfaces, if peak performance is necessary, the power should be supplied separately from the signal cable. In PoE, the supply voltage is based on a standard that uses a higher voltage than standard cameras can supply; this necessitates more electronics and causes more power dissipation which requires sophisticated thermal design to avoid an increase in thermal noise and thus loss of image quality.

Although many digital camera interfaces are accessible to laptop computers, it is highly recommended to avoid standard laptops for high-quality and/or high-speed imaging applications. Often, the data busses on the laptop will not support full transfer speeds and the resources are not available to take full advantage of high performance cameras and software. In particular, the Ethernet cards standard in most laptops perform at a much lower level than the PCIe cards available for desktop computers.

Camera Software

In general, there are two choices when it comes to imaging software: camera-specific software-development kits (SDKs) or third-party software. SDKs include application programming interfaces with code libraries for development of user-defined programs, as well as simple-image viewing and acquisition programs that do not require any coding and offer simple functionality. With third-party software, camera standards (GenICam, DCAM, GigE Vision) are important to ensure functionality. Third-party software includes NI LabVIEW (Figure 6.32c), MATLAB, OpenCV, and the like. Often, third-party software is able to run multiple cameras and support multiple interfaces, but it is ultimately up to the user to ensure functionality.

Though a host of camera types, interfaces, power requirements, and software exist for imaging applications, understanding the pros and cons of each allows the user to pick the best combination for any application. Whether an application requires high-data transfer, long cable lengths, and/or daisy chaining, a camera combination exists to achieve the best results.

Camera and Lens Selection for a Vision Project

Choosing a camera for an embedded vision system is important to determine project requirements. Resolution is the first criteria to when choosing a camera. The classic resolution of a camera is based on pixels; such as 600 pixels x 400 pixels. The other type of resolution is about the spatial resolution. The spatial resolution is how close the pixels are to each other; how many pixels-per-inch (ppi) are on the sensor. The spatial resolution really controls how an image will look.

Focal length is the next requirement. Selecting a focal length is a trade-off between being what zoom level one needs. A larger focal length (such as 200) will be zoomed in, while a smaller focal length (such as 10) will be zoomed out. A focal length of 30–50 is approximately what we see with our eyes. Smaller than that will look larger than life (and is often called a wide-angle lens). An extreme example of a small focal length can be a fish-eye lens that can see around 180° (with a fairly distorted image). If the focal length is specified as a range it is probably an adjustable zoom lens.

The next is the maximum aperture or f number for camera choice. The f number is often specified as f/2.8 or F2.8. The larger the number the less light can enter the aperture. If you need to take images in a low-light condition you will want a small f number to avoid needing external lighting of the scene. The smaller the f number, the shallower the depth of field will be. The depth of field is the part of the image that appears in focus (and sharp).

The programmer will also need to look at the field of view or FOV. The FOV is the angular measurement window that can be seen by the lens. The FOV is usually specified with two numbers, such as 60° x 90°. In this case 60° is the horizontal FOV and 90° is the vertical FOV. Sometimes instead of giving two numbers people will just specify one number based on the diagonal. FOV can be related to the focal length of the lens. FOV is computed by knowing the size of the camera imaging array (i.e., CCD) from the datasheet and the focal length of the lens. So for example, if your lens has an FOV of 60° x 90°, the working distance from the sensor (i.e., where the camera hits the target) is 2 meters away, and the smallest object

that you need to be able to detect (with a minimum of 2 pixels on it) is 0.01meters: So we now know that when we choose a camera it must have a minimum resolution of 460 x 800 to meet the requirement of seeing a 0.01m object from 2m away. Also remember if you require 2 pixels to be on the object in each direction that is a total of 4 (2 x 2) pixels that will be on the full object. Often the camera resolution will be specified as a total such as 2MP (or 2 mega pixels). In that case you can multiply the 460 x 800 to get a total of 368,000 pixels required. So in this case a 2MP camera would be more than sufficient.

With all lenses, and particularly with cheaper lenses, distortion and vignetting can be a big problem. Common distortion modes are where the X or Y axis appears to no longer be straight and perpendicular with each other (rectilinear). Example of that is an image look wavy or bulging. With vignetting the edges, and particularly the corners, become darkened.

The mounting lens and camera system need to be checked. There are many different styles for mounting a lens to a camera. There are different filters that can be attached to lens for things such as polarization and blocking certain parts of the spectrum (such as IR). And the iPhone camera typically has a built in IR filter in the lens to remove IR components from images.

Another necessity is a black and white or color camera for the project. In many cases black and white cameras will be easier for the algorithms to process. However having color can give more channels for humans to look at and for algorithms to learn from. For many applications, for example in intelligent traffic systems, a combination of b/w and color cameras are also frequently used to satisfy the specific national legal requirements for evidence-grade images. In addition to black and white, there are hyper-spectral cameras that see various frequency bands. Some of the common spectral ranges are ultra-violet (UV), near-infrared (NIR), visible near-infrared (VisNIR), and infrared (IR).

Dynamic range of the camera is another requirement. The larger the dynamic range, the greater the light difference that can be handled within an image. If the robot is operating outdoors in sun and shade, you need a high-dynamic range (HDR) camera.

Another camera interface requirement would be both the electrical and software interface. On the electrical side the common camera protocols are camera-link, USB2.0, USB3, FireWire (IEEE1394), and gigabit Ethernet

(GigE). GigE cameras are good since they just plug into an Ethernet port making wiring easy. Many of them can use power-over-Ethernet (PoE) so the power goes over the Ethernet cable and only have one cable to run. The downside of GigE is the latency and nondeterministic nature of Ethernet. The other interfaces might be better if latency is a problem. Also generally one can only send full video over GigE with a wired connection, and not from a wireless connection. The reason for this is that one need to set the Ethernet port to use jumbo packets, which can't be done on a standard wireless connection. Probably the most common reason GigE cameras are not working properly is that the Ethernet port on the computer is not configured for jumbo packets.

For software interface, make sure that camera has an SDK that will be easy to use. If project is using ROS or OpenCV, verify that the drivers are good. Also check in what format will the image be supplied: will it be a file format such as png or jpg? Or raw image format?

For stereo vision, make sure the cameras have a triggering method, so that when the programmer issues an image-capture command, the user gets images from both cameras simultaneously. The triggering can be in the software or the hardware, but hardware is typically better and faster.

For video, the frame rate is typically specified in frames per second (fps) or “line rate” or “line frequency” for line-scan cameras is important. The frame rate describes the number of images that the sensor can capture and transmit per second. The higher the frame rate, the quicker the sensor. The quicker the sensor, the more images it captures per second. The more images, the higher the data volumes. For fast-moving applications like inspections of printed images, with newspapers moving at high speeds past the camera inspection point, the cameras must be able to “shoot” in milliseconds. Some microscopic inspections used in medicine and industry require only low frame rates.

6.4 THERMAL-IMAGING CAMERA

For manufacturing and other industries using heavy electrical and mechanical machines, thermal imaging is a machinery testing for production level and efficiency. Thermal imagers are one of the most valuable diagnostic tools for industrial applications. These identify anomalies that are invisible to the human eye, thus allowing necessary corrective measures to avoid costly system failures. Thermal-imaging cameras record the intensity of infrared

wave radiation (part of EM waves) and convert it into a visible image. This helps users to find out when and where maintenance is required. Electrical and mechanical installations tend to heat up before failure. By identifying those hot spots with a thermal-imaging camera, preventive actions can be taken. This can avoid costly production breakdowns or, even worse, fire. Thermal-imaging cameras can scan temperature distribution throughout the entire surface of machinery and electrical equipment without requiring contact. This is shown in Figures 6.33 and 6.34.

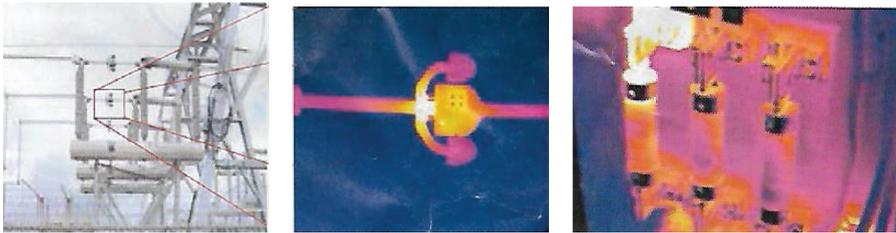


FIGURE 6.33. High-power transmission line inspection for overheating by thermal-imaging cameras.

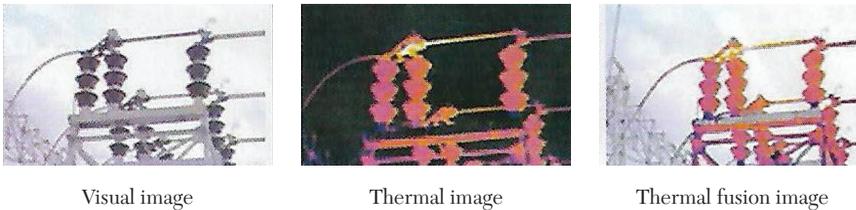


FIGURE 6.34. Inspection of a substation revealing overheated components.

Thermal imagers find application in almost every industry including defense, manufacturing, electronics, electrical, mobile, automotive, and petrochemical. Applications for electrical systems can be divided into two categories: high-voltage installations and low-voltage installations. Thermal-imaging cameras can be used for inspection of high-voltage substations, switchgear, transformers, and outdoor circuit breakers from a safe distance without entering the risk zone. In low-voltage installations, thermal-imaging cameras help to locate hot spots of loose connections, load imbalances, corrosion, and increase in impedance to current, and so on. The temperature of mechanical components rises as these wear out and become less efficient. This increases overall temperature of equipment, causing these to fail. Thermal-imaging cameras can monitor many mechanical systems including motors, couplings, gearboxes, bearings, pumps, compressors, belts, blowers, and conveyor systems.

There are many other areas where thermal-imaging cameras play an important role. These include flare detection, tank-level detection, hot-spots detection in welding robots, aeronautical materials inspection, moulds inspection, and inspection in paper mills, pipe-work in refractories and petrochemical installations. Thermal-imaging cameras, scan entire components, giving instant diagnostic insights with the full extent of problems. These are easier to use and find problems faster with higher accuracy. It is shown in Figure 6.35.



FIGURE 6.35. Modern thermal-imaging cameras are small, lightweight, and easy to use.

The hotter the object, the more its infrared radiation. Infrared energy coming from the object is focused by the optics onto an infrared detector. The detector sends this information to a sensor for image information to a sensor for image processing using complex calculations. The sensor translates the data into an image that can be viewed on the LCD screen. Each pixel of the image is a measure of the temperature at different points. This is achieved by complex algorithms performed in the thermal imaging camera.

Summary

- In digital camera CCD array of sensors is used for image formation and number of sensors is equal to number of pixels.
- The size of image depends on number of rows, columns, and bits per pixel.
- In analog cameras, the image formation is due to the chemical reaction where as in digital it is a bit.
- The shutter speed corresponds to the exposure time of the amount of incident light on the sensor.

- Gain is a digital-camera setting that controls the amplification of the signal from the camera sensor.
- Gamma is a digital-camera setting that controls the grayscale reproduced on the image.
- Dynamic range is the difference between the lowest detectable light level and the highest detectable light level.
- Zooming refers to increase the quantity of pixels using three methods such as pixel replication, zero-order hold method, and zooming k times.
- Spatial resolution refers clarity measured in pixels per inch and gray-level resolution refers levels of gray in image and equal to number of bits per pixel.
- Thermal imaging cameras help to locate hot spots of loose connections and monitor mechanical electrical systems.

References

<https://www.tutorialspoint.com/>

<http://www.vision-systems.com/articles/print/volume-20/issue-9/features/>

<https://www.edmundoptics.com/resources/application-notes/imaging/>

Learning Outcomes

- 6.1 How is image formed in camera?
- 6.2 Define pixel.
- 6.3 List a few color models.
- 6.4 Differentiate average and weighted grayscale image methods.
- 6.5 Differentiate analog versus digital, area-scan versus line-scan cameras.
- 6.6 Among interlaced versus progressive cameras, which is best one?
- 6.7 What is zooming and what types of zooming are there?
- 6.8 What is meant by spatial resolution and gray-level resolution?
- 6.9. Write the charge coupled device camera sensor construction.
- 6.10 Use a diagram to explain the CMOS sensor construction.
- 6.11 What is meant by spectral properties of monochrome and color cameras?

- 6.12 What are the parameters needed for digital-camera settings?
- 6.13 Which pixel depth is suited for grayscales and why?
- 6.14 List different camera interfaces that are available. Compare them.
- 6.15 Write a note on camera imaging software.
- 6.16 Write about thermal-imaging cameras.

Further Reading

The Digital Photography Book by Scott Kelby

EMBEDDED VISION PROCESSORS AND SENSORS

Overview

Embedded vision processors are fully programmable with DSP, CNN, and open CV library support. Any of the following can be selected for embedded vision application: high-performance embedded CPU, ASSP CPU, GPU CPU, DSP CPU, and FPGA CPU. Proximity, position, velocity, level, temperature, force, vibration, flow, altimeter, oxygen, gyroscopes, image, biometric, biosensors, and pressure are some of sensors used in EV applications. Range, resolution, target composition, repeatability, and form factor are the selection parameters for the sensor.

Learning Objectives

After reading this one will know the

- blocks in embedded vision processors and its features;
- different types of processors for vision system;
- different sensor for different EV application areas; and
- selection criteria for sensor and processor.

7.1 VISION PROCESSORS

The estimated vision-system shipments are above 300 billion dollars in the year 2020 market as shown in Figure 7.1(a). The video-surveillance market, used for things such as home surveillance, retail, healthcare, security in airports, government, banks, and casinos, are expected to grow

at a rate of 37.3% from the years 2012–2020. ADAS vision market also continues to increase as shown in Figure 7.1(b).

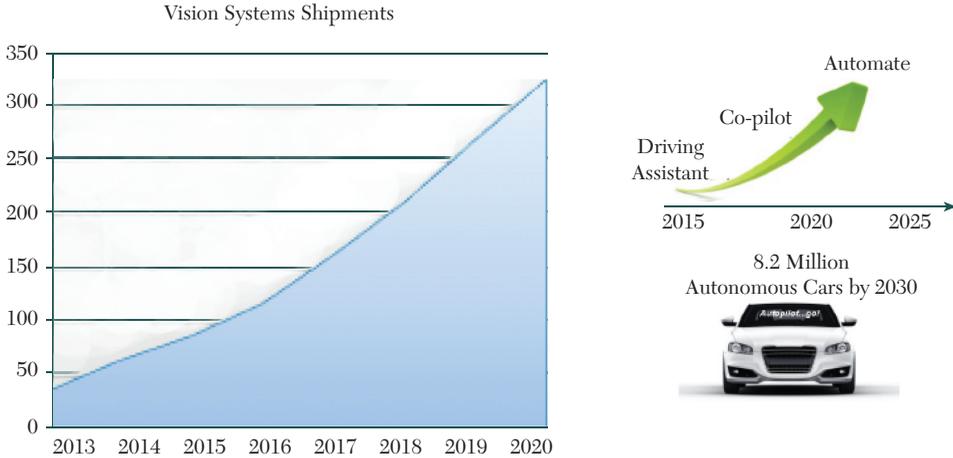


FIGURE 7.1. (a) Growth of vision-system shipments (b) ADAS vision market.

Embedded vision demands lots of processing performance. This is because most embedded vision systems operate on video data (which is high data rate) using complex algorithms (which are needed in order to reliably extract meaning from images). In addition to being complex, vision algorithms tend to evolve continuously, which means that ease of programming is a key attribute for processors competing in this space.

When designers have data-intensive applications demanding lots of computer performance with low power, low cost, and programmability the answer is usually highly parallel, specialized processor architectures. And, indeed, that's a common theme among processors targeting vision applications. But despite this commonality, there's extreme diversity among architectures being promoted for vision applications. These include DSPs, GPUs, FPGAs, and a variety of more specialized, vision-specific architectures.

Each type of processor has its strengths and weaknesses, and each application has its unique requirements. There is no one class of processor that's best for all vision applications. In fact, even for a single-vision application, often a combination of two or three different processor types is optimal. For system and chip designers, there is a robust field of vision-oriented processors to choose from. But they have to determine which offerings are best suited for specific application. And this work is made

more challenging by the vast diversity of architectures and programming models.

Once system and software developers get comfortable using a particular type of processor for a particular purpose, it's hard to change both because the engineers are comfortable and productive with the processor they're familiar with, and also because of the big investments already made in designing and optimizing hardware and software around that processor. So even if another type of processor emerges which offers better performance, energy efficiency, or programmability, it can be difficult to displace the serving processor. Hence, big advantages are accruing to the processor suppliers who are getting to market early with credible offerings for embedded vision applications.

Embedded vision applications typically require very high performance, programmability, low cost and energy efficiency processors for its operations. Achieving all of these together is difficult. Dedicated logic yields high performance at low cost, but with little programmability. General purpose CPUs provide programmability, but with weak performance or energy efficiency. Demanding embedded vision applications will most often use a combination of processing elements, for example,

- CPU for complex decision making, network access, user interface, storage management, overall control
- high-performance DSP-oriented processor for real time, moderate rate processing with moderately complex algorithms.
- highly parallel engine(s) for pixel-rate processing with simple algorithms

Any one of the following processor types is selected for vision applications depending on requirements.

1. High performance embedded CPUs—Although challenged with respect to performance and efficiency, unaided high-performance embedded CPUs are attractive for some vision applications. CPUs are easiest for using tools, operating systems, middleware, etc. Most systems need a CPU for other tasks. However, performance and/or efficiency is often inadequate and memory bandwidth is a common bottle neck. For example, the Intel Atom Z510 is best suited for applications with modest performance needs.
2. Application-specific standard product + CPU—Application-specific standard products (ASSPs) are designed for specific applications. They usually

include strong application-specific software-development infrastructure and/or application software. However, the specialization may not be right for designer particular applications. They may come from small suppliers, which can mean more risk. They use unique architectures, which can make programming them, and migration to other solutions, more difficult. Some are not user programmable. For example PrimeSense PS1080 A2 is best suit for ultrahigh volume, low-cost applications.

3. Graphics processing unit (GPU) + CPU—It often used for vision algorithm development. It's widely available and easy to get started with parallel programming. It is well integrated with CPU, sometimes on one chip. Typically it cannot be purchased as a chip, only as a board, with a limited selection of CPUs. For example, NVIDIA GT240 is best suit for performance hungry applications with generous size/power/cost budgets.
4. DSP Processor + Coprocessors + CPU—DSPs are suitable for vision. However, DSPs often lack sufficient performance, and aren't as easy to use as CPUs. Hence, DSPs are often augmented with specialized co-processors and a CPU on the same chip. For example, Texas Instruments DaVinci is best suited for applications with moderate performance needs and moderate size/power/cost budgets.
5. Mobile “application processor”—It is energy efficient and often had strong development support. However, specialized coprocessors are usually not user programmable. For example, Qualcomm QSD8560 is best suited for applications with moderate performance needs, wireless connectivity, and tight size/power/cost budgets.
6. FPGA + CPU—FPGA flexibility is very valuable for embedded vision applications. It enables custom specialization and enormous parallelism. However, FPGA design is hardware design, typically done at low level. It ease of use improving due to platforms, IP block libraries, and emerging high-level synthesis tools. Low performance CPUs can be implemented in the FPGA and high performance integrated CPUs on the horizon. For example, Xilinx Spartan-3 XC3S4000 is best suit for high-performance needs with tight size/power/cost budgets.

Selecting a processor for an embedded vision application steps are:

Step 1: Will it fit on the CPU?

Step 2: Is there a suitable ASSP?

Step 3: Implement on a GPU, FPGA, or + DSP + accelerators.

Hardware Platforms for Embedded Vision, Image Processing, and Deep Learning

TABLE 7.1. Devices required for embedded vision.

Device Family: Fastest Calculations: Battery Usage:	Micro controller <0.2 GFLOPS < 0.3 Watts
Features:	Most microcontrollers (e.g., Arduino, AVR, PIC) are far too slow for camera processing, but a 32-bit ARM Cortex-M4 (e.g., 168MHz STM32F407) might handle some extremely basic camera applications. Microcontrollers only support very minimal OSs, so typically software runs as low-level firmware, and write most algorithms and code by designers, with the advantage of direct access to the hardware such as I/O pins and timers and hard real-time operation.
Device Family: Fastest Calculations: Battery Usage:	Mobile SoC dev board or tablet 1-25 GFLOPS 1-6 Watts
Features:	The latest mobile ARM CPUs can provide both great speed and low battery draw. For doing mostly integer processing, a cheaper Cortex A8 (e.g., BeagleBone Black) or even an ARM11 (e.g., Raspberry Pi) might be good enough. For doing a lot of floating point, a Cortex A9 (e.g., quad core ODROID U3) or Cortex A15 (e.g., quad core ODROID XU or quad core Jetson TK1) board are enough. Their FPU hardware is many times faster than the FPU in Cortex-A8. For most performance available then, the GPU acceleration of Jetson TK1 is fastest. For smallest size, then a Gumstix Overo is tiny. For most efficient CPU, then Cortex A7 (e.g., A20 OLinuXino LIME2) is available. For visualizing images on a screen, then a rooted tablet running Android or Linux might be a better option (using Wifi or Bluetooth to a microcontroller for the needed I/O access). Software development for an ARM SoC is similar to desktop software and libraries like OpenCV are supported on ARM, but it's not as easy as x86.
Device Family: Fastest Calculations: Battery Usage:	Net book or small laptop 15-110 GFLOPS 30-100 Watts

Features:	Portable computers (e.g., Mini ITX M/B + quad core 3.5GHz Core i7 CPU) can have a really fast CPU and are really easy to develop code on just like a PC, but are a lot more power-hungry and larger. For visualizing images, then a netbook might be a better option.
Device Family: Fastest Calculations: Battery Usage:	x86 laptop with dGPU 240-2200 GFLOPS 40-110 Watts
Features:	Some larger laptops include a dedicated GPU capable of CUDA or OpenCL GPU acceleration (e.g., MSI GE60 or Alienware 14), so are very well suited to intense computer vision. These are also really easy to develop code on just like a PC, but are a lot more power-hungry, heavy, and larger, even compared to x86 SBCs & net books
Device Family: Fastest Calculations: Battery Usage:	FPGA/DSP/ASIC/DPU /CV hardware design 50-1000 GFLOPS 0.5-3 Watts
Features:	FPGAs (e.g., Cyclone II Starter Kit + 5MP camera) can be extremely fast with extremely low battery usage, but are very complex to design. Some high-end DSPs are powerful enough for vision or deep learning and are basically CPUs with large-scale SIMD instructions, and thus need specialized programming but are much easier to develop algorithms on them than FPGAs (e.g., TI C6x / EVE DSPs, Qualcomm “Machine Learning Platform” DSPs, and analog devices). Some offer specialized CPU designs that require specialized programming such as multi-processor (e.g., Adapteva Parallella FPU) or VLIW + Vector CPUs or DSPs (e.g., Mobileye EyeQ, Tensilica Xtensa, CEVA CEVA-XM and Qualcomm Hexagon). There are also some fixed-function imaging or vision accelerator ASIC chips that are extremely efficient at certain very specific algorithms (e.g., Ambarella, Movidius Myriad 2, TI EVE, NEC IMAPCAR, Inuitive, FotoNation IPU, Renesas IMP, Visconti, Sensity / Eutecus, and Analog Devices). There are vision IP cores to put into designer’s own silicon chips (e.g., CEVA, Tensilica, Synopsis, Videantis, Apical, CogniVue, Imagination Technologies, Vivante / VeriSilicon, and Adapteva “Epiphany”). There are

	also some highly parallel Neural Network / Deep Learning Processor DPU AI accelerator chips on the market (e.g., Nervana/Intel, Wave Computing, DeePhi, IBM TrueNorth Neuromorphic computer, and Toshiba TDNN).
--	---

When developing embedded vision or image-processing applications for a desktop computer, the choices in hardware platforms are very simple. If the project doesn't need much processing then use a cheap laptop or PC with integrated GPU, or if it needs lots of processing power, use a fast CPU with a powerful dedicated GPU card. But for embedded systems, there are a lot more options to choose from, and there is no single device that is suitable for all embedded platforms because each has some advantages and some disadvantages. Table 7.1 is a basic summary of embedded hardware that can be used for embedded vision and image processing.

Requirements of Computers for Embedded Vision Application

When selecting a PC for a vision application there are many factors that need to be considered in order to ensure that the solution chosen delivers the performance needed, combined with both long-term reliability and stability of supply. Only a few years ago compact and embedded PCs just did not have the performance needed for an embedded vision system. In recent years we have seen significant reductions in the power to performance ratio and are now seeing very capable fanless and compact industrial PCs without internal cables or internal moving parts, which can deliver performance with high environmental specifications suitable for both the harsh environments of the factory floor and the processing power required for imaging. As in most markets there is a wide variety of options and invariably you get what you pay for.

Selecting the right components can make a huge difference to the performance of a vision system and its long-term reliability, especially as a vision system is expected to handle very large amounts of data (images) in quick succession. Not all PC systems are well suited for these tasks and most off-the-shelf domestic computers often have never been stress-tested with continuous high throughput of data. Some of the key considerations when specifying vision computing hardware are listed below.

What is the maximum data rate that the system needs to handle?

- Number of cameras / image size / frame rate?
- Memory bandwidth required?

How many simultaneous images need to be held in memory for immediate access?

- Memory required?

Does the system have to process this data at the same rate?

- Combination of speed / number of processors / memory bandwidth?

Does the system have to record all this data to disk?

- Disk configuration, data rate?

What level of hardware redundancy is required to ensure data integrity?

- Disk mirroring, redundant power supplies?

Does the software support multiprocessor or multicore systems?

- Number of processors and different configurations?

What level of reliability MTBF (mean time before failure) has to be achieved?

- Consider the PC design including cables used or solid state disks.

What is the environment of the target location in terms of temperature, vibration?

- Expected production start / life cycle?

- Component obsolescence and supply security?

Mechanical considerations?

- Housing size and design?

Environmental aspects?

- Lead-free, PC fans/filters, electrical noise, EMC, etc.

Depending on the environment, a computer has to meet certain demands. When a system is deployed in a harsh industrial environment, considerations such as shock, vibration, thermal cycling, humidity, and dust have to be considered, whereas a system used in a clean room can have lesser demands. Some environments specify extreme vibration resistance (e.g., mobile applications). Other imaging applications demand stability even at extreme temperatures (e.g., traffic surveillance where computers are installed outside). The same requirements have to be considered for the insides of a vision system.

Processor Configuration Selection

There are many PC processor configurations, all with varying performances. In contrary to other applications, processor speed is not the only important factor in choosing a proper processor for an imaging application. In general, suitable processors can be divided into three application areas:

- Workstation: Intel Core i7/i5
- Mobile: Intel Atom & ULV Core i5 & i7
- Server: Intel Xeon

Workstation processors are most commonly found in standard imaging applications, while mobile processors are often found in embedded vision systems due to their low power consumption. For the most demanding applications that need multiple processors or higher data throughput rates, server processors are often used as they offer more memory ports, increasing the data flow through the processor.

In most cases a single chip includes two or four processors that allow different applications or tasks to run on each processor. In theory, this doubles or quadruples the performance of the PC compared to a single processor-based machine. This however is only beneficial for a vision application if the software supports “multi-threading.” Multi-core processors offer a viable alternative to using two separate server processors such as the Intel Xeon. Typically, server motherboard designs support four processors, providing systems with 16 cores and more. The core count continues to increase as does memory bandwidth.

7.2 EMBEDDED VISION PROCESSORS

The embedded vision processors are fully programmable and combine the flexibility of software solutions with the high performance and low power consumption of dedicated hardware as shown in Figure 7.2.

Features of an embedded vision processor combine a high-performance 32-bit scalar core with a 512-bit vector DSP, an optimized CNN engine, and support for user-defined APEX hardware accelerators. The processors are highly scalable and configurable enabling users to tailor them to their specific application requirements to maximize performance while minimizing power and area. The optional CNN engine is a programmable

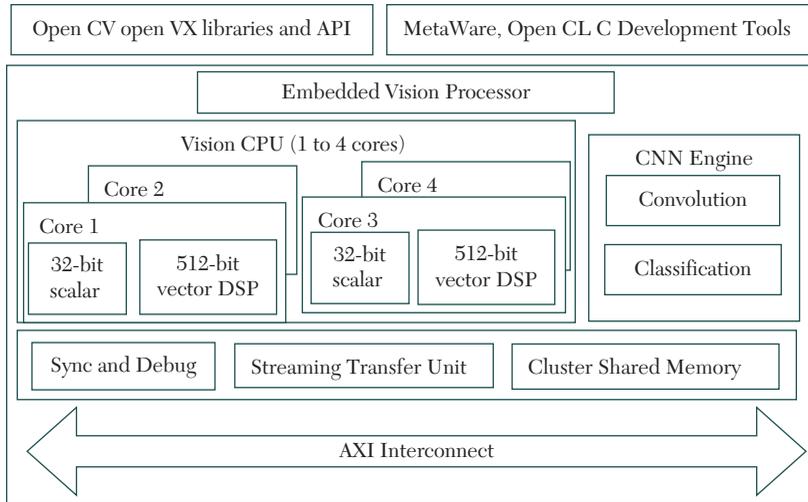


FIGURE 7.2. Embedded vision processor.

object detection engine that implements a convolutional neural network (CNN) enabling fast and accurate detection of a wide range of objects such as faces, pedestrians and hand gestures. It has a quad-core vision CPU. It supports 1080p–4K vision streams and open CV, open VX, and open CL C high-productivity toolset.

The vision CPU features a quad-issue super-vector architecture with a 32-bit, high-performance scalar pipeline and a 512-bit wide SIMD VDSP that are optimally balanced to achieve excellent performance with low power consumption. The core executes one scalar and three vector instructions (128-bit instruction bundle) per cycle. The pipeline is based on the ARC HS architecture and sophisticated branch prediction unit and a late-stage ALU that improves instruction throughput.

The vision processors feature separate instruction and data L1 cache that can be independently configured for 4K, 8K, 16K, 32K, or 64KB size. The processors support 8KB to 128KB of data closely coupled memory, and 32KB to 256KB of vector memory for the VDSP. The processors register file has 32 32-bit registers. The register file can be constructed from fast, single-cycle access memory or flip flops, and supports one or two write ports and two read ports. The processors feature 64-bit load double and store double instructions. These are single instructions that load or store 64-bits of data to and from register pairs.

Convolution Neural Networks (CNN) Engine

CNNs are inspired by the way our brains work to process vision. CNNs are based on a deep learning algorithm that is trained with many images of an object and then generalized to a graph that can be used by the algorithm to find the object in pictures or video. They perform image analysis using a successive refinement process that uses multiple-feature extraction layers. At each layer, the image is matched against the patterns that result from the training phase. Each successive layer extracts increasingly complex features until a final match decision is made.

The processors can be configured with an optional programmable embedded deep neural-network engine used to run CNN executables. The engine is optimized for object detection, image classification, and scene segmentation with excellent performance efficiency. The configurability of the CNN engine enables the flexible mapping of CNN graphs into the engine resources. CNN graph training is done off line, typically on a server farm, and the resulting graph is programmed into the object detection engine by the user with the CNN graph mapping tool.

Cluster Shared Memory

A low-latency shared-data memory is included in the processor to support information passing and coordination between the multiple CPUs and the CNN processing element cores. This memory is used as a software managed scratch pad and is configurable from 0 to 8MB. To allow for larger sizes, the memory is internally multi-banked, but this is invisible to the software.

Streaming Transfer Unit

The processor has a configurable DMA controller to permit efficient transfer of data between the processor and different memory regions on or off chip.

Bus Interface

The processor has native support for the ARM bus protocol. The AXI bus is 64 or 128 bits wide to improve system throughput. The processor supports up to four output interrupt pins, and up to three input interrupt lines. These can be used, for example, to synchronize with an external host. The host can also raise an interrupt by writing in a memory mapped register or by driving an interrupt input pin on the processor.

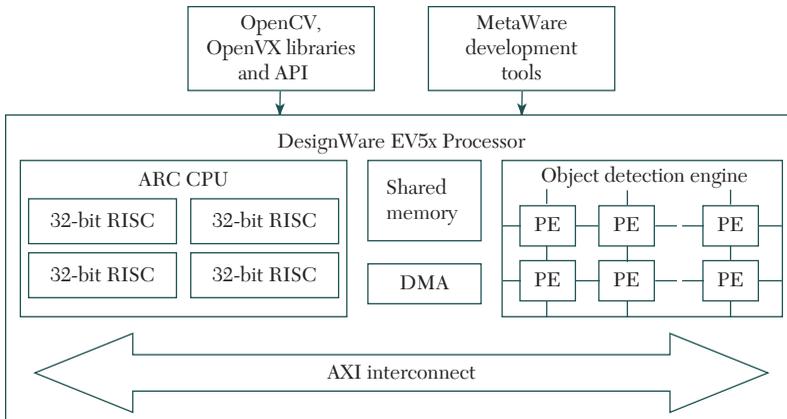
Complete Suite of Development Tools

The processors are supported by a complete suite of development tools that include the MetaWare Development Toolkit, which includes a C/C++ Compiler that generates highly efficient scalar code, a source level debugger, and the ARC nSIM instruction set simulator. MetaWare EV SDK Option consists of the OpenCV Library, OpenVX Runtime framework, and kernels and an OpenCL C language compiler. OpenCL C is a C like language and is used with the processors to develop kernels that are executed in the OpenVX graph. OpenCV (an Open Source Computer Vision library) is a software library of more than 2,500 algorithms that can be used with the MetaWare tools and provides a software infrastructure for embedded vision applications. OpenCV can be used to detect and recognize objects, and a full range of machine vision capabilities. OpenVX is a Khronos standard for acceleration of embedded vision algorithms. Client defined OpenVX functions are supported, with the kernels described in standard C/ C++ or in OpenCL. OpenVX graphs are automatically mapped and executed on the processor and object detection engine.

Intel Movidius Myraid X Vision Processors

The Intel embedded vision processor EV52 features a dual-core 32-bit CPU with a programmable CNN object-detection engine that is user configurable with up to eight processing elements. The EV54 features a quad-core CPU and the programmable CNN object detection engine. To speed application software development, the EV5x processor family is supported by a comprehensive software programming environment based on existing and emerging embedded vision standards including OpenCV and OpenVX, as well as the Synopsys ARC MetaWare Development Toolkit. This is shown in Figure 7.3.

The OpenCV source libraries available for EV processors provide more than 2,500 functions for real-time computer vision. The processors are programmable and can be trained to support any object detection graph. The OpenVX framework includes 43 standard computer-vision kernels that have been optimized to run on the EV processors, such as edge detection, image pyramid creation, and optical flow estimation. Users can also define new OpenVX kernels, giving them flexibility for their current vision applications and the ability to address future object-detection requirements. The OpenVX runtime distributes tiled kernel execution over the EV processors multiple execution units, simplifying the programming



PE = processing element

FIGURE 7.3. Intel Movidius Myriad X.

of the processor. The full suite of tools and libraries, along with available reference designs, enable designers to efficiently build, debug, profile, and optimize their embedded vision systems.

The Intel Movidius Myriad 2 VPU delivers high-performance machine vision and visual awareness in severely power constrained environments. Myriad 2 gives developers immediate access to its advanced vision-processing core, while allowing them to develop proprietary capabilities that provide true differentiation. The Myriad 2 reference board includes reference camera and MEMS sensors, and an integration kit for applications processors, MIPI to USB Bridge for rapid PC/board prototyping, enabling developers to efficiently prototype applications.

Matrox RadientPro CL

The Matrox RadientPro CL is a vision-processor board supporting the highest camera link acquisition rates with FPGA-based processing offload capabilities customizable by Matrox or using the optional Matrox FPGA Development Kit. It capture images at the highest Camera Link rates with support for the Full and 80-bit modes at up to 85MHz. It eliminate lost pixels through a PCIe 2.0 x8 host interface and ample on-board buffering. It reduce cabling and eliminate power supplies by way of Power over Camera Link (PoCL) support. It offload and accelerate image processing to free and assist the host CPU using an Altera Stratix V FPGA. It simplify the development of custom on-board image processing using the optional Matrox FPGA Development Kit. It reduce development and validation costs

through a managed lifecycle offering consistent long term availability. It implements image capture with ease and confidence using Matrox Imaging Library (MIL) application development toolkit. It maintain flexibility and choice by way of 64-bit Windows 7/8.1/10, Linux and RTX64 (RTOS) support. It is shown in Figure 7.4(a).

Single-board Computer Raspberry Pi

The Raspberry Pi is a more powerful CPU, a 900 MHz quad-core processor, with a lot of memory and significantly more power for imaging applications. The ARM Cortex-A7 quad-core CPU can take control of an application and support machine vision tasks. As an example, the use of machine vision software HALCON in combination with IDS cameras and Raspberry Pi 2 is suitable to identify fonts or barcodes. The entire range of USB 2.0 industrial cameras and GigE cameras can be operated on Raspberry Pi 2 with the new driver. A powerful image processing solution can be developed on a PC and then transferred to the embedded system where it will run independently. In addition, there are a variety of software development kits available that will provide interfaces to a wide range of camera types. Smart cameras contain the image capture and processing capabilities within the camera unit itself, while compact, or multipoint imaging systems feature a self-contained unit for image well, there is a real scalable choice of embedded vision solutions and the goals of the application must be used to drive the selection. It is shown in Figure 7.4 (b).

Nvidia Jetson TX1

Nvidia Jetson TX1 is loaded with a 64-bit quad-core ARM Cortex-A57 CPU with a 256-core Maxwell GPU. The Nvidia Jetson TX1 is one of the most powerful devices in the market for embedded computer vision, as shown in Figure 7.4(c). What makes it more impressive is that it consumes just 10W of power to deliver 1 Teraflop 16FP performance. It is the right choice for high-end embedded vision applications.

Nvidia Jetson TK1

Nvidia Jetson TK1 is the predecessor of Jetson TX1. With 192-core Kepler GK20a GPU, it is priced at \$1 per CUDA core, and it delivers a performance of 300 GigaFlops. TK1 doesn't have onboard WiFi or Bluetooth. However, these can be added via USB or the mini-PCIe port. It is shown in Figure 7.4(d).

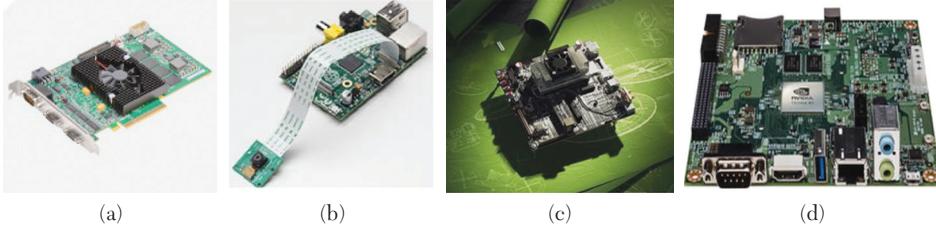


FIGURE 7.4. (a) Matrox RadiantPro CL. (b) Raspberry Pi and camera. (c) Nvidia Jetson TX1. (d) Nvidia Jetson TK1.

Beagle board: Beagle Bone Black

Beagle Bone Black is popular for IoT applications. It is shown in Figure 7.5(a). As compared to Raspberry Pi which has a single 26-pin header that can be used as 8 GPIO pins, or as a serial bus, the Beaglebone Black has two 48-socket headers that can be utilized for virtually limitless I/O hardware. It also includes a number of analog I/O pins that allow it to connect to a variety of sensor hardware that can't be used with an out-of-the-box Raspberry Pi. Compared to the Raspberry Pi, it is double the price and exhibits inferior performance. With that said, BeagleBone Black isn't an excellent choice for embedded vision as video decoding, 3D rendering, and general GUI performance are all much better on Raspberry Pi 3.

Orange Pi

Orange Pi has slightly better hardware than Raspberry Pi for the price point. It also has some features missing from Raspberry Pi like SATA, Gigabit Ethernet, IR, and Mic. This is shown in Figure 7.5(b).

ODROID-C2

ODROID-C2 packs double the RAM and much faster processor than Raspberry Pi 3. It is shown in Figure 7.5(c). Features like Gigabit Ethernet and 4K video support make it superior to Raspberry Pi 3. Software support and the strength of the community is nowhere close to Raspberry Pi. Another plus point for ODROID-C2 is its easy availability as opposed to Raspberry Pi.

Banana Pi

Banana Pi has the same processing per dollar as that of Raspberry Pi. Banana Pi is shown in Figure 7.5(d). There are a few more devices, for example the Intel Edison, that are more suitable for IoT use cases. Arduino board is another device that is extremely popular and a lot of hobbyists and students run some embedded vision algorithms on it.

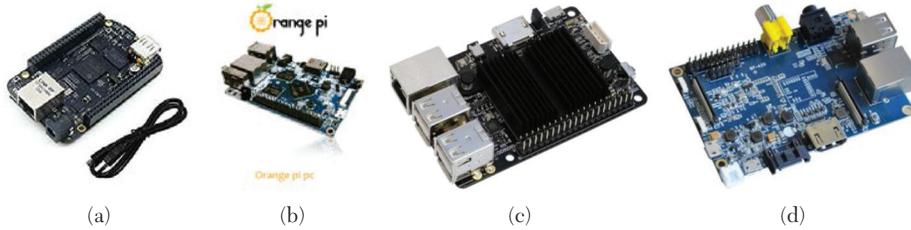


FIGURE 7.5 (a) BeagleBone Black. (b) Orange Pi. (c) ODROID-C2. (d) Banana Pi.

CEVA-XM4 Imaging and Vision Processor

The CEVA-XM4 imaging and computer vision processor IP solves the most critical issues for the development of energy-efficient embedded vision systems where die size and power budget are extremely constrained, yet algorithms require intensive processing as shown in Figure 7.6. The CEVA-XM4 and its associated tools and libraries combine to deliver a comprehensive vision IP platform that allows developers to simply and easily address the key elements of intelligent vision processing, namely 3D-vision, computational photography, visual perception, and analytics.

CEVA-XM4 Block Diagram

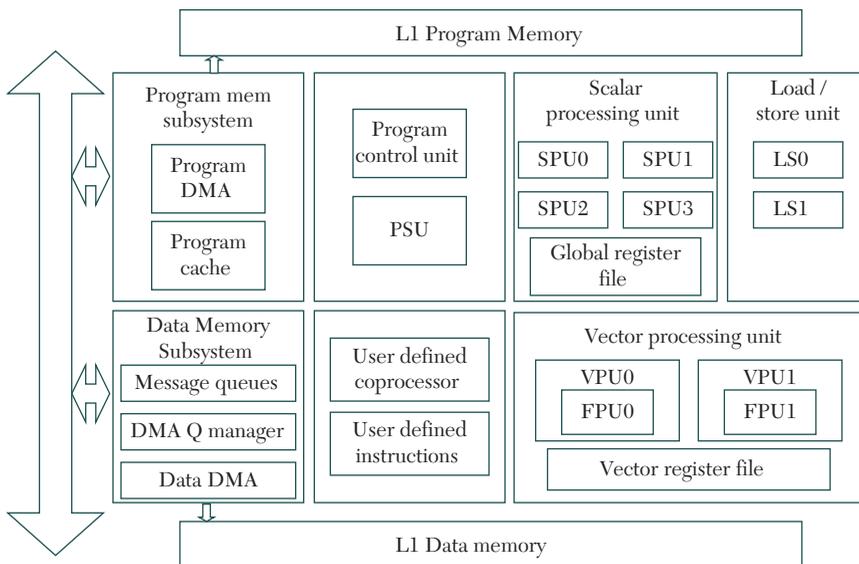


FIGURE 7.6. CEVA XM4 block diagram.

The CEVA-XM4 includes running embedded vision on video streams (1080p, 4K), combining depth generation with vision processing, enabling multi-application processing (for example, combining gesture, face detection, emotions, eye-tracking, and depth), and implementing multi-image algorithms in high resolution. Fully programmable in high-level languages. The CEVA-XM4 is an extremely high-performance, low-power, fully synthesizable DSP and memory subsystem IP core that incorporates a wide vector architecture with fixed and floating point processing, multiple simultaneous scalar units, and a vision-oriented, low-power instruction set.

MAX10 FPGA

MAX10 FPGA integrates features such as analog-to-digital converters (ADCs) and dual-configuration flash allows to store and dynamically switch between two images on a single chip. It also includes soft-core embedded processor support, digital-signal processing blocks, and soft DDR3 memory controllers. MAX 10 FPGAs offer system-level cost savings through increased integration of system component functions.

Dual configuration flash—A single, on die flash memory supports dual configuration, for true fail-safe upgrades with thousands of possible reprogram cycles.

Analog blocks—Integrated analog blocks with ADCs and temperature sensor provide lower latency and reduced board space with more flexible sample sequencing.

Instant on—MAX 10 FPGAs can be the first usable device on a system board to control bring up of other components such as high-density FPGAs, ASICs, ASSPs, and processors.

Nios II soft-core embedded processor—MAX 10 FPGAs support the integration of the soft-core Nios II embedded processors, providing embedded developers a single chip, fully configurable, instant on processor subsystem.

DSP blocks—As the first nonvolatile FPGA with DSP, MAX 10 FPGAs are ideal for high performance, high-precision applications using integrated 18x18 multipliers.

DDR3 external memory interfaces—MAX 10 FPGAs support DDR3 SDRAM and LPDDR2 interfaces through soft intellectual property (IP) memory controllers, optimal for video, data path, and embedded applications.

Complex control management software—controlled system management through Nios II soft core embedded processors.

Single-core voltage support—Single supply offering required for power-up sequence management.

User flash—With up to 736 KB of on die user flash code storage, MAX 10 FPGAs enable advanced single-chip Nios II embedded applications. The amount of user flash available depends on configuration options.

Intel FPGAs are ideal for machine vision (MV) cameras, allowing designs to accommodate a wide variety of image sensors as well as MV-specific interfaces. FPGAs can also be used as vision-processing accelerators inside the edge computing platform to harness the power of artificial intelligence deep learning for analysis of the MV data. It implements various bus interfaces, such as PCI, PCIe, Gbps Ethernet, USB, and others. It integrates a wide range of functions such as image capture, camera interfaces, preprocessing, and communication functions, all within a single FPGA.

The Cyclone V SoC, combines image-signal processing pipeline with machine-vision algorithms executing the ARM A9 hard-processor system to build complete machine vision systems on chip. It uses Simulink and Embedded Coder from the MathWorks to generate C/C++ code for Cyclone V SoCs. It has flexibility to interface many types of image sensors. It has fast processing to incorporate a full-image sensor pipeline (ISP) intellectual property (IP) that includes techniques, such as defect pixel correction, gamma correction, dynamic range correction, and noise reduction. It has cost-effective solutions that can incorporate functions such as sensor interfacing, image compression, and even pan-tilt-zoom (PTZ) control.

Vision DSPs for Imaging and Vision

The Vision Q6 DSP is the latest DSP for embedded vision and AI built on a new, faster processor architecture. The fifth-generation Vision Q6 DSP offers 1.5X greater performance than its predecessor, the Vision P6 DSP, and 1.25X better power efficiency at the Vision P6 DSP's peak performance. The Vision P6 DSP, introduced in 2016, set a new standard in AI performance for a general purpose embedded vision DSP by offering 4X the peak performance compared to the Vision P5 DSP. The Vision P5 DSP, introduced in 2015, has been highly successful in the mobile market. It offers up to 4X-100X the performance relative to traditional mobile CPU+GPU systems at a fraction of the power/energy.

The DSP family offers general purpose imaging and vision products that were designed for the complex algorithms in imaging and embedded vision,

including innovative multi-frame noise reduction, video stabilization, high-dynamic range (HDR) processing, object and face recognition and tracking, low-light image enhancement, digital zoom, and gesture recognition, plus many more.

The Vision DSP family offloads the host CPU for lower energy consumption running intensive imaging and vision applications. Multicore host CPUs can't handle these power-hungry, bandwidth-demanding applications. Hardwired accelerators are restricted to a fixed set of functions, and GPUs offer pipelines that are not required or not efficient in image and vision processing applications. Imaging and vision algorithms can run on a DSP that's specifically optimized for the imaging and vision functions required.

The instruction set, memory system, and data types have all been optimized for high throughput 8, 16, and 32-bit pixel processing for all Vision P5, P6, and Q6 DSPs. The Vision DSP family is available as licensable, synthesizable IP with rich libraries and advanced software tools, allowing you to write your code in C/C++.

Vision Q6 DSP Features and Benefits

A deeper, 13-stage processor pipeline and system architecture designed for use with large local memories enable the Vision Q6 DSP to achieve 1.5GHz peak frequency and 1GHz typical frequency at 16nm, in the same floor-plan area as the Vision P6 DSP. As a result, designers using the Vision Q6 DSP can develop high-performance products that meet increasing vision and AI demands and power efficiency needs.

- An enhanced DSP instruction set results in up to 20% fewer cycles than the Vision P6 DSP for embedded vision applications/kernels such as Optical Flow, Transpose, and warpAffine, and for commonly used filters such as Median and Sobel.
- 2X system data bandwidth with separate master/slave AXI interfaces for data/instructions and 2-channel DMA alleviates memory bandwidth challenges in vision and AI applications, and also reduces latency and overhead associated with task switching and DMA setup.
- Backwards compatibility with the Vision P6 DSP, so customers can preserve their software investment for an easy migration.
- Optional vector floating point unit (VFPU) also supports half precision (FP16).

Vision P6 DSP Features and Benefits

With new instructions, increased math throughput, and other enhancements, the Vision P6 DSP sets a new standard in imaging and computer vision benchmarks, increasing the performance by up to 4X compared to the highly successful Vision P5 DSP.

For AI applications, the Vision P6 DSP boosts performance by up to 4X with quadruple the available MAC horsepower, which is a major computation block for AI applications. Compared to commercially available GPUs, the Vision P6 DSP will achieve twice the frame rate at much lower power consumption on a typical AI implementation. For a wide range of other key vision functions, such as convolution, FIR filters, and matrix multiplies, the Vision P6 DSP increases performance by up to 2X with its improved 8-bit and 16-bit arithmetic. Figure 7.7 shows the Vision P6 DSP block diagram.

- Processes 9728 bits per cycle
- Offers 256 MACs: 4X compared to Vision P5 DSP

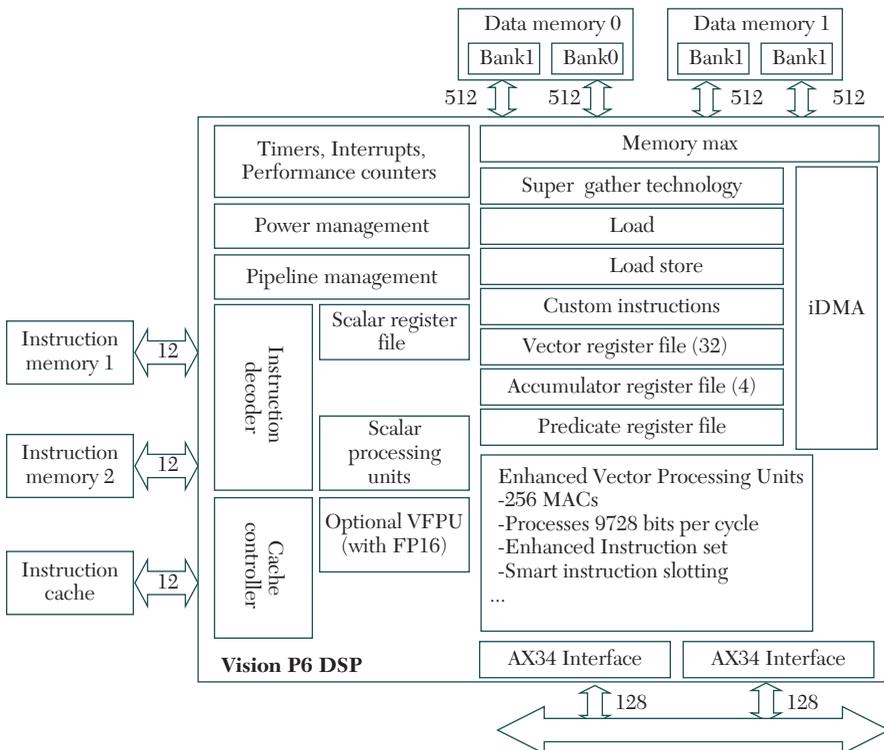


FIGURE 7.7. Vision P6 DSP block diagram.

- Enhanced instruction set and instruction slotting
- Fully software compatible with Vision P5 DSP
- Optional VFPU with single-precision 32-bit and/or half precision 16-bit floating point support offers performance and flexibility for porting existing GPU code

Vision P5 DSP Features and Benefits

- Offers up to 13X vision processing performance improvement over the previous generation Vision DSP
- Processes 7168 bits per cycle
- Optional vector floating point unit (VFPU) with single precision 32-bit floating point support offers flexibility to provide high precision math at a minimal area penalty.

VFPU

The Vision Q6, P6, and P5 DSPs provide an optional VFPU for those applications that need this precision or as a quick way to port existing code. The VFPU offers significant performance improvement with very little area increase. The Vision P6 and Q6 DSPs offer optional support for a 32-way VFPU with half precision (FP16) format. Table 7.2 gives the different features of Vision DSP processors.

TABLE 7.2. Features of Vision P5 DSP and Vision Q6/P6 DSP

	Vision P5 DSP	Vision Q6/P6 DSP
Number of bits processed per cycle	7168	9728
MACs	64	256
16-bit (FP16) VFPU support (optional)	No	Yes
32-bit (FP16) VFPU support (optional)	Yes	Yes

Wide vector SIMD data processing for superior performance

The VLIW issue of vector operations gives an almost arbitrary mix of loads, stores, multiplies, and ALU operations, resulting in a rich set of pixel computations. Up to 320 operations can be issued per cycle and 256 of these can be ALU operations.

SuperGather

The Vision Q6, P6, and P5 DSPs integrate the highly sophisticated SuperGather technology, which provides the ability to quickly and efficiently read/write from noncontiguous local memory locations. The SuperGather unit enables the full utilization of the available SIMD capabilities for algorithms such as warping, lens distortion correction, and canny edge tracing.

Imaging instructions

The Vision Q6, P6, and P5 DSPs include many imaging specific operations that accelerate 8-, 16-, and 32-pixel data-types and video operation patterns. Some examples of these instructions are arithmetic operations (ADD, SUB, COMPARE, MUL, DIVIDE), bit manipulation operations, and data reorganization operations.

Highly energy efficient

The Vision Q6, P6, and P5 DSPs are highly energy efficient compared to CPUs or GPUs for all kinds of pixel operations.

High performance

The Vision Q6, P6, and P5 DSPs offer a 5-way VLIW architecture, where each VLIW slot can perform 64-way SIMD 8-bit operations. The Vision family is designed to provide 320 operations per clock cycle. The Vision Q6 and P6 DSPs can achieve even higher efficiency with its wide SIMD multiply accumulates, offering significantly enhanced performance for the pixel filtering and image analysis features common in computer vision applications.

OpenCV-like library support

The Vision Q6, P6, and P5 DSPs come with over 1700 OpenCV like functions. These functions are highly optimized to achieve the best performance on these DSPs. While OpenCV has over 2500 functions, Cadence has chosen the most common 1700 functions to optimize. Cadence continues to add more functions with quarterly library updates.

OpenVX 1.1

The Vision Q6, P6, and P5 DSPs are the first imaging/vision DSPs to pass Khronos Group's conformance tests for the OpenVX 1.1 specification.

Application developers can now take advantage of Vision P5 and P6 DSP functionality without detailed knowledge of the hardware architecture and still achieve high performance. Cadence provides an application programming kit (APK) that supports all 40 library functions required by OpenVX 1.1. All of these functions are already fully optimized on the Vision P5, P6, and Q6 DSPs. Applications developed using the standard OpenVX 1.1 API can be compiled and run on Vision P5, P6, and Q6 DSPs without any code changes. Cadence's OpenVX framework automatically schedules and executes the appropriate DMA transfers for efficient memory access, and runs highly optimized DSP vision processing kernels in parallel with the DMA transfers.

AI software support

The Vision Q6 and P6 DSPs support AI applications developed in the Caffe, TensorFlow, and TensorFlowLite frameworks through the Tensilica Neural Network Compiler. The Tensilica Neural Network Compiler maps neural networks into executable and highly optimized high performance code for the target DSP, leveraging a comprehensive set of optimized neural network library functions. The Vision Q6 and P6 DSPs also support the Android Neural Network API (ANN) for on device AI acceleration in Android powered devices.

Rich third-party application software support

Along with math library support, Cadence also supports a very rich set of third-party applications targeting the Vision DSP family. Some of these third-party companies offer video WDR, image stabilization, super resolution, CNN, and various ADAS applications. These applications are ported and optimized on these DSPs for a fast time-to-market.

CPU or FPGA selection for vision project

As multicore CPUs and powerful FPGAs proliferate, vision system designers need to understand the benefits and tradeoffs of using these processing elements. More vision systems that include the latest generations of multi core CPUs and powerful FPGAs reach the market, vision system designers need to understand the benefits and tradeoffs of using these processing elements. They need to know not only the right algorithms to use on the right target but also the best architectures to serve as the foundations of their designs.

Inline vs. co-processing

Before selecting which types of algorithms are best suited for each processing element, a designer should understand which types of architectures are best suited for each application. When developing a vision system based on the heterogeneous architecture of a CPU and an FPGA, a designer needs to consider two main use cases: inline and co-processing.

With FPGA co-processing (see Figure 7.8), the FPGA and CPU work together to share the processing load. This architecture is most commonly used with GigE Vision and USB3 Vision cameras because their acquisition logic is best implemented using a CPU. Acquire the image using the CPU and then send it to the FPGA via direct memory access (DMA) so the FPGA can perform operations such as filtering or color-plane extraction. Then send the image back to the CPU for more advanced operations such as optical character recognition (OCR) or pattern matching. In some cases, implement all of the processing steps on the FPGA and send only the processing results back to the CPU. This allows the CPU to devote more resources to other operations such as motion control, network communication, and image display.

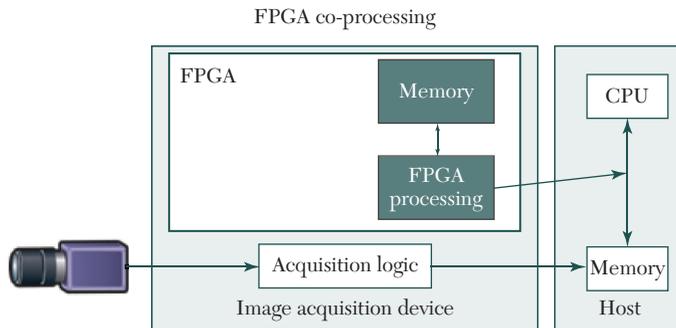


FIGURE 7.8. In FPGA co-processing, images are acquired using the CPU and then sent to the FPGA via DMA so the FPGA can perform operations.

In an inline FPGA processing architecture as shown in Figure 7.9, connect the camera interface directly to the pins of the FPGA so the pixels are passed directly to the FPGA as the designer sends them from the camera. This architecture is commonly used with Camera Link cameras because their acquisition logic is easily implemented using the digital circuitry on the FPGA.

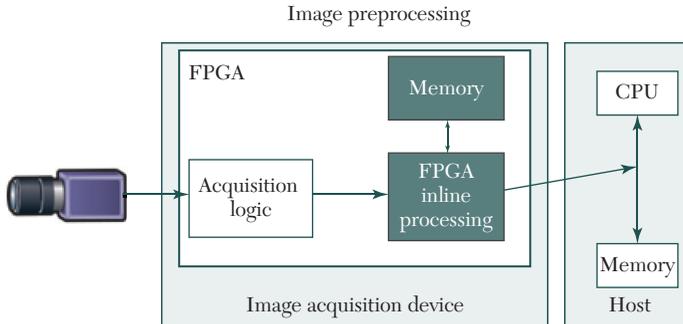


FIGURE 7.9. In the inline FPGA processing architecture, the camera interface is connected directly to the pins of the FPGA so the pixels are passed directly to the FPGA as they are sent from the camera.

This architecture has two main benefits. First, just like with co-processing, designer can use inline processing to move some of the work from the CPU to the FPGA by performing preprocessing functions on the FPGA. For example, use the FPGA for high-speed preprocessing functions such as filtering or thresholding before sending pixels to the CPU. This also reduces the amount of data that the CPU must process because it implements logic to only capture the pixels from regions of interest, which increases the overall system throughput. The second benefit of this architecture is that it allows for high-speed control operations to occur directly within the FPGA without using the CPU. FPGAs are ideal for control applications because they can run extremely fast, highly deterministic loop rates. An example of this is high-speed sorting during which the FPGA sends pulses to an actuator that then ejects or sorts parts as they pass by.

The advantages of an FPGA for image processing depend on each case, including the specific algorithms applied, latency or jitter requirements, I/O synchronization, and power utilization. Often using an architecture featuring both an FPGA and a CPU presents the best of both worlds and provides a competitive advantage in terms of performance, cost, and reliability. Unfortunately, one of the biggest challenges to implementing an FPGA-based vision system is overcoming the programming complexity of FPGAs. Vision algorithm development is, by its very nature, an iterative process. Most of the time, the designer needs to determine which approach works best and “best” is different from application to application. For some applications, speed is paramount. In others, it’s accuracy. At a minimum, designers need to try a few different approaches to find the best one for any specific application.

7.3 SENSORS FOR APPLICATIONS

Sensors for Industrial Applications

Industrial sensors cover almost all of the sensor categories including proximity, position, velocity, level, temperature, force, and pressure. These find application in extremely diverse areas including industrial process control, energy, aviation, safety and security, automobiles, healthcare, and building automation to name a few. Industrial process control involves monitoring and control of machinery, systems, and processes across a large number of industries including chemicals, pharmaceuticals, biotechnology, energy, water, oil and gas, plastic, paper, and food and beverage. The modern industrial processing and manufacturing systems are highly automated, ensuring that raw materials and energy are consumed in an efficient manner. There are many parameters to be monitored and controlled including pressure, temperature, level, flow rate, humidity, dust, corrosives, explosives, liquids, and gases. Sensors are the first element in a process control and measurement system. Precise control and accurate displacement/position measurement in extremely small scales such as nano meters and pico meters has become increasingly significant during the past few years. The main performance criteria for industrial sensors are sensitivity, resolution, compactness, long-term stability, thermal drift, and power efficiency.

1. Proximity sensors

These can be mechanical, optical, inductive, and capacitive. They are widely used in industrial automation like conveyor lines for counting and jam detection, machine tools for safety interlock and sequencing. They are also used in detecting the presence or absence of objects. Mechanical sensors are basically mechanical switches for on/off operations. Optical sensors could be light sources like light emitting diodes (LEDs) and phototransistors. These sensors are essentially noncontact-type with no moving parts. They are small, fast switching, and insensitive to vibration and shock. However, they require alignment, can be blinded by ambient light conditions, and may require a clean, dust- and water-free environment. Inductive and capacitive proximity sensors generally have a short range but are very robust and reliable. Inductive sensors use magnetic properties to detect the presence of metal objects. Capacitive sensors are normally used to detect the capacitance caused by nonmetallic objects. Figure 7.10a shows the capacitive and inductive proximity sensors.

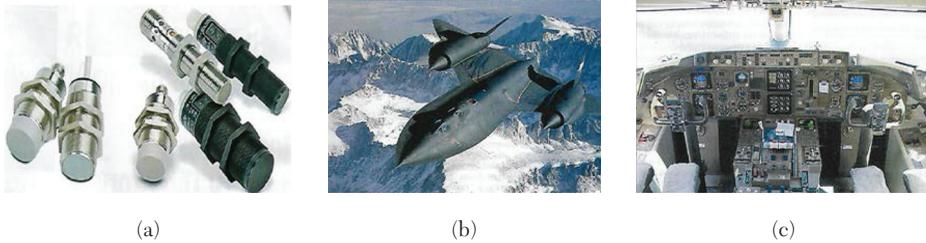


FIGURE 7.10 (a) Capacitive and inductive proximity sensors. (b) SR-71 Blackbird jet aircraft. (c) Cockpit of a Boeing jet.

2. Position sensors

These include encoders and linear variable differential transformers (LVDTs). Encoders are digital sensors commonly used to provide position feedback for actuators. These consist of a glass or plastic disk that rotates between a light source (LED) and a pair of photo detectors. A plastic disk is encoded with alternate light and dark sectors, so pulses are produced as the disk rotates. LVDTs consist of a magnetic core that moves in a cylinder. These are commonly used for position feedback in servomechanisms, automated measurement, and many other industrial and scientific applications.

3. Force sensors

Force and pressure sensors usually act as transducers. They generate signals as a function of the pressure being imposed on them. Force-sensing resistors are two pin-force sensors whose resistance changes when a force, pressure, or mechanical stress is applied on the sensor surface. The FN3050 sensor from TE connectivity is a rugged force-load cell, highly suited for process industry and test bench applications.

4. Vibration/acceleration sensors

Ceramic piezoelectric sensors or accelerometers are most commonly used to detect vibration. Tri-axial accelerometers are used in mobile systems, cars, turbines, and aircrafts. These provide vibration information and position data. Inertial measurement units measure linear and angular motion usually with gyroscopes and accelerometers. These are widely used in aircraft and missile navigation and guidance.

5. Level sensors

These sensors are very common in industrial process control. Selection of a suitable level sensor depends on its size and geometry. Industrial

process control includes hydrostatic and optical level sensors, ranging from simple limit value detection to precision continuous level sensing. Hydrostatic sensors can be installed as submersible sensors for positioning in the fluid or with a screw thread for attachment to the exterior tank wall.

6. Wireless sensors

Wireless sensing of acceleration, vibration, and velocity provides a more effective way of preventing equipment failures. Wireless sensors are not only wearable but also provide advantages of real-time, continuous, long-distance sensing and ease of operation. There are many technologies used for wireless data transmission, including mobile phones, Zigbee, GPS, WiFi, satellite, IR, RF, and Bluetooth. Wireless sensors enable easy access to inaccessible locations, rotating machinery, hazardous or restricted areas, and mobile equipment, which otherwise are difficult to access using wired technology. Wireless sensors are also employed to collect data for environmental monitoring. For example, they can be used for monitoring the temperature inside a refrigerator or monitoring the water level in the overflow tank of a nuclear power plant. Wireless sensors are suitable for monitoring the quality of underground or surface water and preventing wastage of water.

Sensors for Aviation and Aerospace

Accurate feedback systems and ease of control of aircraft systems are enabled by electronic sensors. Sensors help in measuring various parameters like monitoring, control, and navigation. Thus, avionics or electronic systems play an important role in modern aviation. Avionic systems include systems for communication, navigation and display, searchlights, and complex tactical systems for airborne early warning signals. For example, the Lockheed SR-71 Blackbird, the world's fastest and highest-flying operational manned, air-breathing aircraft (Figure 7.10 (b)), is equipped with electronic systems including signal-intelligence sensors, side-looking airborne radars, and camera sensors. If it detects a missile launched by the enemy, it simply accelerates and out flies the missile.

All sensors installed in aircraft are for flight instruments. These include tachometers, engine-temperature gauges, fuel- and oil-quantity gauges, pressure gauges, altimeters, airspeed-measurement meters, vertical-speed indicators, and others. Then, there are sensors for ground testing, flight testing, vibration, environment, angle of attack and static. There are also Doppler radars, lightning-detection radars, terrain radars, anti-collision

warning systems, and stall-warning systems. Many of these instruments and control sensors supply additional signals to cockpit indicators (Figure 7.10c), informing the pilots to take proper action and precaution, and prevent any kind of disaster or accident.

Aircraft computer systems receive data from various sensors, including air-temperature probes, angle of attack probes and pilot static-pressure systems. These computers process inputs from these sensors, apply compensating factors and transmit information to the displays in the cockpit. The pilots continuously monitor the status of the engine and environment from the cockpits.

There are thousands of sensors for different commercial aircraft. It is too broad a category, not to mention the hundreds of models and makes of planes covering different generations of aircraft. This dealt about common sensors used in aircraft for various purposes. Types of sensors are used in aircraft are given below.

1. *Flow sensors*—These are used to monitor the quantities of lubrication oil and liquid coolant fluid in fuel transfer and bleed air systems. Firms like Esterline Corp. and Crane Aerospace & Electronics manufacture liquid sensors and fuel flow sensors. Sensata Technologies Inc. manufactures airflow sensors.
2. *Pressure sensors*—Pressure sensors monitor the pressure in hydraulic systems, braking, raising and lowering landing gear, engine oil, oxygen tanks, heating and coolant fluids. Firms like Esterline Corp. and Custom Control Sensors make pressure sensors for aviation.
3. *Temperature sensors*—Temperature sensors monitor the conditions of hydraulic oils, fuels, refrigerants, and environmental cooling systems. These sensors include bi-metallic temperature gauges, thermometers, wheatstone bridge indicators, ratio meters, and thermocouples. Omega Engineering Inc. makes high precision temperature sensors for reliable, easy to assemble, extended life and preflight applications. Hydra Electric makes pressure and temperature sensors based on latest technology.
4. *Altimeters*—These measure changes in static air pressure to determine the altimeter of the craft. For example, MS5803-02B, based on MEMS technology, is a high-resolution altimeter sensor from TE Connectivity. It includes a high-linearity pressure sensor and an ultra-low power 24-bit $\Delta\Sigma$ ADC with internal factory calibrated coefficients.

5. *Airspeed indicators*—These calculate true air speed based on pilot tube, static pressure, and temperature data.
6. *Position sensors*—Position sensors such as linear variable differential transformers (LVDTs) and rotary variable differential transformers (RVDTs) sense the displacement of aircraft components. Companies that make such sensors include Magnetic Sensors Corp., BEI sensors, and Active sensors.
7. *Oxygen sensors*—Oxygen sensors are at the very heart of the control of inerting systems in Airbus and other civilian aircraft. Different types of oxygen sensors are available from SST Sensing Ltd.
8. *Force and vibration sensors*—Such sensors are used on aircraft to measure torque and force in braking and actuation systems as well as in flight controls.
9. *Compasses and magnetometers*—These are extremely useful for indicating where the aircraft is headed by measuring Earth's magnetic fields.
10. *Gyroscopes*—Gyroscopes are used for direction indication as well as controlling the turning and attitude of aircraft. MEMS gyroscopes are used in modern aircrafts. Watson Industries makes different gyroscopes for the aviation industry.
11. *Attitude heading and reference systems*—Altitude heading and reference systems (AHRSes) have replaced most gyroscopes in modern aircraft. Data from MEMS devices, GPSes, magnetometers and accelerometers, and attitude information are received by AHRSes. Watson industries makea AHRSes for aviation. A typical AHRS module is shown in Figure 7.11.
12. *Tachometers*—Tachometers indicate engine rpm. Tachometer probes are used in turbine engines. These sense the changes in magnetic field flux density, as rotating gear wheels move at the same speed as compressor shafts travelling through the probes magnetic field. Resulting voltage signals are directly proportional to engine speed.

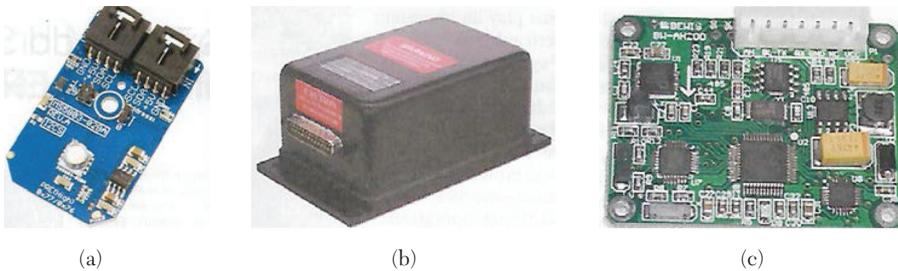


FIGURE 7.11. (a) Altimeter. (b) AHRS. (c) AHRS module.

Other aircraft instruments include aneroid barometers, direction indicators, artificial horizons, attitude indicators, laser sensors, sound sensors, IR and RF sensors, among others. The sensors are either mounted or placed inside the engine to measure various internal and external environmental conditions. These are designed and manufactured with high reliability and specification standards set by the aviation industry. The above mentioned sensors can be found in the following sensor subsystems.

1. *Communication*—Communications connect the flight deck to the ground, and also to the passengers onboard. Very high-frequency (VHF) radio band is used for line of sight communication such as aircraft to aircraft and aircraft to air traffic control. High frequency (HF) radio is used for transoceanic lights or satellite communication. Various electronic sensors for communication and navigation systems are available with Bharat Electronics Ltd.
2. *Navigation*—Navigation is the determination of position and direction on or above the surface of Earth. Avionics may use satellite-based, ground-based, or other systems. Navigation systems calculate the position automatically and display it to the flight crew on moving map displays.
3. *Monitoring*—Glass cockpits or computer monitors are used in modern aircraft, instead of gauges and other analogue displays. Cockpit equipment include control, monitoring, communication, navigation, weather, and anti-collision systems. Honeywell makes advanced monitoring and display systems for aircraft, helicopters, and space vehicles. Ametek makes cockpit indicators and display systems.
4. *Auto control systems*—Modern aircraft have autopilot mode to automatically control flight. Most commercial planes are also equipped with air-

craft flight control systems to reduce pilot errors and workload at landing or takeoff. In helicopters, auto-stabilization is used in a similar fashion. The advent of electromechanical systems has increased safety.

5. *Automatic traffic control*—Automatic traffic control ensures sufficient space between two or more aircraft, either horizontally or vertically, to prevent collisions. Controllers may coordinate position reports provided by pilots. Radars are also used to check positions of aircraft. Central and control towers, oceanic controllers, and terminal controllers enable automatic traffic control.
6. *Collision avoidance systems*—To supplement air traffic control, most aircraft use traffic alert and collision avoidance systems (TCAS) to detect nearby aircraft and prevent midair collisions. On terrain, ground proximity warning systems or radar altimeters are usually employed. Some leading manufacturers of such systems are Rockwell Collins, Honeywell, and Thales Group. A less accurate but inexpensive device is the portable collision avoidance system (PCAS) shown in Figure 7.12 (a). Pilots all over the world use it. It is a passive device similar in function to a traffic alert and collision avoidance system.
7. *Flight recorders*—Commercial aircraft cockpit data recorders, commonly known as black boxes, store flight information and audio from the cockpit. These are often recovered from an aircraft after a crash to determine control settings and other parameters during the incident. Komoline is an Indian firm that manufactures black box recorders (Figure 7.12 b).
8. *Weather systems*—Weather systems such as weather radar and lightning detectors or meteorological instruments are used by pilots to view the weather ahead. Incidences like heavy precipitation or severe turbulence due lightning activity sensed by radars allow pilots to deviate flight paths.

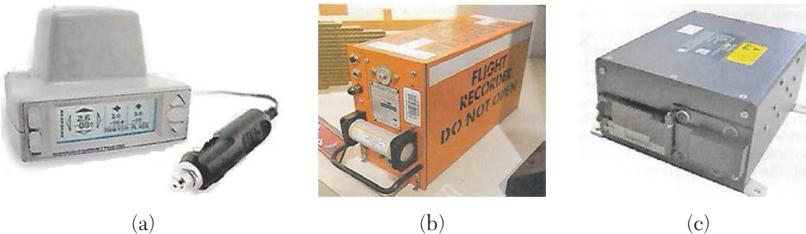


FIGURE 7.12. (a) PCAS. (b) Black box. (c) HUMS for helicopters.

9. *Aircraft management systems*—These provide centralized control of the multiple complex systems fitted in aircraft, including engine monitoring and management systems. Health and usage monitoring systems (HUMS) are integrated with aircraft management computers to give maintainers early warnings of parts that need replacement. It is shown in Figure 7.12 (c).
10. *Military mission-control systems*—Military aircraft are designed to deliver weapons or monitor other weapon systems. The vast array of sensors available to the military is used for any and all tactile means. Bigger sensor platforms have mission management computers. Many electronic sensor components for civil and military applications are available from Bharat Electronics Ltd.

Sensors in aircraft play an important role in monitoring the risks associated with aviation activities, operation of aircraft, controlling aircraft to acceptable levels, air traffic control to communicate with aircraft to help maintain separation between two or more aircraft to prevent collisions, and so on. Hundreds of types of sensors are installed in aircraft to monitor different conditions. These feed information to flight computers and displays for pilots to handle aircraft effectively. Some leading manufacturers of sensors for aviation are Active Sensors, AMETEK Inc., BEI Sensors, Bharat Electronics Ltd., Crane Aerospace & Electronics, Esterline Corp., Honeywell, Hydra-Electric, Komoline Aerospace Ltd., Magnetic Sensors Corp., Meggitt SA, OMEGA Engineering, Oxsensis Ltd., Rockwell Collins Inc., Sensata Technologies Inc., SST Sensing Ltd., Thales Group, and Watson Industries, etc.

Sensors for the Automobile Industry

Modern cars make thousands of decisions based on the data provided by various sensors that are interfaced to the vehicles onboard computer systems. A car-engine management system consists of a wide range of sensor devices working together, including engine sensors, relays, and actuators. Many of these sensors operate in rough and harsh conditions that involve extreme temperatures, vibrations, and exposure to environmental contaminants. These provide vital data parameters to the electronic-control unit (ECU) that governs the various engine functions effectively. Digital computers now control engines through various sensors. Luxury cars have a multitude of sensors for controlling various features as shown in Figure 7.13.

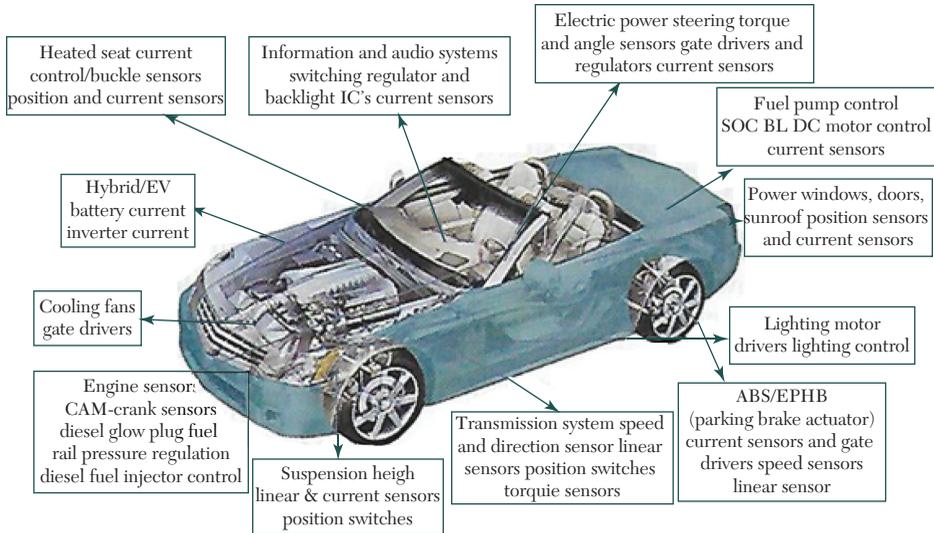


FIGURE 7.13. Sensors used in cars.

Sensors play an important role in the automotive industry. These enable greater degrees of vehicle automation and futuristic designs. For example, at manufacturing facilities, sensorized robotic arms are used for painting car bodies and measuring the thickness of the coatings being applied. Manufacturers can simply monitor the thickness of the paint being sprayed on instruments, airbag claddings, and various internal parts of the vehicles using sensors. Sensors monitor vehicle engines, fuel consumption, and emissions, along with aiding and protecting drivers and passengers. These allow car manufacturers to launch cars that are safer, more fuel efficient and comfortable to drive.

Electronic control unit

All sensors inside the vehicle are connected to the ECU, which contains the hardware and software (firmware). Hardware consists of electronic components on a printed circuit board (PCB) with a microcontroller (MCU) chip as the main component. The MCU processes the inputs obtained from various sensors in real time. All mechanical and pneumatic controls have been replaced by electronic/electrical systems that are more flexible, easier to handle, lighter and cheaper. Moreover, the ECU has reduced the number of wires and emissions, and enabled diagnosing problems with ease. Controlling and monitoring in the modern vehicle is much easier with the ECU (Figure 7.14a).

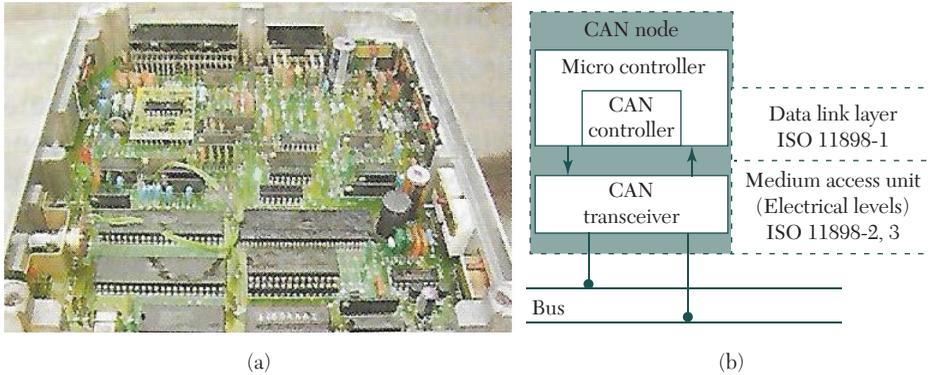


FIGURE 7.14 (a) ECU. (b) CAN bus node

Communication and control

The ECU simplifies the communication between various components and devices, because long wires for each function are not required. It is installed in the vehicle and connected to the nearest vehicle bus, including controller area network (CAN Figure 7.14b), local interconnect network (LIN), FlexRay, and BroadR-Reach, among others. A CAN bus standard is designed to allow MCUs, sensors, and other devices to communicate with each other without a host computer.

Emission control

After sensing fuel level and calculating fuel quantity, the ECU sends signals to various relays and actuators, including ignition circuit, spark plugs, fuel injectors, engine-idling air-control valve, and exhaust gas re-circulation (EGR) valve. Then, it extracts the best possible engine performance while keeping emissions as low as possible.

Engine fault diagnosis

ECU collects signals from various sensors, including faulty ones, and stores these in its memory. Sensors diagnose these faults either by reading ECU memory directly or engine diagnostic equipment supplied by the vehicle manufacturer. Modern luxury cars contain hundreds of ECUs, but cheaper and smaller cars only a handful. The number of ECUs goes up with ever-increasing features. Depending on the vehicle make and model, the ECU(s) can be found beneath the wiper, under the bonnet in engine bay, passenger front foot-well under the carpet, or beneath the glove compartment. Some common vehicle sensors include ambient light,

battery current, differential oil temperature, door open warning, anti-lock braking system (ABS), auto door-lock position, battery temperature, brake power booster, camshaft position, crankshaft position, cylinder-head temperature, diesel emissions-fluid temperature, headlight level, humidity, hybrid battery voltage, hybrid circuit breaker, ignition pass-lock, manifold absolute pressure (MAP), mass air-flow (MAF), oil level, oxygen, power-steering fluid level, speed, steering angle, temperature, throttle position, transmission oil pressure, and windshield washer level. Table 7.3 gives a list of some popular sensors used in modern vehicles.

TABLE 7.3. Sensors Used in Modern Vehicles

Sensor	Function
Engine speed sensor	Monitors spinning speed of crankshaft
Fuel temperature sensor	Ensures right amount of fuel is injected to keep motion smooth
Spark knock sensor	Ensures fuel is burned correctly
Voltage sensor	Manages car speed and ensures speed is controllable
MAP sensor	Measure manifold pressure
Oxygen sensor	Measures unburden oxygen presented in exhaust pipe
MAF sensor	Calculates density and volume of air taken in by engine
Air-fuel ratio meter	Monitors correct air-fuel ratio of engine
Crank position sensor	Monitors piston's top dead center (TDC) position in engine
Cam position sensor	Monitors position of valves in engine
Throttle position sensor	Monitors position of throttle in engine
Knock sensor	Detects engine knocking due to timing advance
Engine coolant temperature sensor	Measures engine temperature

Modern sensors and technology can even help a driverless car to drive at high speeds on open road. Autonomous cars use many sensors including radars and cameras. Lidar is the primary sensor used in most driverless cars. It helps in sensing the world around the vehicle and by bouncing laser light off nearby objects to create 3D maps of the surroundings. Lidar does not detect objects; it profiles these by illuminating them and analyzing the path

of reflected light. It uses emitted light and yields high-resolution images. It is not affected much by the intensity of light at any time (day or night) and, hence, the result is extremely accurate.

Artificial intelligence used in Lidar sensing enables the vehicle to avoid rough paths, collisions, obstacles, potholes, traffic, etc. Therefore it is an indispensable component for fully autonomous cars. Autonomous vehicles demand complex integration of sophisticated algorithms running on powerful processors, making critical decisions based on real-time data coming from a diverse and complex array of sensors. The vehicles need good and reliable sensors including GPS, cameras, MEMS-based gyroscopes, and accelerometers. Some sensors used in advanced driver-assistance systems (ADASes) include fuel delivery, lane departure warning, parking aid, tank pressure measurement, adaptive cruise control (ACC), blind-spot detection, brake booster system, collision avoidance system, filter monitoring, lidar, power-assisted steering, reversing aid, start-stop system, tank air intake and extraction, tank leakage diagnostics, traffic sign recognition, and so on.

Autonomous vehicle sensors

Safety, fast identification, timely action, reliable notification, and warning messages are of prime importance in autonomous vehicles. These are monitored and controlled by embedded MEMS sensors used in autonomous vehicles as shown in Figure 7.15a. Embedded sensor systems include cameras, radars, light-based radar (LiDAR) systems, ultrasonic sensors, wheel-speed sensors, and global positioning systems (GPS).

1. *Camera*—Cameras and radar systems are prerequisites for all levels of automation. Using image sensors, cameras provide information like speed, distance, and outlines of obstacles and moving objects. Autonomous cars use rear, front, and 360-degree cameras. Cameras help the driver with a better representation of their surroundings and the environment outside the vehicle. Normally, four to six cameras are required to obtain realistic 3D images. Front camera systems are used to automatically detect objects within 100–250 meters. The algorithm in cameras can identify pedestrians, motor vehicles, side strips, road margins, and so on. These can also detect traffic signs and signals. A good dynamic range of the camera is required to provide a clear image even when sun rays are falling directly onto the camera lens.

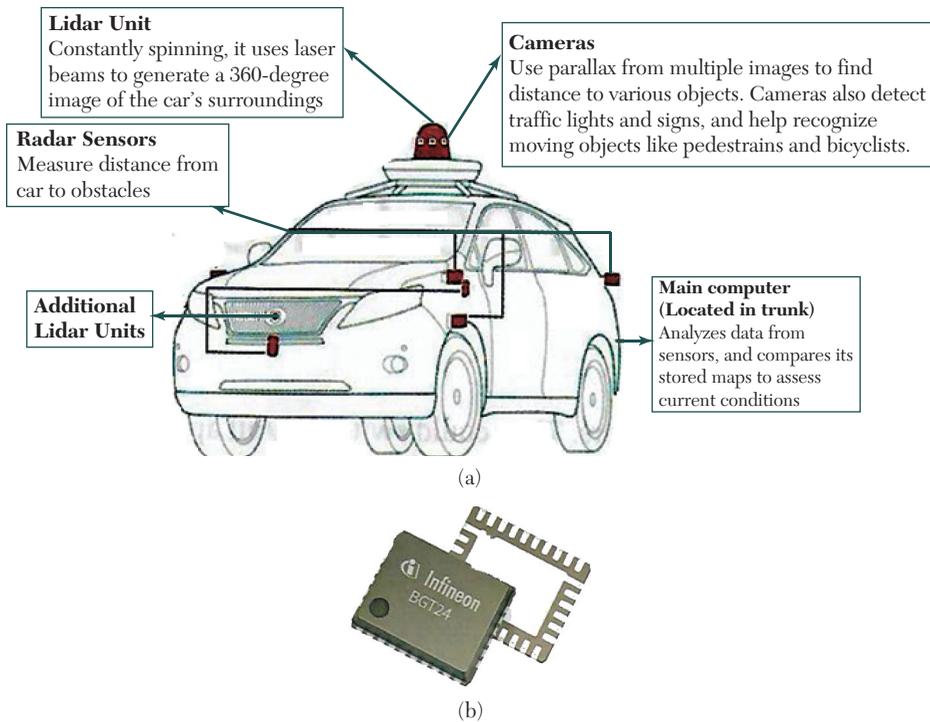


FIGURE 7.15. (a) Sensors used in autonomous vehicles. (b) 24GHz chip for radar front end application.

2. *Radar*—Radar is used for detection and localization of objects using radiowaves. Advanced driver-assistance systems employ many radar sensors making an important contribution to the overall function of autonomous driving. 24GHz BGT24MTR11 for radar front ends application is shown in Figure 7.15b.
3. *Lidar*—Lidar is a laser-based system primarily used to measure distances to stationary as well as moving objects. It calculates the distance of an object from a moving car. It employs special procedures to provide 3D images of the detected objects. The system is a complex mechanical mirror system, normally mounted on top of the autonomous car, which captures spatial images of objects with 360° all-round visibility.
4. *Wheel speed sensor*—This is used to determine the speed at which the car is moving. Usually, it consists of a toothed ring and magnetic pickup used by anti-lock brake systems, traction control, and other systems to indicate wheel speed.

5. *Ultrasound sensor*—These are used for parking and detecting objects very close to the vehicle.
6. *GPS*—This provides the information for accurate positioning and location of the car.

Agricultural Sensors

Smart agriculture, also known as precision agriculture, allow farmers to maximize yields using minimum resources, such as water, fertilizers, and seeds. By deploying sensors and mapping fields, farmers can understand their micro-scale, conserve resources and reduce impact on the environment. Several sensing technologies are used in precision agriculture. These provide data that assist farmers to monitor and optimize crops as well as adapt to changing environmental factors. Some sensing technologies used in precision agriculture are given in the following list.

1. Location sensors use signals from GPS satellite to determine latitude, longitude, and altitude within the required area. A minimum of three satellites are required to triangulate a position. The NJRNJG1157PCDTEI GPS integrated circuit is a good example of location sensor.
2. Optical sensors use light to measure soil properties. These measure different frequencies of light reflectance in near-infrared (IR), mid-IR, and polarized light spectra. Sensors can be placed on vehicles or aerial platforms, such as drones or satellites, to determine clay, organic matter, and moisture content of the soil.
3. Electro-chemical sensors provide key information in the form of pH and soil nutrient levels. Sensor electrodes work by detecting specific ions in the soil. Currently, sensors mounted on specially-designed sleds are helping gather, process, and map soil's chemical data.
4. Mechanical sensors measure soil compaction or mechanical resistance.
5. Dielectric soil moisture sensors assess moisture levels by measuring dielectric constant of the soil.
6. Air-flow sensors measure soil air permeability. Measurements can be made at a single location or dynamically while in motion. Desired output is the pressure required to push a predetermined amount of air into ground at prescribed depth. Various types of soil properties, including compaction structure, soil type, and moisture level, produce unique identifying signatures.

Agriculture weather stations

Agricultural weather stations are self-contained units placed at various locations throughout the agricultural field. These stations have several sensors that are appropriate for local crops and climate. Information such as air temperature, soil temperature at various depths, rainfall, leaf wetness, chlorophyll, wind speed, dew point temperature, wind direction, relative humidity, solar radiation, and atmospheric pressure are measured and recorded at predetermined levels. This data is analyzed and sent wirelessly to a central data server at programmed intervals.

Smartphone tools

A number of smartphone tools can be adapted to farming applications. For instance, crop and soil observations can be logged in the form of pictures, pin-pointing locations, soil colors, water, plant leaves, and light properties. Tools such as cameras, GPSs, microphones, accelerometers, gyroscopes, and smartphone applications can greatly help farmers. Using these, farmers can identify crop diseases and make diagnoses. They can even calculate the amount of fertilizer required, and study the soil and water. Drones are being used in large-scale farming for insecticide and pesticide spraying purposes. For example, the Indian Space Research Organization (ISRO) has developed an Android-based application that collects real-time information to assess the damage caused to agricultural crops due to hail storms. This application allows farmers to process faster insurance claims. It is currently used for rice and cotton crops in the states of Karnataka, Madhya Pradesh, Haryana, and Maharashtra. Mobile-operated solar-based pumps reduce the cost of electricity for farmers. E-Fences help them save their crops from animals like elephants.

Smart Sensors

A sensor is an input device that received and responds to a signal or stimulus. Modern sensors have more features including user friendliness, accessibility, and flexibility. Sensors are becoming more and more intelligent, providing higher accuracy, flexibility, and easy integration into distributed systems.

Intelligent sensors use standard bus or wireless network interfaces to communicate with one another or with microcontrollers (MCUs). The network interface makes data transmission easier while also expanding the system. An intelligent sensor may consist of a chain of analog and digital

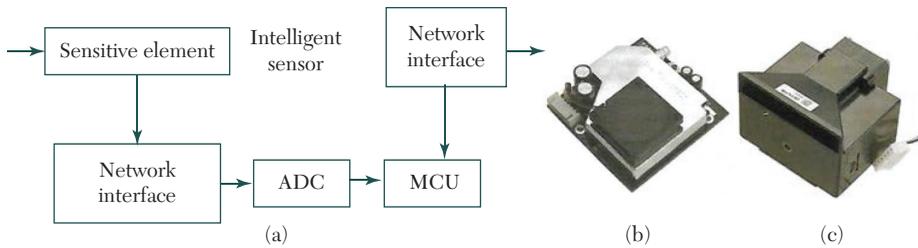


FIGURE 7.16. (a) Intelligent sensor structure. (b) PM2.5 / PM 10 sensor. (c) PM2.5 sensor.

blocks, each of which provides a specific function. Data processing and analogue-to-digital conversion (ADC) functionalities help improve sensor reliability and measurement accuracy. The typical structure of an intelligent sensor is shown in Figure 7.16(a).

There are a wide variety of sensors depending on the technology (analog/digital) and application such as IoT sensors, pollution sensors, RFID sensors, image sensors, biometric sensors, printed sensors, and MEMS and NEMS sensors.

IoT sensor

IoT sensors include temperature sensors, proximity sensors, pressure sensors, RF sensors, pyroelectric infrared (PIR) sensors, water-quality sensors, chemical sensors, smoke sensors, gas sensors, liquid-level sensors, automobile sensors, and medical sensors. These sensors are connected to a computer network for monitoring and control purposes. Using sensors and the Internet, IoT systems have wide applications across industries with their unique flexibility in providing enhanced data collection, automation, and operation.

Pollution sensor

Air pollution sensors are used to detect and monitor the presence of air pollution in the surrounding area. These can be used for both indoor and outdoor environments. Although there are various types of air pollution sensors, most of these sensors focus on five parameters: particulate matter, ozone, carbon monoxide, sulfur dioxide, and nitrous oxide. Sensors capable of detecting particulate matter with a diameter between 2.5 and $10\mu\text{m}$ (PM10) and a diameter less than $2.5\mu\text{m}$ (PM2.5) are available. Figure 7.15(b) shows a typical PM sensor and Figure 7.15 (c) shows PM2.5 sensor with a detection time of ten seconds.

RFID sensors

RFID chips are as small as the size of rice grains and can be inserted directly under the skin for use as ID cards. There is a trend to use RFID chips in many products including contactless banks cards and the UK “Oyster cards.” There are also cases where chips are implanted in pet and cattle for monitoring. It is shown in Figure 7.17(a).

Wearable sensors

These sensors include medical sensors, GPS, inertial measurement unit (IMU), and optical sensors. With modern techniques and miniature circuits, wearable sensors can now be deployed in digital health-monitoring systems. Sensors are also integrated into various accessories such as cloths, wrist bands, eye glasses, head phones, and smartphones. Optical, IMU, and GPS sensors to dominate the market in terms of revenue by 2022 as shown in Figure 7.17(b).

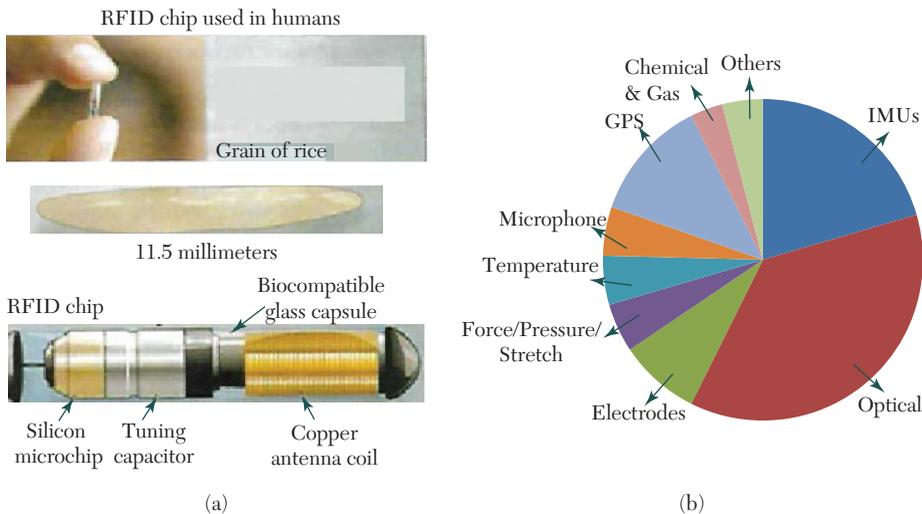


FIGURE 7.17. (a). Grain-size RFID chip. (b) Wearable chart.

Image sensors

It is found commonly in smartphone camera. An image sensor detects and conveys the information that constitutes an image. Digital imaging is fast replacing analog imaging. Digital cameras use CMOS sensors, which allow faster speed with lower power consumption. A Renesas 8.48 MP High Sensitivity CMOS Image Sensor for 4K Video Network Cameras is shown in Figure 7.18(a).

Biometric sensors

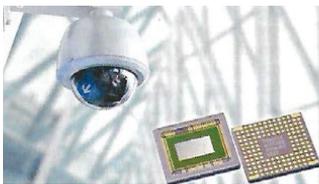
The most common biometric sensor is the fingerprint module. The latest fingerprint sensor consists of sensors for display, glass and metal, detection of directional gestures, and underwater fingerprint match and device wake-up.

Printed sensors

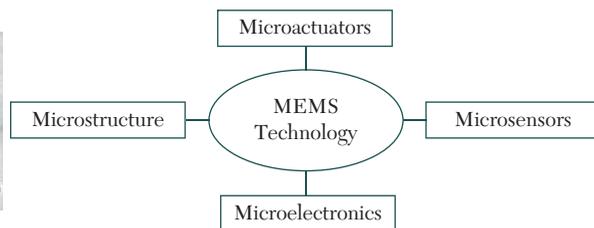
Sensors printed on flexible substrates are popular and enable applications ranging from human-machine interfaces to environmental sensing. Printed sensors may have a very simple structure with only a few electrodes and capability to be manufactured on plastic substrates. They offer advantages in terms of mechanical flexibility, thinness, and weight reduction.

7.4 MEMS

Microelectromechanical systems (MEMS) are devices characterized both by their small size and the manner in which they are made. They are made up of component sizes between 1 and 100 micrometers. The most notable elements are microsensors and micro-actuators. MEMS devices can vary from simple structures to extremely complex electro-mechanical systems with multiple moving elements under the control of integrated microelectronics. In other words, the MEMS sensor is a precision device in which mechanical parts and microsensors along with a signal-conditioning circuit are fabricated on a small piece of silicon chip. Generally, MEMS consist of mechanical microstructures, microactuators, microsensors, and microelectronics in one package. Figure 7.18(b) shows the block diagram of a MEMS device.



(a)



(b)

FIGURE 7.18. (a) Image sensor. (b) MEMS device.

Microsensors detect changes in a system's environment by measuring thermal, chemical, electrical, or mechanical information. These variables are processed by microelectronics and then microactuators act according to the changes in the environment. Some common types of MEMS sensors available are:

1. MEMS accelerometers—These are used to measure static or dynamic force of acceleration. The major categories are silicon capacitive, piezoresistive, and thermal accelerometers. These are used in smartphones for various controls including switching between landscape and portrait modes, anti-blur capture, and pocket-mode operation.
2. MEMS gyroscopes—These detect the angular rate of an object. MEMS gyros are used for vehicle stability control with a steering-wheel sensor and rollover detection.
3. MEMS pressure sensors—These sensors measure three types of pressures: gauge, absolute, and differential pressure. The sensor is integrated with a diaphragm and a set of resistors on integrated chips so that pressure is detected as change in resistance. These sensors are used in automotive, industrial, medical, defense, and aerospace applications. In automotive systems, these are widely used in oil-pressure sensor, crash detection, fuel-tank vapor pressure monitoring, exhaust gas recirculation, engine management system, and so on.
4. MEMS magnetic field sensors—These sensors detect and measure magnetic fields, and find use in position sensing, current detection, speed detection, vehicle detection, space exploration, and so on.
5. Fluxgate sensors—Fluxgate sensors are used to measure DC or low-frequency AC magnetic fields. These find many applications like space research, geophysics, mineral prospecting, automation, and industrial process control. MEMS-based fluxgate sensors consume less power, are small in size, and provide better performance.

NEMS

Nanoelctromechanical systems (NEMS) are a class of device like MEMS but on the nanoscale. These are the next miniaturization step after MEMS devices. Nanoresonators and nanoaccelerometers are examples of NEMS. Usually, NEMS rely on carbon-based materials, including diamond, carbon nanotubes, and graphene. One of their most promising applications is the combination of biology and nanotechnology. Nanoresonators would

find application in wireless communication technologies, while nanomotors might be used in nano-fluidic pumps for biochips or sensors.

Biosensors

Biosensors are analytical devices that combine a bioreceptor (biological recognition element) and a transducer. The bioreceptors can be organisms, tissues, cells, enzymes, antibodies, nucleic acids, and so on. It detects the target analyte. The transducer can be electrochemical, optical, thermal, or mechanical in nature. It converts the detected analyte into a measurable signal. Therefore biosensors involve cross-functional interaction among disciplines such as electronics, electrical engineering biology, and chemistry. In a typical biosensor, the biological recognition unit interacts with biological samples / bio-elements, which include enzymes, living tissues, and antibodies. Subsequently, transducers transform the signals generated from this interaction into an electrical signal.

The application segment comprises food toxicity detection, industrial process control, medicine/diagnostics/ point-of-care (POC) testing and other application areas such as environment and agriculture. POC applications can be segmented into glucose monitoring, cardiac markers, infectious disease diagnosis, coagulation monitoring, pregnancy and fertility testing, blood gas and electrolytes testing, tumor or cancer markers, urinalysis testing, cholesterol tests, and so on. The elements of a biosensor are shown in Figure 7.19.

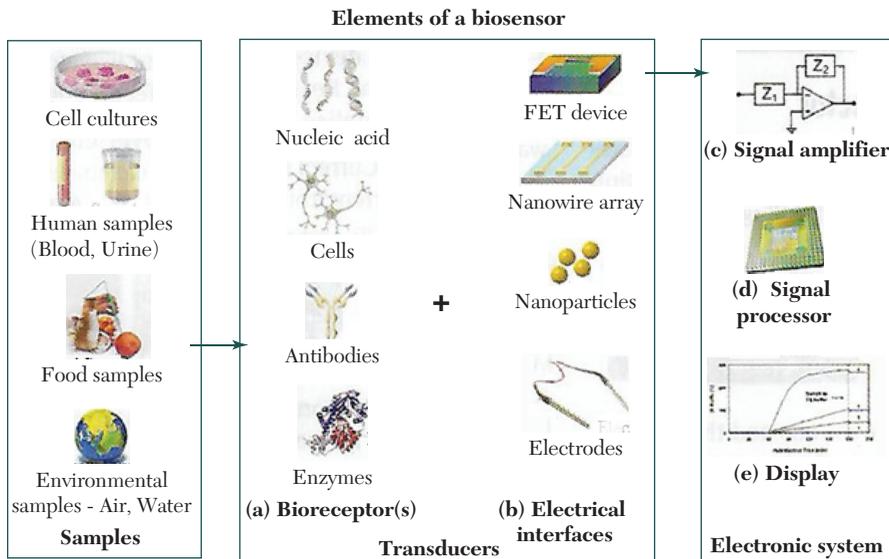


FIGURE 7.19. Elements of a biosensor.

Medical Sensors

A medical device is used to diagnose, monitor, or treat diseases, or as a supportive aid for physically disabled persons. Various types of sensors are used in medical applications, including temperature probes, force sensors in kidney dialysis machines, airflow sensors in anesthesia delivery systems, and pressure sensors in infusion pumps and sleep apnea machines. An implanted pacemaker is a real-time embedded sensor system that delivers a synchronized rhythmic electric stimulus to the heart muscle in order to maintain an effective cardiac rhythm. Sensors/electrodes of pacemakers detect the heart's electrical activity and send data through wires to the computer. So, these are life-saving and safety-critical medical applications of sensors. A typical pacemaker is shown in Figure 7.20(a).

A high-volume pressure sensor is shown in Figure 7.20(b). It has integrated thin-film temperature compensation and calibration, and it's based on piezo resistive technology. It is intended for use in medical diagnostics, infusion pumps, blood-pressure monitors, pressure catheter applications, and patient monitoring. Digital airflow sensors featuring precise measurement, fast response time, high accuracy, high stability, and high sensitivity are suitable for critical medical applications like anesthesia delivery machines, laparoscopy, and heart pumps. These support ASIC-based I2C digital output, which allows easy integration to microprocessors or microcontrollers.

Nuclear Sensors

Alpha, beta, and gamma rays, as well as neutrons are the most common forms of ionizing radiation. Sensors and detectors play a major role in ensuring safety of nuclear plant workers. Solid-state sensors directly convert the incident radiation into electrical current using materials such as silicon,

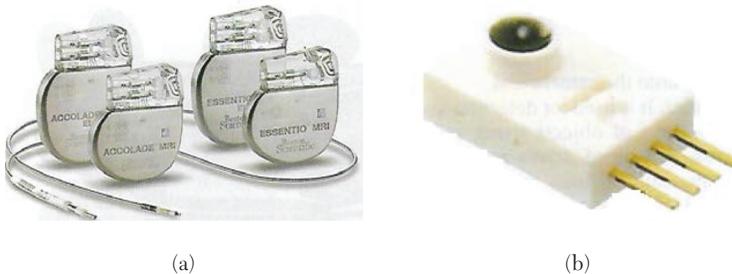


FIGURE 7.20. (a) Typical pacemaker. (b).MPX2300DT1 pressure sensor.

germanium, and cadmium zinc telluride. These sensors have high-energy resolutions, which makes them suitable for detecting the exact amount of radiation energy. Alpha, beta, gamma, and X-ray radiations can be detected using silicon PIN photodiodes. Figure 7.21 shows PIN diode based X100-7 direct absorption detector for nuclear applications.

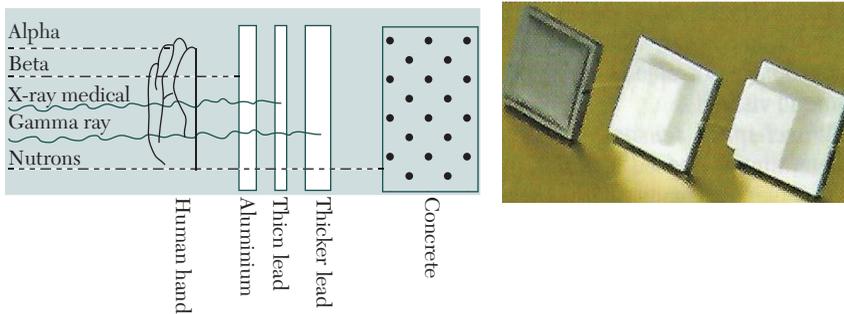


FIGURE 7.21. Penetration power of different radiations and X100-7 gamma radiation detector.

Sensors for Deep-Sea Applications

Modern electronic instruments in oceanography have helped humans explore and observe the greatest depths of the ocean. Electronic systems with low-power, accurate, and delicate sensors collect vital information, which has helped researchers measure and record the deep sea environment and tidal data.

Nereus is the world's deepest driving underwater vehicle. It can be configured to operate as a remotely operated vehicle (ROV), or operate independent of human control as an autonomous underwater vehicle (AUV). Nereus is capable of exploring and mapping the sea floor using sensors and cameras as shown in Figure 7.22(a).

A wide variety of sensor instruments are used in the ocean, including acoustic Doppler current profiler, benthic flow meter, bottom pressure and tilt meter, conductivity temperature depth (CTD), dissolved oxygen sensor, digital still camera, high-definition video camera, hydrophone, mass spectrometer, optical attenuation sensor, pH and carbon dioxide sensor, pressure sensor, remote-access fluid and DNA sampler, resistivity probe, seismometer, sonar, thermistor array, and turbulent-flow current meter. Let us explore various types of deep sea sensors.

1. *DO sensor*—A dissolved oxygen (DO) meter assesses water quality and measures the amount of dissolved oxygen in a liquid, and is one of the most important instruments because of its influence on the organisms living in water. DO sensor for depths of up to 6000 meters consists of a preamplifier covered by a titanium housing, as shown in Figure 7.22(b). This complete sensor is used to interface CTD probe systems.
2. *Pressure sensor*—Seismic activity under the water is monitored by placing pressure sensors on the sea floor. Piezoelectro (quartz crystals) sensors produce electric charges when placed under pressure. These can measure the pressure or the weight of the water above. The pressure of different sections along a fault-line determine where tectonic plates are locked up or are trying to move past each other. Pressure builds up between the plates, resulting in earthquakes when these break free. Pressure sensors are prone to drift and lose accuracy over time. Low-cost self-calibrating pressure sensors can be deployed on the sea floor to monitor long-term seismic activity. Data from deep-sea sensor networks helps scientists understand what happens along fault lines. A pressure sensor is shown in Figure 7.22 (c).
3. *pH sensor*—Increasing atmospheric carbon dioxide is driving a long-term decrease in ocean pH. Measuring pH in sea water with glass electrodes demands a special sensor construction to avoid mistakes caused by high ionic strength of sea water. The ion sensitive field effect transistor (ISFET) pH sensor can be immersed directly in sea water. It is capable of reporting pH with good accuracy.
4. *Sonar*—Remote underwater observation is much easier now, because sound navigation and ranging (sonar) technology has become much more advanced. Both passive and active sonar are used in modern naval warfare from waterborne vessels, aircraft, and fixed installations. Submarines depend on sonar to a great extent for underwater communication.

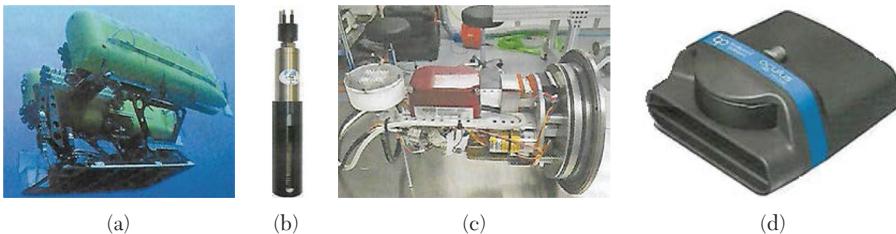


FIGURE 7.22. (a) Nereus. (b) DO sensor. (c) Pressure sensor. (d) Multi-beam sonar.

A new generation of multi-beam sonar designated for use across a wide variety of underwater applications is shown in Figure 7.22(d).

5. *Imaging sensor*—Underwater photogrammetry in the deep sea is different from that of land or in space. Rough conditions, high pressures, absence of natural light, and refraction are some problems. Attenuation and scattering degrade the radiometric image quality and limit effective visibility. An optical system for AUVs for the purpose of visual mapping of large areas of the seafloor up to 6000 meters has been developed. Schematic overview of the components of the high-altitude camera system for GEOMAR Remus 6000 AUV is shown in Figure 7.23.

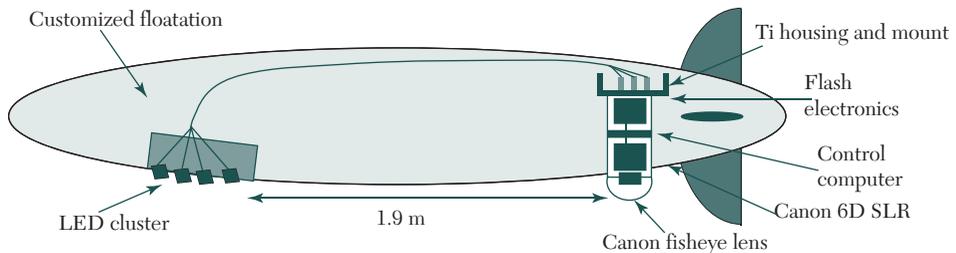


FIGURE 7.23. Schematic overview of GEOMAR Remus 6000 AUV.

6. *Underwater sensor network*—Deep sea exploration requires a different approach for communication as compared to shallow water communication. Underwater applications include monitoring, disaster management, military, navigation, and sports. Underwater sensor network (UWSN) is used for underwater explorations. It is a network of autonomous sensor nodes that are spatially distributed underwater to collect temperature, pressure, and other water-related data. UWSN acoustic transceivers used for communication are low-frequency waves, but have long wavelengths suitable for long-distance communications. UWSN finds use in a wide range of applications for management and recovery of disaster monitoring and preventive mechanisms. UWSN with acoustic sensor networks and underwater mobile ROVs, AUVs sensor networks, and embedded vision can be used on a large scale for exploration in the deep sea.

Various environmental conditions of the sea, including temperature, pressure and currents can be easily monitored with sensors and image sensor camera. Modern electronic sensors, microelectronics, nanotechnology,

powerful computers, and high-bandwidth cables in the oceans will likely lead to unprecedented advancements in ocean exploration and sensing in the next few decades. The deep sea needs to be explored not only for extracting rich resources for human needs, but also for determining earthquakes and other related natural disasters for human safety. Underwater sensor networks along with vision systems and ROVs and AUVs will help humans explore deeper in the sea.

Sensors for Security Applications

CCTV surveillance sensors are installed in many industrial setups and high-risk security areas to monitor and guard against threats. There is a wide range of national security systems to guard against rogue states and terrorism, and monitor weather reports and natural calamities such as cyclones, earthquakes, and tsunamis. A radar sensor is one of the most important types of electronic security system. A radar transmits an electronic signal that bounces off objects and returns to the receiver for analysis. A radar sensor monitors areas such as national and international borders, military bases, airports, seaports, refineries, and other critical industries. It detects movements at ground level, such as individuals walking or crawling toward a specific area, from a range of several kilometers. Radar sensor is also used on aircrafts, ships, and submarines.

Satellite is another vital electronic system used in national security. Unlike radar, it is placed in an orbit above the Earth and uses cameras to take pictures of the Earth. Meteorologists use weather satellites and radars to monitor and forecast weather and atmospheric measurements, and provide important information about rain and snow, among others. Optical sensors like lidar (light detection and ranging) along with unmanned aerial vehicles (UAVs) or drones could be used for efficient and reliable security applications. A lidar sensor mounted on a UAV along with sophisticated software can process images for analysis.

Home security

Motion sensors and CCTVs are the two most common home security systems. Motion flood-lights trigger light when these sensors sense motion. A motion guard-dog sounds like the barking of a dog when someone approaches. Speed, volume, and type of the bark may vary depending on how far the detected motion is. There are different types of motion sensors available in the market, including passive infrared (PIR), ultrasonic, and tomographic.

PIR sensors—All warm-blooded creatures including humans emit IR radiation. A PIR sensor triggers a burglar alarm when it detects a human or an animal. It is common in indoor alarms.

Ultrasonic sensors—These can be active or passive. Active ones emit pulses of ultrasonic waves and then measure reflected signals from a moving object. Animals can hear ultrasonic frequencies and so these signals may drive them away.

Tomographic sensors—These emit radio waves and sense when those waves are disturbed. These can detect waves through walls/objects, and are often positioned in a way that creates a radio wave around the area. These sensors are useful for warehouses and large storage units.

Smart-home security systems—Smart security systems work in a seamless environment and can be manipulated using customized rules as shown in Figure 7.24. These communicate with one or more wireless protocols such as Wi-Fi, Z-Wave, Zigbee, or a proprietary mesh network. A smart-home security system connects to the home Wi-Fi network to monitor and control security devices using a smartphone. Using an app, one can monitor motion, windows, door locks, indoor and outdoor surveillance cameras, lights, sirens, smoke, and more.

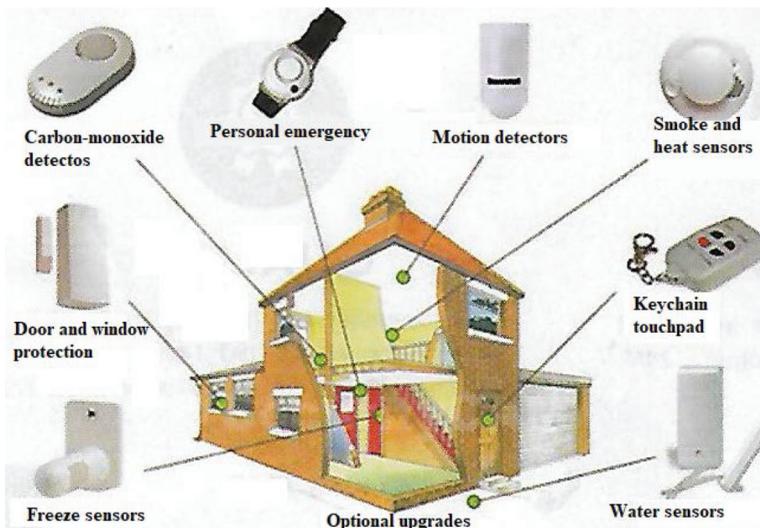


FIGURE 7.24. Typical home-security systems (<http://armetalarm.com>).

Personal security

The main goal of personal security alarms is to make a loud sound and attract attention in an emergency, and alert those nearby that the user is in danger. These gadgets are small, wireless, portable, and easy to conceal. Personal alarms are an excellent self-defense solution, especially for women and children.

Panic alarms—These electronic devices can be attached to almost anything. The sensor unit activates a loud alarm and flash light. These devices are lightweight, compact, and portable, and can be used to scare off potential intruders and prevent robberies.

Stun guns—Electronic stun devices often come with multiple features including flash LED light and strobe function, and deliver a powerful electrical shock siren to deter threats. These may also have safety-pin ropes that prevent others from using the devices against the person.

Child locators—These wireless systems consist of transmitter and receiver units. The range of the transmitter is about 45.7m (150 feet) and can activate a sound alarm in the receiver unit. Most come with a wrist strap, which makes it easy to attach the receiver to shoelaces, belts, or wrists.

Mobile SOS buttons—These are portable devices and can be worn around the neck by disabled people as a necklace or on the wrist as a bracelet. On pressing the button, these automatically call a landline or mobile number persistently until the call is answered. The unit then allows the person to talk through the pendant or bracelet.

Implantable microchips—Use of microchips in livestock allows farmers to track animals. This technology can also be used for human beings. It can be particularly useful for parents who are concerned about the safety of their children. The microchip implanted in the child's body can transmit information related to the location of the child and alert parents of any danger.

Industrial security

Apart from CCTV systems, there are other security systems meant for industrial setups and related establishments. Due to the variety and complexity of commercial setups and industries, it is important to go beyond basic physical security, from simple to very complex systems. Emerging technologies, industry, an industrial IoT have given way to many security threats, including cyber-attacks. Industrial security systems may include fire

alarms, chemical sensors, access control systems, video surveillance units, and intrusion detection systems, for a complete solution for protecting workers and their assets.

Biometric access control systems—These are fingerprint access and time attendance control systems mostly found in industries, commercial establishments, and offices. These use fingerprints to first register in the database for authentication.

Proximity access control systems—RFID-based proximity access systems are normally used in offices, factories, banks, and so on. These are inexpensive, quick, and easy to use for door- and gate-entry systems. Sometimes these are even more effective than video surveillance.

Chemical sensors—An array of chemical sensors is used to detect organic compounds present in gases. Some of these sensors are used in homeland security, analysis, radio-frequency detection, sensing toxic industrial materials, bomb detection, toxic vapors, and chemical agent simulants.

Magnetic sensors—These sensors are used in many security and military systems. Traditional sensors are complemented by new sensor-types such as anisotropic magneto resistor (AMR), giant magneto-resistance (GMR) and giant magneto-impedance (GMI) sensors. These are used for the detection of ferromagnetic and conducting objects, navigation, position tracking, and in anti-theft systems.

Selection Criteria for Sensor

The criteria below provide a frame of reference for how to correctly pair sensors with their specific applications:

1. Consider what is being sensed: Are you striving to sense a process parameter (temperature pressure, flow), an object's presence, the distance to the target, or the position of a mechanism?
2. Environmental condition: Is the sensor suited for the environmental condition which it will inhabit? What are the unique environmental conditions?
3. Range: What is the measurement limit of the sensor? Will the target sit within range?
4. Control interface: What type of controller interface and switching logic is required?

5. Resolution: What is the most granular increment detected by the sensor?
6. Composition of the target: What is the material composition of the substance that will be sensed? Is it metal, is it plastic?
7. Repeatability: Is the variable sensed consistently measured under the same environment?
8. Form factor: How much physical space is available for the sensor and what form best fits the application?
9. Special requirements: Newly added components might create new conditions to be considered. For example, piezoelectric vibration transducers succeed at converting mechanical energy into electrical by stressing a piezoelectric crystal. Its electrical energy output is a direct function of the transducer design and the mechanical energy input. As a result, the crystal may need to be protected against vibrations exceeding beyond a set threshold.
10. Protection class, voltage range, discrete or analog output, response speed, sensing range, electrical connection, mounting type, and size are the parameters that need to be considered for the selection of a proper sensor for the vision applications.

Summary

- Embedded vision processors are programmable low-power high-performance DSP CNN CPU.
- Embedded vision applications typically require very high performance, programmability, low cost and energy efficiency processors for its operations.
- Data rate, memory requirements, number of processors, software support, environment aspects, and size are the selection parameters for processors.
- Convolution neural networks are based on a deep learning algorithm that is trained with many images of an object and can be used by the algorithm to find the object in pictures or video.
- Movidius EV52 is a dual-core 32-bit CPU with programmable CNN object-detection engine which is user configurable.

- Matrox RadiantPro CL, Raspberry Pi, Nvidia Jetson TX1, Nvidia Jetson TK1, BeagleBone Black, Orange Pi, ODROID C2, and Banana Pi are few boards for EV purpose.
- All sensors in a vehicle are connected to an electronic control unit which contains hardware and software.
- Location sensors use signals from satellites to determine latitude, longitude, and altitude.
- IoT sensors include temperature sensors, proximity sensors, pressure sensors, RF sensors, pyroelectric infrared (PIR) sensors, water-quality sensors, chemical sensors, smoke sensors, gas sensors, liquid-level sensors, automobile sensors, and medical sensors.
- Wearable sensors include medical sensors, GPS, inertial measurement units (IMU), and optical sensors.
- Microelectromechanical systems (MEMS) are devices characterized both by their small size and the manner in which these are made.
- Biosensors are analytical devices that combine a bio-receptor (biological recognition element) and a transducer.
- A medical device is used to diagnose, monitor, or treat diseases, or as a supportive aid for physically disabled persons.

References

<http://www.vision-systems.com/articles/print/volume-20/issue-9/features/>

<https://www.movidius.com/myriadx>

<https://www.stemmer-imaging.com/en/knowledge-base/computers/>

<https://ip.cadence.com/vision>

Learning Outcomes

- 7.1 Explain embedded vision processors with the help of blocks.
- 7.2 Write about CPU or FPGA architecture selection.
- 7.3 List the processor type selected for vision applications.
- 7.4 How will you select a suitable processor for application areas?
- 7.5 Write the importance of a CNN module.

- 7.6 Draw an Intel Movidius Myriad X block diagram.
- 7.7 List the features of Matrox RadientPro CL.
- 7.8 What are the advantages of Raspberry Pi?
- 7.9 What are different boards available for EV application?
- 7.10 Draw a block diagram of CEVA-XM4.
- 7.11 What are the features of MAX10 FPGA?
- 7.12 Write about the different vision DSPs for imaging and vision.
- 7.13 Give the blocks in Vision P6 DSP.
- 7.14 What is meant by SuperGather technology in DSP?
- 7.15 Which is suitable for vision, CPU or FPGA?
- 7.16 Differentiate FPGA co-processing and inline FPGA architecture.
- 7.17 List sensors for industrial applications.
- 7.18 Write about sensors for aviation and aerospace.
- 7.19 Write a short note about sensors for automobile industry.
- 7.20 Give the importance of ECU in vehicles.
- 7.21 List different sensors used in modern vehicles.
- 7.22 Write about agricultural sensors.
- 7.23 Write a short note on RFID sensors and MEMS device.
- 7.24 Write about available MEMS sensors.
- 7.25 Sketch the elements of biosensor.
- 7.26 Write a short note on medical sensors.
- 7.27 What is a smart sensor?
- 7.28 Explain the deep sea application sensors and security sensors.
- 7.29 What are the selection criteria for a sensor?

Further Reading

Computer Vision: Specialized Processors for Real Time Image Analysis
by Eduard Montseny and Joan Frau.

COMPUTER VISION

Overview

Computer vision is an area of studies to make computers efficiently perceive, process, and understand visual data such as images and videos. Robot vision involves using a combination of camera hardware and computer algorithms to allow robots to process visual data from the world. Embedded vision needs knowledge of different fields such as embedded system, computer vision, image processing, robotics, signal processing, cameras, sensors, physics, mathematics, imaging, AI, machine learning, and machine vision.

Learning Objectives

After reading this one will know the

- different technologies associated with embedded vision,
- components of computer vision,
- algorithms for feature extraction, optical flow, tracking, machine learning, object detection, etc.,
- semantic segmentation recognition, instance segmentation, object recognition algorithms,
- few computer-vision applications, and
- robotic vision application areas, sensors, and testing in different industries.

8.1 EMBEDDED VISION AND OTHER TECHNOLOGIES

Humans use all parts of their body and their brains to see, visually sense, and realize the world around them. Embedded vision is the science that aims to give a similar, if not better, capability to a machine or computer. Embedded vision combines computer vision and embedded systems. Knowing computer vision is important when we design embedded vision products. *Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images.* It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding. It needs knowledge from the fields of computer science, electrical engineering, mathematics, physiology, biology, and cognitive science in order to understand and stimulate the operation of human vision system.

Computer vision is a field of artificial intelligence and computer science that aims at giving computers a visual understanding of the world, with the help of powerful algorithms. It is one of the main components of machine understanding. Embedded vision combines image sensors, sensors, and circuitry along with computer vision algorithms.

Computer vision is an area of studies to make computers efficiently perceive, process, and understand visual data such as images and videos. The ultimate goal is for computers to emulate the striking perceptual

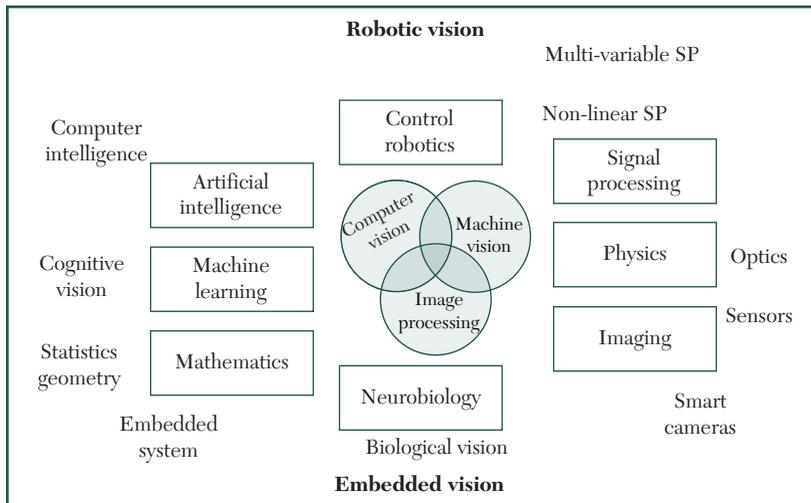


FIGURE 8.1. Various fields involved in embedded vision.

capability of human eyes and brains, or even to surpass and assist the human in certain ways.

Computer vision overlaps significantly with the fields of image processing, pattern recognition, and photogrammetry. Image processing focuses on image manipulation. Pattern recognition studies various techniques to classify patterns. Photogrammetry is concerned with obtaining accurate measurements from images. The fields involved in embedded vision are shown in Figure 8.1.

Robot Vision

In basic terms, robot vision involves using a combination of camera hardware and computer algorithms to allow robots to process visual data from the world (Figure 8.11 (d) and (e)). For example, the system could have a 2D camera which detects an object for the robot to pick up. A more complex example might be to use a 3D-stereo camera to guide a robot to mount wheels onto a moving vehicle. Without robot vision, the robot is essentially blind. This is not a problem for many robotic tasks, but for some applications robot vision is useful or even essential. Robot vision is closely related to machine vision and embedded vision. They are all closely related to computer vision. In a family tree, computer vision could be seen as their “parent,” and the “grandparent” is signal processing as shown in Figure 8.2 (a).

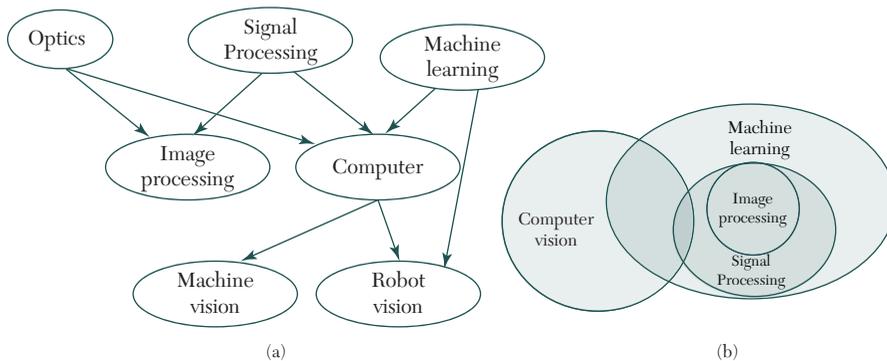


FIGURE 8.2. (a) Family tree. (b) Comparison of all technologies.

Robot vision and machine vision are used interchangeably. However, there are a few subtle differences. Some machine vision applications, such as part inspection, have nothing to do with robotics—the part is merely placed in front of a vision sensor that looks for faults. Also, robot vision is not only an engineering domain. It is also a science with its own specific areas of research. Unlike pure computer vision research, robot vision must

incorporate aspects of robotics into its techniques and algorithms, such as kinematics, reference frame calibration, and the robot's ability to physically affect the environment. Visual servo is a perfect example of a technique that can only be termed robot vision, not computer vision. It involves controlling the motion of a robot by using the feedback of the robot's position as detected by a vision sensor. *Robot vision is a set of algorithms that renders vision to the robotic components.* At its core, robot vision is a combination of computer algorithms, cameras, and other hardware components that work unanimously in order to provide visual insights to the robot or machine. This helps the robot to accomplish complex tasks that require visual understanding.

Signal Processing

Signal processing involves processing electronic signals to either clean them up (e.g., removing noise), extract information, prepare them to output to a display, or for further processing. Anything can be a signal. There are various types of signals which can be processed, for example, analog electrical signals, digital electronic signals, frequency signals, and so on. Images are basically just a two- (or more) dimensional signal. Signal processing is an umbrella and image processing lies under it. The amount of light reflected by an object in the physical world (3D world) is passed through the lens of the camera and it becomes a 2D signal and hence result in image formation. This image is then digitized using methods of signal processing and then this digital image is manipulated in digital image processing.

Image Processing

Image processing techniques are primarily used to improve the quality of an image, convert it into another format (like a histogram), or otherwise change it for further processing. Computer vision, however, is more about extracting information from images to make sense of them. So, image processing is used to convert a color image to grayscale, and then computer vision is used to detect objects within that image. In a family tree (Figure 8.2 a), both of these domains are heavily influenced by the domain of physics, specifically optics.

Pattern Recognition and Machine Learning

Pattern recognition branch of the family is focused on recognizing patterns in data of robot vision. For example, to be able to recognize an object from its image, the software must be able to detect if the object it sees is similar to previous objects. Machine learning, therefore, is another

parent of computer vision alongside signal processing. However, not all computer vision techniques require machine learning. Machine learning is used on signals that are not images. In practice, the two domains are often combined like this: computer vision detects features and information from an image, which are then used as an input to the machine learning algorithms. For example, computer vision detects the size and color of parts on a conveyor belt, then machine learning decides if those parts are faulty based on its learned knowledge about what a good part should look like. Pattern recognition involves study from image processing and from various other fields that include machine learning (a branch of artificial intelligence). In pattern recognition, image processing is used for identifying the objects in an image and then machine learning is used to train the system for the change in pattern. Pattern recognition is used in user-aided diagnosis, recognition of handwriting, recognition of images, and so on.

Machine Vision

Machine vision refers to the industrial use of vision for automatic inspection, process control, and robot guidance. The rest of the “family” are scientific domains, whereas machine vision is an engineering domain. In some ways, it is a child of computer vision because it uses techniques and algorithms for computer vision and image processing. But, although it’s used to guide robots, it’s not exactly the same thing as robot vision.

Computer Graphics

Computer graphics deals with the formation of images from object models, rather than the image being captured by some device. For example, Object rendering is when an image is generated from an object model (See Figure 8.3 (a)).

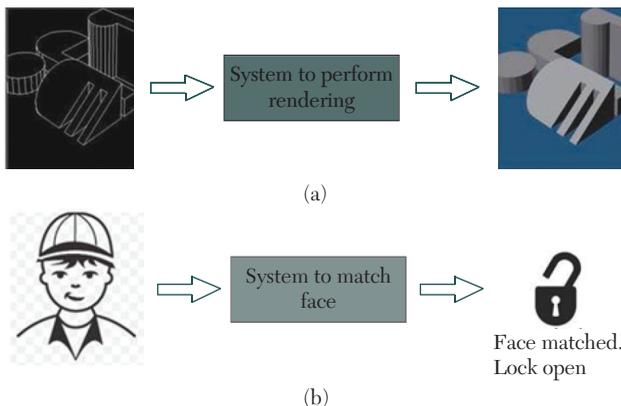


FIGURE 8.3. (a) Computer graphics. (b) Machine / computer vision.

Artificial Intelligence

Artificial intelligence is more or less the study of putting human intelligence into machines. Artificial intelligence has many applications in image processing. For example: developing computer aided diagnosis systems that help doctors in interpreting images of X-rays, MRI, and so on, and then highlighting conspicuous or clearly visible sections to be examined by a doctor.

Color Processing

Color processing includes the processing of colored images and different color spaces that are used. For example, RGB color model, YCbCr, HSV. It also involves studying transmission, storage, and encoding of these color images.

Video Processing

A video is a very fast movement of pictures. The quality of the video depends on the number of frames/pictures per minute and the quality of each frame being used. Video processing involves noise reduction, detail enhancement, motion detection, frame-rate conversion, aspect-ratio conversion, color space conversion, and so on.

Computer Vision Versus Machine Vision

Computer vision refers in broad terms to the capture and automation of image analysis with an emphasis on the image analysis function across a wide range of theoretical and practical applications. Machine vision traditionally refers to the use of computer vision in an industrial or practical application or process where it is necessary to execute a certain function or outcome based on the image analysis done by the vision system. This is shown in Figure 8.3 (b). The vision system uses software to identify preprogrammed features. The system can be used to trigger a variety of set “actions” based on the findings. The components of a basic computer vision and machine vision system are the generally the same:

- an imaging device, usually a camera that contains an image sensor and a lens
- an image capture-board or frame-grabber may be used (in some digital cameras that use a modern interface (USB port), a frame-grabber is not required)

- lighting appropriate for the specific application
- a computer, but in some cases, as with “smart cameras” where the processing happens in the camera, a computer may not be required
- image processing software

The term machine vision is used in nonindustrial environments such as high-end surveillance, biomedical or life science applications, and even in the effort to improve an Internet search engine’s ability to provide image-based recognition in search. Machine vision work just as human inspectors working on assembly lines visually inspect parts to judge the quality of workmanship. Machine vision systems are programmed to perform narrowly defined tasks such as counting objects on a conveyor, reading serial numbers, and searching for surface defects. Though machine vision systems have neither the intelligence nor the learning capability of human inspectors, they are considered useful in many applications. Manufacturers favor machine vision systems for visual inspections that require high speed, high magnification, 24-hour operation, and repeatability of measurements.

Commercial and open-source machine vision software packages typically include a number of different image processing techniques such as the following:

- pixel counting: counts the number of light or dark pixels
- thresholding: converts an image with gray tones to simply black and white
- connectivity and segmentation: used to locate and/or count parts by differentiating between light and dark connected regions of pixels
- barcode reading: decoding of 1D and 2D codes designed to be read or scanned by machines
- optical character recognition: automated reading of text such as serial numbers
- gauging: measurement of object dimensions in inches or millimeters
- edge detection: finding object edges
- template matching: finding, matching, and/or counting specific patterns
- robust pattern recognition: location of an object that may be rotated, partially hidden by another object, or varying in size

In most cases, a machine vision system will use a combination of these processing techniques to perform a complete inspection. A system that reads a barcode may also check a surface for scratches and measure the length and width of a machined component. Machine vision or computer vision deals with developing a system in which the input is an image and the output is some information. For example: developing a system that scans the human face and opens any kind of lock (See Figure 8.3 (b)).

Computer Vision versus Image Processing

Computer vision tries to do what a human brain does with the retinal data that means understanding the scene based on image data. That mainly involves segmentation, recognition, and reconstruction (3D) and these work together to give us the scene understanding. Computer vision attempts what biological vision attempts. That is reading 2D images / videos of object surfaces, usually to identify or track single or multiple objects. It often employs AI techniques to do this, such as pattern recognition, machine learning, semantic ontology, Kalman filters, and may model object kinematics to predict their motion or behavior.

Image processing is the step by step transformation of input image into an output image. In between it extracts some information from the image to assist in the transformation. Basic image processing includes rotation, color-scale changes, crop, filter effects, and so on.

Computer vision employs image processing and machine learning as well as some of the other mathematical methods (e.g., variational methods, combinatorial approaches, etc.) to do the mentioned tasks. However, image processing, is mainly focused on processing raw images without giving any knowledge feedback on them. For example, if you want to do a semantic image segmentation (a computer vision task) you might apply some filtering on the image during the process (an image processing task) or try to recognize the objects in the scene (a machine learning task). Computer vision, however, is the process of studying an image or a group of images, using image processing and machine learning techniques, to get information from an image other than its properties, for example, computer vision can detect the number of windows in the image of a building. The output of a computer vision technique is not just a transformed image but much more. Just like the human brain, when it views something it stores information, like features of a face. Similarly, computer vision can also be used to extract such information from images (or videos). Computer vision algorithms can then be taught to decipher meaning in images from existing information fed

to it earlier, and can be taught to recognize patterns, distinguish between objects, and so forth. So, image processing takes in an input image and outputs an image after some defined transformations. Whereas computer vision takes an input image and outputs desired information that the algorithm was trained to process.

The Difference between Computer Vision, Image Processing, and Machine Learning

A signal is a sequence of discrete measurable observations obtained using a capturing device, be it a camera, a radar, ultrasound, a microphone, and so on. The dimensionality of the input signal gives us the first distinction between the fields. Monochannel sound waves can be thought of as a one-dimensional signal of amplitude over time, whereas a picture is a two-dimensional signal, made up of rows and columns of pixels. Recording consecutive images over time produces video which can be thought of as a three-dimensional signal.

Input of one form can sometimes be transformed to another. For example, ultrasound images are recorded using the reflection of sound waves from the object observed, and then transformed to a visual modality. X-ray can be considered similarly to ultrasound, only that radioactive absorption is transformed into an image. Magnetic resonance imaging (MRI), records the excitation of ions and transforms it into a visual image. In this sense, signal processing might actually be understood as image processing.

Assume a single image is acquired from an X-ray machine. Image processing engineers (or software) would often have to improve the quality of the image before it passes to the physician's display. Image processing is, as its name implies, all about the processing of images. Both the input and the output are images. Methods frequently used in image processing are filtering, noise removal, edge detection, color processing, and so forth. Software packages dedicated to image processing are, for example, Photoshop and Gimp.

In computer vision, quantitative and qualitative information from visual data is received. Much like the process of visual reasoning of human vision; we can distinguish between objects, classify them, sort them according to their size, and so forth. Extending beyond a single image, in computer vision one tries to extract information from video. For example, counting the number of cats passing by a certain point in the street as recorded by a video camera or measuring the distance run by a soccer player during a game and then extracting other statistics.

But not all processes are understood to their fullest which delays designer ability to construct a reliable and well-defined algorithm for tasks. Here comes machine learning methods to solve. Methodologies like support vector machine (SVM) and neural networks are aimed at mimicking our way of reasoning without having full knowledge of how we do this. For example, a sonar machine placed to alert for intruders in oil-drill facilities at sea needs to be able to detect a single diver in the vicinity of the facility. It is not possible for sonar alone to detect the difference between a big fish and a diver. More in-depth analysis is needed.

Characterizing the difference between the motions of the diver compared to a fish by sonar would be a good start. Features related to this motion, such as frequency, speed, and so on are fed into the support vector machine or neural network classifier. With training, the classifier learns to distinguish a diver from a fish. After the training set is completed, the classifier is intended to repeat the same observation as the human expert will make in a new situation. Thus, machine learning is quite a general framework in terms of input and output. Like humans, it can receive any signal as an input and give almost any type of output. The relationships of all technologies are presented in a Venn diagram as shown in Figure 8.2 (b). Table 8.1 summarizes the input and output of each domain.

TABLE 8.1. Input/Output of all technology.

Domain	Input	Output
Image processing	Image	Image
Signal processing	Electrical signal	Electrical signal, quantitative information; e.g., peak location,
Computer vision	Image/video	Image, quantitative/ qualitative information/ features; e.g., size, color, shape, classification, etc.
Machine learning	Any feature signal, from image, video, sound, etc.	Signal, quantitative/ qualitative information, image
Pattern Recognition	Information/features	Information
Machine vision	Images	Information
Robot Vision	Images	Physical action
Embedded Vision	Images and signals from sensors	Information, signal, action

8.2 TASKS AND ALGORITHMS IN COMPUTER VISION

The goal of computer vision is to emulate human vision using digital images through three main processing components, executed one after the other as shown in Figure 8.4: (1) image acquisition, (2) image processing, and (3) image analysis and understanding. As our human visual understanding of the world is reflected in our ability to make decisions through what we see, providing such a visual understanding to computers would allow them the same power.



FIGURE 8.4. Components of computer vision.

Image Acquisition

A digital image is produced by one or several image sensors which, besides various types of light sensitive cameras, includes range sensors, tomography devices, radar, ultra-sonic cameras, and so on. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or color images), but can also be related to various physical measures, such as depth, absorption, or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

Image acquisition is the process of translating the analog world around us into binary data composed of zeros and ones, interpreted as digital images. Most of the time, the raw data acquired by the devices needs to be post processed in order to be more efficiently exploited in the next steps.

Different tools have been created to build such datasets as shown in Figure 8.5: ((1) webcams and embedded cameras; (2) digital compact cameras and DSLR; (3) consumer 3D cameras and laser range finders.



FIGURE 8.5. From left to right and from top to bottom: webcam, digital SLR, laser range finder, 3D camera, and embedded camera.

Image Processing

The second component of computer vision is the low-level processing of images. Algorithms are applied to the binary data acquired in the first step to infer low-level information on parts of the image. This type of information is characterized by image edges, point features, or segments, for example. They are all the basic geometric elements that build objects in images as shown in Figure 8.6.

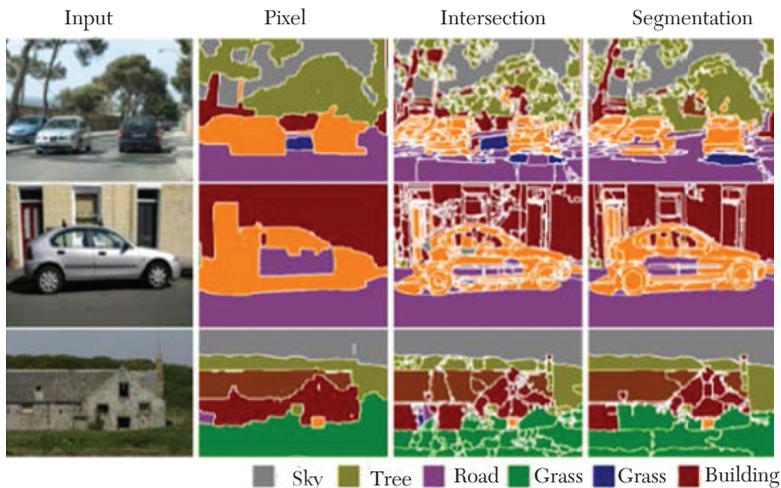


FIGURE 8.6. Classification of parts of images.

This second step usually involves advanced applied mathematics algorithms and techniques. Low-level image-processing algorithms include: (1) edge detection; (2) segmentation; (3) classification; and (4) feature detection and matching. Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to preprocess the data in order to assure that it satisfies certain assumptions implied by the method. Examples are:

- resampling in order to assure that the image coordinate system is correct
- noise reduction in order to assure that sensor noise does not introduce false information
- contrast enhancement to assure that relevant information can be detected

- scale-space representation to enhance image structures at locally appropriate scales
- feature extraction: image features at various levels of complexity are extracted from the image data. Typical examples of such features are:
 - lines, edges, and ridges
 - localized interest points such as corners, binary large objects in visual systems (BLOBs) (\bullet), or points

More complex features may be related to texture, shape, or motion.
- Detection/Segmentation: At some point in the processing a decision is made about which image points or regions of the image are relevant for further processing. Examples are:
 - selection of a specific set of interest points
 - segmentation of one or multiple image regions which contain a specific object of interest

Image Analysis and Understanding

The last step of the computer vision pipeline is the actual analysis of the data, which will allow the decision making. High-level algorithms are applied, using both the image data and the low-level information computed in previous steps. At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example:

- verification that the data satisfy model based and application specific assumptions
- estimation of application specific parameters, such as object pose or object size
- classifying a detected object into different categories

Examples of high-level image analysis are shown in Figure 8.7. They are (1) 3D scene-mapping; (2) object recognition; and (3) object tracking.

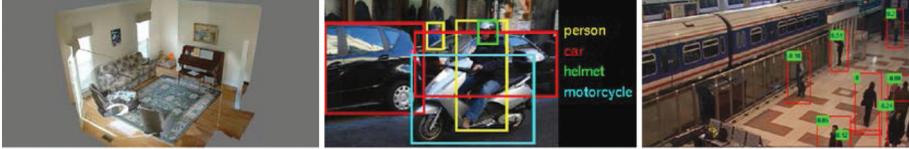


FIGURE 8.7. 3D mapping of a living room. Recognition of objects. Tracking people in a train station.

Each of the application areas employ a range of computer vision tasks; more or less well-defined measurement problems or processing problems, which can be solved using a variety of methods. Some examples of typical computer vision tasks are presented below.

Recognition

The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactorily solved in computer vision for the general case. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects (e.g., polyhedrons), human faces, printed or hand-written characters, vehicles, and in specific situations typically described in terms of well-defined illumination, background, and pose of the object relative to the camera. Different varieties of the recognition problem are described as follows:

- *Recognition:* One or several pre-specified or learned objects or object classes can be recognized, usually together with their 2D positions in the image or 3D poses in the scene.
- *Identification:* An individual instance of an object is recognized. Examples: identification of a specific person's face or fingerprint, or identification of a specific vehicle.
- *Detection:* The image data is scanned for a specific condition. Examples: detection of possible abnormal cells or tissues in medical images or detection of a vehicle in an automatic road-toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interesting image data. It can be further analyzed by more computationally demanding techniques to produce a correct interpretation.

Several specialized tasks based on recognition exist, such as:

- *Content based image retrieval*: Finding all images in a larger set of images that have specific content. The content can be specified in different ways, for example in terms of similarity relative a target image (give me all images similar to image X), or in terms of high-level search criteria given as text input (give me all images that contains many houses, are taken during winter, and have no cars in them).
- *Pose estimation*: Estimating the position or orientation of a specific object relative to the camera. An example application for this technique would be assisting a robot arm in retrieving objects from a conveyor belt in an assembly line situation.
- *Optical character recognition (OCR)*: Identifying characters in images of printed or handwritten text, usually with a view for encoding the text in a format more easy handle for editing or indexing (e.g., ASCII).

Motion

Several tasks relate to motion estimation, in which an image sequence is processed to produce an estimate of the velocity either at each point in the image or in the 3D scene. Examples of such tasks are:

- *Egomotion*: determining the 3D rigid motion of the camera.
- *Tracking*: following the movements of objects (e.g. vehicles or humans).

Scene reconstruction

Given one or (typically) more images of a scene, or a video, scene reconstruction aims at computing a 3D model of the scene. In the simplest case the model can be a set of 3D points. More sophisticated methods produce a complete 3D surface model.

Image restoration

The aim of image restoration is the removal of noise (sensor noise, motion blur, etc.) from images. The simplest possible approach for noise removal is various types of filters such as low-pass filters or median filters. More sophisticated methods assume a model of how the local image structures look like, a model that distinguishes them from the noise. By first analyzing the image data in terms of the local image structures, such as lines or edges, and then controlling the filtering based on local information from the analysis step, a better level of noise removal is usually obtained compared to the simpler approaches.

When developing computer vision algorithms, one has to face different issues and challenges, related to the very nature of the data or the application to be created and its context such as (1) noisy or incomplete data; (2) real-time processing; and (3) limited resources of power and memory.

Algorithms

Some of the following algorithms are used in computer vision.

- SIFT and SURF algorithms are for *feature point extraction*. Scale invariant feature transform (SIFT) and sped-up robust features (SURF) are used for object recognition and image registration.
- Viola Jones algorithm used for object (especially face) detection in real time. One of the most elegant algorithms.
- “Eigen faces” approach using principal component analysis (PCA) is for dimension reduction. It is used in face recognition. It has a very intuitive approach and yet it is mathematically strong.
- Lucas-Kanade algorithm and Horn-Schunk algorithm are for optical flow calculation. They are used for tracking and stereo registration.
- Mean-shift algorithm is for fast tracking of an object. It is not very robust, but easy to use, and very useful in specific applications.
- Kalman filter is again for object tracking which uses point features for tracking. It has great use in many fields like computer vision, control systems, etc.
- Adaptive thresholding and other thresholding techniques are used in computer vision.
- Machine-learning algorithms like support vector machine (SVM), k nearest neighbor (KNN), Naive Bayes, etc. are also very important in the field of computer vision.
- Edge detection, interest point detection, histogram of oriented gradient (HoG), optical flow, pyramids, motion models, global motion, mean shift, camera model, fundamental matrix, face recognition, structure from motion, stereo, bag of features, and Hough transform are algorithms in computer vision used to extract information from images.
- Face recognition: Snapchat and Facebook use face detection algorithms to apply filters and recognize a person in pictures.

- Image retrieval: Google Images uses content-based queries to search relevant images. The algorithms analyze the content in the query image and return results based on best-matched content.
- Gaming and controls: A great commercial product in gaming that uses stereo vision is Microsoft Kinect.
- Surveillance: Surveillance cameras are ubiquitous at public locations and are used to detect suspicious behaviors.
- Biometrics: Fingerprint, iris, and face matching remain some common methods in biometric identification.
- Smart cars: Vision remains the main source of information to detect traffic signs, lights, and other visual features.

Feature Extraction

A primary component of the computer-vision software pipeline is feature extraction, which identifies and encodes relevant image features.

A typical vision software pipeline illustrated in Figure 8.8 takes an image or video and distills the data down to key relevant information components called features. The features are then processed, typically with machine learning algorithms, to gain semantic and geometric information about objects in the scene, while identifying the objects type and location. Objects can be observed over time to gain understanding of the context

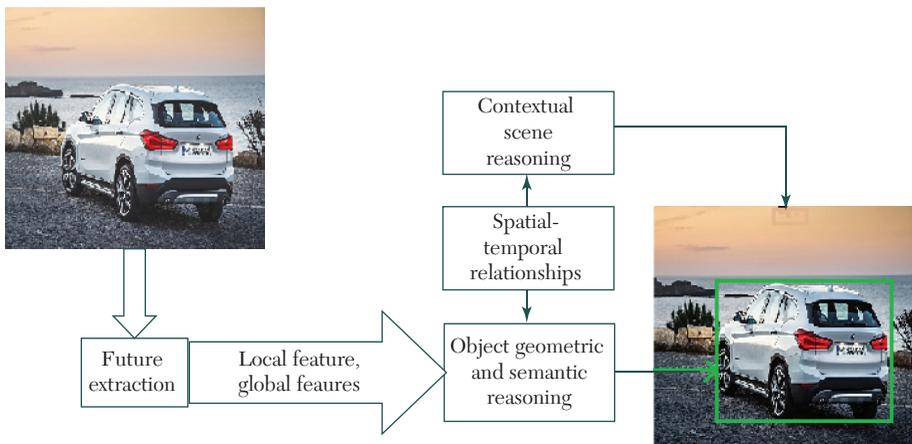


FIGURE 8.8. Overview of computer-vision feature extract processing. The image on the left has features extracted and the features are used to understand the object (car) and the scene in an iterative process.

of the scene. Typically the process is iterative, once enough object and context understanding is gained, such information can be used to refine knowledge of the scene. Features within an image are identified by the feature extraction algorithm, a principle component of the vision software pipeline.

A capable feature extraction algorithm must distill important image information into scale, illumination, viewpoint, and rotation invariant signatures, called feature descriptors. Feature descriptors are vital to the algorithmic process of recognizing objects. For example to recognize a car, the feature extraction algorithm could enable the identification of the wheels, side-mirrors, and windshield. Given the relative location of these features in the image, a system could then recognize that the scene contains a car.

The “quality” of any particular algorithm lies in its ability to consistently identify important features, regardless of changes in illumination, scale, viewpoint, or rotation. In general, the more capable an algorithm is at ignoring these changes, the more computationally expensive it becomes. Features from the Accelerated Segment Test (FAST) is a Corner detection method to extract feature points. To demonstrate this tradeoff, the unsophisticated FAST corner detector executes in 13ms for a 1024x768 image on a desktop machine, but provides no robustness to changes in illumination, scale, or rotation. In contrast, the highly capable SIFT algorithm, which is illumination, scale, viewpoint, and rotation invariant, processes the same image in 1920ms, which is 147 times slower. One method to address the performance issues of feature extraction on mobile-embedded platforms is to utilize cloud-computing resources to perform vision computation. However, this approach requires much more wireless bandwidth compared to a system with a capable feature detector. For example, transmitting compressed SIFT features would require about 84kB for a large number of features (over 1000) compared to 327kB to send a compressed image. Since existing wireless mediums are already straining to carry existing data, there is significant value for communication mediums to perform feature extraction locally, even if cloud resources are used to analyze the feature data.

Feature Extraction Algorithms

A typical feature extraction algorithm, as illustrated in Figure 8.9, is composed of five steps.

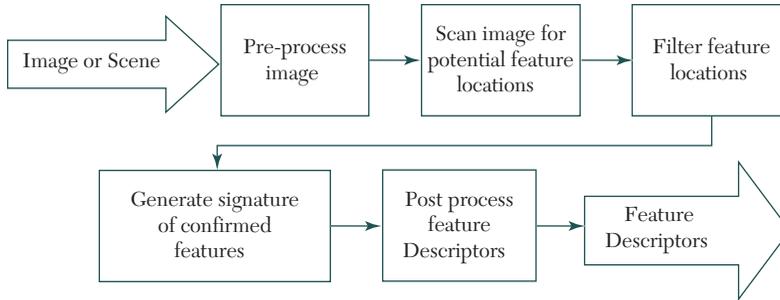


FIGURE 8.9. Overview of feature extraction.

This figure shows the general steps involved in feature extraction. The first three steps of the process locate feature points. The final two steps create feature vectors. The quality of a feature extraction algorithm is evaluated on four major invariance characteristics: illumination, scale, viewpoint, and rotation. A very capable feature extraction algorithm will produce feature sets for an image that are nearly identical, despite changes in lighting, object position, or camera position. FAST, HoG, and SIFT algorithms represent a wide trade-off of quality and performance. From the high-speed low-quality FAST algorithm to the very high-quality and expensive SIFT algorithm are used to extract information. In addition, these algorithms are widely representative of the type of operations that are typically found in feature extraction algorithms.

1. The first step is to preprocess the image, an operation which typically serves to highlight the intensity discontinuities (i.e., object boundaries) by, for example, eliminating the DC components (mean values) of the image.
2. The second step scans the processed image for potential feature point locations; the specifics of this phase are highly dependent on the underlying algorithm.
3. The third step of feature extraction works to filter out weak or poorly represented features through, for example, sorting the features found based on a key characteristic and then dropping the non-prominent results. The second and third steps implement a process typically called feature-point localization.
4. Once feature points are localized, the fourth step computes the feature descriptor. A feature descriptor is a compact representation of an image

feature that encodes key algorithm-specific image characteristics, such as variations of pixel intensity values (gradients). The feature descriptor implements the illumination, scale, and rotation invariance supported by a particular algorithm.

5. The fifth and final step of feature extraction performs another filter pass on the processed feature descriptors based on location constraints.

FAST corner detection is designed to quickly locate corners in an image for position tracking. It is the least computationally intensive and the least robust of the algorithms. The FAST feature matching degrades when the scene is subject to changes in illumination, object position, or camera location and image noise. The algorithm locates corners by comparing a single pixel to the 16 pixels around it. To perform this comparison, the target pixel and surrounding pixels must be fetched from memory, and then the target pixel must be compared to all the pixels in the enclosing circle. The descriptor is made by concatenating the pixel intensities of the 16 surrounding pixels. Speeding up these comparisons, through a combination of functional unit and thread-level parallelism, greatly improves the performance of FAST.

HoG is commonly used for human or object detection. The HoG algorithm is more computationally intensive than FAST, because it provides some illumination and rotation invariance. The descriptor is built using the histogram of the gradients of pixel intensities within a region, which are subsequently normalized. The major operations for this phase of HoG computation are the fetching of image data from memory and the calculation of histograms using integral images. This phase of the HoG algorithm utilizes a sliding window of computation in which each window is independent. Consequently, much parallelism is available to exploit. The second major time component is preprocessing. The preprocessing for HoG is computation of the integral image. This is comprised mainly of memory operations, the computation of the image gradient, and finally the histogram binning of gradient values based on direction. More efficient computation of these components, through functional unit support and thread-level parallelism, significantly speeds up processing.

SIFT is a feature extraction algorithm widely used for object recognition. It is the most computationally expensive and algorithmically complex of the feature extraction algorithms, but it provides a high level of invariance to most scene changes. This portion of the algorithm involves computing and binning the gradient directions in a region around the feature point,

normalizing the feature descriptor, and accessing pixel memory. The operations in this phase are performed on each feature point and benefit from specialized hardware. The second largest component of SIFT is the feature point localization. This portion is dominated by compare and memory operations to locate the feature points. There are also 3D curve fitting and gradient operations to provide sub-pixel accuracy and filter weaker responses, respectively. This phase of SIFT provides ample thread level parallelism. The preprocessing step, the third most expensive component in SIFT, involves iterative blurring of the image which is a convolution operation. The convolution requires multiplying a region of the image by coefficients and summing the result, operations which can benefit from specialized functional unit support.

Visual recognition tasks such as image classification, localization, and detection are key components of computer vision.

Image Classification

The problem of image classification is explained as given a set of images that are all labeled with a single category. Designers are asked to predict these categories for a novel set of test images and measure the accuracy of the predictions. There are a variety of challenges associated with this task, including viewpoint variation, scale variation, intra-class variation, image deformation, image occlusion, illumination conditions, and background clutter.

Instead of trying to specify what every one of the image categories of interest look like directly in code, designers provide the computer with many examples of each image class and then develop learning algorithms that look at these examples and learn about the visual appearance of each class. In other words, they first accumulate a training dataset of labeled images, then feed it to the computer to process the data.

Given that fact, the complete image classification pipeline can be formalized as follows:

- Input is a training dataset that consists of N images, each labeled with one of K different classes.
- Then, use this training set to train a classifier to learn what every one of the classes looks like.
- In the end, evaluate the quality of the classifier by asking it to predict labels for a new set of images that it's never seen before and then compare the true labels of these images to the ones predicted by the classifier.

The most popular architecture used for image classification is *convolutional neural networks* (CNNs). A typical case for the use of CNNs is when one feeds the network images and the network classifies the data. CNNs tend to start with an input “scanner” which isn’t intended to parse all the training data at once. For example, to input an image of 100 x 100 pixels, designer wouldn’t want a layer with 10,000 nodes. Rather, the designer creates a scanning input layer of say 10 x 10 which he feed the first 10 x 10 pixels of the image. Once the designer passed that input, he feeds it the next 10 x 10 pixels by moving the scanner one pixel to the right. This technique is known as *sliding windows*.

This input data is then fed through convolutional layers instead of normal layers. Each node only concerns itself with close neighboring cells. These convolutional layers also tend to shrink as they become deeper, mostly by easily divisible factors of the input. Besides these convolutional layers, they also often feature *pooling layers*. Pooling is a way to filter out details. A commonly found pooling technique is *max pooling*, where one takes, say, 2 x 2 pixels and passes on the pixel with the most amount of a certain attribute.

Most image classification techniques nowadays are trained on *ImageNet*, a dataset with approximately 1.2 million high-resolution training images. Test images will be presented with no initial annotation (no segmentation or labels) and algorithms will have to produce labeling specifying what objects are present in the images. Typically, computer-vision systems use complicated multi-stage pipelines and the early stages are typically hand-tuned by optimizing a few parameters.

Object Detection

The task to define objects within images usually involves outputting bounding boxes and labels for individual objects. This differs from the classification/localization task by applying classification and localization to many objects instead of just a single dominant object. You only have two classes of object classification, which means object bounding boxes and non-object bounding boxes. For example, in car detection, you have to detect all cars in a given image with their bounding boxes. If we use the sliding window technique in the same way we classify and localize images, we need to apply a CNN to many different crops of the image. Because CNN classifies each crop as object or background, we need to apply CNN to huge numbers of locations and scales, which is very computationally expensive!

Object Tracking

Object tracking refers to the process of following a specific object of interest, or multiple objects, in a given scene. It traditionally has applications in video and real-world interactions where observations are made following an initial object detection. Now, it's crucial to autonomous driving systems such as self-driving vehicles from companies like Uber and Tesla.

Object-tracking methods can be divided into two categories according to the observation model: generative method and discriminative method. The generative method uses the generative model to describe the apparent characteristics and minimizes the reconstruction error to search the object, such as principal component analysis (PCA). The discriminative method can be used to distinguish between the object and the background, its performance is more robust, and it gradually becomes the main method in tracking. The discriminative method is also referred to as tracking-by-detection, and deep learning belongs to this category. To achieve the tracking by detection, we detect candidate objects for all frames and use deep learning to recognize the wanted object from the candidates. There are two kinds of basic network models that can be used: *stacked auto encoders* (SAE) and *convolutional neural network* (CNN).

The most popular deep network for tracking tasks using SAE is *deep learning tracker*, which proposes offline pre-training and online fine-tuning the net. The process works like this:

- Off-line unsupervised pre-train the stacked denoising auto encoder using large-scale natural image datasets to obtain the general object representation. Stacked denoising auto encoder can obtain more robust feature expression ability by adding noise in input images and reconstructing the original images.
- Combine the coding part of the pre-trained network with a classifier to get the classification network, then use the positive and negative samples obtained from the initial frame to fine tune the network, which can discriminate the current object and background. DLT uses particle filter as the motion model to produce candidate patches of the current frame. The classification network outputs the probability scores for these patches, meaning the confidence of their classifications, then chooses the highest of these patches as the object.
- In the model updating, DLT uses the way of limited threshold.

Because of its superiority in image classification and object detection, CNN has become the mainstream deep model in computer vision and in visual tracking. Generally speaking, a large-scale CNN can be trained both as a classifier and as a tracker. CNN-based tracking algorithms are *fully-convolutional network tracker* (FCNT) and *multi-domain CNN* (MD Net).

FCNT analyzes and takes advantage of the feature maps which is a pre-trained ImageNet, and results in the following observations.

- CNN feature maps can be used for localization and tracking.
- Many CNN feature maps are noisy or unrelated for the task of discriminating a particular object from its background.
- Higher layers capture semantic concepts on object categories, whereas lower layers encode more discriminative features to capture intra-class variation.

Semantic Segmentation



FIGURE 8.10. Semantic segmentation recognition.

The process of segmentation divides whole images into pixel groupings which can then be labeled and classified. Particularly, semantic segmentation tries to semantically understand the role of each pixel in the image (e.g., is it a car, a motorbike, or some other type of class?). For example, in Figure 8.10, apart from recognizing the person, the road, the cars, the trees, and so on, we also have to delineate the boundaries of each object. Therefore, unlike classification, we need dense pixel-wise predictions from the models.

As with other computer vision tasks, CNNs have had enormous success on segmentation problems. One of the popular initial approaches was patch classification through sliding window, where each pixel was separately classified into classes using a patch of images around it. This, however, is very inefficient computationally because we don't reuse the shared features between overlapping patches.

The solution, is *fully convolutional networks* (FCN) that popularized end-to-end CNN architectures for dense predictions without any fully connected layers. This allowed segmentation maps to be generated for images of any size and was also much faster compared to the patch classification approach. Almost all subsequent approaches on semantic segmentation adopted this paradigm.

Instance Segmentation

Beyond semantic segmentation, instance segmentation segments different instances of classes, such as labeling five cars with five different colors. In classification, there's generally an image with a single object as the focus and the task is to say what that image is. But in order to segment instances, the designer needs to carry out far more complex tasks. The designer sees complicated sights with multiple overlapping objects and different backgrounds, and he not only classifies these different objects, but also identifies their boundaries, differences, and relations to one another!

Object Recognition Algorithms

Object recognition is a process for identifying a specific objects in a digital image or a video. Object recognition algorithms merrily rely on matching, pattern recognition or learning algorithms using a feature based technique or appearance based technique. Although various object recognition algorithms are introduced in recent days, the algorithms SIFT, ASIFT, SURF, and ORB techniques are more distinct in terms of performance accuracy, and speed. In these algorithms ORB is more suitable for embedded environments. These algorithms are widely used in many applications of robotics, industries, military, home serve applications, and so forth.

SIFT: Scale Invariant Feature Transforms Algorithm

The SIFT algorithm is an algorithm in computer vision which is to detect and describe local features in images. It describes the distinctive features that have properties which help for matching different images of an object and these distinctive features are invariant to various transformations of images. In SIFT algorithm there are four major stages of computation used to generate the set of image features that are briefed as follows:

- The first step is scale space extreme detection. In this step the algorithm searches key points over all scales and image locations. It has been implemented by using the difference of Gaussian (DoG) function to

identify potential point of interest that are invariant to image scale and orientation.

- The second step is key point localization. In this step the key points are filtered so that only stable key points are retained. At each candidate location, a detailed model should determine location and scale. Hence, the key points are selected depending on the measurement of stability. Unstable key points with low contrast will be rejected.
- The third step is orientation assignment. In this stage each key point is assigned an orientation to make the description invariant to rotation. Here key point locations are found at particular scales and orientations are assigned to them. So, this step has ensured invariance to the image location, scale, and rotation.
- The fourth step is called keypoint descriptor. This step involves the computation of descriptor vector for each keypoints obtained so that the descriptor will be invariant to the descriptor. This step is performed on the scale of an image that is close to scale of the key points. Consider, key point descriptor use and orientation histograms on a 4 x 4 grid. Each of the orientation histograms has eight orientation bins that are created over a 4 x 4 pixel window. Therefore the feature vectors will have a total of 128 elements which is computed from a 16 x 16 pixels window. Thus, in this algorithm, object recognition will be done by matching the descriptor elements of input image and the reference image will be obtained as the result of this algorithm.

SURF: Speed Up Robust Features Algorithm

In computer vision, SURF algorithm is a local feature detector and descriptor. It has been partly inspired by the scale-invariant feature transform (SIFT) descriptor, but the SURF key point detector is a better algorithm compared to SIFT in its speed. The SURF algorithm is efficient at rotation and other transformations of image. In SURF algorithm there are four major stages of computation to generate the image features which are briefed as follows:

- The first step is integral image creation. An input image is obtained and an integral image is created with respect to the input image. It determines which are to be extracted in the following steps are to be obtained from this integral image.

- The second step is called as fast-Hessian detector. Here determines are extracted from the DoG in the Hessian detector stage. When this determines are greater than the threshold, then it is designated as key point candidate. Then a key point candidate will be selected as a key point, if the determine is greater than 18 neighboring determines of up-scale and down-scale, and also greater than eight neighboring determines of same scale. This step extracts key points in each size of the box filter in the following pattern, 9x9, 15x15, 21x21, and 27x27. Hence the processing speed is slow.
- The third step is called the get orientation stage. This step is to decide a major direction for which a partial image with a size of 6 scale based on each key point. This step gets Haar response using Haar Wavelet and accumulates response included in 0~60 degrees. Then this process is repeated 72 times as 5 degrees for each unit. The major direction is decided by the largest vector and the cos and sin values of major direction are calculated and interpolated for the partial image with size of 20* scale based on each key point.
- The fourth step is the get descriptor stage. In this step the size of the descriptor is obtained. The window formulated in the previous step is divided into 4 x 4 of 16 areas, in which each pixel is calculated using the Haar wavelet. Each of the 16 areas of 4 descriptors are created, so a total 16 x 4, i.e., 64 descriptors are created in this step. As the result, the matching patterns are found by comparing the descriptors obtained from different images.

ORB: Oriented Fast and Rotated Brief Algorithm

The ORB algorithm from in computer vision is an efficient alternative to SIFT or SURF. The ORB is very strong in object recognition at image rotation condition. The ORB overcomes the drawbacks faced by the SIFT and SURF algorithms. In ORB algorithm there are four major stages of computation that are briefed as follows.

- The first step is called features from accelerated test (FAST) corner detection. The FAST corner detection is used to find the key point in the input image. Moreover, the number of corners is likely to become key point by this FAST corner detection. It also uses image pyramid to produce multi-scale features. But, FAST does not use computer orientation.

- The second step is called the Harris corner detection. Once the key points are obtained, this step is implemented to find the top N points among the obtained key points of the input image. ORB extracts key points in each scale using FAST and Harris corner detection by orientation and rotation invariance are not computed in this step.
- The third step is to get the orientation by intensity centroid. In this step the intensity weighted centroid of patch with the located corner at center is computed. Hence, the direction of the vector from this corner point to the centroid, forms the orientation. For obtaining the rotation invariance, the moments are computed within a circular region of a defined radius on which the radius will be defined by the size of the patch.
- The fourth step is to get the descriptor by binary robust independent elementary features (rBRIEF—rotation BRIEF). In this algorithm rBRIEF is used because BRIEF is weak at rotation. The rBRIEF determines a test point what is strong at rotational changes. This step runs a greed search among all binary tests to find the ones that have high rotational invariance and mean close to 0.5, where rBRIEF enumerates a binary test for 31×3 pixel patch and sub-window 5×5 based on key points taken from many images. rBRIEF compares all test points, and if the point is larger than the threshold, it is removed. These procedures are repeated until 256 tests are created. The result is called rBRIEF. The final result will be obtained by descriptor matching where object is recognized from the image. ORB is faster than SIFT and SURF and ORB descriptor work better than SURF.

The SURF algorithm will be an extension of the SIFT algorithm where this is more efficient than the SIFT algorithm in case of robustness. The important speed gain is due to the use of integral images. Object recognition has highlighted the potential of SIFT and SURF algorithms in a wide range of computer vision applications. In spite of these, SIFT and SURF are too slow to recognize objects in an embedded environment. Hence an ORB algorithm is introduced which efficiently replaced SIFT and SURF algorithms and is much faster and evidentially suitable for object recognition in an embedded environment. ORB algorithm increases the speed in real time and enhances by integrating with other algorithms for moving tracking or multiple-object detection.

Optical Flow and Point Tracking

Optical flow and tracking are techniques that are needed for motion analysis in video sequences. When analyzing video data, motion is probably the most important cue, and the most common techniques to exploit this information are difference images, optical flow, and point tracking. Difference images are useful when the camera is static, as most of the scene is stationary and there is limited motion. In that case, taking the difference of two images directly gives us the location and magnitude of motion. Optical flow calculates a displacement map or flow field for every pixel in every frame of the video. There are no restrictions on camera or scene motion and it is usually calculated between successive frames of a video sequence. Hence, it is extremely localized in time, that is optical flow is dense spatially, but sparse temporally. Point tracking tries to track points over several frames, thus it is temporally denser compared to optical flow. It is usually sparse spatially, that is only a few pixels are tracked in each frame.

These major computer vision techniques can help a computer extract, analyze, and understand useful information from a single image or from a sequence of images. There are many other advanced techniques including style transfer, colorization, action recognition, 3D objects, human pose estimation, and more.

Commercial Computer Vision Software Providers

In addition to hardware choices, computer vision Development Company to provide the software or algorithms are given below, such as:

- VectorBlox (vision FPGA development)
- ZMicro (vision FPGA development)
- AIotive / AdasWorks (software for self-driving cars and ADAS)
- Algolux, Itseez / Intel (creators & maintainers of OpenCV)
- QuEST Global (computer vision firmware development)
- Mitek (image based POS sales)
- Numenta (software for anomaly detection)
- PathPartner Technology
- Morpho
- StradVision
- Sighthound

8.3 APPLICATIONS OF COMPUTER VISION

Techniques developed for computer vision have many applications in the fields of robotics, human-computer interaction, and visualization. Some example are: (1) motion recognition, (2) augmented reality, (3) autonomous cars, (4) domestic/service robots, and (5) image restoration such as denoising, etc. (see Figure 8.11). Some of the applications are analyzed and discussed in the chapters on industrial vision, medical vision, and machine vision.

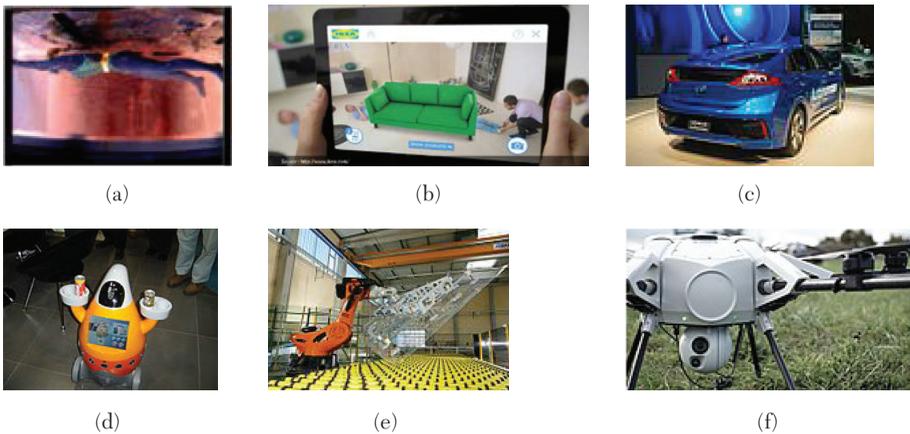


FIGURE 8.11. Some applications of computer vision. (a) Motion tracking. (b) Augmenting living room before buying new sofa. (c) Autonomous car. (d) A domestic robot. (e) Robotic vision. (f) Delivery of packages via drones.

Computers can create a 3-D image from a 2-D image, such as those in cars, and provide important data to the car and/or driver. For example, this intelligent device could provide inputs to the driver or even make the car stop if there is a sudden obstacle in the road. When a human who is driving a car sees someone suddenly move into the path of the car, the driver must react instantly. In a split second, human vision has completed a complex task, that of identifying the object, processing data, and deciding what to do.

One of the most prominent application fields is medical computer vision or medical image processing. This area is characterized by the extraction of information from image data for the purpose of making a medical diagnosis of a patient. Generally, image data is in the form of microscopy images, X-ray images, angiography images, ultrasonic images, and tomography images. An example of information that can be extracted from such image data is

detection of tumors, arteriosclerosis, or other malign changes. It can also be measurements of organ dimensions, blood flow, and so on. This application area also supports medical research by providing new information, for example, about the structure of the brain, or about the quality of medical treatments. In Chapter 3, “Medical Vision,” it is clearly discussed.

A second application area in computer vision is in industry. Here, information is extracted for the purpose of supporting a manufacturing process. One example is quality control where details or final products are being automatically inspected in order to find defects. Another example is measurement of position and orientation of details to be picked up by a robot arm. More examples are discussed in Chapter 2, “Industrial Vision.”

Military applications are probably one of the largest areas for computer vision. The obvious examples are detection of enemy soldiers or vehicles and missile guidance. More advanced systems for missile guidance send the missile to an area rather than a specific target, and target selection is made when the missile reaches the area based on locally acquired image data. Modern military concepts, such as “battlefield awareness,” imply that various sensors, including image sensors, provide a rich set of information about a combat scene which can be used to support strategic decisions. In this case, automatic processing of the data is used to reduce complexity and to fuse information from multiple sensors to increase reliability.

One of the newer application areas is autonomous vehicles, which include submersibles, land-based vehicles (small robots with wheels, cars or trucks), aerial vehicles, and unmanned aerial vehicles (UAV). The level of autonomy ranges from fully autonomous (unmanned) vehicles to vehicles where computer-vision-based systems support a driver or a pilot in various situations. Fully autonomous vehicles typically use computer vision for navigation, that is for knowing where it is, or for producing a map of its environment and for detecting obstacles. It can also be used for detecting certain task specific events, for example, a UAV looking for forest fires. Examples of supporting systems are obstacle warning systems in cars, and systems for autonomous landing of aircraft. There are enough examples of military autonomous vehicles ranging from advanced missiles, to UAVs for reconnaissance missions or missile guidance. Space exploration is already being made with autonomous vehicles using computer vision, for example, NASA Mars Exploration Rover. Other application areas include support of visual effects creation for cinema and broadcast, for example, camera tracking (match moving) and surveillance.

In the entire area of logistics automatic sorting equipment must offer continually greater capacity in terms of speed, quantity, and safety data. It must be ensured effective feedback monitoring throughout the creation of the value chain, right up to the final customer. The key to future success are clear advantages of intelligent reading devices based on image processing in comparison with conventional laser scanners. An example might be a common situation at the supermarket checkout, when a user cannot read the barcode label by laser scanner and the code must be entered manually. The reason why the laser scanner could not read the barcode could be because it was damaged, faded, dirty, fuzzy, warped, poorly printed, and so on. (Figure 8.12). When we relate this situation to the entire field of the logistics industry, similar downtimes when reading codes in terms of economy, efficiency, and quality for automatic sorting machines are unacceptable. These failures, when reading codes, can result in a range of

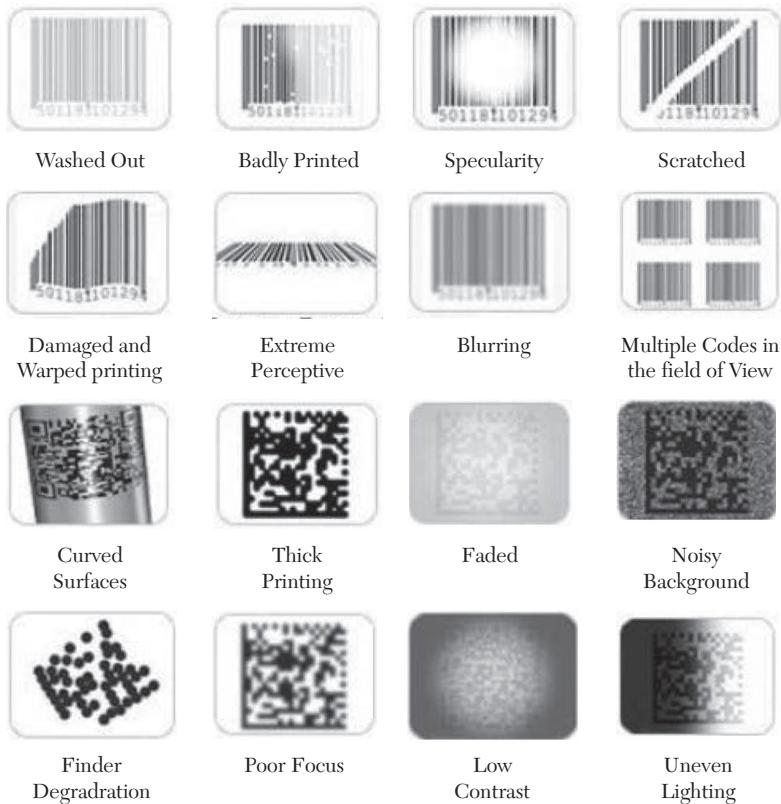


FIGURE 8.12. Sample a variety of damage 1D and 2D codes.

negative impacts associated with higher costs or liability for quality. Devices based on computer vision and the utilization of advanced algorithms make it possible to read these damaged codes, unlike laser scanners.

Besides simple reading of 1D and 2D barcodes for identification, it is possible to use vision systems for other requirements, such as:

- additional identification of un-coded text so called optical character recognition (OCR),
- correct placement of labels (e.g., their orientation)
- verification of the presence of the logo
- quality control
 - control of surface defects—scratches, cracks, packaging integrity
 - dimensional control relative to standards and tolerances (measurement accuracy up to 0.05 mm)
 - control of the packaging—shape, color
- automatic sorting packages, etc.

Figure 8.13 (c) is a prime example of sorting lines that show control of correct position of the cap using a camera system. If the cap is okay the product goes on the conveyor for palletizing. When an incorrect fit it is detected, the product is automatically eliminated.

Similarly as in the above case transported boxes can be sorted as well on the basis of their size and thus achieve efficient use of transport space

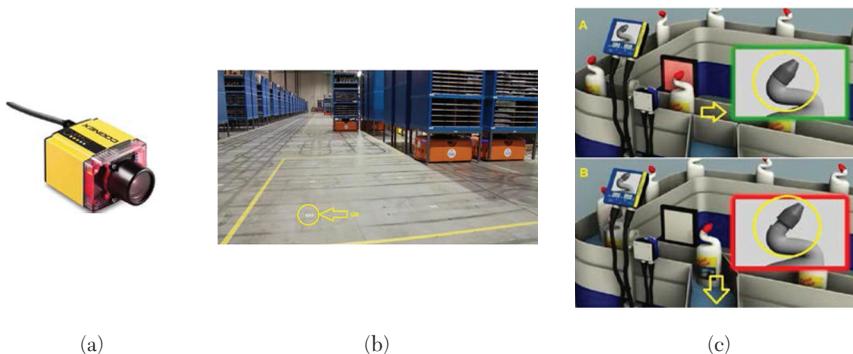


FIGURE 8.13. (a) Cognex DataMan. 500 (b) Kiva robots orientation. (c) Automatic control by camera.

of trucks or containers. Some of the advantages of scanners based on image processing are:

- high-speed reading,
- high reliability,
- omnidirectional reading of the code and reading using OCR,
- image storage—successful or unsuccessful capture of code, and
- feedback about the quality of codes.

Code readers using computer vision can be used just like standard laser scanners: stationary or handheld. An example of the executive stationary readers is DataMan 500 (Figure 8.13 (a)) by company Cognex, which is based on its own chip technology for computer vision, Cognex VSoC (vision system on a chip). This technology ensures ultrafast automatic exposure and speed shooting up to 1000 frames per second. Results of a read can be sent at speeds of up to 90 per second, and each result contains up to three chains of codes, giving a total of 270 readings per second. Significant is the ability to load up to six different codes in one image, and independently of their position. Compared to laser readers, users can monitor exactly what they see on the device, either in real time on a monitor, or later, thanks to the ability to archive the image—this allows important feedback for analysis of potential damage to codes and on this basis the user can take appropriate measures. These types of readers have no moving parts, so they have double the lifetime and a much higher reliability than laser readers. Curiously, the DataMan 500 is the first sensor for the logistics industry, which utilizes technology of auto focus with liquid lenses that maximize the depth of focus for greater reliability in situations where the object is changing positions.

Use of computer vision in logistics can be found not only in a device for the identification and control, but also in transport devices. For example, several Amazon warehouses already operate with mobile Kiva robots which stock high shelving units with goods (Figure 8.13 b).

These robots go to below the standardized shelving unit and automated instructions guide them to the destination. The robots move in a reserved area, and using computer vision follow a route to the destination using QR codes that appear in various places on the warehouse floor. The robots pass over the QR codes and load data. Robots are also littered with side sensors, through which they communicate with their environment and with each

other. Therefore there are no collisions and traffic is smooth and efficient. At the same a fully automated central system oversees all of the robots, knows where they are at any given time, and ensures there are no collisions and errors. Amazon warehouses that use the robots can hold about 50% more goods than traditional warehouses. This is because the spaces between shelves that are necessary for the movement of people can be eliminated because Kiva robots are moving directly below the shelving unit.

Another logistical area where computer vision is used is for the transport of packages via drones. Big companies like Amazon, Google, or DHL courier companies would be very happy if they could use drones for delivering packages (Figure 8.11 f). Other than legal issues (flight levels, air traffic control, etc.) that in many countries currently make it impossible for the mass use of drones in the supply chain, the biggest obstacle is with the “vision” of these machines. For the future it is necessary for the drones to avoid unknown obstacles when flying in unfamiliar surroundings as is partly possible already with advanced autonomous vehicles. This proves necessary to ensure sufficiently powerful hardware and software capable of processing and analyzing large amounts of data in real time. For this, neural networks to machine learning is used. However, at present the drones are dependent on GPS navigation and by a relatively primitive technology for avoiding obstacles that are built on the same technology as a parking assistant for cars.

Computer vision brings to the process of identification many advantages such as increased speed, accuracy, or for example feedback, thanks to the analysis of captured images. More advanced use of computer vision in autonomous machines such as transport robots in warehouses or drones for the transportation of packages directly to customers without the need for human involvement into the process of their operation, will allow future integration of these devices into the Internet of things and thereby further improve the efficiency in the logistics supply chain.

Packages and Frameworks for Computer Vision

OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Usage ranges from interactive art, mines inspection, stitching maps on the web, or through advanced robotics.

SimpleCV is an open-source framework for building computer vision applications and it can access several high-powered computer vision libraries such as OpenCV without having to first learn about bit depths, file

formats, color spaces, buffer management, eigen values, or matrix versus bitmap storage.

Mahotas is a computer vision and image processing library for Python. It includes many algorithms implemented in C++ for speed while operating in NumPy arrays and with a very clean Python interface. Mahotas currently has over 100 functions for image processing and computer vision and it keeps growing.

OpenFace is a Python and Torch implementation of face recognition with deep neural networks.

FaceNet is unified embedding for face recognition and clustering by Google. PyTorch allows the network to be executed on a CPU or with compute unified device architecture (CUDA).

Ilastik is a simple, user-friendly tool for interactive image classification, segmentation, and analysis. It is built as a modular software framework, which currently has workflows for automated (supervised) pixel and object level classification, automated and semi-automated object tracking, semi-automated segmentation and object counting without detection. Using it requires no experience in image processing.

8.4 ROBOTIC VISION

Robotic vision is a combination of sensors, computer software, and cameras to perform tasks. Robots are becoming more flexible, accurate, safer, and faster. These are more energy efficient, compact, and lighter. These are easier to program and capable of integrating with a wider range of machines. Some examples of robots at work are shown in Figure 8.14.

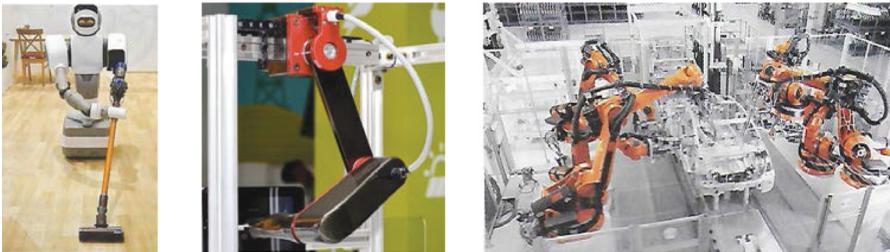


FIGURE 8.14. Aeolus smart-home robot cleaning the floor; robot testing a smartphone, a robot working in the automotive industry.

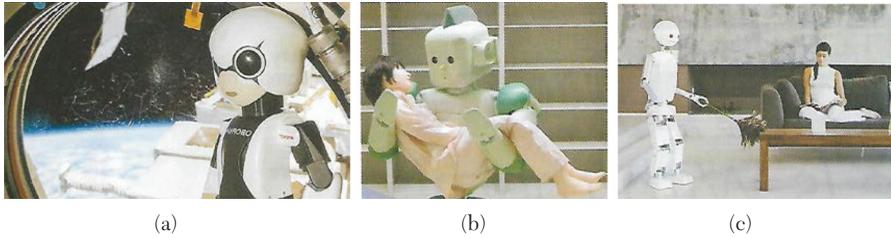


FIGURE 8.15. (a) ISS talking robot KIROBO. (b) Medical care. (c) Household chores Robos.

On August 21, 2013, Japan created a talking robot, named Kirobo who went to the International Space Station (ISS). As shown in Figure 8.15 (a), robots can be used to executive repetitive tasks like loading and unloading expensive CNC machines, freeing the human operators to take up value-addition jobs. New generation robots, called “cobots,” come with an intuitive user interface and additional features allow any person to control them through a modern smartphone. A collaborative robot, or cobot, works interactively with humans in a shared a workspace. It helps the human operator(s) to fulfill tasks and minimize risk in situations such as transportation and handling of sharp, pointed, hot, or heavy work pieces.

1. *Medical care.* This can be a defining point for elderly, ill, or disabled people who need someone to look after them. A 24-hour humanoid robot assistant who understands speech and gesture commands can be programmed to handle emergency situations. This would ease the lives of many who depend on nurses or nannies. See Figure 8.15 (b).
2. *Household chores.* iRobots is one of the first organizations to venture into this space for exclusive jobs like auto mapping, auto sewage cleaning, and auto lawn mowing. Maid robots initiate and complete household tasks. See Figure 8.15 (c).
3. *Exploring the unexplored.* Robots are involved in exploring the unexplored in the field of medical, training, education, entertainment, and so on as shown in Figure 8.16 (a).
4. *Security and armed forces.* A robot army can save many innocent lives at war. If robots can be programmed and firewalled well, these would be a great help to stop terrorism and infiltration across borders as a country would have a 24-hour fatigue-free army manning its borders.



FIGURE 8.16. (a). Exploring the unexplored. (b) Mars path finder.

Mars Path Finder

The Mars Path Finder has a small robot called the Sojourner which used computer vision techniques to maneuver itself on the surface of the planet Mars. It is shown in Figures 8.16 (b) and 8.17.

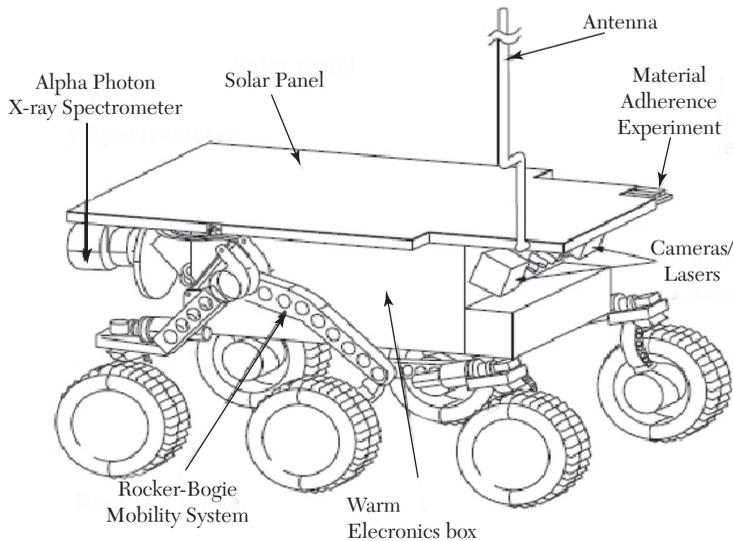


FIGURE 8.17. Mars Pathfinder's Sojourner rover.

Cobots versus Industrial Robots

Cobots are designed to work along with human operators, while industrial ones work in place of humans. Cobots are capable of self-learning during a job, while the latter require an engineer to write new code for any change in process. Cobots are not designed for heavy manufacturing as these work closely with humans. Hence, these are safe enough to function

around humans. Industrial robots, however, can handle heavier and larger materials and require fencing to keep humans out of the workspace. Cobots immobilize at the slightest touch due to sophisticated sensors and, thus prevent any danger to people who are nearby.

A *smart robot* detects its environment, learns from it, and responds accordingly. To detect the environment, it requires sensors like LIDAR, temperature, depth, proximity, and camera. The sensors interact with the environment in real time and generate the required information and responses. The robot checks the information using various algorithms to generate the required responses as per the situation or scenario. It then decides how to act. Applications that utilize smart robots can perform the tasks better, faster, and accurately. Various fields of application of smart robots include aerial (drones), industrial (heavy-duty automated machines), underwater robotics, robotics for intelligent transportation systems, scientific research, manufacturing, mining, agriculture, construction, search-and-rescue operations, space research, medical assistance, and personal assistance.

Machine Learning in Robots

Machine learning has had a significant impact on smart robotic technologies. It integrates vision, imitation learning, self-supervised learning, and assistive and medical technologies. Robot vision helps in structured prediction learning techniques and inspection systems like identification and sorting of objects. Imitation learning provides observational learning used for humanoid robots. It is required for construction, agriculture, search and rescue, military, and other areas. Self-supervised learning enables robots to generate their own training courses to improve performance. It includes road detection algorithms and more to improve performance. A smart camera on a robot is required for machine learning. It acts as a set of eyes that help the robot compare the image of a product against a set of criteria, and makes sure that the criteria are met. Mounting a camera on an industrial arm provides a variety of features for inspection. Robotic applications that require a smart camera are pick-and-place, assembly, packaging and palletizing, quality inspection, screw driving, polishing, labeling, welding, molding, lab testing, and others. For example, in assembly, machine vision guides the parts to be assembled, while the barcode reader reads the label on the parts to make sure correct parts are assembled.

Sensors in Robotic Vision

A typical robot has a movable physical structure driven by motors, sensor circuitry, power supply, and a computer that controls these elements. Some known robots are as follows:

1. Educational robots are mostly robotic arms used to lift small objects, or wheels to move forwards, backwards, left or right, with some other features.
2. Honda's ASIMO robot can walk on two legs like a person.
3. Industrial robots are automated machines that work on assembly lines.
4. BattleBots are remote-controlled fighters.
5. DRDO's Daksh is a battery-operated, remote-controlled robot on wheels. Its primary role is to recover bombs.
6. Robomow is a lawn-mowing robot.
7. MindStorms are programmable robots based on Lego building blocks.

Some robots only have motorized wheels, while others have dozens of movable parts, including spin wheels and pivot-jointed segments with some sort of actuators. Some robots use electric motors and solenoids as actuators, while others use a hydraulic or a pneumatic system. Most autonomous robots have sensors that can move and perform a specific task without human intervention or control.

One interesting technology under development in the field of robotics is the micro-robot for biological environment. A micro-robot may be integrated with sensors, signal processing, a memory unit, and a feedback system working at a micro-scale level. This development can have important implications for integrated micro-bio-robotic systems for applications in biological engineering and research. Some common robotic sensors are given in Table 8.2.

TABLE 8.2. Common sensors used in robots.

COMMON SENSORS USED IN ROBOTS	
Sensors	Functions
Touch	Sensing an object's presence or absence
Force	Measuring force along a single axis
Vision	Detecting edges, holes, and corners

COMMON SENSORS USED IN ROBOTS	
Sensors	Functions
Proximity	Noncontact detection of an object
Physical orientation	Coordinates of objects in space
Heat	Wavelength of infrared (IR) or ultra violet (UV) rays, temperature, magnitude, and direction
Chemicals	Presence, identity, and concentration of chemicals or reactants
Light	Presence, color, and intensity of light
Sound	Presence, frequency, and intensity of sound

Biosensors

Biosensors use a living organism or biological molecules, especially enzymes or antibodies, to detect the presence of chemicals. Examples include blood glucose monitors, electronic noses, and DNA biosensors.

Raindrops sensors

These sensors are used to detect rain and weather conditions, and convert the same into number of reference signals and analogue output. It is shown in Figure 8.18 (a).

Multi-axis force torque sensors

Such sensors are fitted onto the wrist of robots to detect forces and torques that are applied to the tool. It is shown in Figure 8.18 (b).

Optoelectronic torque sensors

These sensors are used for collaborative robot applications, ensuring safer and more effective human-machine-collaboration. These have less than one microvolt of noise even in low-torque range, can self-calibrate

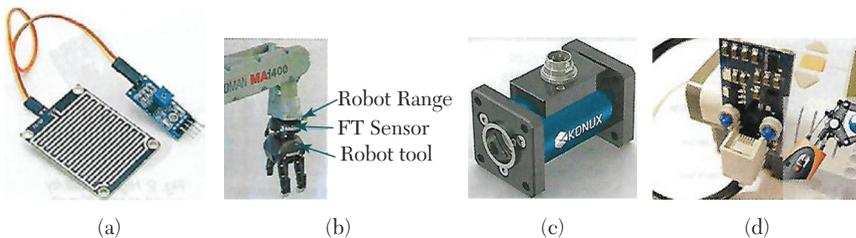


FIGURE 8.18. (a) Raindrop sensor. (b) 6-axis force torque sensor fitted onto robotic arm. (c) Optoelectronic torque sensor. (d) Inertial sensor mounted on a robot.

and measure to an accuracy of 0.01%. A contactless measurement principle ensures that the sensors are insensitive to vibration and are wear resistant. It is shown in Figure 8.18 (c).

Inertial sensors

Motion sensing using MEMS inertial sensors is applied to a wide range of consumer products including computers, cell phones, digital cameras, gaming, and robotics. There are various types of inertial sensors depending on applications. It is shown in Figure 8.18 (d).

Sound sensors

Sound sensors are used in robots to receive voice commands. High-sensitivity microphones or sound sensors are essential in voice assistants like Google Assistant, Alexa, and Siri. It is shown in Figure 8.19 (a).

Emotion sensors

Robots can react to human facial expressions using emotion sensors. The B5T HVC face-detection sensor module from Omron Electronics is a fully integrated human vision component (HVC) plug-in module that can identify faces with speed and accuracy. The module can evaluate the emotional mood based on one of the five programmed expressions. The sensor is shown in Figure 8.19 (b).

Grip sensors

RoboTouch Twenty-One gripper features embedded digital output, which is ideal for OEM integration into a gripper. These sensors comprise multiple sensing pads, each having multiple sensing elements, and so on. Data from sensors is transferred to grippers via I2C or SPI digital interfaces, enabling easy integration of the sensors into grippers. The gripper is shown in Figure 8.19 (c).

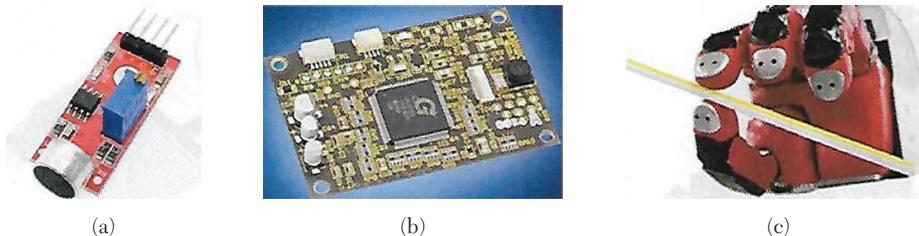


FIGURE 8.19. (a) High sensitivity sound sensor. (b) BST HVC face-detection sensor module. (c) RoboTouch Twenty-One gripper.

Artificial Intelligence Robots

Sensors used in AI robots are the same as, or are similar to, those used in other robots. Fully functional human robots with AI algorithms require numerous sensors to simulate a variety of human and beyond-human capabilities. Sensors provide the ability to see, hear, touch, and move like humans. These provide environmental feedback regarding surroundings and terrain.

Distance, object detection, vision, and proximity sensors are required for self-driving vehicles. These include camera, IR, sonar, ultrasound, radar, and LIDAR. A combination of various sensors allows an AI robot to determine size, identify an object, and determine its distance. Radio-frequency identification (RFID) tools are wireless sensor devices that provide identification codes and other information. Force sensors provide the ability to pick up objects. Torque sensors can measure and control rotational forces. Temperature sensors are used to determine temperature and avoid potentially harmful heat sources from the surroundings. Microphones are acoustical sensors that help the robot receive voice commands and detect sound from the environment. Some AI algorithms can even allow the robot to interpret the emotions of the speaker.

Humanoid robots with AI algorithms can be useful for future distant space-exploration missions. Atlas is a 183cm (6-feet) tall, bipedal, humanoid robot, designed for a variety of search-and-rescue tasks for rough outdoor terrains. It has an articulated sensor head that includes a stereo camera and a laser range finder.

Robotic Vision Testing in the Automotive Industry

Automobiles in modern times are becoming as much computer as they are vehicle. They include GPS, navigation systems, touchscreen and/or voice-activated audio and video systems, rear-view camera systems, lane-change assist device, and so on. These infotainment systems are quickly becoming standard issue for the new automotive industry. There is a fine line between enhancement of the driving experience and distraction. A simple “glitch” in one of these systems could possibly lead to enough distraction in busy traffic to cause an accident that results in injury or death. Reliability and accuracy of infotainment systems is critical, and rigorous testing should be applied to components, both during product development and on the assembly line.



FIGURE 8.20. Robotic vision testing in the automotive industry.

Business Insider predicts there will be approximately 10 million self-driving cars by the year 2020. Testing of reliability of automobile components has reached critical mass. Failure of some of these systems cannot be an option. Robotic testing of these components is a great solution to provide unmatched duplication, speed, and accuracy. It is shown in Figure 8.20.

Recently, tactile testing of touchscreen devices, levers/ buttons, and keypads could only be accomplished by human hands. SR-SCARA-Pro is a lightweight SCARA robot that can be customized for testing Infotainment system components. For touchscreen testing, the SR-SCARA-Pro goes light years beyond human capability, performing up to 800 touches per minute, without the element of human error. Easily programmable, and having the ability to interface with existing software used for testing, it has the capability to provide remote accurate testing in a fraction of the required time. With the ability to touch, press, and swipe, it can easily master the demand of a multi-faceted control panel of any type, thus providing the repeated and accurate testing needed to ensure maximum safety and reliability of automobile infotainment systems.

Automotive infotainment systems are one area where rigorous testing cannot be undercut. SR-SCARA-Pro has the unique ability to adapt to a large range of testing environments. Basic control modules are included with the package, making it possible for developers to create their own software solutions to provide testing of their prototypes. Using this unique, robotic solution can insure ultimate reliability under all anticipated conditions, which means safer driving for the end user. Consumers want Infotainment systems that offer PC-grade responsiveness, easily navigated menus, connectivity with PDAs and Smartphones, and a plethora of options. These are features that drive competition in the marketplace, and innovation is moving at a rapid pace. Integrated automation is the trend, and testing needs to be scaled up to accommodate that need. SR-SCARA Pro is ready to adapt to the need for testing, insuring safety for vehicles of the future.

8.5 ROBOTIC TESTING IN THE AVIATION INDUSTRY

The robot arm quickly activates a well-placed camera, and begins the preflight check of the instrument panel, physically activating the toggles, switches, and touch-panel functions, to insure operational readiness. A lightweight SCARA robot can be customized for testing of commercial control panels. As stated earlier the robot performs up to 800 touches per minute. It is shown in Figures 8.21 and 8.22 (a).

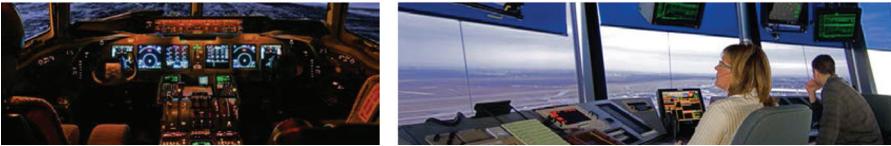


FIGURE 8.21. Robotic testing in the aviation industry.

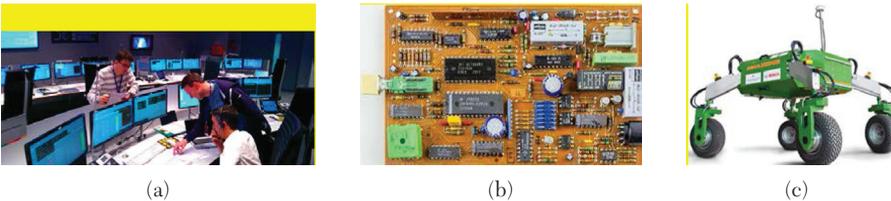


FIGURE 8.22. (a) Aviation industry test. (b) Electronics industry test. (c) AgBot II, a solar-powered micro-robot.

Easily programmable and having the ability to interface with existing software used for testing, the SR-SCARA-Pro has the capability to provide remote, accurate testing, in a fraction of the required time. With the ability to touch, press, and swipe, SR-SCARA-Pro can easily master the demand of a multi-faceted control panel. An example of this would be an air traffic controller in the United States remotely managing and directing air traffic in Asia.

Safe operation of an aircraft depends on every component being able to perform correctly under both normal and emergency conditions. Aircraft are historically rarely subject to mechanical failure, but with increasing automation being the trend in the aviation industry, the possibility of system-related error also increases. The need for in-depth testing of these flight systems is on the rise, to include:

- in-flight navigation systems,
- communications systems,

- air-traffic control systems,
- in-flight entertainment systems,
- cockpit controls,
- security systems,
- flight-simulation systems,
- aerospace industrial-engineering systems, and
- high-end prototypes of aircraft control systems and components.

Intensive checks in the production of these systems are time-consuming, as well as subject to human error. Robotic checking can provide superior efficiency, reducing time, cost, and the number of errors that will allow for further expansion and growth. With the ability for remote testing and remote monitoring of test results, the SR-SCARA-Pro can allow for multinational teams to evaluate the reliability and efficiency of their designs with accurate repetitive techniques.

Robotic Testing in the Electronics Industry

The global electronics industry is worth a staggering \$1.8 trillion every year in value. It is one of the most sought out after sectors for technology innovations because the success of key players depends on how effectively they are able to offer great devices at fabulous prices at profitable margins to win customer hearts.

Price optimization is one area that depends on how well you can source components and how effectively you can manage logistics to come up with favorable market supply. Being undoubtedly one of the largest industries in the world electronics, manufacturers are facing newer challenges in ensuring quality of the devices they produce in mass scales. Over the past decade, people are increasingly relying on electronic devices that do more than what they did previously, or in other words people are buying smarter electronic devices. From the smallest handheld MP3 players to large smart-refrigerators, the avenue of smart consumer devices has grown tremendously. Nearly all of them have touchscreen-enabled interfaces and also have embedded software installed in them to operate as a smart device according to user interactions.

In fact the Internet of Things (IoT) enabled smart homes will be one of the most invested sector in the coming years and smart homes will definitely include smart appliances such as smart televisions, refrigerators, and so on.

To offer great and reliable electronic devices, however, it is imperative to offer great levels of product stability, intuitiveness, and most importantly a fault-free product. Over 24% of all accidental home fires are reported to have occurred due to faulty electronic appliances.

This is where product testing is viewed as a major requirement. Every year companies spent billions of dollars on testing product quality and reports suggest that nearly 35% of the overall QA budgets in the electronics industry is reserved for testing user interfaces, such as touchscreen displays embedded on appliances. Employing a significant workforce to achieve this testing feat will push up operational expenses by a huge margin, and thus companies are always looking for newer avenues to test electronic device interfaces with minimal cost.

In India alone the market for electronics and home appliances would reach a staggering \$400 billion by 2020 according to expert studies.

The SR-SCARA-Pro robot is the game changer for this scenario as it ensures that electronic devices are well tested before deployment towards actual usage without human intervention. Each device needs to pass through various rounds of quality assurance (QA) measures to ensure that end users do not face any hassles while using them on a regular basis. And this QA process starts not after a final product comes out of the production line but right from the time successful prototypes are designed from great ideas. With the SR-SCARA-Pro robot, there is a huge opportunity for companies to prevent faulty devices from being shipped to stores from their production lines.

The robot is intelligently designed to collaborate with both external hardware components such as touch screens, buttons, knobs, and other control devices, as well as with software-control modules that will ensure the end product is trustworthy and market-ready. The SR-SCARA-Pro robot can integrate with renowned software quality-assurance frameworks and control systems to automate the testing of all software components and check against desired functionality outputs. On the hardware side, the robot ensures that sensitivity, touchscreen intuitiveness, and usability are thoroughly tested during the entire production lifecycle. With the SR-SCARA-Pro robot, manufacturers have several advantages when compared to legacy manual QA measures. They are:

- faster QA checks,
- significant cost-cutting in the long run,

- zero errors due to avoidance of human contact,
- lower workforce requirement,
- better products,
- improved demand-supply balance, and
- increased consumer adaptation.

Automation of electronics testing will help manufacturers save costs significantly and invest on higher value processes such as research and development. Employing intelligent robotic assistants such as the SR-SCARA-Pro will enable manufacturers to come out with ready to deploy electronic devices and appliances that have been thoroughly tested. Since all test cases for the hardware interfaces such as touchscreen displays, switches, and knobs could be automated, there is no scope for miss outs and errors. It is shown in Figure 8.22 (b).

The SR-SCARA-Pro will definitely bring about a digital transformation in the electronics appliance testing segment and allow manufacturers to save significantly in terms of costs and lower manpower requirements. However, the best thing about it is the ability to produce error-free products that are shipped to consumers, which in turn translates into satisfied and loyal buyers.

The Use of Drones and Robots in Agriculture

Smart agriculture can be revolutionized by the use of drones and robotics. One exciting application is crop imaging. Using drones equipped with multispectral sensors, farmers can survey their land, take images that reveal information like fertility of specific patches of soil, determine the amount of water the crops need, and more. In the past, farmers had to rely on satellite imaging to get detailed maps of their land. This process often took 14 days. With drones, the same can be carried out when the farmers want. Agricultural drones can be used for monitoring plant health, counting plants, spraying insecticide, and pesticides on crops, scheduling seeding/harvesting, classifying management zones, reducing usage of scarce resources, recording data for future analysis, and increasing yields by using resources effectively. Some real-life applications of robots for smart farming are given below.

Nursery planting —There is a rising need for nursery automation. Companies like HETO Agrotechnics and Harvest Automation provide

automation solutions for seeding, potting, and warehousing for plants in greenhouses.

Crop seeding—Autonomous precision seeding combines robotics with geo-mapping. A map is generated that shows different properties (quality, density, etc.) at every point in the field. A tractor with robotic seeding attachment places the seeds at precise locations and depths, so that each seed has the best chance of growing. Crop monitoring and analysis: Drone-making companies like Precision Hawk offer farmers packages that include robotic hardware and analysis software.

Ground-based robots, like Boni Rob provide detailed monitoring as these can work in closer proximity to the crops. Some robots can also be used for tasks like weeding and fertilizing. AgBot II is a solar-powered micro-robot. This spraying robot uses computer-vision technologies to detect weeds and to spray targeted drops of herbicide onto them. Lettuce Bot is a thinning robot that uses computer vision to detect lettuce plants as it drives over them. It decides at that moment which plants to keep and which to remove. Wall-Ye, an autonomous vineyard robot, can prune grapevines. The company behind Wall-Ye has developed a blueberry pruning robot, too. Robots are developed for picking, harvesting, shepherding, herding, and milking. Figure 8.22 (c) shows AgBot II, a solar-powered micro-robot.

Underwater Robots

Underwater drones and portable submersible robots with intelligent sensors for monitoring water temperature, auto-compass heading, and depth, turns, pitch, and roll have been developed. Due to technological advancements, underwater drones can now move around freely, faster, and more easily. Robots and artificial intelligence have been developed to explore the oceans where humans cannot reach. The OceanOne robot has a human-like body with touch sensors that can carry out a variety of



FIGURE 8.23. (a) Robotic mermaid with touch sensors. (b) Knightscope robot. (c) Robotic patrolling prison. (d) Ayuda robot.

tasks in the deep sea. Its wrists are fitted with force sensors that give haptic feedback to the operator's control. These feedback signals can identify actions things such as the robot grasping at something firm and heavy, or light and delicate. The brain of the robot reads data and makes sure that its hands keep a firm grip on objects. It is shown in Figure 8.23.

Autonomous Security Robots

Knightscope K5 is a 1.8m- (6-feet) tall autonomous security robot, packed with sensors like lidar array and cameras. It is shown in Figure 8.23(b). It moves about parking lot aisles, hallways, offices, stadiums, and shopping malls on the prowl for suspicious activity. It can differentiate between a harmless passerby and potential criminal, and feed all that data to the cloud. K5 is not meant to replace security guards. It is fully-autonomous security data machine to fill blind spots. This robot can be deployed where it is not safe for human guards to patrol e.g., under dangerous bridges and crime-ridden public parking lots.

Protectors of law dedicate their lives to keep our cities safe. However, it is not possible for them to reach every place in need, in time. In this regard, robots can bring great benefits. As they are becoming more interactive and intelligent, robots have become helping hands for police personnel. It is shown in Figure 8.23 (c). Robots can be put to use in different applications on the streets. With the amalgamation of various sensors, motors and output channels, these can feed on necessary information in real time, and provide assistance as required. Main sensory components of these machines are lidars, auto-sonic sensors, touch sensors, cameras with computer vision, microphones, and input touchscreens and so on. Inputs picked up by the sensors are processed in a microcontroller (MCU) to create relative outputs.

Taiwan police recently deployed robots for citizen assistance purposes. The Ayuda robot, built by Syscom Computer Engineering, can file and print formal police complaints for citizens, run citizen-ID verification, provide road assistance and more. It can also be used for surveillance and security purposes powered by the various technologies lying underneath the design. It is shown in Figure 8.23 (d). The robot combines multiple sensors and utilizes techniques like facial recognition for citizen identification, natural-language processing-based speech for interaction and computer vision for surroundings situation monitoring. A built-in microphone facilitates two-way communication. Ayuda's body is made of acrylonitrile-butadiene-styrene (ABS). It is 1.6 meters tall and weighs 100 kilograms. Using a lithium battery, it can run for eight hours, following four hours of charge.

Summary

- Machine / Computer Vision basic components are imaging device, computer, and image-processing software.
- Image acquisition, image processing, image analysis, understanding, and finally decision making are the components of computer vision.
- Preprocess image, scan image for potential feature locations, filter feature locations, generate signatures of confirmed features, and post-process feature descriptors are the steps in feature extraction.
- Object detection defines an object within images and involves outputting bounding box and labels for individual objects.
- Object tracking refers to the process of following a specific object of interest in a given scene.
- Object recognition is a process for identifying a specific object in a digital image or video using any of the following algorithms: SIFT, ASIFT, SURF, and ORB.
- Optical-flow techniques are needed for motion analysis in video sequences.
- The Mars Path Finder is a small robot which uses computer-vision techniques to maneuver itself on the surface of the planet Mars.
- A robot has movable physical structures driven by motors, sensor circuitry, and power supply. A computer controls these elements.
- Touch, force, vision, proximity, physical orientation, heat, chemicals, light, and sound are the common sensors used in robots.
- The automotive industry, aviation industry, electronics industry use robotic vision for testing.

References

<https://blog.robotiq.com/robot-vision-vs-computer-vision-whats-the-difference>

<https://www.techopedia.com/definition/32309/computer-vision>

https://www.visiononline.org/Vision-vs-Machine-Vision/content_id/4585

http://computervision.wikia.com/wiki/Machine_vision

<http://www.rsipvision.com/defining-borders/>

<http://shervinemami.info/embeddedVision.html>

https://heartbeat.fritz.ai/the-5-computer-vision-techniques-that-will-change-https://www.tutorialspoint.com/dip/pdf/dip_quick_guide.pdf

www.digitaltrends.com

Learning Outcomes

- 8.1 Write about the impact of other technologies in embedded vision.
- 8.2 Compare image processing, computer vision, and embedded vision.
- 8.3 Define robot vision.
- 8.4 Write the similarities between computer vision and machine vision,
- 8.5 Differentiate computer vision, image processing, and machine learning.
- 8.6 Draw the components of computer vision. Explain them.
- 8.7 List a few algorithms used in computer vision.
- 8.8 What is meant by feature extraction?
- 8.9 What are different algorithms for feature extraction?
- 8.10 Draw the block diagram of feature extraction.
- 8.11 What is the image classification?
- 8.12 Write about the object detection.
- 8.13 Give a short note on object tracking.
- 8.14 What is meant by semantic segmentation?
- 8.15 What are the different algorithms for object recognitions?
- 8.16 Write the scale invariant feature transform algorithm (SIFT).
- 8.17 Write about speed-up robust feature algorithm.
- 8.18 Write about ORB algorithm.
- 8.19 Write a short note on optical-flow algorithm.
- 8.20 List few applications of computer vision.
- 8.21 Write different sensors used in robots.
- 8.22 Write about the application area of robotic vision.
- 8.23 Write a short note on sensors in robotic vision.

8.24 Write about the role of testing work by robots in industry.

8.25 What is the role of robots in security work?

Further Reading

1. *Programming computer Vision with python: Tools and algorithms for analyzing images* by Jan Erik Solem,
2. *Robot visions* by Isaac Asimov.

ARTIFICIAL INTELLIGENCE FOR EMBEDDED VISION

Overview

Artificial Intelligence (AI) with embedded vision aims to create systems that can function intelligently and independently just like humans. Supervised learning, unsupervised learning, and reinforcement learning are some of the learning algorithms used to make machines that are artificially intelligent. Smart cameras with built-in AI capabilities offer conveniences in human lifestyles. Artificial vision is the use of special technology to give some vision to people without natural sight. Some examples of 3D-imaging technologies are stereo vision, structured light, laser triangulation, and the starting point of a file (ToF). Stereo vision is the process of extracting 3-D information from multiple 2-D views of a scene using multiple cameras. Camera choice, hardware scalability, ease of the use of software, algorithm accuracy, price, and integration ability are the choices in the embedded-vision software selection.

Learning Objectives

After reading this one will be able to know the

- role of AI in embedded vision applications,
- AI in personalized styling, shopping, and other applications,
- working of intelligence surveillance system,
- AI embedded in cameras,
- digital object-recognition audio-assistant (DORA) for visually impaired persons,

- 3D-imaging technologies working principles, comparison, applications, and advantages,
- embedded-vision safety standards, and
- considerations in choosing embedded-vision software.

9.1 Embedded Vision-based Artificial Intelligence

Artificial intelligence (AI) technology primarily comes in the form of machine learning and deep convolutional neural networks to help vision systems learn, distinguish between objects, and even recognize objects. AI is helping bring vision technology into unprecedented territory.

One primary reason for the use of AI in vision systems is the rise of the Industrial Internet of Things (IIoT). The IIoT features machine-to-machine communication in a highly automated environment that's dependent upon embedded vision to identify a wide range of objects within the factory and throughout the process of the flow of goods. Further, a vision system's ability to actually recognize objects, as well as a variety of defects in an object, can significantly improve the accuracy of a vision system. For inspection applications, for example, this accuracy translates directly into productivity and profitability.

Consider the following situation: A terrorist bombing on a busy street was full of cameras, both those permanently installed by law enforcement organizations and businesses, and those carried by race spectators and participants. But none of them was able to detect the forthcoming threat represented by the backpacks, each containing a pressure cooker-implemented bomb, with sufficient advance notice to prevent the tragedy. And the resultant flood of video footage was predominantly analyzed by the eyes of the police department attempting to identify and locate the wrongdoers, due to both the slow speed and low accuracy of the alternative computer-based image analysis algorithms.

Consider, too, the ongoing military presence in any of the country, as well as the ongoing threat to country embassies and other facilities around the world. Only a limited number of human surveillance personnel are available to look out for terrorist activities such as the installation of improvised explosive devices (IEDs) and other ordinances, the crowd and movement of enemy forces, and the like. And these human surveillance

assets are further delayed by fundamental human shortcomings such as distraction and fatigue.

Computers, however, don't get sidetracked, and they don't need sleep. More generally, an abundance of ongoing case studies, domestic and international alike, provide ideal opportunities to tackle the tireless analysis assistance that embedded vision processing can deliver. Automated analytics algorithms are conceptually able, for example, to search through an abundance of security camera footage in order to pinpoint an object left at a scene and containing an explosive device, cash, or smuggled or other contents of interest to investigators. And after capturing facial features and other details of the person(s) who left the object, analytics algorithms can conceptually also index image databases both public (Facebook, Google Image Search, etc.) and private in order to rapidly identify the suspect(s) and instruct for necessary preventive activity.

Such automated surveillance technology shortcomings are rapidly being surmounted, however, as cameras (and the image sensors contained within them) become more feature rich, as the processors analyzing the video outputs similarly increase in performance, and as the associated software therefore becomes more robust. *The term "embedded vision" refers to the use of computer vision in embedded systems, mobile devices, PCs, and the cloud.* Historically, image-analysis techniques have typically only been implemented in complex and expensive, therefore niche, surveillance systems. However, the previously mentioned cost, performance, and power consumption advances are now paving the way for the proliferation of embedded vision into intelligence diverse surveillance and other applications.

Vision processing has added artificial intelligence to surveillance networks, enabling "aware" systems that help protect property, manage the flow of traffic, and even improve operational efficiency in retail stores. In fact, vision processing is helping to fundamentally change how the industry operates, allowing it to deploy people and other resources more intelligently while expanding and enhancing situational awareness.

Motion detection, as its name implies, allows surveillance equipment to automatically signal an alert when frame-to-frame video changes are noted. A historically popular technique to detect motion relies on "codex" motion vectors, a byproduct of the motion estimation employed by video compression standards such as MPEG-2 and MPEG-4/H.264. Because

these standards are frequently hardware accelerated, scene-change detection using motion vectors can be efficiently implemented even on modest IP camera processors, needing no additional computing power. However, this technique is susceptible to generating false alarms, because motion vector changes do not always coincide with motion from objects of interest. It can be difficult to impossible, using only the motion vector technique, to ignore unimportant changes such as trees moving in the wind or casting shifting shadows, or to adapt to changing lighting conditions.

These annoying “false positives” have unfortunately contributed to the perception that motion-detection algorithms are unreliable. Nowadays, however, an increasing percentage of systems are adopting intelligent motion-detection algorithms that apply adaptive background-modeling along with other techniques to help identify objects with much higher accuracy levels, while ignoring meaningless motion artifacts. Even under more challenging environmental conditions, such as poor or wildly fluctuating lighting, precipitation (rain, snow, etc.) induced substantial image degradation, or heavy camera vibration, accuracy can still be near 70%.

The capacity to accurately detect motion has several related event-based applications, such as object counting and trip zone. As the name implies, counting tallies the number of moving objects crossing a user defined imaginary line, while tripping flags an event each time an object moves from a defined zone to an adjacent zone. Other common applications include loitering, which identifies when objects linger too long, and object left-behind/removed, which searches for the appearance of unknown articles, or the disappearance of designated items.

Robust artificial intelligence often requires layers of advanced vision know-how, from low-level imaging processing to high-level behavioral or domain models. As an example, consider a demanding application such as traffic and parking lot monitoring, which maintains a record of vehicles passing through a scene. It is often necessary to first deploy image stabilization and other compensation techniques to retard the effects of extreme environmental conditions such as dynamic lighting and weather. Compute-intensive pixel-level processing is also required to perform background modeling and foreground segmentation.

To equip systems with scene understanding sufficient to identify vehicles in addition to traffic lanes and direction, additional system competencies handle feature extraction, object detection, object classification (i.e., car, truck, pedestrians, etc.), and long-term tracking. License plate recognition

(LPR) algorithms and other techniques locate license plates on vehicles and discern individual license plate characters. Some systems also collect metadata information about vehicles, such as color, speed, direction, and size, which can then be streamed or archived in order to enhance subsequent forensic searches.

In terms of the market, the most spectacular progress can be noted in the car, as these technologies are used in advanced driver-assistance systems (ADAS) for the detection of obstacles and the recognition of signs, traffic lights, cars, pedestrians, and assorted others. The images come from a bank of cameras arranged on and around the car, while the training is performed in data centers in dedicated computing machines, and the inference algorithm is embedded either in an engine control unit (ECU) in the case of semi-autonomous cars or in a complete computer in the case of robotic or fully autonomous cars.

In order for a machine to actually view the world like people or animals do, it relies on artificial intelligence to embedded vision and image recognition. It's how Apple's Face ID can tell whether a face its camera is looking at is yours. Basically, whenever a machine processes raw visual input—such as a JPEG file or a camera feed—it's using embedded vision to understand what it's seeing. It's easiest to think of embedded vision as the part of the human brain that processes the information received by the eyes not the eyes themselves. One of the most interesting uses of embedded vision, from an AI standpoint, is image recognition, which gives a machine the ability to interpret the input received through embedded vision and categorize what it “sees.”

Any AI system that processes visual information usually relies on embedded vision, and those capable of identifying specific objects or categorizing images based on their content are performing image recognition. This is incredible important for robots that need to quickly and accurately recognize and categorize different objects in their environment. Driverless cars, for example, use vision and image recognition to identify pedestrians, signs, and other vehicles.

AI can be used in numerous ways along with vision systems. Inspection applications are some of the first jobs that AI has been profitable in, specifically when leveraging machine-learning algorithms for defect detection and classification. The cost of acquiring and labeling large datasets has decreased in the past few years due to advances in IIoT, making machine learning more accessible than ever for inspection applications.

The other main way that AI is used in vision systems is for continuous improvement in recognition applications. This could be deployed in nearly any scenario in which vision systems are used for object recognition. Typically, incorrect predications can be identified and associated with recorded data, so a vision system can continuously learn and improve itself based on its own mistakes. Of course, there are many other ways in which AI can be used in tandem (racing bike) with vision systems, but inspection applications and continuous improvement for object recognition tasks are some of the most common and practical uses.

AI is a revolutionary technology and it is controlled to transform the way we use vision systems in practically every type of deployment around the globe. AI enhances the capabilities of vision systems, making them smarter, more flexible, and constantly improving.

Embedded vision is a scholastic term that depicts the capability of a machine to get and analyze visual information along with sensor information about environment all alone and afterward settle on decisions about it by using actuators. That can consist of text, sensed information, photographs, and videos, yet more comprehensively may consist of “pictures” from thermal, or infrared sensor, indicators and different sources. Embedded vision permits computers, robots, other computer-controlled vehicles, and everything from processing plants and farm equipment to semi-independent cars and drones, to run all the more productively and shrewdly and even securely. In any case, embedded vision’s significance has turned out to be considerably increasingly evident in a world bombarded with digital pictures.

Since the advent of camera-prepared smartphones, we’ve been accumulating amazing measures of visual symbolism that, without somebody or something to process everything, is far less valuable and usable than it ought to be. We’re now witnessing computer vision that enables purchasers to compose and get access to their photograph gallery in, say, Google Photos, with over the billions of pictures shared online.

To get a thought of the amount discussed here, a year ago photograph printing service Photoworld did the math and discovered it would take an individual 10 years to try and review all the photographs shared on Snapchat, that had been posted in one hour. What’s more, obviously, in those 10 years, an additional 880,000 years’ worth of photographs would have been produced if things proceeded at a similar rate. Basically, our

reality has turned out to be progressively loaded up with digital pictures and we require computers to understand everything. Currently, it's well beyond human capacities to keep up.

AI-Based Solution for Personalized Styling and Shopping

To help retail stores deliver personalized styling suggestions to customers, the company FR Tech Innovations, came up with the idea of iLUK. It is a computer vision and artificial-intelligence-based solution. An iLUK pod is placed in a retail outlet. Customers enter the pod, and get photographed and scanned from different angles, using a set of RGB cameras. Once the cameras capture the images of the customers, the images are passed through custom algorithms to construct a 3D model of the customer called an *avatar*. iLUK encrypts the 3D-model avatar using the customer's credentials (identification) and stores them on a cloud. The customer can access the 3D-model avatar from any device tablet, phone, or PC using a profile ID and password and shop for any part of a wardrobe and get personalized styling suggestions. Computer vision is used to capture and create 3D models and digitize the garments in 3D. AI is used to analyze body type and suggest appropriate clothing options based on body type and fashion sensibilities.

Cashier-less stores have recently opened in several countries. At the store, users need to follow three simple steps: download the store's app, open it, and scan the unique QR code while entering the store. Customers can pick the items they want and pay for them via different online payment options. The system combines technologies such as QR codes, computer vision, sensor fusion, embedded vision, deep learning, and AI. The QR code, which appears on the user's phone screen after logging into the app, acts as a unique identification for customer. After scanning it at the entrance, the system registers the information that the user has entered the store. Cameras provide vision to the system without compromising privacy. Sensor fusion or embedded vision combines sensors like proximity, motion, and weight. As soon as an item is picked up, multiple sensors complement the computer vision. This gives credibility to the results. Sensors smoothly integrate into the store. These generate data for the AI-powered system to make decisions. AI and deep learning analyze the interaction between the customer and store.

ModiFace gives clients a chance to try cosmetics utilizing just their cell phones. Topology does likewise for eyewear. MTailor makes custom-fitted

pants and shirts employing a similar procedure. Outside of fashion, Pottery Barn (a U.S.-based home furnishings store) gives clients a chance to perceive what new furniture may look like in their homes, and Hover transforms clients' photos of their homes into completely estimated 3D models.

Microsoft made an algorithm had a mere 3.5% inaccuracy rate when identifying what was in pictures. That implies it was right 96.5% of the time. Tesla cars depend on a large group of cameras and sonar that not just keep vehicle from floating out of a path, however, but can also perceive what different objects and vehicles are around it and furthermore obey signs and traffic signals. Computer vision will empower better approaches for doing diagnostics and for analyzing X-rays, MRI, CAT, mammography, and different outputs. All things considered, nearly 90% of every medical data is picture-based.

Deep learning has been very successful in many segments over the past 10 years. Image-based technologies include facial recognition, iris and gesture monitoring, object and free-space detection, and more recently, behavioral recognition.

Biometrics is another major segment where deep learning is widely used. These algorithms are used for the authentication of an individual. The latest Apple phone, the iPhone X, is a notable example thanks to facial recognition in 3D. In surveillance and homeland security, facial recognition is used in border controls and in the production of identity papers through the use of specialized cameras. Iris recognition based on deep learning for the authentication of an individual is also increasingly used with a desire for use in mobile devices. Finally, behavioral recognition is added in this segment.

Deep learning is currently integrated in gesture recognition though mainly in the entertainment segment, with on-board computers in the car, gaming, commercial drone controls, and so on. The major players in each of these areas are well known. There are Google, Amazon, Facebook, and Apple. The investment of these companies in the AI field has been consistent over the last 10 years.

In order to make a machine capable of understanding the world around it, technology has been inspired by biology. The information enabling humans to locate their place in, and interact with, their universe passes through their eyes 80% of the time. Much of the research in AI has therefore focused on the ability to analyze images from vision systems. The other

main inspiration from biology is the mathematical structure that allows the machine to analyze these images: artificial neural networks, a miniature structural copy of the human brain.

There are a multitude of different neural networks depending mainly on the topology of the connections between neurons, the aggregation function used, the threshold function, and the back propagation method (if present, the network is called a convolutional neural network CNN). These mathematical methods are all part of the field of artificial intelligence called “deep learning,” and are broken down into two parts: training and inference.

The vast majority of neural networks have a very variable “training” algorithm (supervised or not) according to the goal to be achieved. The algorithm modifies the synaptic weights according to a data set presented at the input of the network. The goal of this training is to enable the neural network to “learn” from examples.

If the training is properly performed, the network will provide output responses very similar to the input values of the training data set. An inference engine is a software algorithm corresponding to a simulation of deductive reasoning, the neuron network in the case of deep learning. This software is often embedded in the device.

The AI development also cannot be disassociated from specialized hardware development. It is interesting to note that designers and builders of vision processors also provide a software layer via an embedded operating system and/or a software development kit (SDK).

This makes it very easy to implement software solutions and allows the hardware to be used to the best of its capabilities, while also requiring platform-specific development skills using tools such as embed OS from ARM (Cambridge, UK; www.arm.com), Jetson from NVIDIA (Santa Clara, CA, USA; www.nvidia.com), XSDK from Xilinx Inc. (San Jose, CA, USA; www.xilinx.com), and CDNN toolkit from CEVA (Mountain View, CA; USA; www.ceva-dsp.com).

The market for computer vision is developing nearly as fast as the capacities. It’s anticipated to reach \$26.2 billion by 2025, developing more than 30% each year. Artificial intelligence is the future, and computer vision is the most amazing appearance of that future. Before long, it will be anyplace and all over the place, to such an extent that you won’t even notice it.

AI Learning Algorithms

Artificial Intelligence (AI) is a broad branch of computer science, aiming to create systems that can function intelligently and independently just like humans. *AI is an imitation of human intelligence processes by machines.* The intelligence processes include learning, reasoning, and self-correction. Specific AI applications include embedded vision, machine vision, speech recognition, and expert systems. The subfields of AI are outlined in the following sections.

Speech recognition

Humans can speak and listen to communicate through language; this is the field of speech recognition. Since speech recognition is statistically based, hence it's called statistical learning.

Natural language processing

Humans can write and read the text in a language; this is the field of NLP or natural language processing.

Machine vision

Humans can see with their eyes and process what they see; this is the field of computer vision. Computer vision falls under the symbolic way that computers process information. Moreover, they can recognize the surroundings around them through their eyes which creates images of that world. This field of image processing which even though is not directly related to AI but it is required for computer vision. This in turn is used in embedded vision.

Robotics

Humans can understand their environment and move around fluidly; this is the field of robotics.

Pattern recognition

Humans can see patterns such as grouping of like objects; this is the field of pattern recognition. Machines are even better at pattern recognition because they can use more data and dimensions of data; this is the field of machine learning.

If we can replicate the structure and the function of the human brain, we might be able to get cognitive capabilities in machines; this is the field

of neural networks. If these networks are more complex and more in-depth and we use those to learn complicated things that is the field of deep learning. There are different types of deep learning and machines which are fundamentally different techniques to replicate what the human brain does.

If we get the network to scan images from left to right, top to bottom, it's a convolution neural network (CNN). A CNN is used to recognize objects in a scene; this is how computer vision fits in object recognition is accomplished through AI.

Humans can remember the past as what you had for dinner last night, well at least most of us. We can get a neural network to recognize a limited past this is a recurrent neural network. As you see there are two ways an eye works, one is symbolic-based, and another is data-based. For the database side, it is called machine learning as we need to feed the machine lots of data before it can learn.

For example, if you had lots of data for sales versus advertising spend you can plot that data to see some pattern. If the machine can learn this pattern, then it can make predictions based on what it has learned.

While one or two or even three dimensions is natural for humans to understand and learn, machines can learn in many more aspects like even hundreds or thousands. That's why devices can look at lots of high-dimensional data and determine patterns. Once it learns these patterns, it can make predictions that humans can't even come close to. We can use all these machine-learning techniques to do one of two things: classification or prediction.

As an example, when you use some information about customers to assign new customers to a group like young adults, then you are classifying the customer.

There are certain learning algorithms which are used in AI-development process to run the machines smartly. Here are some of the learning algorithms used to make machines artificially intelligent.

Supervised learning

If you train an algorithm with data, which also contains the answer, then it's called supervised learning. For example, when you train a machine to recognize your friends by name you'll need to identify them for the computer.

Unsupervised learning

If you train an algorithm with data where you want the machine to Figure out the patterns, then it is called unsupervised learning. For example, you might want to feed the data about celestial objects in the universe and expect the machine to come up with patterns in that data.

Reinforcement learning

If you give any algorithm a goal and expect the machine to achieve that goal through trial-and-error, it's called reinforcement learning. A robot's attempt to climb over a wall until it succeeds is a good example.

Deep learning and vertical image recognition

AI development has the potential to reconstruct the businesses with its innovative features like natural language processing, machine learning, image processing, and robotic process automation. Artificial intelligence is already changing how many industries operate.

In a world where the biggest players like Facebook and Instagram limit how much of their content other actors are able to tap into, there's been a rise in open-source projects such as ImageNet. ImageNet's mission is to create a large-scale image database that researchers can tap into in order to train and manufacture their algorithms.

The challenge is that in order for computers to index and catalogue these huge sets of data, they initially need to have some human input in terms tagging and classifying their "training images." Deep-learning algorithms then use this information to create benchmarks to compare with future images, but need to be fed as many as tens of millions of training images. Steps in identifying a horse in images is shown in Figure 9.1

Industrial image processing is where AI helps capture many different states and/or product features, and evaluates these via intelligence pattern recognition to enable more reliable quality control. Forbes states that AI can increase error detection rate by up to 90%. Cooperative and collaborative

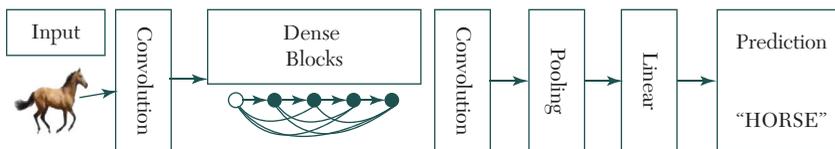


FIGURE 9.1. DenseNet visualization.

robots that share the same work space with humans can react flexibly to unforeseen events, making decisions based on situational awareness. Similar requirements apply to autonomous industrial vehicles.

AI applications in the semiconductor industry showed, for example, a 30% reduction in reject rates when using big data to analyze root-cause data, which must be collected with high precision and in real time. There is the large area of real-time production planning and control of the new “industry 4.0 factories.” Here AI helps optimize the processes at the edge, thereby increasing utilization of machines and, ultimately, the productivity of the entire factory.

Algorithm Implementation Options

With the introduction of high-end vision processors, all image-analysis steps can now optionally be entirely performed in dedicated function equipment. Embedded systems based on digital signal processors (DSPs), application system-on-chips (SoCs), graphics processors (GPUs), field-programmable logic devices (FPGAs), and other processor types are now entering the mainstream, primarily driven by their ability to achieve comparable vision-processing performance at lower cost and power consumption.

Standalone cameras and analytics digital video recorders (DVRs) and networked video recorders (NVRs) increasingly rely on embedded vision processing. Large remote monitoring systems, however, are still fundamentally based on one or more cloud servers that can aggregate and simultaneously analyze numerous video feeds. However, even emerging “cloud” infrastructure systems are beginning to adopt embedded solutions, in order to more easily address performance, power consumption, cost, and other requirements. Embedded vision coprocessors can assist in building scalable systems, offering higher net performance in part by redistributing processing capabilities away from the central server core and toward cameras at the edge of the network.

Although the evolution to an architecture based on distributed intelligence is driving the proliferation of increasingly autonomous networked cameras, complex algorithms often still run on infrastructure servers. Networked cameras are commonly powered by power over Ethernet (PoE) and therefore have a very limited power budget. Further, the lower the power consumption, the smaller and less conspicuous the camera is. To quantify the capabilities of modern semiconductor devices,

consider that an ARM Cortex-A9-based camera consumes only 1.8W in its entirety, while compressing H.264 video at 1080p30 (1920x1080 pixels per frame, 30 frames per second) resolution. Networked cameras with local vision processing “intelligence,” on the other hand, have direct access to raw video data and can analyze and respond to events with low latency. Distributed intelligence surveillance systems block diagram is shown in Figure 9.2.

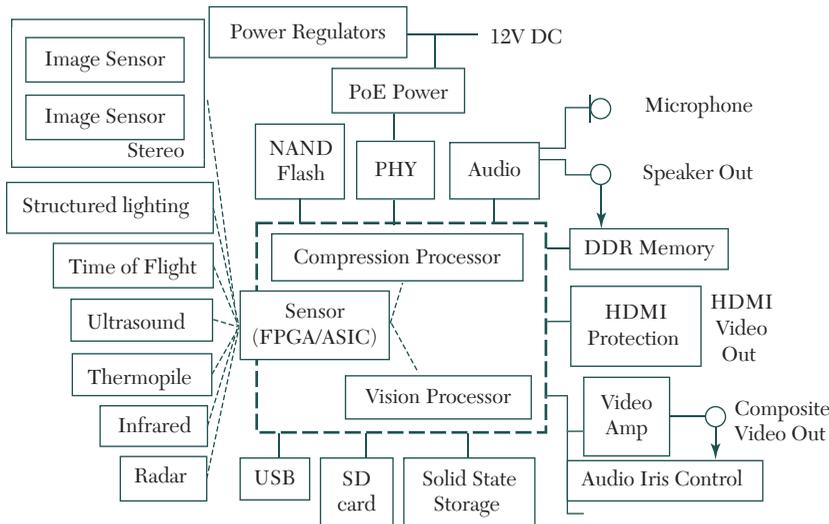


FIGURE 9.2. Distributed intelligence surveillance systems.

It’s relatively easy to recompile PC-originated analytics software to run on an ARM processor, for example. However, as the clock frequency of a host CPU increases, the resultant camera power consumption also increases significantly as compared to running some-to-all of the algorithm on a more efficient DSP, FPGA, or GPU. Interfacing a dedicated vision coprocessor will reduce the power consumption even more. And further assisting software development, a variety of computer vision software libraries is available. Some algorithms, such as those found in OpenCV (the Open Source Computer Vision Library), are cross-platform, while others, such as Texas Instruments IMGLIB (the Image and Video Processing Library), VLIB (the Video Analytics and Vision Library) and VICP (the Video and Imaging Coprocessor Signal Processing Library), are vendor-proprietary. Leveraging pre-existing code speeds time to market, and to the extent that it exploits on-chip vision acceleration resources, it can also produce much

higher performance results than those attainable with generic software. Vision software libraries can speed a surveillance system's time to market (left) as well as notably boost its frame rate and other attributes (right) as shown in Figure 9.3.

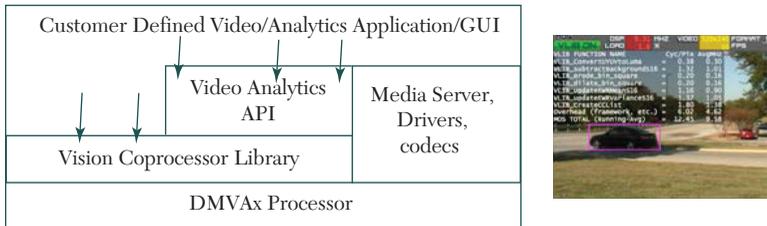


FIGURE 9.3. Vision software libraries (left) its frame rate and other attributes (right).

Initial vision applications such as motion detection sought to draw the attention of on-duty surveillance personnel, or to trigger recording for later forensic analysis. Early in-camera implementations were usually elementary, using simple DSP algorithms to detect gross changes in grayscale video, while those relying on PC servers for processing generally deployed more sophisticated detection and tracking algorithms. Over the years, however, embedded vision applications have substantially narrowed the performance gap with servers, benefiting from more capable function tailored processors. Each processor generation has integrated more potent discrete components, including multiple powerful general computing cores as well as dedicated image and vision accelerators.

As a result of these innovations, the modern portfolio of embedded vision capabilities is constantly expanding. And these expanded capabilities are appearing in an ever wider assortment of cameras, featuring multi-megapixel CMOS sensors with wide dynamic range and/or thermal imagers, and designed for every imaginable installation requirement, including dome, bullet, hidden/concealed, vandal-proof, night vision, pan-tilt-zoom, low light, and wirelessly networked devices. Installing vision-enabled cameras at the “edge” has reduced the need for expensive centralized PCs and backend equipment, lowering the implementation cost sufficient to place these systems in reach of broader market segments, including retail, small business, and residential.

The future is bright for embedded vision systems. Sensors capable of discerning and recovering 3-D depth data, such as stereo vision, ToF (time-of-flight), and structured light technologies, are increasingly appearing

in surveillance applications, promising significantly more reliable and detailed analytics. 3-D techniques can be extremely useful when classifying or modeling detected objects while ignoring shadows and illumination artifacts, addressing a problem that has conventional 2-D vision systems. In fact, systems leveraging 3-D information can deliver detection accuracies above 90%, even for highly complex scenes, while maintaining a minimal false detection rate which is shown in Figure 9.4.

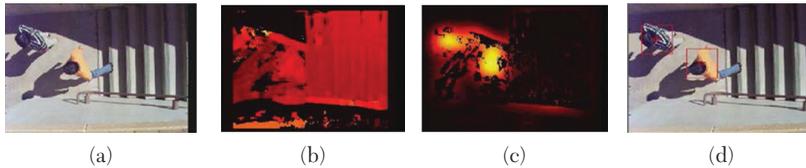


FIGURE 9.4. (a) Left image. (b) Stereo disparity image. (c) 3D-person model. (d) Output 3D-camera accuracy.

However, these 3-D technology advantages come with associated tradeoffs that also must be considered. For example, stereo vision, which uses geometric “triangulation” to estimate scene depth, is a passive, low-power approach to depth recovery that is generally less expensive than other techniques and can be used at longer camera-to-object distances, at the tradeoff of reduced accuracy. Figure 9.5 shows how stereo-vision technique uses a pair of cameras, reminiscent of a human’s left- (left) and right-eye perspectives (middle), to estimate the depths of various objects in a scene (right).



FIGURE 9.5. The stereo-vision technique to estimate the depths of various objects in a scene.

Time-of-flight (ToF), however, is an active, higher-power sensor that generally offers more detail, but at higher cost and with a shorter operating range. Both approaches, along with structured light and other candidates, can be used for detection. But the optimum technology for a particular application can only be fully understood after prototyping. Figure 9.6 shows how although the depth map generated by a ToF 3-D sensor is more dense

than its stereo vision created disparity map counterpart, with virtually no coverage “holes” and therefore greater accuracy in the ToF case, stereo-vision systems tend to be lower power, lower cost, and usable over longer distances. Section 9.3 discusses these technologies in detail.

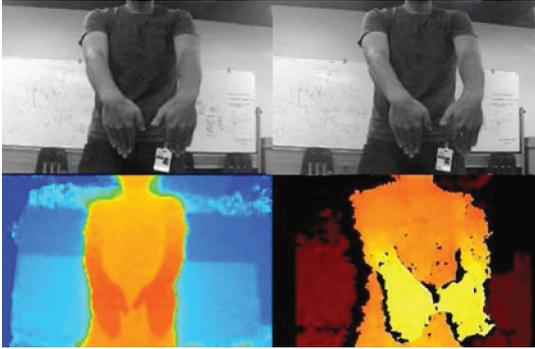


FIGURE 9.6. Depth map generated by ToF (left) 3-D sensor and stereo vision (right) created disparity map.

As new video compression standards such as H.265 become established, embedded-vision surveillance systems will need to process even larger video formats (4k x 2k and beyond), which will compel designers to harness hardware-processor combinations that may include some or all of the following: CPUs, multi-core DSPs, FPGAs, GPUs, and dedicated accelerators. Addressing often contending embedded-system complexity, cost, power, and performance requirements will likely lead to more distributed-vision processing, whereby rich object and feature metadata extracted at the edge can be further processed, modeled, and shared “in the cloud.” And the prospect of more advanced compute engines will enable state-of-the-art vision algorithms, including optical flow and machine learning.

AI Embedded in Cameras

AI is a machine that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. One tool used to create AI is the artificial neural network. Inspired by human or animal biological neural networks, these systems “learn” without task-specific programming. In computer image recognition these commonly called “neural networks” might learn to identify images that contain cats by analyzing sample images that have been manually labeled as “cat” or “no cat” and then using the results to identify cats in other images. They do

this without any prior knowledge about cats, e.g., that they have fur, tails, whiskers and cat-like faces. Instead, they evolve their own set of relevant characteristics from the learning material that they process.

Horizon Robotics offers a camera with sufficient processing power to run its own neural network. In a very real sense, the entire camera now becomes an artificial intelligence. *It is a device capable of learning and making decisions that will enable it to attain its specified objectives.* With the ability to process 1080p video at 30fps it can detect, identify, and track up to 200 objects—faces maybe—per frame. With a typical power consumption of only 1.5W, one now have a powerful server built into each camera. A device able to provide facial recognition or any other analytic you desire, at scale; think finding the bad guy in a stadium full of potential faces.

The advantages that immediately come to mind are the economics of bandwidth and processing power. If the camera is an AI device it will send back to the head end, or to the cloud, only the images or frames that meet the objective—only what the operator is looking for. And, if the processing is done at the edge, no robust server is needed to process and analyze the video stream.

Modern life is one big photo shoot. The glassy eyes of closed-circuit TV cameras watch over streets and stores, while smartphone owners continually surveil themselves and others. Tech companies like Google and Amazon have convinced people to invite ever-watching lenses into their homes via smart speakers and Internet-connected security cameras.

Now a new breed of chips tuned for artificial intelligence is arriving to help cameras around stores, sidewalks, and homes make sense of what they see. Even relatively cheap devices will be able to know your name, what you're holding, or that you've been loitering for exactly 17.5 minutes. It's the latest development in the tech industry's campaign to build out the internet of things, a slogan for linking everyday devices to the internet so they are interactive and gather data.

Smart cameras with built-in AI capabilities can offer new conveniences, like a phone notification that a child just arrived home safely, or that the dog walker really did walk the dog. They will also bring new risks to privacy in public and private spaces.

9.2 ARTIFICIAL VISION

Artificial vision is the use of special technology to give some vision to people without natural sight. Some artificial vision devices target the *retina*, the tissue lining the inside of the back of the eye. Light-sensitive cells in the retina, called *rods* and *cones*, absorb light rays and change them into electrical signals. The signals are sent to the brain, which interprets them as visual images. Several diseases can damage the retina, causing *retinal degeneration* that eventually leads to blindness. In some cases, disease causes the cones and rods to die but leaves undamaged other cells in the retina as well as the *optic nerve*, the retina's connection to the brain. Patients in this condition may benefit from artificial vision systems implanted directly into the eye to restore sight.

One form of artificial vision, called a *retinal prosthesis*, uses small electrodes to stimulate certain cells in the retina. A surgeon places the electrodes in the back of the eye, in front of or behind the retina. In many designs an external camera, usually mounted in a pair of eyeglasses, captures visual imagery and relays it to a small computer. The computer sends radio or infrared signals to the implanted electrodes, which convert them to electrical signals. The stimulated retina cells send these signals to the brain, where they are interpreted as vision. Other designs do not require an external camera, and instead try to use the retinal implant as a more direct replacement for damaged rods and cones.

The image created by a retinal prosthesis appears as a pattern of lighted dots that are large and fuzzy. The retinal prosthesis cannot create a detailed picture, but it may help a blind person, for example, to find his or her way around obstacles and to locate objects at close range. In patients with damaged optic nerves, signals cannot get from the eye to the portion of the brain responsible for vision. In these patients, physicians can implant electrodes directly in the brain to receive signals. In combination with an external camera and a computer, the electrodes stimulate brain cells to produce a crude image. It is shown in Figure 9.7.

A digital camera mounted on the patient's eyeglasses or head takes images on demand, for example, at the push of a button or a voice command. By using near real-time image-processing algorithms, parameters such as the brightness (i.e., bright, medium, dark), color (according to a predefined color palette), and content of the captured image frame are determined. The content of an image frame is determined as follows: First,

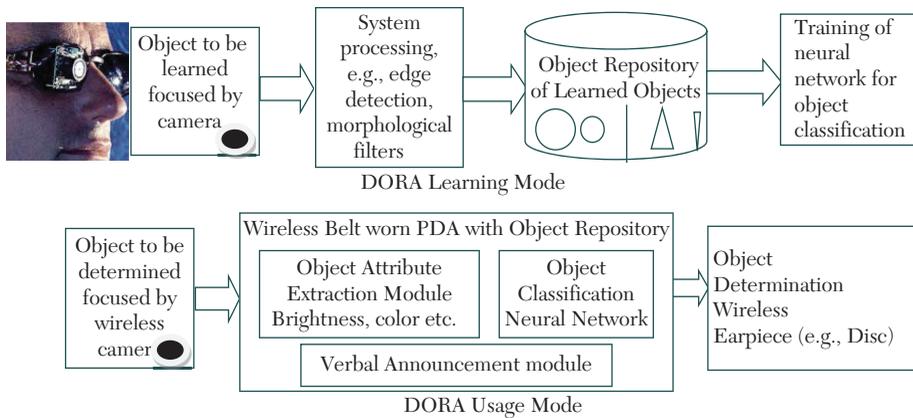


FIGURE 9.7. Digital object-recognition audio (DORA)-assistant for the visually impaired.

the captured image frame is processed for edge detection within a central region of the image only, to avoid disturbing effects along the borderline. The resulting edge pattern is subsequently classified by artificial neural networks that have been trained previously on a list of “known,” that is, identifiable, objects such as table, chair, door, car, and so on. Following the image processing, a descriptive sentence is constructed consisting of the determined/classified object and its descriptive attributes (brightness, color, etc.): *“This is a dark blue chair.”* Using a computer-based voice synthesizer, this descriptive sentence is then announced verbally to the severely visually impaired or blind patient.

Severely visually impaired patients, blind patients, or blind patients with retinal implants alike can benefit from a system such as the object-recognition assistant presented here. With the basic infrastructure (image capture, image analysis, and verbal image content announcement) in place, this system can be expanded to include attributes such as object size and distance, or, by means of an IR-sensitive camera, to provide basic “sight” in poor visibility (e.g., foggy weather) or even at night.

AI for Industries

Use of AI in industrial environments requires highly advanced integrated logic. This is because, in fast processes as in the case of inspection systems, there is often no further control authority. This is different, for example, in medical technology, where AI results of automatic image analysis are always controlled by a doctor and where AI’s job is simply to make recommendations, accelerating data evaluation steps in the process.

For this reason, AI systems for use in industrial environments must always ensure that AI decision-making processes are traceable and as regards machine and occupational safety requirements, 100 percent correct. Training the AI is therefore much more complex in an industrial environment. There are also hardly any negative examples. This aspect, too, is different from medical technology, where thousands of negative and positive diagnoses can be used to train and teach systems. In industry, on the other hand, errors must be avoided right from the start. This is why digital twins, that is, digital images of machines and systems, are often used here to simulate negative findings and then, for instance, rule out certain robot movements from the outset.

Machine learning is the system where knowledge is constantly expanded with new clear information and deep learning is a system where systems train themselves using large amounts of data and independently interpret new information. Many computing units, usually general purpose graphics units (GPGPUs) are combined to form a deep neural network (DNN). This deep learning network must then be trained. In the field of image processing, for example, pictures of various trees can be used to train the system. The amount of image data required is immense. Real-life research projects speak of 130,000 to 700,000 images. Using all this information, neural networks then develop parameters and routines based on case specific algorithms to reliably identify a tree.

With significantly increased computing and graphics performance, the ultra-low power and industrially robust AMD Ryzen Embedded V1000 series is a good choice. With a total of 3.6 TFLOPs from a multipurpose CPU and powerful GPGPU combined, it offers flexible computing power that until a few years ago was only achievable with systems that consumed several hundred watts. Today this computing power is available from 15 watts. This makes the processors also suitable for integration in fanless, completely enclosed and, hence, highly robust devices for factory use. As real-time processes, these also support memory with error correction code (ECC), which is essential for most industrial machines and systems.

As far as necessary software environment for fast and effective introduction of AI and deep learning is concerned, AMD-embedded processors offer comprehensive support for tools and frameworks, such as TensorFlow, Caffe and Keras. At <https://gpuopen.com/professional-compute/>, a wide range of software tools and programming environments for deep learning and AI applications are available. These include popular open source platform

ROCm (Radeon Open Compute platform) for GPGPU applications. HIPfy is an available tool with which proprietary applications can be transferred into portable HIP C++ applications, so that dangerous dependence on individual GPU manufacturers can be effectively avoided. AI development has also become much easier with the availability of OpenCL2.2 because since then. OpenCL C++ kernel language has been integrated into OpenCL, which makes writing parallel programs much easier. With such an ecosystem, both knowledge-based AI and deep learning are comparatively easy to implement, and are no longer just the reserve of billion-dollar IT giants like Google, Apple, Microsoft, and Facebook.

Big Data is the name given to huge volumes of data that can be analyzed to get useful patterns and make business predictions. The IoT Big Data is found in many industrial applications including predictive models, analysis, smart devices, and machine learning. AI is required to make sense of Big Data that is generated from various sensors. Hence, it becomes a prerequisite for an IoT system to work intelligently for analyzing, collecting, manipulating, and generating insights, from the enormous amount of data at high speed.

Edge computing is important in machine learning. It makes decisions by applying intelligence based on the deployed machine-learning models. For example, IoT devices are deployed for monitoring gas pipelines. Here, costly human inspection is replaced by technology that can rapidly detect leakages or other anomalies, and automatically alert the concerned person(s) for appropriate action. In such cases, distributed computing at the edge can dramatically cut response times.

Edge-computing architecture can be visualized using three-tier architecture. The first layer has local devices and applications, the second is the edge layer and the third is a public cloud. Devices are sensors and actuators that are responsible for collecting data and controlling devices. These are connected to the cloud through the local edge-computing layer.

9.3 3D-IMAGING TECHNOLOGIES: STEREO VISION, STRUCTURED LIGHT, LASER TRIANGULATION, AND TOF

1. Stereo Vision

Stereo vision is the process of extracting 3-D information from multiple 2-D views of a scene. It is a technique aimed at inferring depth from two or

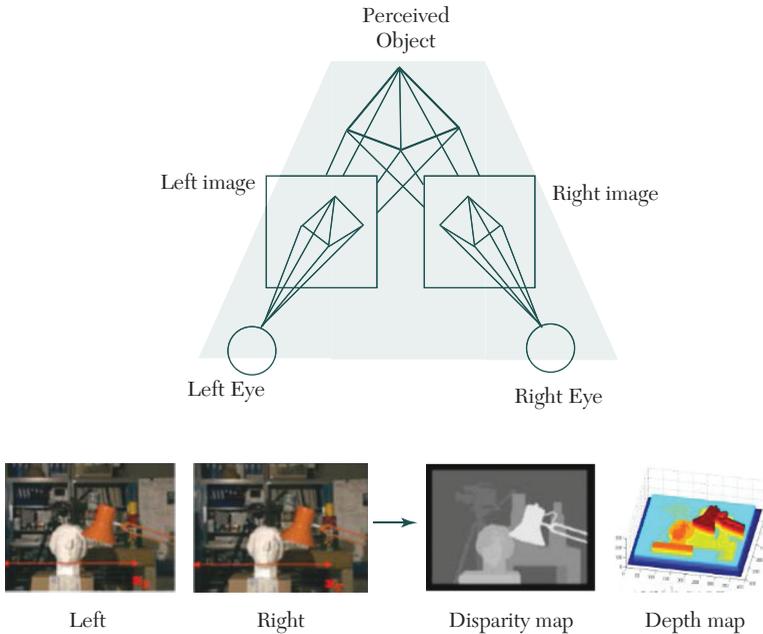


FIGURE 9.8. Stereo vision.

more cameras. It is shown in Figure 9.8. Stereo vision is used in applications such as advanced driver-assistance systems (ADAS) and robot navigation where stereo vision is used to estimate the actual distance or range of objects of interest from the camera. The 3-D information can be obtained from a pair of images, also known as a stereo pair, by estimating the relative depth of points in the scene. These estimates are represented in a stereo disparity map, which is constructed by matching corresponding points in the stereo pair. Stereo vision is also used in applications such as 3-D movie recording and production, object tracking, machine vision, and range sensing.

The recovery of the 3D structure of a scene using two or more images of the 3D scene, each acquired from a different viewpoint in space. The images can be obtained using multiple cameras or one moving camera. The term binocular vision is used when two cameras are employed. The calibration techniques used for this specific 3D imaging technique involve the alignment of the pixel information between the cameras, as well as the extraction of the necessary information on the depth of the image. These calibration techniques are performed in a similar manner to how our brains visually measure distance. The transposition of the cognitive process into a system therefore requires a significant computational effort by the imaging system.

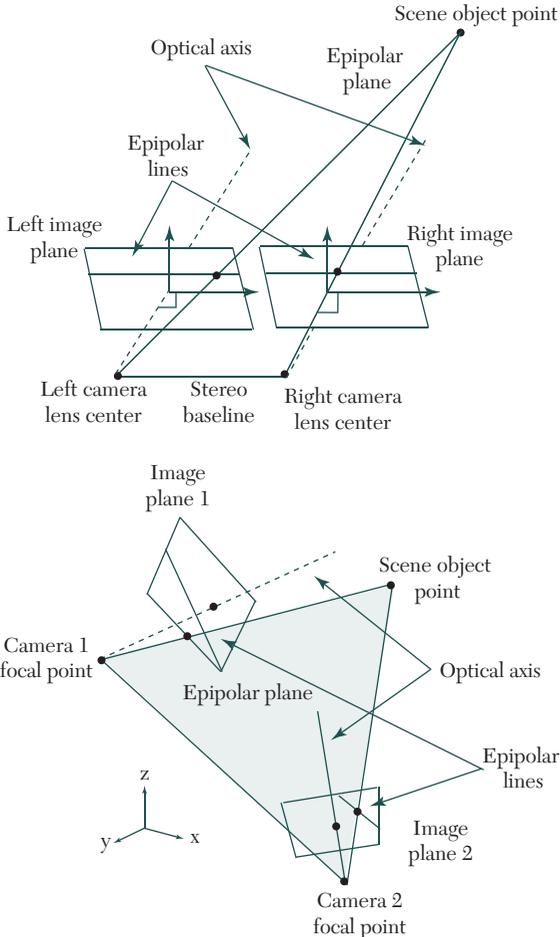


FIGURE 9.9. Two cameras in arbitrary position and orientation.

The incorporation of standard image sensors into stereo vision reduce the overall costs of these cameras, as compared to when more sophisticated sensors are used, such as a high-performance sensor or global shutter, which typically result in a higher cost of the complete system. However, it is important to note that the distance range is limited by mechanical constraints, as the requirement to achieve a physical baseline will require the use of larger dimension modules. The precise mechanical alignment and recalibration of these systems are also important in ensuring their accurate measurements. One limitation of the stereo vision technique is

that it often does not work well in changing light conditions, as it is heavily dependent upon the object's reflective characteristics.

Fixation point is the point of intersection of the optical axis. *Baseline* is the distance between the centers of projection. *Epipolar plane* is the plane passing through the centers of projection and the point in the scene. *Epipolar line* is the intersection of the epipolar plane with the image plane. *Conjugate pair* is any point in the scene that is visible in both cameras and will be projected to a pair of image points in the two images. Two camera position and terminology are shown in Figure 9.9.

Triangulation—the principle underlying stereo vision

The 3D location of any visible object point in space is restricted to the straight line that passes through the center of projection and the projection of the object point. Binocular stereo vision determines the position of a point in space by finding the intersection of the two lines passing through the center of projection and the projection of the point in the each image. It is explained in Figure 9.10.

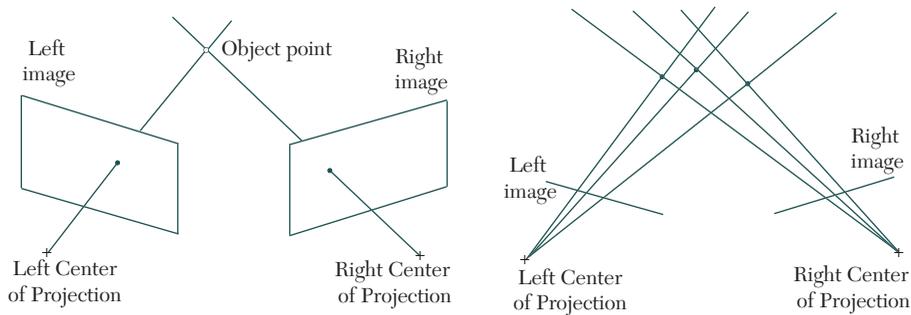


FIGURE 9.10. Binocular stereo vision.

The two problems of stereo are the correspondence problem and reconstruction problem. The correspondence problem is finding pairs of matched points such that each point in the pair is the projection of the same 3D point. Triangulation depends crucially on the solution of the correspondence problem. Ambiguous correspondence between points in the two images may lead to several different consistent interpretations of the scene. The reconstruction problem can be solved by disparity map. Given the corresponding points, one can compute the disparity map. The disparity map can be converted to a 3D map of the scene (i.e., it can recover the 3D structure) if the stereo geometry is known.

Recovering depth (reconstruction)

Stereo images are rectified to simplify matching, so that a corresponding point in one image can be found in the same row in the other image. This reduces the 2D-stereo correspondence problem to a 1D problem. There are two approaches to stereo image rectification, calibrated, and uncalibrated rectification. Uncalibrated stereo image rectification is achieved by determining a set of matched interest points, estimating the fundamental matrix, and then deriving two projective transformations. Calibrated stereo rectification uses information from the stereo camera calibration process. Calibration is carried out acquiring and processing 10 + stereo pairs of a known pattern such as typically a checkerboard. Calibration is available in both OpenCV and MATLAB software. Figure 9.11 gives the overview of stereo vision system.

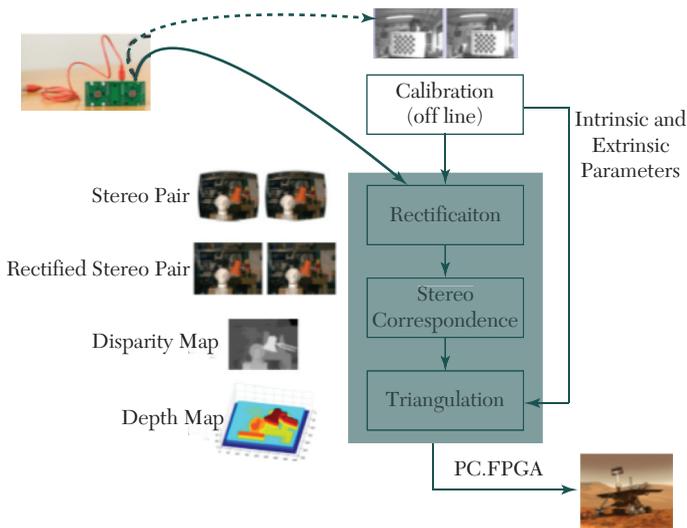


FIGURE 9.11. Overview of a stereo vision system.

Stereo camera calibration is used to determine the intrinsic parameters and relative location of cameras in a stereo pair, this information is used for stereo rectification and 3D reconstruction. The rectification process uses the information from the calibration step. The rectification process removes lens distortions and turns the stereo pair in standard form. Stereo correspondence aims at finding homologous points in the stereo pair. The next to last step is triangulation. This process give the disparity map, the base line T and the focal length f (calibration). Triangulation computes the position of the correspondence in the 3D. Figure 9.12 explains this.

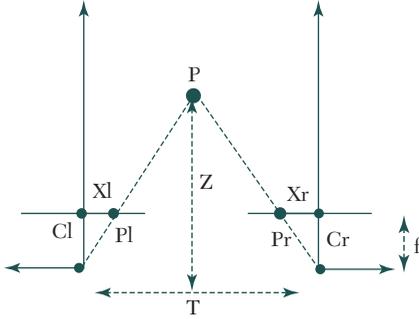


FIGURE 9.12. Triangulation process.

Consider recovering the position of P from its projections Pl and Pr . Given with base line T and focal length f .

$$x_l = f \frac{X_l}{Z_l} \quad \text{or} \quad X_l = \frac{x_l Z_l}{f} \quad \text{and} \quad x_r = f \frac{X_r}{Z_r} \quad \text{or} \quad X_r = \frac{x_r Z_r}{f}$$

In general, the two cameras are related by the following transformation:

$$P_r = R(P_l - T)$$

Using $Z_r = Z_l = Z$ and $X_r = X_l - T$ we have:

$$\frac{x_l Z}{f} - T = \frac{x_r Z}{f} \quad \text{or} \quad Z = \frac{Tf}{d}$$

Where $d = x_l - x_r$ is the *disparity* (i.e., the difference in the position between the corresponding points in two images)

Intrinsic parameters of the two cameras are focal length, image center, parameters of lenses distortion, and so on. They characterize the transformation from image plane coordinates to pixel coordinates, in each camera. Extrinsic parameters (R , T) describe the relative position and orientation of the two cameras.

$P_r = R(P_l - T)$ (R and T aligns right camera with left camera). They can be determined from the extrinsic parameters of each camera:

$$R = R_r R_l$$

$$T = T_l - R^T T_r$$

Disparity is the distance between corresponding points when the two images are superimposed. Disparity is higher for points closer to the

camera. The disparity map is the distance of all points from the disparity map that can be displayed as an image.

2. Structured Light

In the structured-light technique as shown in Figure 9.13, a predetermined light pattern is projected onto an object to obtain an image's depth information through the analysis of the distorted pattern. This imaging technique prevents motion blur from occurring, as there is no conceptual limit on frame times, which thereby provides this robust technique protection from multipath interfaces. Note that active illumination requires a complex camera, in which precise and stable mechanical alignment between the lens and pattern projector are required, however there is a risk of decalibration that can occur in these situations. Additionally, the reflected pattern is sensitive to optical interference in the environment and must therefore only be utilized for use in indoor applications.

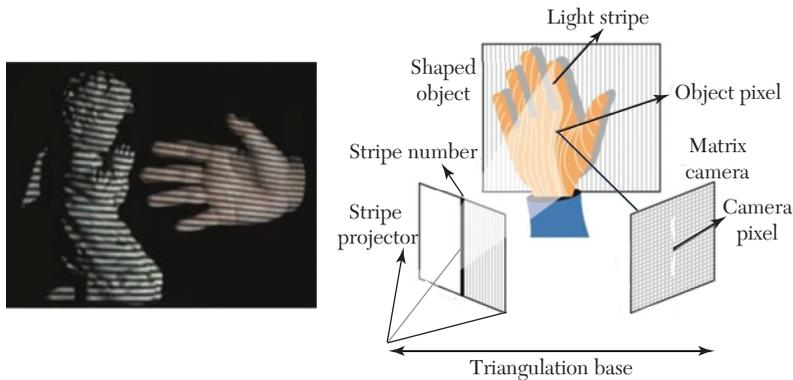


FIGURE 9.13. Structured light.

3. Laser Triangulation

Laser triangulation systems measure the geometrical offset of a line of light, whose value is directly related to the height of the object. Based on the ability of the camera to scan the object, this one-dimensional imaging technique is entirely dependent on the determining the distance between the laser and its point on the surface of the object, which will provide information on the position of the laser dot as it appears in the camera's field of view. The term triangulation indicates that the laser dot, the camera, and the laser emitter form a triangle. Laser triangulation is shown in Figure 9.14.

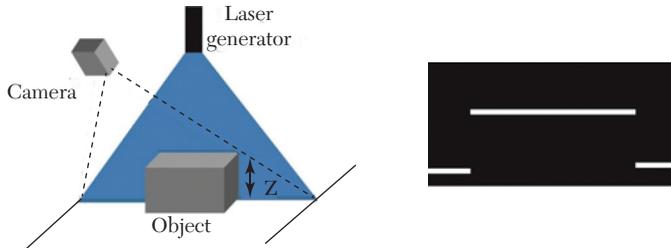


FIGURE 9.14. Laser triangulation.

High-resolution lasers are typically used for monitoring applications where high accuracy, stability, and low temperature drift are required to investigate the displacement and position of objects. This technique is limited to scanning applications only, as it is only capable of covering only a short range, and is sensitive to ambient light, as well as structured and/or complex structures. Complex algorithms and calibration are also required for this technique.

4. Time-of-Flight Camera for 3D imaging

3D Time-of-Flight (ToF) technology is revolutionizing the machine vision industry by providing 3D imaging using a low-cost CMOS pixel array together with an active modulated light source. Compact construction, easy-of-use, together with high accuracy and frame-rate makes ToF cameras an attractive solution for a wide range of applications. Time-of-flight (ToF) is a relatively new method by which it is possible to gather detailed 3D information. By illuminating a scene, usually using infrared or near-infrared light, a ToF camera can measure the distance between itself and objects within that scene. Compared to other techniques for acquiring 3D information, such as use of scanning or stereoscopic vision, ToF cameras are capable of exhibiting greater accuracy, while being extremely fast and affordable. This opens up the advantages of 3D imaging to a much greater variety of applications than was feasible previously, covering gaming, medicine, manufacturing, and so on.

ToF is a term used to denote each of the methods that measure the distance from a direct calculation of the double time flight of photons that are present between the camera and the scene. This measurement is performed either directly (D-ToF) or indirectly (I-ToF). Whereas D-ToF requires a complex and constraining time-resolved apparatus, I-ToF is simpler, as it's a light source is synchronized with an image sensor.

The pulse of light is then emitted from the sensor in phases with the shuttering of the camera, during which the synchronization of the light pulse is used to calculate the ToF of the photons to determine the distance between the point of emission and the object. During this procedure, a direct measurement of the depth and amplitude of every measurable pixel is used to create the final image, which is otherwise referred to as the depth map. The ToF operating technique is shown in Figure 9.15 (a).

The ToF system has a small aspect ratio and monocular approach that only requires calibration once in the entire lifetime of the device. Additionally, these properties of the ToF system allow for its consistent successful operation in ambient light conditions. Despite these advantages, the ToF system requires active illumination synchronization, which can ultimately lead to multipath interference and distance aliasing of the depth map.

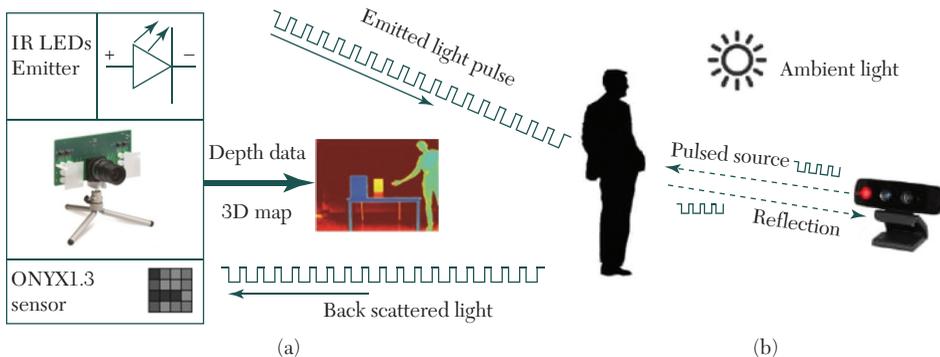


FIGURE 9.15. (a) ToF operating principle (b) 3D time-of-flight camera operation.

Theory of Operation

A 3D time-of-flight (ToF) camera works by illuminating the scene with a modulated light source and observing the reflected light. The phase shift between the illumination and the reflection is measured and translated to distance. Figure 9.15 (b) illustrates the basic ToF concept. Typically, the illumination is from a solid-state laser or a LED operating in the near-infrared range (~850nm) invisible to the human eyes. An imaging sensor designed to respond to the same spectrum receives the light and converts the photonic energy to electrical current. Note that the light entering the sensor has an ambient component and a reflected component. Distance (depth) information is only embedded in the reflected component. Therefore, high ambient component reduces the signal to noise ratio (SNR). Various types

of signals (also called carriers) are used with ToF, with sound and light being the most common.

Time of flight light carrier

Using light sensors as a carrier is common, because it is able to combine speed, range, low weight, and eye-safety. Infrared light ensures less signal disturbance and easier distinction from natural ambient light, resulting in very high-performing sensors for their given size and weight.

Time of flight sound carrier

Ultrasonic sensors are used for determining the proximity of objects (reflectors) while the robot or UAV is navigating. For this task, the most conventional implementation is the time-of-flight sensor, which calculates the distance of the nearest reflector using the speed of sound in air and the emitted pulse and echo arrival times.

To detect phase shifts between the illumination and the reflection, the light source is pulsed or modulated by a continuous-wave (CW), source, typically a sinusoid or square wave. Square wave modulation is more common because it can be easily realized using digital circuits. Pulsed modulation can be achieved by integrating photoelectrons from the reflected light, or by starting a fast counter at the first detection of the reflection. The latter requires a fast photo-detector, usually a single-photon avalanche diode (SPAD). This counting approach necessitates fast electronics, since achieving 1 millimeter accuracy requires timing a pulse of 6.6 picoseconds in duration. This level of accuracy is nearly impossible to achieve in silicon at room temperature.

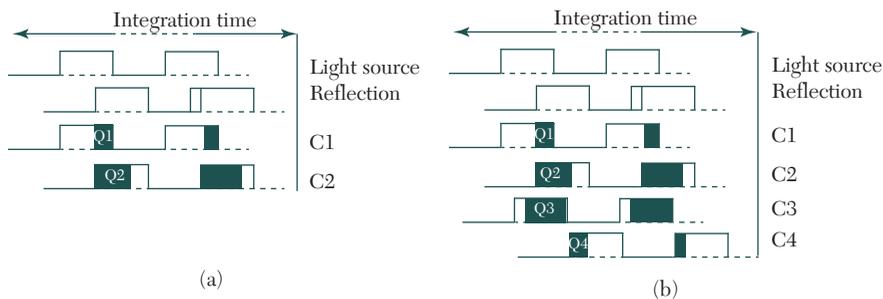


FIGURE 9.16. (a) Two time-of-flight methods: pulsed. (b) ToF method: continuous-wave.

The pulsed method is straightforward (Figure 9.16 a). The light source illuminates velocity “ c ” light for a brief period t , and the reflected energy is sampled at every pixel, in parallel, using two out-of-phase windows, C_1 and C_2 , with the same time period t . Electrical charges accumulated during these samples, Q_1 and Q_2 , are measured and used to compute distance using the formula:

$$d = \frac{1}{2} c \Delta t \left(\frac{Q_2}{Q_1 + Q_2} \right) \quad (1)$$

In contrast, the CW method (Figure 9.16 b) takes multiple samples per measurement, with each sample phase-stepped by 90 degrees, for a total of four samples. Using this technique, the phase angle between illumination and reflection φ and the distance, d , can be calculated by

$$\varphi = \arctan \left(\frac{Q_3 - Q_4}{Q_1 - Q_2} \right) \quad (2)$$

$$d = \frac{c}{4\pi f} \varphi \quad (3)$$

In ToF sensors, distance is measured for every pixel in a 2D addressable array, resulting in a depth map. A depth map is a collection of 3D points (each point also known as a voxel). As an example, a QVGA sensor will have a depth map of 320 x 240 voxels. 2D representation of a depth map is a grayscale image, as is illustrated by the soda cans example in Figure 9.17 the brighter the intensity, the closer the voxel. Figure 9.17 (a) shows the depth map of a group of soda cans.

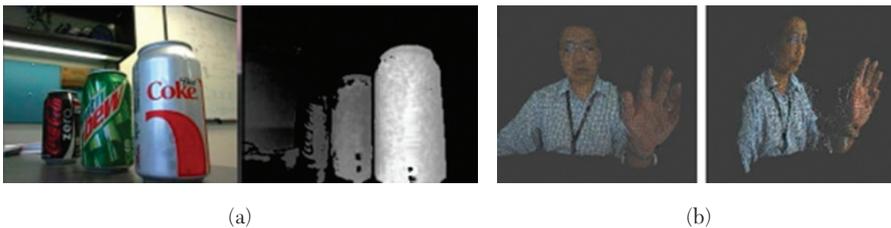


FIGURE 9.17. (a): Depth map of soda cans. (b) Avatar formed from point-cloud.

Alternatively, a depth map can be rendered in a three-dimensional space as a collection of points, or point-cloud. The 3D points can be mathematically connected to form a mesh onto which a texture surface can be mapped. If the texture is from a real-time color image of the same

subject, a life-like 3D rendering of the subject will emerge, as is illustrated by the avatar in Figure 9.17 (b). One may be able to rotate the avatar to view different perspectives.

Working of ToF

ToF camera systems consist of an image sensor, image processing chip, and modulated light source. In simple terms, these systems work by illuminating a scene with a modulated light source, and then measure the phase shift of the wave that is reflected back. Since light has a constant speed, ToF cameras are able to calculate the distance to each point in the scene based on the time it took for that light to return to the camera. Rather than scanning an image line by line, a ToF camera system will illuminate the entire scene all at once and then measures phase shift in the light reflected back to the image sensor. This raw data can be captured quickly and the calculation required to derive the distance is relatively straightforward, with ToF cameras thus achieving extremely high frame rates (even beyond what human vision can detect). This means that unlike many other 3D-vision arrangements, ToF allows 3D-depth information to be extracted from a scene in real-time using embedded processors. Figure 9.18 shows a ToF imaging system using Melexis hardware.

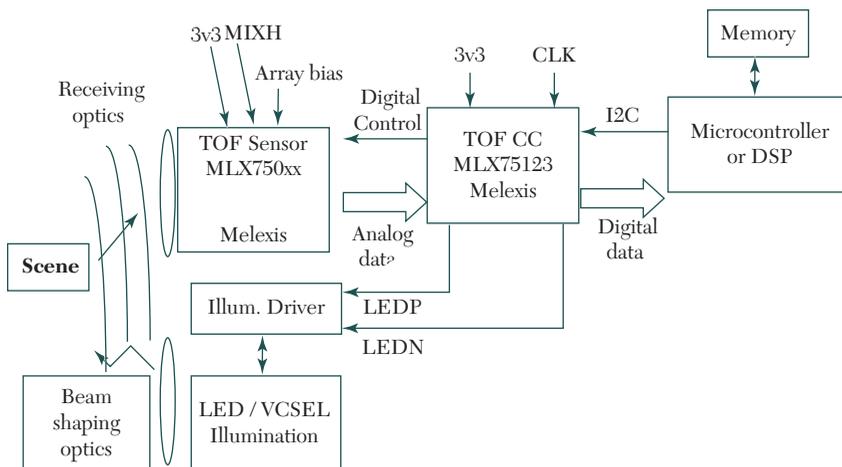


FIGURE 9.18. A ToF imaging system using Melexis hardware.

While it is possible to source the components needed for a ToF camera system individually, several manufacturers supply compact, ready-made solutions that are generally more convenient. Targeted at the automotive

sector, Melexis' solution comprises the MLX75x23 image sensor paired with the MLX75123 companion chip. The MLX75x23 is a sunlight-robust image sensor with QVGA resolution, while the MLX75123 controls the sensor, modulates the light source and communicates with the host processor. Besides Melexis, STMicroelectronics VL6180 provides a compact, integrated ToF solution. This is aimed at smartphone designs and enables gesture recognition functionality to be benefitted from. The OPT8241-CDK-EVM evaluation hardware from Texas Instruments is based on the company's OPT8241 320×240 resolution ToF imaging device, which supports up to 150fps operation.

Comparison of 3D-imaging Technologies

Stereo vision versus ToF

Stereo vision generally uses two cameras separated by a distance, in a physical arrangement similar to the human eyes. Given a point-like object in space, the camera separation will lead to measurable disparity of the object positions in the two camera images. Using a simple pin-hole camera model, the object position in each image can be computed, which we will represent them by α and β . With these angles, the depth, z , can be computed as shown in Figure 9.19(a).

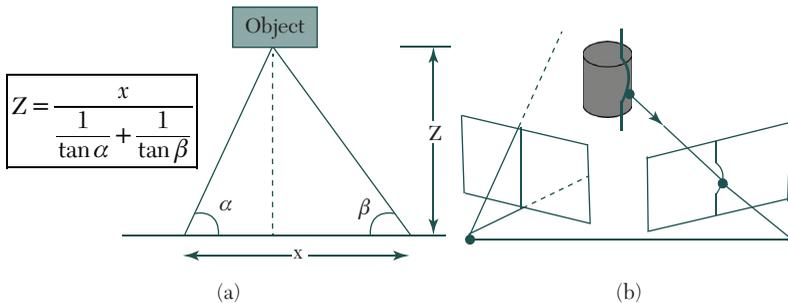


FIGURE 9.19. (a) Stereopsis depth through disparity measurement. (b) Structured light concept.

A major challenge in stereo vision is solving the correspondence problem: giving a point in one image, how to find the same point in the other camera? Until the correspondence can be established, disparity, and therefore depth, cannot be accurately determined. Solving the correspondence problem involves complex, computationally intensive algorithms for feature extraction and matching. Feature extraction and matching also require sufficient intensity and color variation in the image

for robust correlation. This requirement renders stereo vision less effective if the subject lacks these variations—for example, measuring the distance to a uniformly colored wall. ToF sensing does not have this limitation because it does not depend on color or texture to measure the distance.

In stereo vision, the depth resolution error is a quadratic function of the distance. By comparison, a ToF sensor, which works off reflected light, is also sensitive to distance. However, the difference is that for ToF this shortcoming is remedied by increasing the illumination energy when necessary; and the intensity information is used by ToF as a “confidence” metric to maximize accuracy using Kalman filter-like techniques.

Stereo vision has some advantages. The implementation cost is very low, as most common off-the-shelf cameras can be used. Also, the human-like physical configuration makes stereo vision well-suited for capturing images for intuitive presentation to humans, so that both humans and machines are looking at the same images.

Structured-light versus ToF

Structured-light works by projecting known patterns onto the subject and inspecting the pattern distortion. Successive projections of coded or phase-shifted patterns are often required to extract a single-depth frame, which leads to a lower frame rate. Low-frame rate means the subject must remain relatively still during the projection sequence to avoid blurring. The reflected pattern is sensitive to optical interference from the environment; therefore, structured-light tends to be better suited for indoor applications. A major advantage of structured-light is that it can achieve relatively high spatial (X-Y) resolution by using off-the-shelf DLP projectors and HD color cameras. Figure 9.18 (b) shows the structured-light concept. By comparison, ToF is less sensitive to mechanical alignment and environmental lighting conditions, and is more mechanically compact. The current ToF technology has lower resolution than today’s structured-light, but is rapidly improving.

The comparison of a ToF camera with stereo vision and structured-light is summarized in Table 9.1. The key takeaway is that ToF is a cost-effective, mechanically compact depth-imaging solution unaffected by varying environmental illumination and vastly simplifies the Figure-ground separation commonly required in scene understanding. This powerful combination makes the ToF sensor well-suited for a wide variety of applications.

TABLE 9.1. Comparison of 3D-Imaging Technologies

Considerations	Stereo Vision	Structured Light	Laser Triangulation	Time-of-Flight (ToF)
Software complexity	High	Medium	High	Low
Material cost	Low	High	High	Low
Compactness	Medium	Medium	Medium	Very compact
Response time	Medium	Slow	Medium	Fast
Depth accuracy	Medium	Medium to very high in short range	Very high	Medium
Low light performance	Weak	Good	Good	Good
Bright light performance	Good	Weak	Medium	Good
Power consumption	Low	Medium	Medium	Scalable
Range	Limited 2m to 5m	Scalable cm to 2m	Limited cms	Scalable 30-50 cm to 20-50 m
Distance	Medium to far (depending on the distance of the cameras)	Short to medium	Short	Far
Resolution	Medium	Medium	Varies	High
Real-time capability	Low	Low	Low	High
Outdoor light	Good	Weak	Weak	Weak to good
Total operating cost (including calibration efforts)	High	Medium to high	High	Medium
APPLICATIONS				
Game		X	X	X
3D movies	X			

Considerations	Stereo Vision	Structured Light	Laser Triangulation	Time-of-Flight (ToF)
3D scanning		X		X
User interface control			X	X
Augmented reality	X			X

Although few in number, many of the currently used 3D systems are based on 3D stereo vision, structured-light cameras or laser triangulation, all of which typically operate at fixed working distances that require significant calibration to achieve specific areas of detection. The TOF systems are therefore particularly advantageous as compared to these other 3D-imaging techniques as a result of their unique ability to overcome these limiting challenges. In doing so, TOF systems often provide users with a greater amount of flexibility in its use in potential applications. Due to the pixel complexity and/or power consumption of most commercial solutions, image resolution is often limited to video graphics array (VGA) or less.

Applications of ToF 3D-Imaging Technology

ToF technology can be applied to applications from automotive to industrial to healthcare, to smart advertising, gaming, and entertainment. A ToF sensor can also serve as an excellent input device for both stationary and portable computing devices. In automotive, ToF sensors can enable autonomous driving and increased surrounding awareness for safety. In the industrial segment, ToF sensors can be used as human machine interface (HMI), and for enforcing safety envelopes in automation cells where humans and robots may need to work in close proximity. In smart advertising, using ToF sensors as gesture input and human recognition, digital signage could become highly interactive, targeting media contents to the specific live audience. In healthcare, gesture recognition offers noncontact human-machine interactions, fostering a more sanitary operating environment. The gesturing capability is particularly well-suited for consumer electronics, particularly in gaming, portable computing, and home entertainment. ToF sensors natural interface provides an intuitive gaming interface for first-person video games. This same interface can also replace remote controls, mice, and touchscreens. These ToF camera sensors can be used for object scanning, measuring distance, indoor navigation, obstacle avoidance, gesture recognition, tracking objects, measuring volumes,

reactive altimeters, 3D photography, augmented reality games, and much more. Generally speaking, ToF applications can be categorized into gesture and non-gesture. Gesture applications emphasize human interactions and speed, while non-gesture applications emphasize measurement accuracy.

Gesture Applications

Gesture applications translate human movements (faces, hands, fingers, or whole-body) into symbolic directives to command gaming consoles, smart televisions, or portable computing devices. For example, channel surfing can be done by the waving of hands, and presentation can be scrolled by using finger flickering. These applications usually require fast response time, low- to medium-range, centimeter-level accuracy, and power consumption.

Non-Gesture Applications

ToF sensors can be used in non-gesture applications as well. For instance, in automotive, a ToF camera can increase safety by alerting the driver when it detects people and objects in the vicinity of the car, and in computer-assisted driving. In robotics and automation, ToF sensors can help detect product defects and enforce safety envelopes required for humans and robots to work in close proximity. With 3D printing rapidly becoming popular and affordable, ToF cameras can be used to perform 3D scanning to enable “3D copier” capability. In all of these applications, spatial accuracy is important.

ToF camera technology is allowing machines to see beyond simple 2D images and explore the third dimension, thereby enabling depth perception and better object recognition. Compared to other 3D machine-vision techniques, ToF is much faster, and its ability to generate real-time depth information means that a wide variety of applications can be served. These include the following:

Augmented Reality—ToF-enabled 3D vision is allowing exciting new augmented reality (AR) applications to be explored, and existing ones to work better. The point clouds generated by a TOF camera enable AR software to map out its surroundings for an enhanced 3D understanding of the environment around it. This lets it place in-software objects more accurately and facilitates more dynamic interaction between virtual and actual elements of the environment. TOF can also detect the user’s movements and posture, so that they are able to interact with virtual

elements using their body directly, without having to rely on handheld controllers or gloves.

Industrial Robots—For the industrial segment, the capacity to recognize objects and produce real-time 3D depth maps will prove invaluable to robotics. Manufacturing robots involved in automated quality inspection will be able to quickly and accurately produce a 3D scan of an object. ToF may also be used in collaborative robot designs, to prevent collisions with humans nearby or to provide interactive gesture control. For logistics, it will allow robots to grab and place objects more accurately.

Medical, Scientific, Engineering—In the medical field there can often be a need to interface with electronics, but the risk of cross-contamination means that touch-based interaction is undesirable. Gesture-based control using ToF cameras will allow doctors and nurses to manipulate images or utilize software without having physical contact with the device. For scientific investigation, ToF cameras would enable gesture-based manipulation of 3D images—such as DNA strands or protein molecules. In engineering, being able to quickly and affordably 3D scan items will be helpful for hardware prototyping and design activities.

Drones and Vehicles—ToF cameras could also bring greater intelligence to drones and unmanned ground vehicles. Drones using ToF would have a better awareness of their 3D environment and be able to create 3D maps or perform automated obstacle avoidance. Similarly, unmanned ground vehicles could use ToF cameras to provide obstacle sensing capabilities, allowing for autonomous navigation.

Time of Flight Sensor Advantages

As an emerging technology, ToF has a number of advantages over conventional point (single pixel) scanner cameras and stereoscopic cameras, including:

Simplicity—In contrast to stereo vision or triangulation systems, the whole system is very compact: the illumination is placed just next to the lens, whereas the other systems need a certain minimum base line. In contrast to laser scanning systems, no mechanical moving parts are needed. A great advantage of a ToF cameras is that they are capable of composing 3D images of a scene in just one shot. Many other 3D vision systems require more images and movement.

Efficient—It is a direct process to extract the distance information out of the output signals of the ToF sensor. As a result, this task uses only a small amount of processing power, whereas with stereo vision complex-correlation algorithms are implemented requiring much more processing power using more energy. After the distance data has been extracted, object detection is also a straightforward process to carry out because the algorithms are not disturbed by patterns on the object.

Speed—ToF 3D cameras are able to measure the distances within a complete scene with a single shot. As the cameras reach up to 160 frames per second, they are ideally suited to be used in real-time applications.

Price—In comparison to other 3D depth-range scanning technology such as structured-light camera/projector systems or laser-range finders, ToF technology is quite inexpensive.

Other advantages are lightweight, full-frame time-of-flight data (3D image) collected with a single laser pulse, unambiguous direct calculation of range, blur-free images without motion distortion, coregistration of range and intensity for each pixel, pixels are perfectly registered within a frame, have the ability to represent objects in the scene that are oblique to the camera, there's no need for precision scanning mechanisms, they combine 3D flash LIDAR with 2D cameras for 2D texture over 3D depth, it's possible to combine multiple 3D Flash LIDAR cameras to make a full volumetric 3D scene, smaller and lighter than point scanning systems, there are no moving parts, they have low-power consumption, and the ability to "see" into obscurants known as range-gating (fog, smoke, mist, haze, rain).

9.4 SAFETY AND SECURITY CONSIDERATIONS IN EMBEDDED VISION APPLICATIONS

Embedded vision (EV) systems are used across a wide range of applications from advanced driver assistance systems (ADAS) to machine vision in medical imaging, augmented reality, and many other applications. While the inclusion of an EV system adds significant benefits to the end application, it is unavoidable on developers to also ensure that the inclusion of the system cannot result in loss of life, injury, or damage to property. Achieving this requires that considering not only the safety of the design by following an engineering life-cycle and agreed upon standards, but also considering the security of the EV system to prevent it from being modified either maliciously or otherwise.

The end application for the EV system will drive the safety and security requirements. For example, a consumer application will have significantly fewer requirements than an ADAS or machine-vision system. To aid us in these design considerations and safety and security requirements, there are several well-known international standards such as IEC61508, which acts as an umbrella for many electronic systems requiring functional safety. There are also more application specific standards such as ISO26262 for automotive applications, IEC62061 for machinery, and DO178 / DO254 for flight applications. Additionally, commercial applications also require CE, UL, or CSA standards marking depending upon the end market. Each of these standards comes with development and verification requirements that need to be realized within the implementing organization's engineering, as well as in the delivery life-cycle to ensure compliance. While the heart of an EV system is the processing core, the system will typically contain an FPGA or programmable system on a chip (SoC) within which can be addressed a number of the considerations raised thus far.

What do these standards actually mean?

Many of these safety standards define levels of safety with different names, from Safety Integrity Level (SIL) for IEC61508, to Design Assurance Level (DAL) for DO254, and Automotive SIL (ASIL) for ISO26262. Within SIL, DAL, and ASIL there are also a number of differing levels which can be applied to the applications depending upon the criticality of the application. Typically, these levels are defined by the number of hours to failure, or correctly specified as the time to failure in hours. While the differing standards are generally aligned, there are some differences as shown in Table 9.2.

TABLE 9.2. SIL, ASIL, and DAL Safety Standards

SIL	ASIL	DAL	Time to Failure (hours)	Failure in Time (FIT)
		A	1,000,000,000	1
4	D		100,000,000	10
3	C	B	10,000,000	100
2	B		1,000,000	1000
1	A	C	100,000	10,000
		D	100,000	10,000
		E	1,000	1,000,000

When the design analysis is performed, it demonstrates how the required level for certification is achieved. Engineers tend to work with FIT rates, which is the reciprocal of the time to failure (hours). When working at the SIL 4 and DAL A levels, this requires a correctly architected system to achieve these requirements.

Systematic considerations

The development of safe systems requires excellent systems engineering practice with clearly defined and traceable requirements at each level of development (Figure 9.20). Figure 9.20 shows excellent systems engineering practice with clearly defined and traceable requirements at each level of development is a must for engineers developing safe and secure embedded systems.

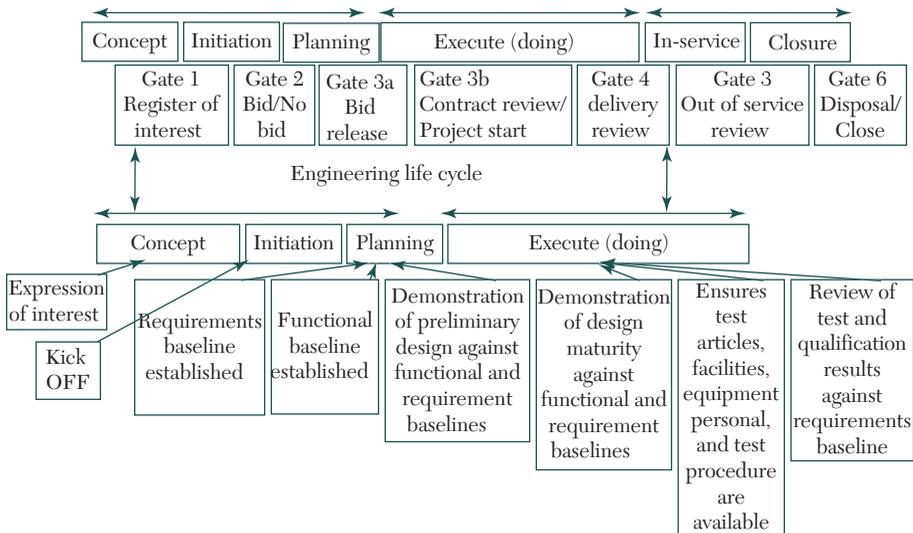


FIGURE 9.20. Excellent systems engineering practice for safe and secure embedded systems.

As in Figure 9.20, the engineering life cycle will be determined by the end application and the resultant certification required. This life cycle will define the overall engineering approach to be taken from concept to production and disposal of the EV System.

It is within this life cycle that engineering review gates controlling the progress of the project must be defined. During these reviews, independent technical experts will examine requirements, designs, technical reports

and test results to ensure the design maturity is suitable to progress to the next stage, or if further work needs to be performed to achieve the desired standard of evidence.

The engineering plan will also outline the verification and validation process at every level, which is undertaken to gain the body of evidence to achieve compliance against the applicable standard. This may require testing of the EV system across environmental operating ranges, dynamic vibration, and shock. Subjecting the EV system to accelerated life testing also ensures that the operating life of the system can be achieved. When it comes to security, consider the high-level issues engineers face attempting to secure their designs. These include the following:

Designing in quality

Obviously, depending upon the end application, component selection and manufacturing standards must be carefully considered to ensure compliance with the quality requirements for the application. When it comes to the processing core it's best to use FPGAs and SoC devices that are appropriately rated. Whether the application requires conformance to regular commercial quality standards or more demanding standards such as industrial, automotive, aerospace, or defense, the engineering team can build in quality from day one by specifying the correct component grade at the start of the project.

There are also a number of design techniques used to help achieve the tough requirements of these standards. To help ensure the design meets the reliability requirements often called the probability of success-reliability engineering techniques such as creating a reliability block diagram of the functions within the system and ensuring that any dangerous failure modes and single points of failure are eliminated, if necessary.

Within the design itself, perform failure mode effect criticality analysis, (FMECA), but keep in mind that the level at which this is performed can vary on an application by application basis from functional block to component level. The FMECA will consider the potential failure modes, the next effect, and the end effect upon the system. It will also consider if the fault can be detected by the build in a self-testing and monitoring system. If developing a component level FMECA then consider the part stress analysis (PSA) of each component within the design to ensure that it is operating with the correct derating. The level of derating applied will

depend upon the chosen standard commonly used. Standards include Department of Defense (Mil-STD 1547) and the European Space Agency (ESCC-Q-30-11A).

If a PSA is not performed, it is possible to use devices that will be over stressed and, as a result, could become the life-limiting factor on the equipment. Failure of which may or may not lead to loss or degradation of the system depending upon the FMECA predictions. Finally, along with the reliability aspects, it's also critical to perform a threat analysis on the system that will determine the threats to the system based upon the use cases and the potential mitigation strategies for the threats identified.

Architecture Case Study

At the hardware level, it's important to consider the functionality of the system and how proper implementation of the functional safety and security will be achieved. While this can be implemented from scratch, it is much better to select components that already support these features, for example, the Xilinx Zynq All Programmable SoC. The heart of any EV System is the image processing pipeline. This requires high-bandwidth processing ability combined with supervisory and control capability. The Zynq AP SoC enables a tightly integrated architecture, as opposed to the traditional processor and FPGA combination.

This tighter integration between the processor and logic fabric not only allows for a better solution but also provides for a more secure system because the interaction between the two is not available externally for malicious or other access. Within the electronic architecture, the embedded security architecture of the Zynq AP SoC can be used to provide for secure configuration. Within both the PS and the PL there is a three-stage process to ensure system partitions are secure. These comprise a hashed message authentication code (HMAC), advanced encryption standard (AES) decryption, and RSA authentication. Both the AES and HMAC use 256-bit private keys while the RSA uses 2048-bit keys, the security architecture of the Zynq AP SoC also allows for JTAG access to be enabled or disabled. These security features are enabled when upon generating the boot file and the configuration partitions for the nonvolatile boot media. It is also possible to define a fall-back partition such that should the initial first stage boot loader fail to load its application it will fall back to another copy of the application stored at a different memory location.

Once the device is successfully up-and-running, the ARM trust-zone

architecture can be used to implement orthogonal worlds which limits access to hardware functions within the Zynq AP SoC including programmable logic (PL) peripherals, as well as segment memory and L2 cache to ensure secure and non-secure worlds limiting limit interaction. When it comes to implementing the image-processing pipeline within the Zynq AP SoC PL fabric, it's also possible to use trust zone to provide secure or non-secure access to IP cores within the programmable logic fabric. This enables secure access for critical aspects of the image-processing chain preventing the ability for unauthorized changes to the configuration. The image-processing pipeline can be implemented using either custom developed or modules from the IP library.

Some safety and security implementations (IEC61508, for example) may require isolation of design elements from each other. This may be a result of the modular redundancy, differing safety areas or test functions. Enforcing physical separation between the identified zones by the use of the isolation design flow (IDF) is supported for the Zynq when used with Vivado Design Suite (Figure 9.21).

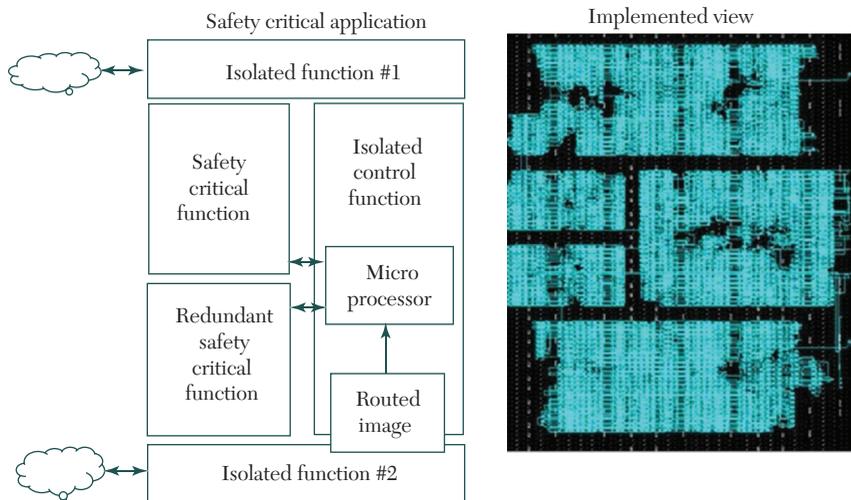


FIGURE 9.21. Isolation design flow.

Isolation design flow can be very useful when implementing majority voting within the processing chain or other control logic. Use of this ensures that the only interconnection between redundant modules is via trusted paths. When it comes to implementing the design there are also a number

of device and tool-specific implementation considerations to use. Of course, the end application and overall engineering management plan will outline the necessity to implement these techniques.

- Use of error detecting and correcting (EDAC) codes on memories, if necessary, can be combined with a scrubbing function which periodically reads and corrects the data in memory whether or not the application is accessing the memory.
- Exploiting the Hamming difference when defining control words: increasing the Hamming distance between command words while requiring more bits to implement can help with the reliability of the design.
- For critical commands, use the ARM and FIRE approach, which requires two separate commands to action critical functions.
- Use EDAC codes on external communication interfaces.
- When using the comprehensive built-in test (BIT) capability, which can otherwise report on the health or status of the system, the Zynq XADC makes for a very capable element of a BIT system as it allows the device voltages and temperatures to be monitored along with the bringing in of external signals through the mux.

Choosing Embedded Vision Software

When choosing embedded vision software, keep in mind the following considerations:

1. *Camera Choice*—The first consideration when picking embedded vision software is to determine if it works with the camera that is best suited for application. It is easy to find low-cost analog cameras, but, often, an application needs more than VGA resolution, frame rates faster than 30 frames/s, and an overall greater image quality than a standard embedded-vision camera has.
2. *Hardware Scalability*—Camera scalability is another important consideration. Because camera technologies are advancing rapidly, someday designers may want to upgrade cameras to improve image quality or measure additional features.
3. *Software Ease of Use*—Once a designer acquires an image, the next step is to process it. With the choices in algorithms today, finding the correct

tools through trial and error in a programming language can be tedious and ineffective. With this in mind, the designer needs vision software tools to help him to make the most of the algorithms.

4. *Algorithm Breadth and Accuracy*—When choosing vision software, you must determine whether the software tools can correctly and accurately measure important part or object features down to the subpixel. If the software is not accurate and reliable, then it does not matter how fast the computer is or how many pixels camera has. Keep in mind that it is much easier to make accurate code faster than to make fast code more accurate. The five most common embedded-vision application areas are listed below along with the most popular algorithms.
 - Enhancing an image—Use filtering tools to sharpen edges, remove noise, or extract frequency information. Use image calibration tools to remove nonlinear and perspective errors caused by lens distortion and camera placement. One also can use the image calibration tools to apply real-world units to their measurements, so the tools return values in microns, millimeters, or miles instead of pixels.
 - Checking for presence—This is the simplest type of vision inspection. To check for part or feature presence, one can use any of the color, pattern-matching, or histogram tools. A presence check always results in a yes/no or pass/fail.
 - Locating features—Locating features is important when aligning objects or determining exact object placement, serving as a standard for all subsequent inspections. Edge detection, grayscale pattern matching, shape matching, geometric matching, and color pattern matching are all tools you can use to locate features. The tools return the object position (X, Y) and rotation angle down to one-tenth of a pixel. Geometric matching is immune to overlapping objects or objects that change in scale.
 - Measuring features—The most common reason to use a vision system is to take a measurement. Typically, the designer uses edge detection, particle analysis, and geometric function tools to measure distance, diameter, total count, angles, and area. Whether the designer is calculating the total number of cells under a microscope or the angle between two brake-caliper edges, these tools always return a number instead of a location or pass/fail value.

- Identifying parts—Part identification is important for part compliance, tracking, and verification. Straightforward identification methods include reading a bar code or data code such as DataMatrix and PDF 417 (barcode). Newer methods use trainable OCR or object classification. Part identification often results in text or a string rather than a measurement or a pass/fail determination.

5. *Heterogeneous Processing*

One of the biggest advancement in embedded vision has been processing power. With processor performance doubling every two years, and a continued focus on parallel processing technologies like FPGAs, vision system designers can now apply highly sophisticated algorithms to visualize data and create more intelligent systems. This increase in performance means designers can achieve higher data throughput to conduct faster image acquisition, use higher resolution sensors, and take full advantage of some of the latest cameras on the market that offer the highest dynamic ranges. An increase in performance helps designers not only acquire images faster, but also allows them to process them faster. Preprocessing algorithms such as thresholding and filtering or processing algorithms such as pattern matching can execute much more quickly. This ultimately gives designers the ability to make decisions based on visual data faster than ever. Unfortunately, one of the biggest challenges to implementing an FPGA-based vision system is overcoming the programming complexity of FPGAs. Vision-algorithm development is, by its very nature, an iterative process. Designers know up front that they will have to try a few approaches with any task. Most of the time, they need to determine not which approach works but which approach works best, and “best” is different from application to application. To maximize productivity, immediate feedback and benchmarking information for algorithms is needed regardless of the processing platform being used. Seeing algorithm results in real time is a huge time-saver when using an iterative exploratory approach. However, the traditional approach to FPGA development can slow down innovation due to the compilation times required between each design change of the algorithm. One way to overcome this is to use an algorithm development tool that helps you develop for both CPUs and FPGAs from the same environment while not getting bogged down in FPGA compilation times.

6. *Integration with Other Devices*

If a designer has ever completed a vision application, then the designer knows that vision is often part of a much larger control system. In industrial automation, designed vision application may need to, control actuators to sort products, communicate inspection results to a robot controller, programmable logic controller (PLC), or embedded system, save images and data to network servers, communicate inspection parameters and results to a local or remote user interface. Often, for scientific imaging applications, the designer must integrate vision with motion stages, data acquisition systems, microscopes specialized optics, and advanced triggering.

7. *Price*

Vision software packages come in many variations. Many cater to OEM customers by splitting up their development libraries and selling algorithms. While each algorithm bundle seems lower in cost, the total vision development package cost is often quite high. Add to that the cost of a license for each component, and application deployment becomes complicated as well as costly. A vision development module features all the algorithms a designer needs to meet the toughest vision challenges so that the designer can avoid researching, buying, and maintaining multiple software bundles. Plus, deploying applications is quite inexpensive—with a single vision deployment license, the designer can deploy an executable that uses any number of vision algorithms.

Summary

- Artificial intelligence (AI) technology in the form of machine learning and deep convolutional neural networks to help vision systems learn, distinguish between objects and even recognize objects.
- Speech recognition, natural language processing, machine vision, robotics, and pattern recognition are the subfields of AI.
- A training algorithm with data that contains answers is called supervised learning. For example, identifying friend's photo by computer
- A training algorithm with data where the machine figures out the patterns is called unsupervised learning. For example, a pattern from celestial objects data.

- Giving a goal to an algorithm and expecting the machine to achieve that goal through trial and error is called reinforcement learning. For example, a robot will attempt to climb a wall until it is successful.
- In poor visibility, an artificial vision system captures an image through an IR-sensitive camera embedded in eyeglasses and analysis attributes such as object size and distance, and then gives information to the patients.
- Stereo vision is a technique to infer depth of image information from two or more cameras.
- Calibration, rectification, stereo correspondence, and triangulation are the steps in a stereo vision system.
- 3D time-of-flight (ToF) technology is revolutionizing the machine-vision industry by providing 3D imaging using a low-cost CMOS pixel array together with an active modulated light source.
- ToF applications are of two types. Gesture applications emphasize human interactions and speed, while non-gesture applications emphasize measurement accuracy.
- Simplicity, efficient, speed, price, light, and weight are some of the advantages of ToF 3D technology.
- EV safety standards are measured by time to failure and failure in time (FIT) rates.
- Enhancing an image, checking for presence, locating features, measuring features, and identifying parts are common embedded-vision application areas.

References

<https://in.mathworks.com/discovery/stereo-vision.html>

<https://www.mouser.in/applications/time-of-flight-robotics/>

<https://www.allaboutcircuits.com/capturing-3d-images-with-tof-camera-technology/>

<https://www.azosensors.com/article.aspx?ArticleID=115>

<http://www.vision-systems.com/safety-security-in-embedded-vision-applications.html>

<https://www.pathpartnertech.com/embedded-vision-approach-for-real-time-classifiers/>

Learning Outcomes

- 9.1 Write about the role of artificial intelligence technology in embedded vision.
- 9.2 Compare four different 3D-imaging technologies.
- 9.3 Define stereo vision.
- 9.4 List few applications of AI in EV.
- 9.5 Draw a block diagram of distributed-intelligence surveillance systems.
- 9.6 Draw the components of artificial vision.
- 9.7 Write about the triangulation principle in stereo vision.
- 9.8 Write the steps in achieving stereo-vision system.
- 9.9 Give a note on structured-light 3D technique.
- 9.10 Write about laser-triangulation 3D technology.
- 9.11 Write the theory of operation of a ToF camera for 3D imaging.
- 9.12 Compare stereo vision and ToF.
- 9.13 Compare structured light and ToF.
- 9.14 List a few applications of ToF.
- 9.15 List the advantages of a ToF sensor.
- 9.16 What do you mean by safety and security considerations in EV applications?

Further Readings

1. *Artificial Intelligence with Python* by Prateek Joshi
2. *Deep Learning for Computer Vision* by Isaac Rajalingappa Shanmugamani
3. *Artificial Intelligence and Machine Learning* by Vinod Chandra and Anand Hareendran

VISION-BASED REAL-TIME EXAMPLES

Overview

Embedded vision algorithms are classified in to three classes such as point, local, and global. Point-processing algorithms take an input image and generate an output image where the output value of a specific pixel is only dependent on the input value at that same coordinate. Local image-processing algorithms generate an output image at a specific pixel based on the input values in the neighborhood of the input pixel. Global transforms produce an output pixel value at every (x, y) coordinate based on all the values in the input image. Active shape model, clustering algorithms, thinning morphological operations, and Hough transform are a few methods applied in the image to extract required information. A few algorithms, R & D in EV, and a few real-time EV examples are discussed.

Learning Objectives

After reading this one will know the

- three different classification of algorithms and their usage in vision system,
- different kind of clustering algorithms,
- working of thinning operation and active-shape model,
- circle Hough transform algorithm applications,
- examples such as measurement, defect detection, bottle inspection, UAV target following, lift axle design, object tracking, ADAS, diagnostic imaging, and electronic pill, and
- ongoing and completed RD projects.

10.1 ALGORITHMS FOR EMBEDDED VISION

Most embedded vision algorithms were developed on general purpose computer systems with software written in a high-level language. This section refers to both general purpose operations (e.g., edge detection) and hardware optimized versions (e.g., parallel adaptive filtering in an FPGA). Many sources exist for general purpose algorithms.

Listed here are some general-purpose computer-vision algorithms: general image-processing functions, segmentation, transforms, machine-learning detection, machine-learning recognition, geometric descriptors, features, tracking, matrix math, robot support, image pyramids, camera calibration, stereo, 3D, utilities and data structures, fitting, and more than 500 functions are available.

One of the most-popular sources of computer vision algorithms is the OpenCV Library. OpenCV is open-source and currently written in C, with a C++ version under development. NVIDIA works closely with the OpenCV community, for example, and has created algorithms that are accelerated by GPGPUs. MathWorks provides MATLAB functions/objects and Simulink blocks for many computer vision algorithms within its Vision System Toolbox, while also allowing vendors to create their own libraries of functions that are optimized for a specific programmable architecture. National Instruments offers its LabView Vision module library. And Xilinx is another example of a vendor with an optimized computer-vision library that it provides to customers as Plug and Play IP cores for creating hardware accelerated vision algorithms in an FPGA. Other vision libraries are Halcon, Matrox Imaging Library (MIL), Cognex VisionPro, VXL, CImg, and Filters.

Three Classes

Such algorithms can be classified into three different classes: point, local and global. Point-processing algorithms take an input image and generate an output image where the output value of a specific pixel is only dependent on the input value at that same coordinate. Such point-processing algorithms can be used in a number of different ways to enhance images. One of the most useful types of point-processing algorithms is image subtraction, in which one image is subtracted from another.

Although simple, the concept is very powerful. Such algorithms are commonly used in fluoroscopy techniques such as digital subtraction angiography where it is necessary to highlight blood vessels. In this process,

an image of the tissue is first imaged before a radiological contrast agent is injected into the patient. After taking a second image, the images are subtracted, resulting in a final image where the blood vessels become more clearly visible.

Such simple algorithms are also useful in machine-vision applications, where, for example, an object may be improperly illuminated. In such applications, it is often necessary to highlight specific features within an image. If a backlight is used to image an automotive part, for example, and the task is to measure the diameter of the part, then simply thresholding the image may suffice. To perform this task, first a histogram of the image showing the number of pixels in an image at different pixel values is computed. An intensity threshold is then set and each pixel in the image compared with this threshold. If the pixel value is higher than the threshold, the pixel value is set to white. If the pixel value is lower than the threshold, the pixel value is set to black. This global thresholding results in a segmented binary image of the part that can be more easily measured.

Where lighting may be nonuniform, however, such a simple thresholding technique alone may not suffice. In some cases, it is necessary to eliminate any background nonuniformity. To accomplish this, a light field of the scene under illumination is first imaged. This can be performed by imaging the scene with a uniform white surface. A captured image of this uniform surface will then show the nonuniform light distribution. If this image is then subtracted from an image of an object under the same illumination conditions, any variation in background illumination will be reduced. After this is performed, the image can then be threshold to more accurately segment the captured object.

Generating an image histogram of an image is also useful when contrast enhancement needs to be performed. In a method known as histogram equalization, pixel values in the histogram of the original image are remapped to another histogram distribution that features a wider and more uniform



FIGURE 10.1 (a). Enhancing by histogram of image [point] (b). Reduce noise by median filter (local).

distribution of pixel values (Figure 10.1 a). By spreading these intensity values, the contrast of the image is enhanced. In many machine vision applications, enhancing the contrast of an image is used as a preprocessing step to local image processing algorithms.

Local Operators

Rather than generate an output image where the output value of a specific pixel is only dependent on the input value at that same coordinate, local-image processing algorithms generate an output image at a specific pixel based on the input values in the neighborhood of the input pixel. This technique is particularly useful to perform image sharpening, smoothing, or finding edges in an image.

To perform this task, a convolution kernel is first chosen to produce the desired effect. To reduce the amount of noise in an image, for example, a mean filter can be used. This filter replaces each pixel value in the image with an average or mean value of both its neighbors and itself. A typical 3×3 convolution kernel to perform this task may then consist of a 3×3 array with values of $1/9$ th. When this kernel is convolved across the image, the result is an image with any high-frequency components such as noise reduced. In this respect the algorithm acts as a low-pass frequency filter.

Although mean filters are useful, a more robust method of reducing noise in an image is the median filter. Here, instead of replacing each pixel value in the image with an average of both its neighbors and itself, the pixel is replaced with the median value. Because this median value represents one of the pixels in the image, the median filter retains the edge information in the image while at the same time removing noise. Figure 10.1 (b) shows that in order to reduce noise in an image, a median filter replaces pixels in the image with the median value of itself and its neighbors.

While such filters are useful for noise removal, it is often necessary to highlight the edges within an image so that some form of measurement can be later made. Here again, convolution kernels can be used to perform this task. One of the most popular of these is the Sobel filter. To highlight edges within an image, it is necessary to determine the transition points from light to dark in the image. This is accomplished by calculating the gradient of the image intensity at each point by convolving two kernels across the image to highlight the gradient in both horizontal and vertical directions. Other commonly used filters that use this same principle include the Prewitt and Roberts Cross and Laplacian operators.

While such filters may be effective in highlighting the edges of an image, in many cases, there will be missing pixels in these edges, making automated measurement difficult. To ensure lines curves and circles can be more accurately located, the Hough transform can be used after an edge-detection algorithm is performed.

Knowing that the (x, y) coordinates of any pixel along a line can be represented in polar coordinates as $\rho = x\cos\theta + y\sin\theta$, each value of x and y in the image will generate a sinusoidal curve in polar coordinate space. Performing this for bright pixels that represent edges in the image will then result in a number of different overlapping sinusoidal curves. If the curves of two different points intersect in the $\rho \theta$ plane, then both points are classified as belonging to the same line. This line then can be graphically overlaid on the image and used to compute image features. Similarly, the Hough transform can be used to determine other geometric features such as circles and ellipses by substituting the relevant Cartesian to polar coordinate equations (Figure 10.2 a).

While finding such features in images is extremely useful, it is often necessary to locate regions (or “blobs”) that are similar within an image. Perhaps the most common transform to perform this task is the Laplacian of Gaussian, an algorithm that can be implemented using convolution. After the image is first blurred using a Gaussian convolution kernel to remove noise, another convolution kernel is applied that approximates the second spatial derivative or intensity change of pixel values across the image. Applying both these kernels then results in an image where the edges of the object have been highlighted. This information can then be used, for example to find the area of the object or its center of gravity. In Figure 10.2(b) to locate edges within an image, a Laplacian of Gaussian can be applied using convolution methods.

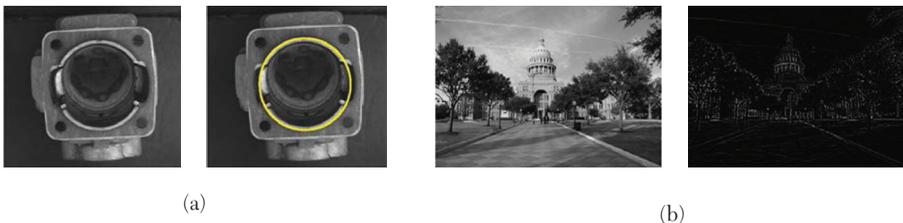


FIGURE 10.2. (a) Hough transform circle detected image. (b) Laplacian of Gaussian for edge in image

Global Transformations

While point and local transforms form the basis of many commercially available image processing libraries, the third class of image operators, global transforms produce an output pixel value at every (x, y) coordinate based on all the values in the input image. One of the most commonly used global transforms in image processing is the Fourier transform. Applying the transform to an image results in an image where each point represents a particular frequency of the original image. Figure 10.3(a) for applying the transform to an image results in an image where each point represents a particular frequency of the original image.

Because image convolution in the spatial domain (such as Sobel filtering) is equivalent to multiplication in the Fourier of frequency domain, using the Fourier transform can prove useful in applications where large, computationally intensive kernels would otherwise need to be employed. Perhaps the most important application of this transform, however has been in image compression where a variations of the algorithm, the discrete cosine transform (DCT) is used in JPEG, MPEG, and H.261 standards.

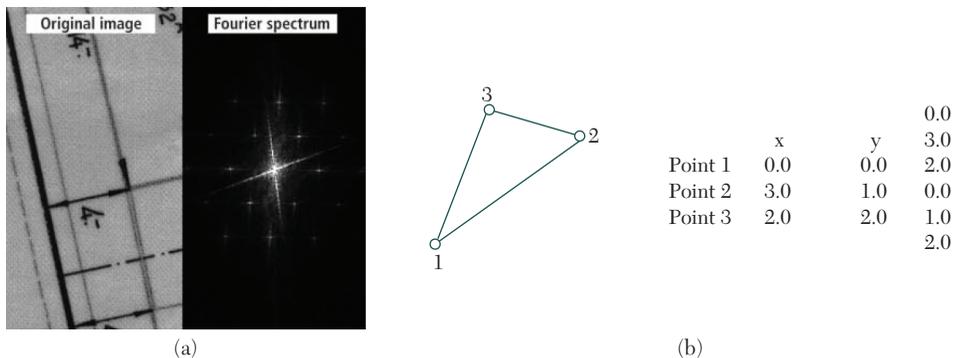


FIGURE 10.3. (a) Example for global. (b) Active model point representation.

10.2 METHODS AND MODELS IN VISION SYSTEMS

1. Shapes and Shape Models

Two-dimensional shapes and shape-models theory forms the foundation on which the ASM is built. Figure 10.3(b) left a simple shape with three points. The middle shows the same shape as an array. On the right the shape is a vector. The points are related to each other in some invariant

sense. Move, expand, or rotate the shape, it is still the same shape. Edges between points are not part of the shape but are often drawn to clarify the relationship or ordering between the points. In practice it is convenient to represent a shape array of (x, y) coordinates as a vector: first all the x and then all the y coordinates. The distance between two points is the Euclidean distance between the points. The distance between two shapes is the sum of the distances between their corresponding points. The Procrustes distance between two shapes x_1 and x_2 is the root mean square distance between the shape points $\sqrt{(x_1 - x_2) \cdot (x_1 - x_2)}$ after alignment. The centroid \bar{x} (also simply called the position) of a shape x is the mean of the point positions. The size of a shape is the root mean square distance between the shape points and the centroid.

In the face recognition system, locating facial features such as the location points of the eyes, nose, and mouth plays a significant role, because if the facial feature points are situated accurately, the following features extraction and classification stages would be more powerful and efficient. A perfect located method can upgrade the performances of linked research areas, such as a face reconstruction, face recognition, and expression recognition. Although humans can easily recognize the accurate location of the facial feature points from a face image, for the computer it is not an easy task, because the computer does not have the complex brain of a human.

The face has a complicated three-dimensional surface structure, thus, for the formation of a two-dimensional image, the change is very large, particularly for different face poses and facial expressions, as well as various lighting conditions. The two-dimensional image distinction is very evident, and a precise and effective method is a very challenging task. Trait location is very important for the analysis of linked face issues; its accuracy is immediately linked to the reliability of the following application. It is not only to supply a significant geometric information for face image processing and analysis but also plays a significant application for face recognition, facial animation, face synthesis, model-based face-image coding, expression analysis of the face pose, and so on.

Interpreting images, including objects whose appearance can vary is hard. A robust approach has been to employ deformable models, which can perform the variations in shape and/or texture (intensity) of the target objects. Precise and strong location of trait point is a complicated and difficult issue in face recognition.

2. Active Shape Model (ASM)

The shape of an object is represented through landmarks which are one chain of consecutive trait points, each of which is important, point existent in most of the images being considered, for example, the location of the right eye. Enough number of trait points should be provided to cover the comprehensive shape and details. A group of landmarks forms a shape. Meanwhile, the shapes are represented as vectors: all the x coordinates pursued by the y coordinates of the points in the form. Align one shape to other with a correspondence transform (allowing rotation, scaling, and translation) that reduce the Euclidean distance average between shape points. The mean shape is declared the middle of the stratified training shapes. The ASM beginning the search for facial landmarks from the mean shape aligned to the place and size of the face specified by a global face detector.

It then reiterates the following two steps until convergence:

- i. Propose a temporary shape by determining the positions of shape points through template appropriating from the image texture concerning each point.
- ii. Confirms the temporary shape to a universal shape model. The special template suitability is uncertain and the form model gathers the outcomes of the weakened form matches to form a powerful classifier. The complete search is reiterated at each level in an image pyramid, from poor to fine resolution.

Active shape model location

One approach to locating examples of objects whose shape can vary (e.g., faces or internal organs) is to use active shape models (ASMs). This method relies on building a statistical model of shape variation from examples in a training set. Each example object is represented using a fixed number of landmark points (x_i, y_i) ($i = 1 \cdots N_p$), each of which marks a particular point on the object. The training examples are aligned into a common co-ordinate frame and each is then represented by $2N_p$ element vector $x = (x_1, y_1, \cdots, x_{N_p}, y_{N_p})^T$. If we make the assumption that such vectors have a Gaussian distribution for the training set we can build a linear model as follows: $x = \hat{x} + Pb$. Where \hat{x} is the mean of the training set, b is a vector of t shape parameters and P is a $2N_p \times t$ matrix formed from the t principle eigen vectors of the covariance matrix of the training set.

The t shape parameters, b_i , are then mutually independent and the i^{th} is Gaussian with zero mean and variance λ_i (the i^{th} largest eigen value of the covariance matrix). For instance Figure 10.4 show the effects of varying the parameters of a face model which uses 169 points to represent the shape of various facial features.

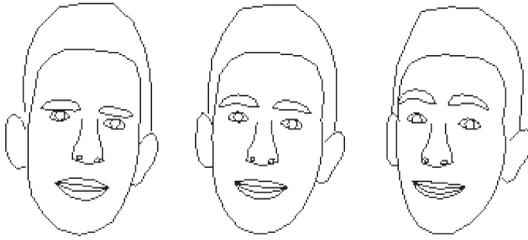


FIGURE 10.4 Effect of varying shape parameter 169 point face model.

3. Clustering Algorithms

Clustering is a machine learning technique that involves the grouping of data points. Given a set of data points, a clustering algorithm is used to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

K-means clustering

1. First select a number of classes/groups to use and randomly initialize their respective center points. The center points are vectors of the same length as each data point vector.
2. Each data point is classified by computing the distance between that point and each group center, and then classifying the point to be in the group whose center is closest to it.
3. Based on these classified points, recompute the group center by taking the mean of all the vectors in the group.
4. Repeat these steps for a set number of iterations or until the group centers don't change much between iterations. One can also opt to randomly initialize the group centers a few times, and then select the run that looks like it provided the best results.

K-means has the advantage that it's pretty fast, as all we're really doing is computing the distances between points and group centers; very few computations! It thus has a linear complexity $O(n)$.

On the other hand, K-means has a couple of disadvantages. First, one has to select how many groups/classes there are. This isn't always trivial and ideally with a clustering algorithm we would want it to figure those out for us because the point of it is to gain some insight from the data. K-means also starts with a random choice of cluster centers and therefore it may yield different clustering results on different runs of the algorithm. Thus, the results may not be repeatable and lack consistency. Other cluster methods are more consistent.

K-medians is another clustering algorithm related to K-means, except instead of recomputing the group center points using the mean, we use the median vector of the group. This method is less sensitive to outliers (because of using the median) but is much slower for larger datasets as sorting is required on each iteration when computing the median vector.

Mean shift clustering

Mean shift clustering is a sliding-window-based algorithm that attempts to find dense areas of data points. It is a centroid-based algorithm meaning that the goal is to locate the center points of each group/class, which works by updating candidates for center points to be the mean of the points within the sliding window. These candidate windows are then filtered in a post-processing stage to eliminate near duplicates, forming the final set of center points and their corresponding groups. Mean shift clustering for a single sliding window is explained here:

1. In mean shift, consider a set of points in two-dimensional space and begin with a circular sliding window centered at a point C (randomly selected) and having radius r as the kernel. Mean shift is a hill-climbing algorithm that involves shifting this kernel iteratively to a higher-density region on each step until convergence.
2. At every iteration the sliding window is shifted toward regions of higher density by shifting the center point to the mean of the points within the window (hence the name). The density within the sliding window is proportional to the number of points inside it. Naturally, by shifting to the mean of the points in the window it will gradually move toward areas of higher-point density.

3. Continue shifting the sliding window according to the mean until there is no direction at which a shift can accommodate more points inside the kernel. That is, keep moving the circle until you are no longer increasing the density (i.e., number of points in the window).
4. This process of steps 1 to 3 is done with many sliding windows until all points lie within a window. When multiple sliding windows overlap the window containing the most points is preserved. The data points are then clustered according to the sliding window in which they reside.

In contrast to K-means clustering there is no need to select the number of clusters as mean shift automatically discovers this. That's a massive advantage. The fact that the cluster centers converge toward the points of maximum density is also quite desirable as it is quite intuitive to understand and fits well in a naturally data-driven sense. The drawback is that the selection of the window size/radius “ r ” can be nontrivial.

Density-based spatial clustering of applications with noise (DBSCAN)

DBSCAN is a density based clustered algorithm similar to mean shift, but with a couple of notable advantages.

1. DBSCAN begins with an arbitrary starting data point that has not been visited. The neighborhood of this point is extracted using a distance epsilon ϵ (All points which are within the ϵ distance are neighborhood points).
2. If there are a sufficient number of points (according to minPoints) within this neighborhood, then the clustering process starts and the current data point becomes the first point in the new cluster. Otherwise, the point will be labeled as noise (later this noisy point might become the part of the cluster). In both cases that point is marked as “visited.”
3. For this first point in the new cluster, the points within its ϵ distance neighborhood also become part of the same cluster. This procedure of making all points in the ϵ neighborhood belong to the same cluster is then repeated for all of the new points that have been just added to the cluster group.
4. This process of steps 2 and 3 is repeated until all points in the cluster are determined, that is, all points within the ϵ neighborhood of the cluster have been visited and labeled.

5. Once done with the current cluster, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise. This process repeats until all points are marked as visited. Since at the end of this all points have been visited, each point will have been marked as either belonging to a cluster or being noise.

DBSCAN poses some great advantages over other clustering algorithms. First, it does not require a preset number of clusters at all. It also identifies outliers as noises unlike mean shift which simply throws them into a cluster even if the data point is very different. Additionally, it is able to find arbitrarily sized and arbitrarily shaped clusters quite well.

The main drawback of DBSCAN is that it doesn't perform as well as others when the clusters are of varying density. This is because the setting of the distance threshold ϵ and `minPoints` for identifying the neighborhood points will vary from cluster to cluster when the density varies. This drawback also occurs with very high dimensional data since again the distance threshold ϵ becomes challenging to estimate.

Expectation maximization (EM) clustering using Gaussian mixture models (GMM)

One of the major drawbacks of K-means is its naive use of the mean value for the cluster center. K-means also fails in cases where the clusters are not circular, again as a result of using the mean as cluster center.

Gaussian mixture models (GMMs) give us more flexibility than K-means. With GMMs assume that the data points are Gaussian distributed; this is a less restrictive assumption than saying they are circular by using the mean. That way, two parameters to describe the shape of the clusters: the mean and the standard deviation. Taking an example in two dimensions means that the clusters can take any kind of elliptical shape (since we have standard deviation in both the x and y directions). Thus, each Gaussian distribution is assigned to a single cluster. In order to find the parameters of the Gaussian for each cluster (e.g., the mean and standard deviation) an optimization algorithm called expectation maximization (EM) is used.

1. Begin by selecting the number of clusters (like K-means does) and randomly initializing the Gaussian distribution parameters for each cluster. One can try to provide a good guesstimate for the initial parameters by taking a quick look at the data too.

2. Given these Gaussian distributions for each cluster, compute the probability that each data point belongs to a particular cluster. The closer a point is to the Gaussian's center, the more likely it belongs to that cluster. This should make intuitive sense since with a Gaussian distribution we are assuming that most of the data lies closer to the center of the cluster.
3. Based on these probabilities, compute a new set of parameters for the Gaussian distributions such that maximize the probabilities of data points within the clusters. Compute these new parameters using a weighted sum of the data-point positions, where the weights are the probabilities of the data point belonging in that particular cluster.
4. Steps 2 and 3 are repeated iteratively until convergence, where the distributions don't change much from iteration to iteration.

There are really 2 key advantages in using GMMs. Firstly GMMs are a lot more flexible in terms of cluster covariance than K-means; due to the standard deviation parameter, the clusters can take on any ellipse shape, rather than being restricted to circles. K-means is actually a special case of GMM in which each cluster's covariance along all dimensions approaches 0. Secondly, since GMMs use probabilities, they can have multiple clusters per data point. So if a data point is in the middle of two overlapping clusters, simply define its class by saying it belongs X-percent to class 1 and Y-percent to class 2. In other words, GMMs support mixed membership.

Agglomerative hierarchical clustering

Hierarchical clustering algorithms actually fall into two categories: top down or bottom up. Bottom-up algorithms treat each data point as a single cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all clusters have been merged into a single cluster that contains all data points. Bottom-up hierarchical clustering is therefore called hierarchical agglomerative clustering or HAC. This hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample.

1. Begin by treating each data point as a single cluster, that is, if there are X data points in dataset then have X clusters. Then select a distance metric that measures the distance between two clusters. As an example average linkage is used which defines the distance between two clusters to be the average distance between data points in the first cluster and data points in the second cluster.

2. On each iteration combine two clusters into one. The two clusters to be combined are selected as those with the smallest average linkage. That is, according to selected distance metric, these two clusters have the smallest distance between each other and therefore are the most similar and should be combined.
3. Step 2 is repeated until the root of the tree is reached, or only have one cluster which contains all data points. In this way one can select how many clusters want in the end, simply by choosing when to stop combining the clusters, that is, when one stops building the tree!

Hierarchical clustering does not require us to specify the number of clusters and one can even select which number of clusters looks best since we are building a tree. Additionally, the algorithm is not sensitive to the choice of distance metric; all of them tend to work equally well whereas with other clustering algorithms, the choice of distance metric is critical. A particularly good-use case of hierarchical clustering methods is when the underlying data has a hierarchical structure and you want to recover the hierarchy; other clustering algorithms can't do this. These advantages of hierarchical clustering come at the cost of lower efficiency, as it has a time complexity of $O(n^3)$, unlike the linear complexity of K-means and GMM.

4. Thinning Morphological Operation

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening. It can be used for several applications, but is particularly useful for skeletonization. In this mode it is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output.

Thinning working

Like other morphological operators, the behavior of the thinning operation is determined by a structuring element. The binary structuring elements used for thinning are of the extended type described under the hit and miss transform (i.e., they can contain both ones and zeros). The thinning of an image I by a structuring element J is:

$$\text{thin}(I, J) = I - \text{hit} - \text{andmiss}(I, J)$$

Where the subtraction is a *logical subtraction* defined by, $X - Y = X \cap \text{NOT } Y$.

In everyday terms, the thinning operation is calculated by translating the origin of the structuring element to each possible pixel position in the image, and at each such position comparing it with the underlying image pixels. If the foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image, then the image pixel underneath the origin of the structuring element is set to background (zero). Otherwise it is left unchanged. Note that the structuring element must always have a one or a blank at its origin if it is to have any effect.

The choice of structuring element determines under what situations a foreground pixel will be set to background, and hence it determines the application for the thinning operation. In fact, the operator is normally applied repeatedly until it causes no further changes to the image (i.e., until *convergence*). Alternatively, in some applications, for example, *pruning*, the operations may only be applied for a limited number of iterations. Thinning is the dual of thickening, that is, thickening the foreground is equivalent to thinning the background.

Uses of thinning

One of the most common uses of thinning is to reduce the threshold output of an edge detector such as the Sobel operator, to lines of a single pixel thickness, while preserving the full length of those lines (i.e., pixels at the extreme ends of lines should not be affected). A simple algorithm for doing this is the following: Figure 10.5 shows the result of this thinning operation on a simple binary image.

Thinning is often used in combination with other morphological operators to extract a simple representation of regions. A common example is the automated recognition of hand written characters. In this case,

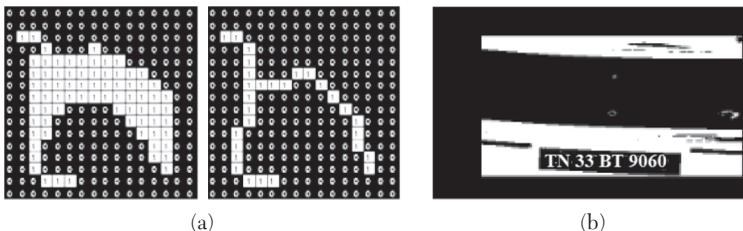


FIGURE 10.5. (a) Morphological thinning of binary shape. (b) Plate edge detection.

morphological operators are used as preprocessing to obtain the shapes of the characters which then can be used for the recognition.

Localization of license plate using morphological operations

License plate recognition (LPR) is an image-processing technology used to identify vehicles by their license plates. Intelligent transport system is a real time, accurate, and efficient transportation management system that can solve the various road problems generated by traffic congestion, thus receiving more and more attention. Morphological operations are used to fill the gaps between characters in an edge image to make rectangular regions. The algorithm uses morphological operations on preprocessed, edge images of the vehicles. Characteristic features such as license plate width and height, character height, and spacing are considered for defining structural elements for morphological operations. The basic morphological operations are erosion and dilation. Dilation is used to fill the gaps or holes. A morphological operator is used to dilate the image once horizontally and the other time vertically. Another horizontal dilation is employed on the common bright pixels. The structuring elements of dilations are pixel horizontal or vertical lines. Due to digits and characters, a license plate contains many vertical edges. This feature is employed for locating the plate in an image.

This method is one of the techniques used to detect license plates. RGB images are more difficult to process, so the original image is converted into a grayscale image. Smoothing is used to extract more information from the data. The median filter is normally used to reduce noise in an image. An image mask isolates parts of an image for processing. The binary thresholding method is used to separate object and background. Binarization is used to process each pixel in an image that is converted into one bit. The Sobel operator performs a 2-D spatial-gradient measurement on an image. The canny operator inputs a grayscale image, and outputs an image showing the positions of tracked intensity discontinuities. Erosion is often used to remove irrelevant details from a binary image. Opening is the combination of erosion dilation. Morphological methods were used to achieve the detection of a license plate, as shown in Figure 10.5 (b).

5. Hough Transform (HT)

The Hough transform is a way of finding the most likely values that represent a line (or a circle, or many other things). To the Hough transform

a picture of a line is given as input. This picture will contain two types of pixels: those that are part of the line, and those that are part of the background.

For each pixel that is part of the line, all possible combinations of parameters are calculated. For example, if the pixel at coordinate $(1, 100)$ is part of the line, then that could be part of a line where the gradient $(m) = 0$ and y -intercept $(c) = 100$. It could also be part of $m = 1, c = 99$; or $m = 2, c = 98$; or $m = 3, c = 97$; and so on. The line equation $y = mx + c$ is solved to find all possible combinations.

Each pixel gives one vote to each of the parameters $(m$ and $c)$ that could explain it. If your line has 1000 pixels in it, then the correct combination of m and c will have 1000 votes. The combination of m and c , which has the most votes, is what is returned as the parameters for the line.

Circle Hough transform (CHT)

In the line detection case, a line was defined by two parameters (r, θ) . In the circle case, we need three parameters to define a circle $(x_{center}, y_{center}, r)$. Here, (x_{center}, y_{center}) define the center position and r is the radius, which allows us to completely define a circle. The HT is used in detecting lines. However, it has evolved over the years to identify other analytical shapes such as circles and ellipses. Both of the HT and the circular Hough transform (CHT) usually depend on converting grayscale images to binary images. They both use a preliminary edge detection technique such as Sobel or Canny. With respect to ellipse detection, an ellipse can be defined by five parameters, its center (X_c, Y_c) , the major axis a , the minor axis b , and the slope θ as shown in Figure 10.6.

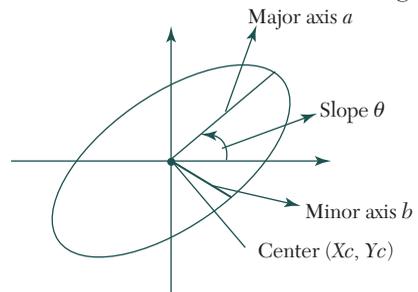


FIGURE 10.6. Ellipse parameters.

Due to the noise immunity and other merits of the HT, it has been widely used in many applications. For example, 3D applications, detection of objects and shapes, lane and road sign recognition, industrial and medical applications, pipe and cable inspection, and underwater tracking.

10.3 REAL-TIME EXAMPLES

1. Embedded-Vision-Based Measurement

Due to continuing and rapid advances of both hardware and software technologies in camera and computing systems, we continue to have access to cheaper, faster, higher quality, and smaller cameras and computing units. As a result, vision-based methods consisting of image processing and computational intelligence can be implemented more easily and affordably than ever using a camera and its associated operations units.

Instrumentation and measurement (I&M) as a field is primarily interested in measuring, detecting, monitoring, and recording of a phenomenon referred to as the measurand, and associated calibration, uncertainty, tools, and applications. While many of these measurands are invisible to the human eye, for example, the amount of electrical current in a wire, there are many others that can be seen visually, such as the number of people in a room. As such, it is intuitive to develop tools and methods that would “see” the measurand similar to the human eye and measure it. Such tools would be primarily electrical and/or electronic devices, possibly (though not necessarily) computer based, and would receive a picture of the scene from a camera or similar visual sensor, sometime sensible to a

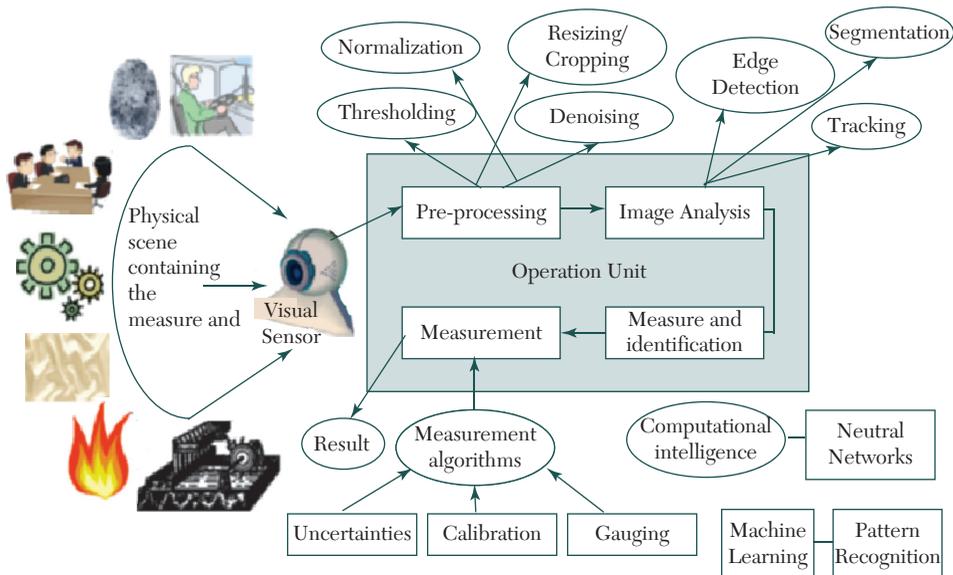


FIGURE 10.7. Vision-based measurement.

wider band of electromagnetic radiation (infrared, UV, X-ray, etc.) than the human eye, and perform certain operations and/or computational processes to measure or detect the subject of interest. Figure 10.7 shows high-level architecture of vision-based measurement. Left to right: An image is acquired by a *visual sensor*, and is fed to the *operations unit* to perform image processing, computational intelligence, and measurement operations.

Visual sensor

The visual sensor can be a visible-light camera, an infrared camera, a laser scanner, an X-ray scanner, or any other sensor that can obtain an image of the physical scene containing the measurand. Since the most commonly used visual sensor is visible-light camera such as a complementary metal-oxide-semiconductor (CMOS) or higher resolution charge-coupled device (CCD), the captured image is most of the time very similar to a picture of the scene as seen by a human. But for other type of sensors such as laser or X-ray, this image is different from what a human sees and is mostly meant for the consumption of the operations unit. Irrespective of the type of visual sensor, a key contributing factor for accurate measurements is the calibration of the camera and precise knowledge of its position, orientation, focal length, aspect ratio, principle point, distortion, and so on.

Operations unit

The operations unit receives the image acquired by the visual sensor and performs the necessary operations to obtain the desired measurements. This unit can be implemented in either software or hardware; that is, it can either be programmed into a generic microprocessor based system, such as the processing unit of a smart camera, or it can be implemented in dedicated hardware, such as field programmable gate array (FPGA) or application-specific integrated circuit (ASIC). The unit itself consists of the following four major stages:

1. **Preprocessing:** The purpose of this stage is to prepare the raw image for the next stage of operations. The image as acquired by the visual sensor could have deficiencies such as glare, noise, blurs, and so on. In addition, it might not be in the form required by ensuing operations. For example, a fingerprint image is typically acquired in grey scale, but to be processed it typically needs to be converted to pure black and white without any background. Preprocessing takes care of such needs and performs

operations such as normalization which modifies the pixel intensity and contrast of parts of the image, thresholding which converts the image into a binary black and white image, denoising which rids the image from additive white Gaussian noise or other types of noise, resizing, cropping, and so on. These operations are signal processing, specifically image processing, with many methods and algorithms available for their implementation.

2. **Image Analysis:** The purpose of this stage is to analyze the image and extract the necessary information for finding the measurand and doing the measurements later. This stage also uses image-processing operations, such as segmentation, which divides the image into multiple segments each representing something meaningful in the scene, edge detection which finds the edges of objects in the scene and helps us identify objects of interest, tracking of objects after they have been detected and as they move through the scene, and so on. For example, we can see color analysis and contour detection applied to food images in order to detect individual ingredients. At the end of the image analysis stage, the output is either the measurand itself, or is information that can lead to the identification of the measurand. In the former case, we can skip the next stage, measurand identification, and move straight to the measurement stage. For example, to count the number of people in a room by counting the number of faces, once the faces have been detected in the image analysis stage, we can move straight to counting them without any further operations. However, in some applications, more operations are needed to identify the measurand. For example, in food images, even though individual ingredients have been detected, we still don't know what they are exactly (apple? orange? bread? etc.). Hence, an additional identification stage is needed to answer this question. This stage is typically performed using computational intelligence operations, as discussed next.
3. **Measurand Identification:** The purpose of this stage is to identify the specific measurand in the image, if it hasn't already been identified in the previous stage of image analysis. Techniques that are used here are mostly based on computational intelligence, especially machine learning, and specifically pattern recognition and pattern matching, the former providing a reasonable "most likely" matching of the given inputs to an output and hence introducing some uncertainties, while the latter looks for and reports exact matches of the given inputs to an a priori pattern. In this stage, we can find, match, and identify specific patterns, shapes, and

classes of objects in order to identify our measurand. Optical character recognition and neural networks are also done at this stage if needed. For example, by feeding the output of food images (after color analysis and contour detection) into a Support Vector Machine engine that has been previously trained with similar food images in terms of color, texture, shape, and size, we can identify what ingredients exist in the food, with a certain degree of accuracy. In some applications where the physical phenomenon needs to be only detected, as opposed to gauged, such as gesture detection, our task is finished at this stage with the detection and identification of the measurand. But in many other applications, the measurand has to go through further measurement operations, as discussed next.

4. **Measurement:** at this stage we have the measurand and we can perform the required measurement operation such as gauging which gives us the dimensions of the measurand and its circumference, area, volume, etc., as well as temporal measurements when tracking the measurand and its state over time. For example, where the area of a single food ingredient that has been identified in the previous stage is determined. By assuming a more or less constant thickness of the ingredient, we can measure its volume from the area, use readily available food density tables to find the mass of the ingredient, and use nutritional tables to measure its calories and nutrition. Calibration is another requirement at this stage. In the food example, we need a reference to know the dimensions of the food ingredient, which may be the user's thumb that has been measured before and can be used for calibration here. As another example for temporal measurements, consider a driver monitoring application where, to detect yawning, we must first detect and track a closed mouth, then detect if the same mouth opens according to a certain pattern over a certain time, and then is closed again. The temporal relationship between the various states of the mouth is of outmost importance, otherwise there will be false positives because singing or talking will be mistaken for yawning.

Uncertainties and their sources

Visual-based measurement (VBM) systems, like every system employed for measurement purposes, can be considered as actual measurement systems, if they provide measurement results.

Under this assumption, the only significant remaining effects are random and, consequently, the dispersion of values that could reasonably be

attributed to the measurand, can be represented by the standard deviation of a given or assumed PDF. This standard deviation is called *standard uncertainty* and represents the fundamental stone on which measurement uncertainty is evaluated, also when the measurement result is not directly provided by a single instrument, but is obtained as a combination of measurement results.

According to these concepts, to characterize a VBM system as a measuring instrument, it is imperative that the following steps are accomplished:

1. All significant systematic effects shall be identified and recognized, and proper corrections shall be applied.
2. The dispersion of values that could reasonably be attributed to the measurand shall be characterized in terms of standard uncertainty.
3. If different parts of the instrument, both hardware components and algorithms, are expected to contribute to the dispersion of values that could reasonably be attributed to the measurand, steps 1 and 2 shall be repeated for all of them, and the individual obtained standard uncertainty values shall be suitably combined in order to obtain the final combined standard uncertainty associated to the measured value provided by the VBM system.

Specifically, as far as the VBM visual sensors are concerned, we can list the following main sources of uncertainties:

- **Lighting:** The lighting of the scene directly affects the values of the pixels of the resulting image, which affect the image processing parts in Figure 10.7, and since the output of the image processing parts are input to the remaining parts, we can see that lighting conditions in fact affect the entire measurement system. Hence, applications in which the lighting condition may vary are affected by this parameter. Lighting conditions can be seen either as systematic effects (for instance the presence of shadows is a systematic effect if they do not change during the whole measurement process) and random effects (for instance due to short term fluctuations of the lighting conditions). Both effects shall be taken into account when evaluating uncertainty.
- **Camera angle:** The angle with which the image is taken is also important in applications where the camera has a free angle and is not fixed, since

the angle directly affects the shape and position of the measurand in the image. Also in this case a systematic effect shall be considered and compensated for (due to the camera position) and the random effects shall be also considered, related to fluctuations of the camera position due to imperfections of the camera bearing system, vibrations, and so on.

- Camera equipment: Different cameras have different lenses, hardware, and software components, all affecting the resulting image taken with that camera. Hence, an application that is not using a specific and predefined camera can be affected by this parameter. Again, this may originate systematic effects as well as random effects and both shall be carefully considered.

There are also other uncertainties introduced in the particular image processing or computational intelligence algorithms used in the VBM system, which must also be taken into account. As an example, denoising algorithms are not 100% efficient and some noise is still present in the output image. This noise represents a contribution to uncertainty, and as such, it has to be evaluated and combined with other contributions to define the uncertainty associated with the final measurement result.

Identifying and evaluating all individual contributions to uncertainty is also essential to compare different possible architectures (hardware and software) and understand which one provides the best performance, from the metrological perspective, under the different possible measurement conditions. This can be efficiently done only if well-established standards and techniques are used.

2. Defect Detection on Hardwood Logs Using Laser Scanning

Over the past few decades a broad variety of scanning technologies have emerged for wood processing. Several scanning and optimization systems are on the market that aid in the sawing of logs into lumber. Among them are defect detection and classification systems for logs and stems. Defect detection on hardwood trees and logs is categorized into internal and external detection. Internal detection determines defects inside logs, while external detection identifies defects on a log's surface. Currently, most available scanning systems are external methods that use a laser-line scanner to collect rough log profile information. These systems were typically developed for softwood (i.e., pine, spruce, fir) log processing and for gathering information about external log characteristics such as

diameter, taper, curvature, and length. Once log shape data are obtained, a previously generated cutting pattern or template is selected that best fits the log. Optimization systems then use this profile information to better position the log on the carriage with respect to the saw and to improve the sawyer's decision-making ability. Adding external defect information to the optimization process is a natural extension of current technology.

A portable, demonstration laser log-scanner was used to collect the log-surface data. The scanner comprises four laser-line generator/camera units stationed at 90-degree intervals around the log's circumference. The scanner utilizes triangulation to determine locations of log surface points covered by the laser-line. The log stands still while the carriage holding the four scanning units moves on rails along the log's length. A transducer records the lineal position of the scanner accurate to 0.01 in. Depending on the circumference of the log at any specific location, the number of points in each cross-section varies. However, on average the distance between points in each cross section is 0.04 in. When a sequence of cross sections is assembled, a three-dimensional map of the log surface is obtained. An additional computer program has been developed using OpenGL to render realistic views of the scanned log surfaces. This program is especially useful for visually examining the logs and comparing detected defects with both the visible and manually recorded defect locations.

3. Reconstruction of Monocular Fiberscopic Images

In many industrial and medical applications, glass fiber endoscopes are used to acquire images from complex hollows for diagnostic and interventional purposes. For a complete exploration and understanding of a hollow cave, 3D reconstruction is necessary. Typical examples in the field of industrial machine inspection are complex drill holes or coolant bores in turbine blades. In the medical domain, fiberscopes are applied for diagnostic tasks such as laryngoscopy, bronchoscopy, and for minimal invasive surgery scenarios.

Due to the nature of light transmission into the observed cavity and image transmission out of the cavity by the sole use of glass fibers, the acquired and observed image scene is subsampled by the amount of glass fibers that form the flexible image conductor. The amount of fibers depends on the diameter of the endoscope and the physical dimensions of the fibers. Thus, using fiberscopes in combination with cameras and the goal to observe, acquire, digitize, and reconstruct a complex scene of

a cavity, a crude subsampling is made on the tip of the fiberscope, while an oversampling takes place, where the image transmitted by the glass fibers is acquired with a camera. An image of a test-chart, observed with a fiberscope is depicted in Figure 10.8, with subregions enlarged to show the typical fiber-effect. It can be seen, that each group of pixels—denoting the intensity of an image point transmitted by a glass fiber is surrounded by a dark ring. This is due to the transmission property of glass fibers. Every optical fiber consists of a core with a high refractive index n_1 and a cladding with a lower refractive index n_2 . Light rays, which enter the fiber at one end, are guided along the core by total internal reflections at the core-cladding interface. This total internal reflection can be described by Snellius' law: $\sin \alpha_c = n_2/n_1$, where α_c denotes the critical angle of total reflection. The reflected light rays follow the bends in the fiber and exit it at its other end. For fiberscopes, bundles of optical fibers are combined with appropriate end terminations and protective sheathing to form light guides.

The elastic image conductor of current available fiberscopes consists of a sorted bundle of fibers made of glass or quartz crystal. According to the described core-cladding relationship, fibers show a homogeneous alignment of bright transmission points, surrounded by a dark borders over the entire image, the so-called “comb” structure (see Figure 10.8).

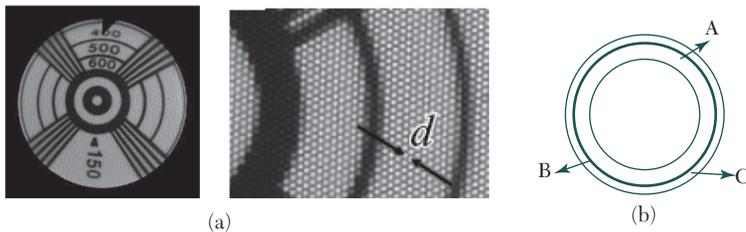


FIGURE 10.8. (a). Test-chart and enlarged honeycomb structure. (b). Regions of dividing of a bottle mouth.

The elimination of the comb-structure pattern consists of four major steps. The first step consists of an adaptive filtering approach, making the feature and correspondence detection more robust. The second step deals with the calibration and distortion correction. The third step consists of the correspondence-calculation and feature tracking. This part also includes the 3D reconstruction process as well as self-calibration. The fourth step deals with two knowledge-driven extensions of the reconstruction process, that enhance the results significantly.

4. Vision Technologies for Empty Bottle Inspection Systems

Vision technologies have two different approaches for inspecting glass bottles: the ones in the first group locate and track bottle mouth, bottle bottom and walls while the other group technologies involve defect detecting. Such vision inspection systems are required to perform with high accuracy and adaptability under high speed and mechanical vibration working conditions.

Bottle bottom inspection

Image pre-processing

Because of the high-speed movement of empty bottles on transmission, the original images captured by a CCD camera are not ideal for things such as noise interference and uneven illumination, which may appear when the target deviates from the center of the light source. Therefore, the five nearest-neighbor smoothing filter is used to remove noise and to achieve the grayscale transformation to enhance image contrast and improve the effect of output images.

Edge detection

After enhancing the contrast of images, edge detection of a bottle bottom is achieved.

Bottom positioning

In the process of empty bottles transmission at high speed, the bottle bottom position in each image may vary because of mechanical vibration and time error of image acquisition. Therefore, it is necessary to calculate the position of the bottle bottom in each image accurately to ensure it is in the region of interest. In practical application, the required velocity of empty bottles detecting is up to 20 bottles, so it is crucial to develop a quick and efficient positioning algorithm to improve the performance of the system. Bottom positioning is actually a circle detection in the image of a bottle bottom. Therefore, in order to obtain all parameters of the circle quickly and efficiently, the clutter edge is filtered first and then gets the parameters of the circle on this basis.

After edge detection, the image of a bottle bottom contains abundant edge information, in which some edges unimportant or irrelevant to bottom positioning will influence the effect of location greatly. Therefore, these unnecessary edges must be filtered. Then the unnecessary edges can be

removed according to the perimeter threshold set after calculating each edge perimeter by chain-code tracing. After clutter edge filtering, circle detection algorithm is employed to improve the real-time performance of circle detection.

Bottle mouth inspection

Bottle mouth location and tracking are the most difficult tasks in empty bottle vision inspection. The speed and accuracy of this step have a great influence on the performance of the inspection system. The region of interest in a bottle mouth image is defined in order to improve the processing speed. The regions of rings in the image are defined as areas A, B, and C as shown in Figure 10.8 (b). Region A, the circular area, is the image of sealing surface; region B is the middle circular ring in the image; and region C is the area from out edge of the middle circular ring to the outside of bottle mouth illuminated by a light source.

The bottle mouth position is different in each image because of high speed, mechanical vibration, time error of image acquisition, and the poor quality of a bottle neck. This means that the processing region is unfixed. Therefore, it is necessary to determine the size and position of the region that needs to be processed. This means that we need to determine the center coordinates and the inner and outer diameters of the three circles according to the feature of a bottle mouth image.

The difficulty here is that the width of region B is only four to five pixels, so the absolute errors of the extracted features, such as the coordinate of circle center, inner radius, and outer radius, must be within two pixels. Therefore, the automatic detection and tracking algorithm should have the advantages of minor calculation and high accuracy. Therefore, the focus shifted to the domain of lines crossing the bottle mouth edge and grayscale scanning along the lines is applied. Then the edge point sets of bottle mouth image can be acquired by classifying the saltation grayscale pixels obtained. Finally, the least square circle fitting is applied to the image edges to get bottle mouth parameters.

Bottle mouth visual defect detection algorithm for empty bottles

The processing area of a bottle mouth can be obtained after getting its actual location. In order to improve the processing speed, fixed threshold is applied in the binary segmentation of the bottle mouth image because of the obvious contrast between the target and background regions. According

to the image characteristics and the regions defined in Figure 10.18(b), the defect detection of bottle mouth involves fracture detection in region B and spottiness detection in regions A and C.

Feature extraction

In the binary image obtained by threshold segmentation, the gray value of the background area in region B is 255 and the target area, that is, the defect area, is 0. In contrast, the value of background area of regions A and C are 0 and the target area is 255. The run-length encoding algorithm is applied to analyze the connectivity. Numbered the connected domains obtained successively and the serial number N_d of the last one is the number of connected domains, which are likely the defect areas. Suppose the area of the connected domain is A_j and it is used to prejudge the defects. Set an area threshold value T_s , and if A_j is greater than T_s , the connected domain may be defect area, otherwise it may be the noise area and should be removed.

Defect recognition

The method based on a connected domain search is used to extract the features in the three regions in a bottle mouth image. The features are number of defect areas, total area of defect areas, area of the largest in the defect areas, width of the defect areas, distance from the center of defect area to the center of circular ring, and posture ratio of defect areas. After features extraction, defect areas are judged. The three target regions in a bottle mouth image and output are analyzed and the result denotes whether there are defects in the bottle.

5. Unmanned Rotorcraft for Ground Target Following Using Embedded Vision

Unmanned aerial vehicles (UAVs) have recently aroused much interest in the civil and industrial markets, ranging from industrial surveillance, agriculture, and academic research, to wildlife conservation. The unmanned rotorcraft has received much attention in the defense and security community. More specifically, an unmanned rotorcraft equipped with a vision payload can perform a wide range of tasks, such as search and rescue, surveillance, target detection and tracking, and so on, as vision provides a natural sensing modality, in terms of human comprehension for feature detection and tracking. A real-time vision software is developed, and is running on the real-time operating system QNX.

Hardware configuration of the vision system

The hardware configuration of on-board vision system for the UAV is as illustrated in Figure 10.9 and consists of the following five main parts: a visual sensor, an image acquisition module, a vision processing module, a pan/tilt servo mechanism, and a video and data link.

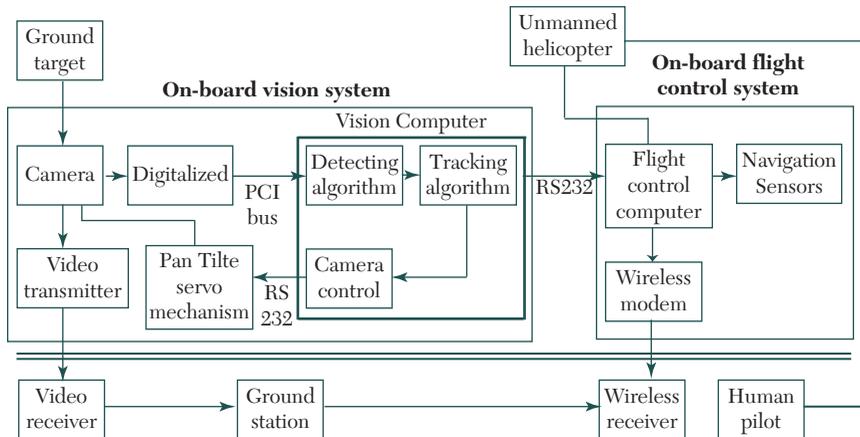


FIGURE 10.9. The configuration of an overall vision system.

A. Visual Sensor: Video Camera

A visual sensor is employed on-board to obtain in-flight visual information of the surrounding environment of the UAV. Interested visual information is composed of silent and dynamic features, such as the color and shape of land marks, and motions of vehicles. A color video camera is selected as the on-board visual sensor, which has a compact size and a weight less than 30g and 40-degree field of view.

B. Image Acquisition Module: Frame Grabber

The primary function of a frame grabber is to perform the A/D conversion of the analog video signals and then output the digitalized data to a host computer for further processing. PC/104 standard frame grabber, a Colory104, has the following features: (1) high resolution: Colory 104 is capable of providing a resolution up to 720×576 (pixels), which is sufficient for on-line processing; (2) multiple video inputs: it is able to collect data from multiple cameras; (3) sufficient processing rate; (4) and featured processing method: two tasks are used alternatively to convert the digital video signal into specified formats.

C. Vision Processing Module: Vision Computer

As shown in Figure 10.9, the digitalized visual signals provided by the frame grabber is transferred to the on-board vision computer that is the key unit of the vision system. The vision computer coordinates the overall vision system, such as image processing, target tracking, and communicating with the flight-control computer. Two separated embedded computers in the on-board system for UAVs are used. One is for flight control, and another one is for machine vision algorithms. Such a configuration for on-board system helpful because of the following reasons: (1) the computation consumption of flight-control task and vision program are very heavy, which can hardly be carried out together in a single embedded computer; (2) the sampling rate of the flight-control computer is faster than the vision computer, since the faster sampling rate is required to stabilize the unmanned rotorcraft; (3) the decoupled structure reduces the negative effect of data blocking caused by the vision program and flight-control system, and thus makes the overall system more reliable. Separated on-board PC104 embedded computer, Cool RoadRunner III, is employed to process the digitalized video signal and execute the vision algorithms. The core of the board is an Intel LV Pentium- III processor running at 933 MHz. A compact flash memory card is used to save the captured images.

D. Pan/Tilt Servo Mechanism

In the application of the ground target following, it is required to keep the target objects in the field of view of the camera to increase the flexibility of vision based tracking. As such, mounted the camera on a pan/tilt servo mechanism that can rotate in the horizontal and vertical directions.

E. Wireless Data Link and Video Link

In order to provide ground operators with clear visualization to monitor the work that the on-board vision is processing during flight tests, the video captured by the on-board camera is transmitted and displayed in a ground control station. An airborne 2.4 GHz wireless video link is used to transmit the live video captured to the ground control station.

Configuration of the vision software system

The purpose of the vision software system is to coordinate the work of on-board devices and implement vision algorithms. Since the vision software system targets for real-time applications and runs in an embedded PC104 computer, QNX Neutrino, a real-time embedded operating system,

is employed as the developing platform. QNX Neutrino has a micro kernel that requires fewer system resources, and performs more reliably and efficiently for embedded systems during runtime compared to the traditional monolithic kernel.

The vision software program coordinates tasks such as capturing video, controlling pan/tilt servo mechanism, as well as performing the vision detecting and tracking algorithms. To make the vision software system easy to design and robust to perform, the entire vision software system is divided into several main blocks. Each block is assigned a special task.

1. Reading RGB image from the buffers assigned to the frame grabber. The reading rate is set. In order to reduce the risk of damaging the image data, two buffers are used to store the captured images by the frame grabber alternatively.
2. Processing the captured images, carrying out the vision algorithms, such as the automatic tracking and camera control.
3. Controlling the rotation of the pan/tilt servo mechanism to keep the ground target in a certain location of the image.
4. Saving the captured and processed images to a high-speed compact flash.
5. Communicating with the flight-control computer. The flight-control computer sends the states of the UAV and commands from the ground station to the vision computer, and the vision computer sends the estimated relative distance between the UAV and the ground target to the flight-control computer to guide the flight of the UAV.
6. Providing a mean for users to control the vision program such as running and stopping the tracking as well as changing the parameters of the vision algorithms.
7. Managing and scheduling the work of the entire vision software system.

Coordinate frames used in vision systems

Coordinate systems adopted in the UAV vision systems are shown in Figure 10.10.

1. The local north-east-down (NED) coordinate system (labeled with a subscript “n”) is an orthogonal frame on the surface of the earth, whose origin is the launching point of the aircraft on the surface of the Earth.

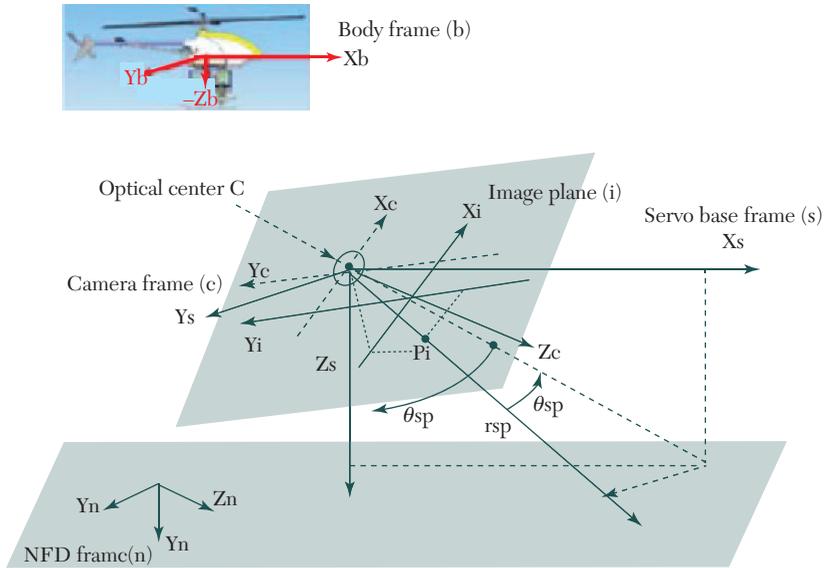


FIGURE 10.10. Coordinate frames used in unmanned vision system.

2. The body coordinate system (labeled with a subscript “b”) is aligned with the shape of the fuselage (main body) of the aircraft.
3. The servo base-coordinate system (labeled with a subscript “s”) is attached to the base of the pan/tilt servo mechanism, which is aligned with the body coordinate system of the UAV.
4. The spherical coordinate system (labeled with a subscript “sp”) is also attached to the base of the pan/tilt servo mechanism. It is used to define the orientation of the camera and the target with respect to the UAV.
5. The camera coordinate system (labeled with a subscript “c”), whose origin is the optical center of the camera. The Z_c -axis is aligned with the optical axis of the camera and points from the optical center C towards the image plane.
6. The image frame (or the principle image coordinate system) (appended with a subscript “i”) has the origin at the principal point. The coordinate axis, X_i and Y_i are aligned with the camera coordinate axis, X_c and Y_c , respectively.

Vision-based ground target following

The vision-based ground target detection many vision approaches are used worldwide, such as template matching, background subtraction, optical flow, stereo-vision-based technologies, and feature-based approaches. A sophisticated vision-based target detection and tracking scheme is illustrated in Figure 10.11, and employs robust feature descriptors and efficient image-tracking techniques. Based on the vision-sensing data and navigation sensors, the relative distance to the target is estimated. Such estimation is integrated with the flight-control system to guide the UAV to follow the ground target in flight.

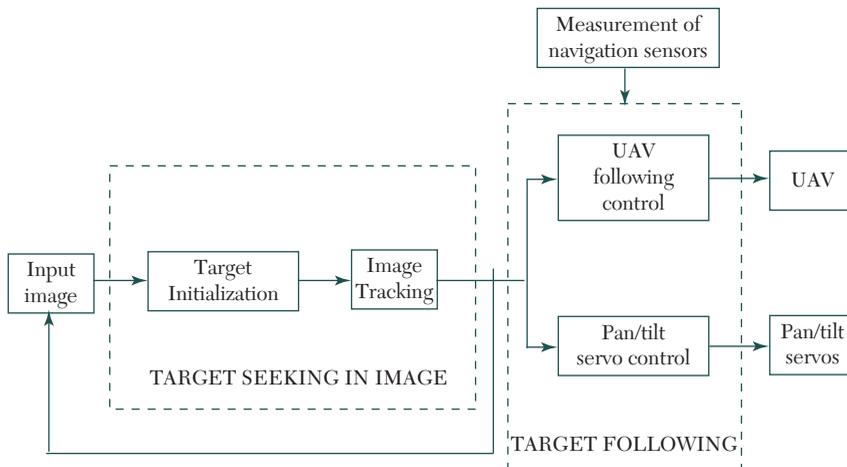


FIGURE 10.11. Flow chart of the ground-target detection, tracking, and following.

A. Target Detection

The purpose of the target detection is to identify the target of interest from the image automatically based on a database of preselected targets. A car is chosen as the ground target. A classical pattern recognition procedure is used to identify the target automatically, which includes three main steps: They are segmentation, feature extraction, and pattern recognition.

1. Segmentation: The segmentation step aims to separate the objects of interest from background. To simplify further processing, some assumptions are made. First, the target and environments exhibit Lambertian reflectance, and in other words, their brightness is unchanged regardless of viewing directions. Second, the target has a distinct color distribution compared to the surrounding environments.

Step 1: Threshold in color space. To make the surface color of the target constant and stable under the varying lighting condition, the color image is represented in HSV space, which stands for hue, saturation, and value 3 channels. Precalculated threshold ranges are applied these three channels. Only the pixel values falling in these color ranges are described as the foreground points, and pixels of the image that fall out of the specified color range are removed.

Step 2: Morphological operation. Normally, the segmented image is not smooth and has many noise points. Morphological operations are then employed to filter out noise, fuse narrow breaks and gulfs, eliminate small holes, and fill gaps in the contours. Next, a contour-detection approach is used to obtain the complete boundary of the objects in the image, which will be used in the feature extraction.

2. **Feature Extraction:** Generally, multiple objects will be found in the segmented images, including the true target and false objects. The geometric and color features are used as the descriptors to identify the true target.

Geometry feature extraction: To describe the geometric features of the objects, the four lowest moment invariants are employed, since they are independent of position, size, and orientation in the visual field. It can be easily proven that compactness is invariant with respect to translation, scaling, and rotation.

Color feature extraction: To make the target detection and tracking more robust, a color histogram is employed to represent the color distribution of image area of the target, which is not only independent of the target orientation, position, and size, but also robust to partial occlusion of the target and easy to implement. Due to the stability in outdoor environments, only hue and value are employed to construct the color histogram for object recognition.

Dynamic features: Besides the static features extracted from the foreground objects, further their dynamic motion using the Kalman filtering technique is calculated. The distance between the location of each object z_i and the predicted location of the target z is employed as a dynamic feature. Both the static and dynamic features of them are then employed in the pattern recognition. The extracted features of an object need to be arranged in a compact and identifiable form. A straightforward way is to convert these features in a high-dimension vector.

3. Pattern Recognition: The purpose of the pattern recognition is to identify the target from the extracted foreground objects in terms of the extracted features. The straightforward classifier is to use the nearest neighbor rule. It calculates a metric or “distance” between an object and a template in a feature space, and assign the object to the class with the highest score. But to take advantage of a priori knowledge of the feature distribution, the classification problem is formulated under the model-based framework, and solved by using a probabilistic classifier. A discriminant function, derived from Bayes theorem, is employed to identify the target. This function is computed based on the measured feature values of each object and the known distribution of features obtained from training data.

Step 1. Prefilter: Before classifying the objects, a pre-filter is carried out to remove the objects whose feature values are outside certain regions determined by a priori knowledge. This step aims to improve the robustness of the pattern recognition and speed up the calculation.

Step 2. Discriminant function: The discriminant function, derived from Bayes theorem is used to determine the target based on the measured feature values of each object and the known distribution of features of the target obtained from training data.

B. Image Tracking

As shown in Figure 10.11, after initialization, the image-tracking techniques are employed. The purpose of image tracking is to find the corresponding region or point to the given target. Unlike the detection, the entire image search is not required. Thus, the processing speed of image tracking is faster than the detection. The image-tracking problem can be solved by using two main approaches, (1) filtering and data association, and (2) target representation and localization.

Filtering and Data Association

The filtering and data association approach can be considered as a top-down process. The purpose of the filtering is to estimate the states of the target, such as static appearance and location. Typically, the state estimation is achieved by using filtering technologies. It is known that most of tracking algorithms are model based because a good model-based tracking algorithm will greatly outperform any model-free tracking algorithm if the underlying model is found to be a good one. If the measurement noise satisfied the Gaussian distribution, the optimal solution can be achieved by the Kalman

filtering technique. In some more general cases, particle filters are more suitable and robust. However, the computational cost increases and the sample degeneracy is also a problem. When multiple targets are tracked in the image sequence, the validation and association of the measurements become a critical issue. The association techniques, such as probabilistic data association filter (PDAF) and joint probabilistic data association filter (JPDAF) are widely used.

Target Representation and Localization

Besides using the motion prediction to find the corresponding region or point, target representation and localization is considered another efficient bottom-up approach. Among the searching methods, the mean-shift approach using the density gradient is commonly used, and tries to search the peak value of the object probability density. However, the efficiency will be limited when the spatial movement of the target becomes significant. To take advantage of the aforementioned approaches, using multiple trackers are widely adopted in applications of image tracking. In the tracking scheme motion, color, and geometric features are integrated to realize robust image tracking.

Instead of using multiple trackers simultaneously, a hierarchical tracking scheme is used to balance the computational cost and performance, as illustrated in Figure 10.12. In the model-based image tracking, the Kalman filtering technique is employed to provide accurate estimation and prediction of the position and velocity of a single target, referred to as dynamic information. If the model-based tracker fails to find the target, a mean shift based image tracking method will be activated to retrieve the target back in the image.

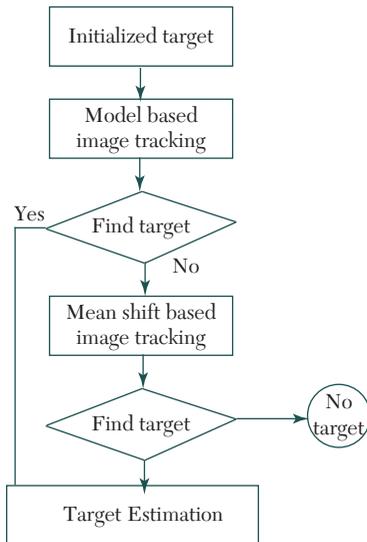


FIGURE 10.12. Flow chart of image tracking.

1. Model-Based Image Tracking

Model-based image tracking will predict the possible location of the target in the subsequent frames, and then do the data association based on an updated likelihood function. The advantage of the model-based image tracking is to combine dynamic features with geometric features of the target in the

image tracking under noise and occlusion condition. In addition, several methods are employed to make the tracking more robust and efficient, which are given by:

- Narrowing the search window in terms of the prediction of the Kalman filter; and
- Integrating the spatial information with appearance and setting the different weightings for the discriminant function.

Most of time, the model-based tracker can lock the target in the image sequence, but sometime it may fail due to the noise or disturbance, such as partial occlusion. Thus, a scheme is required to check whether the target is still in the image, and then activate other trackers.

2. Switching Mechanism

The purpose of the switching mechanism is to check whether the target is still in the image when the target is lost by the model-based tracker. If it is, the mean-shift tracker will be activated. The loss of the target can be attributed to the poor match of features due to noise, distortion, or occlusion in the image. An alternative reason may be the maneuvering motion of the target, and the target is out of the image. Therefore, in order to know the reason and take the special way to find target again, it is necessary to formulate the decision making as the following hypothesis testing problem: H_0 : The target is still in the image. Or H_1 : The target is not in the image due to maneuvers. The estimation error is considered as a random variable. If H_0 is true, the Chi-square testing-based switching declares the target is still in the image and enables the mean-shift based tracker.

3. Mean-Shift-Based Image Tracking

If the target is still in the image, a continuously adaptive mean shift (CAMSHIFT) algorithm is employed, as shown in Figure 10.12. This algorithm uses the mean-shift searching method to efficiently obtain the optimal location of the target in the search window. The principle idea is to search the dominated peak in the feature space based on the previous information and certain assumptions. The detected target is verified by comparing with an adaptive target template. The CAMSHIFT algorithm consists of three main steps: back projection, mean-shift searching, and search-window adaptation.

Step 1. Back projection: In order to search the target in the image, the probability distribution image needs to be constructed based on the color distribution of the target. The color distribution of the target is defined in hue channel. Based on the color model of the target, the back projection algorithm is employed to convert the color image to the color probability distribution image. The probability of each pixel in the region of interest is calculated based on the model of the target, which is used to map the histogram results.

Step 2. Mean shift algorithm: Based on the obtained color-density image, a robust nonparametric method, the mean-shift algorithm, is used to search the dominated peak in the feature space. The mean-shift algorithm is an elegant way of identifying these locations without estimating the underlying probability density function.

Step 3. Search window adaptation: The region of interest is calculated dynamically using motion filtering. To improve the performance of the CAMSHIFT algorithm, multiple search windows in the region of interest are employed. The initial locations and sizes of the searching windows are adopted from the centers and boundaries of the foreground objects respectively. These foreground objects are obtained using the color segmentation in the region of interest. In the CAMSHIFT algorithm, the size of the search window will be dynamically updated according to the moments of the region inside the search window. Generally, more than one target candidate will be detected due to multiple search windows adopted. To identify the true target, the similarity between the target model and the detected target candidate is measured using the intersection comparison. This verification can effectively reduce the risk of detecting the false target.

C. Target-Following Control

A target-following system consists of two main layers, the pan/tilt servo-mechanism control and the UAV-following control. The overall structure of the target-following control is depicted in Figure 10.13. A pan/tilt-servo mechanism is employed in the first layer to control the orientation of the camera to keep the target in an optimal location in the image plane, namely eye-in-hand visual serving, which makes target tracking in the video sequence more robust and efficient. In the second layer, the UAV is controlled to maintain a constant relative distance between the moving target and the UAV inflight.

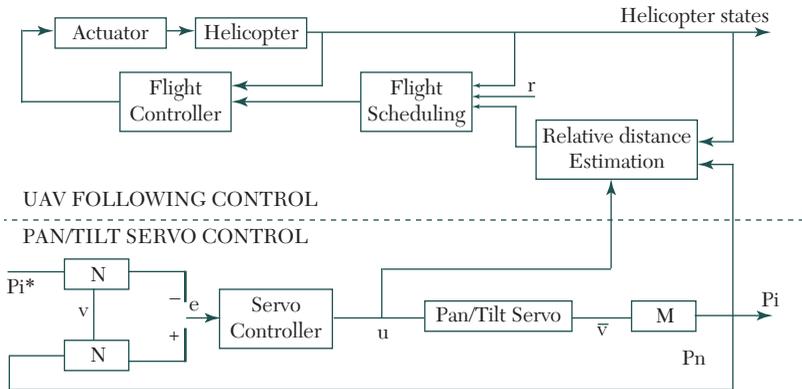


FIGURE 10.13. Block diagram of the tracking-control scheme.

6. Automatic Axle-Lifting System Design

An automatic-lift axle-lifting/dropping system aims to automate the dropping action. It aims to lower damages to the truck, as well as damages to road pavement due to loading conditions. The system also helps in increasing traction when the road surface is wet or icy. By automating the system, the task of deciding when to lift or drop the axles will no longer be a burden to the driver. An electronic control unit to actuate solenoid valves is introduced. All environmental components, for example, switches, lift axle solenoid valves, and sensors, are connected to this control unit and axles are actuated by the algorithm running inside it. The automatic axle-lifting system is shown in Figure 10.14. It can be linked with automatic driving and lane-detection vision systems.

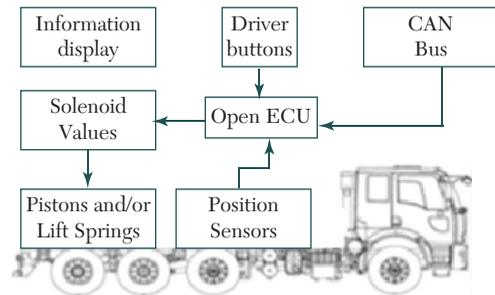


FIGURE 10.14. Automatic axle-lifting system.

System requirements and algorithm

The main requirement for the auto-drop system is the measurement of axle loads. The system must drop the appropriate axle(s) automatically based on the loading condition. If the second axle is lifted and if the rear axle-group load exceeds 18.0 tons, the system will drop the nearest axle, which is second axle, in order to share overload and maintain axle loads under permissible limits and does not allow driver to lift axles again. Other

conditions that do not involve axle loads are also included in the regulation in addition to the conditions given above. For instance, once the ignition is off, all axles should be lowered. Speed limit is taken into account for functional requirements to provide safety driving by disabling any driver input above 30 kph. Additionally, in order to improve emergency brake performance when parking brake is engaged tag axle (4th axle) should be dropped or not allowed to be lifted by driver since wheels should be in contact with the road surface.

There are two liftable axles (the 2nd and 4th axles). The algorithm starts with start button. It checks whether the ignition is turned on or not, as well as the axle positions. If the vehicle is not running, it will automatically drop axle 2. If the vehicle is running, it will check the rear axle load (RAL) value against set weight value like 18 tons. This information is used to determine the state of axle 2. If the RAL is lower than 11.5 tons, then axle 4 will be lifted.

The system considers axle load limits, checks ignition and parking brake conditions. It provides traction assistance when needed and consider speed limits as needed. A supervisory controller is used for implementation of these functions. The lift axle system has two user switches (tag axle and self-steer) in order to send lift/drop request to the ECU algorithm. A cluster screen has been placed on the driver information screen to provide relevant information to the driver. The driver can get information about axle states, request refusals (if any), automatic actions, and failures.

7. Object Tracking Using an Address Event Vision Sensor

Vehicle tracking systems are based on video cameras. In contrast to traditional CCD or CMOS imagers that encode image irradiance and produce constant data volume at a fixed-frame rate, irrespective of scene activity, the asynchronous-address event vision sensor contains an array of autonomous, self-signaling pixels which individually respond in real time to relative changes in light intensity by placing their address on an asynchronous arbitrated bus. Pixels that are not stimulated by a change in illumination are not triggered; hence static scenes produce no output.

Because there is no pixel readout clock, no time quantization takes place at this point. The sensor operates largely independent of scene illumination, directly encodes object reflectance, and greatly reduces redundancy while preserving precise timing information. Because output bandwidth is automatically dedicated to dynamic parts of the scene, a

robust detection of fast-moving vehicles at variable lighting conditions is achieved. The scene information is transmitted event-by-event to a DSP via an asynchronous bus. The pixel location in the imager array are encoded in the event data that are reflected as i, j coordinates in the resulting image space in the form of address-events (AE). An effective way of processing AE data takes advantage of the efficient coding of the visual information by directly processing the spatial and temporal information contained in the data stream.

The high dynamic range of the photosensitive element ($>120\text{dB}$ or 6 decades) makes the imager ideal for applications with uncontrolled light conditions. Figure 10.15 (a) depicts the general architecture of the concerned embedded sensory system, which comprises an imager, a first-in, first-out (FIFO) buffer memory and the Blackfin DSP BF537 from analog device. The location (address) of the event generating pixels within the array is transmitted to a FIFO on a 16-bit parallel bus, implementing a simple 4-phase handshake protocol. The FIFO is placed between the imager sensors and the DSP to cope with peaks of AE activity. In the processing stage, every AE received by the DSP is labeled by attaching the processor clock ticks with 1ms precision as a timestamp. These data are the basis for the vehicle tracking.

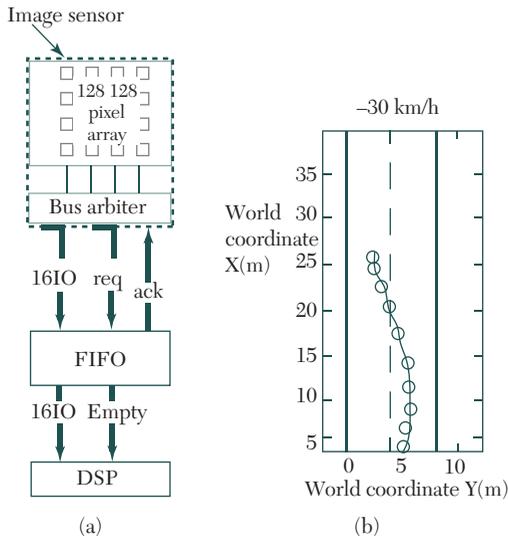


FIGURE 10.15. (a) Embedded system architecture. (b) Vehicle tracks.

AE processing algorithms for the vehicle speed estimation have also been implemented on this DSP. The full processing comprises the AE acquisition and time stamping, clustering and tracking including rough speed estimation.

The mean-shift approach implements a continuous clustering of AEs and tracking of clusters. This algorithm processes each AE as it is received without data buffering (no memory consumption). This is of special importance when using low-cost and low-memory resource systems. Each new event can be assigned to a cluster based on a distance criterion and is used to update this clusters weight and (x,y) —center position for tracking.

The algorithm can be briefly described as follows:

- a. Find one cluster in the cluster list that the new AE with address $x_E=(i,j)$ lies within a seek-distance R_k of its center
- b. If a cluster is found where $R < RK$, update all the clusters features accordingly.
- c. If no cluster is found, create a new one with center x_E and initialize weights and size, creation time and velocity. A new label (unique identification number) is assigned to the cluster.

Very inactive clusters have low AE frequency and consequently low weights. The list of existing clusters is scanned periodically (10 to 20 times/s) for old and inactive clusters which are then deleted from the list. The velocity vector of the cluster is also updated at this occasion. Figure 10.15 (b) depicts examples of vehicle tracks observed on a two-lane road. Imager coordinates have been transformed to world coordinates using a simple geometric projection based on the imager mounting height and optical parameters. The x-coordinate shows the road length in meters (containing the vehicle direction), while the y-coordinate gives of the road width in meters. The distance between two adjacent circles on the vehicle track is 0.2 seconds. Therefore, the traces with closer circles' distance reflect the low-speed vehicle compared to the longer distances between circles.

8. Using FPGA as an SoC Processor in ADAS Design

FPGA technology can address these challenges in key automatic driver assistive system (ADAS) applications. The FPGA is used as a main SoC processor or coprocessor for algorithm acceleration or sensor-interface translation. Table 10.1 shows FPGA functions in ADAS applications.

TABLE 10.1. FPGA functions in ADAS applications.

ADAS Application	FPGA Function
Lane-Departure Warning	Camera-sensor interface and lane-marking detection
Traffic-Sign Recognition	Camera-sensor interface and image processing for sign recognition
Night Vision	Infrared or thermal-camera-sensor interface and image-processing pipeline
Intelligent Headlight Control	Camera-sensor interface, headlight detection algorithm, and controller area network (CAN) bus interface
Adaptive Cruise Control	Radar-sensor interface and CAN-bus interface
Collision Avoidance	Camera- and radar-sensor interface, radar processing, image processing and recognition, and CAN-bus interface
Surround Vision	Image stitching, image analytics, video-data transmission, and fisheye correction (pincushion barrel distortion)
Blind-Spot Warning	Radar-sensor interface and CAN-bus interface
Park Aid	Ultrasonic and camera-sensor interface, parking algorithm, fisheye correction, and CAN-bus interface

Forward camera systems involve high-speed video processing, complex sensor fusion, and real-time data analysis that enable the automobile's corrective action. To achieve this high-level functionality with both mono- and stereo-camera systems, one may also integrate additional sensor types such as radar and laser sensors. Each sensor type is unique in how it provides data, making it a challenge to design for multiple architectures. Traditional DSP processors or microcontrollers do not have enough power for real-time video processing and analytics and automotive-system design needs additional hardware coprocessors to keep up with these real-time processing requirements.

FPGA integration

Instead of using DSP processors or microcontrollers, one can integrate the entire automotive vision system into a single, low-cost FPGA. For optimal system performance, one can develop the hardware parallel processing engines with FPGAs as well as integrate software algorithms running on SoC's hard-processor system. In addition, FPGAs allow for customized I/O interfaces to various image sensors, along with output interfaces, such as

CAN, LVDS, or Ethernet communications. The flexible nature of an FPGA gives several implementation options. Altera's Cyclone V SoC FPGAs to integrate the radar, video, and sensor fusion processing algorithms along with network connectivity in a single device is as shown in Figure 10.16.

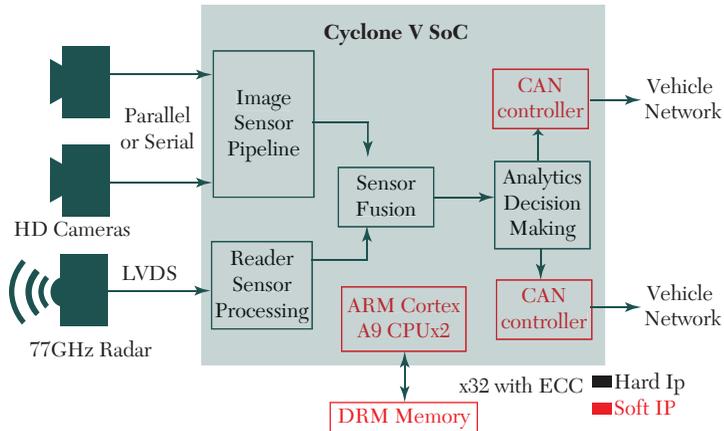


FIGURE 10.16. SoC FPGA.

Forward-looking Sensor-Fusion ECU

Surround-view or 360° car-view cameras in automotive systems allow drivers to see a 360° image surrounding the vehicle and are especially useful in slow-moving parking applications. Surround-view processors require many I/Os to allow connectivity of four or more camera sensors, real time stitching of images, and perspective correction that enables a normal looking view outside a vehicle. Many manufacturers want to place custom camera sensors in different locations around the vehicles, process different sensor

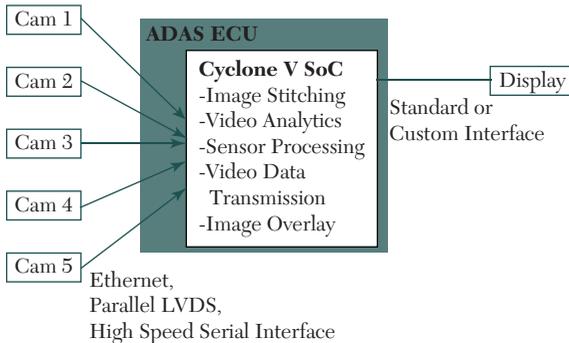


FIGURE 10.17. Surround-view camera block diagram.

resolutions, and output an image onto various size displays in the vehicle as given in Figure 10.17. FPGAs are a cost-effective solution, providing the flexibility to integrate data from multiple sensors and processing power to perform real-time image stitching for high-definition (HD) video streams at 30 frames per second (fps) and above.

Making a smarter rear-view camera

Rear-view cameras or back-up camera systems provide a fisheye distortion-corrected image on the display of a head unit. Next-generation automotive-vision systems will automatically identify threatening objects and actively control the automobile to avoid accidents. This increased functionality requires more computational power from video-processing engines. Traditional digital-signal-processing (DSP) processors or microcontrollers do not have enough power for real-time video processing and analytics as the resolution of the images increases to megapixel sensors.

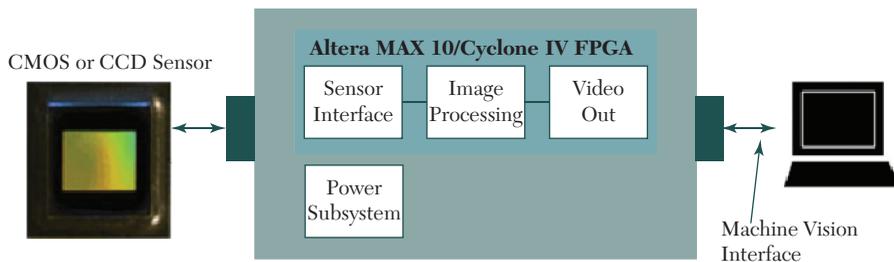


FIGURE 10.18. Flexibility, FPGAs support different sensor and MV interfaces.

GigE vision provides an open, high-performance, scalable framework for image streaming and device control over Ethernet networks. This interface standard provides an environment for networked machine-vision systems based on switched client/server architectures, allowing one to connect multiple cameras to multiple computers as shown in Figure 10.18.

Lane and vehicle detection method for lane-change-assistant systems

According to the WHO, each year lives of approximately 1.25 million people cost as result of road traffic accidents. Between 20 and 50 million people suffer from nonfatal injuries, which sometimes incur disabilities. Road traffic injuries bring considerable economic losses to victims, their families, and nations as a whole. Therefore, in 2016 many firms and corporations declared that they were willing to participate in the development of the automatic vehicle. Volvo Corporation has promised that by 2020, nobody

will face a serious accident involving one of its new cars when using the driving-assistance system and warning.

Intelligent vehicle technologies utilize some kind of sensor such as Lidar, radar, and vision sensors, with Lidar and radar only used for obstacle detection. Unique vision sensors are used for lane detection and vehicle detection. Detecting lane markings and vehicles enables vehicles to evade collisions and support a warning system.

Feature information, a model of the lane markings, and color information are three main methods included in lane detection. Feature information for lane detection includes edge, gradient, and intensity. These features rely on the different intensity between the road surface and the lane markings. The edge information and Hough transform find straight lines, which can be the lane markings.

The second approach uses the road information to make the mathematical model of the lane markings. The B-splines uses a set of candidate points that must be extracted from the lane markings. For the color information, this approach usually converts RGB to HSI or custom-color spaces, or color features.

Using the driver-assistance system, first detect lanes, then detect vehicles based on the lane and vehicle features to support the warning system. This idea consists of two main steps: The information lane will be detected in the first step. And then, the vehicle will be detected inside an area among the detected lanes by vehicle features. Figure 10.19 (a) represents the scheme outline of proposed system. Region of interest (ROI) is defined as an area close to the test vehicle. The full image is used to detect vehicles. There are two phases: (1) The lane detection on the ROI, and (2) The vehicles

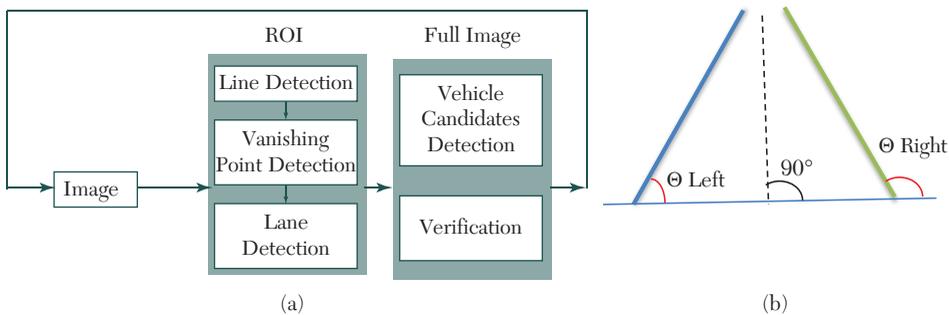


FIGURE 10.19. (a) The Lane, Vehicle detection algorithm (b) Relevant line detection

detection based on the detected lanes on the full image. Kalman filter is used to track vehicle information with error cases or inaccuracy cases.

Lane detection—Line estimation method

The first step of the approach is to detect the line segments from the image inputs. The edge-drawing lines algorithm is used to detect lanes. For each image, the edge-drawing (ED) lines algorithm only takes from 10 ms to 20 ms with an Intel 2.2 GHz CPU for the line detection, which is much faster than other algorithms such as the line segment detector (LSD), without the need for any further processing. Moreover, the ED lines algorithm runs even faster when it is applied to the ROI. For any image, the ROI are defined as the rectangle box. The line segment is detected by the ED lines. After the line segments are extracted from the images, two exclusive steps are performed to remove the irrelevant line segments. First, all of the vertical line segments and horizontal line segments are removed. Next it rejects the irrelevant remaining line segments and keeps the relevant ones—the relevant line segments are the lane markings. The line segments are divided into two subsets, left and right candidate sets. The angles between all of these segments and the horizontal axis are calculated as shown in Figure 10.19 (b), in which θ_{left} is the angle between the left segments and the horizontal axis and θ_{right} is the angle between the right segments and the horizontal axis. A set range of limits for each angle are proposed: 30° to 85° for θ_{left} and 120° to 175° for θ_{right} . It means that any segment that does not belong to the ranges will be removed.

Two clusters of the relevant segments, the left and the right, are available. The vanishing point is the intersection point between the lane markings or where the road boundaries cross. Therefore, in order to define the vanishing point, all line segments on each cluster are extended and each line of the left cluster meets the lines of the right cluster at the intersection points. We create a grid formed by square cells of size equal to the permitted error in estimating the vanishing point position. Finally the cell that has the maximum number of the intersection points is selected and its center weight is returned as an approximation of the vanishing point position.

Lane detection

To detect lane markings the horizontal straight line is created. It intersects with the extended segment at points, and these points are circled. Based on the distances among these points, the close points are clustered in a group. Similarly, the lane-marking detection process for rear-side view

is similar to the lane-marking detection process for the frontal view. It includes steps such as the line detection, the vanishing-point estimation, and the lane-marking detection.

Vehicle detection and monitoring vehicle candidate detection

Vehicle detection is the main work relied on by the detected lane markings from the above section. The original image is filtered by horizontal edge. Then, the received image is converted to binary using the Otsu's thresholding (clustering-based image thresholding). After having the binary image, the image is divided into lane areas based on the detected lane markings. Information of the lane areas is used to detect the vehicles.

9. Diagnostic Imaging

As shown in Figure 10.20, a typical diagnostic imaging system consists of three sets of cards: data acquisition, data consolidation, and image-/data-processing cards.

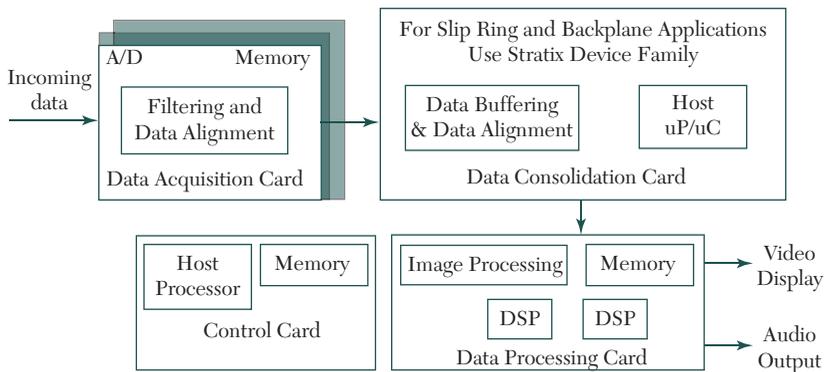


FIGURE 10.20. Example of diagnostic imaging equipment.

The data-acquisition card, which filters incoming data, is the most cost-sensitive system card. Usually a diagnostic imaging system will consist of multiple data-acquisition cards (in some cases, up to 20 cards per system). Once the data is compensated and filtered, it is sent to the data consolidation card for buffering and data alignment. For computer tomography and positron emission tomography scanners where the detectors rotate around the body, the data is serialized and sent across a slip-ring electromechanical subassembly. Once the data has been collected, it is sent to the image-/data-processing cards. These cards perform heavy-duty filtering and the most algorithm-intensive image reconstruction. Once completed the final

imaging and scaling functions for display are usually done on a single-board computer (SBC). There are several variables that one needs to consider before making component selections for the acquisition and processing cards. For example, depending on the number of channels per system and resolution required, one could choose:

1. Off-the-shelf analog components or integrate the analog functionality into an ASIC
2. Two-dimensional (2D) or three-dimensional (3D) imaging
3. To partition image processing between the processing cards and the SBC
4. Intel FPGA SDK for OpenCL for algorithmic acceleration

10. Electronic Pill

The human body is a sensitive system, and sometimes doctors are unable to detect a disease in time and it becomes too late to cure it. Use of electronic pills helps to easily detect diseases, and this can help take prompt action against them. Electronic pill technology makes use of different components/parts such as drug reservoir, delivery pump, electronic microcontroller (MCU), wireless communication, cameras, and sensors. These elements have to be combined in a way so as to preserve small size, reliable manufacturing, and a safety profile fit for medical use. The device containing these parts is built as a small, pill-shaped capsule, which is swallowed and passed through the gastro-intestinal tract. Use of an electronic pill will free users from invasive methods such as catheters, endoscopic instruments, or radioisotopes for collecting information about the digestive tract. Drug delivery using an electronic pill will also be controlled with on-board electronics, enabling precise and adaptable delivery patterns, which are not yet possible by other means.

An electronic pill has multichannel sensors that will prove to be an important tool for healthcare technology towards in-depth and detailed investigation of diseases. In addition, its uses range from drug delivery to reaching specific regions of the human body to target different types of cancer, stimulate damaged tissues, and track gastric problems and measure biomarkers. To carry out these functions, the pill is powered by an edible battery and equipped with appropriate sensors. It is important to assure that the materials used to make an edible battery are not toxic to humans, as this can cause significant complications if it gets into the digestive tract.

An electronic pill contains sensors or tiny cameras that collect information as it travels through the gastro-intestinal tract before being excreted from the body a day or two later. The capsule takes measurements of the local pH and temperature inside the body. This electronic pill transmits information such as acidity, pressure, and temperature levels, or images of the esophagus and intestines to the doctor's computer for analysis. Electronic pills are also being used to measure muscle contraction, ease of passage and other factors of the body to reveal information that was unavailable in the past.

The electronic pill is a medical monitoring system, measuring parameters like temperature, pH, conductivity, and dissolved oxygen, and can also capture images and send it to a system. An electronic pill has a 16mm diameter, 55mm length and 5gm weight, and can be swallowed. It is covered by a chemically-resistant polyether-terketone (PEEK) coating. As soon as the pill moves through the gastrointestinal track, it starts to detect diseases and abnormalities. The pill can easily reach areas such as small and large intestines and deliver real time information to an external system. Data collected is then displayed on a monitor.

Specifications of the electronic pill

An electronic pill consists of four microelectronic sensors. The first sensor consists of a silicon diode attached to the substrate, fabricated on two silicon chips located at the front end of the capsule. It is used to identify body temperature. Use of a silicon-integrated circuit makes this sensor useful, and it comes at a very low cost. The second is ion-sensitive field effect transistor (ISFET), which is used for measuring ion concentration in solutions. The third is direct-contact gold electrode (DCGE), which helps measure conductivity. Conductivity is measured by determining the content of water and salt absorption, and the breakdown of organic compounds into charged colloids and bile secretion. Three electrode electrochemical cell (TEEC) is the fourth sensor. IT is used to calculate the rate of dissolved oxygen, and identify the activity of aerobic bacteria in the small and large intestines.

All these sensors are controlled by an application specific integrated circuit (ASIC). All other components of the electronic pill, namely, tiny camera, 10-bit analogue-to-digital converter (ADC), digital-to-analogue converter (DAC), relaxation oscillator circuit (OSC) and digital signal processing circuit are connected to the ASIC consisting of analogue signal conditioning. The circuit is powered by two SR48 silver-oxide (Ag_2O) batteries having 35 hours working capacity. Supply voltage is about 3.1V with power consumption of 15.5mW. The pH and oxygen sensors are

enclosed in separate 8nL electrolyte chambers containing a 0.1 KOH solution retained in a 0.2 percent calcium alginate gel. The two sensors are covered by a $12\mu\text{m}$ -thick film made of Teflon and Nafion, respectively. These are protected by a $15\mu\text{m}$ thick dialysis membrane of polycarbonate. All data is collected by the ASIC. A radio transmitter transmits data to the base station for the doctor to identify the problem.

Types of electronic pills

An electronic pill can be classified into two types: one including a camera that collects data from disease infected areas and sends it to the system, and the other containing only sensors to measure pH level, temperature, and oxygen level and so on.

1. Intelligent pill

This is basically a plastic capsule usually taken with solid food or water. It is meant to be transported through the digestive system in a natural manner. Then, the drug is dispensed to different parts of the body. The size of the device is similar to that of fat multivitamin and the drug can carries out specialized actions.

2. IntelliCap drug

This electronic pill acts as a drug delivery and monitoring device. It is made of a drug reservoir, wireless communication system, electronic controllers, sensors, and a delivery pump. The device has a minuscule form and, upon ingestion it travels through the gastrointestinal tract. Use of on-board electronics makes drug delivery precise and flexible. The electronic pill network is shown in Figure 10.21.

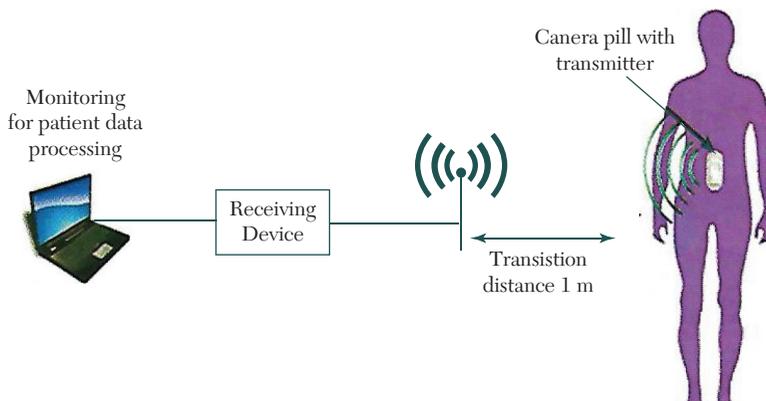


FIGURE 10.21. Electronic pill network.

10.4 RESEARCH AND DEVELOPMENT IN VISION SYSTEMS

This section gives an overview of research undergone in embedded-vision techniques, applications, and methods for the decade. It will give idea for embedded-vision development.

Robotic Vision

Most advanced humanoid robots can tramp along flat and inclined surfaces, climb up and down stairs, and slog through rough terrain. Some can even jump. But despite the progress, legged robots still can't begin to match the agility, efficiency, and robustness of humans and animals. Existing walking robots hog power and spend too much time in the shop. All too often, they fail, they fall and they break. For the robotic helpers we have long dreamed of to become a reality, these machines will have to learn to walk as we do. We must build robots with legs because our world is designed for legs. We step through narrow spaces, we navigate around obstacles, we go up and down steps, Robots on wheels or tracks can't easily move around the spaces we've optimized for our own bodies. Developing human-like robots is an important area of research.

A mobile robot that can detect and catch the target object by using the CCD camera with an embedded system has been developed. The image-processing techniques used were histogram, image spatial resolution and connectivity (neighboring pixels). It is practical to use these techniques with the embedded system for detecting the target object. Moreover, they can be applied to the embedded-system mobile robots to perform small tasks with high flexibility, low cost, ease of construction, and without using a lot of energy.

The implementation of image-processing algorithms over a friendly ARM-embedded system, allows a mobile robot to move around in an autonomous way within an area, where both static and dynamic obstacles are present. Given that the robot has a vision system, this is capable of increase or decreases its speed when it faces a mobile obstacle or simply changes its direction when facing a static obstacle.

Vision-based sensors are a key component for robot systems where many tasks depend on image data. Real-time constraints of the control tasks bind a lot of processing power only for a single-sensor modality. Dedicated

and distributed processing resources are the “natural” solution to overcome this limitation.

Sometimes algorithm and hardware architecture for an embedded-stereo vision system appropriate for robotics applications were designed. Three reprogrammable processor units deemed preprocessing, postprocessing and stereo matching were processed.

The miniature vision module combines an analog VLSI (aVLSI) vision sensor with a digital postprocessor (MPC555). The aVLSI sensor provides grayscale image data as well as smooth optical flow estimates. The particular computational architecture (analog and parallel) of the sensor allows efficient real-time estimation of the smooth optical flow field. The MPC555 controls the sensor read-out and, furthermore, allows for additional higher-level processing of the image and optical flow data. It also provides the necessary standard interface such that the module can be easily programmed and integrated into different robotic platforms. The vision module seems particularly suited for educational projects in robotic sciences.

Stereo Vision

In vision-processing systems, many applications require multi-camera support. For the connection of the cameras to the processing system, multiple interfaces and a platform capable of handling sustained high-data rates are essential. To cope with these requirements, a hardware-based solution using FPGA technology is advisable, especially when targeting space and energy constrained embedded systems. FPGA-based scalable and resource-efficient multi-camera GigE Vision IP core for video and image processing exists. To reduce the number of interfaces needed, the IP core supports the connection of multi-camera interfaces to a single gigabit Ethernet port using an Ethernet switch. The multi-camera GigE Vision IP core is able to extract the raw video data from multiple GigE Vision video streams, reconstruct the video frames from every camera, and pass these data along for further processing. The IP core is implemented on a Xilinx Virtex-4 FPGA and integrated in a complete video processing platform for a full-system realization. In addition to the IP core, bilinear interpolation for image demosaicing with Bayer pattern and an automatic white balance algorithm are implemented for evaluation of the platform. Benchmarking of the hardware implementation has been performed with a total resolution of up to 2048x2048 pixels. Achieved frame rates vary from 25 fps to 345 fps depending on the selected resolution and on the number of cameras used.

The DeepSea G2 stereo-vision system that Tyzx developed features an embedded stereo camera consisting of two CMOS imagers, a Tyzx DeepSea 2-stereo application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a DSP/coprocessor, a PowerPC running Linux, and an Ethernet connection. About the size of a hard-cover book, the G2 delivers real-time, 30 frames-per-second (fps) interpretations of visual data even in challenging environments, and has been deployed in a variety of applications, including person-tracking and autonomous-vehicle navigation.

Stereo-vision algorithms are usually designed to run on standard PCs or PC-based systems. Up to now and due to the limited resources (e.g., internal and external RAM size, CPU power, power consumption, etc.) no general embedded hardware exists where different algorithms can be implemented and run properly. The developed test platform aims to close this gap and implement a convenient platform for the evaluation of different high-level vision algorithms used for embedded systems. Furthermore, it offered an excellent opportunity for testing and evaluating automatic code-generation tools (e.g., MathWorks Real-Time Workshop Embedded Coder).

Real-time method to estimate upper human posture using a stereo-vision embedded system has been developed. Using the stereo-vision system produces precise disparity images, and it is possible to execute real-time posture estimation with high accuracy. By using image processing and trigonometry, the positions of head and hands were acquired and discriminated. Using these positions, shoulder positions are estimated. Finally, inverse kinematics is applied to estimate positions of elbows.

Stereo matching, a key element in extracting depth information from stereo images, is widely used in several embedded consumer electronic and multimedia systems. Such systems demand high-processing performance and accurate depth perception, while their deployment in embedded and mobile environments implies that cost, energy, and memory overheads need to be minimized. Hardware acceleration has been demonstrated in efficient embedded stereo-vision systems. The guided filter design is used in two parts of the stereo-matching pipeline, showing that it can simplify the hardware complexity of the adaptive support weight aggregation step, and efficiently enable a powerful disparity refinement unit, which improves matching accuracy, even though cost aggregation is based on simple, fixed-support strategies. Several design variants were implemented on a Kintex-7 FPGA board, which was able to process HD video ($1,280 \times 720$) in real

time (60 fps) using ~57.5k and ~71k of the FPGA's logic (CLB) and to register resources, respectively.

Recent years have seen the widespread diffusion of 3D sensors, mainly based on active technologies such as structured light and time-of-flight, enabling the development of very interesting 3D-vision applications. Compact 3D cameras based on passive stereo-vision technology suited for mobile/embedded vision applications is described here. The developed 3D camera is very compact, the overall area of the processing unit is smaller than a business card. It's lightweight, it weight less than 100 g including lenses; and it has a reduced power consumption, about 2-watt processing stereo pairs at 30+ fps, and can be easily configured with different baselines and processing units according to specific application requirements. The overall design was mapped on to a low-cost FPGA, making the hardware design easily portable to other reconfigurable devices, and allows us to obtain in real-time, both accurate and dense, depth maps according to state-of-the-art stereo-vision algorithms.

Vision Measurement

Vision measurement is sensor-based and consists of the embedded Linux system, an ARM kernel microprocessor system, a camera, and a line-structured light projector. A laser plane is projected on the object surface, and the vision sensor captures the laser-stripe images. The images are processed and image features are extracted in the embedded system. The depth information of object surface and 3D coordinates are calculated and obtained by model calibration. The measurement data can be transmitted via Ethernet, serial port, and field bus, and an embedded-vision sensor is developed for industrial filed applications.

For roadbed settlement, we generally used a method of ceramic micropressure sensor or silicon micro-pressure sensor measuring the pressure difference of the water produced in the settling tube. In order to overcome the deficiency of existing techniques, the proposed is a novel effective method using laser and embedded computer vision to measure the deviation from the laser spot center. Compared to traditional techniques, it has the advantages of being simple, low cost, and easy to deploy.

The frame rate of commercial off-the-shelf industrial cameras is breaking the threshold of 1000 frames-per-second, the sample rate required in high-performance motion-control systems. On the one hand, it enables computer vision as a cost-effective feedback source; but on the

other hand, it imposes multiple challenges on the vision-processing system. Designed and implemented an FPGA-based embedded-vision system in support of high frame-rate visual servo applications. The vision system will be demonstrated together with a mechanical system for vision based inkjet printing. This demonstration shows that, with off-the-shelf components, a robust, hard real time, low delay, embedded vision system is feasible for industrial applications.

An embedded vision-processing platform for a simple automatic die-bonding system using an ARM microprocessor as the vision processor, Linux as the operation system, and a USB camera for image acquisition equipment was designed. It completed wafer image acquisition, processing, wafer alignment, and defects inspection. The platform was competent for wafer image processing and analysis in die bonders and can be widely applied in the field of industry measurement and control.

Online monitoring of stem diameter information without interrupting natural growth of the crops is important for the water-saving irrigation. In order to build a compact, mobile measurement system, DSP embedded Smart Camera VC-4472 was selected as the main components of device. The smart VC-4472 has integrated three TMS320C64xx DSP cores and has a higher image-processing power than the conventional PC-based image-processing system. The real-time processing of the image can be accomplished on the camera alone without using PC. Secondly, the imaging system of the device uses a background illumination source to improve the quality of images of stems. Programs enable us to automatically measure and record the diameter of plant stem.

Industrial Vision

Here we describe a neuromorphic dual-line vision sensor and signal-processing concepts for object recognition and classification. The system performs ultra high-speed machine-vision with a compact and low-cost embedded-processing architecture. The main innovation includes efficient edge extraction of moving objects by the vision sensor on pixel level and a novel concept for real-time embedded vision processing based on address-event data. The system exploits the very high temporal resolution and the sparse visual-information representation of the event-based vision sensor. The 2×256 pixel dual-line temporal-contrast vision sensor asynchronously responds to relative illumination-intensity changes and consequently extracts contours of moving objects. Data-volume independence from

object velocity was shown and evaluates the data quality for object velocities of up to 40 m/s (equivalent to up to 6.25 m/s on the sensor's focal plane). Subsequently, an embedded-processing concept was presented for real-time extraction of object contours and for object recognition.

Automobile Industry

Embedded-vision engine (EVE), is a novel vision accelerator designed to complement digital signal processors on low- and mid-level vision algorithms. EVE's performance on three important mid-level vision functions, the Hough transform for circles, integral image, and calculating the rotation invariant binary robust independent elementary feature (RBRIEF) descriptor are illustrated. EVE can execute these functions four times faster than the state-of-the-art digital signal processor. With other vision functions accelerated 3-12X, the acceleration of these popular vision functions contribute to the application level speed up of 5x on common automotive vision applications.

Active safety guard in vehicles has become a necessity for the ever-increasing transportation density. An in-vehicle embedded system for vision-based driving environment perception and hazardous situation warning using the image processing and computer-vision technology was developed. The system runs on the TI TMS320DM642 DSP, which is a high-performance digital-signal processor produced by Texas Instruments Corporation. The embedded-driving environment perception algorithm consists of two parts: the lane markings detection part and the preceding vehicles recognition part. The system has been evaluated in series of actual driver assistance tests.

The approaches to solve the problems of aviation combined vision system making use of the best properties and functional characteristics of the enhanced vision system, forming enhanced image from a multiple-sensor vision system and synthetic-vision system forming a virtual district model image by means of digital map, navigation, and flight parameters of the aircraft developing were discussed and developed by researcher.

Medical Vision

Research in the field of medical vision has yielded embedded vision, which is the ability of an embedded system to make logical or intelligent decisions based on the images captured and processed by the system. Point-of-care diagnostics provide clinical diagnostic facilities to the patient at the site, thereby reducing time taken for diagnosis and enhancing the quality of

treatment. The proposed system enables medical professionals to interpret these point-of-care clinical diagnostics and connect the different remote diagnostic sites, so that they can be accessed, monitored, controlled, and maintained by a trained medical professional.

Docking an embedded intelligent wheelchair into a U-shape bed automatically through visual servo, real-time U-shape bed localization method on an embedded-vision system based on FPGA and DSP was proposed. This method locates the U-shape bed through finding its line contours. The task can be done in a parallel way with FPGA does line extraction and DSP does line contour finding.

Embedded Vision System

A new energy-saving system whose multiple switches are operated independently according to the regional movement detection in video stream was researched. This system consists of an embedded-vision system and LED switching circuit, communicating with each other through GPIO. An embedded-vision system is implemented by a 32-bit RISC processor, Open RISC, and peripherals—SDRAM, SSRAM, DMA, TFT-LCD, GPIO, and CIS—connected on a Wishbone on-chip bus. One-bus structure is too busy to provide enough high-speed transferring of image data from the camera both to TFT-LCD and to the processor that is executing motion detecting program. So segmented bus structure is introduced to achieve higher performance. The FPGA implementation confirms higher data-transferring speed, suitable for a real-time video application. Only simulated performance was verified by the researcher.

A kind of remote-vision system based on embedded technology was designed, which can transmit the picture data on Ethernet, being free from the distance restrictions, needless repeated wiring, and not being easily interfered with by other parts of computer system. The software and hardware design of vision system consisting of CMOS image sensors, ARM microprocessor, and embedded OS were introduced. Modularizations of hardware compatible to the standard I²C bus, detailed instruction of embedded processor and image sensor and the realization of application programs were emphasized. The design of hardware parts including a high-speed ARM S3C4510B processor, image-sensor module, and image-buffering control module was undertaken. Driver of uClinux, image collecting, image compressing, and network communication program were also analyzed by researchers.

It was demonstrated how fiducial marker-tracking algorithms can be adopted for operation on Raspberry Pi. Use of the proposed ideas allowed it to achieve around 60fps speed of binary marker tracking and describing the problem of text detection and recognition in an outdoor environment. Experimental results indicated desirable results and good potential to provide low-cost and efficient embedded-vision systems for this purpose.

A small and low-cost 8-bit MCU-based embedded-vision system in the form of an intelligent sensor to track multiple YUV 24-bit color objects in real time was built. The system uses an Atmel AVR 8-bit microcontroller ATmega64, which was connected directly to the C3088 camera module using the Omnivision OV6620 CMOS image sensor, eliminating the need of a frame-grabber. It was a functioning vision system, with a well-defined interface that was accessible through a standard serial port, providing high-level, post-processed image information to a primary system (PC, another microcontroller, etc.).

A simple low-cost embedded vision system which has been developed and can satiate the requirement of a small vision system in various surveillance and control systems and in applications which require an artificial vision to be embedded in them were described by researchers. This system utilizes a low-cost CMOS camera module and all image data was processed by a high-speed low-cost AVR microcontroller. The embedded-vision system implemented simple image-processing algorithms and taps the processing power of the AVR microcontrollers with the firmware which in turn also helps to reduce the hardware cost.

There is also an embedded-vision system, which integrates a web-cam quality CMOS imaging chip with a RISC processor, to perform real-time car-counting functions in the indoor and outdoor environment. The challenge of this application, especially for the outdoor environment, was to develop vision algorithms for day and night, and during the light-transition periods (i.e., dawn and dusk). The vision system also needs to accommodate a tremendous range of illumination change (from sunny summer to snowy winter). The entire system consists of a network of 13 embedded-vision systems covering a parking facility over one square kilometer in size. The vision network has been in daily use for the employee parking guidance.

This list will not end here. Listed are only very few research works for reference. The research in this area is still on-going and there is still a lot of room for development in the embedded-vision research area.

Summary

- Image subtraction, thresholding, histogram, and median filtering are examples for point algorithms.
- Mean filtering, Hough transform, and Laplacian of Gaussian for edge detection are local algorithm examples.
- Fourier spectrum, Sobel filtering, convolution, and compression are global examples.
- In a face-recognition system the location of eyes, nose, and mouth are done by an active-shape model.
- Clustering is grouping data points into a specific group.
- Based on group center value and calculating distance technique is used in K-means clustering.
- Based on dense areas of data points (sliding window) and updating center point to the mean of points in the sliding window is the concept of mean-shift clustering.
- DBSCAN is density-based clustering and it has the advantage of outlining noises in data points.
- EM using GMM clustering is more flexible, supports mixed membership, and based on a mean, standard deviation of data points algorithm.
- Thinning is used to reduce the threshold output of an edge detector such as Sobel operator.
- Morphological methods are used to achieve detection of license plates on cars.
- Hough Transforms are used in applications such as object detection, road-sign recognition, industrial and medical applications, pipe and cable inspection, and underwater tracking.

References

<https://www.embedded-vision.com/technology/computer-vision-algorithms>

<https://www.vision-systems.com/features/applying-algorithms-for-machine-vision.html>

<https://ieeexplore.ieee.org/searchresult.jsp?newsearch=true&queryText=embedded%20vision>

Learning Outcomes

- 10.1 Write about the three classification of algorithms in embedded vision.
- 10.2 What is meant by active shape model?
- 10.3 Define clustering algorithms.
- 10.4 Explain k-means clustering.
- 10.5 Write the steps in mean-shift clustering.
- 10.6 What are the advantages of density-based spatial clustering?
- 10.7 Write in short about the EM using GMM clustering algorithm.
- 10.8 Write a short note on agglomerative hierarchical clustering.
- 10.9 Write about the working of thinning morphological operation.
- 10.10 What is Hough transform?
- 10.11 Explain embedded-vision-based measurement.
- 10.12 Describe defect detection on hardwood logs using laser scanning.
- 10.13 How do vision technologies help in empty-bottle inspection systems?
- 10.14 Give hardware configurations and software used for unmanned rotorcraft in ground-target following.
- 10.15 Write the system requirements and algorithm for automatic-lift axle system design.
- 10.16 Give embedded system architecture for object tracking using an address event-vision sensor.
- 10.17 Explain in detail about the FPGA as SoC in ADAS design.
- 10.18 Draw an example of diagnostic imaging equipment.

Further Readings

1. *Open CV: Computer Vision Projects with Python* by Howse Joseph
2. *Computer Vision: Algorithms and Applications* by Richard Szeliski

Appendix

Embedded Vision Glossary

2-D Sensor: An image sensor that discerns the horizontal and vertical location of objects in front of it, but not their distance from it.

3-D Sensor: An image sensor that discerns not only objects horizontal and vertical locations, but also their distance (i.e., depth) from it, by means of techniques such as stereo sensor arrays, structured light, or time-of-flight.

4-D Sensor: See Plenoptic Camera

Active-Pixel Sensor: An APS, also commonly known as a CMOS sensor, this image sensor type consists of an array of pixels, each containing a photo detector and active amplifier. An APS is typically fabricated on a conventional semiconductor process, unlike the CCD.

Adaptive Cruise Control: An ADAS system that dynamically varies an automobiles speed in order to maintain an appropriate distance from vehicles ahead of it.

ADAS: Advanced Driver Assistance Systems, an “umbrella” term used to describe various technologies used in assisting a driver in navigating a vehicle. Examples include:

- In-vehicle navigation with up-to-date traffic information
- Adaptive cruise control
- Lane-departure warning
- Lane-change assistance
- Collision avoidance
- Intelligent speed adaptation/advice
- Night vision
- Adaptive headlight control
- Pedestrian protection
- Automatic parking (or parking assistance)

- Traffic-sign recognition
- Blind-spot detection
- Driver drowsiness detection
- Inter-vehicular communications, and
- Hill descent control

Algorithm: A method for calculating a function, expressed as a list of instructions. Beginning with an initial state and initial input, the instructions describe a computation that, when executed, will proceed through a finite number of defined states, eventually producing an output and terminating at a final state.

Application Processor: A highly integrated system-on-chip, typically comprising a high-performance CPU core and a constellation of specialized co-processors, which may include a DSP, a GPU, a video processing unit (VPU), an image acquisition processor, etc. The specialized co-processors found in application processors are usually not user-programmable, which limits their utility for vision applications.

Augmented Reality: A live view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. The technology functions by enhancing one's current perception of reality. In contrast, virtual reality replaces the real world with a simulated one.

Analytcs: The discovery, analysis, and reporting of meaningful patterns in data. With respect to embedded vision, the data input consists of still images and/or video frames.

API: Application programming interface, a specification intended for use as an interface to allow software components to communicate with each other. An API is typically source code-based, unlike an ABI (application binary interface) which, as its name implies, is a binary interface.

Background Subtraction: A computational vision process that involves extracting foreground objects in a particular scene, in order to improve the subsequent analysis of them.

Barrel Distortion: An optical system distortion effect that causes objects to become "spherized" or "inflated," that is, resulting in the bulging outward of normally straight lines at image margins. Such distortion is typically

caused by wide-angle lenses, such as the fisheye lenses commonly found in automotive backup cameras. Embedded-vision techniques can be used to reduce or eliminate barrel distortion effects.

Bayer Pattern: A common color-filter pattern used to extract chroma information from a nominally monochrome photo detector array, via filters placed in front of the image sensor. The Bayer pattern contains twice as many green filters as either red or blue filters, mimicking the physiology of the human eye, which is most sensitive to green-frequency light. Interpolation generates an approximation of the remainder of each photo detector's full color spectrum.

Biometrics: The identification of humans by their characteristics or traits. Embedded-vision-based biometric schemes include facial recognition, fingerprint matching, and retina scanning.

Camera: A device used to record and store images; still, video, or both. Cameras typically contain several main subsystems; an optics assembly, an image sensor, and a high-speed data transfer bus to the remainder of the system. Image processing can occur in the camera, the system, or both. Cameras can also include supplemental illumination sources.

CCD: A charge-coupled device, used to store and subsequently transfer charge elsewhere for digital-value conversion and other analysis purposes. CCD-based image sensors employ specialized analog semiconductor processes and were the first technology to achieve widespread usage. They remain popular in comparatively cost-insensitive applications where high-quality image data is required, such as professional, medical, and scientific setting.

CImg: An open-source C++ toolkit for image processing, useful in embedded-vision implementations.

CMOS Sensor: Also referred to as active-pixel sensor. CMOS is a digital, high-speed, low-power device. The charge from photosensitive pixel is converted to a voltage at the pixel site.

Collision Avoidance: An ADAS system that employs embedded vision, radar and/or other technologies to react to an object ahead of a vehicle. Passive collision avoidance systems alert the driver via sound, light, vibration of the steering wheel, etc. Active collision avoidance systems override the driver's manual control of the steering wheel, and the accelerator and/or brakes in order to prevent a collision.

Computer Vision: The use of digital processing and intelligent algorithms to interpret meaning from images or video. Computer vision has mainly been a field of academic research over the past several decades.

Contour: One of a number of algorithms which finds use in delineating the outline of an object contained within a 2-D image.

Core Image: The pixel-accurate nondestructive image-processing technology in Mac OS X (10.4 and later) and iOS (5 and later). Implemented as part of the QuartzCore framework, Core Image provides a plugin-based architecture for applying filters and effects within the Quartz graphics rendering layer.

CPU: Central processing unit, the hardware within a computer system which carries out program instructions by performing basic arithmetical, logical, and input/output operations of the system. Two common CPU functional units are the arithmetic logic unit (ALU), which performs arithmetic and logical operations, and the control unit (CU), which extracts instructions from memory and decodes and executes them.

CUDA: Compute unified device architecture, a parallel computing “engine” developed by NVIDIA, found in graphics processing units (GPUs), and accessible to software developers through variants of industry standard programming languages. Programmers use “C for CUDA” (C with NVIDIA extensions and certain restrictions), compiled through a PathScale or Open64 C compiler, to code algorithms for execution on the GPU. AMD’s competitive approach is known as Stream.

Development Tools: Programs and/or applications that software developers use to create, debug, maintain, or otherwise support other programs and applications. Integrated development environments (IDEs) combine the features of many tools into one package.

DirectCompute: An application programming interface (API) that supports general-purpose computing on graphics processing units on Microsoft Windows Vista and Windows 7. DirectCompute is part of the Microsoft DirectX collection of APIs and was initially released with the DirectX 11 API but runs on both DirectX 10 and DirectX 11 graphics processing units.

DSP: Digital signal processor, a specialized microprocessor with an architecture optimized for the fast operational needs of digital signal processing. Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repeatedly

on a set of data. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within some fixed time, and deferred (or batch) processing is not viable.

Edge Detection: A fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction, which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities.

Embedded Vision: The merging of two technologies: embedded systems and computer vision. An embedded system is any microprocessor-based system that isn't a general-purpose computer. Computer vision is the use of digital processing and intelligent algorithms to interpret meaning from images or video. Today, due to the emergence of very powerful, low-cost, and energy-efficient processors, it has become possible to incorporate vision capabilities into a wide range of embedded systems.

Emotion Discernment: Using embedded-vision image processing to discern the emotional state of a person in front of a camera, by means of facial expression, skin color and pattern, eye movement, etc. One rudimentary example of the concept is the "smile" feature of some cameras, which automatically takes a picture when the subject smiles.

Epipolar Geometry: The geometry of stereo vision. When two cameras view a 3D scene from two distinct positions, a number of geometric relations exist between the 3-D points and their projections onto the 2-D images that lead to constraints between the image points. Epipolar geometry describes these relations between the two resulting views.

Face Detection: Using embedded-vision algorithms to determine that one or multiple human (usually) faces are present in a scene, and then taking appropriate action. A camera that incorporates face detection features might, for example, adjust focus and exposure settings for optimum image capture of people found in a scene.

Face Recognition: An extrapolation of face detection, which attempts to recognize the person or people in an image. In the most advanced case, biometric face recognition algorithms might attempt to explicitly identify an individual by comparing a captured image against a database of already identified faces. On a more elementary level, face recognition can find use in ascertaining a person's age, gender, ethnic orientation, etc.

Feature extraction and detection: Most feature extraction and detection algorithms include edge detection, line tracing, object shape analysis, a

classification algorithm and template matching. Sometimes the image is transformed into a different domain such as Fourier and Wavelet before features are extracted.

FPGA: Field programmable gate array, an integrated circuit designed for configuration by a customer after manufacturing. The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). FPGAs contain programmable logic components called “logic blocks”, and a hierarchy of reconfigurable interconnects that allow the blocks to be “wired together.” Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

Framework: A universal reusable software platform used to develop applications, products, and solutions. Frameworks include support programs, compilers, code libraries, an application programming interface (API) and tool sets that bring together all the different components needed to enable development of a project or solution.

Function: Also known as a subroutine, a segment of source code within a larger computer program that performs a specific task and is relatively independent of the remaining code.

Fusion: AMD’s brand for the combination of a CPU and GPU on a single integrated piece of silicon, with the GPU intended to implement general-purpose operations beyond just graphics processing.

Gaze Tracking: Also known as eye tracking, the process of measuring the eye position and therefore the point of gaze (i.e. where the subject is looking). Embedded-vision-based gaze tracking systems employ noncontact cameras in conjunction with infrared light reflected from the eye. Gaze tracking can be used as a computer user interface scheme, for example, with cursor location and movement that tracks eye movement, and it can also be used to assess driver alertness in ADAS applications.

Gesture Interface: The control of a computer or other electronic system by means of gestures incorporating the position and movement of fingers, hands, arms, and other parts of the human body. Successive images are captured and interpreted via embedded-vision cameras. Conventional 2-D-image sensors enable elementary gesture interfaces; more advanced

3-D sensors that discern not only horizontal and vertical movement but also per-image depth (distance) allow for more complex gestures, at the tradeoffs of increased cost and computational requirements.

GPGPU: General-purpose computing on graphics processing units, the design technique of using a graphics processing unit (GPU), which typically handles computation only for computer graphics, to perform computation in applications traditionally handled by the central processing unit (CPU).

GPU: Graphics processing unit, a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the building of images in a frame buffer intended for output to a display. GPUs are very efficient at manipulating computer graphics, and their highly parallel structure makes them more effective than general-purpose CPUs for algorithms where processing of large blocks of data is done in parallel.

HDR: High dynamic range imaging, a set of methods used to allow a greater dynamic range between the lightest and darkest areas of an image. This wide dynamic range allows HDR images to represent more accurately the range of intensity levels found in real scenes.

Image and vision processing algorithms: At the heart of inspection systems are a host of image and vision processing algorithms. These algorithms can be grouped into several categories, including image enhancement and formation, morphological operations, and feature extraction and detection.

Image Processor: A specialized digital signal processor used as a component of a digital camera. The image processing engine can perform a range of tasks, including Bayer-to-full RGB per-pixel transformation, demosaic techniques, noise reduction, and image sharpening.

Image Search: The process of searching through a database of existing images to find a match between objects contained within one/some of them (such as a face) and content in a newly captured image.

Image Sensor: A semiconductor device that converts an optical image into an electronic signal, commonly used in digital cameras, camera modules and other imaging devices. The most common image sensors are charge-coupled device (CCD) and complementary metal-oxide-semiconductor (CMOS) active pixel sensors.

Image Warping: The process of digitally manipulating an image such that any shapes portrayed in the image are notably altered. In embedded-vision

applications, warping may be used either for correcting image distortion or to further distort an image as a means of assisting subsequent processing.

IMGLIB: Image processing library, Texas Instruments' DSP-optimized still image processing function library for C programmers.

Industrial Vision: See also Computer Vision. In this industrial vision camera, sensors and processors make decision on what it sees and relays information to the handling system so that it introduces automation in industries.

Infrared Sensor: An image sensor that responds to light in the infrared (and near-infrared, in some cases) frequency spectrum. The use of infrared light transmitters to assist in determining object distance from a camera can be useful in embedded-vision applications because infrared light is not visible to the human eye. However, ambient infrared light in outdoor settings, for example, can interfere with the function of infrared-based embedded-vision systems.

Intelligent Video: A term commonly used in surveillance systems, it comprises any solution where the system automatically performs an analysis of the captured video.

IPP: Intel's integrated performance primitives, a library of software functions for multimedia, data processing, and communications applications. Intel IPP offers thousands of optimized functions covering frequently used fundamental algorithms.

Kinect: A motion sensing input add-on peripheral developed by Microsoft for the Xbox-360 video game console and Windows PCs. Kinect enables users to control and interact with the system without the need to touch a game controller, through a natural user interface using gestures and spoken commands.

Lane Transition Alert: An ADAS system that employs embedded vision and/or other technologies to react to a vehicle in the process of transitioning from one roadway lane to another, or off the roadway to either side. Passive lane transition alert systems alert the driver via sound, light, vibration of the steering wheel, etc. Active collision avoidance systems override the driver's manual control of the steering wheel in order to return the vehicle to the previously occupied roadway lane.

Lens Distortion Correction: Employs embedded-vision algorithms to compensate for the image distortions caused by suboptimal optics systems

or those with inherent deformations, such as the barrel distortion of fisheye lenses.

Library: A collection of resources used by programs, often to develop software. Libraries may include configuration data, documentation, help data, message templates, pre-written code and subroutines, classes, values, and type specifications. Libraries contain code and data that provide services to independent programs. These resources encourage the sharing and changing of code and data in a modular fashion, and ease the distribution of the code and data.

Machine Vision: See also computer vision and industrial vision. It includes algorithms to measure and identify items in the image captured through a camera and passes information to control PLC.

Middleware: Computer software that provides services to software applications beyond those available from the operating system. Middleware, which can be described as “software glue,” makes it easier for software developers to perform communication and input/output, so they can focus on the specific purpose of their application.

Microprocessor: See also CPU. An integrated circuit that contains all the functions of a central processing unit of a computer.

Morphological operations: Morphological operations are nonlinear operations which incorporate a “structuring element” that probes the image, providing results on how well an elemental structure fits within the image. The outputs of morphological operations could result in thickening or thinning edges, removing small objects within a larger object, connecting broken edges, eliminating small holes, and filling small gaps.

Motion Capture: Also known as motion analysis, motion tracking, and mocap; the process of recording movement of one or more objects or persons. It is used in military, entertainment, sports, and medical applications, and for validation of computer vision and robotics. In filmmaking, and games, it refers to recording the movements (but not the visual appearance) of human actors via image samples taken many times per second, and using that information to animate digital character models in 2D or 3D computer animation. When it includes face and fingers or captures subtle expressions, it is often referred to as performance capture.

NI Vision for LabVIEW: National Instruments configuration software and programming libraries that assist in building imaging applications. It

comprises NI Vision Builder for Automated Inspection and the NI Vision Development Module, the latter a comprehensive library with hundreds of scientific imaging and machine vision functions that you can program using NI LabVIEW software and several text-based languages.

NPP: The NVIDIA Performance Primitives library, a collection of GPU-accelerated image, video, and signal processing functions. NPP comprises over 1,900 image processing primitives and approximately 600 signal processing primitives.

Object Tracking: The process of locating a moving object (or multiple objects) over time using a camera. The objective is to associate target objects in consecutive video frames. However, this association can be especially difficult when the objects are moving faster than the frame rate. Another situation that increases the complexity of the problem is when the tracked object changes orientation over time.

OCR: Optical character recognition, the conversion of scanned images of handwritten, typewritten, or printed text into machine-encoded text. OCR is widely used as a form of data entry from some sort of original paper data source, whether documents, sales receipts, mail, or any number of printed records. It is crucial to the computerization of printed texts so that they can be electronically searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text-to-speech, and text mining.

OpenCL: Open computing language, a framework for writing programs that execute across heterogeneous platforms consisting of central processing units (CPUs), graphics processing units (GPUs), and other processors. OpenCL includes a language (based on C99) for writing kernels (functions that execute on OpenCL devices), plus application programming interfaces (APIs) that are used to define and then control the platforms. OpenCL provides parallel computing using task-based and data-based parallelism.

OpenCV: A library of programming functions mainly aimed at real-time image processing, originally developed by Intel, and now supported by Willow Garage and Itseez. It is free for use under the open source BSD license. The library is cross-platform.

OpenGL: Open graphics library, a standard specification defining a cross-language, multi-platform API for writing applications and simulating physics, that produces 2D and 3D computer graphics. The interface consists of over 250 different function calls, which can be used to draw complex

three-dimensional scenes from simple primitives. OpenGL functions can also be used to implement some GPGPU operations.

OpenNI: Open natural interaction, an industry-led, nonprofit organization focused on certifying and improving interoperability of natural user interface and organic user interface for natural interaction devices, applications that use those devices, and middleware that facilitates access and use of such devices.

OpenVL: A modular, extensible, and high-performance library for handling volumetric datasets. It provides a standard, uniform, and easy-to-use API for accessing volumetric data. It allows the volumetric data to be laid out in different ways to optimize memory usage and speed. It supports reading/writing of volumetric data from/to files in different formats using plugins. It provides a framework for implementing various algorithms as plugins that can be easily incorporated into user applications. The plugins are implemented as shared libraries, which can be dynamically loaded as needed. OpenVL software is developed openly and is freely available on the web.

Operating System: A set of software that manages computer hardware resources and provides common services for computer programs. For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and will frequently make a system call to an operating system function or be interrupted by it.

Optical Flow: The pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. Optical-flow techniques such as motion detection, object segmentation, time-to-collision and focus of expansion calculations, motion compensated encoding, and stereo disparity measurement utilize the motion of the objects surfaces and edges.

Photogrammetry: The practice of determining the geometric properties of objects from photographic images. Algorithms for photogrammetry typically express the problem as that of minimizing the sum of the squares of a set of errors. This minimization is known as bundle adjustment and is often performed using the Levenberg–Marquardt algorithm.

Pincushion Distortion: The opposite of barrel distortion; image magnification increases with the distance from the optical axis. The visible

effect is that lines that do not go through the center of the image are bowed inward, toward the center of the image, like a pincushion. Embedded-vision techniques can be used to reduce or eliminate pincushion distortion effects.

Plenoptic Camera: Also known as a light-field camera, it uses an array of microlenses, at the focal plane of the main lens and slightly ahead of the image sensor, to capture light field information about a scene. The displacement of image parts that are not in focus can be analyzed and depth information can be extracted. Such a camera system can therefore be used to refocus an image on a computer after the picture has been taken.

Point Cloud: A set of vertices in a three-dimensional system, usually defined by X, Y, and Z coordinates, and typically intended to be representative of the external surface of an object. Point clouds are often created by 3D scanners. These devices measure in an automatic way a large number of points on the surface of an object, and often output a point cloud as a data file. The point cloud represents the set of points that the device has measured.

Processor: See also CPU and microprocessor. A processor is an integrated electronic circuit that performs that calculations that run a computer.

Reference Design: A technical blueprint of a system that is intended for others to copy. It contains the essential elements of the system; recipients may enhance or modify the design as required. Reference designs enable customers to shorten their time to market, thereby supporting the development of next generation products using latest technologies. The reference design is proof of the platform concept and is usually targeted for specific applications. Hardware and software technology vendors create reference designs in order to increase the likelihood that their products will be used by OEMs, thereby resulting in a competitive advantage for the supplier.

Resolution: The amount of detail that an image holds. Higher resolution means more image detail. Resolution quantifies how close lines can be to each other and still be visibly resolved. Resolution units can be tied to physical sizes (e.g., lines per mm, lines per inch), to the overall size of a picture (lines per picture height), to angular subtendant, or to the number of pixels in an image sensor. Line pairs are sometimes used instead of lines; a line pair comprises a dark line and an adjacent light line.

SoC: System-on-a-chip, an integrated circuit (IC) that integrates most if not all components of an electronic system into a single chip. It may contain

digital, analog, mixed-signal, and often radio-frequency functions—all on a single chip substrate.

Stereo Vision: The use of multiple cameras, each viewing a scene from a slightly different perspective, to discern the perceived depth of various objects in the scene. Stereo vision is employed by the human vision system via the two eyes. The varying perspective of each camera is also known as binocular disparity.

Stream: A set of hardware and software technologies originally developed by ATI Technologies and now managed by acquiring company AMD (and renamed App). Stream enables AMD graphics processors (GPUs), working with the system's central processor (CPU), to accelerate applications beyond just graphics (i.e., GPGPU). The Stream Software Development Kit (SDK) allows development of applications in a high-level language called Brook+. Brook+ is built on top of ATI Compute Abstraction Layer (CAL), providing low-level hardware control and programmability. NVIDIA's competitive approach is known as CUDA.

Structured Light: A method of determining the depths of various objects in a scene, by projecting a predetermined pattern of light onto the scene for the purpose of analysis. 3-D sensors based on the structured light method use a projector to create the light pattern and a camera to sense the result. In the case of the Microsoft Kinect, the projector employs infrared light. Kinect uses an astigmatic lens with different focal lengths in the X and Y direction. An infrared laser behind the lens projects an image consisting of a large number of dots that transform into ellipses, whose particular shape and orientation in each case depends on how far the object is from the lens.

Surveillance System: A camera-based system that implements scene monitoring and security functions. Historically, the outputs of surveillance-systems cameras were viewed by humans via television monitors. Embedded-vision-based surveillance systems are now replacing the often-unreliable human surveillance factor, via automated “tripwire,” facial detection, and other techniques.

Time-of-Flight: A method of determining the depths of various objects in a scene. A time-of-flight camera contains an image sensor, a lens and an active illumination source. The camera derives distance from the time it takes for projected light to travel from the transmission source to the object and back to the image sensor. The illumination source is typically either a pulsed laser or a modulated beam, depending on the image sensor type employed in the design.

Video Analytics: Also known as video content analysis, it is the capability of automatically analyzing video to detect and determine temporal events not based on a single image. The algorithms can be implemented on general-purpose computers or specialized embedded-vision systems. Functions that can be implemented include motion detection against a fixed background scene. More advanced functions include tracking, identification, behavior analysis, and other forms of situation awareness.

VLIB: Video processing library, Texas Instruments DSP-optimized video-processing function library for C programmers.

VXL: A collection of open source C++ libraries for vision applications. The intent is to replace X with one of many letters, that is, G (VGL) is a geometry library, N (VNL) is a numerical library, I (VIL) is an image processing library, etc. These libraries can also be used for general scientific computing applications.

Index

3D cameras, 33
3D Depalletizing, 64
3D mapping, 382
3D model, 383
3D pattern matching, 86
3D stereo industrial vision, 57
3D vision sensor technology, 33

A

Acceleration sensor, 339
Accuracy, 25, 27, 31, 40, 53, 58, 64, 69, 78, 79, 81, 82, 83, 98, 108, 143, 173, 174, 178, 184, 185, 199, 247, 290, 309, 337, 352, 353, 358, 360, 389, 393, 401, 403, 410, 412, 423, 424, 426, 438, 439, 451, 453, 457, 458, 460, 469, 472, 481, 495, 500, 501, 521, 528
Active shape model, 52, 475, 482–483
Adaptive smoothing filter, 126
Adaptive thresholding, 384
ADAS design, 516
ADAS, 20, 21
Agbot II, 413, 417
AI embedded in camera, 423
AI learning algorithm, 432–435
Air fuel ratio meter, 348
Airspeed indicators, 342
Algorithm, 117–135
Altimeters, 340
Altitude head and reference system, 342
Analog camera, 14, 255, 256, 257–258, 259, 260, 273, 278, 283, 290, 291, 302, 309, 468,
Analog signal, 198, 200, 201, 248, 260, 302
Aperture size, 263
Aperture, 162, 257, 263, 280, 288, 305
Area of interest, 253, 278, 283, 285–286, 288
Area scan cameras, 60, 71, 72, 73, 253, 261, 262
Artificial intelligence (AI), 94, 117–120, 138,

247, 330, 349, 370, 373, 374, 417, 423–474
Artificial neural networks (ANNs), 189
Artificial vision, 117, 118, 119, 423
Aspect ratio, 271
ASSP, 9, 10, 313, 329
Augmented reality, 2, 21, 88, 186, 398, 459, 460, 462
Automobile industry, 345–351
Avatar, 429, 454, 455,
Average method, 270
Axle lifting, 513–514
Ayuda robot, 418

B

Banana Pi, 327
Barrel distortion, 161, 517
Beagle board, 327
Big data, 133–134
Binary morphological operations, 231
Binning, 66, 71, 253, 278, 283, 286, 288, 388
Binocular stereo vision, 447
Biological area, 245
Biometric sensor, 353, 355
Biometrics, 88, 385, 430
Biopsy, 94
Biosensor, 313, 357
Bits, 259, 266, 268, 291
Black box, 200, 223, 344
Blood flow, 116, 399
Blur image, 66
Blurring, 87, 118, 188, 197, 203, 210, 212, 227, 230, 250, 260, 389, 457,
Box blur, 230, 250

C

Calibration, 63
Cam position, 348
Camera interface, 46, 71, 87, 253, 291, 300,

303, 304, 306, 330, 336
 Camera link, 16, 68, 87, 253, 300, 302–303, 306, 325, 336
 Camera Obscura, 254
 Camera software, 304–305
 Cameras, 14–15
 CAN bus node, 347
 Cancerous cell, 107
 Capture board, 24, 253, 260, 261, 271, 300, 302, 374
 Car cruise control system, 181
 Catalog, 53
 CCD array, 257, 258, 265, 280, 309
 CCD sensor, 73, 165, 248, 258, 272, 275, 279, 281, 286, 289
 CCTV, 46, 150, 157, 158, 255, 362, 364
 CEVA-XM4 vision processor, 328–329
 Character recognition, 58, 50, 68, 79, 81, 83–84, 152, 336, 375, 383, 401, 495
 Circle Hough transform, 475, 491
 Classification of objects, 249
 Classification, 39, 42, 48–56
 Cluster shared memory, 323
 Clustering algorithm, 475, 483–486
 CMOS sensor, 273
 CMYK color model, 269
 CMYK, 233, 297
 CNN engine, 321, 323
 CNN technology enablers, 186–188
 CNN, 28, 93, 132–133, 135, 136, 137, 138, 169, 170, 175, 176, 180, 181, 182, 186–188, 194, 313, 321, 323, 335, 366, 390, 391, 392, 393, 431, 433
 Co-processing, 336
 CoaXPress, 46, 68, 71, 303
 Cobots, 179, 405, 406–407
 Color format, 267, 268
 Color processing, 197, 202, 374, 377
 Color spaces, 197, 232, 374, 404
 Compass, 34, 213, 217, 219, 342, 417
 Computer aided diagnostic processing, 120–123
 Computer graphics, 373
 Computer integrated endoscopy, 110
 Computer neural networks (CNN), 132–133
 Computer tomography (CT), 98–99, 522

Computer vision, 369–421
 Conjugate pair, 447
 Contact image sensor, 71, 72–73
 Continuous system, 201
 Continuous wave ToF, 453
 Contouring, 299
 Contrast enhancement, 129, 154, 380, 477
 Convolution layer, 93, 136–137, 169, 175
 Convolution, 28, 93, 125, 135, 136, 138, 149, 169, 175, 178, 180, 184, 197, 203, 209, 210, 226, 227, 230, 250, 322, 332, 366, 389, 390, 391, 392, 393, 424, 431, 433, 471, 478, 480
 Convolutional neural networks, 28, 138–136, 149, 169, 175, 180, 184, 390, 424, 471
 Corneal image analyzer, 93, 105–106
 CPU, 5, 6, 8, 9, 10, 11, 12, 13, 35, 87, 123, 162, 179, 186, 313, 315, 316, 322, 323, 325, 326, 330, 331, 334, 335, 336, 337, 366, 367, 404, 436, 439, 443, 470, 521, 528
 Crank position, 348
 Cross correlation, 237, 238, 242, 245, 250
 Cross validation, 144
 Curvature driven flows, 94
 CV software provider, 397

D

Decision making, 8, 10, 39, 45, 135, 160, 163, 189, 315, 381, 419, 443, 498, 511
 Decision trees, 170, 173, 174, 189
 Deep learning tracker, 391
 Deep learning, 133, 134, 135, 138, 170, 175–176, 187, 194, 317, 323, 330, 366, 391, 429, 430, 431, 433, 434, 443, 444
 Deep neural networks DNN, 169, 175, 404
 Degrees of freedom, 49
 Demosaic, 164
 Dense optical flow, 162–163
 DenseNet visualization, 434
 Density based spatial clustering, 485
 Depth estimation, 184, 185
 Depth map, 60, 438, 452, 454, 461, 529
 Depth sensing, 33
 Description of objects, 249
 Designing quality, 465
 Detection, 16, 20, 21, 23, 25, 26, 27, 32, 42, 44, 52, 53, 54, 55, 56, 62, 68, 83, 90, 91, 93,

- 96, 104, 110, 116, 117, 120, 121, 125, 126, 129, 130, 132, 133, 135, 139, 150, 151, 156, 157, 158, 162, 175, 182, 183, 184, 185, 186, 187, 189, 190, 194, 197, 202, 203, 210, 213, 214, 215, 217, 239, 240, 244, 245, 246, 249, 250, 287, 309, 322, 323, 324, 329, 338, 340, 349, 350, 353, 355, 356, 357, 362, 365, 366, 369, 374, 375, 377, 380, 381, 382, 384, 386, 387, 388, 389, 390, 391, 392, 393, 395, 396, 399, 404, 407, 409, 410, 411, 419, 425, 426, 427, 430, 434, 437, 438, 453, 459, 469, 475, 476, 490, 491, 494, 495, 497, 499, 500, 501, 502, 507, 509, 513, 515, 517, 519, 520, 523
- Diagnostic imaging, 475, 522
- Dielectric soil moisture sensor, 351
- Digital camera interfaces, 300
- Digital camera, 104, 167, 253, 256, 257, 258, 259, 260, 273, 276, 278, 279, 281, 283, 284, 285, 287, 288, 300, 302, 304, 309, 310, 311, 354, 374, 410, 441
- Digital image, 14, 51, 95, 96, 98, 100, 101, 104, 111, 119, 165, 197–251
- Digital signature, 131, 132, 147
- Digital zoom, 292, 331
- Direct Part Marking, 77–78
- Discrete system, 201
- Disk mirroring, 320
- Dissolved Oxygen (DO) sensor, 359, 360, 524
- Dithering, 299, 300
- DORA, 423, 442
- Drive simulator, 182
- Drones, 2, 18, 88, 351, 352, 362, 398, 403, 407, 416, 417, 428, 461
- DSP, 5, 8, 11, 12, 13, 18, 35, 89, 90, 123, 177, 178, 186, 313, 314, 315, 316, 317, 318, 321, 329, 330, 331, 332, 333, 335, 366
- Dynamic range, 15, 73, 90, 96, 101, 164, 165, 166, 253, 273, 275, 285, 288, 290, 291, 306, 310, 330, 349, 437, 470, 515
- E**
- Edge-based matching, 243, 244
- Edge detection, 44, 52, 90, 120, 125, 126, 197, 203, 210, 213, 214, 215, 217, 249, 250, 324, 375, 377, 380, 384, 442, 469, 476, 491, 494, 500
- Eigen faces approach, 384
- Electric field imaging, 168
- Electro chemical sensor, 351
- Electromagnetic spectrum, 201, 202
- Electronic control unit, 345, 346, 367, 513
- Electronic pill, 475, 523
- Electronic shutter, 15, 279
- EM clustering, 486
- Embedded CPU, 9, 37, 313, 315
- Embedded system, 1, 2, 3, 5, 6, 7, 8, 9, 11, 17, 19, 37, 87, 123, 319, 326, 369, 370, 425, 435, 439, 464, 471, 505, 515, 526, 527, 528, 529, 531
- Embedded vision processor, 2, 9, 313–368
- Embedded vision, 1–37
- Emission control, 347
- Emotion sensor, 410
- Endoscopy, 93, 87, 109, 110
- Engine fault diagnosis, 347
- Engine speed sensor, 348
- Epipolar plane, 446, 447
- Escape rates, 44
- Exposure time, 65, 66, 67, 69, 71, 73, 85, 92, 278, 284, 309
- Eye detection, 245
- Eye, 23, 29, 47, 57, 97, 106, 109, 114, 115, 118, 136, 163, 165, 183, 202, 211, 239, 245, 246, 256, 264, 268, 270, 290, 305, 307, 329, 354, 361, 371, 407, 424, 427, 430, 432, 433, 438, 440, 441, 445, 452, 453, 456, 481, 482, 492, 493, 512
- F**
- Face recognition, 23, 242, 331, 384, 404, 481,
- Facial recognition, 22, 93, 108, 150, 158, 167, 194, 418, 430, 440
- False alarm, 44, 184, 426
- Family tree, 371, 372
- FAST, 386, 387, 388, 395, 396
- Feature computation, 89, 143
- Feature extraction, 502, 507, 508
- Feature selection, 51, 91, 143

Features designing, 149, 191
 Feed-forward neural networks FNN, 168
 Fiberscopic images, 498
 Film, 54, 59, 64, 69, 101, 106, 125, 255, 256, 257
 Filters, 56, 72, 97, 125, 126, 136, 184, 210, 213, 222, 227, 229, 232, 277, 283, 306, 320, 331, 332, 376, 383, 384, 476, 478, 479, 510, 522
 Firewire, 43, 46, 253, 259, 300, 302, 304, 306
 Fitting, 28, 132, 389, 476, 501
 Flow sensors, 341, 366
 Force sensor, 339, 358, 411, 418
 Form factor, 313, 366,
 Fourier series, 224, 225
 Fourier transform, 53, 101, 224, 225, 226, 236, 247, 480
 FPGA, 89, 90, 162, 283, 287, 304, 307, 314, 316, 325, 329, 330, 335, 336, 337, 397, 435, 439, 463, 465, 466, 470, 476, 493, 516, 517, 518, 519, 523, 527, 528, 529, 532
 Frame grabbers, 24, 88, 179
 Frame rate, 65, 66, 67, 69, 90, 165, 259, 260, 278, 279, 281, 284, 285, 287, 288, 290, 307, 319, 332, 374, 437, 451, 455, 457, 468, 514, 527, 529, 530
 Framework, 19, 120, 170, 178, 324, 335, 378, 403, 404, 415, 443, 509, 519
 Frequency domain, 197, 210, 223, 224, 226, 227, 480
 Fringe projection, 61
 Fuel temperature sensor, 348
 Fully connected layer, 93, 136, 137, 181, 393
 Fundus image analyzer, 93, 106
 Fuzzy logic, 90, 189

G

Gain, 85, 86, 99, 102, 121, 173, 253, 281, 283, 284, 288, 290, 291, 310, 385, 396, 465, 484
 Gamma radiation detector, 359
 Gamma, 165, 166, 202, 208, 209, 253, 283, 284, 285
 Gaussian blur, 230
 Gaussian filter, 197, 211
 Gaussian high pass filter, 227, 229

Gaussian low pass, 227, 229
 Gaussian mixture model, 141, 486
 Genetic algorithms, 90, 189
 Geometric pattern matching, 237, 239
 Gesture interface, 116, 117
 Gesture, 21, 22, 186, 322, 329, 331, 335, 405, 430, 456, 459, 460, 461, 472, 495
 GigE vision standard, 303
 Global transformation, 480
 GPS, 340, 349, 351, 352, 354, 367, 403, 411
 GPU, 5, 9, 10, 11, 13, 28, 35, 123, 177, 179, 180, 186, 313, 314, 316, 317, 318, 319, 326, 330, 331, 332, 333, 334, 335, 435, 436, 439, 443, 444, 476
 Gray level resolution, 298
 Grayscale mapping, 126
 Grayscale, 454
 Grayscale based matching, 238
 Grayscale morphological operations, 250
 Grip sensor, 410
 Gyroscope, 313, 319, 342, 349, 352

H

Hann window, 230, 250
 Hardwood logs, 497
 Heart rate, 23, 108
 Heterogeneous processing, 470
 Hex Code, 269
 Hierarchical clustering, 141, 487, 488
 High pass filter, 125, 227, 228, 229
 High speed cameras, 47, 65
 Histograms, 131, 203, 204, 205, 231, 388, 394
 HoG, 384, 387, 388, 526
 Home security, 362, 363
 Honeycomb structure, 499
 Hough transform, 52, 91, 384, 475, 479, 490, 520, 531, 534
 HSV, 232, 250, 374, 508,

I

Identification, 29, 52, 55, 58, 65, 69, 78, 80, 83, 86, 132, 138, 156, 162, 183, 184, 185, 193, 244, 349, 382, 385, 386, 401, 402, 403, 411, 418, 429, 470, 494, 495, 516

- Identity transformation, 207
 - Image acquisition, 13, 39, 44, 67, 82, 90, 106, 117, 149, 379, 419, 470, 500, 501, 503, 530
 - Image classification, 49, 90, 135, 142, 323, 389, 390, 392, 404,
 - Image compression, 93, 127, 129, 130, 225, 234, 235, 330, 480
 - Image correlation matching, 241
 - Image digitalization, 247
 - Image enhancement, 91, 96, 110, 112, 125, 146, 197, 206, 207, 331
 - Image guided surgery (IGS), 95
 - Image guided therapy (IGT), 95
 - Image preprocessing, 36, 149, 160, 249, 337
 - Image processing step, 247
 - Image processing, 32
 - Image registration, 119, 133, 146, 162, 242, 243, 384
 - Image restoration, 44, 53, 97, 152, 154, 383, 398
 - Image segmentation, 50, 119, 120, 121, 133, 248, 376
 - Image sensor, 3, 14, 15, 29, 33, 34, 36, 58, 71, 72, 73, 114, 248, 253–311, 330, 349, 353, 354, 361, 370, 374, 379, 399, 425, 446, 451, 517, 532
 - Image sharpening, 197, 202, 203, 227, 249, 478
 - Image signal processor ISP, 163
 - Image smoothing, 119, 146, 197, 490
 - Image transformation, 206, 249
 - Image, 14, 72, 74, 90, 96, 97, 109, 127, 130, 131, 132, 156, 197, 198, 203, 223, 247, 248, 253, 256, 257, 317, 372, 376, 377, 379, 380, 381, 389, 498
 - ImageNet, 176
 - Indo cyanine green (ICG) imaging, 105
 - Industrial camera, 326, 529
 - Industrial robots, 30, 58, 406, 408, 461
 - Industrial security, 364
 - Industrial vision measurement, 39, 81–91
 - Industrial vision system, 39, 40–48, 49, 52, 54, 55, 56, 57, 82, 86, 87, 88, 89,
 - Industrial vision, 39, 41, 77, 398, 399, 530
 - Inertial sensor, 410
 - Inline processing, 337
 - Inspection, 57
 - Instance segmentation, 369, 393
 - Intel Movidius Myraid X, 324
 - Intelligence surveillance, 423, 436
 - Intelligent pill, 525
 - Intelligent sensor, 352, 417, 533
 - Interlaced, 256, 260, 261
 - Internet of things, 134, 440
 - IoT sensor, 353, 367
 - IR imaging, 71, 167, 281
 - Isolation design flow, 467
- ## J
- JPEG compression, 197, 225, 234
- ## K
- K means clustering, 483, 485
 - K nearest neighbors, 170, 171
 - Kalman filter, 376, 384, 457, 508
 - Kirsch compass mask, 219
 - Knightscope robot, 417
 - Knock sensor, 348
 - K-times zooming, 294
- ## L
- Labeled data, 140
 - Lane detection, 26, 513, 520, 521
 - Laplacian operator, 213, 221, 222, 478
 - Laser profiling, 59, 61
 - Laser scanning, 461, 497
 - Laser triangulation, 423, 444, 450, 459
 - Layer, 20, 93, 105, 106, 109, 130, 133, 136, 137, 138, 140, 168, 169, 170, 175, 176, 180, 181, 323, 377, 390, 392, 414, 426, 430, 431, 434, 444, 512
 - Learning process, 138, 143, 180, 319
 - LEDs, 67, 73, 338
 - Lens distortion correction, 161, 334
 - Lens selection, 254, 305
 - Level sensor, 339, 340, 353
 - License plate recognition, 153, 426, 490
 - LIDAR, 187, 348, 349, 350, 362, 407, 411, 418, 462, 520

Line scan cameras, 47, 60, 66, 71, 72, 73, 74,
75, 92, 253, 261, 262, 307
Line scan inspection, 74
Line scan technology, 70, 71
Linear model, 482
Linear smoothing, 230
Linear transformation, 206, 207
Local operators, 478
Logarithmic transformation, 207
Low pass filter, 125, 227, 228, 229, 383
LSPIHT algorithm, 130
LVDS, 518

M

Machine learning, 3, 28, 92, 93, 132, 135, 136,
137, 138, 139, 140, 143, 144, 145, 147, 168,
169, 170, 172, 173, 174, 176, 177, 178, 179,
180, 186, 189, 190, 194, 237, 318, 369, 372,
373, 376, 377, 378, 385, 403, 407, 424, 427,
432, 433, 434, 439, 443, 471, 476, 483, 494
Machine learning model, 142, 444
Machine vision, 2, 3, 5, 16, 23, 24, 25, 37, 40,
41, 42, 43, 44, 46, 47, 51, 52, 53, 54, 67, 84,
85, 87, 88, 97, 179, 180, 197, 237, 239, 250,
324, 325, 326, 330, 369, 371, 373, 374, 375,
376, 378, 382, 398, 407, 419, 432, 445, 451,
460, 462, 463, 471, 477, 478, 504, 519, 530
Magnetic Resonance Imaging (MRI), 95, 100,
127, 135, 232, 377
Magnetometer, 342
MAP sensor, 348
Mars path finder, 406
Mask, 41, 125, 126, 129, 197, 209, 210, 211,
212, 213, 214, 215, 216, 217, 218, 219, 220,
221, 222, 227, 490
Matrox RadiantPro CL, 325, 367,
Mavica, 256
MAX10 FPGA, 329
Mean filter, 211, 478
Mean shift algorithm, 384, 512
Mean shift clustering, 484
Median filter, 126, 230, 231, 250, 383, 478,
490
Medical imaging system (MIS), 102
Medical sensor, 353, 354, 358, 367
Medical vision image processing, 111

Medical vision, 93–147, 398, 399, 531
MEMS, 325, 341, 342, 349, 353, 355, 356,
367, 410
Metadata search, 149, 151
Micro bolometer, 275
Microcontroller, 168, 317, 346, 352, 358, 418,
517, 519, 523, 533
Min/max filters, 231, 232
Mobile application processor, 13
Model, 19, 21, 22, 35, 45, 49, 51, 52, 54, 55,
56, 61, 64, 75, 83, 86, 94, 95, 113, 115, 116,
117, 122, 123, 132, 134, 139, 140
Modulation transfer function (MTF), 288, 289
Motion blur, 65, 66, 154, 279, 383, 450
Motion tracking, 240, 398
Motion, 22, 25, 31, 65, 66, 67, 69, 117, 150,
154, 160, 162, 165, 166, 184, 188, 194, 232,
240, 244, 262, 263, 279, 329, 336, 339, 348,
351, 362, 372, 374, 376, 378, 383, 391, 397,
410, 425, 426, 429, 437, 450, 462, 471, 503,
508, 510, 512, 529, 532
MOZI, 254
MV cameras, 24, 330

N

Naive Bayes algorithm, 170, 174, 194
Naïve template matching, 241
Natural language processing, 432, 434, 471
Navigation, 110, 168, 339, 340, 343, 360, 361,
365, 399, 403, 411, 413, 445, 459, 461, 507,
528, 531
Negative transformation, 206, 207
NEMS, 353, 356
Neural networks, 3, 28, 44, 45, 51, 53, 55, 90,
132, 135, 137, 149, 168, 169, 170, 175, 180,
184, 189, 190, 194, 323, 335, 378, 390, 403,
404, 424, 431, 433, 439, 442, 443, 471, 495,
Nikon, 256, 258
Node, 99, 140
Noise reduction, 162, 165, 166, 210, 211, 230,
330, 331, 374, 380
Nonlinear smoothing, 230
Normalized cross correlation, 237, 241, 242
Nuclear sensor, 358
Nvidia Jetson Tk1, 326, 367
Nyquist limit, 253, 288, 289, 291

O

Object detection, 16, 20, 21, 27, 151, 156,
158, 175, 186, 187, 189, 322, 323, 324, 366,
369, 388, 390, 392, 396, 411, 419, 426, 462

Object recognition algorithm, 369, 393

Object recognition, 3, 21, 90, 91, 119, 243,
244, 247, 369, 381, 384, 388, 393, 394, 395,
423, 428, 433, 460, 508, 530, 531

Object tracking, 243, 381, 391, 404, 419, 445,
475, 514

Occlusion, 50, 240, 389, 508, 511

ODROID-C2, 327, 253, 283, 287, 288, 450,

Offset, 34, 237, 253, 283, 287, 288, 450

Open source tools, 176

OpenCL, 19, 318, 324, 444, 523

OpenCV, 19, 36, 186, 304, 307, 324, 334, 397,
403, 436, 448, 476

OpenPilot, 187

OPENVX, 177, 178, 186, 187, 324, 334, 335

Ophthalmology, 93, 104, 114, 145

Optical character recognition, 60, 79, 81, 83,
152, 336, 375, 383, 401, 495

Optical flow, 162, 183, 184, 324, 331, 369,
384, 397, 439, 507, 527

Optical zoom, 292

Orange Pi, 327, 367

Oriented fast and rotated brief algorithm, 395

Overfitting, 142, 143, 176

Oxygen sensor, 108, 342, 348, 359, 524

P

Packages, 29, 41, 58, 62, 85, 91, 375, 377, 398,
401, 403, 417, 471

Partial differential equations, 94, 118

Pattern matching, 82, 83, 85, 86, 197, 236,
237, 238, 239, 336, 469, 470, 494

Pattern recognition, 60, 85, 91, 180, 197, 202,
236, 237, 247, 371, 372, 373, 376, 378, 393,
432, 434, 471, 494, 507, 508, 509

PCB inspection, 50, 51, 472

People counting, 150, 157, 159, 194

Perimeter detection, 27, 157

Peripheral vein imaging, 93, 108

Personal security, 364

Perspective transformation, 264

pH sensor, 360

Photo Pac, 256

Picture Archiving and communication systems
(PACS), 102

Pincushion distortion, 161

PIR sensor, 353, 363, 367

Pixel clock, 253, 272, 283, 284, 287, 288, 300,

Pixel count, 253, 288, 291, 375

Pixel density, 297

Pixel depth, 253, 288, 290, 291

Pixel replication zooming, 292

Pixel size, 70, 71, 72, 276, 277, 288, 289,

Pixel, 2, 8, 10, 11, 14, 15, 34, 45, 47, 52, 54,
56, 57, 63, 64, 65, 66, 70, 73, 74, 85, 90, 93,
104, 111, 112, 113, 119, 121, 122, 125, 126,
129, 135, 136, 140, 142, 153, 154, 160, 161,
162, 163, 165, 180, 183, 191, 198, 204, 205,
205, 207, 210, 211, 214, 215, 216, 217, 223,
225, 230, 231, 232, 235, 238, 242, 243, 244,
246, 248, 253, 256, 258, 259, 261, 265, 266,
271, 272, 273, 274, 287, 288, 290, 291, 292,

Point tracking, 397

Pollution sensor, 353

Pool graphic system, 20

Pooling layer, 93, 136, 137, 175, 390

Portable Collision Avoidance System, 344

Pose estimation, 383, 397

Position sensors, 339, 342

Power-law transformation, 208

Pressure sensor, 339, 341, 353, 356, 358, 360,
529

Prewitt operator, 213, 214, 215, 216

Printed sensor, 353, 355

Progressive Scan, 40, 259, 260, 261, 281,

Proximity sensors, 32, 338, 353, 367, 411

Q

Quantization, 131, 151, 166, 200, 299, 514

R

Radar, 340, 344, 348, 349, 350, 362, 377, 379,
411, 517, 518, 520

Raindrop sensor, 409

Raspberry Pi, 5, 317, 326, 367, 533,

Real time radiography, 123

- Recognition, 3, 18, 20, 21, 22, 23, 44, 58, 60, 68, 79, 81, 83, 85, 90, 91, 103, 108, 116, 119, 135, 149, 150, 151, 153, 158, 167, 175, 180, 190, 194, 197, 202, 203, 237, 242, 243, 244, 247, 249, 331, 336, 349, 357, 367, 369, 371, 372, 373, 375, 376, 378, 381, 382, 383, 388, 393, 394, 395, 396, 397, 398, 401, 404, 418, 423, 426, 427, 428, 430, 432, 433, 434, 439, 440, 456, 459, 460, 471, 476, 481, 489, 490, 491, 494, 495, 502, 507, 508, 509, 517, 530, 531, 533
- Rectified linear unit (RELU) layer, 136, 137
- Rectified linear unit, 169
- Recurrent neural networks RNN, 135, 168
- Region of interest, 91, 128, 239, 500, 501, 512, 520
- Reinforcement learning, 141, 423, 434, 471
- Remote sensing, 95, 101, 103, 197, 202, 206, 246
- Repeatability, 30, 82, 83, 85, 313, 366, 375
- Resolution, 3, 41, 60, 66, 67, 70, 75, 82, 85, 86, 94, 104, 114, 128, 130, 133, 153, 162, 165, 184, 232, 238, 240, 253, 256, 259, 260, 271, 275, 277, 278, 282, 285, 286, 288, 289, 290, 291, 296, 298, 305, 306
- Restoration, 44, 53, 96, 152, 154, 197, 202, 249, 383, 398
- Revision stack, 177, 178, 179
- RFID sensor, 353, 354
- RGB color model, 248, 268, 374
- RGB, 71, 72, 97, 163, 166, 232, 233, 248, 268, 270, 277, 282, 374, 429, 490, 505, 520
- Robinson compass mask, 213, 217
- Robot vision, 59, 63, 197, 202, 369, 371, 373, 378, 407
- Robotic surgery, 110
- Robotic vision, 369, 404, 408, 411, 526
- Robotics, 24, 31, 40, 41, 47, 58, 80, 179, 186, 369, 370, 372, 393, 398, 403, 407, 410, 416, 432, 440, 460, 461
- RTOS, 179, 326,
- Sampling, 118, 181, 200, 240, 253, 283, 286, 289, 290, 291, 504
- Scale invariant feature transform algorithm, 384
- Scanning, 33, 54, 71, 72, 98, 106, 247, 260, 277, 390, 429, 451, 455, 459, 461, 497, 501
- Security in EV, 462
- Segmentation, 50, 53, 56, 63, 89, 95, 97, 110, 114, 119, 120, 121, 122, 132, 133, 135, 139, 140, 141, 162, 165, 175, 248, 249, 323, 375, 376, 380, 392, 393, 404, 426, 476, 494, 501, 502, 507, 512
- Semantic segmentation,
- Sensor size, 41, 272, 277, 278
- Sensor taps, 281
- Server, 159, 321, 323, 352, 435, 440, 471, 519
- Shape model, 52, 475, 480, 482
- Shapes, 41, 188, 211, 239, 480, 481, 482, 490, 491, 494
- Sharpening, 90, 197, 202, 203, 210, 213, 227, 248, 478
- Short wave infrared (SWIR), 275
- Shutter speed, 24, 257, 278, 280, 281
- SIFT algorithm, 386, 387, 393, 396
- Sigma filter, 126
- Signal processing, 5, 8, 12, 24, 101, 123, 198, 230, 247, 283, 285, 290, 329, 330, 369, 3790, 371, 372, 373, 377, 378, 408, 436, 494, 519, 524
- Signal to noise ratio, 72, 100, 113, 128, 130, 165, 232, 253, 275, 276, 285, 291, 452
- Signal, 5, 8, 11, 12, 24, 34, 67, 72, 84, 94, 95, 99, 100, 101, 113, 123, 126, 128, 130, 163, 165, 197, 198, 199, 200, 209, 223, 224, 230, 231, 232, 237, 247, 248, 253, 258, 259, 260, 261, 263, 265, 273, 275, 279, 283, 285, 287, 289, 290, 291, 300, 302, 303, 304, 310, 329, 330, 339, 340, 342, 349, 372
- SLAM, 20
- Sliding windows, 390, 485
- Smart camera, 14, 24, 37, 39, 40, 41, 48
- Smart fashion. 188
- Smart sensor, 352
- Sobel operator, 489, 490, 534
- SoC, 6, 15, 177, 179, 186, 317, 330, 435, 438, 463, 465, 466, 467, 516
- Safety in EV, 462
- Safety Standards, 424, 463

Software tools, 34, 39, 56, 60, 84, 89, 90, 331, 443, 469
 Sonar, 14, 359, 360
 Spark knock sensor, 348
 Spatial domain, 223, 224, 226, 480
 Spatial filter, 232, 250
 Spatial resolution, 66, 67, 70, 73, 271, 289, 296
 Spectral properties, 281
 Spectral response, 276, 281
 Speech recognition, 175, 190, 432, 471
 Speed up robust feature algorithm (SURF), 420
 SR-SCARA-Pro, 412
 Stereo disparity image, 438
 Stereo vision, 57, 64, 307, 385, 423, 437, 438, 439, 444, 445, 446, 447, 448, 456, 457, 461, 462, 472
 Stereoscopic endoscope, 98
 Stereoscopic microscope, 108
 Structured light, 14, 33, 34, 423, 437, 438, 450, 457, 462, 529,
 Subsampling, 169, 286
 Sum of absolute difference, 243
 Sum of squared difference, 241, 243
 Super resolution, 153, 162, 335
 SuperGather, 334
 Supervised learning, 135, 141, 407, 423, 433, 471, 483
 Support vector machines, 170, 172
 System, 199

T

Tachometer, 340, 342
 TDI, 262
 Temperature sensor, 329, 341, 348, 353, 411
 Template matching, 55, 63, 197, 239, 240, 241, 243, 244, 245, 246, 247, 375, 507
 Temporal filter, 110, 232, 250
 Testing, 34, 36, 65, 97, 115, 140, 143, 144, 149, 174, 176, 193, 307, 340, 357, 369, 407, 411, 412, 413, 414, 415, 416, 465, 511, 528
 Thermal fusion image, 308
 Thermal imaging camera, 253, 254, 307, 308, 309

Thinning Morphological, 475, 488
 Thinning, 417, 475, 488, 489
 Three sensor color imaging, 72
 Time of flight camera, 61, 92, 451
 Tomographic sensor, 363
 Tracing, 76, 77, 334, 501
 Training, 42, 52, 55, 81, 86, 103, 122, 135, 137, 138, 139, 140, 141, 142, 143, 144, 149, 168, 169, 170, 171, 173, 174, 176, 180, 181, 182, 184, 187, 188, 191, 192, 194, 323, 378, 389, 390, 405, 407, 427, 431, 434, 442, 443, 471, 482, 509
 Transformation, 101, 111, 119, 131, 182, 197, 203, 205, 206, 207, 208, 223, 224, 249
 Triangulation, 34, 57, 59, 60, 63, 423, 438, 444, 447, 450, 459, 461, 472
 Triggering, 24, 40, 47, 67, 70, 253, 263, 279, 283, 287, 307, 471
 TV lines, 253, 259, 288, 291

U

Ultrasonic imaging system, 93, 99
 Ultrasound sensor, 351
 Underwater robots, 417
 Unmanned aerial vehicle, 362, 502
 Unsupervised learning, 135, 141, 423, 434, 471, 483
 USB, 5, 6, 14, 46, 68, 87, 253, 290, 300, 303, 306, 325, 326, 330, 336, 374, 436, 530

V

Validation set, 140
 Vehicle track, 514, 515
 Vertical image recognition, 434
 Vibration sensor, 342
 Video analytics algorithms, 160–168
 Video content analysis, 26, 150, 168
 Video processing, 13, 202, 374, 436, 517, 519, 527
 Vision DSP, 330, 331, 333, 334, 335
 Vision guided robots, 29
 Vision measurement, 529–530
 Vision P6 DSP, 330, 331, 332
 Vision pipeline, 11, 12, 16, 381
 Vision Q6 DSP, 330, 331, 332

Vision software libraries, 436, 437
Visual sensor, 492, 493
Voltage sensor, 348
VxWorks, 179

W

Wearable sensor, 354, 367
Web inspection, 69
Weighted average filter, 211, 212
Weights, 137, 141
Wireless sensor, 340
Workstation, 19, 89, 102, 145, 321

X

X-ray imaging, 93, 95, 101, 202

Z

Zero order hold zooming, 293
Zooming, 96, 203, 211, 253, 291, 292, 293,
294, 295, 296, 310,