GLOBAL EDITION

Student's Manual

Digital Image Processing

Fourth Edition

Rafael C. Gonzalez • Richard E. Woods

Digital Image Processing Global Edition

4th Edition

Student Homework Problem Solutions Manual

Version 1.0

Rafael C. Gonzalez
Richard E. Woods

Pearson/Prentice Hall
Hoboken, NJ

www.ImageProcessingPlace.com

Detailed Table of Book Contents

Contents

Introduction 17

1.1 What is Digital Image Processing? 18
1.2 The Origins of Digital Image Processing 19
1.3 Examples of Fields that Use Digital Image Processing 23
Gamma-Ray Imaging24
X-Ray Imaging 24
Imaging in the Ultraviolet Band 27
Imaging in the Visible and Infrared Bands 28
Imaging in the Microwave Band 33
Imaging in the Radio Band 34
Other Imaging Modalities 35
1.4 Fundamental Steps in Digital Image Processing 41
1.5 Components of an Image Processing System 44

Digital Image Fundamentals 47

2.1 Elements of Visual Perception 48
Structure of the Human Eye 48
Image Formation in the Eye 50
Brightness Adaptation and Discrimination 50
2.2 Light and the Electromagnetic Spectrum 54
2.3 Image Sensing and Acquisition 57
Image Acquisition Using a Single Sensing Element 58
Image Acquisition Using Sensor Strips 60
Image Acquisition Using Sensor Arrays 61
A Simple Image Formation Model 61
2.4 Image Sampling and Quantization 63
Basic Concepts in Sampling and Quantization63
Representing Digital Images 65
Linear vs. coordinate Indexing 70
Spatial and Intensity Resolution 71
Image Interpolation 77
2.5 Some Basic Relationships Between Pixels 79
Neighbors of a Pixel 79
Adjacency, Connectivity, Regions, and Boundaries 79
Distance Measures 82
2.6 Introduction to the Basic Mathematical Tools Used in Digital Image Processing 83

ElementWise versus Matrix Operations 83
Linear versus Nonlinear OperationS 84
Arithmetic Operations 85
Set and Logical Operations 91
Basic Set Operations 91
Logical Operations 96
Spatial Operations 98
Single-Pixel Operations 99
Neighborhood Operations 99
Geometric Transformations 100
Image Registration 103
Vector and Matrix Operations 106
Image Transforms 109
Image Intensities as random variables 112

Intensity Transformations and Spatial Filtering 119

3.1 Background 120
The basics of Intensity Transformations and Spatial Filtering 120
About the Examples in this Chapter 122

3.2 Some Basic Intensity Transformation Functions 122
Image Negatives 122
Log Transformations 124
Power-Law (Gamma) Transformations 125
Piecewise Linear Transformation Functions 128
Contrast Stretching 129
Intensity-Level Slicing 130
Bit-Plane Slicing 131

3.3 Histogram Processing 133
Histogram Equalization 134
Histogram Matching (Specification) 140
Local Histogram Processing 149
Using Histogram Statistics for Image Enhancement 150

3.4 Fundamentals of Spatial Filtering 153
The Mechanics of LINEAR Spatial Filtering 154
Spatial Correlation and Convolution 154
Separable Filter Kernels 161
Some Important Comparisons Between Filtering in the Spatial and Frequency Domains 162
A Word about HOW Spatial Filter kernels

ARE Constructed 164
3.5 Smoothing (Lowpass) Spatial Filters 164
Box Filter Kernels 165
Lowpass Gaussian Filter Kernels 166
Order-Statistic (Nonlinear) Filters 174
3.6 Sharpening (Highpass) Spatial Filters 175
Foundation 176
Using the Second Derivative for Image Sharpening—The Laplacian 178
Unsharp Masking and Highboost Filtering 182
Using First-Order Derivatives for Image Sharpening—The Gradient 184
3.7 Highpass, Bandreject, and Bandpass Filters from Lowpass Filters 188
3.8 Combining Spatial Enhancement Methods 191

Filtering in the Frequency Domain 203

4.1 Background 204
A Brief History of the Fourier Series and Transform 204
About the Examples in this Chapter 206
4.2 Preliminary Concepts 207
Complex Numbers 207
Fourier Series 208
Impulses and their Sifting Properties 208
The Fourier Transform of Functions of One Continuous Variable 210
Convolution 213
4.3 Sampling and the Fourier Transform of Sampled Functions 215
Sampling 215
The Fourier Transform of Sampled Functions 216
The Sampling Theorem 217
Aliasing 221
Function Reconstruction (Recovery) from Sampled Data 224
4.4 The Discrete Fourier Transform of One Variable 225
Obtaining the DFT from the Continuous Transform of a Sampled Function 225
Relationship Between the Sampling and Frequency Intervals 228
4.5 Extensions to Functions of Two Variables 230
The 2-D Impulse and Its Sifting Property 230

The 2-D Continuous Fourier Transform Pair 231
2-D Sampling and the 2-D Sampling Theorem 231
Aliasing in Images 233
Extensions from 1-D Aliasing 233
Image Resampling and Interpolation 237
Aliasing and Moiré Patterns 238
The 2-D Discrete Fourier Transform and Its Inverse 240
4.6 Some Properties of the 2-D DFT and IDFT 240
Relationships Between Spatial and Frequency Intervals 240
Translation and Rotation 241
Periodicity 241
Symmetry Properties 243
Fourier Spectrum and Phase Angle 249
The 2-D Discrete Convolution Theorem 253
Summary of 2-D Discrete Fourier Transform Properties 257
4.7 The Basics of Filtering in the Frequency Domain 260
Additional Characteristics of the Frequency Domain 260
Frequency Domain Filtering Fundamentals 261
Summary of Steps for Filtering in the Frequency Domain266
Correspondence Between Filtering in the Spatial and Frequency Domains 268
4.8 Image Smoothing Using Lowpass Frequency Domain Filters 272
Ideal Lowpass Filters 273
Gaussian Lowpass Filters 277
Butterworth Lowpass Filters 278
Additional Examples of Lowpass Filtering 281
4.9 Image Sharpening Using Highpass Filters 284
Ideal, Gaussian, and Butterworth Highpass Filters from Lowpass Filters 284
The Laplacian in the Frequency Domain 289
Unsharp Masking, High-boost Filtering, and High-Frequency-Emphasis Filtering 291
Homomorphic Filtering 293
4.10 Selective Filtering 296
Bandreject and Bandpass Filters 297
Notch Filters 299
4.11 The Fast Fourier Transform 303
Separability of the 2-D DFT 303

Computing the IDFT Using a DFT Algorithm 304
The Fast Fourier Transform (FFT) 304

Image Restoration and Reconstruction 317

5.1 A Model of the Image Degradation/Restoration process 318

5.2 Noise Models 318
Spatial and Frequency Properties of Noise 319
Some Important Noise Probability Density Functions 319
Gaussian Noise 319
Rayleigh Noise 320
Erlang (Gamma) Noise 321
Exponential Noise 321
Uniform Noise 321
Salt-and-Pepper Noise 322
Periodic Noise 324
Estimating Noise Parameters 325

5.3 Restoration in the Presence of Noise Only—Spatial Filtering 327
Mean Filters 328
Arithmetic Mean Filter 328
Geometric Mean Filter 328
Harmonic Mean Filter 329
Contraharmonic Mean Filter 329
Order-Statistic Filters 330
Median Filter 330
Max and Min Filters 332
Midpoint Filter 332
Alpha-Trimmed Mean Filter 332
Adaptive Filters 333
Adaptive, Local Noise Reduction Filter 336
Adaptive Median Filter 338

5.4 Periodic Noise Reduction Using Frequency Domain Filtering 340
More on Notch Filtering 341
Optimum Notch Filtering 345

5.5 Linear, Position-Invariant Degradations 348

5.6 Estimating the Degradation Function 352
Estimation by Image Observation 352
Estimation by Experimentation 352
Estimation by Modeling 353

5.7 Inverse Filtering 356

5.8 Minimum Mean Square Error (Wiener) Filtering 358

5.9 Constrained Least Squares Filtering 363
5.10 Geometric Mean Filter 367
5.11 Image Reconstruction from Projections 368
Introduction 368
Principles of X-ray Computed Tomography (CT) 370
Projections and the Radon Transform 374
Backprojections 377
The Fourier-Slice Theorem 379
Reconstruction Using Parallel-Beam Filtered Backprojections 380
Reconstruction Using Fan-Beam Filtered Backprojections 386

Color Image Processing 399

6.1 Color Fundamentals 400
6.2 Color Models 405
The RGB Color Model 407
The CMY and CMYK Color Models 408
The HSI Color Model 411
Converting Colors from RGB to HSI 413
Converting Colors from HSI to RGB 415
Manipulating HSI Component Images 417
A Device Independent Color Model 418
6.3 Pseudocolor Image Processing 420
Intensity Slicing and Color Coding 420
Intensity to Color Transformations 423
6.4 Basics of Full-Color Image Processing 429
6.5 Color Transformations 430
Formulation 430
Color Complements 434
Color Slicing 436
Tone and Color Corrections 437
Histogram Processing of Color Images 439
6.6 Color Image Smoothing and Sharpening 442
Color Image Smoothing 442
Color Image Sharpening 444
6.7 Using Color in Image Segmentation 445
Segmentation in HSI Color Space 446
Segmentation in RGB Space 446
Color Edge Detection 450
6.8 Noise in Color Images 452

6.9 Color Image Compression 455

Wavelet and Other Image Transforms 463

7.1 Preliminaries 464
7.2 Matrix-based Transforms 466
Rectangular Arrays 472
Complex Orthonormal Basis Vectors 473
Biorthonormal Basis Vectors 474
7.3 Correlation 478
7.4 Basis Functions in the Time-Frequency Plane 479
7.5 Basis Images 483
7.6 Fourier-Related Transforms 484
The Discrete hartley Transform 485
The Discrete Cosine Transform 487
The Discrete Sine Transform 492
7.7 Walsh-Hadamard Transforms 496
7.8 Slant Transform 500
7.9 Haar Transform 502
7.10 Wavelet Transforms 504
Scaling Functions 505
Wavelet Functions 507
Wavelet Series Expansion 510
Discrete Wavelet Transform in One Dimension 512
The Fast Wavelet Transform 513
Wavelet Transforms in Two Dimensions 520
Wavelet Packets 526

Image Compression and Watermarking 539

8.1 Fundamentals 540
Coding Redundancy 541
Spatial and Temporal Redundancy 543
Irrelevant Information 544
Measuring Image Information 545
Shannon's First Theorem 546
Fidelity Criteria 547
Image Compression Models 549
The Encoding or Compression Process 550
The Decoding or Decompression Process 551
Image Formats, Containers, and Compression Standards 551

8.2 Huffman Coding 553
8.3 Golomb Coding 556
8.4 Arithmetic Coding 561
Adaptive context dependent probability estimates 562
8.5 LZW Coding 564
8.6 Run-length Coding 566
One-dimensional CCITT compression568
Two-dimensional CCITT compression 568
8.7 Symbol-based Coding 572
JBIG2 compression 573
8.8 Bit-plane Coding 575
8.9 Block Transform Coding 576
Transform selection 577
Subimage size selection 582
Bit allocation 583
Zonal Coding Implementation 584
Threshold Coding Implementation 585
JPEG588
8.10 Predictive Coding 594
Lossless predictive coding 594
Motion compensated prediction residuals 599
Lossy predictive coding 605
Optimal predictors 609
Optimal quantization611
8.11 Wavelet Coding 614
Wavlet selection 615
Decomposition level selection 616
Quantizer design 617
JPEG-2000 618
8.12 Digital Image Watermarking 624

Morphological Image Processing 635

9.1 Preliminaries 636
9.2 Erosion and Dilation 638
Erosion 639
Dilation 641
Duality 644
9.3 Opening and Closing 644
9.4 The Hit-or-Miss Transform 648
9.5 Some Basic Morphological Algorithms 652

Boundary Extraction 653
Hole Filling 653
Extraction of Connected Components 655
Convex Hull 657
Thinning 660
Thickening 660
Skeletons 662
Pruning 664
9.6 Morphological Reconstruction 667
Geodesic Dilation and Erosion 667
Morphological Reconstruction by Dilation and by Erosion 668
Sample Applications 669
Opening by Reconstruction 670
Automatic Algorithm for Filling Holes 671
Border Clearing 672
9.7 Summary of Morphological Operations on Binary Images 673
9.8 Grayscale Morphology 674
Grayscale Erosion and Dilation 674
Grayscale Opening and Closing 680
Some Basic GrayScale Morphological Algorithms 682
Morphological Smoothing 682
Morphological Gradient 682
Top-Hat and Bottom-Hat Transformations 683
Granulometry 685
Textural Segmentation 687
Grayscale Morphological Reconstruction 688

Image Segmentation 699

10.1 Fundamentals 700
10.2 Point, Line, and Edge Detection 701
Background 702
Detection of Isolated Points 706
Line Detection 707
Edge Models 710
Basic Edge Detection 716
The Image Gradient and Its Properties 716
Gradient Operators 717
Combining the Gradient with Thresholding 722
More Advanced Techniques for Edge Detection 724
The Marr-Hildreth Edge Detector 724

The Canny Edge Detector 729
Linking Edge Points 735
Local Processing 735
Global Processing Using the Hough Transform 737
10.3 Thresholding 742
Foundation 742
The Basics of Intensity Thresholding 743
The Role of Noise in Image Thresholding 744
The Role of Illumination and Reflectance in Image Thresholding 745
Basic Global Thresholding 746
Optimum Global Thresholding Using Otsu's Method 747
Using Image Smoothing to Improve Global Thresholding 752
Using Edges to Improve Global Thresholding 753
Multiple Thresholds 757
Variable Thresholding 761
Variable Thresholding Based on Local Image Properties 761
Variable Thresholding Based on Moving Averages 763
10.4 Segmentation by Region Growing and by Region Splitting and Merging 764
Region Growing 764
Region Splitting and Merging768
10.5 Region Segmentation Using Clustering and Superpixels 770
Region Segmentation using K-Means Clustering 770
Region Segmentation Using Superpixels 772
SLIC Superpixel Algorithm 774
Specifying the Distance Measure 776
10.6 Region Segmentation Using Graph Cuts 777
Images As Graphs 778
Minimum Graph Cuts 780
Computing Minimal Graph Cuts 783
Graph Cut Segmentation Algorithm 785
10.7 Segmentation Using Morphological Watersheds 786
Background 786
Dam Construction 789
Watershed Segmentation Algorithm 791
The Use of Markers 793
10.8 The Use of Motion in Segmentation 796
Spatial Techniques 796
A Basic Approach 796
Accumulative Differences 797

Establishing a Reference Image 798
Frequency Domain Techniques 799

Feature Extraction 811

11.1 Background 812
11.2 Boundary Preprocessing 814
Boundary Following (Tracing) 814
Chain Codes 816
Freeman Chain Codes 816
Slope Chain Codes 819
Boundary Approximations Using Minimum-Perimeter Polygons 821
Foundation 821
MPP Algorithm 822
Signatures 826
Skeletons, Medial Axes, and Distance Transforms 828
11.3 Boundary Feature Descriptors 831
Some Basic Boundary Descriptors 832
Shape Numbers 834
Fourier Descriptors 835
Statistical Moments 839
11.4 Region Feature Descriptors 840
Some Basic Descriptors 840
Topological Descriptors 843
Texture 846
Statistical Approaches 846
Spectral Approaches 855
Moment Invariants 858
11.5 Principal Components as Feature Descriptors 859
11.6 Whole-Image Features 868
The Harris-Stephens Corner Detector 869
Maximally Stable Extremal Regions (MSERs) 876
11.7 Scale-Invariant Feature Transform (SIFT) 881
Scale Space 883
Detecting Local Extrema 885
Finding the Initial Keypoints 885
Improving the Accuracy of Keypoint Locations 887
Eliminating Edge Responses 889
Keypoint Orientation 890
Keypoint Descriptors 892
Summary of the SIFT Algorithm 894

Image Pattern Classification 903

12.1 Background 904

12.2 Patterns and Pattern Classes 906
Pattern Vectors 906
Structural Patterns 908

12.3 Pattern Classification by Prototype Matching 910
Minimum-Distance Classifier 910
Using Correlation for 2-D prototype matching 915
Matching SIFT Features 917
Matching Structural Prototypes 919
Matching Shape Numbers 919
String Matching 920

12.4 Optimum (Bayes) Statistical Classifiers 923
Derivation of the Bayes Classifier 923
Bayes Classifier for Gaussian Pattern Classes 925

12.5 Neural Networks and Deep Learning 931
Background 931
The Perceptron 934
Multilayer Feedforward Neural Networks 943
Model of an Artificial Neuron 943
Interconnecting Neurons to Form a Fully Connected Neural Network 945
Forward Pass Through a Feedforward Neural Network 948
The Equations of a Forward Pass 948
Matrix Formulation 950
Using Backpropagation to Train Deep Neural Networks 953
The Equations of Backpropagation 953
Matrix Formulation 956

12.6 Deep Convolutional Neural Networks 964
A Basic CNN Architecture 964
Basics of How a CNN Operates 965
Neural Computations in a CNN 971
Multiple Input Images 973
The Equations of a Forward Pass Through A CNN 973
The Equations of Backpropagation used to Train CNNs 974

12.7 Some Additional Details of Implementation 987

Bibiography 995

1 Introduction

1.1 Background

This manual contains solutions to the problems marked with an asterisk (*) in Digital Image Processing, Global Edition, 4th Edition.

1.2 The Book Website

Digital Image Processing is a completely self-contained book. However, the companion website offers additional support in a number of important areas. The URL of the site is

www.ImageProcessingPlace.com

For Students or Independent Reader the site contains

- Reviews in areas such as probability, statistics, vectors, and matrices.
- A Tutorials section containing dozens of tutorials on topics relevant to the material in the book.
- An image database containing all the images in the book, as well as many other image databases.

For Instructors the site contains

- An Instructor's Manual with complete solutions to all the problems in the book.
- The manual is available free of charge to instructors who have adopted the book for classroom use.
- Classroom presentation materials in PowerPoint format.
- Material removed from previous editions, downloadable in convenient PDF format.
- Numerous links to other educational resources.

For the Practitioner the site contains additional specialized topics such as

- Links to commercial sites.
- Selected new references.
- Links to commercial image databases.

The website is an ideal tool for keeping the book current between editions by including new topics, digital images, and other relevant material that has appeared after the book was published. Although considerable care was taken in the production of the book, the website is also a convenient repository for any errors discovered between printings.

1.3 The DIP4E Support Packages

In this edition, we created support packages for students and faculty to organize all the classroom support materials available for the new edition of the book into one easy download. The Student Support Package contains most of the original images in the book and answers to homework problems marked with as asterisk (*) in the book. The Faculty Support Package contains solutions to all exercises, teaching suggestions, and all the art in the book in modifiable PowerPoint slides. One support package is made

available with every new book, free of charge. Applications for the support packages are submitted at the book website.

Chapter 2

Problem Solutions

Problem 2.2

The diameter, x, of the retinal image corresponding to the dot is obtained from similar triangles, as shown in Fig. P2.3. That is,

$$\frac{d/2}{0.2} = \frac{x/2}{0.017}$$

which gives $x = 0.085d$. From the discussion in Section 2.1, and taking some liberties of interpretation, we can think of the fovea as a square sensor array having on the order of 337,000 elements, which translates into an array of approximately 580×580 elements. Assuming equal spacing between elements, this gives 580 elements and 579 spaces, for a total of 1,159, on a line 1.5 mm long. The size of each element and each space is then $s = [(1.5\text{ mm}/1,159] = 1.3 \times 10^{-6}$ m. If the size (on the fovea) of the imaged dot is less than the size of a single resolution element, we assume that the dot will be invisible to the eye. In other words, the eye will not detect a dot if its diameter, d, is such that $0.085(d) < 1.3 \times 10^{-6}$ m, or $d < 15.3 \times 10^{-6}$ m.

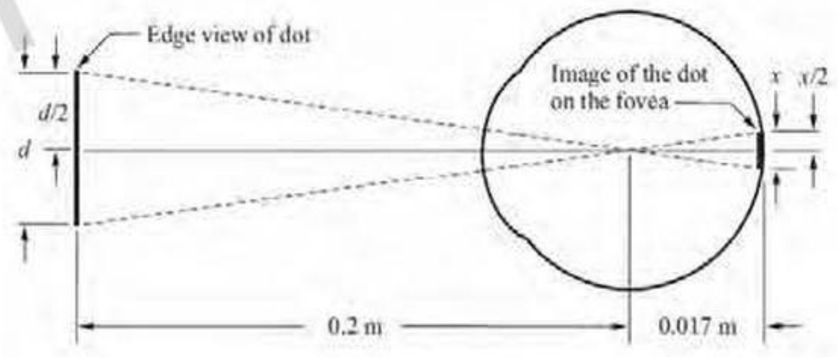


Figure P2.3

Problem 2.4

(a) From the discussion on the electromagnetic spectrum in Section 2.2, the source of the illumination required to see an object must have wavelength the same size or smaller than the object. Because interest lies only on the boundary shape and not on other spectral characteristics of the specimens, a single illumination source in the far ultraviolet (wavelength of .001 microns or less) will be able to detect all objects, and the smallest detail discernible will be .001 microns or less, the same as the wavelength of the illumination. A far-ultraviolet camera sensor would be needed to image the specimens.

Problem 2.5

(a) The vertical (or horizontal) dimension in which the image has to fit is 5 cm or 50 mm. So, we have to fit 2048 lines in 50 mm or approximately 41 lines/mm. Line pairs is half of that, or approximate 20 line pairs per mm.

Problem 2.6

From the geometry of Fig. 2.3, (7 mm)/(35 mm) = (z)/(500 mm) , or z = 100 mm. So the target size is 100 mm on the side. We have a total of 1024 elements per line, so the resolution of 1 line is 1024/100 = 10 elements/mm. For line pairs we divide by 2, giving an answer of 5 lp/mm.

Problem 2.8

Let D be the distance from the center of the camera lens to the area to be imaged, which is given as 1 m. Let H denote the height of the area to be image, which is given as 0.5 m. Let L denote the focal length of the lens, which is given as 200 mm. We want to find d, the height of the imaged area in the focal plane of the lens (this will be the minimum size of the CCD chip needed). Then, from the geometry of Fig. 2.3, $(H/D) = (d/L)$ or

$$d = (LH)/D = (200 \text{ mm})(500 \text{ mm})/1000 \text{ mm} = 100 \text{ mm}$$

Thus, the imaged area is of size 100 mm^2. We are given that the required resolution is 5 line pairs per mm, which means 10 pixels/mm in the vertical (and horizontal) directions. Because $d = 100$ mm, we need to be able to detect (10 pixels/mm)(100 mm) = 1000 pixels in each vertical line of the image. This means that the minimum resolution of our CCD chip is 1000×1000 pixels over a square of size 100×100 mm.

Problem 2.9

(a) The total amount of data (including the start and stop bits) in an 8-bit, 1024×1024 image is $(1024)^2 \times (8 + 2)$ bits. The total time required to transmit 500 such images over a 3 M baud modem is:

$$\text{Trans time} = 500 \times (1024)^2 \times (10)/(3 \times 10^6) = 1,748 \text{ sec}.$$

Problem 2.10

The width-to-height ratio is 16/9 and the resolution in the vertical direction is 1125 lines (or, what is the same thing, 1125 pixels in the vertical direction). It is given that the resolution in the horizontal direction is in the 16/9 proportion, so the resolution in the horizontal direction is $(1125) \times (16/9) = 2000$ pixels per line. The system "paints" a full, 1125×2000 8-bit image every 1/30 sec for each of the red, green, and blue component images. There are 7200 sec in two hours, so the total digital data generated in this time interval is $(1125)(2000)(8)(30)(3)(7200) = 1.166 \times 10^{13}$ bits, or 1.458×10^{12} bytes (i.e., about 1.5 terabytes). These figures show why image data compression (Chapter 8) is so important.

Problem 2.11

(a) From the problem statement,

$$x + M(y + Nz) = s$$

"Moding" both sides of this equation with respect to M gives

$$x = s \bmod M$$

where we used the fact on the left side of the first equation that all quantities are integers, and that the modulus of an integer plus an integer multiple of M is equal to the first integer.

Problem 2.12

The image in question is given by

$$\begin{aligned} f(x, y) &= i(x, y)r(x, y) \\ &= 255e^{-[(x-x_0)^2 + (y-y_0)^2]} \times 1.0 \\ &= 255e^{-[(x-x_0)^2 + (y-y_0)^2]} \end{aligned}$$

A cross section of the image is shown in Fig. P2.12(a). If the intensity is quantized using k bits, then we have the situation shown in Fig. P2.14(b), where $\Delta G = (255 + 1)/2^k$. Since an abrupt change of 8 intensity levels is assumed to be detectable by the eye, it follows that $\Delta G = 8 = 256/2^k$, or $k = 5$. In other words, 32, or fewer, intensity levels will produce visible false contouring.

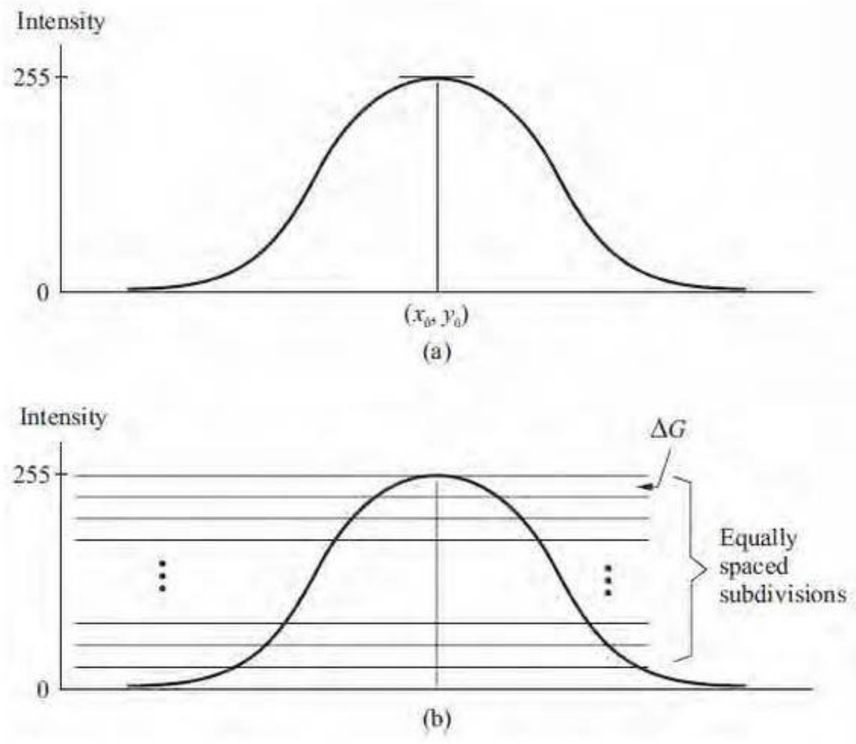


Figure P2.12

Problem 2.14

Let p and q be as shown in Fig. P2.14. Then,

(a) S_1 and S_2 are not 4-connected because q is not in the set $N_4(p)$.

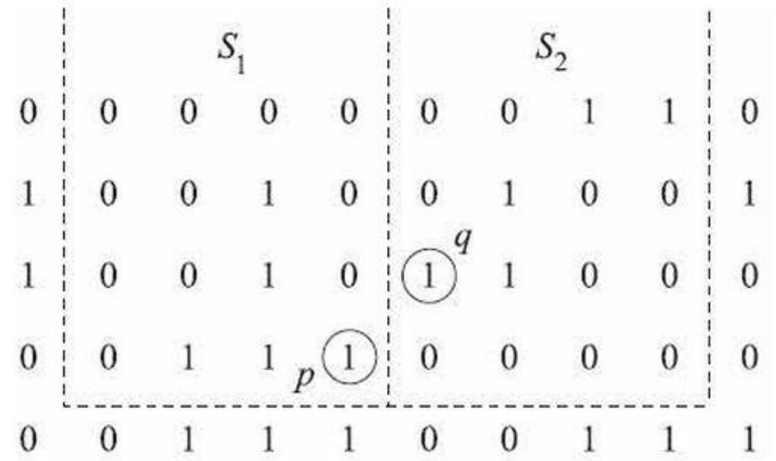


Figure P2.14

Problem 2.15

The solution of this problem consists of defining all possible neighborhood shapes to go from a diagonal segment to a corresponding 4-connected segment, as Fig. P2.15 illustrates. The algorithm then simply looks for the appropriate match every time a diagonal segment is encountered in the boundary.

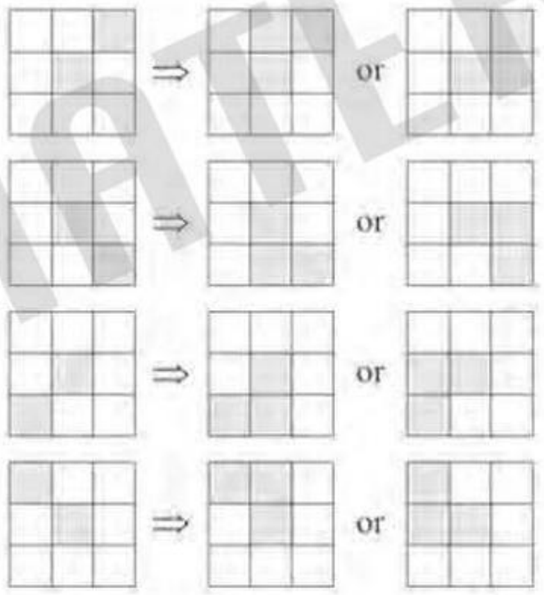


Figure P2.15

Problem 2.18

(a) When $V = \{0,1\}$ a 4-path does not exist between p and q because it is impossible to get from p to q by traveling along points that are both 4-adjacent and also have values from V. Figure P2.18(a) shows this condition; it is not possible to get to q. The shortest 8-path is shown in Fig. P2.18(b); its length is 4. The length of the shortest m-path (shown dashed) is 5. Both of these shortest paths are unique in this case.

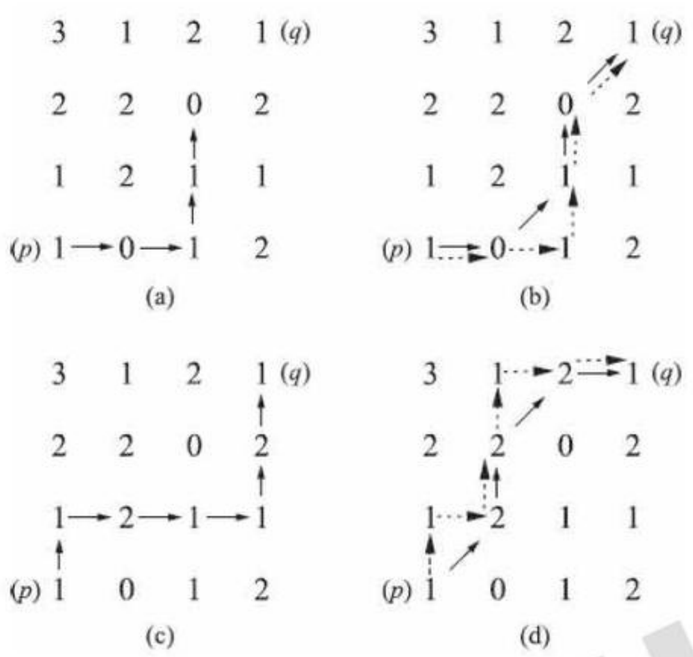


Figure P2.18

Problem 2.19

(a) A shortest 4-path between a point p with coordinates (x, y) and a point q with coordinates (s, t) is shown in Fig. P2.19, where the assumption is that all points along the path are from V. The lengths of the segments of the path are $|x - s|$ and $|y - t|$, respectively. The total path length is $|x - s| + |y - t|$, which we recognize as the definition of the D_4 distance, as given in Eq. (2-20). (Recall that this distance is independent of any paths that may exist between the points.) The D_4 distance obviously is equal to the length of the shortest 4-path when the length of the path is $|x - s| + |y - t|$. This occurs whenever we can get from p to q by following a path whose elements (1) are from V, and (2) are arranged in such a way that we can traverse the path from p to q by making turns in at most two directions (e.g., right and up).

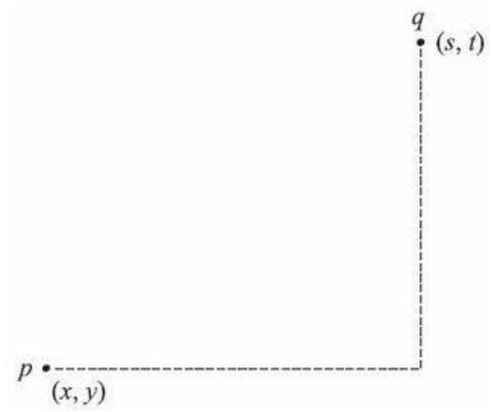


Figure P2.19

Problem 2.22

With reference to Eqs (2-22) and (2-23), let H denote the sum operator, let S_1 and S_2 denote two different small subimage areas of the same size, and let $S_1 + S_2$ denote the corresponding *elementwise* sum of the elements in S_1 and S_2, as explained in Section 2.6. The operator H computes the sum of pixel values in a neighborhood, and thus yields a scalar for a given neighborhood. Then, $H(aS_1 + bS_2)$ means: (1) multiply the pixels in each of the subimage areas by the constants shown, (2) add the pixel-by-pixel values from aS_1 and bS_2 (which produces a single subimage area), and (3) compute the sum of the values of all the pixels in

that single subimage area, which produces a scalar. Let ap_1 and bp_2 denote two arbitrary (but *corresponding*) pixels from $aS_1 + bS_2$. Then we can write

$$\begin{aligned} H(aS_1 + bS_2) &= \sum_{p_1 \in S_1 \text{ and } p_2 \in S_2} ap_1 + bp_2 \\ &= \sum_{p_1 \in S_1} ap_1 + \sum_{p_2 \in S_2} bp_2 \\ &= a\sum_{p_1 \in S_1} p_1 + b\sum_{p_2 \in S_2} p_2 \\ &= aH(S_1) + bH(S_2) \end{aligned}$$

which, according to Eq. (2-23), indicates that H is a linear operator.

Problem 2.23

(a) Addition: The left side of Eq. (2-23) is:

$$\begin{aligned} H[af_1 + bf_2] &= (af_1 + bf_2) + (af_1 + bf_2) \\ &= 2a(f_1) + 2b(f_2) \end{aligned}$$

The right side of Eq. (23) is

$$\begin{aligned} aH[f_1] + bH[f_2] &= a(f_1 + f_1) + b(f_2 + f_2) \\ &= 2af_1 + 2bf_2 \end{aligned}$$

The left and right sides are equal, so the summation operator is linear.

(c) Multiplication: The left side of Eq. (2-23) is:

$$\begin{aligned} H[af_1 + bf_2] &= (af_1 + bf_2) \times (af_1 + bf_2) \\ &= a^2 f_1 \times f_1 + 2abf_1 \times f_2 + b^2 f_2 \times f_2 \end{aligned}$$

where it is understood that, for example, $f_i \times f_j$ is the elementwise product of the two images. The right side of Eq. (2-23) is

$$aH[f_1] + bH[f_2] = a(f_1 \times f_1) + b(f_2 \times f_2)$$

The left does not equal the right so the multiplication operator is not linear.

Problem 2.25

The average of k images is

$$a(k) = \frac{1}{k}\sum_{j=1}^{k} f_j$$

and the average of $k+1$ images is

$$a(k+1) = \frac{1}{k+1}\sum_{j=1}^{k+1} f_j = \frac{1}{k+1}\left[\sum_{j=1}^{k} f_j + f_{k+1}\right]$$
$$= \frac{1}{k+1}\left[\frac{k}{k}\sum_{j=1}^{k} f_j + f_{k+1}\right]$$
$$= \frac{1}{k+1}\left[ka(k) + f_{k+1}\right]$$

Problem 2.26

(a) From Eq. (2-26), at any point (x, y),

$$\bar{g}(x, y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x, y) = \frac{1}{K}\sum_{i=1}^{K} f_i(x, y) + \frac{1}{K}\sum_{i=1}^{K} \eta_i(x, y)$$

Because this equation is applicable at *any* coordinates, (x, y), we can simplify the notation by dropping the coordinates. Then,

$$E\{\bar{g}\} = \frac{1}{K}\sum_{i=1}^{K} E\{f_i\} + \frac{1}{K}\sum_{i=1}^{K} E\{\eta_i\}$$

But all the f_i are the same, so $E\{f_i\} = f$. In other words, the noisy images are formed by adding noise to the same image. The noise changes from image to image, but f remains the same. Also, it is given that the noise has zero mean, so $E\{\eta_i\} = 0$. Thus, it follows that $E\{\bar{g}\} = f$, or $E\{\bar{g}(x, y)\} = f(x, y)$, which proves the validity of Eq. (2-27).

Problem 2.27

(a) Pixels have integer values, and 8 bits allow representation of 256 contiguous integer values. In our work, the range of intensity values for 8-bit images is $[0, 255]$. The subtraction of values in this range cover the range $[-255, 255]$. This range of values cannot be covered by 8 bits, but it is given in the problem statement that the result of subtraction has to be represented in 8 bits also, and thus are limited to the range $[0, 255]$. What this means is that any subtraction of 2 pixels that yields a negative quantity will be clipped at 0.

The process of repeated subtractions of an image $b(x, y)$ from an image $a(x, y)$ can be expressed as

$$d_K(x, y) = a(x, y) - \sum_{k=1}^{K} b(x, y)$$
$$= a(x, y) - Kb(x, y)$$

where $d_K(x, y)$ is the difference image resulting after K subtractions. Because image subtraction is an elementwise operation (see Section 2.6), we can focus attention on the subtraction of any corresponding pair of pixels in the images. We have already stated that negative results are clipped at 0. Once a 0 result is obtained, it will remain so because subtraction of any nonnegative value from 0 is a negative quantity

which, again, is clipped at 0. Similarly, any location (x_0, y_0) for which $b(x_0, y_0) = 0$, will produce the result $d_K(x_0, y_0) = a(x_0, y_0)$. That is, repeatedly subtracting 0 from any value results in that value. The locations in $b(x, y)$ that are not 0 will eventually decrease the corresponding values in $d_K(x, y)$ until they are 0. The maximum number of subtractions in which this takes place in the context of the present problem is 255, which corresponds to the condition at a location in which $a(x, y)$ is 255 and $b(x, y)$ is 1. Thus, we conclude from the preceding discussion that repeatedly subtracting an image from another (and representing the results in 8 bits) will result in a difference image whose components are 0 in the locations in $b(x, y)$ that are not zero and equal to the original values of $a(x, y)$ at the locations in $b(x, y)$ that are 0. This result will be achieved in, at most, 255 subtractions.

Problem 2.28

Let $g(x, y)$ denote the golden image, and let $f(x, y)$ denote any input image acquired during routine operation of the system. Change detection via subtraction is based on computing the difference $d(x, y) = g(x, y) - f(x, y)$. The resulting image, $d(x, y)$, can be used in two fundamental ways for change detection. One way is to use pixel-by-pixel analysis. In this case we say that $f(x, y)$ is "close enough" to the golden image if all the pixels in $d(x, y)$ fall within a specified threshold band $[T_{\min}, T_{\max}]$ where $T_{\min}$ is negative and $T_{\max}$ is positive. Usually, the same value of threshold is used for both negative and positive differences, so that we have a band $[-T, T]$ in which all pixels of $d(x, y)$ must fall in order for $f(x, y)$ to be declared acceptable. The second major approach is simply to sum all the pixels in $|d(x, y)|$ and compare the sum against a threshold Q. Note that the absolute value needs to be used to avoid errors canceling out in the sum. This is a much cruder test, so we will concentrate on the first approach.

There are three fundamental factors that need tight control for difference-based inspection to work: (1) proper registration, (2) controlled illumination, and (3) noise levels that are low enough so that difference values are not affected appreciably by variations due to noise. The first condition basically addresses the requirement that comparisons be made between corresponding pixels. Two images can be identical, but if they are displaced with respect to each other, comparing the differences between them makes no sense. Often, special markings are manufactured into the product for mechanical or image-based alignment.

Controlled illumination ("illumination" is not limited to visible light) obviously is important because changes in illumination can affect dramatically the values in a difference image. One approach used often in conjunction with illumination control is intensity scaling based on actual conditions. For example, the products could have one or more small patches of a tightly controlled color, and the intensity (and perhaps even color) of the pixels in the entire image would be modified based on the actual versus expected intensity and/or color of the patches in the image being processed.

Finally, the noise content of a difference image needs to be low enough so that it does not materially affect comparisons between the golden and input images. Good signal strength goes a long way toward reducing the effects of noise.

There are a number if variations of the basic theme just described. For example, additional intelligence in the form of tests that are more sophisticated than pixel-by-pixel threshold comparisons can be implemented. A technique used often in this regard is to subdivide the golden image into different regions

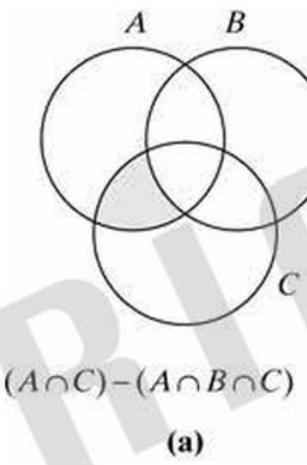
and perform different (usually more than one) tests in each of the regions, based on expected region content.

Problem 2.29

(a) $K = 2^k - 1$.

Problem 2.30

(a) See Fig. P2.30.



(a)

Figure P2.30

Problem 2.31

(a) See Fig. P2.31(a)

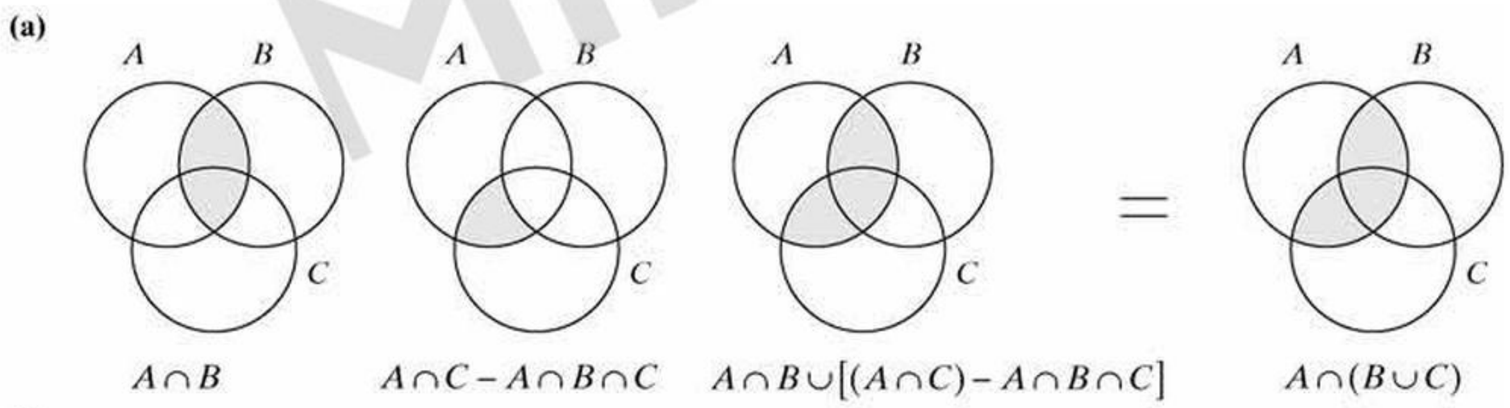


Figure P2.31

Problem 2.32

(a) $[(A \cap B) \cup (A \cap C] - A \cap B \cap C$.

Problem 2.33

(a) We have to show that the following three properties hold under the relation "less than or equal to."

Reflexivity: Any real number is related to itself by the relation $(\leq)$. In other words any real number is equal to itself.

Transitivity: For any three real numbers a, b, and c, if a is less than or equal to b, and b is less than or equal to c, then a is less than or equal to c. So, transitivity holds.

Antisymmetry: For any two real numbers a and b, if a is less than or equal to b, and b is less than or equal to a, then it follows that a and b are the same number.
Since all three properties hold, set S is a partially ordered set under the relation "less than or equal to."

(b) We have to show that the properties of antireflexivity and transitivity hold under the relation "less than."

Antireflexivity: Any real number cannot be less than itself, so this property holds.

Transitivity: For any three real numbers a, b, and c, if a is less than b, and b is less than c, then it follows that a must be less than c. So, transitivity holds.

Problem 2.34

To solve the problem, we have to show that the relation "divisible by" is reflexive, transitive, and antisymmetric.

(a) *Reflexivity:* any positive integer is divisible by itself (i.e., $k = 1$ in the problem statement) so this condition is satisfied.

Problem 2.36

(a)

$$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_x & 0 & c_x t_x \\ 0 & c_y & c_y t_y \\ 0 & 0 & 1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_x & 0 & c_x t_x \\ 0 & c_y & c_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_x \cos\theta & -c_x \sin\theta & c_x t_x \\ c_y \sin\theta & c_y \cos\theta & c_y t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Problem 2.37

(a) The forward scaling transformations is:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

and the corresponding inverse transformation is

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1/c_x & 0 & 0 \\ 0 & 1/c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

(d) The forward rotation transformation is

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

and the corresponding inverse rotation transformation is

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

(e) A composite translation/rotation transformation is

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

and the corresponding inverse transformation is

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & -t_x\cos - t_y\sin\theta \\ -\sin\theta & \cos\theta & t_x\sin\theta - t_y\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Note the order of the matrices in the forward *vs* the inverse composite transformations.

Problem 2.39

(a) The Fourier transformation kernel is separable because

$$\begin{aligned} r(x,y,u,v) &= e^{-j2\pi(ux/M+vy/N)} \\ &= e^{-j2\pi(ux/M)}e^{-j2\pi(vy/N)} \\ &= r_1(x,u)r_2(y,v) \end{aligned}$$

It is symmetric because

$$\begin{aligned} e^{-j2\pi(ux/M+vy/N)} &= e^{-j2\pi(ux/M)}e^{-j2\pi(vy/N)} \\ &= r_1(x,u)r_1(y,v). \end{aligned}$$

Problem 2.40

From Eq. (2-59) and the definition of separable kernels,

$$\begin{aligned} T(u,v) &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)r(x,y,u,v) \\ &= \sum_{x=0}^{M-1} r_1(x,u)\sum_{y=0}^{N-1} f(x,y)r_2(y,v) \\ &= \sum_{x=0}^{M-1} T(x,v)r_1(y,v) \end{aligned}$$

where

$$T(x,v) = \sum_{y=0}^{N-1} f(x,y)r_2(y,v)$$

For a fixed value of x, this equation is recognized as the 1-D transform along one row of $f(x,y)$. By letting x vary from 0 to $M-1$ we compute the entire array $T(x,v)$. Then, by substituting this array into the last line of the previous equation we have the 1-D transform along the columns of $T(x,v)$. In other words, when a kernel is separable, we can compute the 1-D transform along the rows of the image. Then we compute the 1-D transform along the columns of this intermediate result to obtain the final 2-D transform, $T(u,v)$. We obtain the same result by computing the 1-D transform along the columns of $f(x,y)$ followed by the 1-D transform along the rows of the intermediate result.

This result plays an important role in Chapter 4 when we discuss the 2-D Fourier transform. From Eq. (2-61), the 2-D Fourier transform is given by

$$T(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M + vy/N)}$$

We know from Problem 2.41 that the Fourier transform kernel is separable, so we can write this equation as

$$\begin{aligned} T(u,v) &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M + vy/N)} \\ &= \sum_{x=0}^{M-1} e^{-j2\pi(ux/M)}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(vy/N)} \\ &= \sum_{x=0}^{M-1} T(x,v)e^{-j2\pi(ux/M)} \end{aligned}$$

where

$$T(x,v) = \sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(vy/N)}$$

is the 1-D Fourier transform along the rows of $f(x,y)$, as we let $x = 0, 1, 2, \ldots, M-1$.

Chapter 3

Problem Solutions

Problem 3.2

(a) $$s = T(r) = \frac{1}{1 + (m/r)^E} \tag{1}$$

Problem 3.3

(a) The transformations required to produce the individual bit planes are mappings of the truth table for eight binary variables. In this truth table, the values of the 8th bit are 0 for image values 0 to 127, and 1 for image values 128 to 255. Thus a transformation function to generate the most significant (8th) bit plane would be $T(r) = 0$ for r in the range $[0, 127]$ and $T(r) = 1$ for r in the range $[128, 255]$.

Continuing with the truth table concept, the transformation required to produce an image of the 7th bit plane outputs a 0 for image values in the range [0, 63], a 1 for values in the range [64, 127], a 0 for values in the range [128, 191], and a 1 for values in the range [192, 255]. For the 6th bit plane, the transformation function outputs a 0 for image values in the range [0, 31], a 1 for values in the range [32, 63], 0 for values in the range [64, 91], and so forth. The transformation function to generate the 5th bit plane would generate a 0 for image values in the range [0, 15], a 1 for values in the range [16, 31], and so forth. A similar approach is used for the other bit planes. Finally, the output of the transformation for the lowest-order bit plane alternates between 0 and 1, depending on whether the byte values are even or odd.

As you can see, the number of "flips" in the transformation function increases as the order of the bit planes decreases. This is the reason why the lowest-order bit plane is the "busiest."

Problem 3.5

(a) The number of pixels having different intensity level values would decrease, thus causing the count in the lower-order bins in the histogram to decrease. Because the number of pixels would not change, this would cause the numebr of counts of the higher order bins to increase, resulting in taller histogram peaks in the higer values. This will brighten the image but its tonality would decrease.

Problem 3.9

(b) If none of the intensity levels r_k, $k = 1,2,\ldots,L-1$, are 0, then $T(r_k)$ will be strictly monotonic. This implies a one-to-one mapping both ways, meaning that both forward and inverse transformations will be single-valued.

Problem 3.10

The purpose of this simple problem is to make the student think of the meaning of histograms and arrive at the conclusion that histograms carry no information about spatial properties of images. Thus, the only time that the histogram of the images formed by the operations shown in the problem statement can be determined in terms of the original histograms is when one (both) of the images is (are) constant. In (d) we have the additional requirement that none of the pixels of $g(x,y)$ can be 0. It is given that the histograms are not normalized, so, for example, $h_f(r_k)$ is the number of pixels in $f(x,y)$ having intensity level r_k. Assume that all the pixels in $g(x,y)$ have constant value c. The pixels of both images are assumed to be positive. Finally, let u_k denote the intensity levels of the pixels of the images formed by any of the arithmetic operations given in the problem statement. Under the preceding set of conditions, the histograms are determined as follows:

(a) We obtain the histogram $h^+(u_k)$ of the sum of f and g by letting $u_k = r_k + c$, and also $h^+(u_k) = h_f(r_k)$ for all k. In other words, the values (height) of the components of h^+ are the same as the components of h_f, but their locations on the intensity axis are shifted right by an amount c.

Problem 3.11

(a) The histogram equalization transformation for the interval $[0, L-1]$:

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = \frac{2}{(L-1)}\int_0^r w\,dw = \frac{r^2}{(L-1)}$$

By definition, this transformation is 0 for values outside the range $[0, L-1]$. Squaring the values of the input intensities and dividing them by $(L-1)$ will produce an image whose intensities, s, have a uniform PDF because this is a histogram-equalization transformation, as discussed earlier.

(b) We are interested in an image with a specified histogram, so we find next

$$G(z) = (L-1)\int_0^z p_z(w)dw = \frac{3}{(L-1)^2}\int_0^r w^2\,dw = \frac{z^3}{(L-1)^2}$$

over the interval $[0, L-1]$; this function is 0 elsewhere by definition. Finally, we require that $G(z) = s$, but $G(z) = z^3/(L-1)^2$, so $z^3/(L-1)^2 = s$, and we have

$$z = G^{-1}(s) = \left[(L-1)^2 s\right]^{1/3}$$

Thus, if we multiply every histogram equalized pixel by $(L-1)^2$ and raise the product to the power $1/3$, the result will be an image whose intensities, z, have the PDF $p_z(z) = 3z^2/(L-1)^3$ in the interval $[0, L-1]$, as desired.

Problem 3.13

Yes. Transformation (2) is the mapping from s_k to z_q constructed based on Steps 2 and 3 of the histogram specification procedure. In Fig. 3.25(b) s_k is the ordinate and z_q the abscissa. Thus, the graph is presented as the standard way in which you would visualize a mapping from z_q to s_k. If you wanted to visualize the transformation from s_k to z_q constructed based on Steps 2 and 3 of the procedure, you would choose s_k as the abscissa and z_q as the ordinate. Figure 3.25(b) simply has combined two different orientations into one plot. Thus, the functions appear as mirror images of each other when shown in the same graph.

Problem 3.14

The value of the histogram component corresponding to the kth intensity level in a neighborhood is

$$p_r(r_k) = \frac{n_k}{n}$$

for $k = 0,1,2,\ldots,K-1$, where n_k is the number of pixels having intensity level r_k, n is the total number of pixels in the neighborhood, and K is the total number of possible intensity levels. Suppose that the neighborhood is moved one pixel to the right (we are assuming rectangular neighborhoods). This deletes the leftmost column and introduces a new column on the right. The updated histogram then becomes

$$p'_r(r_k) = \frac{1}{n}\left[n_k - n_{L_k} + n_{R_k}\right]$$

for $k = 0,1,2,\ldots,K-1$, where n_{L_k} is the number of occurrences of level r_k on the left column and n_{R_k} is the similar quantity on the right column. The preceding equation can be written also as

$$p'_r(r_k) = p_r(r_k) + \frac{1}{n}\left[n_{R_k} - n_{L_k}\right]$$

for $k = 0,1,2,\ldots,K-1$. The same concept applies to other modes of neighborhood motion:

$$p'_r(r_k) = p_r(r_k) + \frac{1}{n}\left[b_k - a_k\right]$$

$k = 0,1,2,\ldots,K-1$, where a_k is the number of pixels with value r_k in the neighborhood area deleted by the move, and b_k is the corresponding number introduced by the move.

Problem 3.17

Figure 3.28 redrawn with the kernel rotated by 180° :

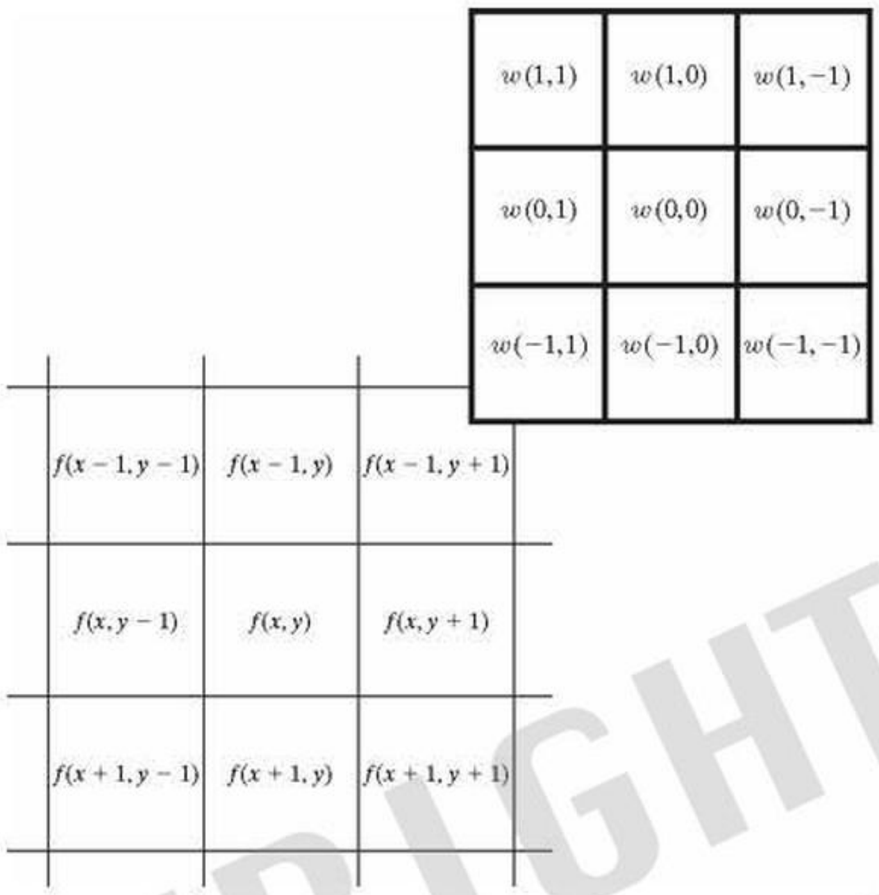


Figure P3.17

Expand Eq. (3-34):

$s = -1, \quad t = -1, 0, 1$

$$w(-1,-1)f(x+1,y+1) + w(-1,0)f(x+1,y) + w(-1,1)f(x+1,y-1)$$

$s = 0, \quad t = -1, 0, 1$

$$w(0,-1)f(x,y+1) + w(0,0)f(x,y) + w(0,1)f(x,y-1)$$

$s = 1, \quad t = -1, 0, 1$

$$w(1,-1)f(x-1,y+1) + w(1,0)f(x-1,y) + w(1,1)f(x-1,y-1)$$

We see that these results correspond to the sum of products for the arrangement in the preceding figure.

Problem 3.18

(a)

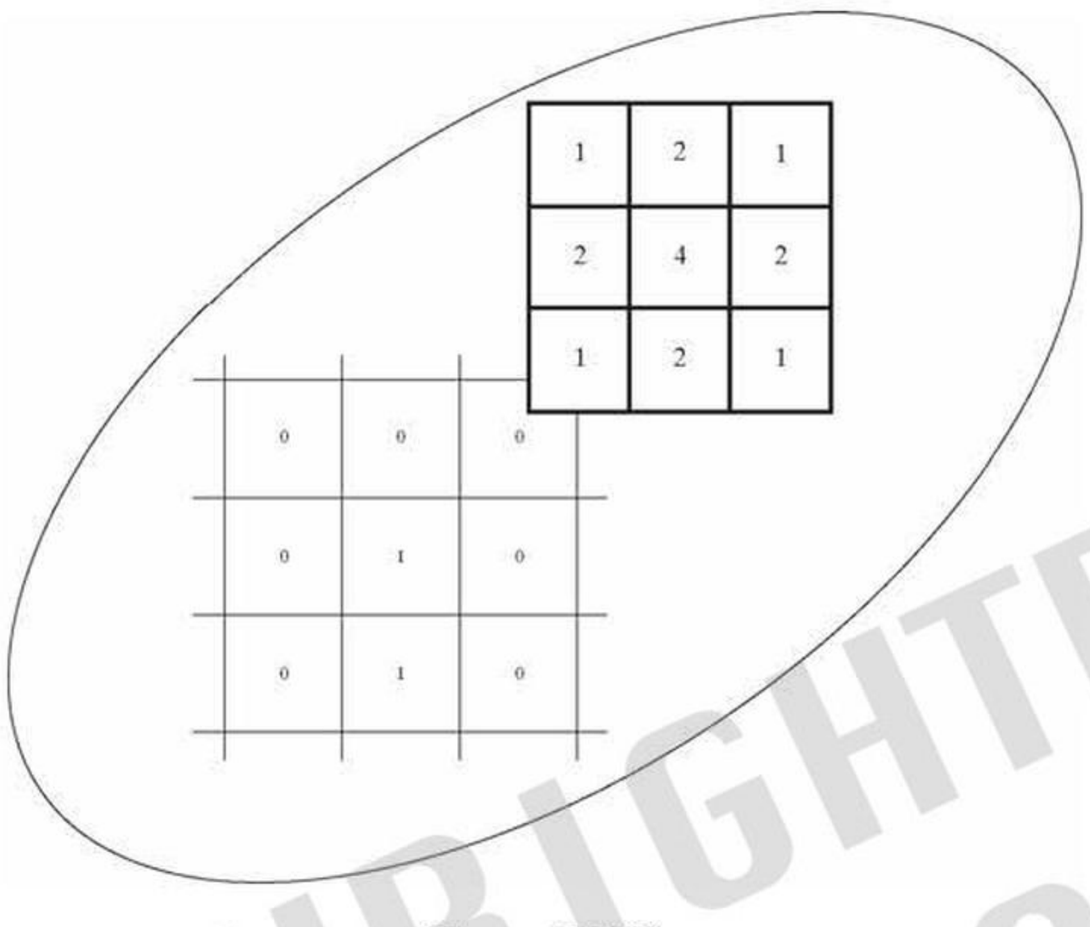


Figure P3.18

(b) Because the kernel is only of size 3×3, the minimum padding needed is a one-element border of 0s all around the image. This turns the $(2,3)$ point in the original image into $(3,4)$ in the padded image. We use the symbol f_p denote the 7×7 padded image:

$$f_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

From Eq. (3-34):

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

$$= \sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s,t) f(3-s, 4-t)$$

where the second equation corresponds to the kernel being centered at point $(3,4)$ of the padded image. The only sum-of-product values that are not 0 are $w(0,0) f_p(3,4) = (4)(1) = 4$ and $w(-1,0) f(4,4) = (2)(1) = 2$. Thus we have that the convolution value when the kernel is centered at $(2,3)$ is $(w \star f)(3,4) = 4 + 2 = 6$. The full convolution is:

$$(w \star f)(x,y) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 3 & 6 & 3 & 0 & 0 \\ 0 & 0 & 4 & 8 & 4 & 0 & 0 \\ 0 & 0 & 3 & 6 & 3 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Problem 3.19

The sizes of correlation and convolution are identical, so we prove the validity of Eqs. (3-36) and (3-37) only for correlation. Let w represent a kernel of size $m \times n$, and let and f be an image of size $M \times N$. We assume that $m \leq M$ and $n \leq N$. We prove the validity of S_h first. Suppose the the bottom right point of w is coincident with the origin of the image, which is at the top, left. To complete the correlation of w with the first row of f so the leftmost bottom point of w is coincident with with the rightmost point of the first row of f we have to shift w to the right a total of $n-1$ times. The width of the image is N, so the total width of the correlation result for the first row will be $N+n-1$. Of course the width of all rows is the same. Thus, as shown in Eq. (3-37), the width of the image resulting from correlation is $S_h = N+n-1$. A similar analsis in the vertical direction would show that the the height of the correlation result is $S_v = M+m-1$, as given in Eq. (3-36). Keep in mind that these results are applicable to the cofiguration in which we start with the rightmost bottom point of w coinciding with the origin of f (top, left) and ending rith the top left point of w coinciding with the bottom right pixel of f.

Problem 3.20

(a) By inspection,

$$w_1 = \begin{bmatrix} w_1(-1) \\ w_1(0) \\ w_1(1) \end{bmatrix} = \mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}. \text{ Similalry, } w_2 = \begin{bmatrix} w_2(-1) & w_2(0) & w_2(1) \end{bmatrix} = \mathbf{w}^T = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

NOTE: For the solution of parts (b) and (c) of this problem, assume that the starting configuration is with the center of the kernel coincident with the origin of the unpaddedimage, as described in Fig. 3.30.

Problem 3.22

(a) By definition, a separable kernel is formed as $\mathbf{v}\mathbf{w}^T$. So, yes, the kernel will be separable. Another way to answer this is that a matrix formed as the outer product of two vectors is always of rank 1. And a kernel with rank 1 is separable.

Problem 3.23

(a) As stated at the beginning of the subsection entitled Separable Filter Kernels in Section 3.4, a function $G(x,y)$ is separable if it can be written as $G(x,y) = G_1(x)G_2(y)$. The 2-D Gaussian function can be written as

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} = G_1(x)G_2(y)$$

Thus, the Gaussian kernel is separable.

Problem 3.24

Let the column vector be denoted by

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

and the row vector as

$$\mathbf{r} = \begin{bmatrix} r_1 & r_2 & \cdots & r_n \end{bmatrix}$$

Their outer product is

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \begin{bmatrix} r_1 & r_2 & \cdots & r_n \end{bmatrix} = \begin{bmatrix} c_1r_1 & c_1r_2 & \cdots & c_1r_n \\ c_2r_1 & c_2r_2 & \cdots & c_2r_n \\ \vdots & \vdots & \cdots & \vdots \\ c_mr_1 & c_mr_2 & \cdots & c_mr_n \end{bmatrix}$$

From Fig. 3.28, we know that the convolution of two, 2-D functions involves rotating one of the functions $180°$ and sliding it past the other so that each of the elements of one function "visits" every element of the other . [This is "full convolution", as explained in connection with Eqs. (3-36) and (3-37).] If we rotate $\mathbf{r}$ by $180°$ we get

$$\mathbf{r} = \begin{bmatrix} r_n & r_{n-1} & \cdots & r_1 \end{bmatrix}$$

You can see that sliding $\mathbf{r}$ past the first row of $\mathbf{c}$ will generate the convolution elements

$$r_1c_1 \quad r_2c_1 \quad \cdots \quad r_mc_1$$

which is equal to the first row of the product above. It's easy to see that when $\mathbf{r}$ slides past the second row of $\mathbf{c}$ that the second row of the product will be generated, and so on, so that at the end of the convolution process we will have generated the entire product of the two vectors.

Problem 3.25

(a) For two functions, Table 3.6 shows that the mean and variance of the product of two functions is

$$m_{g_1 \times g_2} = \frac{m_1\sigma_2^2 + m_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

and

$$\sigma^2_{g_1 \times g_2} = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

If a third function is included in the product, it follows that

$$m_{g_3 \times (g_1 \times g_2)} = \frac{m_3\sigma^2_{g_1 \times g_2} + m_{g_1 \times g_2}\sigma_3^2}{\sigma^2_{g_1 \times g_2} + \sigma_3^2}$$

and

$$\sigma^2_{g_3 \times (g_1 \times g_2)} = \frac{\sigma_3^2\sigma^2_{g_1 \times g_2}}{\sigma_3^2 + \sigma^2_{g_1 \times g_2}}$$

This pattern generalizes to

$$m_{g_K \times (g_{K-1} \times g_{K-2} \times \cdots \times g_1 \times g_2)} = \frac{m_{g_K}\sigma^2_{g_{K-1} \times g_{K-2} \times \cdots \times g_1 \times g_2} + m_{g_{K-1} \times g_{K-2} \times \cdots \times g_1 \times g_2}\sigma_K^2}{\sigma^2_{g_{K-1} \times g_{K-2} \times \cdots \times g_1 \times g_2} + \sigma_K^2}$$

and

$$\sigma^2_{g_K \times (g_{K-1} \times g_{K-2} \times \cdots \times g_1 \times g_2)} = \frac{\sigma_K^2\sigma^2_{g_{K-1} \times g_{K-2} \times \cdots \times g_1 \times g_2}}{\sigma_K^2 + \sigma^2_{g_{K-1} \times g_{K-2} \times \cdots \times g_1 \times g_2}}$$

The standard deviation is the square root of the variance.

Problem 3.26

(a) The number of boundary points between the black and white regions is much larger in the image on the right. When the images are blurred, the boundary points will give rise to a larger number of different values for the image on the right, so the histograms of the two blurred images will be different.

Problem 3.27

(a) From Eqs. (3-39) and (3-40), the composite kernel is of size

$$[Q(m-1)+m]\times[Q(m-1)+m]=[4(3-1)+3]\times[4(3-1)+3]$$
$$=11\times11$$

As you know from our discussion of the Gaussian lowpass kernel in Section 3.5, a kernel whose size is only $3\sigma\times3\sigma$ will have a behavior between a box kernel and a Gaussian kernel. We need on the order of $6\sigma\times6\sigma$ to obtain the benefits of a kernel with smooth skirts.

Problem 3.28

(a) Yes. As Table 3.6 indicates, the convolution of Gaussians is a Gaussian.

Also, as you know from our discussion of the Gaussian lowpass kernel in Section 3.5, a kernel whose size is not on the order of $6\sigma\times6\sigma$ will not have the benefits of a the long skirts of a Gaussian kernel. The kernels in this problem will have a behavior between a box kernel and a Gaussian kernel.

Problem 3.29

One of the easiest ways to look at repeated applications of a spatial filter is to use superposition. Let $f(x,y)$ and w denote the image and a kernel, respectively. Assuming square images of size $N\times N$ for convenience, we can express $f(x,y)$ as the sum of at most N^2 images, each of which has only one nonzero pixel (initially, we assume that N can be infinite). Then, the process of filtering with kernel w is given by the convolution

$$w\star f=w\star\left[f_1+f_2+\cdots+f_{N^2}\right]$$

Suppose for illustrative purposes that f_i has value 1 at its center, while the other pixels in that image are valued 0 (see Fig. P3.29(a)). If w is a 3×3 kernel of 1/9's (Fig. P3.29(b)), then convolving w with f_i will produce an image with a 3×3 array of 1/9's at its center and 0s elsewhere, as Fig. P3.29(c) shows. If w is now applied to this image, the resulting image will be as shown in Fig. P3.39(d). Note that the sum of the nonzero pixels in both Figs. P3.29(c) and (d) is the same, and equal to the value of the original pixel. Thus, it is intuitively evident that successive applications of w will "diffuse" the nonzero value of f_i (not an unexpected result, because w is a blurring kernel). Since the sum remains constant, the values of the nonzero elements will become smaller and smaller, as the number of applications of the filter increases. The overall result is given by adding all the convolved f_k, for $k=1,2,\ldots,N^2$.

Every iteration of blurring further diffuses the values outwardly from the starting point. In the limit, the values would get infinitely small, but, because the average value remains constant, this would require an image of infinite spatial proportions. It is at this junction that border conditions become important. Although it is not required in the problem statement, it is instructive to discuss in class the effect of successive applications of w to an image of finite proportions. The net effect is that, because the values cannot diffuse outward past the boundary of the image, the denominator in the successive applications of averaging eventually overpowers the pixel values, driving the image to zero in the limit. A simple example of this is given in Fig. P3.29(e), which shows an array of size 1×7 that is blurred by successive applications

of the 1×3 kernel $w=\begin{bmatrix}1 & 1 & 1\end{bmatrix}/3$. We see that, provided that the values of the blurred 1 can diffuse out, the sum, S, of the resulting pixels is 1. However, when the boundary is met, an assumption must be made regarding how kernel operations on the border are treated. Here, we used the common assumption that pixel value immediately past the boundary are 0. The kernel operation does not go beyond the boundary, however. In this example, we see that the sum of the pixel values begins to decrease with successive applications of the kernel. In the limit, the term $1/(3)^n$ would overpower the sum of the pixel values, yielding an array of 0's.

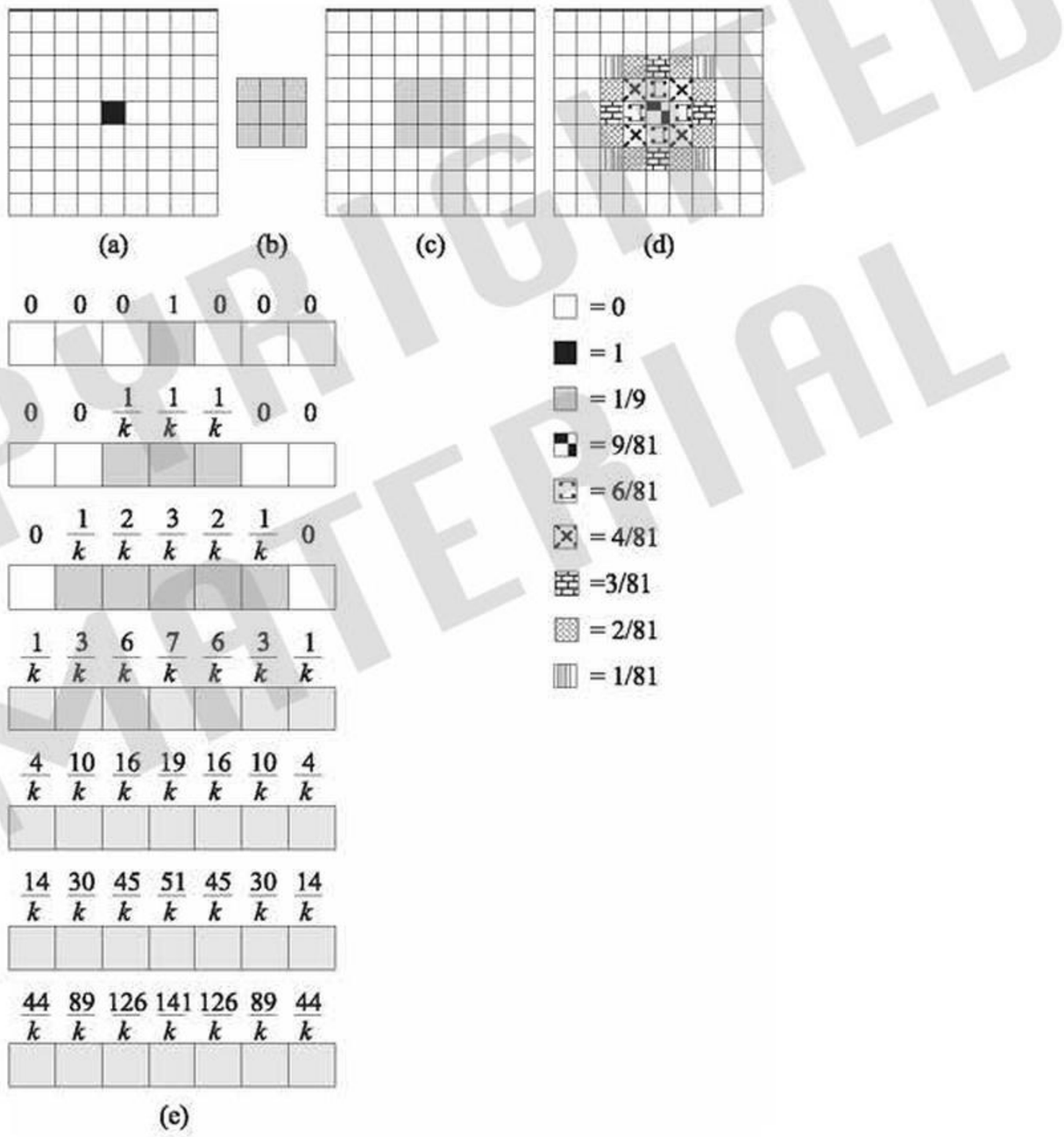


Figure P3.29

Problem 3.31

We assume full convolution, as explained in connection with Eqs. (3-36) and (3-37). Let the coefficients of a general kernel be denoted by $c_1, c_2, c_3, \ldots c_K$. We know that in full convolution each coefficient in the kernel multiplies each pixel in the input image one time. The only operation performed is a sum of products. Therefore, if we were to expand the convolution summation, add all the terms, and collect the coefficients multiplying every pixel, $f(i, j)$, of the original image, we would find that they appear as follows:

$$f(i, j)[c_1 + c_2 + c_2 + \cdots + c_K]$$

But the sum of the coefficients is assumed to be 1. So, if we carry out this opearation for all locations (i,j) in the image and sum the results, we conclude that the sum of the pixels in a filtered image will be equal the sum of the pixels in the original image

Problem 3.36

(a) There are n^2 points in an $n \times n$ median filter. Because n is odd, the median value, ζ, is such that there are $(n^2-1)/2$ points with values less than or equal to ζ and the same number with values greater than or equal to ζ. However, because the area A (number of points) in the cluster is less than one half n^2, and A and n are integers, it follows that A is always less than or equal to $(n^2-1)/2$. Thus, even in the extreme case when all cluster points are encompassed by the filter, there are not enough points in the cluster for any of them to be equal to the value of the median (remember, we are assuming that all cluster points are lighter or darker than the background points). Therefore, if the center point in the filter is a cluster point, it will be set to the median value, which is a background shade, and thus it will be eliminated from the cluster. This conclusion obviously applies also to the less extreme case when the number of cluster points encompassed by the median filter is less than the maximum size of the cluster.

Problem 3.37

(a) Numerically sort the n^2 values. The median is

$$\zeta = \left[(n^2+1)/2\right]\text{-th largest value.}$$

Problem 3.39

The Laplacian operator is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

for the unrotated coordinates, and

$$\nabla'^2 f = \frac{\partial^2 f}{\partial x'^2} + \frac{\partial^2 f}{\partial y'^2}$$

for the rotated coordinates. It is given that

$$x' = x\cos\theta - y\sin\theta \quad \text{and} \quad y' = x\sin\theta + y\cos\theta$$

where θ is the angle of rotation. We want to show that the right sides of the first two equations are equal.

We start with

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial x'}\frac{\partial x'}{\partial x} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial y}$$
$$= \frac{\partial f}{\partial x'}\cos\theta + \frac{\partial f}{\partial y'}\sin\theta$$

Taking the partial derivative of this expression again with respect to x yields

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial x'^2}\cos^2\theta + \frac{\partial}{\partial x'}\left(\frac{\partial f}{\partial y'}\right)\sin\theta\cos\theta + \frac{\partial}{\partial y'}\left(\frac{\partial f}{\partial x'}\right)\cos\theta\sin\theta + \frac{\partial^2 f}{\partial y'^2}\sin^2\theta$$

Next, we compute

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial x'}\frac{\partial x'}{\partial y} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial y}$$
$$= -\frac{\partial f}{\partial x'}\sin\theta + \frac{\partial f}{\partial y'}\cos\theta$$

Taking the derivative of this expression again with respect to y gives

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial x'^2}\sin^2\theta - \frac{\partial}{\partial x'}\left(\frac{\partial f}{\partial y'}\right)\cos\theta\sin\theta - \frac{\partial}{\partial y'}\left(\frac{\partial f}{\partial x'}\right)\sin\theta\cos\theta + \frac{\partial^2 f}{\partial y'^2}\cos^2\theta$$

Finally, adding the two expressions for the second derivatives gives the result:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial x'^2}\left[\cos^2\theta + \sin^2\theta\right] + \frac{\partial^2 f}{\partial y'^2}\left[\sin^2\theta + \cos^2\theta\right]$$
$$= \frac{\partial^2 f}{\partial x'^2} + \frac{\partial^2 f}{\partial y'^2}$$

which proves that the Laplacian operator is independent of rotation.

Problem 3.40

The Laplacian kernel with a -4 in the center performs an operation proportional to differentiation in the horizontal and vertical directions. Consider for a moment a 3×3 "Laplacian" kernel with a -2 in the center and 1's above and below the center. All other elements are 0. This kernel will perform differentiation in only one direction, and will ignore intensity transitions in the orthogonal direction. An image processed with such a kernel will exhibit sharpening in only one direction. A Laplacian kernel with a -4 in the center and 1's in the vertical and horizontal directions will obviously produce an image with sharpening in both directions and in general will appear sharper than with the previous kernel. Similarly, a kernel with a -8 in the center and 1's in the horizontal, vertical, and diagonal directions will detect the same intensity changes as the kernel with the -4 in the center but, in addition, it will also be able to detect changes along the diagonals, thus generally producing sharper-looking results.

Problem 3.41

With reference to Eqs. (3-55) and (3-56), and using $k = 1$ we can write the following equation for unsharp masking:

$$g(x, y) = f(x, y) + f(x, y) - \overline{f}(x, y)$$

Convolving $f(x, y)$ with the kernel in Fig. P3.58(a) produces $f(x, y)$. Convolving $f(x, y)$ with the kernel in Fig. 3.31(a) produces $\overline{f}(x, y)$. Then, because these operations are linear, we can use superposition, and we see from the preceding equation that using two kernels of the form in Fig. P3.41(a) and the kernel in Fig. 3.31(a) produces the composite kernel in Fig. P3.41(b). Convolving this kernel with $f(x, y)$ produces $g(x, y)$, the unsharp result.

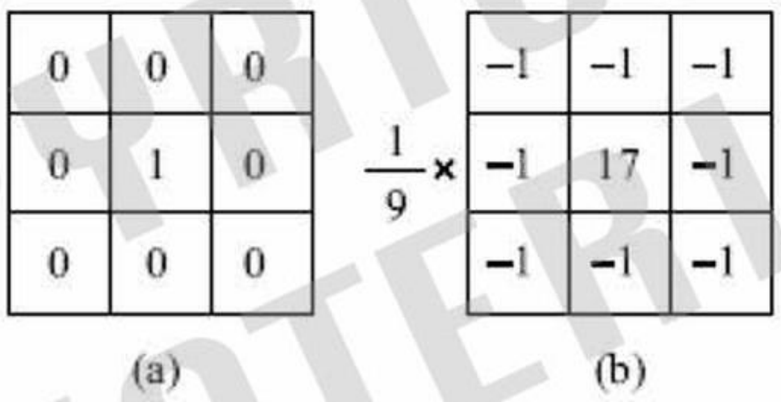


Figure P3.41

Problem 3.43

(a) The magnitude of the gradient is

$$\text{mag}(\nabla f) = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

for the unrotated coordinates and

$$\text{mag}(\nabla' f) = \left[\left(\frac{\partial f}{\partial x'}\right)^2 + \left(\frac{\partial f}{\partial y'}\right)^2\right]^{1/2}$$

for the rotated coordinates. It is given in the statement of Problem 3.39 that

$$x' = x\cos\theta - y\sin\theta \quad \text{and} \quad y' = x\sin\theta + y\cos\theta$$

where θ is the angle of rotation. We want to show that the right sides of the first two equations are equal.

We start by writing

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial x'}\frac{\partial x'}{\partial x} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial y}$$
$$= \frac{\partial f}{\partial x'}\cos\theta + \frac{\partial f}{\partial y'}\sin\theta$$

and

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial x'}\frac{\partial x'}{\partial y} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial y}$$
$$= -\frac{\partial f}{\partial x'}\sin\theta + \frac{\partial f}{\partial y'}\cos\theta$$

from which it follows that

$$\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 = \left(\frac{\partial f}{\partial x'}\right)^2 + \left(\frac{\partial f}{\partial y'}\right)^2$$

or

$$\left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} = \left[\left(\frac{\partial f}{\partial x'}\right)^2 + \left(\frac{\partial f}{\partial y'}\right)^2\right]^{1/2}$$

thus proving that the magnitude of the gradient is an isotropic operator.

Problem 3.44

(c) The rank of both Sobel kernels is 1, so they are separable. For the kernel in Fig. 3.50(d), the vectors $\mathbf{v}$ and $\mathbf{w}$ such that the kernel is equal to $\mathbf{v}\mathbf{w}^T$ are: $\mathbf{v} = [1 \quad 0 \quad -1]^T$ and $\mathbf{w} = [-1 \quad -2 \quad -1]^T$. For the kernel in Fig. 3.50(e) the vectors are $\mathbf{v} = [-1 \quad -2 \quad -1]^T$ and $\mathbf{w} = [1 \quad 0 \quad -1]^T$.

Problem 3.45

(a) The most extreme case is when the kernel is positioned on the center pixel of a 3-pixel gap, along a thin segment, in which case a 3×3 kernel would encompass a completely blank field. Since this is known to be the largest gap, the next (odd) kernel size up is guaranteed to encompass some of the pixels in the segment. Thus, the smallest kernel that will do the job is a 5×5 averaging kernel.

Chapter 4

Problem Solutions

Problem 4.1

(a)

$$\delta(t-t_0)=\begin{cases}\infty & \text{if } t=t_0\\ 0 & \text{if } t\neq t_0\end{cases}$$

(c) Yes. The first expression, $\delta(t-a)$, says that the impulse is located at $t=a$. The second, $\delta(a-t)$, says that the impulse is located at $a=t$, which of course is the same thing.

Problem 4.3

(a) From the definition of 1-D continuous convolution, Eq. (4-24),

$$\begin{aligned}\delta(t)\star\delta(t-t_0)&=\int_{-\infty}^{\infty}\delta(\tau)\delta(t-t_0-\tau)d\tau\\&=\delta(t-t_0)\end{aligned}$$

where the second line follows directly from the sifting property of the impulse, Eq. (4-13). Because the arguments of the functions on the left are different, it is clearer to show the functions and their arguments explicitly, rather than combining them in the form of Eq. (4-24). Using that form of the equation we could write the left side of the above equation as $(\delta\star\delta)(t,t_0)$ which is not as clear. We will use the above, more explicit notation whenever necessary to enhance clarity.

Problem 4.4

Convolution of two, 1-D continuous functions is defined by Eq. (4-24):

$$f(t)\star h(t)=\int_{-\infty}^{\infty}f(\tau)h(t-\tau)d\tau$$

As mentioned in Problem 4.3, we are using the equivalent notation $f(t)\star h(t)$ rather than $(f\star h)(t)$ in some solutions to enhance clarity.

Consider first an impulse located at the origin: The sifting property of a 1-D impulse at the origin is defined by Eq. (4-11):

$$\int_{-\infty}^{\infty} f(t)\delta(t) = f(0)$$

If we let $h(t) = \delta(t)$ in the first expression we obtain

$$f(t) \star \delta(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau$$

Note that the independent variable in the definition of the sifting property of the impulse is different than for convolution. However, we can use the commutative property of convolution to write this equation as:

$$f(t) \star \delta(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau = \int_{-\infty}^{\infty} f(t-\tau)\delta(\tau)d\tau$$

We can now apply the sifting property, which gives the value of the function at $\tau = 0$, or

$$f(t) \star \delta(t) = f(t)$$

In other words, a function convolved with an impulse at the origin does not change the function.

An impulse located at t_0 is written as $\delta(t - t_0)$. Following the same line of reasoning as above, we write:

$$f(t) \star \delta(t - t_0) = \int_{-\infty}^{\infty} f(\tau)\delta([t - t_0] - \tau)d\tau = \int_{-\infty}^{\infty} f(t - t_0 - \tau)\delta(\tau)d\tau = f(t - t_0)$$

which agrees with Eq. (4-13). In other words, the function is now copied so that, a value of the function at any point t, now appears at location $(t - t_0)$. For instance, the origin of $f(t)$ is now at t_0 instead of at 0.

Problem 4.5

Figure P4.5 shows a solution. The aliased function consists of the samples shown.

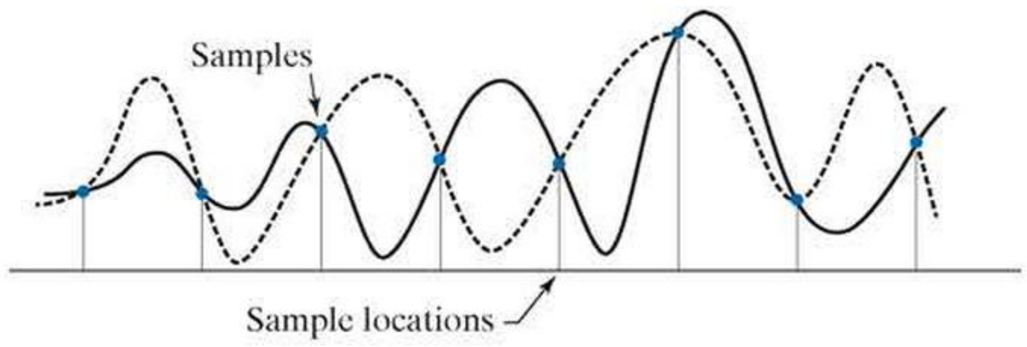


Figure P4.5

Problem 4.6

(a) The sampling rate has to be greater than 1 sample/sec. Figure P4.6 shows a sampling rate slightly higher than 1 sample/sec.

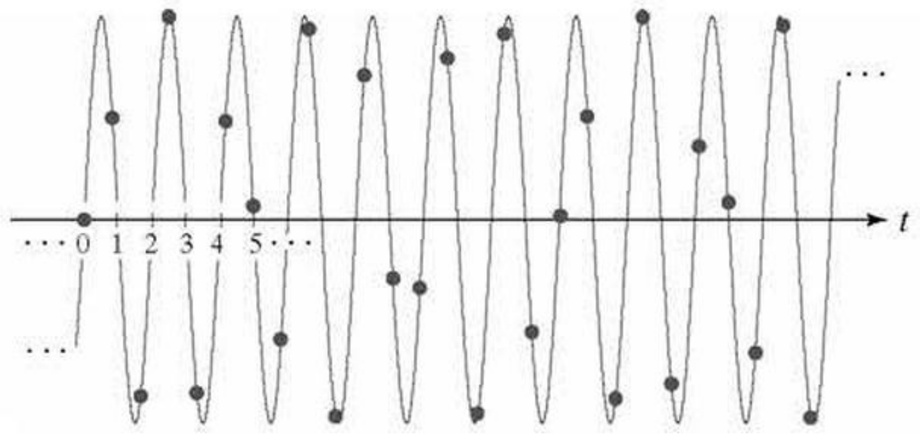


Figure P4.6

Problem 4.7

(b) The Nyquist rate is twice the highest frequency or $(2)(1/2) = 1$ sample/sec.

Problem 4.8

We know from Example 4.2 that

$$\Im\{\delta(t - t_0)\} = e^{-j2\pi t_0 \mu}$$

We know this to be true, so it must follow that the inverse Fourier transform of the exponential must be the impulse. That is,

$$\Im^{-1}\left\{e^{-j2\pi t_0 t}\right\} = \int_{-\infty}^{\infty} e^{-j2\pi t_0 t} e^{j2\pi\mu t} d\mu$$
$$= \int_{-\infty}^{\infty} e^{j2\pi\mu(t-t_0)} d\mu = \delta(t - t_0)$$

By definition, the Fourier transform of $f(t) = e^{j2\pi kt}$ is

$$F(\mu) = \Im\left\{e^{j2\pi t_0 t}\right\} = \int_{-\infty}^{\infty} e^{j2\pi t_0 t} e^{-j2\pi\mu t} dt$$
$$= \int_{-\infty}^{\infty} e^{j2\pi t(t_0-\mu)} dt$$
$$= \delta(t_0 - \mu)$$
$$= \delta(\mu - t_0)$$

where we used the form of the second line in the preceding equation. The last line follows from Problem 4.1(c).

Problem 4.9

(a) It follows from Euler's formula that

$$\cos(2\pi\mu_0 t) = \frac{e^{j2\pi\mu_0 t} + e^{-j2\pi\mu_0 t}}{2}$$

Substituting this expression directly into the definition of the Fourier transform gives us:

$$\begin{aligned}
\Im\{\cos(2\pi\mu_0 t)\} &= \int_{-\infty}^{\infty} \cos(2\pi\mu_0 t) e^{-j2\pi\mu t} dt \\
&= \int_{-\infty}^{\infty} \left[\frac{e^{j2\pi\mu_0 t} + e^{-j2\pi\mu_0 t}}{2}\right] e^{-j2\pi\mu t} dt \\
&= \frac{1}{2}\int_{-\infty}^{\infty} \left[e^{j2\pi\mu_0 t} + e^{-j2\pi\mu_0 t}\right] e^{-j2\pi\mu t} dt \\
&= \frac{1}{2}\left[\int_{-\infty}^{\infty} e^{j2\pi\mu_0 t} e^{-j2\pi\mu t} dt + \int_{-\infty}^{\infty} e^{-j2\pi\mu_0 t} e^{-j2\pi\mu t} dt\right] \\
&= \frac{1}{2}\left[\int_{-\infty}^{\infty} e^{j2\pi(\mu-\mu_0)t} dt + \int_{-\infty}^{\infty} e^{j2\pi(\mu+\mu_0)t} dt\right] \\
&= \frac{1}{2}[\delta(\mu-\mu_0) + \delta(\mu+\mu_0)]
\end{aligned}$$

where the last step follows from Problem 4.8.

Problem 4.10

(a) The period is such that $2\pi nt = 2\pi$, or $t = 1/n$.

(b) The frequency is 1 divided by the period, or n.

(c) The continuous Fourier transform of the given sine wave looks as in Fig. P4.10(a) (see Problem 4.9), and the transform of the sampled data (showing a few periods) has the general form illustrated in Fig. P4.10(b) (the dashed box is an ideal filter that would allow reconstruction if the sine function were sampled, with the sampling theorem being satisfied). The sampled function would look as in Problem 4.6(a) if the sampling theorem were satisfied.

Problem 4.11

Starting from Eq. (4-24),

$$f(t) \star g(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau$$

(Refer to the solution of Problem 4.4(a) regarding the notation we are using to denote convolution). The Fourier transform of this expression is

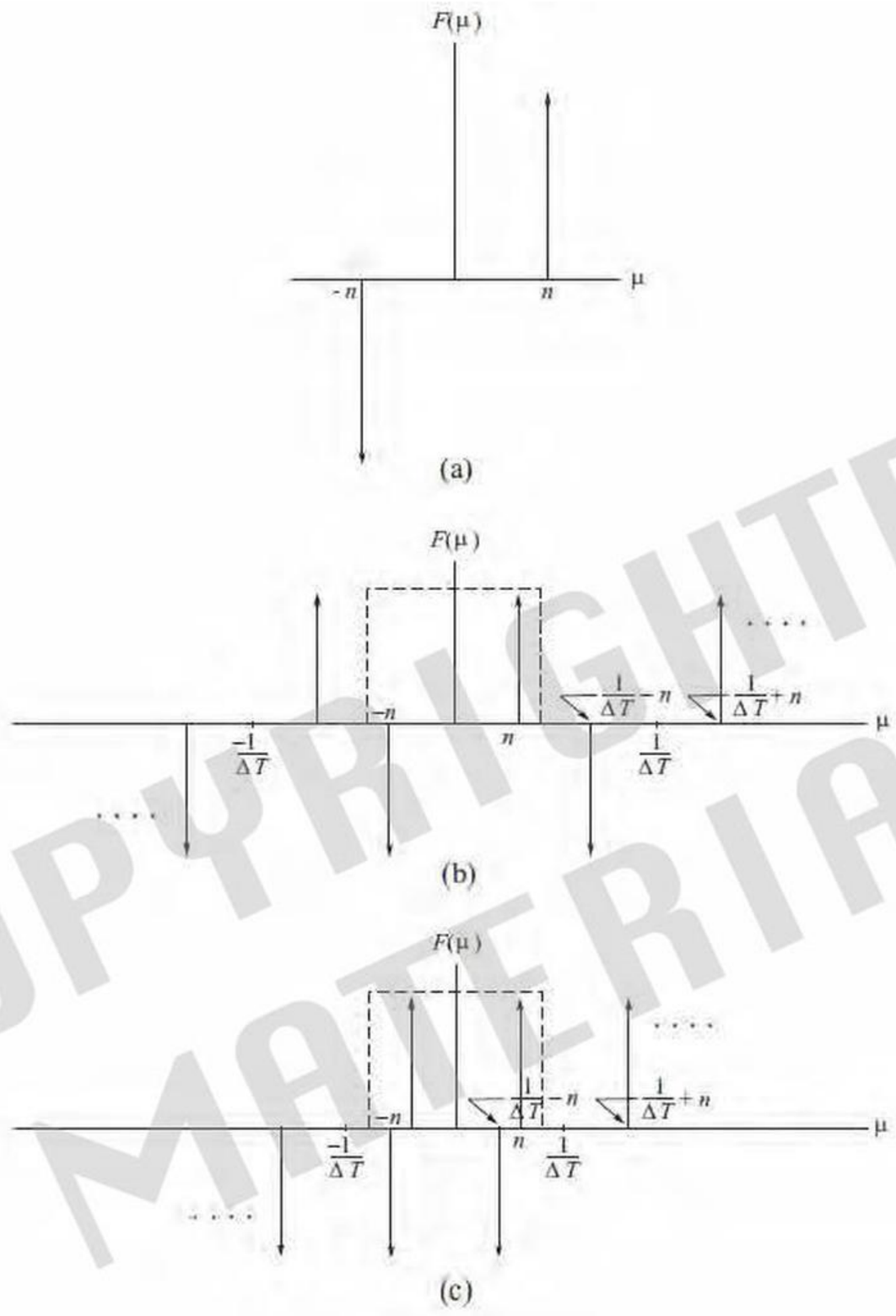


Figure P4.10

$$\Im[f(t) \star g(t)] = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau\right]e^{-j2\pi\mu t}dt$$

$$= \int_{-\infty}^{\infty} f(\tau)\left[\int_{-\infty}^{\infty} g(t-\tau)e^{-j2\pi\mu t}dt\right]d\tau$$

The term inside the inner bracket is the Fourier transform of $g(t-\tau)$. But, we know from the translation property (Table 4.4) that

$$\Im[g(t-\tau)] = G(\mu)e^{-j2\pi\mu\tau}$$

so,

$$\begin{aligned}\Im[f(t)\star g(t)] &= \int_{-\infty}^{\infty} f(\tau)\left[G(\mu)e^{-j2\pi\mu\tau}\right]d\tau \\ &= G(\mu)\int_{-\infty}^{\infty} f(\tau)e^{-j2\pi\mu\tau}d\tau \\ &= G(\mu)F(\mu)\end{aligned}$$

This proves that multiplication in the frequency domain is equal to convolution in the spatial domain. The proof that multiplication in the spatial domain is equal to convolution in the spatial domain is done in a similar way.

Problem 4.13

$$\begin{aligned}f(t) &= h(t)\star\tilde{f}(t) \\ &= \int_{-\infty}^{\infty} h(z)\tilde{f}(t-z)dz \\ &= \int_{-\infty}^{\infty}\frac{\sin(\pi z/\Delta T)}{(\pi z/\Delta T)}\sum_{n=-\infty}^{\infty} f(t-z)\delta(t-n\Delta T-z)dz \\ &= \sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{\sin(\pi z/\Delta T)}{(\pi z/\Delta T)}f(t-z)\delta(t-n\Delta T-z)dz \\ &= \sum_{n=-\infty}^{\infty} f(n\Delta T)\frac{\sin\left[\pi(t-n\Delta T)/\Delta T\right]}{\left[\pi(t-n\Delta T)/\Delta T\right]} \\ &= \sum_{n=-\infty}^{\infty} f(n\Delta T)\operatorname{sinc}\left[(t-n\Delta T)/\Delta T\right]\end{aligned}$$

Problem 4.15

(b) We solve this problem by direct substitution and using orthogonality. First we have to show that $F(u)$ the DFT of $f(x)$. Substituting Eq. (4-45) into (4-44) yields

$$\begin{aligned}F(u) &= \sum_{x=0}^{M-1}\left[\frac{1}{M}\sum_{r=0}^{M-1}F(u)e^{j2\pi rx/M}\right]e^{-j2\pi ux/M} \\ &= \frac{1}{M}\sum_{r=0}^{M-1}F(r)\left[\sum_{x=0}^{M-1}e^{j2\pi rx/M}e^{-j2\pi ux/M}\right] \\ &= \frac{1}{M}F(u)M \\ &= F(u)\end{aligned}$$

where, because of orthogonality, the third step is 0 unless $r = u$.

Next, we have to show that $f(x)$ is the inverse DFT of $F(u)$. Substituting Eq. (4-44) into (4-45) we get

$$
\begin{aligned}
f(x) &= \frac{1}{M}\sum_{u=0}^{M-1}\left[\sum_{r=0}^{M-1} f(r)e^{-j2\pi ur/M}\right]e^{j2\pi ux/M} \\
&= \frac{1}{M}\sum_{r=0}^{M-1} f(r)\left[\sum_{u=0}^{M-1} e^{-j2\pi ur/M}e^{j2\pi ux/M}\right] \\
&= \frac{1}{M} f(x)M \\
&= f(x)
\end{aligned}
$$

where, because of orthogonality, the third step is 0 unless $r = x$. By showing that $F(u)$ is the DFT of $f(x)$ and that $f(x)$ is the IDFT of $F(u)$, we have established that Eqs. (4-44) and (4-45) constitute a Fourier transform pair.

Problem 4.17

(a) We show that the DFT of the left side of the double arrow equals the right side by substituting directly into the forward DFT, Eq. (4-44):

$$
\begin{aligned}
\Im[f(x)e^{j2\pi u_0 x/M}] &= \sum_{x=0}^{M-1}\left[f(x)e^{j2\pi u_0 x/M}\right]e^{-j2\pi ux/M} \\
&= \sum_{x=0}^{M-1} f(x)e^{-j2\pi[(u-u_0)x/M]} \\
&= F(u-u_0)
\end{aligned}
$$

Because we did this by direct substitution into Eq. (4-44), and this equation and Eq. (4-45) are a Fourier transform pair, it must follow that the left side of the double arrow is the IDFT of the right.

Problem 4.18

(a) We have to show first that the Fourier transform of the convolution $f(x)\star h(x)$ is the product $F(u)H(u)$, and vice versa. Using the definition of the 1-D convolution theorem in Eq. (4-24), the DFT in Eq. (4-44), and 1-D discrete convolution in Eq. (4-48), we write

$$
\begin{aligned}
\Im[(f\star h)(t)] = \Im\left[f(x)\star h(x)\right] &= \sum_{x=0}^{M-1}\left[\sum_{m=0}^{M-1} f(m)h(x-m)\right]e^{-j2\pi ux/M} \\
&= \sum_{m=0}^{M-1} f(m)\left[\sum_{x=0}^{M-1} h(x-m)e^{-j2\pi ux/M}\right] \\
&= \sum_{m=0}^{M-1} f(m)H(u)e^{-j2\pi um/M} \\
&= H(u)\sum_{m=0}^{M-1} f(m)e^{-j2\pi um/M} \\
&= H(u)F(u) \\
&= (H\bullet F)(u)
\end{aligned}
$$

where the third step follows from the translation property of the Fourier transform (see Problem 4.17). Because we showed the preceding properties by substituting into the DFT, and the DFT and IDFT are a transform pair, it must be true that the reverse, i.e., that $f(x) \star h(x)$ is the IDFT of $F(u)H(u)$, is true also (see the discussion on this concept following Eqs. (4-44) and (4-45) in the book).

Problem 4.19

$$(f \star h)(t,z) = f(t,z) \star h(t,z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha,\beta) h(t-\alpha)(z-\beta) d\alpha d\beta$$

Problem 4.21

(a) Because all rows of the image are identical, we can focus attention on one row, which is a 1-D square wave with a period, P, of four pixels. Therefore, the frequency of this signal is $f = 1/4 = 0.25$ cycles/pixel. If the stripes are now four pixels wide, then the period is eight pixels, and the frequency of the signal is $f = 1/8 = 0.125$ cycles/pixel, which is one-half the frequency of the original signal. The center peak in the spectrum shown in the problem statement is the dc term, and the other two dominant peaks appear on the horizontal axis of the spectrum, exactly half-way between the center and ends of the horizontal axis of the spectrum. The corresponding peaks in the new spectrum have half the frequency, so they will appear midway between the original peaks and the center of the spectrum. That is, one-quarter of the axis length on either side of center. (The spectrum contains other harmonic frequency components that are of lower amplitude and are not shown.)

Problem 4.23

A simple example of an image that would not be aliased by pixel replication is a black rectangle on a white background, provided that the rectangle is perfectly aligned with the image borders. Zooming would result in larger rectangles, but the edges would be perfectly preserved and the enlarged object would be a perfect rectangle with no edge degradation (such as jaggies).

Problem 4.24

(a) From Eq. (4-59),

$$F(\mu,\nu) = \Im[f(t,z)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t,z) e^{-j2\pi(\mu t + \nu z)} dt dz$$

From Eq. (2-23), we know that the Fourier transform is a linear operator if

$$\Im[a_1 f_1(t,z) + a_2 f_2(t,z)] = a_1 \Im[f_1(t,z)] + a_2 \Im[f_2(t,z)]$$

Substituting into the definition of the Fourier transform yields

$$
\begin{aligned}
\Im[a_1 f_1(t,z) + a_2 f_2(t,z)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[a_1 f_1(t,z) + a_2 f_2(t,z)] \\
&\qquad \times e^{-j2\pi(\mu t + vz)} dtdz \\
&= a_1 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z) e^{-j2\pi(\mu t + vz)} dtdz \\
&\qquad + a_2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_2(t,z) e^{-j2\pi(\mu t + vz)} dtdz \\
&= a_1 \Im[f_1(t,z)] + a_2 \Im[f_2(t,z)]
\end{aligned}
$$

where the second step follows from the distributive property of the integral. The linearity of the inverse transform is proved in exactly the same way.

Problem 4.25

(a) We solve the problem by direct substitution into the forward Fourier transform, Eq. (4-59):

$$
\begin{aligned}
\Im\left[f(x,y)e^{j2\pi(\mu_0 t + v_0 z)}\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{j2\pi(\mu_0 t + v_0 z)} e^{-j2\pi(\mu t + vz)} dtdz \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-j2\pi([\mu-\mu_0]t + [v-v_0]z)} dtdz \\
&= F(\mu - \mu_0, v - v_0)
\end{aligned}
$$

Because we used direct substitution into Eq. (4-59), and this equation and Eq. (4-60) are a Fourier transform pair, it must follow that the left side of the double arrow is the inverse transform of the right, that is, $\Im^{-1}[F(\mu - \mu_0, v - v_0)] = f(x,y)e^{j2\pi(\mu_0 t + v_0 z)}$.

Problem 4.26

(a) By direct substitution into Eq. (4-59) we obtain

$$
\begin{aligned}
\Im[\delta(t,z)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(t,z) e^{-j2\pi(\mu t + vz)} dtdz \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi(\mu t + vz)} \delta(t,z) dtdz \\
&= e^{-j2\pi(\mu 0 + v 0)} \\
&= 1
\end{aligned}
$$

where the third line follows from the sifting property of the impulse, Eq. (4-54). Because we used direct substitution into Eq. (4-59), and this equation and Eq. (4-60) are a Fourier transform pair, it must follow that the left side of the double arrow is the inverse transform of the right, that is, $\Im^{-1}[1] = \delta(t,z)$.

(b) We solve this problem starting with the inverse Fourier transform

$$\begin{aligned}\Im^{-1}\left[\delta(\mu,\nu)\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(\mu,\nu)e^{j2\pi(\mu t + \nu z)}d\mu d\nu \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{j2\pi(\mu t + \nu z)}\delta(\mu,\nu)d\mu d\nu \\ &= e^{j2\pi(0t + 0z)} \\ &= 1\end{aligned}$$

where the third line follows from the sifting property of the impulse, Eq. (4-54). Because we used direct substitution into Eq. (4-60), and this equation and Eq. (4-59) are a Fourier transform pair, it must follow that the right side of the double arrow is the forward transform of the left, that is, $\Im\left[1\right] = \delta(t,z)$.

(c) We solve the problem by direct substitution into Eq. (4-59):

$$\begin{aligned}\Im\left[\delta(t-t_0,z-z_0)\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(t-t_0,z-z_0)e^{-j2\pi(\mu t + \nu z)}dtdz \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi(\mu t + \nu z)}\delta(t-t_0,z-z_0)dtdz \\ &= e^{-j2\pi(\mu t_0 + \nu z_0)}\end{aligned}$$

where the third step follows from the sifting property of the 2-D impulse, Eq. (4-55). Because we used direct substitution into Eq. (4-59), and this equation and Eq. (4-60) are a Fourier transform pair, it must follow that the left side of the double arrow is the inverse transform of the right, that is, $\Im^{-1}\left[e^{-j2\pi(\mu t_0 + \nu z_0)}\right] = \delta(t-t_0,z-z_0)$.

We could have solved the problem directly using the results from Problem 4.25(b) by letting $f = \delta$, and using the knowledge from (a) above that, when $f = \delta$, the Fourier transform is $F = 1$.

(e) By direct substitution into Eq. (4-59), and using Euler's formula to expand the cosine we obtain

$$\begin{aligned}\Im\left[\cos(2\pi\mu_0 t + 2\pi\nu_0 z)\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\frac{e^{j2\pi(\mu_0 t + \nu_0 z)} + e^{-j2\pi(\mu_0 t + \nu_0 z)}}{2}\right]e^{-j2\pi(\mu t + \nu z)}dtdz \\ &= \frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{j2\pi(\mu_0 t + \nu_0 z)}e^{-j2\pi(\mu t + \nu z)}dtdz + \frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{j2\pi(-\mu_0 t - \nu_0 z)}e^{-j2\pi(\mu t + \nu z)}dtdz\end{aligned}$$

We recognize these two integrals as the Fourier transforms of exponentials, which we solved in part(d). Using those results we obtain

$$\Im\left[\cos(2\pi\mu_0 t + 2\pi\nu_0 z)\right] = \frac{1}{2}\left[\delta(\mu-\mu_0,\nu-\nu_0) + \delta(\mu+\mu_0,\nu+\nu_0)\right]$$

Because we used direct substitution into Eq. (4-59), and this equation and Eq. (4-60) are a Fourier transform pair, it must follow that the left side of the double arrow is the inverse transform of the right, that is, $\Im^{-1}(1/2[\delta(\mu-\mu_0,\nu-\nu_0)+\delta(\mu+\mu_0,\nu+\nu_0)])=\cos(2\pi\mu_0 t+2\pi\nu_0 z)$.

Problem 4.27

(b) It is easier to show that the left side is the inverse DFT of the right. By substituting into Eq. (4-68) we obtain,

$$\begin{aligned}\Im^{-1}\left[F(u,v)e^{-j2\pi(ux_0+vy_0)}\right] &= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left[F(u,v)e^{-j2\pi(ux_0+vy_0}\right]e^{j2\pi(ux/M+vy/N)}\\ &= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{j2\pi[(u(x-x_0)/M+v(y-y_0)/N)]}\\ &= f(x-x_0,y-y_0)\end{aligned}$$

Thus, we have shown that the left side of the double arrow is the inverse DFT of the expression on the right. Because we did this by direct substitution into Eq. (4-68), and this equation and Eq. (4-67) are a Fourier transform pair, it must follow that the converse is true; that is that $F(u,v)e^{-j2\pi(x_0u/M+y_0v/N)}$ is the forward Fourier transform of $f(x-x_0,y-y_0)$.

Problem 4.28

(a) We solve the problem by direct substitution into Eq. (4-67):

$$\begin{aligned}\Im\left[\delta(x,y)\right] &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\delta(x,y)e^{-j2\pi(ux/M+vy/N)}\\ &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}e^{-j2\pi(ux/M+vy/N)}\delta(x,y)\\ &= e^{-j2\pi(u[0]/M+v[0]/N)}\\ &= 1\end{aligned}$$

where the third step follows from the sifting property of the 2-D impulse, Eq. (4-58). Because we used direct substitution into Eq. (4-67), and this equation and Eq. (4-68) are a Fourier transform pair, it must follow that the left side of the double arrow is the IDFT of the right: $\Im^{-1}\left[1\right]=\delta(x,y)$.

(b) We solve this problem by starting with the inverse DFT, Eq. (4-68):

$$\begin{aligned}\Im^{-1}\left[MN\delta(u,v)\right] &= \frac{MN}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\delta(u,v)e^{j2\pi(ux/M+vy/N)}\\ &= e^{j2\pi(0x/M+0y/N)}\\ &= 1\end{aligned}$$

where the second step follows from the sifting property of the 2-D impulse, Eq. (4-58). Because we used direct substitution into Eq. (4-68), and this equation and Eq. (4-67) are a Fourier transform pair, it must follow that the right side of the double arrow is the DFT of the left: $\Im[1] = MN\delta(x, y)$.

(d) We solve the problem by direct substitution into the inverse DFT, Eq. (4-68):

$$\begin{aligned}\Im^{-1}[MN\delta(u-u_0, v-v_0)] &= \frac{MN}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\delta(u-u_0, v-v_0)e^{j2\pi(ux/M+vy/N)} \\ &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}e^{j2\pi(ux/M+vy/N)}\delta(u-u_0, v-v_0) \\ &= e^{j2\pi(u_0x/M+v_0y/N)}\end{aligned}$$

where the last step follows from the sifting property of the 2-D discrete impulse, Eq. (4-58). Because we used direct substitution into Eq. (4-68), and this equation and Eq. (4-67) are a Fourier transform pair, it must follow that the right side of the double arrow is the DFT of the left: $\Im\left[e^{j2\pi(u_0x/M+v_0y/N)}\right] = MN\delta(u-u_0, v-v_0)$.

We could have solved this problem directly with the result of Problem 4.27(a) by letting $f(x, y) = 1$, and recognizing that when $f = 1$, then $F = \delta$ (see part (b)).

(f) We solve this problem by direct substitution into the forward DFT, Eq. (4-67), and using Euler's formula to express the sine in terms of exponentials:

$$\begin{aligned}\Im[\sin(2\pi\mu_0x/M + 2\pi v_0y/N)] &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\frac{e^{j2\pi(\mu_0x/M+v_0y/N)} - e^{-j2\pi(\mu_0x/M+v_0y/N)}}{2j}\right]e^{-j2\pi(ux/M+vy/N)} \\ &= \frac{1}{2j}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}e^{j2\pi(\mu_0x/M+v_0y/N)}e^{-j2\pi(ux/M+vy/N)} - \frac{1}{2j}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}e^{j2\pi(-\mu_0x/M-v_0y/N)}e^{-j2\pi(ux/M+vy/N)} \\ &= \frac{MN}{2j}[\delta(u-u_0, v-v_0) - \delta(u+u_0, v+v_0)] \\ &= \frac{jMN}{2}[\delta(u+u_0, v+v_0) - \delta(u-u_0, v-v_0)]\end{aligned}$$

where the second step puts the expressions in the form of Fourier transforms of exponentials. The results of these transforms are given in the statement of part (d), as the third step shows.

Problem 4.30

(a) The period, P, of a function $f(x)$ is such that $f(x+P) = f(x)$ for all x. The period is 2 units (or 2 samples because in this case the samples are the numbers themselves and they are one unit apart each) (i.e., in this case it takes two samples to complete each period). The frequency is the reciprocal of 1/2 cycle/unit or 1/2 cycle/sample or 1/2 period/sample.

Problem 4.31

(a) Consider a 4-point sequence. If the sequence is even, its values have to satisfy

$$f(0) = f(4), \qquad f(1) = f(3), \qquad f(2) = f(2), \qquad f(3) = f(1)$$

As we explained in Example 4.10, $f(4)$ is out of range. Thus, it does not matter what its value is, and $f(0)$ can always be equal to it. However, note that the third term (this is the term at $M/2 = 2$) requires that $f(2) = f(2)$. This is satisfied for any arbitrary number. This idea generalizes directly to any even value of M.

Problem 4.32

(a) The symmetry of this sequence is even, so the first element is arbitrary. In order to embed the sequence in a field of 0s, we set the first element to 0. The length of the sequence is 5 elements, so its center is at $M = 2$, which is the 3rd point in the sequence (remember, we start counting at 0). The center of a 1-D field of 0's nine elements long is at 4, which is the 5th point from the left. Thus, the sequence to embed is now $\{0, \ b, \ c, \ c, \ b\}$ which, when embedded (with coinciding centers) in a field of 9 zeros, looks like Fig. P4.32(a). The fact that that all paired elements have the same values proves that this is an even sequence. As usual, the first element of an even sequence can have any value.

(a)

Figure P4.32

Problem 4.34

(a) To determine if the following array

$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

is odd we apply the definition in Eq. (4-81). The array is odd if

$$w_o(x, y) = -w_o(M - x, N - y) = -w_o(6 - x, 6 - y)$$

for $x, y = 0, 1, 2, 3, 4, 5$. When $(x, y) = (0, 0)$, we have to test against $-w_o(6, 6)$, which is outside the range being examined, and could be any value, so the value of $w(0, 0)$ is immaterial in the test for oddness. The

same is true for all other terms in the first column or row of the array because one of the two coordinates is 0 there. Thus, all we have to do is test the preceding array for we can either test the above array for $x, y = 1,2,3,4,5$. For example, when $(x, y) = (3,2)$, we have $w_o(3,2) = -2$, and $w_o(6-3,6-2) = w_o(3,4) = 2$. In fact, if you compare all pairs you will find that $w_o(x, y) = -w_o(6-x,6-y)$, provided that you limit that values of the variables to $x, y = 1,2,3,4,5$. So, the array is indeed odd. Note that when we are talking about even and odd arrays with this test, we are referring both to the locations *and* values of the elements.

Problem 4.35

The following are proofs of some of the properties in Table 4.1. Proofs of the other properties are given in Chapter 4. Recall that when we refer to a function as imaginary, its real part is zero. We use the term complex to denote a function whose real and imaginary parts are not zero. We prove only the forward part the Fourier transform pairs. Similar techniques are used to prove the inverse part.

(a) *Property 2*: If $f(x, y)$ is imaginary, $f(x, y) \Leftrightarrow F^*(-u,-v) = -F(u,v)$. Because $f(x, y)$ is imaginary, we can express it as $jg(x, y)$, where $g(x, y)$ is a real function. Then the proof is as follows:

$$
\begin{aligned}
F^*(-u-v) &= \left[\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} jg(x,y)e^{j2\pi(ux/M+vy/N)}\right]^* \\
&= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} -jg(x,y)e^{-j2\pi(ux/M+vy/N)} \\
&= -\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [jg(x,y)]e^{-j2\pi(ux/M+vy/N)} \\
&= -\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)} \\
&= -F(u,v).
\end{aligned}
$$

(b) *Property 4*: If $f(x, y)$ is imaginary, then $R(u,v)$ is odd and $I(u,v)$ is even.

Proof: F is complex, so it can be expressed as

$$
\begin{aligned}
F(u,v) &= \text{real}[F(u,v)] + j\,\text{imag}[F(u,v)] \\
&= R(u,v) + jI(u,v).
\end{aligned}
$$

Then, $-F(u,v) = -R(u,v) - jI(u,v)$, and $F^*(-u,-v) = R(-u,-v) - j\,I(-u,-v)$. But, because $f(x, y)$ is imaginary, $F^*(-u,-v) = -F(u,v)$ (see Property 2). It then follows from the previous two equations that $R(u,v) = -R(-u,-v)$ (i.e., R is odd) and $I(u,v) = I(-u,-v)$ (I is even).

(d) *Property 7*: When $f(x,y)$ is complex, $f^*(x,y) \Leftrightarrow F*(-u,-v)$.

Proof:

$$\begin{aligned}\Im\left[f^*(x,y)\right] &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f^*(x,y)e^{-j2\pi(ux/M+vy/N)} \\ &= \left[\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{j2\pi(ux/M+vy/N)}\right]^* \\ &= F^*(-u,-v).\end{aligned}$$

Problem 4.36

(a) The average value of the original image is

$$\overline{f}(x,y) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

and, for the padded image,

$$\begin{aligned}\overline{f_p}(x,y) &= \frac{1}{PQ}\sum_{x=0}^{P-1}\sum_{y=0}^{Q-1} f_p(x,y) \\ &= \frac{1}{PQ}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \\ &= \frac{MN}{PQ}\overline{f}(x,y)\end{aligned}$$

where the second step is result of the fact that the image is padded with 0s. Thus, the ratio of the average values is

$$R = \frac{PQ}{MN}$$

The ratio increases as a function of PQ, indicating that the average value of the padded image decreases as a function of PQ. This is as expected; padding an image with zeros decreases its average value.

Problem 4.38

(b) Unlike part (a) where we substituted into the DFT, it is easier here to substitute the definition of the convolution, $F(u,v)\star H(u,v)$, directly into the IDFT:

$$
\begin{aligned}
\Im^{-1}\left[\frac{1}{MN}(F\star H)(u,v)\right] &= \Im^{-1}\left[\frac{1}{MN}F(u,v)\star H(u,v)\right] = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left[\frac{1}{MN}F(u,v)\star H(u,v)\right]e^{j2\pi(ux/M+vy/N)} \\
&= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left[\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}F(m,n)H(u-m,v-n)\right]e^{j2\pi(ux/M+vy/N)} \\
&= \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}F(m,n)\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}H(u-m,v-n)e^{j2\pi(ux/M+vy/N)} \\
&= \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}F(m,n)e^{j2\pi(um/M+vn/N)}h(x,y) \\
&= f(x,y)h(x,y) \\
&= (f\bullet h)(x,y)
\end{aligned}
$$

where the fourth step follows from the translation property (see Problem 4.27). Because we showed the preceding by substituting into the IDFT, and the DFT and IDFT are a transform pair, then the reverse, i.e., that $(F\star H)(u,v)$ is the DFT of $(f\bullet h)(x,y)$, must be true also.

Problem 4.39

(a) We solve this problem by substituting the definition of correlation from Table 4.3 directly into the DFT, Eq. (4-67)

$$
\begin{aligned}
\Im\left[(f\star h)(x,y)\right] &= \Im\left[f(x,y)\star h(x,y)\right] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\star h(x,y)e^{-j2\pi(ux/M+vy/N)} \\
&= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)h(x+m,y+n)\right]e^{-j2\pi(ux/M+vy/N)} \\
&= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}h(x+m,y+n)e^{-j2\pi(ux/M+vy/N)} \\
&= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)e^{j2\pi(um/M+vn/N)}H(u,v) \\
&= F^*(u,v)H(u,v) \\
&= (F^*\bullet H)(u,v)
\end{aligned}
$$

where the fourth step follows from the translation property (see Problem 4.27). Because we showed the preceding by substituting into the DFT, and the DFT and IDFT are a transform pair, then the reverse, i.e., that $f(x,y)\star h(x,y)$ is the IDFT of $F^*(u,v)H(u,v)$, must be true also.

Problem 4.40

We begin with one variable:

$$
\Im\left[\frac{df(z)}{dz}\right] = \int_{-\infty}^{\infty}\frac{df(z)}{dz}e^{-j2\pi vz}dz
$$

Integration by parts has the following general form

$$\int s\,dw = sw - \int w\,ds$$

Let $s = e^{-j2\pi\nu z}$ and $w = f(z)$. Then, $dw/dz = df(z)/dz$ or

$$dw = \frac{df(z)}{dz}dz \qquad \text{and} \qquad ds = (-j2\pi\nu)e^{-j2\pi\nu z}dz$$

so it follows that

$$\begin{aligned}\Im\left[\frac{df(z)}{dz}\right] &= \int_{-\infty}^{\infty}\frac{df(z)}{dz}e^{-j2\pi\nu z}dz \\ &= f(z)e^{-j2\pi\nu z}\Big]_{-\infty}^{\infty} - \int_{-\infty}^{\infty}f(z)(-j2\pi\nu)e^{-j2\pi\nu z}dz \\ &= (j2\pi\nu)\int_{-\infty}^{\infty}f(z)e^{-j2\pi\nu z}dz \\ &= (j2\pi\nu)F(\nu)\end{aligned}$$

because $f(\pm\infty) = 0$ by assumption (see Table 4.4). Consider next the second derivative. Define $g(z) = df(z)/dz$. Then,

$$\Im\left[\frac{dg(z)}{dz}\right] = (j2\pi\nu)G(\nu)$$

where $G(\nu)$ is the Fourier transform of $g(z)$. But $g(z) = df(z)/dz$, so $G(\nu) = (j2\pi\nu)F(\nu)$, and

$$\Im\left[\frac{d^2f(z)}{dz^2}\right] = (j2\pi\nu)^2F(\nu)$$

Continuing in this manner would result in the expression

$$\Im\left[\frac{d^nf(z)}{dz^n}\right] = (j2\pi\nu)^nF(\nu)$$

If we now go to 2-D and take the derivative of only one variable, we would get the same result as in the preceding expression, but we have to use partial derivatives to indicate the variable to which differentiation applies and, instead of $F(\mu)$, we would have $F(\mu,\nu)$. Thus,

$$\Im\left[\frac{\partial^n f(t,z)}{\partial z^n}\right] = (j2\pi\nu)^nF(\mu,\nu)$$

Define $g(t,z) = \partial^n f(t,z)/\partial t^n$. Then,

$$\Im\left[\frac{\partial^m g(t,z)}{\partial t^m}\right] = (j2\pi\mu)^m G(\mu,\nu)$$

But $G(\mu,\nu)$ is the transform of $g(t,z) = \partial^n f(t,z)/\partial t^n$, which we know is equal to $(j2\pi\mu)^n F(\mu,\nu)$. Therefore, we have established that

$$\Im\left[\left(\frac{\partial}{\partial t}\right)^m \left(\frac{\partial}{\partial z}\right)^n f(t,z)\right] = (j2\pi\mu)^m (j2\pi\nu)^n F(\mu,\nu)$$

Because the Fourier transform is unique, we know that the inverse transform of the right of this equation would give the left, so the equation constitutes a Fourier transform pair (keep in mind that we are dealing with continuous variables).

Problem 4.42

Unless all borders on of an image are black, padding the image with 0s introduces significant discontinuities (edges) at one or more borders of the image. These can be strong horizontal and vertical edges. These sharp intensity transitions in the spatial domain introduce strong high-frequency components along the vertical and horizontal axes of the spectrum.

Problem 4.44

The Fourier transform of the original image can be expressed in polar form as $F(u,v) = |F(u,v)|e^{j\phi(u,v)}$. The transform of the image in Fig. 4.34(b) is $|F(u,v)|e^{-j\phi(u,v)}$ (i.e., the phase angle in this image is the negative of the phase angle in the original). But this is the complex conjugate, $F^*(u,v)$. According to property 5 in Table 4.1, the inverse transform of $F^*(u,v)$ is $f(-x,-y)$, which is precisely the image in Fig. 4.34(b).

Problem 4.45

(a) $g(x,y) = -f(x,y)$.

Problem 4.47

(a) The spatial average (excluding the center term) is

$$g(x,y) = \frac{1}{4}\left[f(x,y+1) + f(x+1,y) + f(x-1,y) + f(x,y-1)\right]$$

From property 3 in Table 4.4,

$$\begin{aligned} G(u,v) &= \frac{1}{4}\left[e^{j2\pi v/N} + e^{j2\pi u/M} + e^{-j2\pi u/M} + e^{-j2\pi v/N}\right]F(u,v) \\ &= H(u,v)F(u,v) \end{aligned}$$

where

$$H(u,v) = \frac{1}{2}\left[\cos(2\pi u / M) + \cos(2\pi v / N)\right]$$

is the filter transfer function in the frequency domain.

(b) To see that this is a lowpass filter transfer function, consider its values in the range $[-M/2, M/2]$. The function assumes its highest value at the origin and decreases on either side of it, so it passes the dc term and low frequencies, and suppresses higher frequencies. Thus, it acts as a lowpass filter transfer function.

Problem 4.48

We want to show that

$$\Im^{-1}\left[Ae^{-(\mu^2+v^2)/2\sigma^2}\right] = A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)}.$$

The explanation will be clearer if we start with one variable. We want to show that, if

$$H(\mu) = e^{-\mu^2/2\sigma^2}$$

then

$$\begin{aligned} h(t) &= \Im^{-1}\left[H(\mu)\right] \\ &= \int_{-\infty}^{\infty} e^{-\mu^2/2\sigma^2} e^{j2\pi\mu t}\, d\mu \\ &= \sqrt{2\pi}\sigma^{-2\pi^2\sigma^2 t^2} \end{aligned}$$

We can express the integral in the preceding equations as

$$h(t) = \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}[\mu^2 - j4\pi\sigma^2\mu t]} d\mu$$

Making use of the identity

$$e^{-\frac{(2\pi)^2\sigma^2 t^2}{2}} e^{\frac{(2\pi)^2\sigma^2 t^2}{2}} = 1$$

in the preceding integral yields

$$h(t) = e^{-\frac{(2\pi)^2\sigma^2t^2}{2}}\int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}[\mu^2 - j4\pi\sigma^2\mu t - (2\pi)^2\sigma^4t^2]}d\mu$$
$$= e^{-\frac{(2\pi)^2\sigma^2t^2}{2}}\int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}[\mu - j2\pi\sigma^2t]^2}d\mu$$

Next, we make the change of variables $r = \mu - j2\pi\sigma^2 t$. Then, $dr = d\mu$ and the preceding integral becomes

$$h(t) = e^{-\frac{(2\pi)^2\sigma^2t^2}{2}}\int_{-\infty}^{\infty} e^{\frac{r^2}{2\sigma^2}}dr$$

Finally, we multiply and divide the right side of this equation by $\sqrt{2\pi}\sigma$ and obtain

$$h(t) = \sqrt{2\pi}\sigma e^{-\frac{(2\pi)^2\sigma^2t^2}{2}}\left[\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{\frac{r^2}{2\sigma^2}}dr\right]$$

The expression inside the brackets is recognized as the Gaussian probability density function whose value from $-\infty$ to ∞ is 1. Therefore,

$$h(t) = \sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2t^2}$$

With the preceding results as background, we are now ready to show that

$$h(t,z) = \Im^{-1}\left[Ae^{-(\mu^2+\nu^2)/2\sigma^2}\right]$$
$$= A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)}$$

By substituting directly into the definition of the inverse Fourier transform we have:

$$h(t,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Ae^{-(\mu^2+\nu^2)/2\sigma^2}e^{j2\pi(\mu t+\nu z)}d\mu d\nu$$
$$= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} Ae^{\left(-\frac{\mu^2}{2\sigma^2}+j2\pi\mu t\right)}d\mu\right]e^{\left(-\frac{\nu^2}{2\sigma^2}+j2\pi\nu z\right)}d\nu$$

The integral inside the brackets is recognized from the previous discussion to be equal to $A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2t^2}$. Then, the preceding integral becomes

$$h(t,z) = A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2t^2}\int_{-\infty}^{\infty} e^{\left(-\frac{\nu^2}{2\sigma^2}+j2\pi\nu z\right)}d\nu$$

We recognize the remaining integral to be equal to $\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2z^2}$, from which we have the final result:

$$h(t,z) = \left(A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2 t^2}\right)\left(\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2 z^2}\right)$$
$$= A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)}.$$

Problem 4.49

(a) One application of the filter gives

$$G(u,v) = H(u,v)F(u,v)$$
$$= e^{-D^2(u,v)/2D_0^2}F(u,v)$$

Similarly, K applications of the filter would give

$$G_K(u,v) = e^{-KD^2(u,v)/2D_0^2}F(u,v)$$

The inverse DFT of $G_K(u,v)$ would give the image resulting from K passes of the Gaussian filter. If K is "large enough," the Gaussian LPF will become a notch pass filter, passing only $F(0,0)$. We know that this term is equal to the average value of the image. So, there is a value of K after which the result of repeated lowpass filtering will simply produce a constant image. The value of all pixels in this image will be equal to the average value of the original image. Note that the answer applies even as K approaches infinity. In this case the filter will approach an impulse at the origin, and this would still give us $F(0,0)$ as the result of filtering.

Problem 4.52

(b) Generate a $P \times Q$ array centered on $[P/2, Q/2]$:

$$H(u,v) = -4\pi^2([u - P/2]^2 + [v - Q/2]^2)$$

for $u = 0, 1, 2, \ldots, P-1$ and $v = 0, 1, 2, \ldots, Q-1$ where P and Q are sizes to which the input image is padded prior to filtering. Then use $H(u,v)$ as any other filter transfer function.

Problem 4.53

The answer is no. The Fourier transform is a linear process, while the square and square roots involved in computing the gradient are nonlinear operations. The Fourier transform could be used to compute the derivatives as differences (as in Problem 4.50), but the squares and square root values, must be computed directly in the spatial domain.

Problem 4.56

$$
\begin{aligned}
H_{\text{HP}} &= 1 - H_{\text{LP}} \\
&= 1 - \frac{1}{1 + \left[D(u,v)/D_0\right]^{2n}} \\
&= \frac{\left[D(u,v)/D_0\right]^{2n}}{1 + \left[D(u,v)/D_0\right]^{2n}} \\
&= \frac{1}{\dfrac{1}{\left[D(u,v)/D_0\right]^{2n}} + \dfrac{\left[D(u,v)/D_0\right]^{2n}}{\left[D(u,v)/D_0\right]^{2n}}} \\
&= \frac{1}{1 + \left[D_0/D(u,v)\right]^{2n}}
\end{aligned}
$$

Problem 4.57

(a) The ring in fact has a dark center area as a result of the highpass operation only (the following image shows the result of highpass filtering only). However, the dark center area is averaged out by the lowpass filter. The reason the final result looks so bright is that the discontinuity (edge) on boundaries of the ring are much higher than anywhere else in the image, thus dominating the display of the result.

Figure P4.57

Problem 4.59

(a) Express filtering as convolution to reduce all processes to the spatial domain.

$$g(x,y) = h(x,y) \star f(x,y)$$

where h is the spatial filter (inverse Fourier transform of the frequency-domain filter) and f is the input image. Histogram processing this result yields

$$g'(x,y) = T\left[g(x,y)\right]$$
$$= T\left[h(x,y) \star f(x,y)\right]$$

where T denotes the histogram equalization transformation. If we histogram equalize first, then

$$g(x,y) = T\left[f(x,y)\right]$$

and

$$g'(x,y) = h(x,y) \star T\left[f(x,y)\right]$$

In general, T is a nonlinear function determined by the nature of the pixels in the image from which it is computed. Thus, in general, $T\left[h(x,y) \star f(x,y)\right] \neq h(x,y) \star T\left[f(x,y)\right]$ and the order does matter.

Problem 4.63

Because $M = 2^n$, we can write Eqs. (4-171) and (4-172) as

$$m(n) = \frac{1}{2}Mn \text{ and } a(n) = Mn$$

Proof by induction begins by showing that both equations hold for $n = 1$:

$$m(1) = \frac{1}{2}(2)(1) = 1 \text{ and } a(1) = (2)(1) = 2$$

We know these results to be correct from the discussion in Section 4.11. Next, we assume that the equations hold for n. Then, we are required to prove that they hold also for $n + 1$. From Eq. (4-169),

$$m(n+1) = 2m(n) + 2^n$$

Substituting for $m(n)$ from above,

$$m(n+1) = 2\left(\frac{1}{2}Mn\right) + 2^n$$
$$= 2\left(\frac{1}{2}2^n n\right) + 2^n$$
$$= 2^n(n+1)$$
$$= \frac{1}{2}\left(2^{n+1}\right)(n+1)$$

Therefore, Eq. (4-171) is valid for all $n \geq 1$.

From Eq. (4-170),

$$a(n+1) = 2a(n) + 2^{n+1}$$

Substituting the expression above for $a(n)$ yields

$$\begin{aligned} a(n+1) &= 2Mn + 2^{n+1} \\ &= 2(2^n n) + 2^{n+1} \\ &= 2^{n+1}(n+1) \end{aligned}$$

Therefore, Eq. (4-172) is valid for all $n \geq 1$. This completes the proof.

1 Introduction

1.1 Background

This manual contains solutions to the problems marked with an asterisk (*) in Digital Image Processing, Global Edition, 4th Edition.

1.2 The Book Website

Digital Image Processing is a completely self-contained book. However, the companion website offers additional support in a number of important areas. The URL of the site is

www.ImageProcessingPlace.com

For Students or Independent Reader the site contains

- Reviews in areas such as probability, statistics, vectors, and matrices.
- A Tutorials section containing dozens of tutorials on topics relevant to the material in the book.
- An image database containing all the images in the book, as well as many other image databases.

For Instructors the site contains

- An Instructor's Manual with complete solutions to all the problems in the book.
- The manual is available free of charge to instructors who have adopted the book for classroom use.
- Classroom presentation materials in PowerPoint format.
- Material removed from previous editions, downloadable in convenient PDF format.
- Numerous links to other educational resources.

For the Practitioner the site contains additional specialized topics such as

- Links to commercial sites.
- Selected new references.
- Links to commercial image databases.

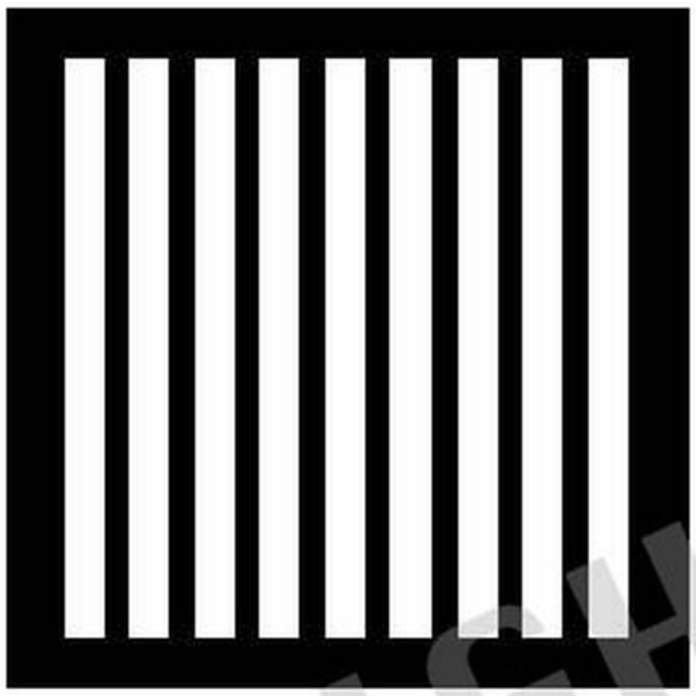
The website is an ideal tool for keeping the book current between editions by including new topics, digital images, and other relevant material that has appeared after the book was published. Although considerable care was taken in the production of the book, the website is also a convenient repository for any errors discovered between printings.

1.3 The DIP4E Support Packages

In this edition, we created support packages for students and faculty to organize all the classroom support materials available for the new edition of the book into one easy download. The Student Support Package contains most of the original images in the book and answers to homework problems marked with as asterisk (*) in the book. The Faculty Support Package contains solutions to all exercises, teaching suggestions, and all the art in the book in modifiable PowerPoint slides. One support package is made

available with every new book, free of charge. Applications for the support packages are submitted at the book website.

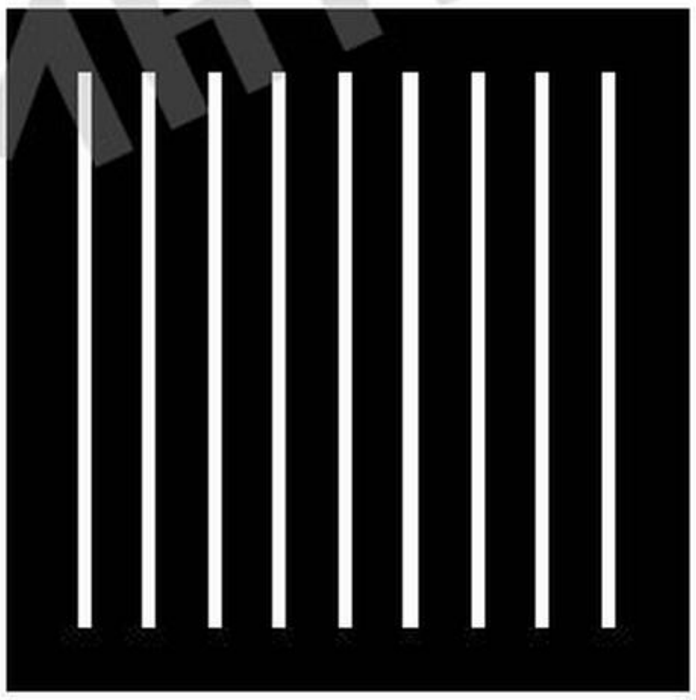
(b)



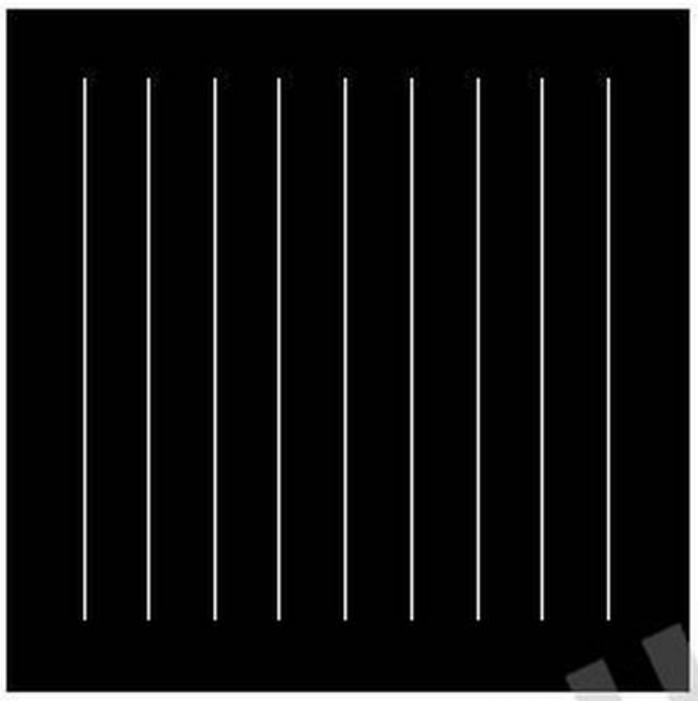
(c)

Problem 5.5

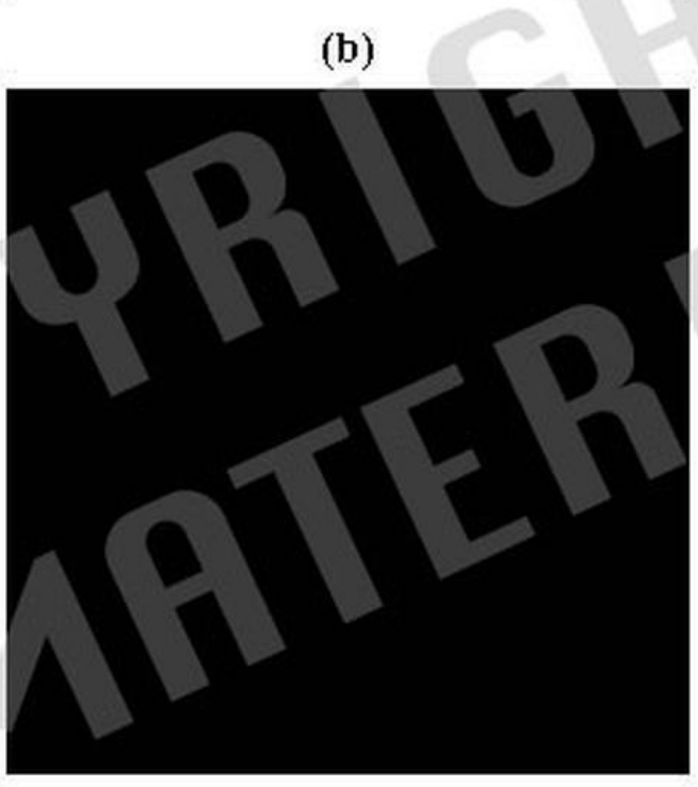
The solutions are as shown in the following figures. The student has two options: (1) give a brief word description that matches the salient differences between these images and the original image given in the problem statement, or (2) provide the solution images directly.



(a)



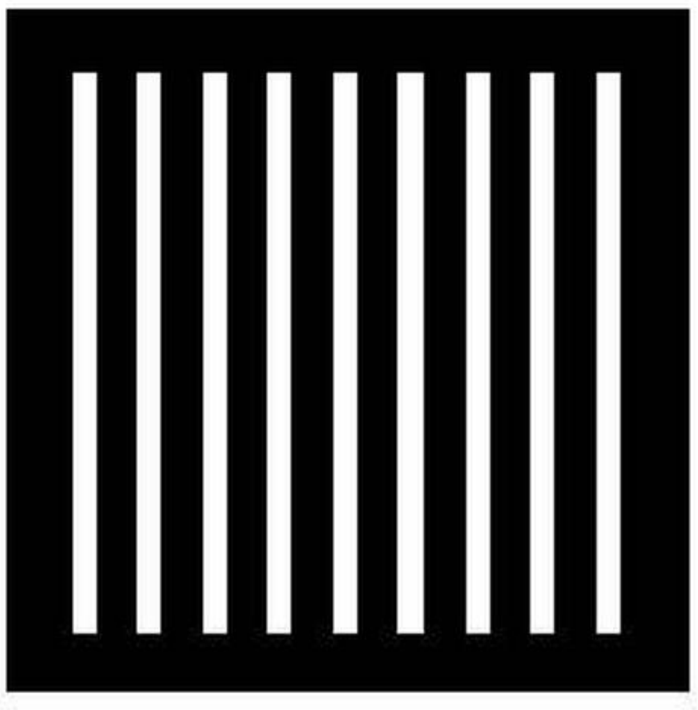
(b)



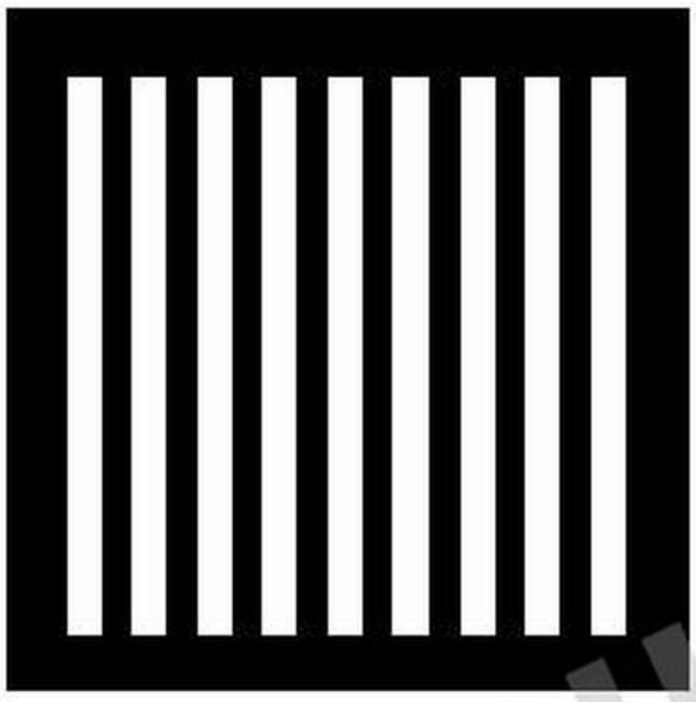
(c)

Figure 5.7

The solutions are as shown in the following figures. The student has two options: (1) give a brief word description that matches the salient differences between these images and the original image given in the problem statement, or (2) provide the solution images directly.



(a)

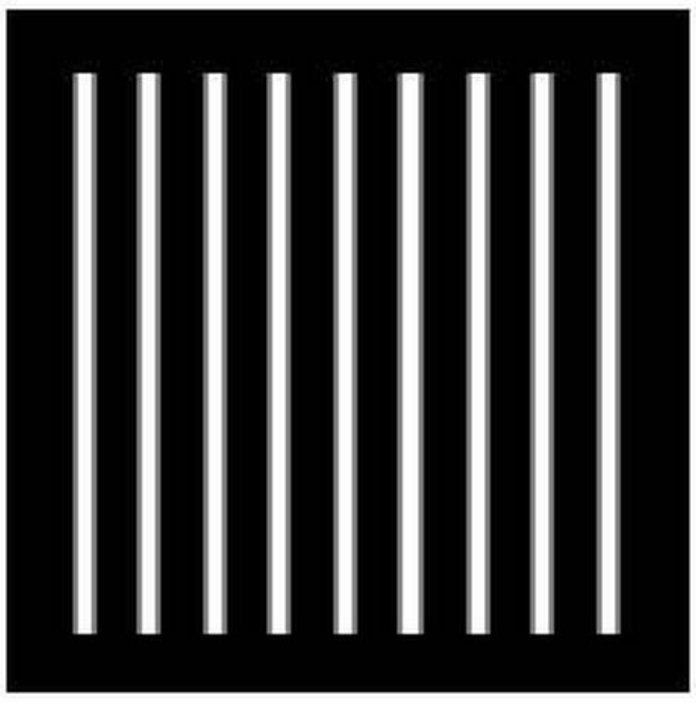


(b)

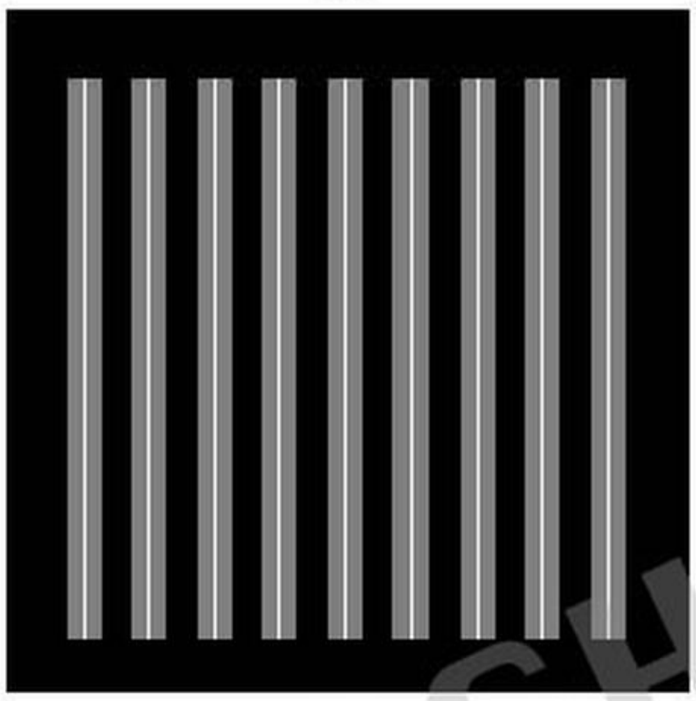
(c)

Problem 5.9

The solutions are as shown in the following figures. The student has two options: (1) give a brief word description that matches the salient differences between these images and the original image given in the problem statement, or (2) provide the solution images directly.



(a)



(b)

(c)

Problem 5.10

The key to understanding the behavior of the contra-harmonic filter is to think of the pixels in the neighborhood surrounding a noise impulse as being constant, with the impulse noise point being in the center of the neighborhood. For the noise spike to be visible, its value must be considerably larger than the value of its neighbors. Also keep in mind that the power in the numerator is 1 plus the power in the denominator.

(a) By definition, salt noise is a low value (really 0). It is most visible when surrounded by light values. The center pixel (the pepper noise) will have little influence in the sums. If the area spanned by the filter is approximately constant, the ratio will approach the value of the pixels in the neighborhood—thus reducing the effect of the low-value pixel. For example, here are some values of the filter for a dark point of value 1 in a 3×3 region with pixels of value 100: For $Q = 0.5$, filter = 98.78; for $Q = 1.0$, filter = 99.88, for $Q = 2.0$, filter = 99.99; and for $Q = 5.0$, filter = 100.00.

(c) When the wrong polarity is used, the large numbers in the case of the salt noise will be raised to a positive power, thus the noise will overpower the result. For salt noise the image will become very light. The opposite is true for pepper noise—the image will become dark.

Problem 5.12

(a) Setting $d = 0$ in Eq. (5-29) yields

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(m,n) \in S_{xy}} g_R(r,c)$$

where, because $d = 0$, and the alpha-trim filter deletes the lowest $d/2$ and the highest $d/2$ intensity values. $g_R(r,c)$ represents *all* the pixels in neighborhood S_{xy}. This is the definition of the arithmetic mean filter given in Eq. (5-23).

Problem 5.13

Bandpass filter transfer functions are obtained by subtracting the corresponding bandreject functions from 1:

$$H_{\text{BP}}(u,v) = 1 - H_{\text{BR}}(u,v)$$

(b) Gaussian bandpass filter transfer function.

$$H(u,v) = 1 - \left[1 - e^{-\left[\frac{D^2(u,v) - C_0^2}{D(u,v)W}\right]^2}\right]$$

$$= e^{-\left[\frac{D^2(u,v) - C_0^2}{D(u,v)W}\right]^2}$$

Problem 5.14

The general form of a notch filter transfer function with Q notch pairs is

$$H_{\text{NR}}(u,v) = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v)$$

where $H_k(u,v)$ and $H_{-k}(u,v)$ are highpass filter transfer functions with centers at (u_k, v_k) and $(-u_k, -v_k)$, respectively.

(a) An ideal notch filter transfer function has the form shown above, with

$$H_k(u,v) = \begin{cases} 0 & \text{if } D_k(u,v) \le D_0 \\ 1 & \text{if } D_k(u,v) > D_0 \end{cases}$$

$$H_{-k}(u,v) = \begin{cases} 0 & \text{if } D_{-k}(u,v) \le D_0 \\ 1 & \text{if } D_{-k}(u,v) > D_0 \end{cases}$$

where

$$D_k(u,v) = \left[(u - M/2 - u_k)^2 + (v - N/2 - v_k)^2\right]^{1/2}$$

and

$$D_{-k}(u,v) = \left[(u - M/2 + u_k)^2 + (v - N/2 + v_k)^2\right]^{1/2}$$

Problem 5.16

(a) The plot would be an integer number of complete periods of the sine wave.

(b) A pair of pure conjugate impulses located at u_0 and $-u_0$.

Problem 5.17

From Eq. (5-46),

$$\sigma^2(x,y) = \frac{1}{mn}\sum\left\{[g(\gamma) - w\eta(\gamma)] - [\bar{g} - w\bar{\eta}]\right\}^2$$

where, to simplify the notation we used "γ" to denote the terms affected by the summation. Letting $K = 1/mn$, taking the partial derivative of σ^2 with respect to w and setting the result equal to zero gives

$$\begin{aligned}
\frac{\partial\sigma^2}{\partial w} &= K\sum 2[g(\gamma) - w\eta(\gamma) - \bar{g} + w\bar{\eta}][-\eta(\gamma) + \bar{\eta}] = 0 \\
&= K\sum -g(\gamma)\eta(\gamma) + g(\gamma)\bar{\eta} + w\eta^2(\gamma) - w\eta(\gamma)\bar{\eta} + \bar{g}\eta(\gamma) - \bar{g}\bar{\eta} - w\bar{\eta}\eta(\gamma) + w\bar{\eta}^2 \\
&= -\overline{g\eta} + \bar{g}\bar{\eta} + w\overline{\eta^2} - w\bar{\eta}^2 + \bar{g}\bar{\eta} - \bar{g}\bar{\eta} - w\bar{\eta}^2 + w\bar{\eta}^2 \\
&= -\overline{g\eta} + \bar{g}\bar{\eta} + w(\overline{\eta^2} - \bar{\eta}^2) = 0
\end{aligned}$$

where, for example, we used the fact that

$$\frac{1}{mn}\sum g(\gamma)\eta(\gamma) = \overline{g\eta}$$

Solving for w gives us

$$w = \frac{\overline{g\eta} - \overline{g}\,\overline{\eta}}{\overline{\eta^2} - \overline{\eta}^2}$$

which agrees with Eq. (5-48).

Problem 5.19

(a) The system transfer function is the Fourier transform of the impulse response. From entries 3 and 8 in Table 4.4,

$$H(u,v) = \Im\{\delta(x-a, y-b)\} = 1e^{-j2\pi(ua/M + vb/M)}$$

(b) The input in the frequency domain is $F(u,v) = \Im\{K\} = K\delta(u,v)$, and the output is

$$\begin{aligned} G(u,v) &= H(u,v)F(u,v) \\ &= e^{-j2\pi(ua/M + vb/M)} K\delta(u,v) \\ &= K\delta(u,v)e^{-j2\pi(ua/M + vb/M)} \end{aligned}$$

From entry 3 in Table 4.4, the output in the spatial domain is

$$g(x,y) = \Im^{-1}\{K\delta(u,v)e^{-j2\pi(ua/M + vb/M)}\} = K$$

Problem 5.20

The solution to Problem 5.19(b) using Eq. (5-61) directly is as follows:

$$\begin{aligned} g(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x-\alpha, y-\beta)d\alpha d\beta \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} K\delta(x-a-\alpha, y-b-\beta)d\alpha d\beta \\ &= K \end{aligned}$$

because K is independent of a, b, α, and β. This agrees with the solution in Problem 5.19(b). Similarly, the solution to Problem 5.19(c) is

$$\begin{aligned} g(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x-\alpha, y-\beta)d\alpha d\beta \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(\alpha,\beta)\delta(x-a-\alpha, y-b-\beta)d\alpha d\beta \\ &= \delta(-x+a, -y+b) \end{aligned}$$

because $\delta(x-a-\alpha, y-b-\beta)$ is 0 unless $x-a=-\alpha$ and $y-b=-\beta$. The above solution agrees with the solution to Problem 5.19(c) because $\delta(-x+a,-y+b)=\delta(x-a,y-b)$ (see Problem 4.1(c)).

Problem 5.21

From Eq. (5-61),

$$g(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x-\alpha,y-\beta)d\alpha d\beta$$

It is given that $f(x,y)=\delta(x-a)$, so $f(\alpha,\beta)=\delta(\alpha-a)$. Then, using the impulse response given in the problem statement,

$$\begin{aligned} g(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(\alpha-a)e^{-\left[(x-\alpha)^2+(y-\beta)^2\right]}d\alpha d\beta \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(\alpha-a)e^{-\left[(x-\alpha)^2\right]}e^{-\left[(y-\beta)^2\right]}d\alpha d\beta \\ &= \int_{-\infty}^{\infty} \delta(\alpha-a)e^{-\left[(x-\alpha)^2\right]}d\alpha \int_{-\infty}^{\infty} e^{-\left[(y-\beta)^2\right]}d\beta \\ &= e^{-\left[(x-a)^2\right]}\int_{-\infty}^{\infty} e^{-\left[(y-\beta)^2\right]}d\beta \end{aligned}$$

where we used the fact that the integral of the impulse is nonzero only when $\alpha=a$. Next, we note that

$$\int_{-\infty}^{\infty} e^{-\left[(y-\beta)^2\right]}d\beta=\int_{-\infty}^{\infty} e^{-\left[(\beta-y)^2\right]}d\beta$$

which is in the form of a constant times a Gaussian density with variance $\sigma^2=1/2$ or standard deviation $\sigma=1/\sqrt{2}$. In other words,

$$e^{-\left[(\beta-y)^2\right]}=\sqrt{2\pi(1/2)}\left[\frac{1}{\sqrt{2\pi(1/2)}}e^{-(1/2)\left[\frac{(\beta-y)^2}{(1/2)}\right]}\right]$$

The quantity inside the square brackets is a Gaussian density, and we know that its integral from minus to plus infinity is 1 so we have that

$$g(x,y)=\sqrt{\pi}e^{-\left[(x-a)^2\right]}$$

which is a blurred version of the original image. This is what we would expect because the impulse response of the system is in the form of a lowpass filter transfer function.

Problem 5.24

(a) Blurring would be in the negative x-direction (up in the picture) and positive y-direction (to the right), as Fig. P5.24(a) shows.

Figure P0524(a)

Problem 5.25

Following the image coordinate convention in the book, vertical motion is in the x-direction and horizontal motion is in the y-direction. Then, the components of motion are as follows:

$$x_0(t) = \begin{cases} at/T_1 & 0 \le t \le T_1 \\ a & T_1 < t \le T_1 + T_2 \end{cases}$$

and

$$y_0(t) = \begin{cases} 0 & 0 \le t \le T_1 \\ \dfrac{b(t - T_1)}{T_1} & T_1 < t \le T_1 + T_2 \end{cases}$$

Then, substituting these components of motion into Eq. (5-74) yields

$$\begin{aligned} H(u,v) &= \int_0^{T_1} e^{-j2\pi[uat/T_1]} dt + \int_{T_1}^{T_1+T_2} e^{-j2\pi[ua + vb(t-T_1)/T2]} dt \\ &= \frac{T_1}{\pi ua} \sin(\pi ua) e^{-j\pi ua} + e^{-j\pi ua} \int_{T_1}^{T_1+T_2} e^{-j2\pi vb(t-T_1)/T2} dt \\ &= \frac{T_1}{\pi ua} \sin(\pi ua) e^{-j\pi ua} + e^{-j\pi ua} \int_0^{T_2} e^{-j2\pi vb\tau/T2} d\tau \\ &= \frac{T_1}{\pi ua} \sin(\pi ua) e^{-j\pi ua} + e^{-j\pi ua} \frac{T_2}{\pi vb} \sin(\pi vb) e^{-j\pi vb} \end{aligned}$$

where in the third line we made the change of variables $\tau = t - T_1$. The blurred image is then

$$g(x,y) = \Im^{-1}\{H(u,v)F(u,v)\}$$

where $F(u,v)$ is the Fourier transform of the input image.

Problem 5.27

From Eq. (5-74),

$$H(u,v) = \int_0^T e^{-j2\pi u x_0(t)} dt = \int_0^T e^{-j2\pi u\left[at^2/2\right]} dt$$
$$= \int_0^T e^{-j\pi u a t^2} dt$$

Using Euler's formula we obtain

$$\int_0^T e^{-j\pi u a t^2} dt = \int_0^T \left[\cos(\pi u a t^2) - j\sin(\pi u a t^2)\right] dt$$
$$= \sqrt{\frac{T^2}{2\pi u a T^2}}\left[C(\sqrt{\pi u a}T) - jS(\sqrt{\pi u a}T)\right]$$

where the forms

$$C(z) = \int_0^z \cos t^2 dt$$

and

$$S(z) = \int_0^z \sin t^2 dt$$

are Fresnel cosine and sine integrals. They can be found, for example, the Handbook of Mathematical Functions, by Abramowitz, or other similar reference.

Problem 5.29

Measure the average value of the background. Set all pixels in the image, except the cross hairs, to that intensity value. Denote the Fourier transform of this image by $G(u,v)$. Because the characteristics of the cross hairs are given with a high degree of accuracy, we can construct an image of the background (of the same size) using the background intensity levels determined previously. We then construct a model of the cross hairs in the correct location (determined from the given image) using the dimensions provided and intensity level of the cross hairs. Denote by $F(u,v)$ the Fourier transform of this new image . The ratio $G(u,v)/F(u,v)$ is an estimate of the blurring function $H(u,v)$. In the likely event of vanishing values in $F(u,v)$, we can construct a radially-limited filter using the method discussed in connection with Fig. 5.27. Because we know $F(u,v)$ and $G(u,v)$, and an estimate of $H(u,v)$, we can refine our estimate of the blurring function by substituting G and H in Eq. (5-85) and adjusting K to get as close as possible to a good result for $F(u,v)$ (the result can be evaluated visually by taking the inverse Fourier transform). The resulting filter in either case can then be used to deblur the image of the heart, if desired.

Problem 5.32

This is a simple plug in problem. Its purpose is to gain familiarity with the various terms of the **W**iener filter. From Eq. (5-81),

$$\begin{aligned} H_{\mathbf{W}}(u,v) &= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] \\ &= \left[\frac{1}{H(u,v)}\frac{H(u,v)H(u,v)}{|H(u,v)|^2 + K}\right] \\ &= \frac{H(u,v)}{|H(u,v)|^2 + K} \end{aligned}$$

because $H(u,v)$ is real. Then, using the results from Problem 5.31,

$$\begin{aligned} |H(u,v)|^2 &= H^*(u,v)H(u,v) \\ &= H^2(u,v) \\ &= 64\pi^8\sigma^4(u^2 + v^2)^2 e^{-4\pi^2\sigma^2(u^2 + v^2)} \end{aligned}$$

and

$$H_{\mathbf{W}}(u,v) = \frac{-8\pi^4\sigma^2(u^2 + v^2)e^{-2\pi^2\sigma^2(u^2+v^2)}}{\left[64\pi^8\sigma^4(u^2 + v^2)^2 e^{-4\pi^2\sigma^2(u^2 + v^2)}\right] + K}$$

Problem 5.36

Because the system is assumed linear and position invariant, it follows that Eq. (5-63) holds. Furthermore, we can use superposition (additivity, as discussed in Section 2.6) and obtain the response of the system first to $F(u,v)$ and then to $N(u,v)$ because we know that the image and noise are uncorrelated. The sum of the two individual responses then gives the complete response. First, using only $F(u,v)$,

$$G_1(u,v) = H(u,v)F(u,v)$$

and

$$|G_1(u,v)|^2 = |H(u,v)|^2 |F(u,v)|^2$$

Then, using only $N(u,v)$,

$$G_2(u,v) = N(u,v)$$

and

$$|G_2(u,v)|^2 = |N(u,v)|^2$$

so that

$$\begin{aligned}|G(u,v)|^2 &= |G_1(u,v)|^2 + |G_2(u,v)|^2 \\ &= |H(u,v)|^2 |F(u,v)|^2 + |N(u,v)|^2\end{aligned}$$

Problem 5.37

(a) It is given that

$$|\hat{F}(u,v)|^2 = |R(u,v)|^2 |G(u,v)|^2$$

Because the image and noise are uncorrelated, it follows from Problem 5.36 that

$$|\hat{F}(u,v)|^2 = |R(u,v)|^2 \left[|H(u,v)|^2 |F(u,v)|^2 + |N(u,v)|^2\right]$$

Forcing $|\hat{F}(u,v)|^2$ to equal $|F(u,v)|^2$ gives

$$R(u,v) = \left[\frac{|F(u,v)|^2}{|H(u,v)|^2 |F(u,v)|^2 + |N(u,v)|^2}\right]^{1/2}$$

Problem 5.39

The basic idea behind this problem is to use the camera and representative coins to model the degradation process and then utilize the results in an inverse filter operation. The principal steps are as follows:

(1) Select coins as close as possible in size and content as the lost coins. Select a background that approximates the texture and brightness of the photos of the lost coins.

(2) Set up the museum photographic camera in a geometry as close as possible to give images that resemble the images of the lost coins (this includes paying attention to illumination). Obtain a few test photos. To simplify experimentation, obtain a TV camera capable of giving images that resemble the test photos. This can be done by connecting the camera to an image processing system and generating digital images, which will be used in the experiment.

(3) Obtain sets of images of each coin with different lens settings. The resulting images should approximate the aspect angle, size (in relation to the area occupied by the background), and blur of the photos of the lost coins.

(4) The lens setting for each image in (3) is a model of the blurring process for the corresponding image of a lost coin. For each such setting, remove the coin and background and replace them with a small, bright dot

on a uniform background, or other mechanism to approximate an impulse of light. Digitize the impulse. Its Fourier transform is the transfer function of the blurring process, $H(u,v)$.

(5) Digitize each (blurred) photo of a lost coin, and obtain its Fourier transform. At this point, we have $G(u,v)$ for each coin.

(6) Obtain an approximation to $F(u,v)$ by using a Wiener filter. Equation (5-85) is particularly attractive because it gives an additional degree of freedom (K) for experimenting.

(7) The inverse Fourier transform of each approximation $\hat{F}(u,v)$ gives the restored image for a coin. In general, several experimental passes of these basic steps with various different settings and parameters are required to obtain acceptable results in a problem such as this.

Problem 5.41

Figure P5.41 shows the solution. To see why the Radon transform of an image containing a single point in the center is a straight line, refer to Figs. 5.36, 5.37, and Eq. (5-101). This equation tells us that we compute the Radon transform by, for every angle θ, varying ρ to find the sum across the image along line $L(\theta,\rho)$. But, for the given image, and any angle θ, the only line that will contain the single point is the line for which $\rho = \sqrt{2}M/2$ (the point is located half-way on the diagonal of the square image). Furthermore, the sum for any value of θ will be the same an equal to the intensity of the point. Therefore, the Radon transform will be an image of a straight line of constant intensity located at $\rho = \sqrt{2}M/2$ as shown in the figure.

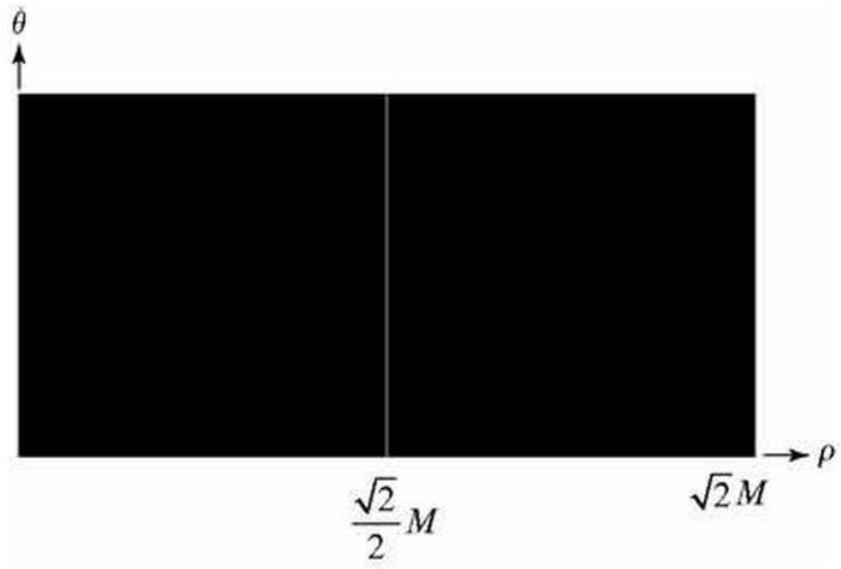


Figure P5.41

Problem 5.42

We know from Fig. 5.38 what the cross section of a solid white disk looks like. Inserting an inner black disk reduces the values of the sum in the Radon transform in that area. Thus, the values of the cross section corresponding to the small black area are reduced, as Fig. P5.42 shows.

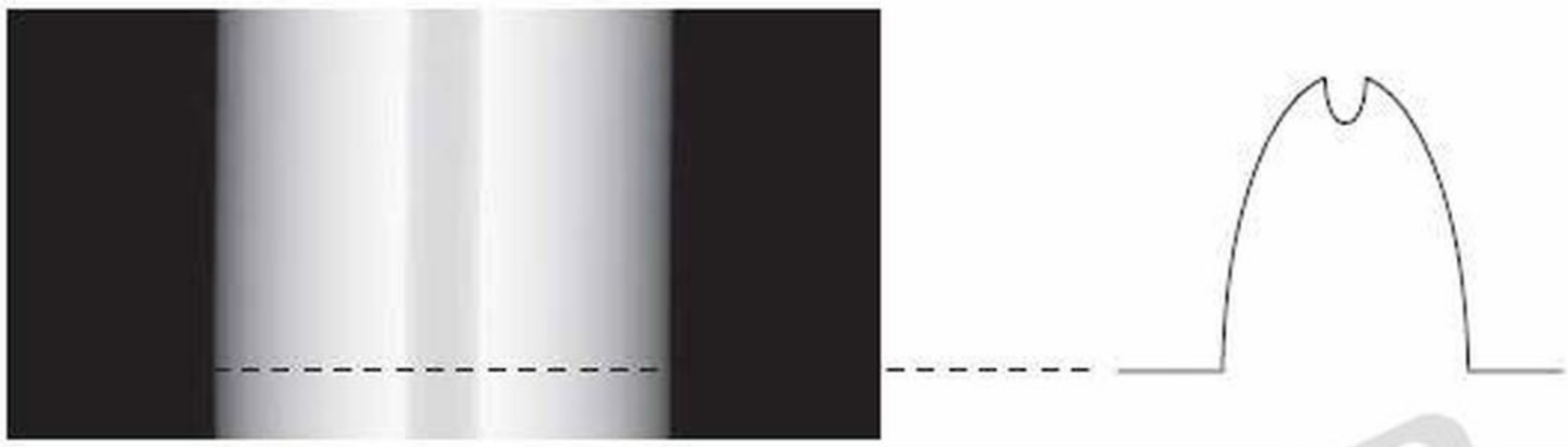


Figure P5.42

Problem 5.44

(a) From Eq. (5-102),

$$\Re\{f(x,y)\} = g(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} 1\times\delta(0 - \rho)dxdy$$
$$= \begin{cases} 1 & \text{if } \rho = 0 \\ 0 & \text{otherwise} \end{cases}$$

which is the equation of a vertical line through the origin of the $\rho\theta$-plane.

Problem 5.45

(a) From Section 2.6, we know that an operator, $\mathcal{H}$, is linear if $\mathcal{H}(af_1 + bf_2) = a\mathcal{H}(f_1) + b\mathcal{H}(f_2)$. From the definition of the Radon transform, Eq. (5-102),

$$\mathcal{H}(af_1 + bf_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (af_1 + bf_2)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$
$$= a\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_1\delta(x\cos\theta + y\sin\theta - \rho)dxdy + b\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_2\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$
$$= \mathcal{H}(af_1) + b\mathcal{H}(f_2)$$

thus showing that the Radon transform is a linear operator.

(c) We know from Chapter 4 that the convolution of two functions f and h is defined as

$$c(x,y) = f(x,y) \star h(x,y) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha,\beta) h(x - \alpha, y - \beta) d\alpha d\beta$$

We want to show that $\Re\{c\} = \Re\{f\} \star \Re\{h\}$, where $\Re$ denotes the Radon transform. We do this by substituting the convolution expression into Eq. (5-102). That is,

$$\Re\{c\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha,\beta) h(x - \alpha, y - \beta) d\alpha d\beta \right] \times \delta(x\cos\theta + y\sin\theta - \rho) dx dy$$

Rearranging terms we obtain

$$\Re\{c\} = \int_{\alpha} \int_{\beta} f(\alpha,\beta) \left[\int_{x} \int_{y} h(x - \alpha, y - \beta) \delta(x\cos\theta + y\sin\theta - \rho) dx dy \right] d\alpha d\beta$$

where we used the subscripts in the integrals for clarity between the integrals and their variables. All integrals are understood to be between $-\infty$ ∞. Working with the integrals inside the brackets with $x' = x - \alpha$ and $y' = y - \beta$ we have

$$\begin{aligned} &\int_{x} \int_{y} h(x - \alpha, y - \beta) \delta(x\cos\theta + y\sin\theta - \rho) dx dy \\ &= \int_{x'} \int_{y'} h(x', y') \delta(x'\cos\theta + y'\sin\theta - [\rho - \alpha\cos\theta - \beta\sin\theta) dx' dy' \\ &= \Re\{h\}(\rho - \alpha\cos\theta - \beta\sin\theta, \theta) \end{aligned}$$

We recognize the second integral as being the Radon transform of h, but, instead of being a function of ρ and θ, it is a function of $\rho - \alpha\cos\theta - \beta\sin\theta$ and θ. The notation in the last line of the previous equation is used to indicate "the Radon transform of h as function of $\rho - \alpha\cos\theta - \beta\sin\theta$ and θ." Then,

$$\begin{aligned} \Re\{c\} &= \int_{\alpha} \int_{\beta} f(\alpha,\beta) \left[\int_{x} \int_{y} h(x - \alpha, y - \beta) \delta(x\cos\theta + y\sin\theta - \rho) dx dy \right] d\alpha d\beta \\ &= \int_{\alpha} \int_{\beta} f(\alpha,\beta) \Re\{h\}(\rho - \rho', \theta) d\alpha d\beta \end{aligned}$$

where $\rho' = \alpha\cos\theta + \beta\sin\theta$. Then, based on the properties of the impulse, we can write,

$$\Re\{h\}(\rho-\rho',\theta)=\int_{\rho'}\Re\{h\}(\rho-\rho',\theta)\delta(x\cos\theta+y\sin\theta-\rho')d\rho'$$

Then,

$$\begin{aligned}\Re\{c\}&=\int_\alpha\int_\beta f(\alpha,\beta)[\Re(\rho-\rho',\theta)]d\alpha d\beta\\&=\int_\alpha\int_\beta f(\alpha,\beta)\left[\int_{\rho'}\Re\{h\}(\rho-\rho',\theta)\delta(\alpha\cos\theta+\beta\sin\theta-\rho')d\rho'\right]d\alpha d\beta\\&=\int_{\rho'}\Re\{h\}(\rho-\rho',\theta)\left[\int_\alpha\int_\beta f(\alpha,\beta)\delta(\alpha\cos\theta+\beta\sin\theta-\rho')d\alpha d\beta\right]d\rho'\\&=\int_{\rho'}\Re\{h\}(\rho-\rho',\theta)\Re\{f\}(\rho',\theta)d\rho'=\int_{\rho'}\Re\{f\}(\rho',\theta)\Re\{h\}(\rho-\rho',\theta)d\rho'\\&=\Re\{f\}\star\Re\{h\}\end{aligned}$$

where the fourth step follows from the definition of the Radon transform and the fifth step follows from the definition of convolution. This completes the proof.

Problem 5.47

The argument of function s in Eq. (5-122) may be written as

$$r\cos(\beta+\alpha-\varphi)-D\sin\alpha=r\cos(\beta-\varphi)\cos\alpha-[r\sin(\beta-\varphi)+D]\sin\alpha$$

From Fig. 5.47,

$$\begin{aligned}R\cos\alpha'&=R+r\sin(\beta-\varphi)\\R\sin\alpha'&=r\cos(\beta-\varphi)\end{aligned}$$

Then, substituting in the earlier expression,

$$\begin{aligned}r\cos(\beta+\alpha-\varphi)-D\sin\alpha&=R\sin\alpha'\cos\alpha-R\cos\alpha'\sin\alpha\\&=R(\sin\alpha'\cos\alpha-\cos\alpha'\sin\alpha)\\&=R\sin(\alpha'-\alpha)\end{aligned}$$

which agrees with Eq. (5-125).

Chapter 6

Problem Solutions

Problem 6.2

Denote by c the given color, and let its coordinates be denoted by (x_0, y_0). The distance between c and c_1 is

$$d(c,c_1) = \left[(x_0 - x_1)^2 + (y_0 - y_1)^2\right]^{1/2}$$

Similarly, the distance between c_1 and c_2 is

$$d(c_1,c_2) = \left[(x_1 - x_2)^2 + (y_1 - y_2)^2\right]^{1/2}$$

The percentage p_1 of c_1 in c is

$$p_1 = \frac{d(c_1,c_2) - d(c,c_1)}{d(c_1,c_2)} \times 100$$

The percentage p_2 of in c_2 is then $p_2 = 100 - p_1$. In the preceding equation we see, for example, that when $c = c_1$, $d(c,c_1) = d(c_1,c_1) = 0$, and it follows that $p_2 = 100 - p_1 = 0\%$. Similarly, when $d(c,c_2) = d(c_1,c_2)$, it follows that $p_1 = 0\%$ and $p_2 = 100\%$. Values in between are easily seen to follow from this simple equation.

Problem 6.4

Use color filters that are tuned to the wavelengths of the colors of the three objects. With a specific filter in place, only the objects whose color corresponds to that wavelength will produce a significant response on the monochrome camera. A motorized filter wheel can be used to control filter position from a computer. If one of the colors is white, then the response of the three filters will be approximately equal and high. If one of the colors is black, the response of the three filters will be approximately equal and low.

Problem 6.6

For the image given, the maximum intensity and saturation requirement means that the RGB component values are 0 or 1. We can create Table P6.6 with 0 and 255 representing black and white, respectively. Thus, we get the monochrome displays shown in Fig. P6.6.

Table P6.6

Color	R	G	B	Mono R	Mono G	Mono B
Black	0	0	0	0	0	0
Red	1	0	0	255	0	0
Yellow	1	1	0	255	255	0
Green	0	1	0	0	255	0
Cyan	0	1	1	0	255	255
Blue	0	0	1	0	0	255
Magenta	1	0	1	255	0	255
White	1	1	1	255	255	255
Gray	0.5	0.5	0.5	128	128	128

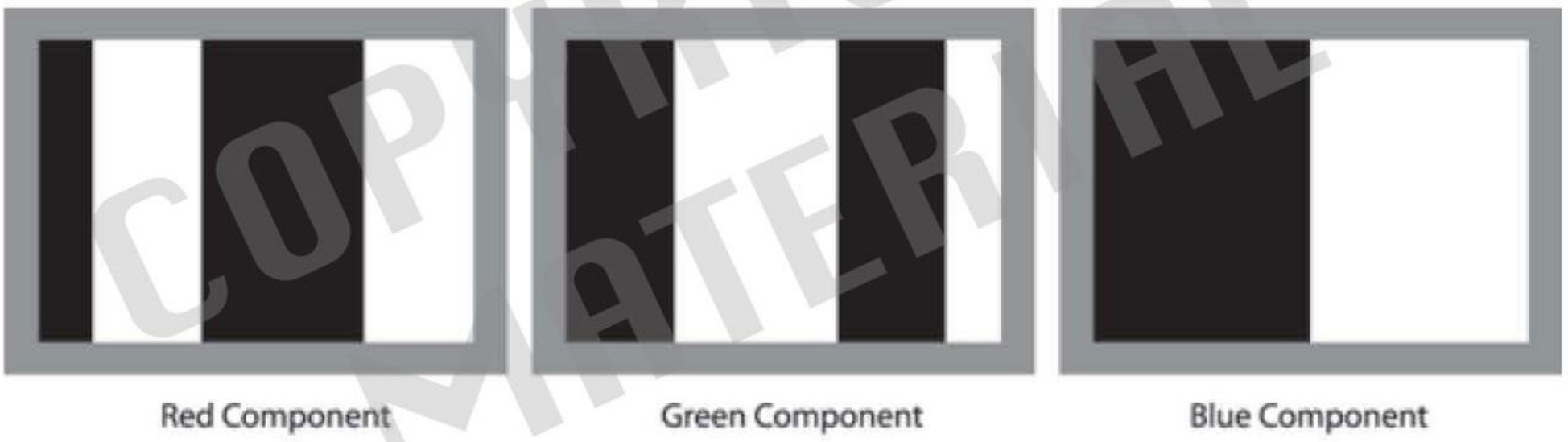


Figure P6.6

Problem 6.8

(a) All pixel values in the Red image are 255. In the Green image, the first column is all 0's; the second column all 1's; and so on until the last column, which is composed of all 255's. In the Blue image, the first row is all 255's; the second row all 254's, and so on until the last row which is composed of all 0's.

Problem 6.9

(a) For the image in Problem 6.6, the maximum intensity and saturation requirement means that the RGB component values are 0 or 1. Based on this, we can create Table P6.9 using Eq. (7-5), where we assumed 8-bit images, in which 1 maps to 255 (white) and 0 maps to 0 (black), and 0.5 maps to 128 (or 127, depending on how you round fractions). A monochrome monitor has a single input, so we can only feed one of the three CMY channels at a time. Thus, we get the monochrome displays shown in Fig. P6.9 from left to right for each of these three components.

Table P6.9

Color	R	G	B	C	M	Y	Mono C	Mono M	Mono Y
Black	0	0	0	1	1	1	255	255	255
Red	1	0	0	0	1	1	0	255	255
Yellow	1	1	0	0	0	1	0	0	255
Green	0	1	0	1	0	1	255	0	255
Cyan	0	1	1	1	0	0	255	0	0
Blue	0	0	1	1	1	0	255	255	0
Magenta	1	0	1	0	1	0	0	255	0
White	1	1	1	0	0	0	0	0	0
Gray	0.5	0.5	0.5	0.5	0.5	0.5	128	128	128

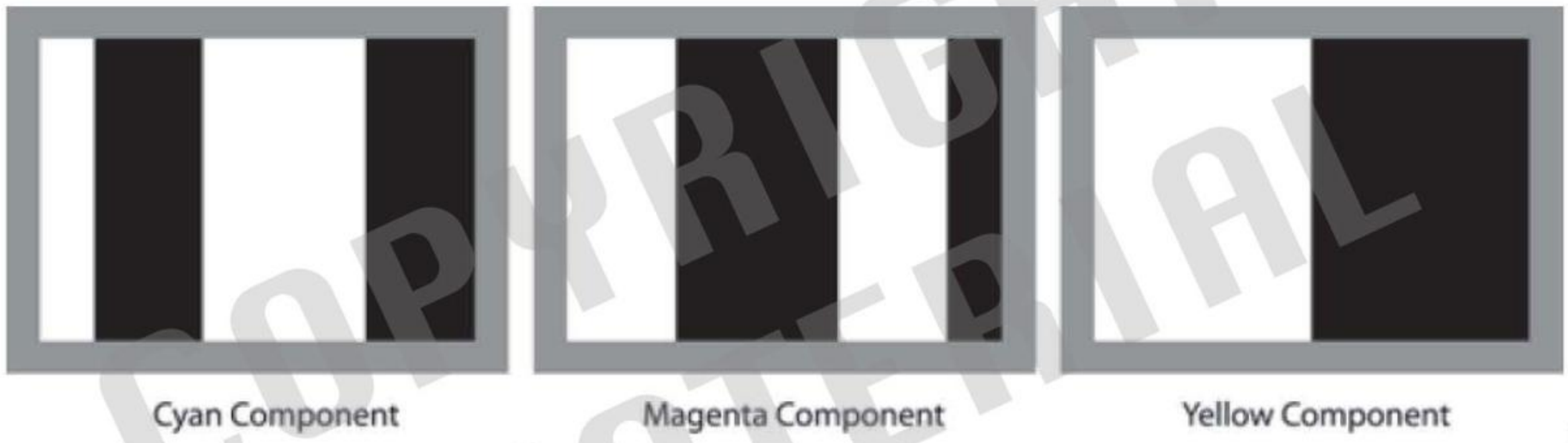


Figure P6.9

Problem 6.10

Using Eqs. (7-16) through (7-19), we get the results in Table P6.10. Based on Eq. (7-16), we see that hue is undefined when $R = G = B$ because $\theta = \cos^{-1}(0/0)$. In addition, saturation is undefined when $R = G = B = 0$ because Eq. (7-18) yields $S = 1 - (3/0)[\min(0,0,0) = 1 - (0/0)$. Thus, we get the monochrome display shown in Fig. P6.10.

Table P6.10

Color	R	G	B	H	S	I	Mono H	Mono S	Mono I
Black	0	0	0	–	0	0	–	–	0
Red	1	0	0	0	1	0.33	0	255	85
Yellow	1	1	0	0.17	1	0.67	43	255	170
Green	0	1	0	0.33	1	0.33	85	255	85
Cyan	0	1	1	0.5	1	0.67	128	255	170
Blue	0	0	1	0.67	1	0.33	170	255	85
Magenta	1	0	1	0.83	1	0.67	213	255	170
White	1	1	1	–	0	1	–	0	255
Gray	0.5	0.5	0.5	–	0	0.5	–	0	128

Figure P6.10

Problem 6.12

There are two important aspects to this problem. One is to approach it in the HSI space and the other is to use polar coordinates to create a hue image whose values grow as a function of angle. The center of the image is the middle of whatever image area is used. Then, for example, the values of the hue image along a radius when the angle is 0° would be all 0's. Then the angle is incremented by, say, one degree, and all the values along that radius would be 1's, and so on. Values of the saturation image decrease linearly in all radial directions from the origin. The intensity image is just a specified constant. With these basics in mind it is not difficult to write a program that generates the desired result.

Problem 6.13

The hue image is shown in Fig. P6.13(a).

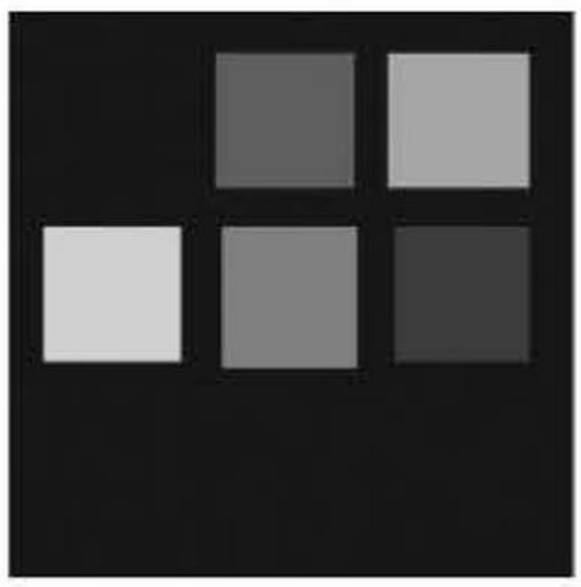


Figure P6.13(a)

Problem 6.14

(a) It is given that the colors in Fig. 6.14(a) are primary spectrum colors. It also is given that the gray-level images in the problem statement are 8-bit images. The latter condition means that hue (angle) can only be divided into a maximum number of 256 values. Because hue values are represented in the interval from 0° to 360°, this means that for an 8-bit image the increments between contiguous hue values are now 360/255.

Another way of looking at this is that the entire [0, 360] hue scale is compressed to the range [0, 255]. Thus, for example, yellow (the first primary color we encounter), which is 60° now becomes 43 (the closest integer) in the integer scale of the 8-bit image shown in the problem statement. Similarly, green, which is 120°, becomes 85 in this image. From this we easily compute the values of the other two regions as being 170 and 213. The region in the middle is pure white [equal proportions of red green and blue in Fig. 6.14(a)] so its hue by definition is 0. This also is true of the black background.

Problem 6.15

The $L^*a^*b^*$ components are computed using Eqs. (6-31) through (6-34). Reference white is $R = G = B = 1$. The computations are best done in a spreadsheet, as shown in Table P6.15.

Table P6.15

Color	R	G	B	X	Y	Z	$\frac{X}{X_w}$	$\frac{Y}{Y_w}$	$\frac{Z}{Z_w}$	$h\left(\frac{X}{X_w}\right)$	$h\left(\frac{Y}{Y_w}\right)$	$h\left(\frac{Z}{Z_w}\right)$	L^*	a^*	b^*
Ref.	1	1	1	0.95	1.00	1.10	1	1	1	1	1	1	100	0	0
Black	0	0	0	0	0	0	0	0	0	0.14	0.14	0.14	0	0	0
Red	1	0	0	0.59	0.29	0	0.62	0.29	0	0.85	0.66	0.14	83	95	105
Yellow	1	1	0	0.77	0.90	0.07	0.81	0.90	0.06	0.93	0.96	0.40	92	-16	113
Green	0	1	0	0.18	0.61	0.07	0.19	0.61	0.06	0.57	0.85	0.40	51	-136	90
Cyan	0	1	1	0.36	0.71	1.09	0.38	0.71	1	0.73	0.89	1	68	-84	-22
Blue	0	0	1	0.18	0.11	1.02	0.19	0.11	0.94	0.58	0.47	0.98	51	53	-101
Magenta	1	0	1	0.77	0.40	1.02	0.81	0.40	0.94	0.93	0.73	0.98	92	100	-49
White	1	1	1	0.95	1.00	1.10	1	1	1	1	1	1	100	0	0
Gray	0.5	0.5	0.5	0.48	0.50	0.55	0.5	0.5	0.5	0.79	0.79	0.79	76	0	0

Problem 6.16

Let C'M'Y' be an intensity-modified version of image CMY. We can relate them to their RGB and R'G'B' counterparts as follows:

$$C = (1 - R) \qquad C' = (1 - R')$$
$$M = (1 - G) \qquad M' = (1 - G')$$
$$Y = (1 - B) \qquad Y' = (1 - B')$$

If you multiply all three components of an RGB image by a constant k, its intensity is changed—that is, the transformations $R' = kR$, $G' = kG$, and $Y' = kY$ change the intensity of an RGB image. To get the equivalent transformation for a CMY image, start with one of these equations, for example,

$$R' = kR$$

And substitute for R and R' from the conversion equations at the top of the page to get

$$1 - C' = k(1 - C)$$

Then rearrange the terms to get

$$\begin{aligned} C' &= 1-k(1-C) \\ &= 1-k+kC \\ &= kC+(1-k) \end{aligned}$$

The equations for M' and Y' follow in the same manner.

Problem 6.18

(a) Because the infrared image, which was used in place of the red component image, has very high gray-level values.

(b) The water appears as solid black (0) in the near infrared image [Fig. 6.25(d)]. Threshold the image with a threshold value slightly larger than 0. The result is shown in Fig. P6.18. It is clear that coloring all the black points in the desired shade of blue presents no difficulties.

Figure P6.18

Problem 6.19

Using Eq. (7-18), we see that the basic problem is that many different colors have the same saturation value. This was demonstrated in Problem 6.10, where pure red, yellow, green, cyan, blue, and magenta all had a saturation of 1. That is, provided that any one of the RGB components is 0, Eq. (7-18) yields a saturation.

Consider RGB colors (1, 0, 0) and (0, 0.59, 0), which respectively represent shades of red and green. The HSI triplets for these colors [per Eqs. (7-16) through (7-19)] are (0, 1, 0.33) and (0.33, 1, 0.2), respectively. Now, the complements of the beginning RGB values (see Section 6.5.2) are (0, 1, 1) and (1, 0.41, 1), respectively; the corresponding colors are cyan and magenta. Their HSI values [per Eqs. (7-16) through (7-19)] are (0.5, 1, 0.66) and (0.83, 0.48, 0.8), respectively. Thus, for the red, a starting saturation of 1 yielded

the cyan "complemented" saturation of 1, while for the green, a starting saturation of 1 yielded the magenta "complemented" saturation of 0.48. That is, the same starting saturation resulted in two different "complemented" saturations. Saturation alone does not have enough information to compute the saturation of the complemented color.

Problem 6.21

The RGB transformations for a complement [see Fig. 6.31(b)] are:

$$s_i = 1 - r_i$$

where $i = 1, 2, 3$ (for the R, G, and B components). But from the definition of the CMY space in Eq. (7-5), we know that the CMY components corresponding to r_i and s_i, which we will denote using primes, are

$$r_i' = 1 - r_i$$
$$s_i' = 1 - s_i$$

Therefore,

$$r_i = 1 - r_i'$$

and

$$s_i' = 1 - s_i = 1 - (1 - r_i) = 1 - (1 - (1 - r_i')) = 1 - r_i'$$

Problem 6.23

Based on the discussion on tone and color corrections is Section 6.5, and with reference to the color wheel in Fig. 6.30, we can decrease the proportion of yellow by (1) decreasing yellow, (2) increasing blue, (3) increasing cyan and magenta, or (4) decreasing red and green.

Problem 6.24

The simplest approach conceptually is to transform every input image to the HSI color space, perform histogram specification per the discussion in Section 3.3 on the intensity (I) component only (leaving H and S alone), and convert the resulting intensity component with the original hue and saturation components back to the starting color space.

Problem 6.25

(b) The saturation image is constant, so smoothing it will produce the same constant value.

Problem 6.26

(a) The cube is composed of six intersecting planes in RGB space. The general equation for such planes is

$$az_R + bz_G + cz_B + d = 0$$

where a, b, c, and d are parameters and the z 's are the components of any point (vector) $\mathbf{z}$ in RGB space lying on the plane. If an RGB point $\mathbf{z}$ does not lie on the plane, and its coordinates are substituted in the preceding equation, the equation will give either a positive or a negative value; it will not yield zero. We say that $\mathbf{z}$ lies on the positive or negative side of the plane, depending on whether the result is positive or negative. We can change the positive side of a plane by multiplying its coefficients (except d) by −1. Suppose that we test the point $\mathbf{a}$ given in the problem statement to see whether it is on the positive or negative side each of the six planes composing the box, and change the coefficients of any plane for which the result is negative. Then, $\mathbf{a}$ will lie on the positive side of all planes composing the bounding box. In fact all points inside the bounding box will yield positive values when their coordinates are substituted in the equations of the planes. Points outside the box will give at least one negative (or zero if it is on a plane) value. Thus, the method consists of substituting an unknown color point in the equations of all six planes. If all the results are positive, the point is inside the box; otherwise it is outside the box. A flow diagram is requested in the problem statement to make it simpler to evaluate the student's line of reasoning.

Chapter 7

Problem Solutions

Problem 7.1

(b) They are not orthonormal since

$$\|\mathbf{s}_0\| = \sqrt{\langle \mathbf{s}_0, \mathbf{s}_0 \rangle} = \sqrt{\mathbf{s}_0^T \mathbf{s}_0} = \left\{ \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \right\}^{1/2} = \sqrt{1+4+1} = \sqrt{6} \neq 0$$

and similarly $\|\mathbf{s}_1\| = \sqrt{2}$ and $\|\mathbf{s}_2\| = \sqrt{3}$.

Problem 7.3

Equation (7-16) defines a generic transform as

$$\begin{aligned} T(u) &= \sum_u f(x) r(x,u) \\ &= f(x) r(x,0) + f(x) r(x,1) + \ldots \end{aligned}$$

We can write the above equation in matrix form as

$$\begin{bmatrix} T(0) \\ T(1) \\ \vdots \\ T(N+1) \end{bmatrix} = \begin{bmatrix} r(0,0) & r(0,1) & \cdots & r(0,N-1) \\ r(1,0) & r(1,1) & & \\ \vdots & & \ddots & \\ r(N-1,u) & & & r(N-1,N-1) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix}$$

We can rewrite this as

$$\mathbf{t} = \mathbf{A}'\mathbf{f} \text{ where } \mathbf{r}_u = \begin{bmatrix} r(0,u) \\ r(1,u) \\ \vdots \\ r(N-1,u) \end{bmatrix} = \begin{bmatrix} r_{u,0} \\ r_{u,1} \\ \vdots \\ r_{u,N-1} \end{bmatrix} \text{ and } \mathbf{A}' = \begin{bmatrix} \mathbf{r}_0^T \\ \mathbf{r}_1^T \\ \vdots \\ \mathbf{r}_{N-1}^T \end{bmatrix}$$

where column vectors **f** and **t** are as defined by Eqs. (7-20) and (7-21), respectively. But Eqs. (7-24) through (7-26) tell us that

$$\mathbf{t} = \mathbf{A}\mathbf{f} = \begin{bmatrix} T(0) \\ T(1) \\ \vdots \\ T(N+1) \end{bmatrix} = \begin{bmatrix} s(0,0) & s(0,1) & \cdots & s(0,N-1) \\ s(1,0) & s(1,1) & & \\ \vdots & & \ddots & \\ s(N-1,u) & & & s(N-1,N-1) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix}$$

Therefore, $\mathbf{A} = \mathbf{A}'$, $\mathbf{r}_u = \mathbf{s}_u$, for all u, and $r(x, u) = s(x, u)$.

$$\mathbf{r} = \begin{bmatrix} r(0,u) \\ r(1,u) \\ \vdots \\ r(N-1,u) \end{bmatrix} = \begin{bmatrix} r_{u,0} \\ r_{u,1} \\ \vdots \\ r_{u,N-1} \end{bmatrix}$$

Problem 7.7

(d) The norm of $f(x) = \cos x$ on the interval $[-\pi, \pi]$ is

$$\|f(x)\| = \sqrt{\langle f(x), f(x) \rangle} = \left[\int_{-\pi}^{\pi} \cos^2 x dx \right]^{\frac{1}{2}} = \left[\frac{1}{2}x + \frac{1}{4}\sin 2x \Big|_{-\pi}^{\pi} \right]^{\frac{1}{2}} = \sqrt{\pi}$$

Problem 7.8

(c) The transform of **g** is $\mathbf{t}_2 = [6.1237 \quad 2.1213 \quad -3.4641]^T$. The transform of **f** was computed in Problem 7.1; its norm is $\sqrt{70}$. The angle between the transformed vectors **t** and $\mathbf{t}_2$ is

$$\theta = \cos^{-1}\left[\frac{[-1.633 \quad -1.4142 \quad 8.0829] \begin{bmatrix} 6.1237 \\ 2.1213 \\ -3.4641 \end{bmatrix}}{(\sqrt{70})(\sqrt{54})} \right] = \cos^{-1}[-0.666] = 132°$$

which is the same as the angle between **f** and **g** in (a).

Similarly, $\mathbf{t} - \mathbf{t}_2 = [-7.7567 \quad -3.5355 \quad 11.5470]^T$ and the distance between them is

$$d = \sqrt{[-7.7567 \quad -3.5355 \quad 11.5470] \begin{bmatrix} -7.7567 \\ -3.5355 \\ 11.5470 \end{bmatrix}} = 14.3527$$

which is the same as the distance between **f** and **g** in (b).

Problem 7.11

(a) The basis is orthonormal and the coefficients are computed by the vector equivalent of Eq. (7-7)

$$\alpha_0 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \frac{5\sqrt{2}}{2}$$

$$\alpha_1 = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \frac{\sqrt{2}}{2}$$

so

$$\frac{5\sqrt{2}}{2}\mathbf{s}_0 + \frac{\sqrt{2}}{2}\mathbf{s}_1 = \frac{5\sqrt{2}}{2}\begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{bmatrix} + \frac{\sqrt{2}}{2}\begin{bmatrix}1/\sqrt{2}\\-1/\sqrt{2}\end{bmatrix} = \begin{bmatrix}3\\2\end{bmatrix}.$$

Problem 7.14

(a) and **(b)** The transform is

$$\mathbf{T} = \mathbf{AFA}^T = \begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\\0.5 & 0.5 & -0.5 & -0.5\\0.5 & -0.5 & 0.5 & -0.5\\0.5 & -0.5 & -0.5 & 0.5\end{bmatrix}\begin{bmatrix}4 & -4 & 4 & 0.5\\-3 & 1 & 5 & -0.5\\2 & -4 & 8 & -0.5\\1 & -3 & 3 & -1.5\end{bmatrix}\begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\\0.5 & 0.5 & -0.5 & -0.5\\0.5 & -0.5 & 0.5 & -0.5\\0.5 & -0.5 & -0.5 & 0.5\end{bmatrix}^T$$

Note that the first column of F (highlighted in blue) is vector **f** of Problem 7.11, the vector whose 1-D transform $\mathbf{t} = [2 \quad -1 \quad 4 \quad 3]^T$. The leftmost multiplication of $\mathbf{AFA}^T$ is

$$\mathbf{AF} = \begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\\0.5 & 0.5 & -0.5 & -0.5\\0.5 & -0.5 & 0.5 & -0.5\\0.5 & -0.5 & -0.5 & 0.5\end{bmatrix}\begin{bmatrix}4 & -4 & 4 & 0.5\\-3 & 1 & 5 & -0.5\\2 & -4 & 8 & -0.5\\1 & -3 & 3 & -1.5\end{bmatrix} = \begin{bmatrix}2 & -5 & 10 & -1\\-1 & 2 & -1 & 1\\4 & -3 & 2 & 1\\3 & -2 & -3 & 0\end{bmatrix}$$

and column 1 of the result is $[2 \quad -1 \quad 4 \quad 3]^T$. Pre-multiplication of **F** by **A** generated 1-D transforms of the columns of **F**. In a similar manner, post-multiplication of intermediate result **AF** by $\mathbf{A}^T$ generates 1-D transforms of the rows of **AF**. This is clear in the following equation, where row 3 of **AF**, which happens to be $\mathbf{f}^T = [4 \quad -3 \quad 2 \quad 1]$, becomes $[2 \quad -1 \quad 4 \quad 3]$ in row 3 of $\mathbf{AFA}^T$:

$$\mathbf{T} = (\mathbf{AF})\mathbf{A}^T = \begin{bmatrix}2 & -5 & 10 & -1\\-1 & 2 & -1 & 1\\4 & -3 & 2 & 1\\3 & -2 & -3 & 0\end{bmatrix}\begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\\0.5 & 0.5 & -0.5 & -0.5\\0.5 & -0.5 & 0.5 & -0.5\\0.5 & -0.5 & -0.5 & 0.5\end{bmatrix} = \begin{bmatrix}3 & -6 & 9 & -2\\0.5 & 0.5 & -2.5 & -0.5\\2 & -1 & 4 & 3\\-1 & 2 & 1 & 4\end{bmatrix}$$

Thus, the pre- and post-multiplication sequence demonstrated above computes the 1-D column transforms of **F** followed by the 1-D row transforms of the column transforms.

$$\mathbf{F} = \mathbf{A}^T\mathbf{T}\mathbf{A} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}^T \begin{bmatrix} 3 & -6 & 9 & -2 \\ 0.5 & 0.5 & -2.5 & -0.5 \\ 2 & -1 & 4 & 3 \\ -1 & 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}$$

$$= \begin{bmatrix} 2.25 & -2.25 & 5.75 & 2.25 \\ 1.25 & -3.25 & 0.75 & -4.75 \\ 2.75 & -4.75 & 7.25 & -1.25 \\ -0.25 & -1.75 & 4.25 & -0.25 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & -4 & 4 & 0.5 \\ -3 & 1 & 5 & -0.5 \\ 2 & -4 & 8 & -0.5 \\ 1 & -3 & 3 & -1.5 \end{bmatrix}$$

Problem 7.17

$\mathbf{T} = \mathbf{A}_M\mathbf{F}\mathbf{A}_N^T$ and $\mathbf{F} = \mathbf{A}_M^{*T}\mathbf{T}\mathbf{A}_N^*$

Problem 7.20

To verify the orthonormality of DFT expansion functions

$$s(x,u) = \frac{1}{\sqrt{N}} e^{j2\pi ux/N} \quad \text{for u} = 0,1,\ldots,N-1$$

we first compute their norms:

$$\begin{aligned} \|s(x,u)\| &= \sqrt{\left\langle \frac{1}{\sqrt{N}} e^{j2\pi ux/N}, \frac{1}{\sqrt{N}} e^{j2\pi ux/N} \right\rangle} \\ &= \frac{1}{\sqrt{N}} \sqrt{\sum_{x=0}^{N-1} \frac{1}{\sqrt{N}} e^{-j2\pi ux/N} \frac{1}{\sqrt{N}} e^{j2\pi ux/N}} \\ &= \frac{1}{\sqrt{N}} \sqrt{\sum_{x=0}^{N-1} \frac{1}{N}} \\ &= 1 \end{aligned}$$

Since the norms are 1, we next verify that they are pairwise orthogonal for all $m \neq n$. We begin with the inner product

$$\begin{aligned} \langle s(x,m), s(x,n) \rangle &= \left\langle \frac{1}{\sqrt{N}} e^{j2\pi km/N}, \frac{1}{\sqrt{N}} e^{j2\pi kn/N} \right\rangle \\ &= \frac{1}{N} \sum_{k=0}^{N-1} e^{-j2\pi km/N} e^{j2\pi kn/N} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} e^{j2\pi k(n-m)/N} \end{aligned}$$

and let $\beta = e^{j2\pi(n-m)/M}$ so that

$$\begin{aligned}\langle s(x,m),s(x,n)\rangle &= \frac{1}{N}\sum_{k=0}^{N-1}\beta^k \\ &= \frac{1}{N}\left(\frac{1-\beta^N}{1-\beta}\right) \\ &= 0\end{aligned}$$

where the last two steps follow from (1) the fact that the sum of the first p terms of infinite geometric series $a+ar+ar^2+\ldots+ar^{p-1}$ is $a(1-r^p)/(1-r)$ and (2) recognizing that $\beta^N = e^{j2\pi(n-m)} = 1$ while $\beta \neq 1$ for all integer $m \neq n$. Thus, the sampled complex exponential expansion functions defined by Eq. (7-37) are orthonormal.

Problem 7.24

The Fourier transform of $g(at)$ for $a>0$ is

$$\Im\{g(at)\} = \int_{-\infty}^{\infty} g(at)e^{-j2\pi ft}dt$$

Let $\tau = at$ so that $t = \tau/a$ and $d\tau = adt$ and $dt = d\tau/a$. Substitution into the above equation then gives

$$\begin{aligned}\Im\{g(\tau)\} &= \int_{-\infty}^{\infty} g(\tau)e^{-j2\pi f\frac{\tau}{a}}\frac{d\tau}{a} \\ &= \frac{1}{a}\int_{-\infty}^{\infty} g(\tau)e^{-j2\pi\left(\frac{f}{a}\right)\tau}d\tau\end{aligned}$$

Since

$$\Im\{g(t)\} = \int_{-\infty}^{\infty} g(t)e^{-j2\pi ft}dt = G(f)$$

we can write

$$\Im\{g(\tau)\} = \Im\{g(at)\} = \frac{1}{a}G\left(\frac{f}{a}\right)$$

Letting $a = 2^s$, $g = \psi$, and $\Psi(f) = \Im\{\psi(t)\}$, we finally get

$$\Im\left\{\psi(2^s t)\right\} = \frac{1}{|2^s|}\Psi\left(\frac{f}{2^s}\right)$$

For $s>0$, time is compressed and the spectrum is expanded; for $s<0$, the opposite occurs.

Problem 7.35

Start with the multiresolution refinement equation [i.e., Eq. (7-125)] with variable k replaced by n,

$$\varphi(x) = \sum_n h_\varphi(k)\sqrt{2}\varphi(2x-n)$$

Scaling x by 2^j, translating it by k, and letting $m = 2k+n$ gives

$$\begin{aligned}\varphi(2^j x - k) &= \sum_n h_\varphi(n)\sqrt{2}\varphi\left(2(2^j x - k) - n\right) \\ &= \sum_n h_\varphi(n)\sqrt{2}\varphi\left(2^{j+1}x - 2k - n\right) \\ &= \sum_m h_\varphi(m - 2k)\sqrt{2}\varphi\left(2^{j+1}x - m\right)\end{aligned}$$

Note that scaling coefficients h_φ can be thought of as "weights" used to expand $\varphi(2^j x - k)$ as a sum of scale $j + 1$ scaling functions. A similar sequence of operations—beginning with Eq. (7-130)—provides an analogous result for $\psi(2^j x - k)$. That is,

$$\psi(2^j x - k) = \sum_m h_\psi(m - 2k)\sqrt{2}\varphi\left(2^{j+1}x - m\right)$$

Substituting Eq. (7-127) into Eq. (7-135), we get

$$d_j(k) = \int f(x)2^{j/2}\psi(2^j x - k)dx$$

which, upon replacing $\psi(2^j x - k)$ in this equation with the right side of the previous equation, becomes

$$d_j(k) = \int f(x)2^{j/2}\left[\sum_m h_\psi(m - 2k)\sqrt{2}\varphi(2^{j+1}x - m)\right]dx$$

Interchanging the sum and integral and rearranging terms then gives

$$d_j(k) = \sum_m h_\psi(m - 2k)\int f(x)2^{(j+1)/2}\varphi(2^{j+1}x - m)dx$$

Where the bracketed quantity is $c_{j_0}(k)$ of Eq. (7-134) with $j_0 = j + 1$ and $k = m$. To see this, substitute Eq. (7-121) into Eq. (7-134) and replace j_0 and k with $j + 1$ and m, respectively. Therefore, we can write

$$d_j(k) = \sum_m h_\psi(m - 2k)c_{j+1}(m)$$

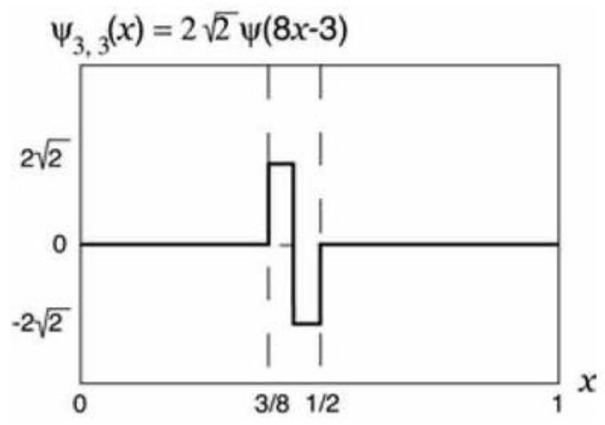
and note the detail coefficients at scale j are a function of the approximation coefficients at scale $j + 1$.

Problem 7.37

From Eq. (7-127), we find that

$$\psi_{3,3}(x) = 2^{3/2}\psi\left(2^3 x - 3\right) = 2\sqrt{2}\psi(8x - 3)$$

and using the Haar wavelet function definition from Eq. (7-132), obtain the following plot:



To express $\psi_{3,3}(x)$ as a function of scaling functions, we employ Eq. (7-130) and the Haar wavelet vector defined in Example 7.6—that is, $h_\psi(0) = 1/\sqrt{2}$ and $h_\psi(1) = -1/\sqrt{2}$. Thus we get

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\varphi(2x-n)$$

so that

$$\psi(8x-3) = \sum_n h_\psi(n)\sqrt{2}\varphi(2[8x-3]-n)$$

$$= \frac{1}{\sqrt{2}}\sqrt{2}\varphi(16x-6) + \left(\frac{-1}{\sqrt{2}}\right)\sqrt{2}\varphi(16x-7)$$

$$= \varphi(16x-6) - \varphi(16x-7)$$

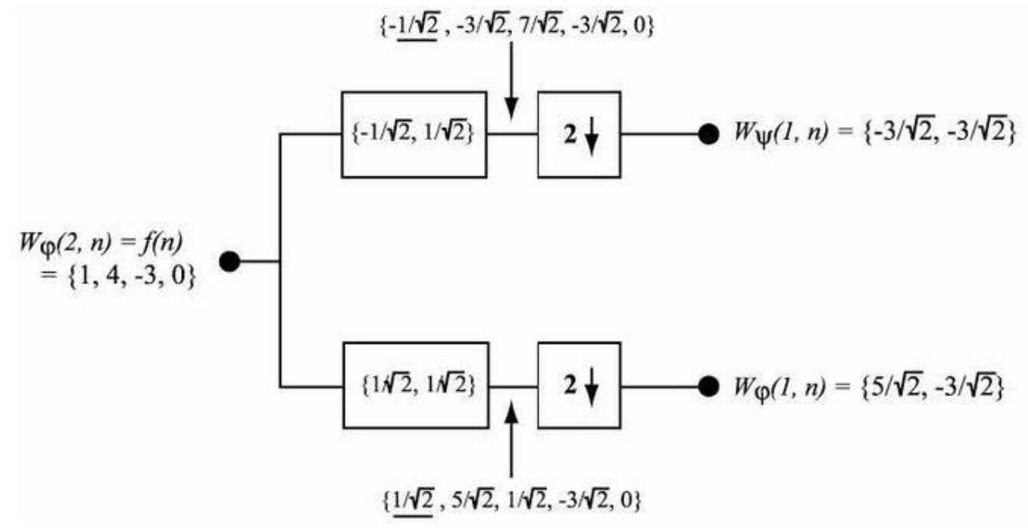
Then, since $\psi_{3,3}(x) = 2\sqrt{2}\psi(8x-3)$ from above, substitution gives

$$\psi_{3,3}(x) = 2\sqrt{2}\psi(8x-3)$$

$$= 2\sqrt{2}\varphi(16x-6) - 2\sqrt{2}\varphi(16x-7)$$

Problem 7.41

The filter bank is the first bank in Fig. 7.23:



Problem 7.43

(a) Input $\varphi(k) = \{1,1,1,1,1,1,1,1\} = \varphi_{0,0}(k)$ for a three-scale wavelet transform with Haar scaling and wavelet functions. Since wavelet transform coefficients measure the similarity of the input to the basis functions, the resulting transform is

$$\{T_\varphi(0,0), T_\psi(0,0), T_\psi(1,0), T_\psi(1,1), T_\psi(2,0), T_\psi(2,1), T_\psi(2,2), T_\psi(2,3)\}$$

$$= \{2\sqrt{2}, 0, 0, 0, 0, 0, 0, 0\}.$$

The $T_\varphi(0,0)$ term can be computed using Eq. (7-137) with $j_0 = k = 0$.

Problem 7.45

As can be seen in the sequence of images that are shown, the DWT is not shift invariant. If the input is shifted, the transform changes. Since all original images in the problem are 128×128, they become the $T_\varphi(7, j, k)$ inputs for the FWT computation process. The filter bank of Fig. 7.22(a) can be used with $j + 1 = 7$. For a single scale transform, transform coefficients $T_\varphi(6, j, k)$ and $T_\psi^i(6, j, nk)$ for $i = \mathrm{H}, \mathrm{V}, \mathrm{D}$ are generated. With Haar wavelets, the transformation process subdivides the image into non-overlapping 2×2 blocks and computes 2-point averages and differences (per the scaling and wavelet vectors). Thus, there are no horizontal, vertical, or diagonal detail coefficients in the first two transforms shown; the input images are constant in all 2×2 blocks (so all differences are 0). If the original image is shifted by 1 pixel, detail coefficients are generated since there are then 2×2 areas that are not constant. This is the case in the third transform shown.

Chapter 8

Problem Solutions

Problem 8.4

(a) Table P8.4 shows the starting intensity values, their 8-bit codes, the IGS sum used in each step, the 4-bit IGS code and its equivalent decoded value (the decimal equivalent of the IGS code multiplied by 16), the error between the decoded IGS intensities and the input values, and the squared error.

(b) Using Eq. (8-10) and the squared error values from Table P8.4, the rms error is

$$e_{rms} = \sqrt{\frac{1}{8}(144 + 25 + 49 + 16 + 16 + 169 + 64 + 9)}$$
$$= \sqrt{\frac{1}{8}(492)} = 7.84$$

or about 7.8 intensity levels. From Eq. (8-11), the signal-to-noise ratio is

$$SNR_{ms} = \frac{96^2 + 144^2 + 128^2 + 240^2 + 176^2 + 160^2 + 64^2 + 96^2}{492}$$
$$= \frac{173824}{492} \simeq 352$$

Table P8.4

Intensity	8-bit Code	Sum	IGS Code	Decoded IGS	Error	Square Error
		00000000				
108	01101100	01101100	0110	96	-12	144
139	10001011	10010111	1001	144	5	25
135	10000111	10001110	1000	128	-7	49
244	11110100	11110100	1111	240	-4	16
172	10101100	10110000	1011	176	4	16
173	10101101	10101101	1010	160	-13	169
56	00111000	01000101	0100	64	8	64
99	01100011	01101000	0110	96	-3	9

Problem 8.6

The conversion factors are computed using the logarithmic relationship

$$\log_a x = \frac{1}{\log_b a} \log_b x$$

Thus, 1 Hartley = 3.3219 bits and 1 nat = 1.4427 bits.

Problem 8.7

Let the set of source symbols be $\{a_1, a_2, \ldots, a_q\}$ with probabilities $\left[P(a_1), P(a2), \ldots, P(a_q)\right]^T$. Then, using Eq. (8-6) and the fact that the sum of all $P(a_i)$ is 1, we get

$$\begin{aligned}\log q - H &= \log q\left[\sum_{j=1}^{q} P(a_j)\right] + \sum_{j=1}^{q} P(a_j)\log P(a_j) \\ &= \sum_{j=1}^{q} P(a_j)\log q + \sum_{j=1}^{q} P(a_j)\log P(a_j) \\ &= \sum_{j=1}^{q} P(a_j)\log qP(a_j)\end{aligned}$$

Using the log relationship from Problem 8.6, this becomes

$$= \log_e \sum_{j=1}^{q} P(a_j)\ln qP(a_j)$$

Then, multiplying the inequality $\ln x \le x - 1$ by -1 to get $\ln 1/x \ge 1 - x$ and applying it to this last result,

$$\begin{aligned}\log q - H &\ge \log e \sum_{j=1}^{q} P(a_j)\left[1 - \frac{1}{qP(a_j)}\right] \\ &\ge \log e\left[\sum_{j=1}^{q} P(a_j) - \frac{1}{q}\sum_{j=1}^{q}\frac{P(a_j)}{P(a_j)}\right] \\ &\ge \log e[1-1] \\ &\ge 0\end{aligned}$$

so that

$$\log q \ge H$$

Therefore, H is always less than, or equal to, $\log q$. Furthermore, in view of the equality condition ($x = 1$) for $\ln 1/x \ge 1 - x$, which was introduced at only one point in the above derivation, we will have strict equality if and only if $P(a_j) = 1/q$ for all j.

Problem 8.9

(d) We can compute the relative frequency of pairs of pixels by assuming that the image is connected from line to line and end to beginning. The resulting probabilities are listed in Table P8.9-2.

Table P8.9-2

Intensity pair	Count	Probability
(21, 21)	8	1/4

(21, 95)	4	1/8
(95, 169)	4	1/8
(169, 243)	4	1/8
(243, 243)	8	1/4
(243, 21)	4	1/8

The entropy of the intensity pairs is estimated using Eq. (8-7) and dividing by 2 (because the pixels are considered in pairs):

$$\frac{1}{2}\tilde{H} = -\frac{1}{2}\left[\frac{1}{4}\log_2\frac{1}{4} + \frac{1}{8}\log_2\frac{1}{8} + \frac{1}{8}\log_2\frac{1}{8} + \frac{1}{8}\log_2\frac{1}{8} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{8}\log_2\frac{1}{8}\right]$$
$$= \frac{2.5}{2}$$
$$= 1.25 \text{ bits/pixel}$$

The difference between this value and the entropy in (a) tells us that a mapping can be created to eliminate (1.811_1.25) =0.56 bits/pixel of spatial redundancy.

Problem 8.15

To decode $G_{\text{exp}}^{k}(n)$:

1. Count the number of 1s in a left-to-right scan of a concatenated $G_{\text{exp}}^{k}(n)$ bit sequence before reaching the first 0, and let i be the number of 1s counted.
2. Get the $k+i$ bits following the 0 identified in step 1 and let d be its decimal equivalent.
3. The decoded integer is then

$$d + \sum_{j=0}^{i-1} 2^{j+k}$$

For example, to decode the first G2 exp (n) code in the bit stream 10111011..., let i = 1, the number of 1s in a left-to-right scan of the bit stream before finding the first 0. Get the 2+1 = 3 bits following the 0, that is, 111 so d = 7. The decoded integer is then

$$7 + \sum_{j=0}^{1-1} 2^{j+2} = 7 + 2^2 = 11$$

Repeat the process for the next code word, which begins with the bit sequence 011...

Problem 8.18

The arithmetic decoding process is the reverse of the encoding procedure. Start by dividing the [0, 1) interval according to the symbol probabilities. This is shown in Table P8.18. The decoder immediately knows the message 0.23355 begins with an "e", since the coded message lies in the interval [0.2, 0.5). This makes it clear that the second symbol is an "a", which narrows the interval to [0.2, 0.26). To further see this, divide the interval [0.2, 0.5) according to the symbol probabilities. Proceeding like this, which is the same procedure used to code the message, we get "eaii!".

Table P8.18

Symbol	Probability	Range
a	0.2	[0.0, 0.2)
e	0.3	[0.2, 0.5)
i	0.1	[0.5, 0.6)
o	0.2	[0.6, 0.8)
u	0.1	[0.8, 0.9)
!	0.1	[0.9, 1.0)

Problem 8.20

The input to the LZW decoding algorithm in Example 8.7 is

39 39 126 126 256 258 260 259 257 126

The starting dictionary, to be consistent with the coding itself, contains 512 locations–with the first 256 corresponding to intensity values 0 through 255. The decoding algorithm begins by getting the first encoded value, outputting the corresponding value from the dictionary, and setting the "recognized sequence" to the first value. For each additional encoded value, we (1) output the dictionary entry for the pixel value(s), (2) add a new dictionary entry whose content is the "recognized sequence" plus the first element of the encoded value being processed, and (3) set the "recognized sequence" to the encoded value being processed. For the encoded output in Example 8.7, the sequence of operations is as shown in Table P8.20.

Note, for example, in row 5 of the table that the new dictionary entry for location 259 is 126-39, the concatenation of the currently recognized sequence, 126, and the first element of the encoded value being processed–the 39 from the 39-39 entry in dictionary location 256. The output is then read from the third column of the table to yield

39 39 126 126

39 39 126 126

39 39 126 126

39 39 126 126

where it is assumed that the decoder knows or is given the size of the image that was received. Note that the dictionary is generated as the decoding is carried out.

Table P8.20

Recognized	Encoded Value	Pixels	Dict. Address	Dict. Entry
	39	39		
39	39	39	256	39-39
39	126	126	257	39-126
126	126	126	258	126-126
126	256	39-39	259	126-39
256	258	126-126	260	39-39-126
258	260	39-39-126	261	126-126-39
260	259	126-39	262	39-39-126-126
259	257	39-126	263	126-39-39
257	126	126	264	39-126-126

Problem 8.24

(a) - (b) Following the procedure outlined in Section 8.2 (Block Transform Coding), we obtain the results shown in Table P8.24.

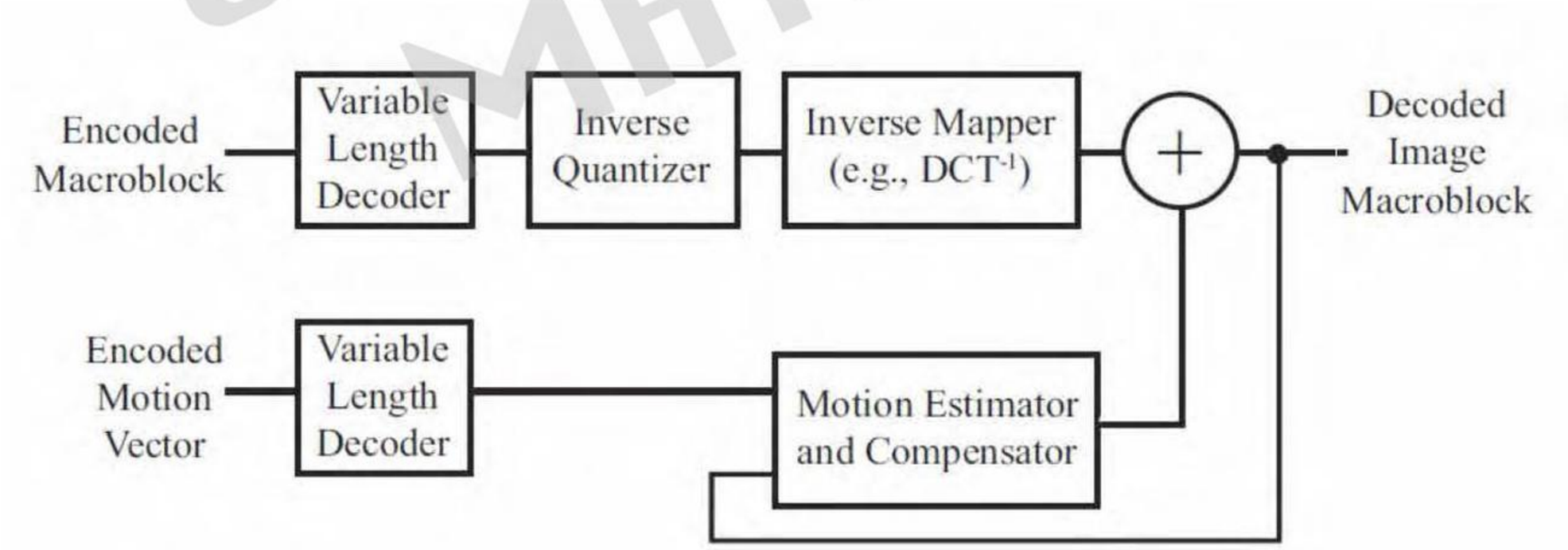
Table P8.24

DC Coefficient Difference	Two's Complement Value	Code
-7	1...1001	00000
-6	1...1010	00001
-5	1...1011	00010
-4	1...1100	00011
4	0...0100	00100
5	0...0101	00101
6	0...0110	00110
7	0...0111	00111

Problem 8.27

The appropriate MPEG decoder is shown in Fig. P8.27.

Figure P8.27



Problem 8.29

The derivation proceeds by substituting the uniform probability function into Eqs. (8-56) - (8-58) and solving the resulting simultaneous equations with $L = 4$. Equation (8-57) yields

$$s_0 = 0$$
$$s_1 = \frac{1}{2}(t_1 + t_2)$$
$$s_2 = \infty$$

Substituting these values into the integrals defined by Eq. (8-56), we get two equations. The first is (assuming $s_1 \leq A$)

$$\int_{s_0}^{s_1}(s-t_1)p(s)ds=0$$

$$\frac{1}{2A}\int_0^{\frac{1}{2}(t_1+t_2)}(s-t_1)ds=\frac{s^2}{2}-t_1s\Big|_0^{\frac{1}{2}(t_1+t_2)}=0$$

$$\begin{aligned}(t_1+t_2)^2-4t(t_1+t_2)&=0\\(t_1+t_2)(t_2-3t_1)&=0\end{aligned}$$

so

$$\begin{aligned}t_1&=-t_2\\t_2&=3t_1\end{aligned}$$

The first of these relations does not make sense since both t_1 and t_2 must be positive. The second relationship is a valid one. The second integral yields (noting that s_1 is less than A so the integral from A to ∞ is 0 by the definition of $p(s)$)

$$\int_{s_1}^{s_2}(s-t_2)p(s)ds=0$$

$$\frac{1}{2A}\int_{\frac{1}{2}(t_1+t_2)}^{A}(s-t_2)ds=\frac{s^2}{2}-t_2s\Big|_{\frac{1}{2}(t_1+t_2)}^{A}=0$$

$$4A^2-8At_2-(t_1+t_2)^2-4t_2(t_1+t_2)=0$$

Substituting $t_2=3t_1$ from the first integral simplification into this result, we get

$$\begin{aligned}8t_1-6At_1+A^2&=0\\\left[t_1-\frac{A}{2}\right](8t_1-2A)&=0\\t_1&=\frac{A}{2}\\t_1&=\frac{A}{4}\end{aligned}$$

Back substituting these values of t_1, we find the corresponding t_2 and s_1 values:

$$t_2=\frac{3A}{2}\text{ and }s_1=A\text{ for }t_1=\frac{A}{2}$$

$$t_2=\frac{3A}{4}\text{ and }s_1=\frac{A}{2}\text{ for }t_1=\frac{A}{4}$$

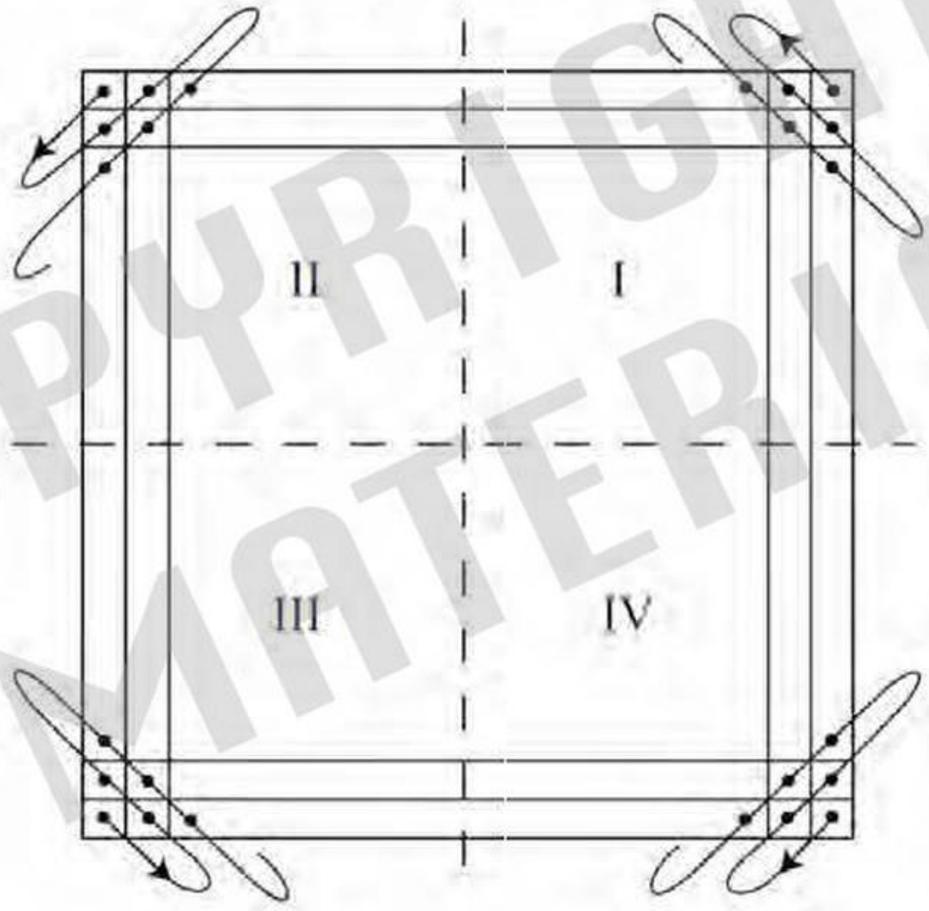
Because $s_1=A$ is not a real solution (the second integral equation would then be evaluated from A to A, yielding 0 or no equation), the solution is given by the second. That is,

$$s_0 = 0 \;\; s_1 = \frac{A}{2} \;\; s_2 = \infty$$
$$t_1 = \frac{A}{4} \;\; t_2 = \frac{3A}{4}$$

Problem 8.34

A variety of methods for inserting invisible watermarks into the DFT coefficients of an image have been reported in the literature. Here is a simplified outline of one in which watermark insertion is done as follows:

Figure P8.34



1. Create a watermark by generating a P-element pseudo-random sequence of numbers, $w_1, w_2, \ldots, w_p$, taken from a Gaussian distribution with zero mean and unit variance.

2. Compute the DFT of the image to be watermarked. We assume that the transform has not been centered by pre-multiplying the image by $(-1)^{x+y}$.

3. Choose $P/2$ coefficients from each of the four quadrants of the DFT in the middle frequency range. This is easily accomplished by choosing coefficients in the order shown in Fig. P8.34 and skipping the first K coefficients (the low frequency coefficients) in each quadrant.

4. Insert the first half of the watermark into the chosen DFT coefficients, c_i for $1 \le i \le \frac{P}{2}$, in quadrants I and III of the DFT using

$$c_i' = c_i(1 + \alpha w_i)$$

5. Insert the second half of the watermark into the chosen DFT coefficients of quadrants II and IV of the DFT in a similar manner. Note that this process maintains the symmetry of the transform of a real-valued image. In addition, constant α determines the strength of the inserted watermark.

6. Compute the inverse DFT with the watermarked coefficients replacing the unmarked coefficients.

Watermark extraction is performed as follows:

1. Locate the DFT coefficients containing the watermark by following the insertion process in the embedding algorithm.
2. Compute the watermark $\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_P$ using

$$\hat{w}_i = c_i' - c_i$$

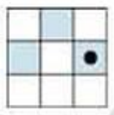
3. Compute the correlation between w and $\hat{w}$ and compare to a pre-determined threshold T to determine if the mark is present.

Chapter 9

Problem Solutions

Problem 9.1

(a)



Problem 9.2

[Note: We recommend that you solve this problem by hand to gain experience. However, if you use **imerode** (MATLAB Image Processing Toolbox) to solve the problem or check your hand-solution, be sure to construct your structuring element using `B = strel('arbitrary', SE)`, where SE is one of the structuring elements in the problem statement. Failure to do this can lead to unexpected results in some cases.]

(a)

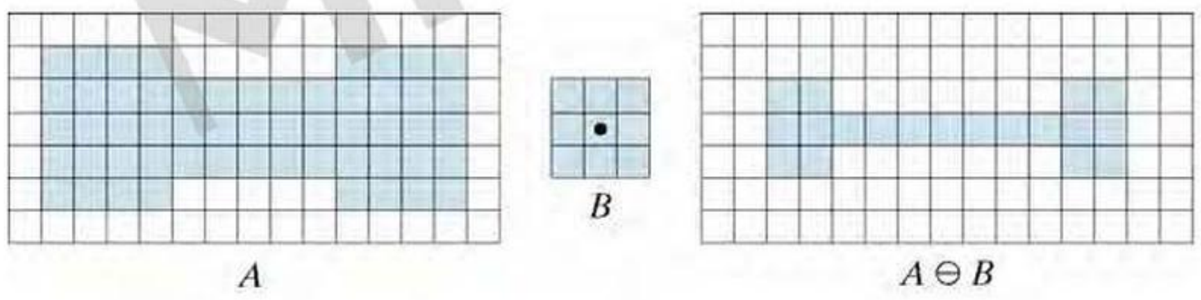


Figure P9.2(a)

Problem 9.3

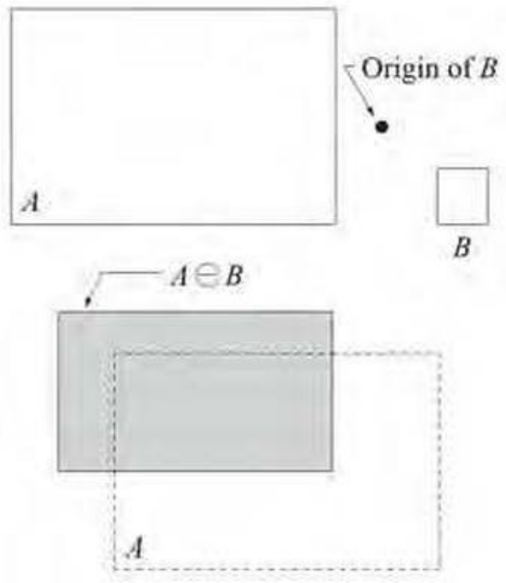


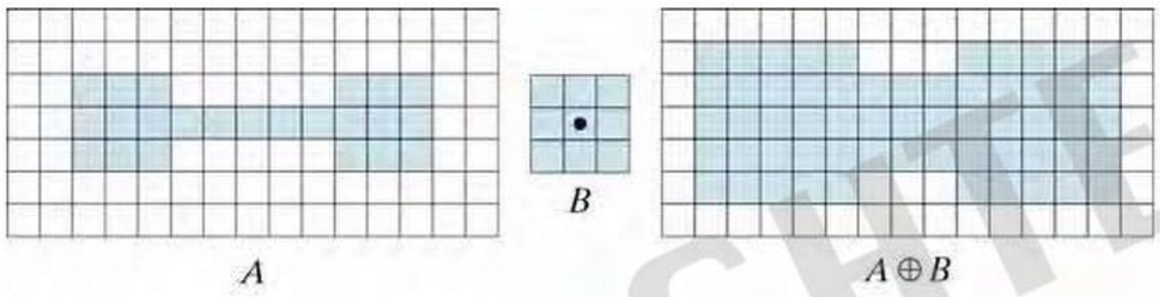
Figure P9.3

Problem 9.4

(a) Erosion is the set of point, $\{z\}$, such that B, translated by z, is contained in A. If B is a single point, this definition will be satisfied only by the points in A. Therefore, erosion of A by B is simply A.

Problem 9.6

(a) Keep in mind in all parts of the problem that dilation is with respect to the shaded elements of B.



Problem 9.8

In the following, the origin of the structuring elements is shown as a black dot.

(a) Erode the original set (shown dashed) with the structuring element shown (note that the origin is at the bottom, right).

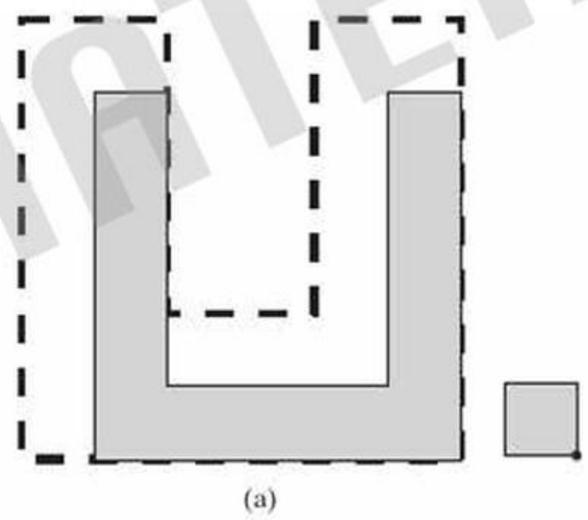


Figure P9.8(a)

(c) First erode the set down to two vertical lines using the rectangular structuring element (note that this element is slightly taller than the center section of the "U" figure). Then dilate the result with the circular structuring element.

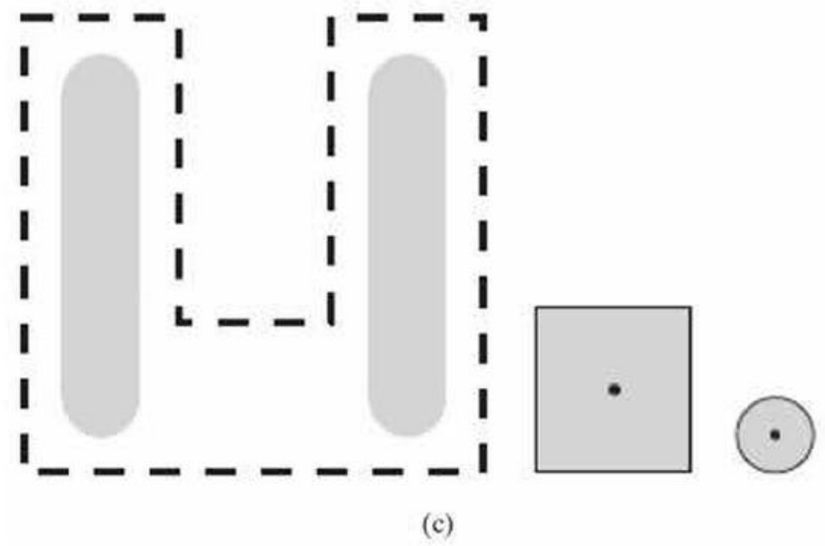
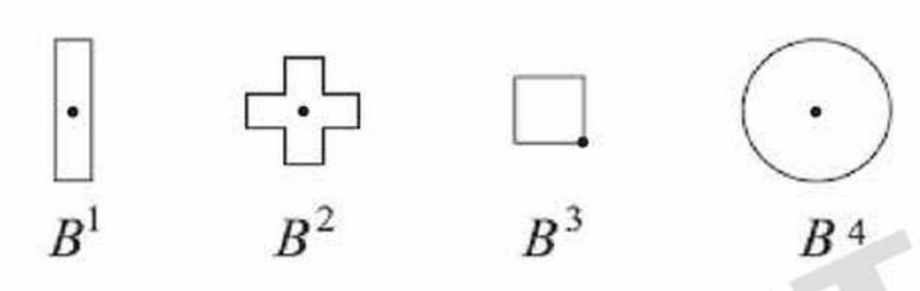


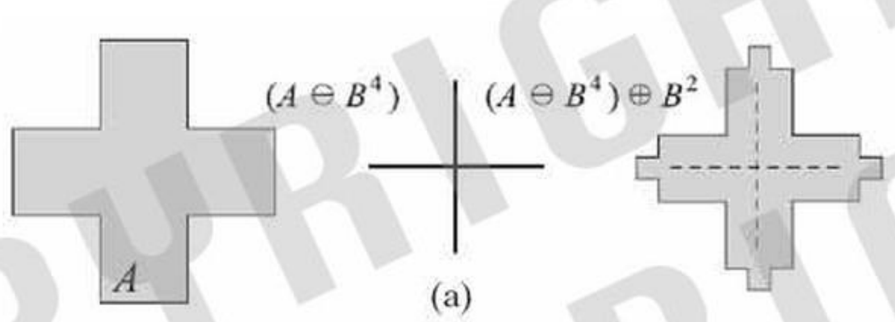
Figure P9.8(c)

Problem 9.9

Structuring elements for all parts of this problem (the dots indicate the origin):



(a)



(a)

Problem 9.10

(a) The dilated set will grow without bound.

Problem 9.12

We have to show that

$$A \ominus B = \left\{ w \in Z^2 \middle| w + b \in A \text{ for every } b \in B \right\} = \left\{ w \in Z^2 \middle| (B)_w \subseteq A \right\}$$

The right side follows directly from the definition of translation because the set $(B)_w$ has elements of the form $w + b$ for $b \in B$. That is, $w + b \in A$ for *every* $b \in B$ implies that $(B)_w \subseteq A$. Conversely, $(B)_w \subseteq A$ implies that *all* elements of $(B)_w$ are contained in A, or, equivalently, that $w + b \in A$ for every $b \in B$.

Problem 9.13

(b) Suppose that $w \in A \ominus B = w \in (A)_{-b}$. Then, for every $b \in B$, $w \in (A)_{-b}$, or $w + b \in A$. But we showed in Problem 9.12 that $w + b \in A$ for every $b \in B$ implies that $(B)_w \subseteq A$, so that $w \in A \ominus B = \left\{ w \in Z^2 \middle| (B)_w \subseteq A \right\}$. Similarly, $(B)_w \subseteq A$ implies that all elements of $(B)_w$ are contained in A, or, equivalently, that $w + b \in A$ for every $b \in B$. But, from (a), $w + b \in A$ implies that $w \in (A)_{-b}$. Thus, if for every $b \in B$, $w \in (A)_{-b}$, then $w \in \bigcap_{b \in B} (A)_{-b}$.

Problem 9.14

We have to show that

$$A \oplus B = \left\{ w \in Z^2 \middle| w = a + b, \text{ for some } a \in A \text{ and } b \in B \right\} = \left\{ w \in Z^2 \middle| (\hat{B})_w \cap A \neq \varnothing \right\}$$

The elements of $\left(\hat{B}\right)_w$ are of the form $w - b$ for $b \in B$. The condition $(\hat{B})_w \cap A \neq \varnothing$ implies that for some $b \in B$, $w - b \in A$, or $w - b = a$ for some $a \in A$ (note that $w = a + b$). Conversely, if $w = a + b$ for some $a \in A$ and $b \in B$, then $w - b = a$ or $w - b \in A$, which implies that $w \in \left[(\hat{B})_w \cap A \neq \varnothing \right]$.

Problem 9.15

(b) Suppose that $w \in \bigcup_{b \in B} (A)_b$. Then for some $b \in B$, $w \in (A)_b$. However, $w \in (A)_b$ implies that there exists an $a \in A$ such that $w = a + b$. But if $w = a + b$ for some $a \in A$, and $b \in B$, then $w - b = a$ or $w - b \in A$, which implies that $w \in \left[(\hat{B})_w \cap A \neq \varnothing \right]$. Now suppose that $w \in \left[(\hat{B})_w \cap A \neq \varnothing \right]$. The condition $(\hat{B})_w \cap A \neq \varnothing$ implies that for some $b \in B$, $w - b \in A$, or $w - b = a$ (i.e., $w = a + b$) for some $a \in A$. But, if $w = a + b$ for some $a \in A$ and $b \in B$, then $w \in (A)_b$ and, therefore, $w \in \bigcup_{b \in B} (A)_b$.

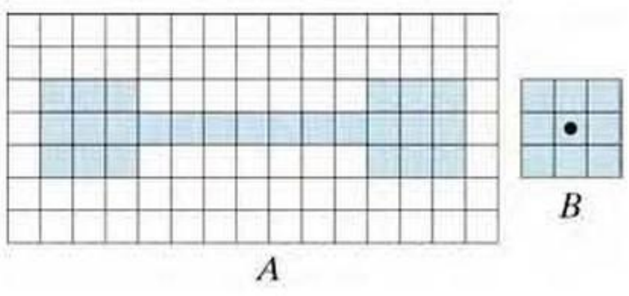
Problem 9.17

(a) When the structuring element B contains part of the boundary of set A during the opening operation, B is contained within the set because B is translated inside A. This satisfies the definition of the opening, so the black straight line segments in (d) indeed correspond to the straight line segments in (a).

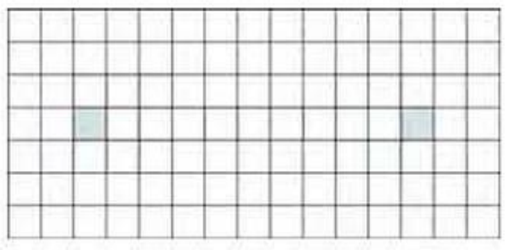
Problem 9.18

(a)

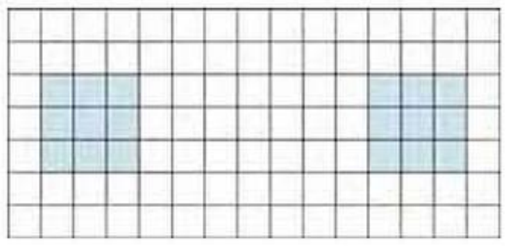
Set A and structuring element B:



Erosion of A by B:



Dilation of the Erosion (Opening)



Problem 9.19

(a) The result will be A because erosion by B will strip off a border of 1s determined by the size of B. The dilation with the same structuring element will restore that border.

Problem 9.20

(a) We start with the definition of opening in Eq. (9-10): $A \circ B = (A \ominus B) \oplus B$. Then,

$$\begin{aligned}(A \circ B)^c &= \left((A \ominus B) \oplus B\right)^c \\ &= (A \ominus B)^c \ominus \hat{B} \\ &= \left(A^c \oplus \hat{B}\right) \ominus \hat{B} \\ &= A^c \bullet \hat{B}\end{aligned}$$

The second line follows from the duality property in Eq. (9-9), the third line follows from the duality property in Eq. (9-8), and the last line follows directly from the definition of the closing given in Eq. (9-11).

Problem 9.21

(a) From Fig. 9.8 and Eq. (9-12), we know that the opening of A by B is the union of all the translations of B such that B is *contained* in A. Therefore, the opening must be a subset of A.

(b) From Eq. (9-12),

$$C \circ B = \bigcup \left\{ (B)_z \middle| (B)_z \subseteq C \right\}$$

and

$$D \circ B = \bigcup \left\{ (B)_z \middle| (B)_z \subseteq D \right\}$$

Therefore, if $C \subseteq D$, it follows that $C \circ B \subseteq D \circ B$.

Problem 9.24

The maximum width that the border can have is one pixel less than the width that would cause it to overlap any part of object C or E. The figure below shows the original B_2 and next to it the structuring element with the maximum border.

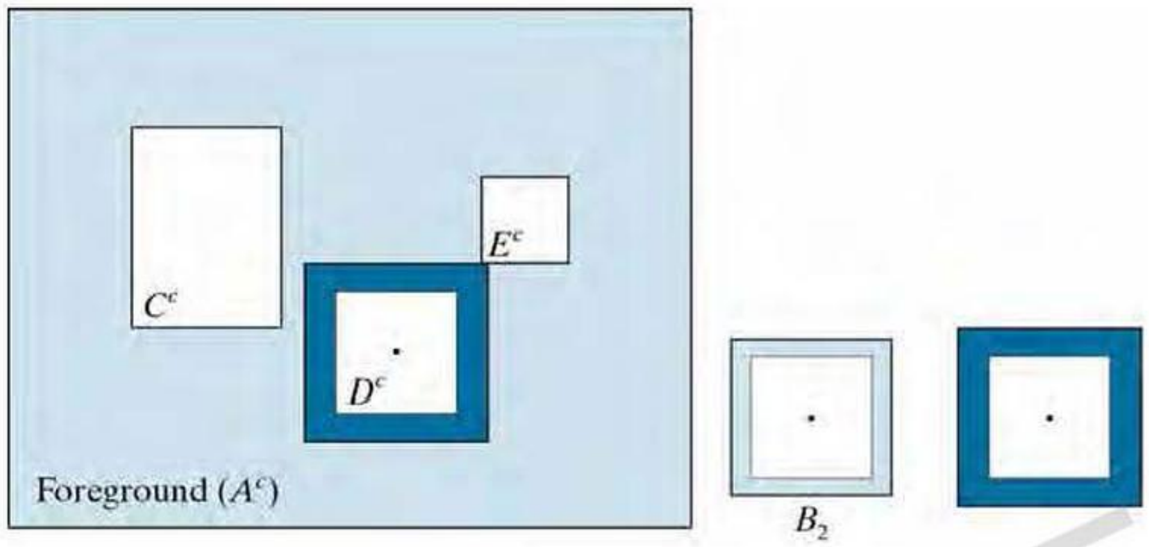


Figure P9.24

Problem 9.27

A 4-connected curve cannot have diagonal connections, so the algorithm only has to check the input image for diagonal terms from the center point of a 3×3 neighborhood to the next point in the curve. For those pixels found, add a 1 to the left (or right) of the center pixel, taking into consideration the direction of travel from the center pixel to the next pixel. Then, given that the curve is one pixel thick with no branches, we only need to check for the possibilities in the first row of the following figure (the reason for labeling the neighborhoods in the manner shown will be clear shortly):

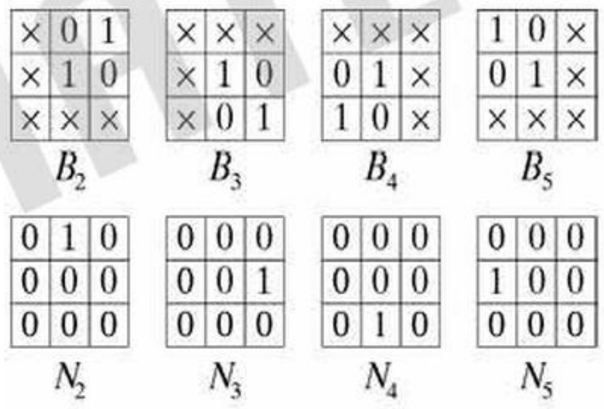


Figure P9.27

The structuring elements shown in the first row of Fig. P9.27 take into account the fact that we only need to look one pixel ahead of every pixel in the direction of travel and make sure all points are visited. Because we are looking only one pixel ahead, we have to guarantee that the pixels on each side of the diagonal are 0; the others do not matter, hence the don't care conditions.

The algorithm consists of the following principal stages.

1) Apply the hit-miss transform to the image containing the curve using each of the structuring elements in the first row of the problem figure, one at a time. Label with a 2 the hits found with B_2, label with a 3 the hits found with B_3, and so on.

2) For each point labeled 2, perform a logical OR of its 3-by-3 neighborhood with N_2 in the second row of Fig. P9.27 (this adds a 1 in the location required to convert the diagonal connection to a 4-connection). Repeat for the other 3 types of hits.

Problem 9.30

The key difference between the Lake and the other two features is that the former forms a closed contour. Assuming that the shapes are processed one at a time, a basic two-step approach for differentiating between the three shapes is as follows:

Step 1. Apply an end-point detector to the object. If no end points are found, the object is a Lake. Otherwise it is a Bay or a Line.

Step 2. There are numerous ways to differentiate between a Bay and a Line. One of the simplest is to determine a line joining the two end points of the object. If the AND of the object and this line contains only two points, the figure is a Bay. Otherwise it is a Line. There are pathological cases in which this test will fail, and additional "intelligence" needs to be built into the process, but these pathological cases become less probable with increasing resolution of the thinned figures.

Problem 9.32

(a) With reference to the example shown in Fig. P9.32, the boundary that results from using the structuring element in Fig. 9.17(c) generally forms an 8-connected path (leftmost figure), whereas the boundary resulting from the structuring element in Fig. 9.15(b) forms a 4-connected path (rightmost figure).

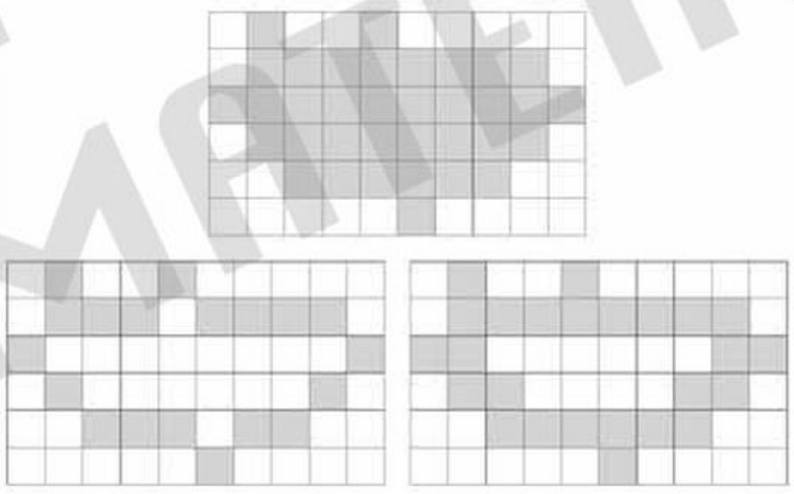


Figure P9.32

Problem 9.33

(a) The entire image would be filled with 1's.

Problem 9.36

(a) If the spheres are not allowed to touch, the solution of the problem starts by determining which points are background (black) points. To do this, we pick a black point on the boundary of the image and determine all black points connected to it using a connected component algorithm (Section 9.6). These connected components are labels with a value different from 1 or 0. The remaining black points are interior to spheres. We can fill all spheres with white by applying the hole filling algorithm in Section 9.6 until all interior black points have been turned into white points. The alert student will realize that if the interior points are already known, they can all be turned simply into white points thus filling the spheres without having to do region filling as a separate procedure.

Problem 9.37

Denote the original image by I and create an array C of zeros (the same size as I) to hold the connected components. Initialize NC, the number of connected components to 0. Create a zero array, Z, the same size as I. Find any 1-valued pixel in I and denote it by p. Set to 1 in Z the same location as p. Apply the reconstruction by dilation algorithm with Z as the marker, and I as the mask, using a 3×3 structuring element of 1s (this defines 8-connectivity over a 3×3 region). This will yield an array R, of the same size as I, containing all the 1-valued pixels in I that are connected to p. Increase NC by 1 and add $NC*R$ to C (multiplying R by NC labels all elements of the connected component just found with the value NC. Subtract R from I. Create a zero array, Z, the same size as I and repeat until there are no 1-valued elements left in R. (Keep in mind that that reconstruction by dilation is basically an implementation of Eq. (9-20) until stability.)

Problem 9.40

The dark image structures that are completely filled by morphological closing remain closed after the reconstruction.

Problem 9.41

(a) In a proof by induction, we prove the expression for $n=1$; assume that it is true for n; and prove it true for $n+1$. The proof for $n=1$ is not difficult:

$$
\begin{aligned}
E_G^{(1)}(F) &= \left[\left[E_G^{(1)}(F)\right]^c\right]^c \\
&= \left[\left[(F\ominus B)\cup G\right]^c\right]^c \\
&= \left[(F\ominus B)^c\cap G^c\right]^c \\
&= \left[(F^c\oplus \hat{B})\cap G^c\right]^c \\
&= \left[(F^c\oplus B)\cap G^c\right]^c \\
&= \left[D_{G^c}^{(1)}(F^c)\right]^c
\end{aligned}
$$

where the first line follows from the fact that the complement of the complement of an expression is the expression itself, second line follows from Eq. (9-40), the third line follows from DeMorgan's law, $(A\cup B)^c = A^c\cap B^c$, the fourth line follows from the duality property of erosion and dilation, Eq. (9-8), the fifth line follows from the symmetry of the SE, and the last line follows from the definition of geodesic dilation.

In step 2 of the proof, we assume that

$$E_G^{(n)}(F) = \left[D_{G^c}^{(1)}\left(D_{G^c}^{(n-1)}(F^c)\right)\right]^c$$

is true. In step 3 we have to prove that

$$E_G^{(n+1)}(F) = \left[D_{G^c}^{(1)} \left(D_{G^c}^{(n)}(F^c) \right) \right]^c$$

is true. We start by recalling from Eq. (9-41) that

$$E_G^{(n)}(F) = E_G^{(1)} \left(E_G^{(n-1)}(F) \right)$$

so,

$$E_G^{(n+1)}(F) = E_G^{(1)} \left(E_G^{(n)}(F) \right)$$

Substituting for $E_G^{(n)}(F)$ (which we assume to be true from step 2) yields

$$E_G^{(n+1)}(F) = E_G^{(1)} \left(\left[D_{G^c}^{(1)} \left(D_{G^c}^{(n-1)}(F^c) \right) \right]^c \right)$$

With reference to Eq. (9-39), we can write the preceding equation as

$$E_G^{(n+1)}(F) = E_G^{(1)} \left(\left[D_{G^c}^{(n)}(F^c) \right]^c \right)$$

For clarity in notation, let $a = \left[D_{G^c}^{(n)}(F^c) \right]^c$, temporarily. Then using the result from step 1 of the proof, we write

$$\begin{aligned} E_G^{(n+1)}(F) &= E_G^{(1)}(a) \\ &= \left[D_{G^c}^{(1)}(a^c) \right]^c \end{aligned}$$

Finally, we substitute for a and obtain

$$E_G^{(n+1)}(F) = \left[D_{G^c}^{(1)} \left(D_{G^c}^{(n)}(F^c) \right) \right]^c$$

This proves the third step of the proof by induction. Therefore, we have proved that

$$E_G^{(n)}(F) = \left[D_{G^c}^{(1)} \left(D_{G^c}^{(n-1)}(F^c) \right) \right]^c$$

Problem 9.43

(a)

$$
\begin{aligned}
(F \ominus nB)^c &= \left[(F \ominus (n-1)B) \ominus B\right]^c \\
&= (F \ominus (n-1)B)^c \oplus \hat{B} \\
&= (F^c \oplus (n-1)\hat{B}) \oplus \hat{B} \\
&= F^c \oplus n\hat{B}.
\end{aligned}
$$

where the second and third lines follow from the duality property in Eq. (9-8). The fourth line clearly is equivalent to the third.

Problem 9.44

(a)

$$
\begin{aligned}
O_R^{(n)}(F) &= R_F^D(F \ominus nB) \\
&= \left[R_{F^c}^E([F \ominus nB]^c)\right]^c \\
&= \left[R_{F^c}^E(F^c \oplus n\hat{B})\right]^c \\
&= \left[R_{F^c}^E(F^c \oplus nB)\right]^c \\
&= \left[C_R^{(n)}(F^c)\right]^c
\end{aligned}
$$

where the second step follows from the duality of reconstruction by dilation (Problem 9.42), the third line follows from the result in Problem 9.43, the fourth line follows from the symmetry of B, and the last step follows from the definition of closing by reconstruction.

Problem 9.45

(a) From Eq. (9-49),

$$
\begin{aligned}
(f \ominus b)^c &= \left[\min_{(s,t)\in b}\{f(x+s, y+t\}\right]^c \\
&= \left[-\max_{(s,t)\in b}\{-f(x+s, y+t\}\right]^c \\
&= \max_{(s,t)\in b}\{-f(x+s, y+t\} \\
&= -f \oplus \hat{b} \\
&= f^c \oplus \hat{b}
\end{aligned}
$$

The second step follows from the definition of the complement of a grayscale function; that is, the minimum of a set of numbers is equal to the negative of the maximum of the negative of those numbers. The third step follows from the definition of the complement. The fourth step follows from the definition of gray-scale dilation in Eq. (9-50), using the fact that $\hat{b}(x,y) = b(-x,-y)$. The last step follows from the definition of the complement, $-f = f^c$.

(d)

$$\begin{aligned}(f \circ b)^c &= \left[(f \ominus b) \oplus b\right]^c \\ &= (f \ominus b)^c \ominus \hat{b} \\ &= (f^c \oplus \hat{b}) \ominus \hat{b} \\ &= f^c \bullet \hat{b}\end{aligned}$$

The first line is from the definition of the opening; the second and third lines are from the duality of erosion and dilation in Eqs. (9-54) and (9-55); and the last line is from the definition of closing in Eq. (9-57).

Problem 9.46

You might find it helpful to review Problem 9.41, where we used proof by induction. This method of proof consists of (1) proving the validity of an expression for $n = 1$; (2) assuming that the expression is true for n; and (3) proving that the expression is true for $n+1$;

(a) For the first step $(n = 1)$, have to show that $D_g^{(1)}(f) = \left[E_{g^c}^{(1)}(f^c)\right]^c$:

$$\begin{aligned}D_g^{(1)}(f) &= \left[\left[D_{g^c}^{(1)}(f)\right]^c\right]^c \\ &= \left[\left[(f \oplus b) \wedge g\right]^c\right]^c \\ &= \left[\left[-\left(-(f \oplus b) \wedge -g\right)\right]^c\right]^c \\ &= \left[-(f \oplus b) \vee -g\right]^c \\ &= \left[(f \oplus b)^c \vee g^c\right]^c \\ &= \left[(f \ominus b) \vee g^c\right]^c \\ &= \left[E_{g^c}^{(1)}(f^c)\right]^c\end{aligned}$$

The second line follows from the definition of geodesic dilation. The third line clearly is equivalent to the second. The fourth and fifth lines follow from the definition of the complement. The sixth line follows from the duality of dilation and erosion (we used the given fact that $\hat{b} = b$). The last line follows from the definition of geodesic erosion.

In the second step, we assume that

$$D_G^{(n)}(f) = \left[E_{g^c}^{(1)}\left(E_{g^c}^{(n-1)}(f^c)\right)\right]^c$$

is true. In the third step, we have to prove that

$$D_G^{(n+1)}(f) = \left[E_{g^c}^{(1)}\left(E_{g^c}^{(n)}(f^c)\right)\right]^c$$

is true. We start by recalling from Eq. (9-41) that

$$D_g^{(n)}(f) = D_g^{(1)}\left(D_g^{(n-1)}(f)\right)$$

so,

$$D_g^{(n+1)}(f) = D_g^{(1)}\left(D_g^{(n)}(f)\right)$$

Substituting for $D_G^{(n)}(f)$ (which we assume to be true) yields

$$D_g^{(n+1)}(f) = D_g^{(1)}\left(\left[E_{g^c}^{(1)}\left(E_{g^c}^{(n-1)}(f^c)\right)\right]^c\right)$$

Then, using Eq. (9-66) we obtain

$$D_g^{(n+1)}(f) = D_g^{(1)}\left(\left[E_{g^c}^{(n)}(f^c)\right]^c\right)$$

For clarity in notation, let $a = \left[E_{g^c}^{(n)}(f^c)\right]^c$, temporarily. Then, using the result from step 1 of the proof, we write

$$\begin{aligned} D_g^{(n+1)}(f) &= D_g^{(1)}(a) \\ &= \left[E_{g^c}^{(1)}(a^c)\right]^c \end{aligned}$$

Finally, we substitute for a and obtain

$$\begin{aligned} D_g^{(n+1)}(f) &= \left[E_{g^c}^{(1)}(a^c)\right]^c \\ &= \left[E_{g^c}^{(1)}\left(E_{g^c}^{(n)}(f^c)\right)\right]^c \end{aligned}$$

which proves that

$$D_g^{(n)}(f) = \left[E_{g^c}^{(1)}\left(E_{g^c}^{(n-1)}(f^c)\right)\right]^c$$

is true.

Problem 9.47

(a)

$$
\begin{aligned}
R_g^D(f) &= D_k^{(k)}(f) \\
&= \left[E_{g^c}^{(1)}\left(E_{g^c}^{(k-1)}\left(f^c\right)\right)\right]^c \\
&= \left[E_{g^c}^{(k)}\left(f^c\right)\right]^c \\
&= \left[R_{g^c}^E\left(f^c\right)\right]^c
\end{aligned}
$$

The first line is from Eq. (9-67), the second from Problem 9.46, the third from Eq. (9-66), and the fourth is from Eq. (9-68).

Problem 9.48

(a) The solution follows the same approach we used in Problem 9.44 for the binary case:

$$
\begin{aligned}
(f \ominus nb)^c &= \left[(f \ominus (n-1)b) \ominus b\right]^c \\
&= (f \ominus (n-1)b)^c \oplus \hat{b} \\
&= (f^c \oplus (n-1)\hat{b}) \oplus \hat{b} \\
&= f^c \oplus n\hat{b}.
\end{aligned}
$$

Problem 9.49

(a) The solution follows the same approach we used in Problem 9.44 for the binary case:

$$
\begin{aligned}
O_R^{(n)}(f) &= R_f^D(f \ominus nb) \\
&= \left[R_{f^c}^E([f \ominus nb]^c)\right]^c \\
&= \left[R_{f^c}^E(f^c \oplus n\hat{b})\right]^c \\
&= \left[R_{f^c}^E(f^c \oplus nb)\right]^c \\
&= \left[C_R^{(n)}(f^c)\right]^c
\end{aligned}
$$

Problem 9.50

(b) The answer in (a) is yes. The boundary is shown in Fig. P9.50(b).

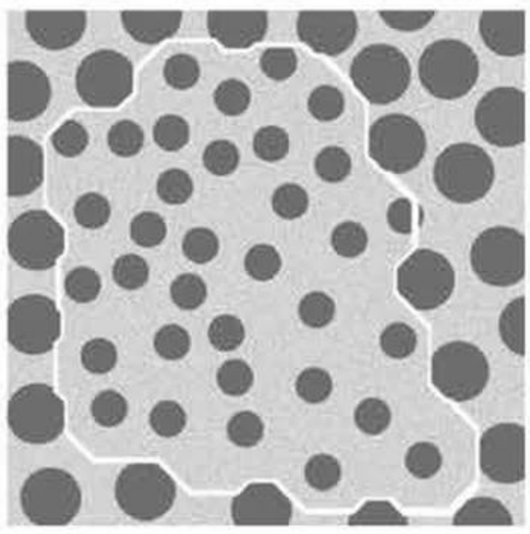


Figure P9.50(b)

Problem 9.51

(a) Color the image border pixels the same color as the particles (white). Call the resulting set of border pixels β. Apply the connected component algorithm (Section 9.6). All connected components that contain elements from β are particles that have merged with the border of the image.

Chapter 10

Problem Solutions

Problem 10.1

The remainder in Eq. (10-4) is

$$R(\text{forward difference}) = \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \frac{1}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \frac{1}{4!}\frac{\partial^4 f(x)}{\partial x^4} + \cdots\cdots$$

Similarly, the remainder in Eq. (10-5) is

$$R(\text{backward difference}) = \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} - \frac{1}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \frac{1}{4!}\frac{\partial^4 f(x)}{\partial x^4} + \cdots\cdots$$

The remainder in Eq. (10-6) is given by the terms left over after subtracting Eq. (10-3) from (10-2):

$$R(\text{central difference}) = \frac{2}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \frac{2}{5!}\frac{\partial^5 f(x)}{\partial x^5} + \frac{2}{7!}\frac{\partial^7 f(x)}{\partial x^7} + \cdots\cdots$$

We see that the lowest-order derivative in this expression is higher than the other two, meaning that the error is lower, as indicated in the problem statement.

Problem 10.2

(a) From Eq. (10-2),

$$f(x+2) = f(x) + 2\frac{\partial f(x)}{\partial x} + 2\frac{\partial^2 f(x)}{\partial x^2} + \frac{4}{3}\frac{\partial^3 f(x)}{\partial x^3} + \cdots\cdots$$

$$f(x-2) = f(x) - 2\frac{\partial f(x)}{\partial x} + 2\frac{\partial^2 f(x)}{\partial x^2} - \frac{4}{3}\frac{\partial^3 f(x)}{\partial x^3} + \cdots\cdots$$

Then,

$$f(x+2) - f(x-2) = 4\frac{\partial f(x)}{\partial x} + \frac{8}{3}\frac{\partial^3 f(x)}{\partial x^3} + \cdots\cdots \quad (1)$$

We are interested in the third derivative, so we have to find a way to eliminate the first derivative term. We can do this by forming $2[f(x-1) - f(x+1)]$ (see Eqs. (10-2) and (10-3)):

$$2[f(x-1) - f(x+1)] = -4\frac{\partial f(x)}{\partial x} - \frac{2}{3}\frac{\partial^3 f(x)}{\partial x^3} + \cdots\cdots \tag{2}$$

Adding Eqs. (1) and (2) and ignoring higher order terms yields,

$$f(x+2) - f(x-2) + 2f(x-1) - 2f(x+1) = 2\frac{\partial^3 f(x)}{\partial x^3}$$

or,

$$\frac{\partial^3 f(x)}{\partial x^3} = \frac{f(x+2) - 2f(x+1) + 2f(x-1) - f(x-2)}{2}$$

which, after including the term $0f(x)$, agrees with Eq. (10-8).

Problem 10.5

The book coordinate system is explained in Fig. 2.19. Angles are measured with respect to the positive x-axis (which points down), with increasing angles measured in the counterclockwise direction. So, horizontal lines are $\pm 90°$ and vertical lines at $0°$ and $180°$.

Problem 10.6

(a) The lines were thicker than the width of the line detector kernels. Thus, when, for example, a kernel was centered on the line it "saw" a constant area and gave a response of 0.

Problem 10.7

(a) The 1st row of Fig. P10.7(a) is the same as Fig. 10.8 in the book. The 2nd row shows the corresponding gradient (magnitude) images and horizontal profiles through their centers. The thin dark borders in the images are included for clarity in defining the borders of the images; they are not part of the image data.

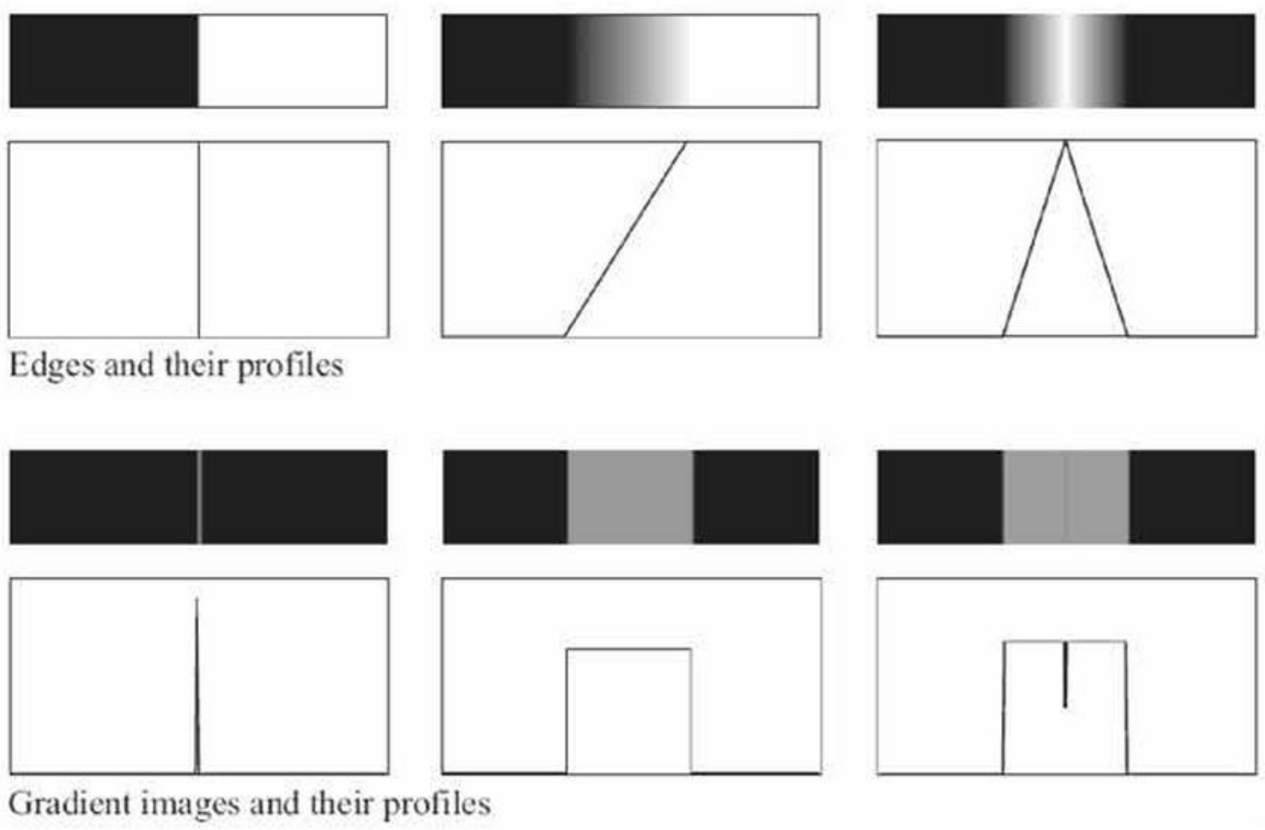


Figure P10.7(a)

Problem 10.9

See Fig. P10.9.

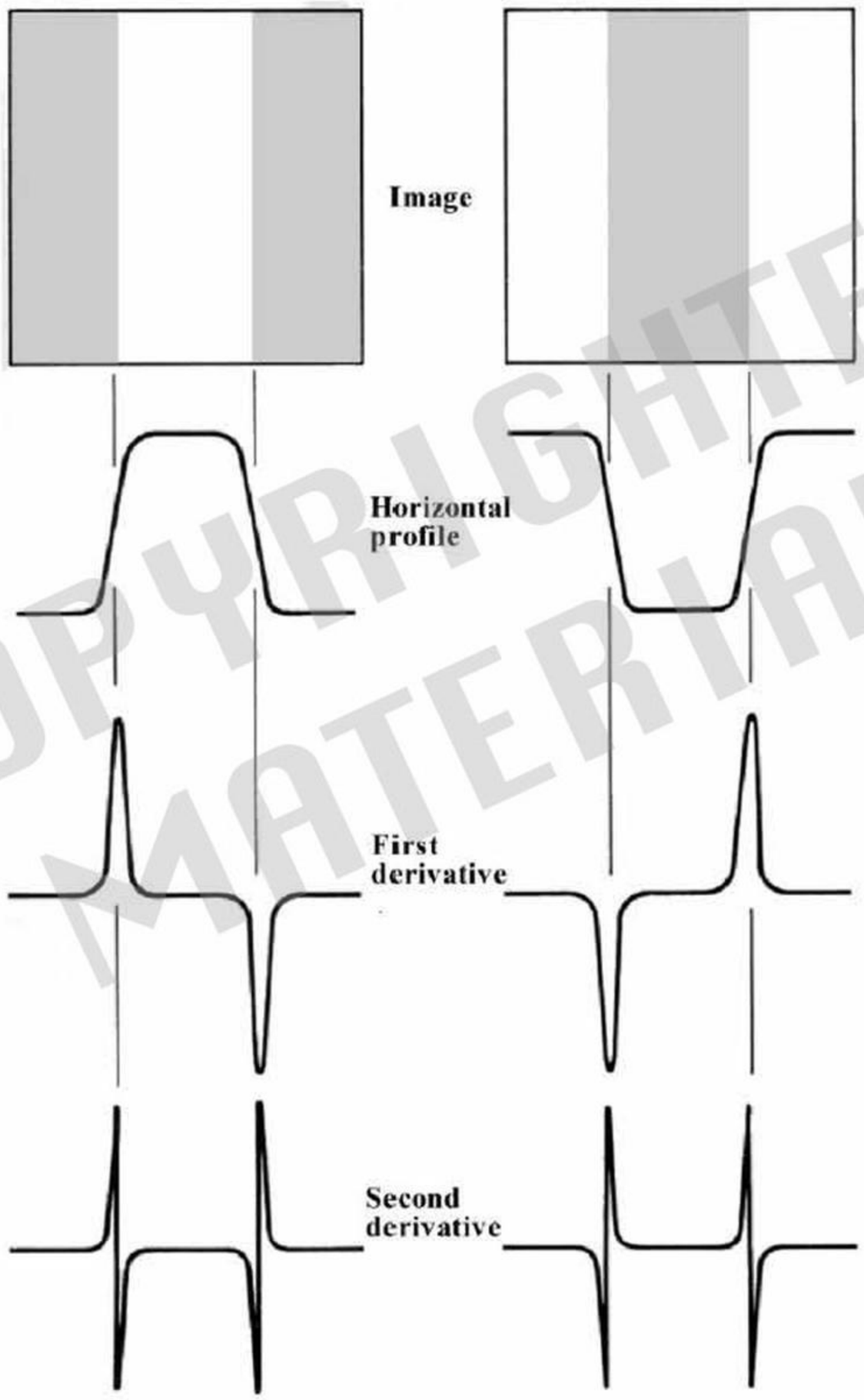


Figure P10.9

Problem 10.10

(a) The gradient is a vector defined as

$$\nabla f = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The derivatives indicate that this vector gives the rate of change along each of the two axes. The question can be reformulated as: In which direction is this directional derivative the maximum? Let **n** be a unit vector oriented in that direction. The dot product of $\nabla f(x, y)$ and **n** is

$$\nabla f(x, y) \cdot \mathbf{n} = \|\nabla f(x, y)\| \|\mathbf{n}\| \cos\theta = \|\nabla f(x, y)\| \cos\theta$$

where we used the fact that the magnitude of the unit vector is 1. The preceding equation achieves its maximum value when $\theta = 0$, which means that **n** is in the same direction as $\nabla f(x, y)$. Since we postulated that **n** points in the direction of maximum rate of change, it follows that $\nabla f(x, y)$ points in the direction of maximum rate of change, and that rate of change is $\|\nabla f(x, y)\|$.

Problem 10.11

(b) Consider first the Sobel kernels in Figs. 10.14 and Fig. P10.11(a) above. A simple way to prove that these kernels give isotropic results for edge segments oriented at multiples of $45°$ is to obtain the kernel responses for the four general edge segments shown in Fig. P10.11(b), which are oriented at increments of $45°$. The objective is to show that the responses of the Sobel kernels are indistinguishable for these four edges. That this is the case is evident from Table P10.11(b), which shows the response of each Sobel kernel to the four general edge segments. We see that in each case the response of the kernel that matches the edge direction is $(4a - 4b)$, and the response of the corresponding orthogonal kernel is 0. The response of the remaining two kernels is either $(3a - 3b)$ or $(3b - 3a)$. The sign difference is not significant because the gradient magnitude is computed by either squaring or taking the absolute value of the kernel responses. The same line of reasoning applies to the Prewitt kernels.

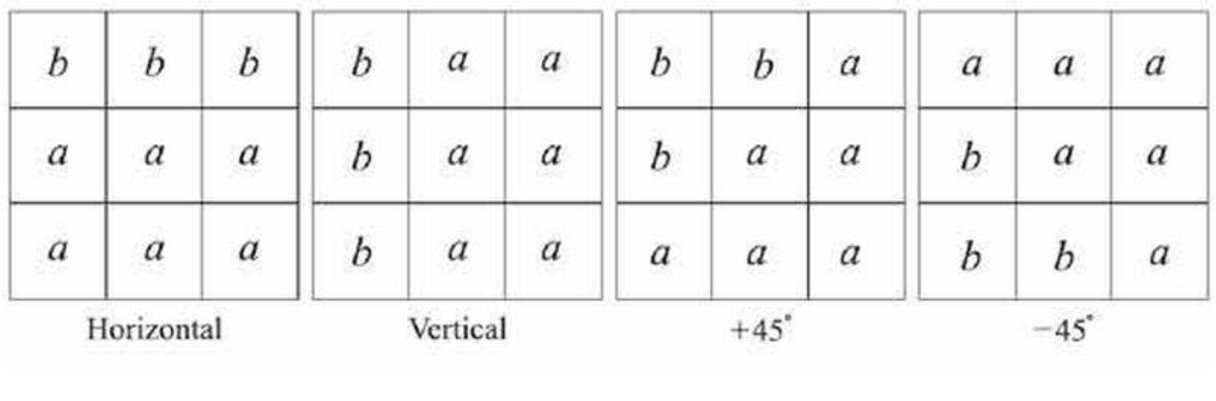


Figure P10.11 (Part b)

Edge direction	Horizontal Sobel (g_x)	Vertical Sobel (g_y)	$+45°$ Sobel (g_{45})	$-45°$ Sobel (g_{-45})
Horizontal	$4a - 4b$	0	$3a - 3b$	$3b - 3a$
Vertical	0	$4a - 4b$	$3a - 3b$	$3a - 3b$
$+45°$	$3a - 3b$	$3a - 3b$	$4a - 4b$	0
$-45°$	$3b - 3a$	$3a - 3b$	0	$4a - 4b$

Table P10.11 (Part b)

Problem 10.13

(a) As discussed in connection with Fig. 10.12, the compass kernels do not use the gradient to compute edge strength. Instead, the strength is obtained by convolving the image with all eight kernels and selecting the largest value. Because every compass kernel has an "opposite" kernel, there will always be a largest positive response in the convolution. What we define as a north (N) kernel, depends on what we define as a north edge. For example, the N kernel in Fig. P10.13(a) would give a positive result for a vertical binary edge with background (0) values on the left, and foreground (1) values on the right. The S kernel would give the same value, but a with a negative sign. For such an edge, the N kernel would give the largest response of all eight kernels. Of course, we could define a north edge with foreground on the left and background on the right, in which case S kernel in Fig. P10.13(a) would become the N kernel. We can elect either edge as our definition of a north edge, but once that is done, we have to select the next (NW) kernel with reference to the coordinate system in Fig. 10.12 and our definition of a positive angle (counterclockwise in our case). That is, once we have defined our N kernel, what is important is that the next (NW) kernel must be equal to the N kernel rotated by +45°. The remaining kernels follow the same pattern of rotation, as Fig. P10.13(a) shows.

−1	0	1	0	1	1	1	1	1	1	1	0	1	0	−1	
−1	0	1	−1	0	1	0	0	0	1	0	−1	1	0	−1	
−1	0	1	−1	−1	0	−1	−1	−1	0	−1	−1	0	−1	−1	
N			NW			W			SE			S			
1	0	−1	0	−1	−1	−1	−1	−1	−1	−1	0	1	0	−1	
1	0	−1	1	0	−1	0	0	0	−1	0	1	1	0	−1	
1	0	−1	1	1	0	1	1	1	0	1	1	1	0	−1	
S			SE			E			NE			SE			

Figure P10.13 (Part a)

Problem 10.14

(a) The solution is shown in Fig. P10.14(a). The numbers in brackets are values of $[g_x, g_y]$.

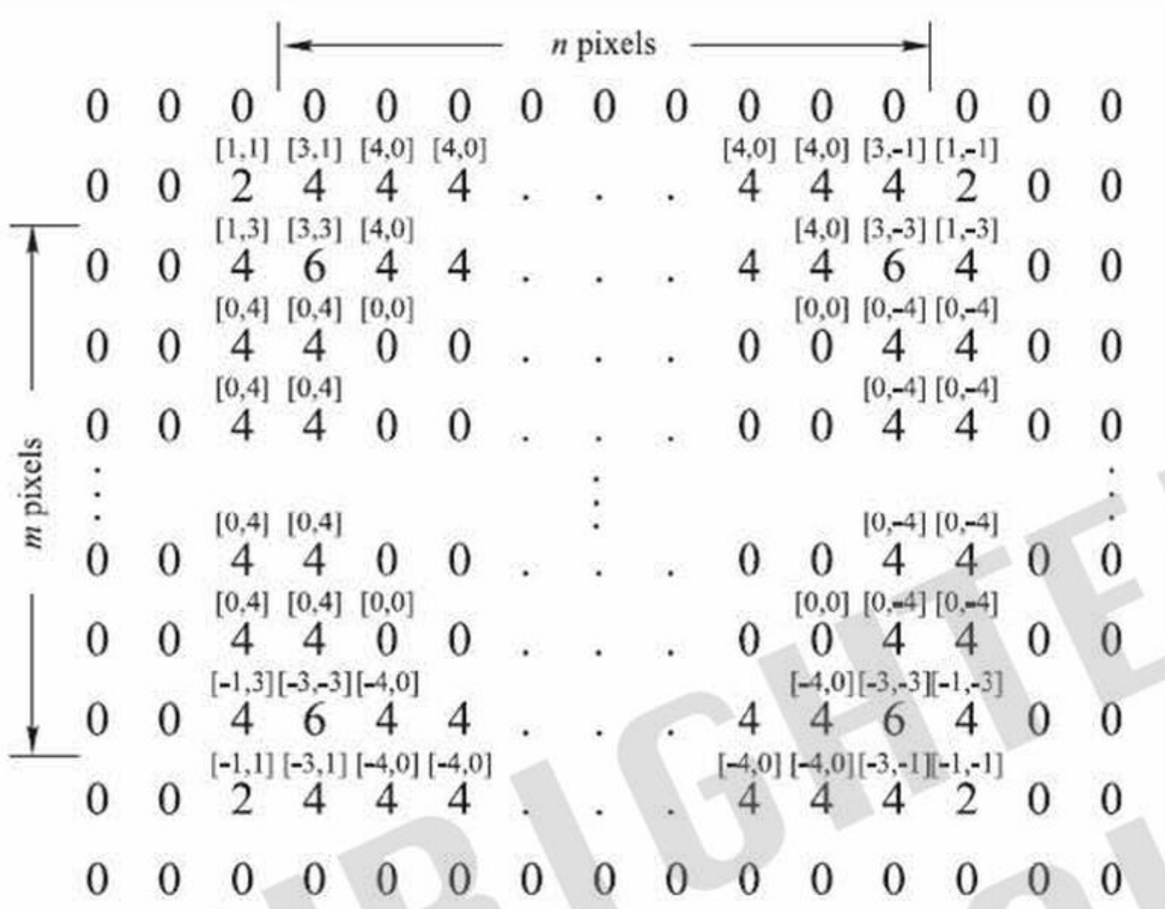


Figure P10.14 (Part a)

Problem 10.15

(a) The smoothing kernel mentioned in the problem statement produces an average value at (x, y) given by

$$\bar{f}(x,y) = \frac{1}{n^2} \sum_{z_i \in S_{xy}} z_i$$

where S_{xy} is the region in the image spanned by the $n \times n$ averaging kernel when it is centered at (x, y) and the z_i are the intensities of the image pixels in that region. The partial

$$\partial \bar{f} / \partial x = \bar{f}(x+1, y) - \bar{f}(x, y)$$

is thus given by

$$\partial \bar{f} / \partial x = \frac{1}{n^2} \sum_{z_i \in S_{x+1,y}} z_i - \frac{1}{n^2} \sum_{z_i \in S_{xy}} z_i$$

The first summation on the right can be interpreted as consisting of all the pixels in the second summation minus the pixels in the first row of the kernel, plus the row picked up by the kernel as it moved from (x, y) to $(x+1, y)$. Thus, we can write the preceding equation as

$$
\begin{aligned}
\partial\bar{f}/\partial x &= \frac{1}{n^2}\sum_{z_i \in S_{x+1,y}} z_i - \frac{1}{n^2}\sum_{z_i \in S_{xy}} z_i \\
&= \left(\left(\frac{1}{n^2}\sum_{z_i \in S_{xy}} z_i\right) + \frac{1}{n^2}(\text{sum of pixels in new row})\right. \\
&\qquad \left. - \frac{1}{n^2}(\text{sum of pixels in 1st row})\right) - \frac{1}{n^2}\sum_{z_i \in S_{xy}} z_i \\
&= \frac{1}{n^2}\sum_{k=y-\frac{n-1}{2}}^{y+\frac{n-1}{2}} f\left(x + \frac{n+1}{2}, k\right) - \frac{1}{n^2}\sum_{k=y-\frac{n-1}{2}}^{y+\frac{n-1}{2}} f\left(x - \frac{n+1}{2}, k\right) \\
&= \frac{1}{n^2}\left[\sum_{k=y-\frac{n-1}{2}}^{y+\frac{n-1}{2}} f\left(x + \frac{n+1}{2}, k\right) - f\left(x - \frac{n+1}{2}, k\right)\right]
\end{aligned}
$$

This expression gives the value of $\partial\bar{f}/\partial x$ at coordinates (x, y) of the smoothed image. Similarly,

$$
\partial\bar{f}/\partial x = \frac{1}{n^2}\left[\sum_{k=x-\frac{n-1}{2}}^{x+\frac{n-1}{2}} f\left(k, y + \frac{n+1}{2}\right) - f\left(k, y - \frac{n+1}{2}\right)\right]
$$

The edge magnitude image corresponding to the smoothed image, $\bar{f}(x, y)$, is then given by

$$
\bar{M}(x, y) = \sqrt{(\partial\bar{f}/\partial x)^2 + (\partial\bar{f}/\partial y)^2}
$$

Problem 10.16

(a) We proceed as follows:

$$
\begin{aligned}
\text{Average}\left[\nabla^2 G(x, y)\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \nabla^2 G(x, y)\, dx\, dy \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}\, dx\, dy \\
&= \frac{1}{\sigma^4}\int_{-\infty}^{\infty} x^2 e^{-\frac{x^2}{2\sigma^2}} dy \int_{-\infty}^{\infty} e^{-\frac{y^2}{2\sigma^2}} dy \\
&\qquad + \frac{1}{\sigma^4}\int_{-\infty}^{\infty} y^2 e^{-\frac{y^2}{2\sigma^2}} dy \int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} dx \\
&\qquad - \frac{2}{\sigma^2}\int_{-\infty}^{\infty} e^{-\frac{x^2+y^2}{2\sigma^2}}\, dx\, dy
\end{aligned}
$$

$$= \frac{1}{\sigma^4}\left(\sqrt{2\pi}\sigma \times \sigma^2\right)\left(\sqrt{2\pi}\sigma\right)$$
$$+ \frac{1}{\sigma^4}\left(\sqrt{2\pi}\sigma \times \sigma^2\right)\left(\sqrt{2\pi}\sigma\right)$$
$$- \frac{2\left(2\pi\sigma^2\right)}{\sigma^2}$$
$$= 4\pi - 4\pi$$
$$= 0$$

The fourth line follows from the fact that

$$\text{variance}(z) = \sigma^2 = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} z^2 e^{-\frac{z^2}{2\sigma^2}} dz$$

and

$$\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{-\frac{z^2}{2\sigma^2}} dz = 1$$

Problem 10.17

(b) The answer is yes for functions that meet certain mild conditions, and if the zero crossing method is based on rotational operators like the LoG function and a threshold of 0. Geometrical properties of zero crossings in general are explained in some detail in the paper "On Edge Detection," by V. Torre and T. Poggio, *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 8, no. 2, 1986, pp. 147-163. Looking up this paper and becoming familiar with the mathematical underpinnings of edge detection is an excellent reading assignment for graduate students.

Problem 10.19

(a) From Eq. (10-32), we see that the DoG function is zero when

$$\frac{1}{2\pi\sigma_1^2}e^{-\frac{x^2+y^2}{2\sigma_1^2}} = \frac{1}{2\pi\sigma_2^2}e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

taking the natural log of both sides yields

$$\ln\left[\frac{1}{2\pi\sigma_1^2}\right] - \frac{x^2+y^2}{2\sigma_1^2} = \ln\left[\frac{1}{2\pi\sigma_2^2}\right] - \frac{x^2+y^2}{2\sigma_2^2}$$

combining terms,

$$\left(x^2+y^2\right)\left[\frac{1}{2\sigma_1^2}-\frac{1}{2\sigma_2^2}\right]=\ln\left[\frac{1}{2\pi\sigma_1^2}\right]-\ln\left[\frac{1}{2\pi\sigma_2^2}\right]$$
$$=\ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$$

The LoG function in Eq. (10-29) is zero when $x^2+y^2=2\sigma^2$. Then, from the preceding equation,

$$\sigma^2\left[\frac{1}{2\sigma_1^2}-\frac{1}{2\sigma_2^2}\right]=\ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$$

Finally, solving for σ^2,

$$\sigma^2=\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2-\sigma_2^2}\ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$$

which agrees with Eq. (10-33).

Problem 10.20

(b) Direct implementation of 2-D spatial convolution requires n^2 multiplications at each location of $f(x,y)$, so the total number of multiplications is $n^2\times M\times N$. On the other hand, 1-D convolution requires n multiplications at each location of every row in the image, for a total of $n\times M\times N$ multiplications for the -pass along all rows. Then, $n\times M\times N$ are required for the pass along all columns, for a total of $2nMN$ multiplications. The computational advantage, A, is then

$$A=\frac{n^2MN}{2nMN}=\frac{n}{2}$$

which is independent of image size. For example, if $n=25$, $A=12.5$, so it takes 12.5 times more multiplications to implement 2-D convolution directly than it does to implement 1-D convolution. This agrees with Eq. (3-44) when $m=n$ in that equation. Of course, the number of multiplications itself is very much dependent on image size.

Problem 10.22

(a) The solution follows the same approach as in Problem 10.21, but using the 1-D kernels in Fig. 10.13 to implement Eqs. (10-19) and (10-20).

Problem 10.23

(a) See Fig. P10.23(a).

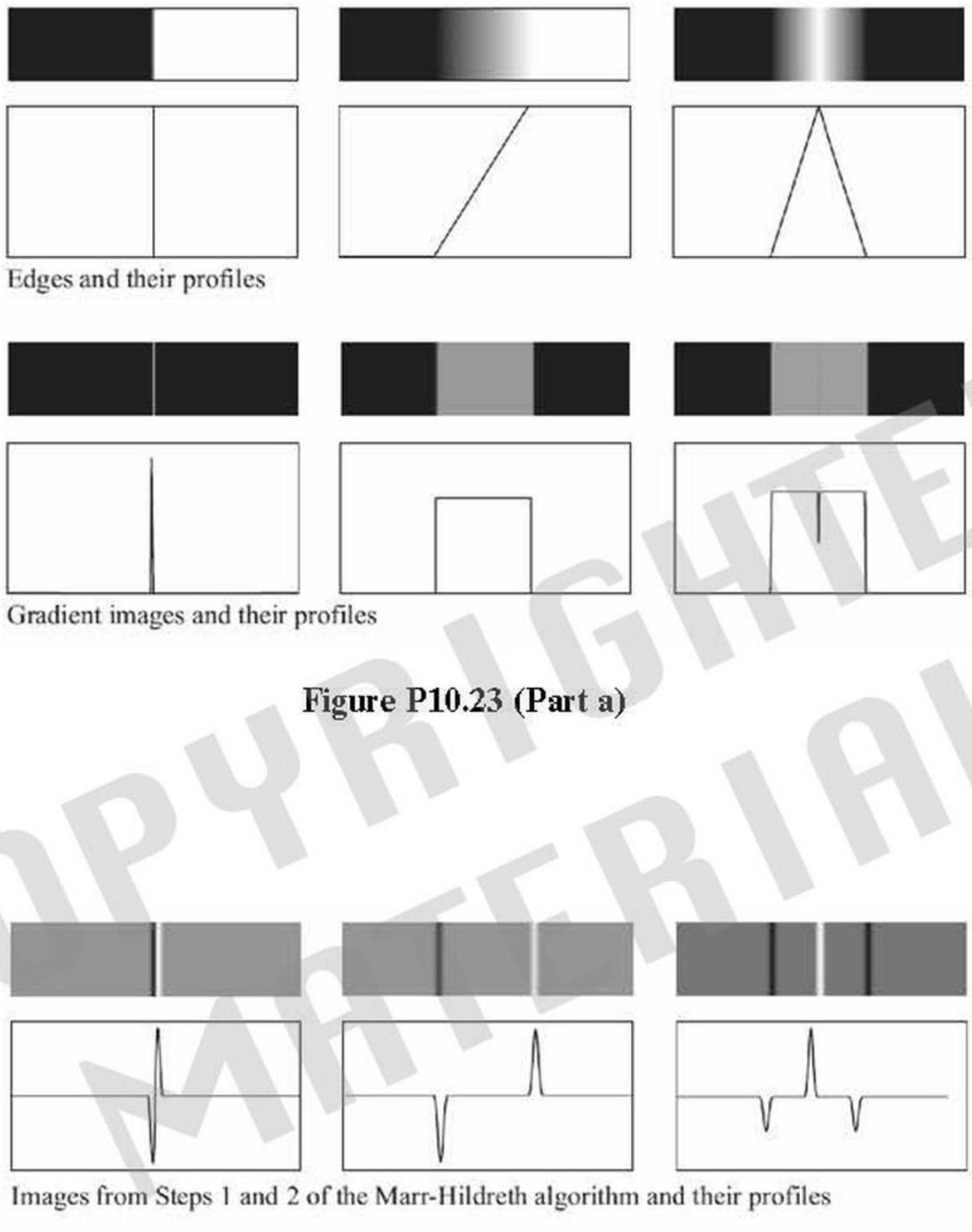


Figure P10.23 (Part a)

(c) See Fig. P10.23(c).

Figure P10.23 (Part c)

Problem 10.25

(b) $\theta = \cot^{-1}(2) = 26.6°$ and $\rho = (1)\sin\theta = 0.45$.

Problem 10.26

(a) Point 1 has coordinates $x = 0$ and $y = 0$. Substituting into Eq. (10-44) yields $\rho = 0$ which is a straight line in a plot of ρ vs. θ.

(b) Only the origin $(0, 0)$ would yield this result.

Problem 10.29

The essence of the algorithm is to compute at each step the mean value, m_1, of all pixels whose intensities are less than or equal to the previous threshold and, similarly, the mean value, m_2, of all pixels with values that exceed the threshold. Let $p_i = n_i/n$ denote the ith component of the image histogram, where n_i is the number of pixels with intensity i, and n is the total number of pixels in the image. Valid values of i are in

the range $0 \le i \le L-1$, where L is the number on intensities and i is an integer. The means can be computed at any step k of the algorithm using

$$m_1(k) = \sum_{i=0}^{I(k-1)} i\,p_i / P(k)$$

where

$$P(k) = \sum_{i=0}^{I(k-1)} p_i$$

and

$$m_2(k) = \sum_{i=I(k-1)+1}^{L-1} i\,p_i / [1 - P(k)]$$

The term $I(k-1)$ is the smallest integer less than or equal to $T(k-1)$, and $T(0)$ is given. The next value of the threshold is then

$$T(k+1) = \frac{1}{2}[m_i(k) + m_2(k)]$$

Problem 10.30

As stated in Section 10.3, we assume that the initial threshold is chosen between the minimum and maximum intensities in the image. To begin, consider the histogram in Fig. P10.30. It shows the threshold at the kth iterative step, and the fact that the mean $m_1(k+1)$ will be computed using the intensities greater than $T(k)$ times their histogram values. Similarly, $m_2(k+1)$ will be computed using values of intensities less than or equal to $T(k)$ times their histogram values. Then, $T(k+1) = 0.5[m_i(k) + m_2(k)]$. The proof consists of two parts. First, we prove that the threshold is bounded between 0 and $L-1$. Then we prove that the algorithm converges to a value between these two limits.

To prove that the threshold is bounded, we write $T(k+1) = 0.5[m_i(k) + m_2(k)]$. If $m_2(k+1) = 0$, then $m_1(k+1)$ will be equal to the image mean, M, and $T(k+1)$ will equal $M/2$, which is less than $L-1$. If $m_1(k+1)$ is zero, the same will be true. Both m_1 and m_2 cannot be zero simultaneously, so $T(k+1)$ will always be greater than 0 and less than $L-1$.

To prove convergence, we have to consider three possible conditions:

1. $T(k+1) = T(k)$, in which case the algorithm has converged.
2. $T(k+1) < T(k)$, in which case the threshold moves to the left.
3. $T(k+1) > T(k)$, in which the threshold moves to the right.

In case (2), when the threshold value moves to the left, m_2 will decrease or stay the same and m_1 will also decrease or stay the same (the fact that m_1 decreases or stays the same is not necessarily obvious. If you don't see it, draw a simple histogram and convince yourself that it does), depending on how much the threshold moved and on the values of the histogram. However, neither mean can increase. If neither mean changes, then $T(k+2)$ will equal $T(k+1)$ and the algorithm will stop. If either (or both) mean decreases, then $T(k+2) < T(k+1)$, and the new threshold moves further to the left. This will cause the conditions just stated to happen again, so the conclusion is that if the thresholds starts moving left, it will always move left, and the algorithm will eventually stop with a value $T > 0$, which we know is the lower bound for T. Because the threshold always decreases or stops changing, no oscillations are possible, so the algorithm is guaranteed to converge in a finite number of steps.

Case (3) causes the threshold to move to the right. An argument similar to the preceding discussion establishes that if the threshold starts moving to the right it will either converge or continue moving to the right and will stop eventually with a value less than $L-1$. Because the threshold always increases or stops changing, no oscillations are possible, so the algorithm is guaranteed to converge.

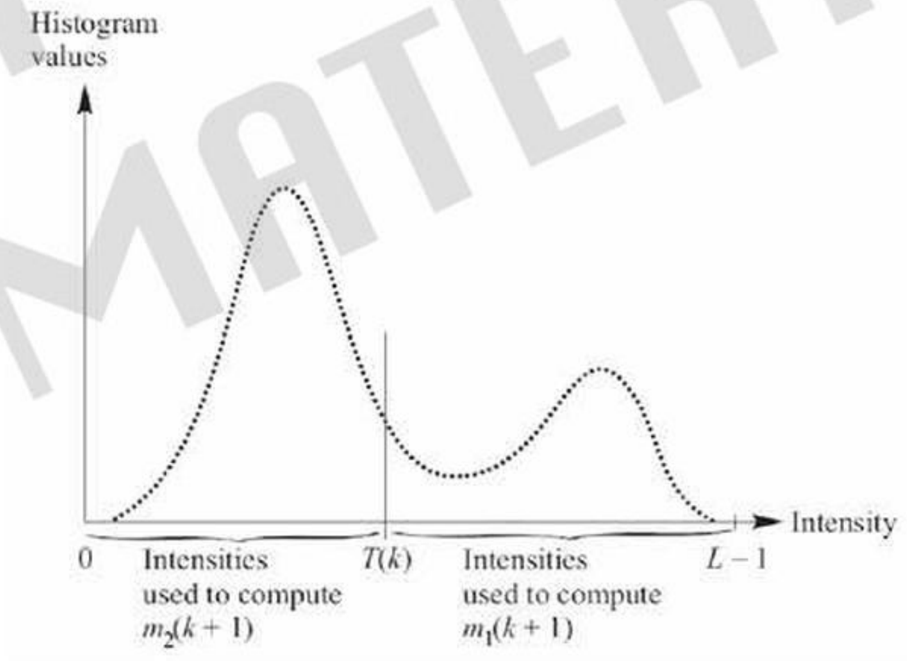


Figure P10.30

Problem 10.32

We know from Problem 10.32 that if the initial threshold value is selected between the minimum and maximum intensity values, the algorithm is guaranteed to converge. However, nothing is stated in the proof about the final value being independent of the initial value. It stands to reason to conclude that the threshold value at convergence is influenced by the shape of the image histogram. Consequently, we may expect that in general the final value will be influenced by the starting value. We can show that this is true using a simple example. Let $f = [0\,1\,2;3\,4\,5;6\,7\,8]/8$. If we choose an initial value of 0.1, the algorithm will converge to $T = 0.4688$. If we start with $T = 0.8$, the algorithm converges to $T = 0.5313$.

Problem 10.33

(a) For a uniform histogram, we can view the intensity levels as points of unit mass along the intensity axis of the histogram. Any values $m_1(k)$ and $m_2(k)$ are the means of the two groups of intensity values G_1 and G_2. Because the histogram is uniform, these are the centers of mass of G_1 and G_2. We know from the solution of Problem 10.30 that if T starts moving to the right, it will always move in that direction, or stop. The same holds true for movement to the left. Now, assume that $T(k)$ has arrived at the center of mass (average intensity). Because all points have equal "weight" (remember the histogram is uniform), if $T(k+1)$ moves to the right G_2 will pick up, say, Q new points. But G_1 will lose the same number of points, so the sum $m_1 + m_2$ will be the same and the algorithm will stop.

Problem 10.34

(a) $A_1 = A_2$ and $\sigma_1 = \sigma_2 = \sigma$, which makes the two modes identical [this is the same as Problem 10.33(b)].

(b) You know from Problem 10.33 that if the modes were symmetric and identical, the algorithm would converge to the point midway between the means. In the present problem, the modes are symmetric but they may not be identical. Then, all we can say is that the algorithm will converge to a point somewhere between the means (from Section 10.3 we know that the algorithm must start at some point between the minimum and maximum image intensities). So, if both A_1 and A_2 are greater than 0, we are assured that the algorithm will converge to a point somewhere between m_1 and m_2.

Problem 10.35

(a)

$$
\begin{aligned}
\sigma_B^2 &= P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \\
&= P_1[(m_1 - (P_1m_1 + P_2m_2)]^2 + P_2[(m_2 - (P_1m_1 + P_2m_2)]^2 \\
&= P_1[(m_1 - m_1(1 - P_2) - P_2m_2)]^2 + P_2[(m_2 - m_1P_1 - m_2(1 - P_1)]^2 \\
&= P_1[P_2m_1 - P_2m_2]^2 + P_2[P_1m_2 - P_1m_1]^2 \\
&= P_1P_2^2(m_1 - m_2)^2 + P_2P_1^2(m_1 - m_2)^2 \\
&= (m_1 - m_2)^2[P_1P_2^2 + P_2P_1^2] \\
&= (m_1 - m_2)^2[P_1P_2(P_2 + P_1)] \\
&= P_1P_2(m_1 - m_2)^2
\end{aligned}
$$

where we used the facts that $m_G = P_1m_1 + P_2m_2$ and $P_1 + P_2 = 1$. This proves the first part of Eq. (10-60).

Problem 10.37

From the definition in Eq. (10-57).

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

where

$$\begin{aligned}\sigma_B^2 &= P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \\ &= P_1 P_2 (m_1 - m_2)^2\end{aligned}$$

and

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$$

As in the text, we have omitted k in σ_B^2 for the sake of notational clarity, but the assumption is that $0 \le k \le L-1$. As explained in the book, the minimum value of η is zero, and it occurs when the image is constant. It remains to be shown that the maximum value is 1, and that it occurs for two-valued images with values 0 and $L-1$.

From the second line of the expression for σ_B^2 we see that the maximum occurs when the quantity $(m_1 - m_2)^2$ is maximum because P_1 and P_2 are positive. The intensity scale extends from 0 to $L-1$, so the maximum difference between means occurs when $m_1 = 0$ and $m_2 = L-1$. But the only way this can happen is if the variance of the two classes of pixels is zero, which implies that the image only has these two values, thus proving the assertion that the maximum occurs only when the image is two-valued with intensity values 0 and $L-1$. It remains to be shown that the maximum possible value of η is 1.

When $m_1 = 0$ and $m_2 = L-1$,

$$\begin{aligned}\sigma_B^2 &= P_1 P_2 (m_1 - m_2)^2 \\ &= P_1 P_2 (L-1)^2\end{aligned}$$

For an image with values 0 and $L-1$,

$$\begin{aligned}\sigma_G^2 &= \sum_{i=0}^{L-1} (i - m_G)^2 p_i \\ &= \sum_{i=0}^{k} (i - m_G)^2 P_1 + \sum_{i=k+1}^{L-1} (i - m_G)^2 P_2 \\ &= (0 - m_G)^2 P_1 + (L-1-m_G)^2 P_2\end{aligned}$$

From Eq. (10-55)

$$\begin{aligned}m_G &= P_1 m_1 + P_2 m_2 \\ &= P_2 (L-1)\end{aligned}$$

so,

$$
\begin{aligned}
\sigma_G^2 &= (0 - m_G)^2 P_1 + (L-1-m_G)^2 P_2 \\
&= P_2^2 (L-1)^2 P_1 + (L-1)^2 (1-P_2)^2 P_2 \\
&= (L-1)^2 (P_2^2 P_1 + P_1^2 P_2) \\
&= (L-1)^2 P_2 P_1 (P_2 + P_1) \\
&= P_2 P_1 (L-1)^2
\end{aligned}
$$

where we used the fact that $P_1 + P_2 = 1$. We see from the preceding results for σ_B^2 and σ_G^2 that $\sigma_B^2 / \sigma_G^2 = 1$ when the image is two-valued, with values 0 and $L-1$. This completes the proof.

Problem 10.38

(a) Let R_1 and R_2 denote the regions whose pixel intensities are greater than T and less or equal to T, respectively. The threshold T is an intensity value, so it gets mapped by the transformation function to the value $T' = 1 - T$. Values in R_1 are mapped to R_1' and values in R_2 are mapped to R_2'. The important thing is that all values in R_1' are below T' and all values in R_2' are equal to or above T'. The sense of the inequalities has been reversed, but the separability of the intensities in the two regions has been preserved.

Problem 10.40

(a) The first column would be black and all other columns would be white. The reason is that a point in the segmented image is set to 1 if the value of the image at that point exceeds b at that point. But $b = 0$, so all points in the image that are greater than 0 will be set to 1 and all other points would be set to 0. But the only points in the image that do not exceed 0 are the points that are 0, which are the points in the first column.

Problem 10.42

The region splitting is shown in Fig. P10.42(a). The corresponding quad tree is shown in Fig. P10.42(b).

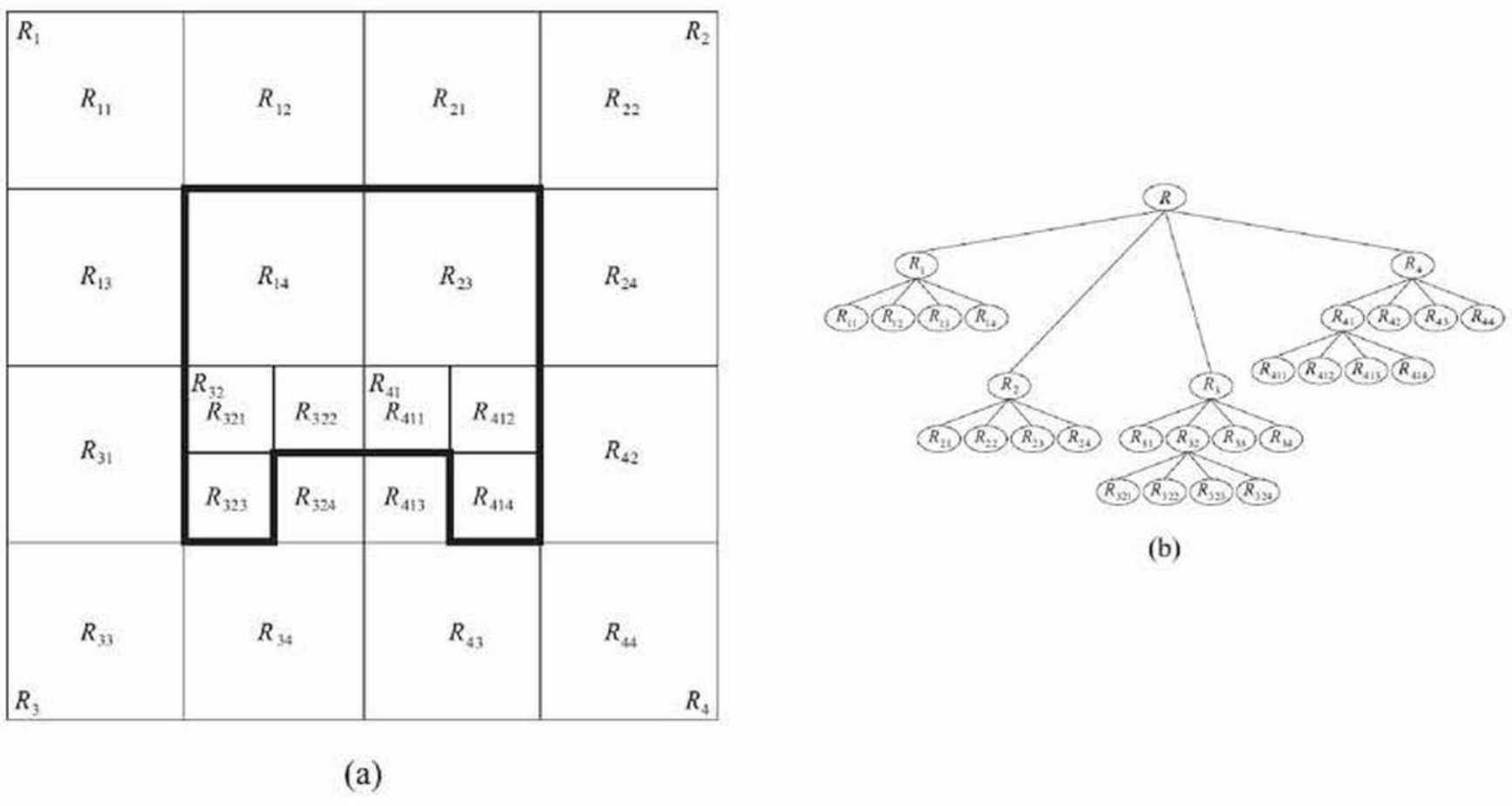


Figure P10.42

Problem 10.45

Equation (10-105)

$$(\mathbf{D}-\mathbf{W})\mathbf{y}=\lambda\mathbf{D}\mathbf{y}$$

can be written as

$$(\mathbf{D}-\mathbf{W})(\mathbf{D}^{-1/2}\mathbf{D}^{1/2})\mathbf{y}=\lambda\mathbf{D}\mathbf{y}$$

Pre-multiplying both sides by $\mathbf{D}^{-1/2}$ gives us

$$\mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-1/2}\mathbf{D}^{1/2}\mathbf{y}=\lambda\mathbf{D}^{-1/2}\mathbf{D}\mathbf{y}$$

But $\mathbf{D}^{-1/2}\mathbf{D}=\mathbf{D}^{1/2}$, so the previous equation becomes

$$\mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-1/2}\mathbf{D}^{1/2}\mathbf{y}=\lambda\mathbf{D}^{1/2}\mathbf{y}$$

Letting

$$\mathbf{A}=\mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-1/2}$$

which agrees with Eq. (107), and

$$\mathbf{z}=\mathbf{D}^{1/2}\mathbf{y}$$

which agrees with Eq. (108), we have the standard eigenvector eigenvalue result

$$\mathbf{A}\mathbf{z}=\lambda\mathbf{z}$$

which agrees with Eq. (10-106).

Problem 10.47

(a) The elements of $T[n]$ are the coordinates of points in the image below the plane $g(x,y)=n$, where n is an integer that represents a given step in the execution of the algorithm. Because n never decreases, the set of elements in $T[n-1]$ is a subset of the elements in $T[n]$. In addition, we note that all the points below the plane $g(x,y)=n-1$ are also below the plane $g(x,y)=n$, so the elements of $T[n]$ are never replaced. Similarly, $C_n(M_i)$ is formed by the intersection of $C(M_i)$ and $T[n]$, where $C(M_i)$ (whose elements never change) is the set of coordinates of *all* points in the catchment basin associated with regional minimum M_i. Because the elements of $C(M_i)$ never change, and the elements of $T[n]$ are never replaced, it follows that the elements in $C_n(M_i)$ are never replaced either. In addition, we see that $C_{n-1}(M_i)\subseteq C_n(M_i)$.

Problem 10.49

The first step in the application of the watershed segmentation algorithm is to build a dam of height max+1 to prevent the rising water from running off the ends of the function, as shown in Fig. P10.49(b). For an image function we would build a box of height max+1 around its border. The algorithm is initialized by setting $C[1] = T[1]$. In this case, $T[1] = \{g(2)\}$, as shown in Fig. P10.49(c) (note the water level). There is only one connected component in this case: $Q[1] = \{q_1\} = \{g(2)\}$.

Next, we let $n = 2$ and, as shown in Fig. P10.49(d), $T[2] = \{g(2), g(14)\}$ and $Q[2\} = \{q_1; q_2\}$, where, for clarity, different connected components are separated by semicolons. We start construction of $C[2\}$ by considering each connected component in $Q[2]$. When $q = q_1$, the term $q \cap C[1]$ is equal to $\{g(2)\}$, so condition 2 is satisfied and, therefore, $C[2] = \{g(2)\}$. When $q = q_2$, $q \cap C[1] = \varnothing$ (the empty set) so condition 1 is satisfied and we incorporate q in $C[2]$, which then becomes $C[2] = \{g(2); g(14)\}$ where, as above, different connected components are separated by semicolons.

When $n = 3$ [Fig. P10.49 (e)], $T[3] = \{2,3,10,11,13,14\}$ and $Q[3] = \{q_1; q_2; q_3\} = \{2,3;10,11;13,14\}$ where, in order to simplify the notation we let k denote $g(k)$. Proceeding as above, $q_1 \cap C[2] = \{2\}$ satisfies condition 2, so q_1 is incorporated into the new set to yield $C[3] = \{2,3;14\}$. Similarly, $q_2 \cap C[2] = \varnothing$ satisfies condition 1 and $C[3] = \{2,3;10,11;13,14\}$. Finally, q3 ∩ C[2] = {14} satisfies condition 2 and C[3] = {2, 3;10,11;13,14}. It is easily verified that C[4] = C[3] = {2, 3;10,11;13,14}.

When $n = 5$ [Fig. P10.49(f)], we have, $T[5] = \{2,3,5,6,10,11,12,13,14\}$ and $Q[5] = \{q_1; q_2; q_3\} = \{2,3;5,6;10,11,12,13,14\}$ (note the merging of two previously distinct connected components). It is easily verified that $q1 \cap C[4]$ satisfies condition 2 and that $q_2 \cap C[4]$ satisfies condition 1. Proceeding with these two connected components exactly as above yields $C[5] = \{2,3;5,6;10,11;13,14\}$ up to this point. Things get more interesting when we consider q_3. Now, $q_3 \cap C[4] = \{10,11;13,14\}$ which, because it contains two connected components of $C[4]$, satisfies condition 3. As mentioned previously, this is an indication that water from two different basins has merged and a dam must be built to prevent this condition. Dam building is nothing more than separating q_3 into the two original connected components. In this case, this is accomplished by the dam shown in Fig. P10.49(g), so that now $q_3 = \{q_{31}; q_{32}\} = \{10,11;13,14\}$. Then, $q_{31} \cap C[4]$ and $q_{32} \cap C[4]$ each satisfy condition 2 and we have the final result for $n = 5$, $C[5] = \{2, 3; 5, 6;10,11;13;14\}$.

Continuing in the manner just explained yields the final segmentation result shown in Fig. P10.49(h), where the "edges" are visible (from the top) just above the water line. A final post-processing step would remove the outer dam walls to yield the inner edges of interest.

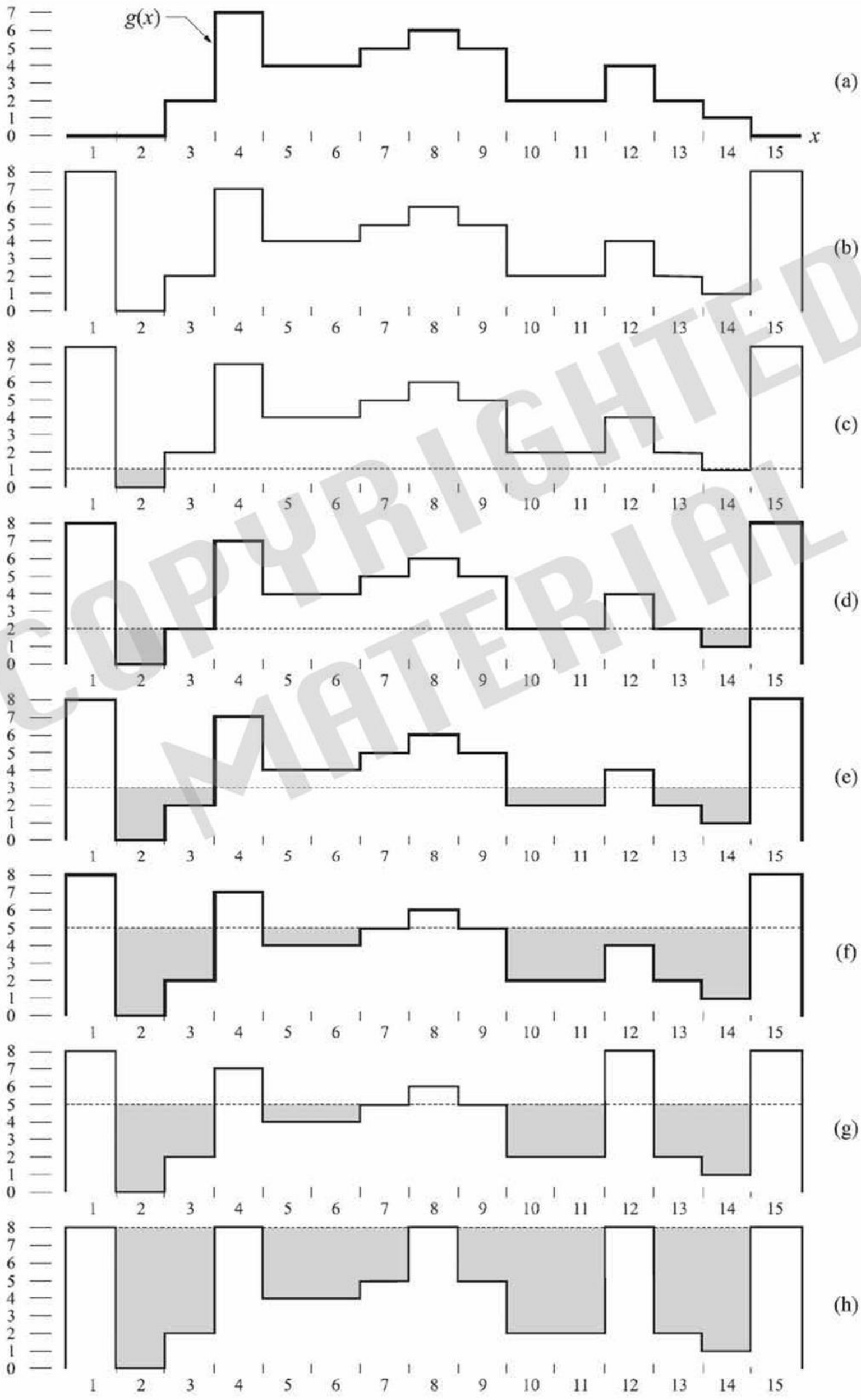


Figure P10.49

Problem 10.51

(a) *True*, assuming that the threshold is not set larger than all the differences encountered as the object moves. The easiest way to see this is to draw a simple reference image, such as the white rectangle on a black background. Let that rectangle be the object that moves. Because the absolute ADI image value at any location is the absolute difference between the reference and the new image, it is easy to see that as the object enters areas that are background in the reference image, the absolute difference will change from zero to nonzero at the new area occupied by the moving object. Thus, as long as the object moves, the dimension of the absolute ADI will grow.

Problem 10.53

Recall that velocity is a vector, whose magnitude is speed. Function g_x is a one-dimensional "record" of the position of the moving object as a function of time (frame rate). The value of velocity (speed) is determined by taking the first derivative of this function. To determine whether velocity is positive or negative at a specific time, n, we compute the instantaneous acceleration (rate of change of speed) at that point; that is we compute the second derivate of g_x. Viewed another way, we determine direction by computing the derivative of the derivative of g_x. But, the derivative at a point is simply the tangent at that point. If the tangent has a positive slope, the velocity is positive; otherwise it is negative or zero. Because g_x is a complex quantity, its tangent is given by the ratio of its imaginary to its real part. This ratio is positive when S_{1x} and S_{2x} have the same sign, which is what we started out to prove.

Problem 10.55

(a) It is given that 10% of the image area in the horizontal direction is occupied by a bullet that is 2.5 cm long. Because the imaging device is square (256 × 256 elements) the camera looks at an area that is 25 cm × 25 cm, assuming no optical distortions. Thus, the distance between pixels is 25/256 =0.098 cm/pixel. The maximum speed of the bullet is 1000m/sec = 100,000 cm/sec. At this speed, the bullet will travel $100{,}000/0.98 = 1.02 \times 10^6$ pixels/sec. It is required that the bullet not travel more than one pixel during exposure. That is, $(1.02 \times 10^6 \text{ pixels/sec}) \times K \text{ sec} \leq 1$ pixel. So, $K \leq 9.8 \times 10^{-7}$ sec.

(c) In a flashing situation with a reflective object, the images will tend to be dark, with the object shining brightly. The spatial techniques discussed in Section 10.8 would then be quite adequate.

Chapter 11

Problem Solutions

Problem 11.1

(a) The solution is shown in Fig. P11.1(a), where the black pixel is b_0, the starting point, and the dark gray point is b_1, the next boundary point found by the algorithm, b_1. As mentioned in the book, the algorithm stops when it returns to b_0 and the next point found after that is b_1.

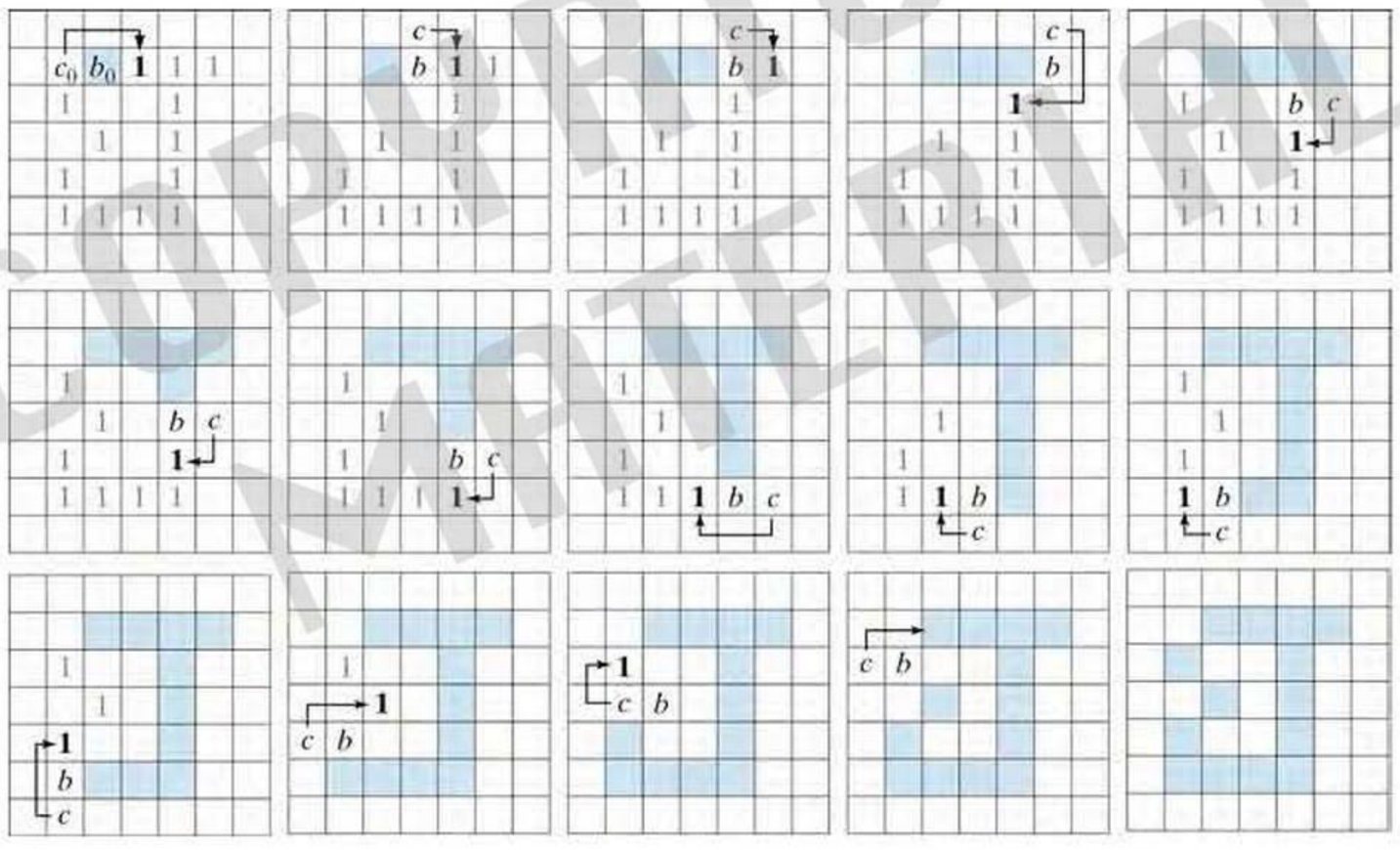


Figure P11.1(a)

Problem 11.2

(a) The algorithm would start at point $(2,4)$. At that point, $b=(2,4)$ and $c=(2,3)$. It would end at point $(3,3)$. At that point, $b=(3,3)$ and $c=(3,2)$. The end point is at coordinates $(3,5)$. The algorithm would arrive at this point from point $(2,4)$. At the end point, $b=(3,5)$ and $c=(2,5)$. The next step would be to look for the first 8-neighbor proceeding clockwise from c. That point would be $(2,4)$ again, so the algorithm would visit that point twice.

Problem 11.3

(a) We look at the Freeman chain code of a closed curve as a circular set of numbers. Selecting the starting point as the beginning of the smallest integer means that we have examined all starting possibilities and selected the one giving the smallest integer. *The* smallest integer, can only be one number, so that number is unique. Therefore, the starting point is unique.

Problem 11.4

(a) The first difference only counts the number of directions that separate adjacent elements of the code. Because the counting process is independent of direction, the first difference is independent of boundary rotation. (It is worthwhile to point out to students that the assumption here is that rotation does not change the code itself).

Problem 11.5

(a) The answer is yes, assuming that the boundary is used without subsampling. Then, because it is 4-connected, any transition from one point to the next is by definition in one of the four directions shown in Fig. 11.3(a). An SCC, applied to such a curve, would only have four possible slope changes, which could be correlated to the four Freeman directions in Fig. 11.3(a). Because the boundary is not subsampled, the separation between pixels is unity, so setting the length of the straight-line segment using SCCs would yield a code that is equivalent to the Freeman code.

Problem 11.6

With reference to Fig. 11.6(d), an accuracy of 10^{-1} means that angle increments are $0, \pm 0.1, \pm 0.2, \ldots \pm 0.9$. Because ± 1 are excluded, the preceding increments total to 19 symbols.

Problem 11.7

(a) As a square boundary is traversed there are four slope changes. In order to return to the starting point (the curve is closed and convex) the sum of the slope changes equals a complete revolution around Fig. 11.6(d). Tortuosity is equal to the sum of the absolute values of the changes. The four slope changes are equal in magnitude: $0.5 + 0.5 + |-0.5| + |-0.5| = 2$ (the order depends on where we start). So, the tortuosity is 2.

(b) If a circle is approximated by n straight-line segments of equal length, the number of slope changes as we traverse the circle is n because the circle is a closed convex curve. As in (a), returning to the starting point is one revolution around Fig. 11.6(d), so the sum of the absolute values of the slope changes must equal 2.

Problem 11.8

We give a simple, intuitive argument. Let P denote the uppermost-leftmost point (vertex) of a polygon. Let B denote the point (vertex) immediately before P as we travel the polygon in the counterclockwise direction. Similarly, let A be the point (vertex) immediately after P. The angle of vertex P is determined by the angle sustained by sequence BPA as we travel counterclockwise through the polygon.

Consider Figure P11.8 which shows three vertices of a polygon, one of which is P. Because P is assumed to be the leftmost point, there are no other points in the polygon that are to the left of it. Now consider a straight line joining points A and B. The only way the P can be the leftmost and uppermost point

simultaneously is if it lies on the left side of that line. But if it does, then the angle of the vertex at P must be convex. The same argument holds if we travel in the clockwise direction, in which case the labels A and B would be simply interchanged in the figure.

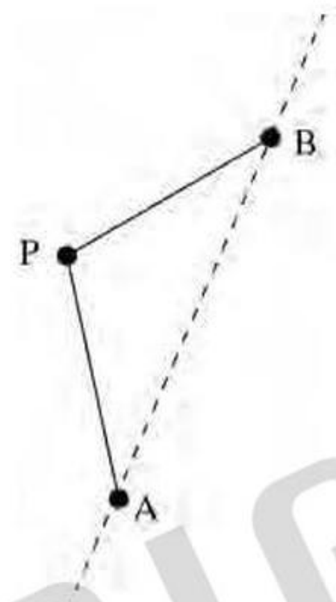


Figure P11.8

Problem 11.10

(a) The rubber-band approach forces the polygon to have vertices at every inflection of the inner and outer walls that surround the gray cell wall illustrated in Fig. 11.7. Because the vertices are joined by straight lines, and the resulting curve is convex, this produces the minimum-perimeter polygon for any given wall configuration, as defined in the discussion in the Section 11.2

Problem 11.11

(a) When the B vertices are mirrored, they coincide with the two white vertices in the corners, so they become collinear with the corner vertices. The algorithm ignores collinear vertices, so the small indentation will not be detected.

(b) When the indentation is deeper than one pixel (but still 1 pixel wide) we have the situation shown in Fig. P11.11. Note that the B vertices cross after mirroring. Referring to the bottom figure, when the algorithm gets to vertex 2, vertex 1 will be identified as a vertex of the MPP, so the algorithm is initialized at that step. Because of initialization, vertex 2 is visited again. It will be collinear with W_C and V_L, so B_C will be set at the location of vertex 2. When vertex 3 is visited, $\text{sgn}(V_L, W_C, V_3)$ will be 0, so B_C will be set at vertex 3. When vertex 4 is visited, sgn(1,3,4) sgn(1, 3,4) will be negative, so V_L will be set to vertex 3 and the algorithm is reinitialized. Because vertex 2 will never be visited again, it will never become a vertex of the MPP. The next MPP vertex to be detected will be vertex 4. Therefore, indentations 2 pixels or greater in depth and 1 pixel wide will be represented by the sequence 1•3•4 in the second figure. Thus, the algorithm solves the crossing caused by the mirroring of the two B vertices by keeping only one vertex. This is a general result for 1-pixel wide, 2-pixel (or greater) deep intrusions.

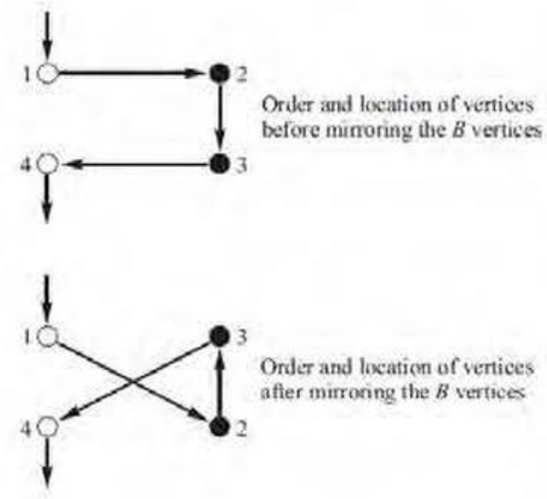


Figure P11.11

Problem 11.12

(a) The solution is shown in Fig. P11.12(a).

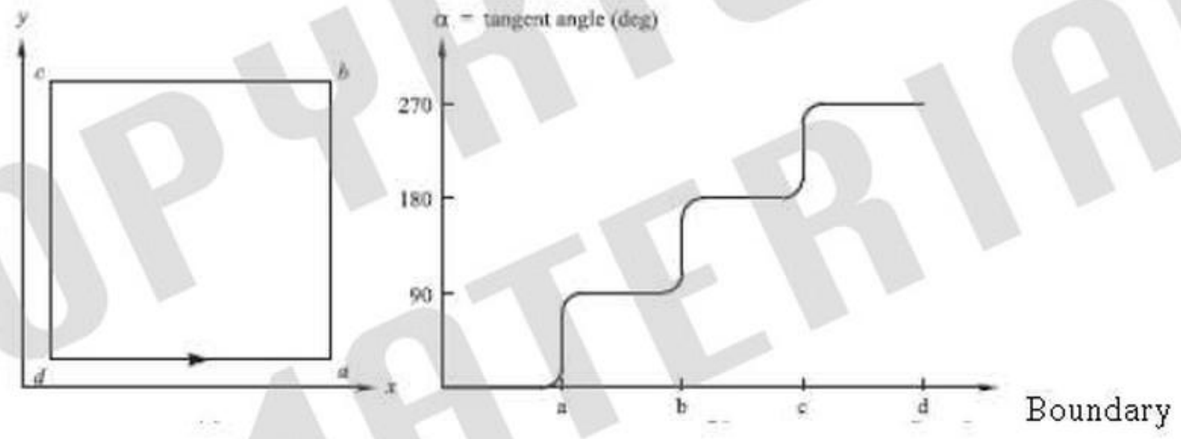


Figure P11.12(a)

Problem 11.13

(a) From Fig. P11.13(a), we see that the distance from the origin to the triangle is given by

$$
\begin{aligned}
r(\theta) &= \frac{D_0}{\cos\theta} \qquad 0^\circ \le \theta < 60^\circ \\
&= \frac{D_0}{\cos(120^\circ - \theta)} \qquad 60^\circ \le \theta < 120^\circ \\
&= \frac{D_0}{\cos(180^\circ - \theta)} \qquad 120^\circ \le \theta < 180^\circ \\
&= \frac{D_0}{\cos(240^\circ - \theta)} \qquad 180^\circ \le \theta < 240^\circ \\
&= \frac{D_0}{\cos(300^\circ - \theta)} \qquad 240^\circ \le \theta < 300^\circ \\
&= \frac{D_0}{\cos(360^\circ - \theta)} \qquad 300^\circ \le \theta < 360^\circ
\end{aligned}
$$

where D_0 is the perpendicular distance from the origin to one of the sides of the triangle, and $D = D_0/\cos(60^\circ) = 2D_0$. Once the coordinates of the vertices of the triangle are given, determining the equation of each straight line is a simple problem, and D_0 (which is the same for the three straight lines) follows from elementary geometry. The signature is shown next to the plot.

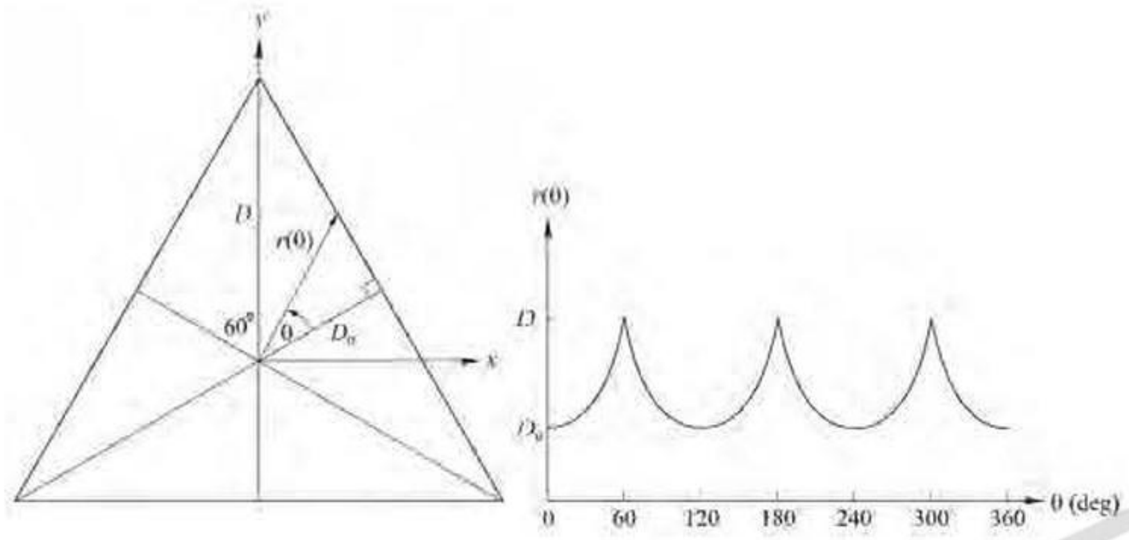


Figure P11.13(a)

Problem 11.14

(a) Threshold a waveform to isolate the peaks. In the thresholded result, look for croups on contiguous points separated by a group of zeros. Three distinct groups of nonzero values imply three peaks, and thus a rectangle. Four groups implies a square. Choose the value of the threshold at approximately 0.6 of the maximum value of the waveforms. This will guarantee that only dominant peaks are detected (see Fig. 11.11). The thresholding approach is more rugged than looking for the peaks themselves.

Problem 11.15

(a) The answer is a single point:

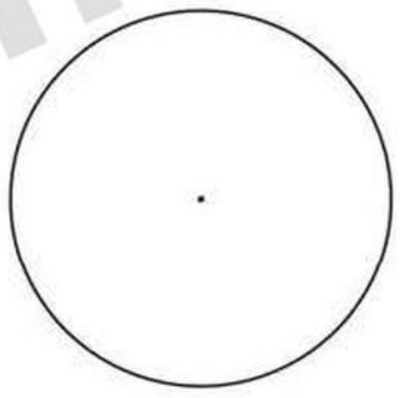


Figure P11.15(a)

(b) See Fig. P11.15(b)

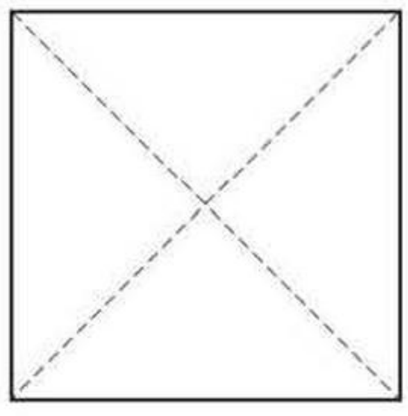


Figure P11.15(b)

Problem 11.16

(a) The number of symbols in the first difference is equal to the number of segment primitives in the boundary, so the shape order is 12.

Problem 11.17

With reference to Chapter 4, the DFT can be real only if the data sequence is conjugate symmetric. Only contours that are symmetric with respect to the origin have this property. The axis system of Fig. 11.18 would have to be set up so that this condition is satisfied for symmetric figures. This can be accomplished by placing the origin at the center of gravity of the contour.

Problem 11.19

The mean is sufficient.

Problem 11.21

This problem can be solved by two descriptors: holes and the convex deficiency (see Section 9.5 regarding the convex hull and convex deficiency of a set). The decision making process can be summarized in the form of a simple decision, as follows: If the character has two holes, it is an 8. If it has one hole it is a 0 or a 9. Otherwise, it is a 1 or an X. To differentiate between 0 and 9 we compute the convex deficiency. The presence of a "significant" deficiency (say, having an area greater than 20% of the area of a rectangle that encloses the character) signifies a 9; otherwise we classify the character as a 0. We follow a similar procedure to separate a 1 from an X. The presence of a convex deficiency with four components whose centroids are located approximately in the North, East, West, and East quadrants of the character indicates that the character is an X. Otherwise we say that the character is a 1. This is the basic approach. Implementation of this technique in a real character recognition environment has to take into account other factors such as multiple "small" components in the convex deficiency due to noise, differences in orientation, open loops, and the like. However, the material in Chapters 3, 9 and 11 provide a solid base from which to formulate solutions. In practice, recognizing such characters would most likely be done using neural networks, as discussed in Chapter 12.

Problem 11.22

(b) Normalize the matrix by dividing it by the sum of its components: 39800:

$$\begin{bmatrix} 0.4925 & 0.0050 \\ 0 & 0.5025 \end{bmatrix}$$

so, $p_{11} = 0.4925$, $p_{12} = 0.005$, $p21 = 0$, and $p_{22} = 0.5025$

Problem 11.24

(a) The image is

$$
\begin{array}{ccccc}
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0
\end{array}
$$

Let $z_1 = 0$ and $z_2 = 1$. Because there are only two intensity levels, matrix $\mathbf{G}$ is of order 2×2. Element g_{11} is the number of pixels valued 0 located one pixel to the right of a 0. By inspection, $g_{11} = 0$. Similarly, $g_{12} = 10$, $g_{21} = 10$, and $g_{22} = 0$. The total number of pixels satisfying the predicate P is 20, so the normalized co-occurrence matrix is

$$
G = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}
$$

Problem 11.25

When assigning this problem, the Instructor may wish to point the student to the review of matrices and vectors in the book website.

(a) From Eq. (11-49),

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_\mathbf{x})$$

Then,

$$
\begin{aligned}
\mathbf{m}_\mathbf{y} &= E\{\mathbf{y}\} = E\{\mathbf{A}(\mathbf{x} - \mathbf{m}_\mathbf{x})\} \\
&= \mathbf{A}\left[E\{\mathbf{x}\} - E\{\mathbf{m}_\mathbf{x}\}\right] \\
&= \mathbf{A}(\mathbf{m}_\mathbf{x} - \mathbf{m}_\mathbf{x}) \\
&= \mathbf{0}.
\end{aligned}
$$

which proves the validity of Eq. (11-50). To prove the validity of Eq. (11-51), we start with the definition of the covariance matrix in Eq. (11-48):

$$\mathbf{C}_\mathbf{y} = E\{(\mathbf{y} - \mathbf{m}_\mathbf{y})(\mathbf{y} - \mathbf{m}_\mathbf{y})^T\}$$

Because $\mathbf{m}_\mathbf{y} = \mathbf{0}$, it follows that

$$
\begin{aligned}
\mathbf{C}_\mathbf{y} &= E\{\mathbf{y}\mathbf{y}^T\} \\
&= E\{[\mathbf{A}(\mathbf{x} - \mathbf{m}_\mathbf{x})][\mathbf{A}(\mathbf{x} - \mathbf{m}_\mathbf{x})]^T\} \\
&= \mathbf{A}E\{(\mathbf{x} - \mathbf{m}_\mathbf{x})(\mathbf{x} - \mathbf{m}_\mathbf{x})^T\}\mathbf{A}^T \\
&= \mathbf{A}\mathbf{C}_\mathbf{x}\mathbf{A}^T
\end{aligned}
$$

Problem 11.26

The mean square error, given by Eq. (11-55), is the sum of the eigenvalues whose corresponding eigenvectors are not used in the transformation. In this particular case, the four smallest eigenvalues are applicable (see Table 11.6), so the mean square error is

$$e_{ms} = \sum_{j=3}^{6} \lambda_j = 1729$$

The maximum error occurs when $k = 0$ in Eq. (11-55) which then is the sum of all the eigenvalues, or 15039 in this case. Thus, the error incurred by using only the two eigenvectors corresponding to the largest eigenvalues is just 11.5 % of the total possible error.

Problem 11.29

We can compute a measure of texture using the expression

$$R(x,y) = 1 - \frac{1}{1+\sigma^2(x,y)}$$

where $\sigma^2(x,y)$ is the variance of the intensity values computed in a neighborhood of (x,y). The size of the neighborhood must be sufficiently large to contain enough samples to have a stable estimate of the mean and variance. Neighborhoods of size 7×7 or 9×9 generally are appropriate for a low-noise case such as this.

Because the variance of normal wafers is known to be 400, we can obtain a value considered normal for $R(x,y)$ by using $\sigma^2(x,y) = 400$ in the equation above. An abnormal region will have a variance of about $(50)^2 = 2500$ or higher, yielding a larger value of $R(x,y)$. The procedure then is to compute $R(x,y)$ at every point (x,y) and label that point as 0 if it is normal and 1 if it is not. At the end of this procedure we look for clusters of 1's using, for example, connected components (see Section 9.5 regarding computation of connected components). If the area (number of pixels) of any connected component exceeds 400 pixels, then we classify the sample as defective.

Problem 11.30

(a) The noisy points that are common to all three patches.

Problem 11.31

(a) Let **M** be a general 2×2 matrix:

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

We know from basic matrix theory that the eigenvalues of **M** are the solution to the quadratic equation

$$\det(\mathbf{M} - \lambda\mathbf{I}) = 0$$

where $\mathbf{I}$ is the 2×2 identity matrix. Then, using the definition of the determinant, we have

$$\begin{vmatrix} a-\lambda & b \\ c & d-\lambda \end{vmatrix} = 0 = (a-\lambda)(d-\lambda) - bc = \lambda^2 - (a+d)\lambda + (ad-bc) = 0$$

This is the equation that must be solved for λ. Using the definitions $\text{Tr}(\mathbf{M}) = a + d$ and $\det(\mathbf{M}) = (ad - bc)$, we can express the equation to solve as:

$$\lambda^2 - \text{tr}(\mathbf{M})\lambda + \det(\mathbf{M}) = 0$$

from which we get the general solution:

$$\lambda_1, \lambda_2 = \frac{\text{tr}(\mathbf{M}) \pm \sqrt{[\text{tr}(\mathbf{M})]^2 - 4\det(\mathbf{M})}}{2}$$

Problem 11.32

All the border points of R_1 in I are either 5 or the value of the boundary of the image, which is given in the problem statement to be 0. For any point p in R_1, and any point q in the boundary of R_1, Eq. (11-64),

$$\forall p \in R_1 \text{ and } \forall q \in \text{boundary}(R_1) : I(p) > I(q)$$

holds, and R_1 is indeed an extremal region.

Problem 11.34

As in the book, using G instead of the g given in the problem statement, we write that statement as

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G$$

We approximate the derivative as

$$\frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

Then,

$$\frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \approx \sigma \nabla^2 G$$

or

$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G$$

which agrees with Eq. (11-70).

Problem 11.35

(b) The last image that can be down-sampled by 2 is of size 2×2, yielding an image of size 1×1. To get from size $2^n \times 2^n$ to size 1×1 (i.e., $2^0 \times 2^0$) requires down-sampling the original image n times. The total number of different size images is therefore n. There is an octave per valid image size, so the total number of octaves is n also.

Problem 11.37

(a) From Eq. (11-71),

$$D(\mathbf{x}) = D + (\nabla D)^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$$

The terms $D, (\nabla D)$, and $\mathbf{H}$ are evaluated at the sample point, so they are constant with respect to $\mathbf{x}$. Taking the derivative with respect to $\mathbf{x}$ and setting the result to 0 gives us

$$\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}} = (\nabla D)^T + \frac{1}{2}\left[2\mathbf{x}^T\mathbf{H}\right] = 0$$

The $\hat{\mathbf{x}}$ that satisfies this equation is

$$\hat{\mathbf{x}}^T\mathbf{H} = -(\nabla D)^T$$

or

$$\hat{\mathbf{x}}^T = -(\nabla D)^T\mathbf{H}^{-1}$$

Taking the transpose of both sides of this equation we get

$$\begin{aligned}\hat{\mathbf{x}} &= -\left[(\nabla D)^T\mathbf{H}^{-1}\right]^T \\ &= -\left(\mathbf{H}^{-1}\right)^T\left((\nabla D)^T\right)^T \\ &= -\mathbf{H}^{-1}(\nabla D)\end{aligned}$$

where the third step follows from the fact that $\mathbf{H}^{-1}$ is symmetric because $\mathbf{H}$ is.

Chapter 12

Problem Solutions

Problem 12.1

(a) By inspection, the mean vectors of the three classes are, approximately, $\mathbf{m}_1 = (1.5, 0.3)^T$, $\mathbf{m}_2 = (4.3, 1.3)^T$, and $\mathbf{m}_3 = (5.5, 2.1)^T$ for the classes Iris setosa, versicolor, and virginica, respectively. The decision functions are of the form given in Eq. (12-4). Substituting the preceding values of mean vectors gives:

$$d_1(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_1 - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 = 1.5x_1 + 0.3x_2 - 1.2$$

$$d_2(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_2 - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2 = 4.3x_1 + 1.3x_2 - 10.1$$

$$d_3(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_3 - \frac{1}{2}\mathbf{m}_3^T\mathbf{m}_3 = 5.5x_1 + 2.1x_2 - 17.3$$

Problem 12.2

From the definition of the Euclidean distance,

$$D_j(\mathbf{x}) = \| \mathbf{x} - \mathbf{m}_j \| = \left[(\mathbf{x} - \mathbf{m}_j)^T (\mathbf{x} - \mathbf{m}_j) \right]^{1/2}$$

Because $D_j(\mathbf{x})$ is non-negative, choosing the smallest $D_j(\mathbf{x})$ is the same as choosing the smallest $D_j^2(\mathbf{x})$ in terms of classification, where

$$\begin{aligned} D_j^2(\mathbf{x}) &= \| \mathbf{x} - \mathbf{m}_j \|^2 = (\mathbf{x} - \mathbf{m}_j)^T (\mathbf{x} - \mathbf{m}_j) \\ &= \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{m}_j + \mathbf{m}_j^T\mathbf{m}j \\ &= \mathbf{x}^T\mathbf{x} - 2\left(\mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \right) \end{aligned}$$

We note that the term $\mathbf{x}^T\mathbf{x}$ is independent of j (that is, it is a constant with respect to j in $D_j^2(\mathbf{x})$). Thus, choosing the minimum of $D_j^2(\mathbf{x})$ is equivalent to choosing the maximum of $\mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j$. .

Problem 12.4

Figure P12.4 shows the solution, where the x's are treated as voltages and the Y's denote impedances. From basic circuit theory, the currents, I's, are the products of the voltages times the impedances. As the figure

shows, these products can be expressed in the same form as the decision functions of the minimum distance classifier. Implementing one resistor bank per decision function and selecting the maximum result is precisely what a minimum-distance classifier does.

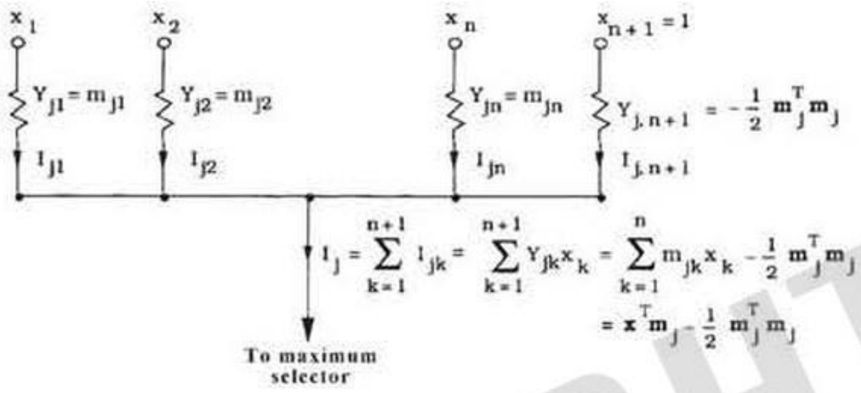


Figure P12.4

Problem 12.5

Assume that the template, w, is of size $J \times K$. For any value of displacement (x, y), we can express both the area of the image under the template, and the template itself , in vector form by letting the first row of the subimage under the template represent the first K elements of a column vector $\mathbf{a}$, the elements of next row the next K elements of $\mathbf{a}$, and so on. At the end of the procedure we subtract the average value of the intensity levels in the subimage from every element of $\mathbf{a}$. The vector $\mathbf{a}$ is of size $JK \times 1$. A similar approach yields a vector, $\boldsymbol{w}$ of the same size as $\mathbf{a}$, for the template w minus its average. (Yypically the template is much smaller than the image. We make it the same size by padding it with 0's). Vector $\boldsymbol{w}$ does not change as (x, y) varies because the coefficients of the template are constants. With this construction in mind, we see that the numerator of Eq. (12-10) is simply the vector inner-product $\mathbf{a}^T\boldsymbol{w}$. Similarly, the first term in the denominator is the norm squared of $\mathbf{a}$, denoted by $\mathbf{a}^T\mathbf{a} = \|\mathbf{a}\|^2$, while the second term has a similar interpretation for $\boldsymbol{w}$. The correlation coefficient then becomes

$$\gamma(x, y) = \frac{\mathbf{a}^T\boldsymbol{w}}{\left[\|\mathbf{a}\|\|\boldsymbol{w}\|\right]^{1/2}}$$

When $\mathbf{a} = \boldsymbol{w}$ (a perfect match), $\gamma(x, y) = \|a\|^2 / \|a\|\|a\| = 1$, which is the maximum value obtainable by the above expression. Similarly, the minimum value occurs when $\mathbf{a} = -\boldsymbol{w}$, in which case, $\gamma(x, y) = 1$. Thus, although the vector $\mathbf{a}$ varies in general for every value of (x, y), the values of $\gamma(x, y)$ are all in the range $[-1,1]$.

Problem 12.7

$Q = 0$ implies that $\max(|A|,|B|) = M$. Suppose that $|A| > |B|$. Then, it must follow that $|A| = M$ and, therefore, that $M = |B|$. But M is obtained by matching A and B, so it must be bounded by

$M \leq \min(|A|,|B|)$. Because we have stipulated that $|A| > |B|$, the condition $M \leq \min(|A|,|B|)$ implies that $M \leq |B|$. But this contradicts the above result, so the only way for $\max(|A|,|B|) = M$ to hold is if $|A| = |B|$. This, in turn, implies that A and B must be identical strings $(A \equiv B)$ because $|A| = |B| = M$ means that all symbols of A and B match. The converse result that if $A \equiv B$ then $Q = 0$ follows directly from the definition of Q.

Problem 12.9

(a) Because it is given that the pattern classes are governed by Gaussian densities, only knowledge of the mean vector and covariance matrix of each class are needed to specify the Bayes classifier. Substituting the given patterns into Eqs. (12-29) and (12-30) we obtain

$$\mathbf{m}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{m}_2 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{C}_1^{-1}$$

and

$$\mathbf{C}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{C}_2^{-1} = \mathbf{C}_1^{-1}$$

Because $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{I}$, the decision functions are the same as those of a minimum-distance classifier:

$$d_1(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_1 - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 = 1.0x_1 + 1.0x_2 - 1.0$$

$$d_2(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_2 - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2 = 5.0x_1 + 5.0x_2 - 25$$

The decision boundary is given by the equation $d(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x}) = 0$, or

$$d(\mathbf{x}) = -4x_1 - 4x_2 + 24 = 0$$

(b) Figure P12.9 shows this boundary in pattern space.

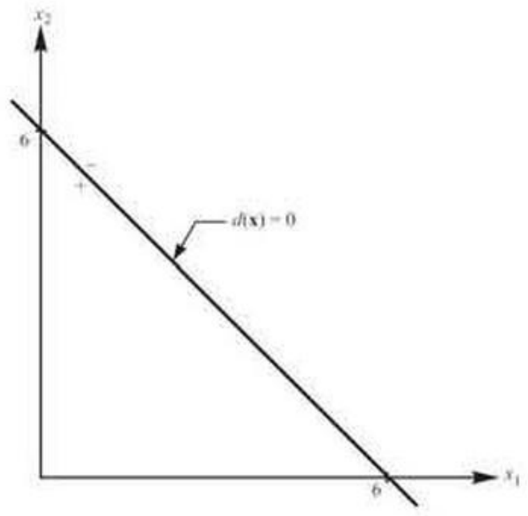


Figure p12.9

Problem 12.12

From basic probability theory,

$$p(\text{correct}) = \sum_{\mathbf{x}} p(\text{correct}/\mathbf{x})p(\mathbf{x})$$

For any pattern belonging to class c_j, $p(\text{correct}/\mathbf{x}) = p(c_j/\mathbf{x})$. Therefore,

$$p(\text{correct}) = \sum_{\mathbf{x}} p(c_j/\mathbf{x})p(\mathbf{x})$$

Substituting into this equation the formula $p(c_j/\mathbf{x}) = p(\mathbf{x}/c_j)\,p(c_j)/p(\mathbf{x})$ gives

$$p(\text{correct}) = \sum_{\mathbf{x}} p(\mathbf{x}/c_j)p(c_j)$$

Because the argument of the summation is positive, $p(\text{correct})$ is maximized by maximizing $p(\mathbf{x}/c_j)p(c_j)$ for each j. That is, if for any pattern $\mathbf{x}$ we compute $p(\mathbf{x}/c_j)p(c_j)$ for $j = 1, 2, \ldots, N_c$, and use the largest value each time as the basis for selecting the class from which $\mathbf{x}$ came, then $p(\text{correct})$ will be maximized. Because $p(\text{error}) = 1 - p(\text{correct})$, the probability of error is minimized by this procedure.

Problem 12.14

We begin by taking the partial derivative of J with respect to $\boldsymbol{w}$:

$$\frac{\partial J}{\partial \boldsymbol{w}} = \frac{1}{2}\left[\mathbf{y}\,\text{sgn}(\boldsymbol{w}^T\mathbf{y}) - \mathbf{y}\right]$$

where, by definition, $\text{sgn}(\boldsymbol{w}^T\mathbf{y}) = 1$ if $\boldsymbol{w}^T\mathbf{y} > 0$, and $\text{sgn}(\boldsymbol{w}^T\mathbf{y}) = -1$ otherwise. Substituting the partial derivative into the general equation given in the problem statement gives

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \frac{\alpha}{2}\left\{\mathbf{y}(k) - \mathbf{y}(k)\text{sgn}\left[\boldsymbol{w}(k)^T\mathbf{y}(k)\right]\right\}$$

where $\mathbf{y}(k)$ is the augmented training pattern being processes at the kth iterative step. Substituting the definition of the sgn function into this result yields,.

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \alpha\begin{cases}\mathbf{0} & \text{if } \boldsymbol{w}(k)^T\mathbf{y}(k) > 0 \\ \mathbf{y}(k) & \text{otherwise}\end{cases}$$

where $\alpha > 0$ and $\boldsymbol{w}(1)$ is arbitrary. This result agrees with the expression given in the problem statement.

Problem 12.15

Let the set of augmented training patterns be denoted by $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$. It is assumed that the training patterns of class c_2 have been multiplied by -1. If the classes are linearly separable, we want to prove that the perceptron training algorithm yields a solution weight vector, $\boldsymbol{w}^*$, with the property

$$\boldsymbol{w}^{*T}\mathbf{y}_i > T_0 \tag{1}$$

for $i = 1, 2, \ldots, N$, where T_0 is a nonnegative threshold. As stated in the problem statement, the perceptron training algorithm (with $\alpha = 1$) is stated as

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \begin{cases}\mathbf{0} & \text{if } \boldsymbol{w}(k)^T\mathbf{y}(k) > T_0 \\ \mathbf{y}(k) & \text{otherwise}\end{cases} \tag{2}$$

Suppose that we retain only the values of k for which a correction takes place (these are the only indices of interest). Then, re-adapting the index notation, we may write

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mathbf{y}_j(k) \tag{3}$$

and

$$\boldsymbol{w}(k)^T\mathbf{y}_j(k) \le T_0 \tag{4}$$

where the subscript j refers to the pattern vectors involved in the corrections. Convergence means that for some finite value, k_m,

$$\boldsymbol{w}(k_m) = \boldsymbol{w}(k_m+1) = \boldsymbol{w}(k_m+2) = \cdots$$

We prove convergence by proving that k_m is finite.

The proof of convergence begins by noting from Eq. (3) that,

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(1) + y_j(1) + y_j(2) + \cdots + y_j(k) \tag{5}$$

Keep in mind that we redefined the iterative step values to those values in which corrections were make. Thus, the values $1, 2, \ldots, k$ are values that refer to instances where corrections were made. Taking the inner product of the solution weight vector with both sides of the preceding equation we get

$$\boldsymbol{w}^T(k+1)\boldsymbol{w}^* = \boldsymbol{w}^T(1)\boldsymbol{w}^* + \mathbf{y}_j{}^T(1)\boldsymbol{w}^* + \mathbf{y}_j{}^T(2)\boldsymbol{w}^* + \cdots + \mathbf{y}_j{}^T(k)\boldsymbol{w}^* \tag{6}$$

Each term $\mathbf{y}_j{}^T(i)\boldsymbol{w}^*$ is less then or equal to T_0, so

$$\boldsymbol{w}^T(k+1)\boldsymbol{w}^* \geq \boldsymbol{w}^T(1)\boldsymbol{w}^* + kT_0 \tag{7}$$

Using the Cauchy-Schwartz inequality, $\| \mathbf{a} \|^2 \| \mathbf{b} \|^2 \geq (\mathbf{a}^T\mathbf{b})^2$ results in the expression

$$\left[\boldsymbol{w}^T(k+1)\boldsymbol{w}^*\right]^2 \leq \| \boldsymbol{w}^T(k+1) \|^2 \| \boldsymbol{w}^* \|^2 \tag{8}$$

or,

$$\| \boldsymbol{w}^T(k+1) \|^2 \geq \frac{\left[\boldsymbol{w}^T(k+1)\boldsymbol{w}^*\right]^2}{\| \boldsymbol{w}^* \|^2} \tag{9}$$

Substituting Eq. (7) into Eq. (9) we obtain

$$\| \boldsymbol{w}^T(k+1) \|^2 \geq \frac{\left[\boldsymbol{w}^T(1)\boldsymbol{w}^* + kT_0\right]^2}{\| \boldsymbol{w}^* \|^2} \tag{10}$$

Another line of reasoning leads to a contradiction regarding $\| \boldsymbol{w}^T(k+1) \|^2$. From Eq. (3),

$$\| \boldsymbol{w}(\ell+1) \|^2 = \| \boldsymbol{w}(\ell) \|^2 + 2\boldsymbol{w}^T(\ell)\mathbf{y}_j(\ell) + \| \mathbf{y}_j(\ell) \|^2 \tag{11}$$

or

$$\| \boldsymbol{w}(\ell+1) \|^2 - \| \boldsymbol{w}(\ell) \|^2 = 2\boldsymbol{w}^T(\ell)\mathbf{y}_j(\ell) + \| \mathbf{y}_j(\ell) \|^2 \tag{12}$$

Let $Q = \max_j\left(\| \mathbf{y}_j(\ell) \|^2\right)$. Then, because $\boldsymbol{w}(\ell)^T\mathbf{y}_j(\ell) \leq T_0$,

$$\| \boldsymbol{w}(\ell+1) \|^2 - \| \boldsymbol{w}(\ell) \|^2 \leq 2T_0 + Q \tag{13}$$

Adding these inequalities for $\ell = 1,2,\ldots,k$ yields

$$\| \boldsymbol{w}(\ell+1) \|^2 \leq \| \boldsymbol{w}(1) \|^2 + [2T_0 + Q]k \tag{14}$$

This inequality establishes a bound on $\| \boldsymbol{w}(\ell+1) \|^2$ that conflicts for sufficiently large k with the bound in Eq. (10). In fact, k can be no larger that k_m, which is a solution to the equation

$$\frac{\left[\boldsymbol{w}^T(1)\boldsymbol{w}^* + k_m T_0\right]^2}{\| \boldsymbol{w}^* \|^2} = \| \boldsymbol{w}(1) \|^2 + [2T_0 + Q]k_m \tag{14}$$

This equation indicates that k_m is finite, thus proving that the perceptron training algorithm converges in a finite number of steps to a solution weight vector $\boldsymbol{w}^*$ if the pattern classes are linearly separable.

The special case $T_0 = 0$ is proved in a slightly different way. In this case, we Eq. (7) becomes

$$\boldsymbol{w}^T(k+1)\boldsymbol{w}^* \geq \boldsymbol{w}^T(1)\boldsymbol{w}^* + ka$$

where

$$a = \min_j \left[\mathbf{y}_j^T(\ell)\boldsymbol{w}^*\right]$$

Because $\boldsymbol{w}^*$ is a solution weight vector by hypothesis, we know that $\mathbf{y}_j^T(\ell)\boldsymbol{w}^* > 0$. Also, because $\boldsymbol{w}^T(\ell)\mathbf{y}(\ell) \leq (T_0 = 0)$,

$$\begin{aligned} \| \boldsymbol{w}(\ell+1) \|^2 - \| \boldsymbol{w}(\ell) \|^2 &\leq \| \mathbf{y}_j(\ell) \|^2 \\ &\leq Q \end{aligned}$$

The rest of the proof remains the same. The bound on the number of iterative steps before a solution is the value of k_m that satisfies the following equation

$$\frac{\left[\boldsymbol{w}^T(1)\boldsymbol{w}^* + k_m T_0\right]^2}{\| \boldsymbol{w}^* \|^2} = \| \boldsymbol{w}(1) \|^2 + Qk_m$$

Problem 12.16

(c) $h(z) = \max(0, z)$

The function is not continuous at $z = 0$, so the derivative is not defined there. But we handle this by *defining* the derivative at that point to be 0. Then,

$$h'(z) = \frac{d}{dz}\max(0, z) = \begin{cases} \frac{d}{dz}(z) & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$
$$= \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

Problem 12.17

The single decision function that implements a minimum-distance classifier for two classes in n-dimensional space has the form

$$d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x}) = (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{x} - \frac{1}{2}(\mathbf{m}_1{}^T \mathbf{m}_1 - \mathbf{m}_2{}^T \mathbf{m}_2)$$

A single neuron with n inputs performs the computation

$$z = \sum_{k=1}^{n} w_k a_k + b$$
$$= \boldsymbol{w}^T \mathbf{a} + b$$

As you can see, this is exactly the same form as above, so out neural network is a two-layer network: an input layer whose values are the value of the input vector $\mathbf{x}$, (so that $a_k = x_k$) and an output layer consisting of a single neuron with weights

$$w_k = m_{1k} - m_{2k}; \quad k = 1, 2, \ldots, n$$

and bias

$$b = -\frac{1}{2}(\mathbf{m}_1{}^T \mathbf{m}_1 - \mathbf{m}_2{}^T \mathbf{m}_2)$$

The network has no hidden layers. If we use a tanh activation function, after the weights and bias are learned via training, this neural network will have an output > 0 for patterns of one class and < 0 for patterns of the other class. This is exactly how a minimum distance classifier with a single decision function would behave for two pattern classes that are linearly separable. Because they are tightly grouped, we assume each mean vector is a good representation of one of the classes.

Problem 12.20

(a) When $P(c_i) = P(c_j)$ and $\mathbf{C} = \mathbf{I}$.

Problem 12.22

The answer is no. The reason is that when we do the first forward pass all outputs will be zero. The first pass of backpropagation would start with error vectors formed from the actual outputs and the desired outputs.

So the error propagated back would not be all zeros. However, the weights would not be corrected until the backward step is completed, so we would be multiplying our error vectors by the initial zero weights and biases, giving zero results for the weight corrections at the end of the backward pass. This cycle without changes would then repeat itself, resulting in a meaningless training session.

Problem 12.24

(c) Choosing large weights can result in correspondingly large values of the intermediate computations, $\mathbf{z}(\ell)$. If a sigmoid activation function is used, the large values in the elements of $\mathbf{z}$ would cause $h(z)$ to become 1. But the outputs of a layer are precisely these $h(z)$, which means that all layers, including the last would output just 1's. The conclusion is that the values of weights learned by the system should be fractions. A way to achieve this is to start training with small values.

Problem 12.25

(a)

$$\begin{aligned}
\delta_j(\ell) &= \frac{\partial E}{\partial z_j(\ell)} = \sum_i \frac{\partial E}{\partial z_i(\ell+1)} \frac{\partial z_i(\ell+1)}{\partial z_j(\ell)} \\
&= \sum_i \delta_i(\ell+1) \frac{\partial}{\partial z_j(\ell)} \left[\sum_j w_{ij}(\ell+1) a_j(\ell) + b_i(\ell+1) \right] \\
&= \sum_i \delta_i(\ell+1) \frac{\partial}{\partial z_j(\ell)} \left[\sum_j w_{ij}(\ell+1) h\big(z_j(\ell)\big) + b_i(\ell+1) \right] \\
&= \sum_i \delta_i(\ell+1) w_{ij}(\ell+1) h'\big(z_j(\ell)\big) \\
&= h'\big(z_j(\ell)\big) \sum_i w_{ij}(\ell+1) \delta_i(\ell+1)
\end{aligned}$$

Problem 12.27

Operator $\odot$ defines elementwise multiplication, and the dimensions of $\mathbf{Z}^\ell$ we know are $n_\ell \times n_p$. It then follows that the dimensions of $\mathbf{D}^\ell$ also must be $n_\ell \times n_p$, assuming that Eq.(12-79) is dimensionally correct, which means that term $\left([\mathbf{W}^{\ell+1}]^T \mathbf{D}^{\ell+1}\right)$ must be of size $n_\ell \times n_p$. Let's see if this is true. We know from forward propagation that matrix $\mathbf{W}^{\ell+1}$ has dimensions $n_{\ell+1} \times n_\ell$, so $[\mathbf{W}^{\ell+1}]^T$ is of size $n_\ell \times n_{\ell+1}$. Each column of matrix $\mathbf{D}^{\ell+1}$ is a vector $\boldsymbol{\delta}$ containing the errors of the neurons in that layer for one of the input pattern vectors. There are $n_{\ell+1}$ neurons in layer $\ell+1$ and we are stuffing the results of every training pattern in a matrix. Since there are n_p patterns being processed, $\mathbf{D}^{\ell+1}$ is of size $n_{\ell+1} \times n_p$. Therefore the dimensions of the product $\left([\mathbf{W}^{\ell+1}]^T \mathbf{D}^{\ell+1}\right)$ are $n_\ell \times n_p$. This agrees with the dimension of $\mathbf{Z}^\ell$ so $\mathbf{D}^\ell$ is indeed of size $n_\ell \times n_p$.

Problem 12.29

Table P1329. Fully-connected multilayer network training algorithm for one pattern vector. One complete iteration through all pattern vectors in the training set constitutes one epoch of training. To initialize, specify a set weights consisting of small random numbers.

Step	Description	Equations
Step 1	Present the training pattern vectors.	$\mathbf{a}(1) = \mathbf{x}$
Step 2	Feedforward	For $\ell = 2, \ldots, L$, compute $\mathbf{z}(\ell) = \mathbf{W}(\ell)\mathbf{a}(\ell-1) + \mathbf{b}(\ell)$, $\mathbf{a}(\ell) = h(\mathbf{z}(\ell))$, and $h'(\mathbf{z}(\ell))$
Step 3	Compute output error	$\boldsymbol{\delta}(L) = \mathbf{r} - \mathbf{a}(L)$
Step 4	Backpropagate the error	For $\ell = L-1, L-2 \ldots, 2$, compute $\boldsymbol{\delta}(\ell) = \left([\mathbf{W}(\ell+1)]^T \boldsymbol{\delta}(\ell+1)\right) \odot h'(\mathbf{z}(\ell))$
Step 5	Update the weights	For $\ell = L, L-1, L-2 \ldots, 2$, let $\mathbf{W}(\ell) = \mathbf{W}(\ell) - \alpha\boldsymbol{\delta}(\ell)(\mathbf{a}(\ell-1))^T$ and $\mathbf{b}(\ell) = \mathbf{b}(\ell) - \alpha\boldsymbol{\delta}(\ell)$

The details of the derivation of the expressions in the table are as follows

We look at the forward process first. Consider a fully-connected neural network with layers $\ell = 1, 2, \ldots, L$, where $\ell = 1$ is the input layer, $\ell = L$ is the output layer, and n_ℓ is the number of neurons in layer ℓ. Let $\mathbf{x}$ denote an n-dimensional pattern vector, and. Let $\mathbf{a}(\ell)$ a vector containing the activation values of all n_ℓ neurons in layer ℓ. By definition the outputs of the first layer are the components of the pattern vector: $\mathbf{a}(1) = \mathbf{x}$, and $\mathbf{a}(L)$ contains the outputs of all the neurons in the output layer. Vector $\mathbf{x}$ is of dimension $n \times 1$ and $\mathbf{a}(\ell)$ is of dimension $n_\ell \times 1$. Let $\mathbf{W}(\ell)$ be a weight matrix of size $n_\ell \times n$ whose rows are the weights of the neurons in layer ℓ. Finally, let $\mathbf{b}(\ell)$ be an $n_\ell \times 1$ vector whose components are the bias values of the neurons in layer ℓ. Using this notation, we can write Eq. (12-54) as

$$\mathbf{z}(\ell) = \mathbf{W}(\ell)\mathbf{a}(\ell) + \mathbf{b}(\ell) \tag{1}$$

It then follows from Eq. (12-55) that

$$\mathbf{a}(\ell) = h(\mathbf{z}(\ell)) \tag{2}$$

At this step, we also compute $h'(\mathbf{z}(\ell))$ for use later in backpropagation. Because we are assuming that activation function h is the same for all neurons, the kth component of vector $\mathbf{a}(\ell)$ is $a_k(\ell) = h(z_k(\ell))$. Starting with $\mathbf{a}(1) = \mathbf{x}$, these equation are applied for $\ell = 2, \ldots, L$ to obtain $\mathbf{a}(\ell)$, a vector of dimension $n_L \times 1$ whose elements are the outputs of the neural network. Each output in this layer is associated with one of the pattern classes that the network is capable of classifying.

The activation of neuron j in the output layer is $a_j(L)$. We define the error of that neuron as

$$E_j = \frac{1}{2}(r_j - a_j(L))^2 \tag{3}$$

$j = 1, 2, \ldots, n_L$, where r_j is the desired response of output neuron $a_j(L)$ for the given $\mathbf{x}$. The output error with respect to a single $\mathbf{x}$ is the sum of the errors of all output neurons with respect to that vector:

$$\begin{aligned} E &= \sum_{j=1}^{n_L} E_j = \frac{1}{2}\sum_{j=1}^{n_L}\left(r_j - a_j(L)\right)^2 \\ &= \frac{1}{2}\|\mathbf{r} - \mathbf{a}(L)\|^2 \end{aligned} \tag{4}$$

where the second line follows from the definition of the Euclidean vector norm. The *total network output error* over all training patterns is defined as the sum of the errors of the individual patterns.

The vector for the error at the output layer is

$$\boldsymbol{\delta}(L) = \mathbf{r} - \mathbf{a}(L) \tag{5}$$

where $\mathbf{r}$ is the target vector for pattern vector $\mathbf{x}$ (remember, the elements of $\mathbf{r}$ are 0, except in the location corresponding to the class of pattern $\mathbf{x}$, where it is 1).

The backpropagation error at one neuron location is given by

$$\delta_j(\ell) = h'\left(z_j(\ell)\right)\sum_{i=1}^{n_{\ell+1}} w_{ij}(\ell+1)\delta_i(\ell+1) \tag{6}$$

Note that the j in the summation is constant, so variable i spans a complete *column* of weight matrix $\mathbf{W}(\ell+1)$, which we denote by $\mathbf{w}_j(\ell+1)$. But the summation is simply the dot product of this column *transposed* and vector $\boldsymbol{\delta}(\ell+1)$:

$$\sum_{i=1}^{n_{\ell+1}} w_{ij}(\ell+1)\delta_i(\ell+1) = [\mathbf{w}_j(\ell+1)]^T\boldsymbol{\delta}\ (\ell+1) \tag{7}$$

which is a scalar. So we now write Eq. (6) as

$$\delta_j(\ell) = [\mathbf{w}_j(\ell+1)]^T\boldsymbol{\delta}\ (\ell+1)\, h'\left(z_j(\ell)\right) \tag{8}$$

where we reversed the order of the two scalars in preparation for the next step, which is to write Eq. (8) in vector form as

$$\boldsymbol{\delta}(\ell) = \left([\mathbf{W}(\ell+1)]^T\boldsymbol{\delta}(\ell+1)\right)\odot\mathbf{h}'(\mathbf{z}(\ell)) \tag{9}$$

As defined in Chapter 2, the symbol $\odot$ means element-wise multiplication of the two vectors.

In scalar form, the weight update equations are

$$\omega_{ij} \rightarrow \omega_{ij} - \eta\,\delta_i a_j \tag{10}$$

and

$$b_j \rightarrow b_j - \eta \delta_j \tag{11}$$

The last step is to write these equations in vector form:

$$\mathbf{W}(\ell) = \mathbf{W}(\ell) - \alpha \boldsymbol{\delta}(\ell)\left(\mathbf{a}(\ell-1)\right)^T \tag{12}$$

and

$$\mathbf{b}(\ell) = \mathbf{b}(\ell) - \alpha \boldsymbol{\delta}(\ell) \tag{13}$$

These expressions are for correcting the weights after each pattern is fed forward through the network. Another approach used often to obtain smoother estimates is to feed all the patterns through the network, compute the average the responses, and do backpropagation using the average, as follows;

$$\mathbf{W}(\ell) \rightarrow \mathbf{W}(\ell) - \frac{\alpha}{n_p}\sum_{\mathbf{x}} \boldsymbol{\delta}_{\mathbf{x}}(\ell)\left(\mathbf{a}_{\mathbf{x}}(\ell-1)\right)^T \tag{14}$$

and

$$\mathbf{b}(\ell) \rightarrow \mathbf{b}(\ell) - \frac{\alpha}{n_p}\sum_{\mathbf{x}} \boldsymbol{\delta}_{\mathbf{x}}(\ell) \tag{15}$$

where n_p is the number of patterns used for training, the summation is over the training patterns, and the subscript indicates that that we are now averaging over all patterns before doing the backpropagation step.

Problem 12.30

(a) If the kernels are square, $w \times w$, of odd size, and the square dimensions of the convolution planes is 504, then $(512 - 2 \cdot (w-1)/2) = 504$, so the spatial dimensions of the kernel are 9×9.

Problem 12.31

(b) The number of neurons has to be an integer. The equation gives fractional values only if one or more of its parameters is not specified properly. For example, if $V = 28$, $P = 0$, and $F = 5$, a stride of 1 would give $N = 24$, so all specifications are valid. However, using a stride of $S = 2$ would give a fractional value for N, so a stride of 2 cannot be used with the values of V, P, and F. One or more of those values would have to change for us to be able to use a stride of 2.

Problem 12.32

Use the chain rule:

$$\frac{\partial E}{\partial b(\ell)} = \sum_x \sum_y \frac{\partial E}{\partial z_{x,y}(\ell)} \frac{\partial z_{x,y}(\ell)}{\partial b(\ell)} = \sum_x \sum_y \delta_{x,y}(\ell)$$

where we used the definition $\delta_{x,y}(\ell) = \partial E / \partial z_{x,y}(\ell)$ and, from the definition of $z_{x,y}(\ell)$, the fact that the partial derivative of this term with respect to $b(\ell)$ is 1.