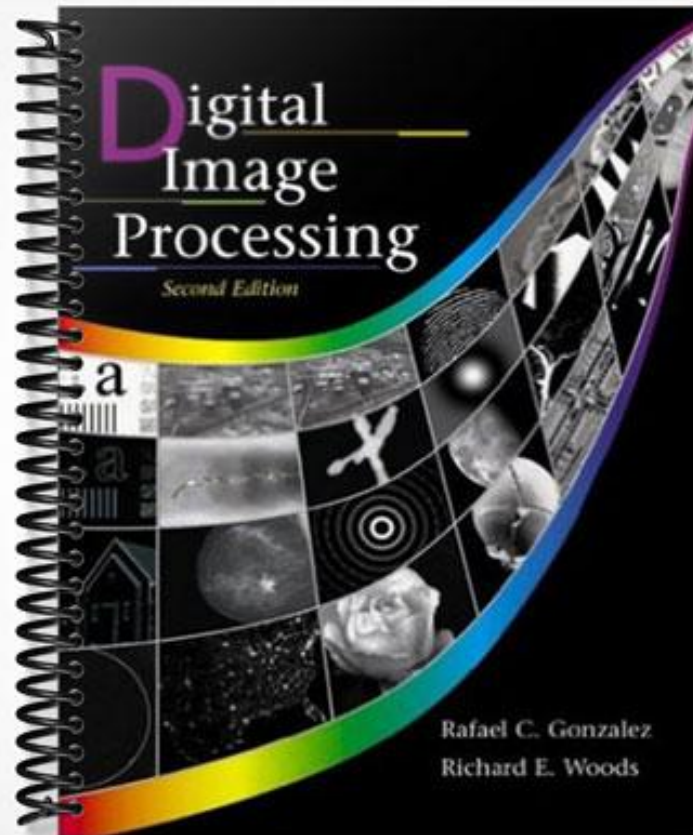


SOLUTIONS MANUAL



The background of the cover is a grayscale image of a globe with a grid overlay, representing digital image processing. The globe is tilted and the grid lines are visible across its surface.

Digital Image Processing

Second Edition

Instructor's Manual

Rafael C. Gonzalez
Richard E. Woods

Digital Image Processing

Second Edition

Instructor's Manual

Rafael C. Gonzalez

Richard E. Woods

Prentice Hall

Upper Saddle River, NJ 07458

www.prenhall.com/gonzalezwoods

or

www.imageprocessingbook.com

Revision history

10 9 8 7 6 5 4 3 2 1

Copyright ©1992-2002 by Rafael C. Gonzalez and Richard E. Woods

Preface

This manual contains detailed solutions to all problems in *Digital Image Processing*, 2nd Edition. We also include a suggested set of guidelines for using the book, and discuss the use of computer projects designed to promote a deeper understanding of the subject matter. The notation used throughout this manual corresponds to the notation used in the text.

The decision of what material to cover in a course rests with the instructor, and it depends on the purpose of the course and the background of the students. We have found that the course outlines suggested here can be covered comfortably in the time frames indicated when the course is being taught in an electrical engineering or computer science curriculum. In each case, no prior exposure to image processing is assumed. We give suggested guidelines for one-semester courses at the senior and first-year graduate levels. It is possible to cover most of the book in a two-semester graduate sequence.

The book was completely revised in this edition, with the purpose not only of updating the material, but just as important, making the book a better teaching aid. To this end, the instructor will find the new organization to be much more flexible and better illustrated. Although the book is self contained, we recommend use of the companion web site, where the student will find detailed solutions to the problems marked with a star in the text, review material, suggested projects, and images from the book. One of the principal reasons for creating the web site was to free the instructor from having to prepare materials and handouts beyond what is required to teach from the book.

Computer projects such as those described in the web site are an important part of a course on image processing. These projects give the student hands-on experience with algorithm implementation and reinforce the material covered in the classroom. The projects suggested at the web site can be implemented on almost any reasonably-equipped multi-user or personal computer having a hard copy output device.

1 Introduction

The purpose of this chapter is to present suggested guidelines for teaching material from this book at the senior and first-year graduate level. We also discuss use of the book web site. Although the book is totally self-contained, the web site offers, among other things, complementary review material and computer projects that can be assigned in conjunction with classroom work. Detailed solutions to all problems in the book also are included in the remaining chapters of this manual.

Teaching Features of the Book

Undergraduate programs that offer digital image processing typically limit coverage to one semester. Graduate programs vary, and can include one or two semesters of the material. In the following discussion we give general guidelines for a one-semester senior course, a one-semester graduate course, and a full-year course of study covering two semesters. We assume a 15-week program per semester with three lectures per week. In order to provide flexibility for exams and review sessions, the guidelines discussed in the following sections are based on forty, 50-minute lectures per semester. The background assumed on the part of the student is senior-level preparation in mathematical analysis, matrix theory, probability, and computer programming.

The suggested teaching guidelines are presented in terms of general objectives, and not as time schedules. There is so much variety in the way image processing material is taught that it makes little sense to attempt a breakdown of the material by class period. In particular, the organization of the present edition of the book is such that it makes it much easier than before to adopt significantly different teaching strategies, depending on course objectives and student background. For example, it is possible with the new organization to offer a course that emphasizes spatial techniques and covers little or no transform material. This is not something we recommend, but it is an option that often is attractive in programs that place little emphasis on the signal processing aspects of the field and prefer to focus more on the implementation of spatial techniques.

2 Chapter 1 Introduction

The companion web site

www.prenhall.com/gonzalezwoods

or

www.imageprocessingbook.com

is a valuable teaching aid, in the sense that it includes material that previously was covered in class. In particular, the review material on probability, matrices, vectors, and linear systems, was prepared using the same notation as in the book, and is focused on areas that are directly relevant to discussions in the text. This allows the instructor to assign the material as independent reading, and spend no more than one total lecture period reviewing those subjects. Another major feature is the set of solutions to problems marked with a star in the book. These solutions are quite detailed, and were prepared with the idea of using them as teaching support. The on-line availability of projects and digital images frees the instructor from having to prepare experiments, data, and handouts for students. The fact that most of the images in the book are available for downloading further enhances the value of the web site as a teaching resource.

One Semester Senior Course

A basic strategy in teaching a senior course is to focus on aspects of image processing in which both the inputs and outputs of those processes are images. In the scope of a senior course, this usually means the material contained in Chapters 1 through 6. Depending on instructor preferences, wavelets (Chapter 7) usually are beyond the scope of coverage in a typical senior curriculum). However, we recommend covering at least some material on image compression (Chapter 8) as outlined below.

We have found in more than two decades of teaching this material to seniors in electrical engineering, computer science, and other technical disciplines, that one of the keys to success is to spend at least one lecture on motivation and the equivalent of one lecture on review of background material, as the need arises. The motivational material is provided in the numerous application areas discussed in Chapter 1. This chapter was totally rewritten with this objective in mind. Some of this material can be covered in class and the rest assigned as independent reading. Background review should cover probability theory (of one random variable) before histogram processing (Section 3.3). A brief review of vectors and matrices may be required later, depending on the material covered. The review material included in the book web site was designed for just this purpose.

Chapter 2 should be covered in its entirety. Some of the material (such as parts of Sections 2.1 and 2.3) can be assigned as independent reading, but a detailed explanation of Sections 2.4 through 2.6 is time well spent.

Chapter 3 serves two principal purposes. It covers image enhancement (a topic of significant appeal to the beginning student) and it introduces a host of basic spatial processing tools used throughout the book. For a senior course, we recommend coverage of Sections 3.2.1 through 3.2.2; Section 3.3.1; Section 3.4; Section 3.5; Section 3.6; Section 3.7.1, 3.7.2 (through Example 3.11), and 3.7.3. Section 3.8 can be assigned as independent reading, depending on time.

Chapter 4 also discusses enhancement, but from a frequency-domain point of view. The instructor has significant flexibility here. As mentioned earlier, it is possible to skip the chapter altogether, but this will typically preclude meaningful coverage of other areas based on the Fourier transform (such as filtering and restoration). The key in covering the frequency domain is to get to the convolution theorem and thus develop a tie between the frequency and spatial domains. All this material is presented in very readable form in Section 4.2. “Light” coverage of frequency-domain concepts can be based on discussing all the material through this section and then selecting a few simple filtering examples (say, low- and highpass filtering using Butterworth filters, as discussed in Sections 4.3.2 and 4.4.2). At the discretion of the instructor, additional material can include full coverage of Sections 4.3 and 4.4. It is seldom possible to go beyond this point in a senior course.

Chapter 5 can be covered as a continuation of Chapter 4. Section 5.1 makes this an easy approach. Then, it is possible give the student a “flavor” of what restoration is (and still keep the discussion brief) by covering only Gaussian and impulse noise in Section 5.2.1, and a couple of spatial filters in Section 5.3. This latter section is a frequent source of confusion to the student who, based on discussions earlier in the chapter, is expecting to see a more objective approach. It is worthwhile to emphasize at this point that spatial enhancement and restoration are the same thing when it comes to noise reduction by spatial filtering. A good way to keep it brief and conclude coverage of restoration is to jump at this point to inverse filtering (which follows directly from the model in Section 5.1) and show the problems with this approach. Then, with a brief explanation regarding the fact that much of restoration centers around the instabilities inherent in inverse filtering, it is possible to introduce the “interactive” form of the Wiener filter in Eq. (5.8-3) and conclude the chapter with Examples 5.12 and 5.13.

Chapter 6 on color image processing is a new feature of the book. Coverage of this

4 Chapter 1 Introduction

chapter also can be brief at the senior level by focusing on enough material to give the student a foundation on the physics of color (Section 6.1), two basic color models (RGB and CMY/CMYK), and then concluding with a brief coverage of pseudocolor processing (Section 6.3).

We typically conclude a senior course by covering some of the basic aspects of image compression (Chapter 8). Interest on this topic has increased significantly as a result of the heavy use of images and graphics over the Internet, and students usually are easily motivated by the topic. Minimum coverage of this material includes Sections 8.1.1 and 8.1.2, Section 8.2, and Section 8.4.1. In this limited scope, it is worthwhile spending one-half of a lecture period filling in any gaps that may arise by skipping earlier parts of the chapter.

One Semester Graduate Course (No Background in DIP)

The main difference between a senior and a first-year graduate course in which neither group has formal background in image processing is mostly in the scope of material covered, in the sense that we simply go faster in a graduate course, and feel much freer in assigning independent reading. In addition to the material discussed in the previous section, we add the following material in a graduate course.

Coverage of histogram matching (Section 3.3.2) is added. Sections 4.3, 4.4, and 4.5 are covered in full. Section 4.6 is touched upon briefly regarding the fact that implementation of discrete Fourier transform techniques requires non-intuitive concepts such as function padding. The separability of the Fourier transform should be covered, and mention of the advantages of the FFT should be made. In Chapter 5 we add Sections 5.5 through 5.8. In Chapter 6 we add the HSI model (Section 6.3.2), Section 6.4, and Section 6.6. A nice introduction to wavelets (Chapter 7) can be achieved by a combination of classroom discussions and independent reading. The minimum number of sections in that chapter are 7.1, 7.2, 7.3, and 7.5, with appropriate (but brief) mention of the existence of fast wavelet transforms. Finally, in Chapter 8 we add coverage of Sections 8.3, 8.4.2, 8.5.1 (through Example 8.16), Section 8.5.2 (through Example 8.20) and Section 8.5.3.

If additional time is available, a natural topic to cover next is morphological image processing (Chapter 9). The material in this chapter begins a transition from methods whose inputs and outputs are images to methods in which the inputs are images, but the outputs are attributes about those images, in the sense defined in Section 1.1. We

recommend coverage of Sections 9.1 through 9.4, and some of the algorithms in Section 9.5.

One Semester Graduate Course (with Background in DIP)

Some programs have an undergraduate course in image processing as a prerequisite to a graduate course on the subject. In this case, it is possible to cover material from the first eleven chapters of the book. Using the undergraduate guidelines described above, we add the following material to form a teaching outline for a one semester graduate course that has that undergraduate material as prerequisite. Given that students have the appropriate background on the subject, independent reading assignments can be used to control the schedule.

Coverage of histogram matching (Section 3.3.2) is added. Sections 4.3, 4.4, 4.5, and 4.6 are added. This strengthens the student's background in frequency-domain concepts. A more extensive coverage of Chapter 5 is possible by adding sections 5.2.3, 5.3.3, 5.4.3, 5.5, 5.6, and 5.8. In Chapter 6 we add full-color image processing (Sections 6.4 through 6.7). Chapters 7 and 8 are covered as in the previous section. As noted in the previous section, Chapter 9 begins a transition from methods whose inputs and outputs are images to methods in which the inputs are images, but the outputs are attributes about those images. As a minimum, we recommend coverage of binary morphology: Sections 9.1 through 9.4, and some of the algorithms in Section 9.5. Mention should be made about possible extensions to gray-scale images, but coverage of this material may not be possible, depending on the schedule. In Chapter 10, we recommend Sections 10.1, 10.2.1 and 10.2.2, 10.3.1 through 10.3.4, 10.4, and 10.5. In Chapter 11 we typically cover Sections 11.1 through 11.4.

Two Semester Graduate Course (No Background in DIP)

A full-year graduate course consists of the material covered in the one semester undergraduate course, the material outlined in the previous section, and Sections 12.1, 12.2, 12.3.1, and 12.3.2.

Projects

One of the most interesting aspects of a course in digital image processing is the pictorial

nature of the subject. It has been our experience that students truly enjoy and benefit from judicious use of computer projects to complement the material covered in class. Since computer projects are in addition to course work and homework assignments, we try to keep the formal project reporting as brief as possible. In order to facilitate grading, we try to achieve uniformity in the way project reports are prepared. A useful report format is as follows:

Page 1: Cover page.

- Project title
- Project number
- Course number
- Student's name
- Date due
- Date handed in
- Abstract (not to exceed 1/2 page)

Page 2: One to two pages (max) of technical discussion.

Page 3 (or 4): Discussion of results. One to two pages (max).

Results: Image results (printed typically on a laser or inkjet printer). All images must contain a number and title referred to in the discussion of results.

Appendix: Program listings, focused on any original code prepared by the student. For brevity, functions and routines provided to the student are referred to by name, but the code is not included.

Layout: The entire report must be on a standard sheet size (e.g., 8.5×11 inches), stapled with three or more staples on the left margin to form a booklet, or bound using clear plastic standard binding products.

Project resources available in the book web site include a sample project, a list of suggested projects from which the instructor can select, book and other images, and MATLAB functions. Instructors who do not wish to use MATLAB will find additional software suggestions in the Support/Software section of the web site.

2 Problem Solutions

Problem 2.1

The diameter, x , of the retinal image corresponding to the dot is obtained from similar triangles, as shown in Fig. P2.1. That is,

$$\frac{(d/2)}{0.2} = \frac{(x/2)}{0.014}$$

which gives $x = 0.07d$. From the discussion in Section 2.1.1, 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 size 580×580 elements. Assuming equal spacing between elements, this gives 580 elements and 579 spaces on a line 1.5 mm long. The size of each element and each space is then $s = [(1.5\text{mm})/1, 579] = 1.3 \times 10^{-6}$ 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 $0.07(d) < 1.3 \times 10^{-6}$ m, or $d < 18.6 \times 10^{-6}$ m.

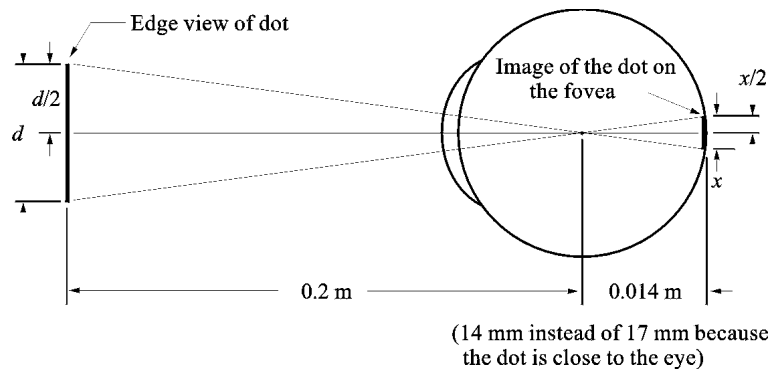


Figure P2.1

Problem 2.2

Brightness adaptation.

Problem 2.3

$$\lambda = c/v = 2.998 \times 10^8(\text{m/s})/60(1/\text{s}) = 4.99 \times 10^6 \text{m} = 5000 \text{Km}.$$

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. A far-ultraviolet camera sensor would be needed to image the specimens. (b) No answer required since the answer to (a) is affirmative.

Problem 2.5

From the geometry of Fig. 2.3, $7\text{mm}/35\text{mm} = z/500\text{mm}$, or $z = 100 \text{mm}$. 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 $1024/100 = 10 \text{ elements/mm}$. For line pairs we divide by 2, giving an answer of 5 lp/mm.

Problem 2.6

One possible solution is to equip a monochrome camera with a mechanical device that sequentially places a red, a green, and a blue pass filter in front of the lens. The strongest camera response determines the color. If all three responses are approximately equal, the object is white. A faster system would utilize three different cameras, each equipped with an individual filter. The analysis would be then based on polling the response of each camera. This system would be a little more expensive, but it would be faster and more reliable. Note that both solutions assume that the field of view of the camera(s) is such that it is completely filled by a uniform color [i.e., the camera(s) is(are) focused on

a part of the vehicle where only its color is seen. Otherwise further analysis would be required to isolate the region of uniform color, which is all that is of interest in solving this problem].

Problem 2.7

The image in question is given by

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

A cross section of the image is shown in Fig. P2.7(a). If the intensity is quantized using m bits, then we have the situation shown in Fig. P2.7(b), where $\Delta G = (255 + 1)/2^m$. Since an abrupt change of 8 gray levels is assumed to be detectable by the eye, it follows that $\Delta G = 8 = 256/2^m$, or $m = 5$. In other words, 32, or fewer, gray levels will produce visible false contouring.

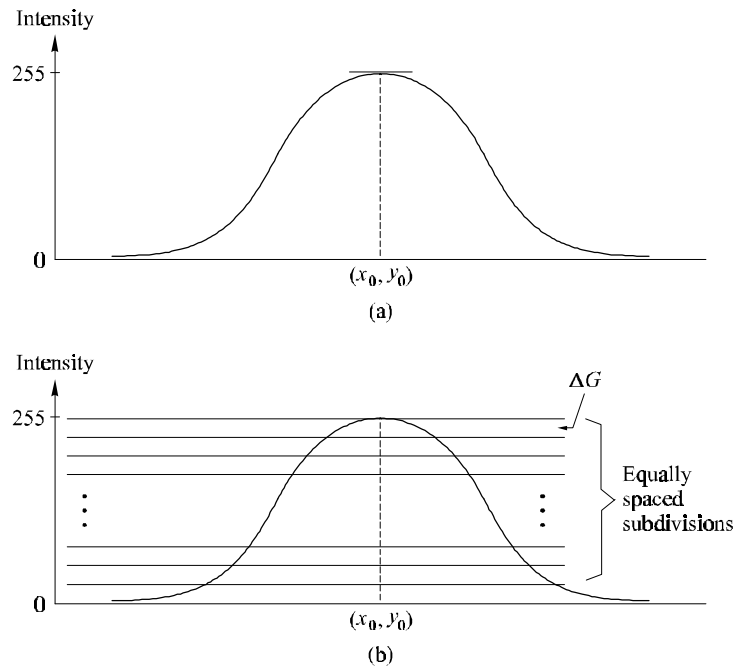


Figure P2.7

Problem 2.8

The use of two bits ($m = 2$) of intensity resolution produces four gray levels in the range 0 to 255. One way to subdivide this range is to let all levels between 0 and 63 be coded as 63, all levels between 64 and 127 be coded as 127, and so on. The image resulting from this type of subdivision is shown in Fig. P2.8. Of course, there are other ways to subdivide the range $[0, 255]$ into four bands.

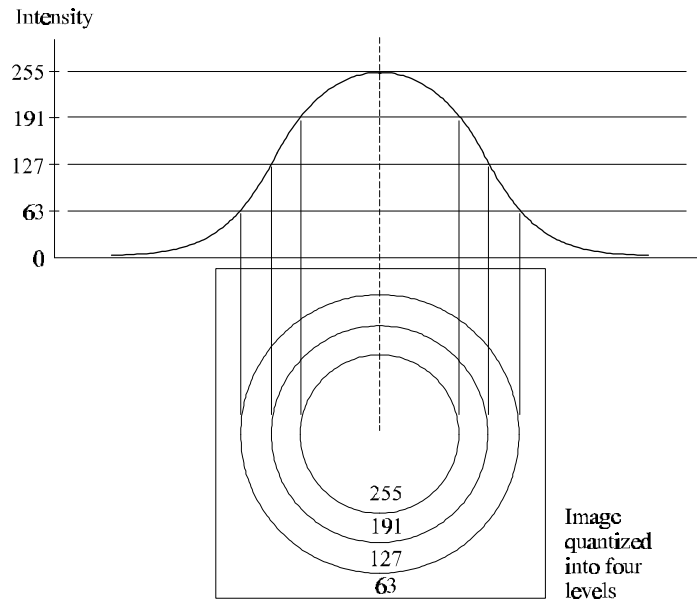


Figure P2.8

Problem 2.9

(a) The total amount of data (including the start and stop bit) in an 8-bit, 1024×1024 image, is $(1024)^2 \times [8 + 2]$ bits. The total time required to transmit this image over a 56K baud link is $(1024)^2 \times [8 + 2] / 56000 = 187.25$ sec or about 3.1 min. (b) At 750K this time goes down to about 14 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 vertical 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 terrabytes). These figures show why image data compression (Chapter 8) is so important.

Problem 2.11

Let p and q be as shown in Fig. P2.11. Then, (a) S_1 and S_2 are not 4-connected because q is not in the set $N_4(p)$; (b) S_1 and S_2 are 8-connected because q is in the set $N_8(p)$; (c) S_1 and S_2 are m -connected because (i) q is in $N_D(p)$, and (ii) the set $N_4(p) \cap N_4(q)$ is empty.

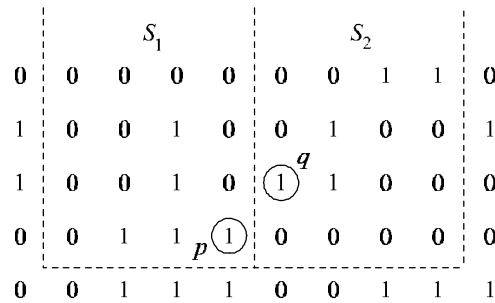


Figure P2.11

Problem 2.12

The solution to this problem consists of defining all possible neighborhood shapes to go from a diagonal segment to a corresponding 4-connected segment, as shown in Fig. P2.12. The algorithm then simply looks for the appropriate match every time a diagonal segment is encountered in the boundary.

Problem 2.13

The solution to this problem is the same as for Problem 2.12 because converting from an m -connected path to a 4-connected path simply involves detecting diagonal segments and converting them to the appropriate 4-connected segment.

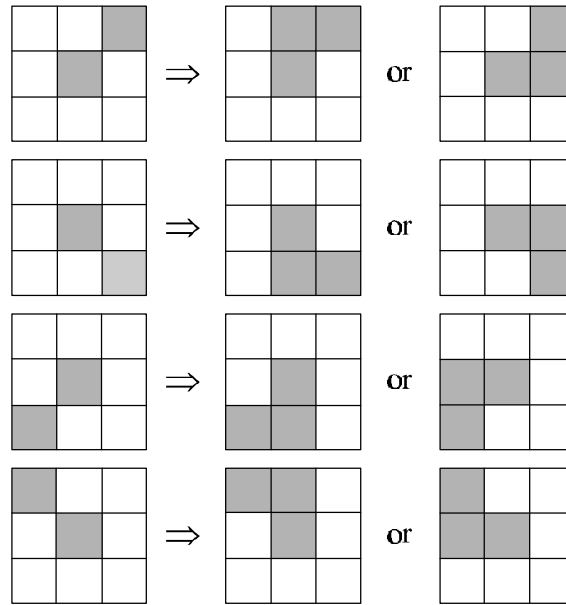


Figure P2.12

Problem 2.14

A region R of an image is composed of a set of connected points in the image. The boundary of a region is the set of points that have one or more neighbors that are not in R . Because boundary points also are part of R , it follows that a point on the boundary has at least one neighbor in R and at least one neighbor not in R . (If the point in the boundary did not have a neighbor in R , the point would be disconnected from R , which violates the definition of points in a region.) Since all points in R are part of a connected component (see Section 2.5.2), all points in the boundary are also connected and a path (entirely in R) exists between any two points on the boundary. Thus the boundary forms a closed path.

Problem 2.15

(a) When $V = \{0, 1\}$, 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 P2.15(a) shows this condition; it is not possible to get to q . The shortest 8-path is shown in Fig. P2.15(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. (b) One

possibility for the shortest 4-path when $V = \{1, 2\}$ is shown in Fig. P2.15(c); its length is 6. It is easily verified that another 4-path of the same length exists between p and q . One possibility for the shortest 8-path (it is not unique) is shown in Fig. P2.15(d); its length is 4. The length of a shortest m -path (shown dashed) is 6. This path is not unique.

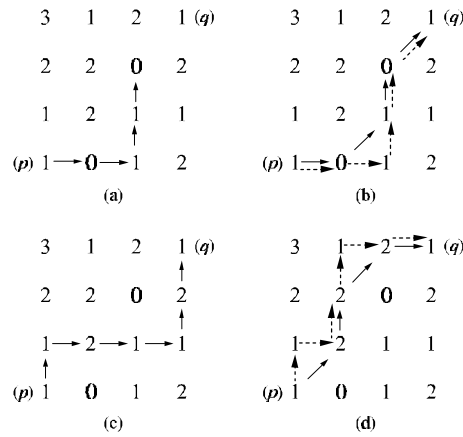


Figure P2.15

Problem 2.16

(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. P2.16, where the assumption is that all points along the path are from V . The length 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. (2.5-16). (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). (b) The path may or may not be unique, depending on V and the values of the points along the way.

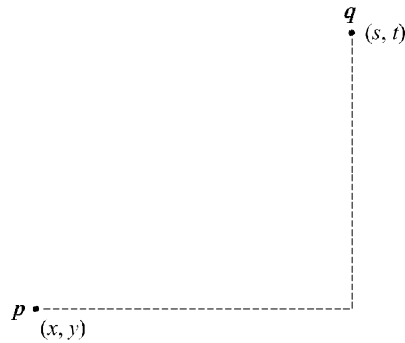


Figure P2.16

Problem 2.17

(a) The D_8 distance between p and q (see Fig. P2.16) is defined as $\max(|x - s|, |y - t|)$. Recall that the D_8 distance (unlike the Euclidean distance) counts diagonal segments the same as horizontal and vertical segments, and, as in the case of the D_4 distance, is independent of whether or not a path exists between p and q . As in the previous problem, the shortest 8-path is equal to the D_8 distance when the path length is $\max(|x - s|, |y - t|)$. This occurs when 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 traveling diagonally in only one direction and, whenever diagonal travel is not possible, by making turns in the horizontal or vertical (but not both) direction. (b) The path may or may not be unique, depending on V and the values of the points along the way.

Problem 2.18

With reference to Eq. (2.6-1), let H denote the neighborhood 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 pixel-by-pixel sum of the elements in S_1 and S_2 , as explained in Section 2.5.4. Note that the size of the neighborhood (i.e., number of pixels) is not changed by this pixel-by-pixel sum. The operator H computes the sum of pixel values in a given neighborhood. Then, $H(aS_1 + bS_2)$ means: (1) multiplying the pixels in each of the subimage areas by the constants shown, (2) adding the pixel-by-pixel values from S_1 and S_2 (which produces a single subimage area), and (3) computing the sum of the values of all the pixels in that single subimage area. 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. (2.6-1), indicates that H is a linear operator.

Problem 2.19

The median, ζ , of a set of numbers is such that half the values in the set are below ζ and the other half are above it. A simple example will suffice to show that Eq. (2.6-1) is violated by the median operator. Let $S_1 = \{1, -2, 3\}$, $S_2 = \{4, 5, 6\}$, and $a = b = 1$. In this case H is the median operator. We then have $H(S_1 + S_2) = \text{median}\{5, 3, 9\} = 5$, where it is understood that $S_1 + S_2$ is the element-by-corresponding-element sum of S_1 and S_2 . Next, we compute $H(S_1) = \text{median}\{1, -2, 3\} = 1$ and $H(S_2) = \text{median}\{4, 5, 6\} = 5$. Then, since $H(aS_1 + bS_2) \neq aH(S_1) + bH(S_2)$, it follows that Eq. (2.6-1) is violated and the median is a nonlinear operator.

Problem 2.20

The geometry of the chips is shown in Fig. P2.20(a). From Fig. P2.20(b) and the geometry in Fig. 2.3, we know that

$$\Delta x = \frac{\lambda \times 80}{\lambda - z}$$

where Δx is the side dimension of the image (assumed square since the viewing screen is square) impinging on the image plane, and the 80 mm refers to the size of the viewing screen, as described in the problem statement. The most inexpensive solution will result from using a camera of resolution 512×512 . Based on the information in Fig. P2.20(a), a CCD chip with this resolution will be of size $(16\mu) \times (512) = 8 \text{ mm}$ on each side. Substituting $\Delta x = 8 \text{ mm}$ in the above equation gives $z = 9\lambda$ as the relationship between the distance z and the focal length of the lens, where a minus sign was ignored because it is just a coordinate inversion. If a 25 mm lens is used, the front of the lens will have to be located at approximately 225 mm from the viewing screen so that the size of the