

**D**

**igital  
Image  
Processing**

Third Edition

**Instructor's  
Manual  
(Evaluation)**

**Rafael C. Gonzalez  
Richard E. Woods**

# Digital Image Processing

*Third Edition*

## Instructor's Manual—Evaluation Copy

*Version 3.0*

Rafael C. Gonzalez

Richard E. Woods

Prentice Hall  
Upper Saddle River, NJ 07458

[www.imageprocessingplace.com](http://www.imageprocessingplace.com)

Copyright © 1992-2008 R. C. Gonzalez and R. E. Woods

## About the Evaluation Manual

The Evaluation Manual has the exact same teaching guidelines as the Full Manual, but only a few problem solutions are included for the purposes of evaluation.

We provide this Evaluation Manual because the Full Manual is available only to two instructors who have adopted the book for classroom use. The Evaluation Manual allows individuals desiring to evaluate the Manual as a possible precursor to adopting the book.

# Chapter 1

## Introduction

The purpose of this chapter is to present suggested guidelines for teaching material from *Digital Image Processing* at the senior and first-year graduate levels. 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. Solutions to a few selected problems are included in this manual for purposes of evaluation. Detailed solutions to all problems in the book are included in the full manual.

### 1.1 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 Tutorials section in the book web site contains review materials on matrix theory and probability, and has a brief introduction to linear systems. PowerPoint classroom presentation material on the review topics is available in the Faculty section of the web site.

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.

## 1.2 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 three 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 prepared with this objective in mind. Some of this material can be covered in class in the first period 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 in the book web site was designed for just this purpose.

Chapter 2 should be covered in its entirety. Some of the material (Sections 2.1 through 2.3.3) can be assigned as independent reading, but more detailed explanation (combined with some additional independent reading) of Sections 2.3.4 and 2.4 through 2.6 is time well spent. The material in Section 2.6 covers concepts that are used throughout the book and provides a number of image processing applications that are useful as motivational background for the rest of the book

Chapter 3 covers spatial intensity transformations and spatial correlation and convolution as the foundation of spatial filtering. The chapter also covers a number of different uses of spatial transformations and spatial filtering for im-

age enhancement. These techniques are illustrated in the context enhancement (as motivational aids), but it is pointed out several times in the chapter that the methods developed have a much broader range of application. For a senior course, we recommend covering Sections 3.1 through 3.3.1, and Sections 3.4 through 3.6. Section 3.7 can be assigned as independent reading, depending on time.

The key objectives of Chapter 4 are (1) to start from basic principles of signal sampling and from these derive the discrete Fourier transform; and (2) to illustrate the use of filtering in the frequency domain. As in Chapter 3, we use mostly examples from image enhancement, but make it clear that the Fourier transform has a much broader scope of application. The early part of the chapter through Section 4.2.2 can be assigned as independent reading. We recommend careful coverage of Sections 4.2.3 through 4.3.4. Section 4.3.5 can be assigned as independent reading. Section 4.4 should be covered in detail. The early part of Section 4.5 deals with extending to 2-D the material derived in the earlier sections of this chapter. Thus, Sections 4.5.1 through 4.5.3 can be assigned as independent reading and then devote part of the period following the assignment to summarizing that material. We recommend class coverage of the rest of the section. In Section 4.6, we recommend that Sections 4.6.1-4.6.6 be covered in class. Section 4.6.7 can be assigned as independent reading. Sections 4.7.1-4.7.3 should be covered and Section 4.7.4 can be assigned as independent reading. In Sections 4.8 through 4.9 we recommend covering one filter (like the ideal lowpass and highpass filters) and assigning the rest of those two sections as independent reading. In a senior course, we recommend covering Section 4.9 through Section 4.9.3 only. In Section 4.10, we also recommend covering one filter and assigning the rest as independent reading. In Section 4.11, we recommend covering Sections 4.11.1 and 4.11.2 and mentioning the existence of FFT algorithms. The  $\log_2$  computational advantage of the FFT discussed in the early part of Section 4.11.3 should be mentioned, but in a senior course there typically is no time to cover development of the FFT in detail.

Chapter 5 can be covered as a continuation of Chapter 4. Section 5.1 makes this an easy approach. Then, it is possible to 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 two of the 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 discuss Examples 5.12 and 5.13. At a minimum, we recommend a brief discussion on image reconstruction by covering Sections 5.11.1-5.11-2 and mentioning that the rest of Section 5.11 deals with ways to generated projections in which blur is minimized.

Coverage of Chapter 6 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 in 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. The amount of material covered depends on the time left in the semester.

### **1.3 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 the material covered, in the sense that we simply go faster in a graduate course and feel much freer in assigning independent reading. In a graduate course we add the following material to the material suggested in the previous section.

Sections 3.3.2-3.3.4 are added as is Section 3.3.8 on fuzzy image processing. We cover Chapter 4 in its entirety (with appropriate sections assigned as independent reading, depending on the level of the class). To Chapter 5 we add Sections 5.6-5.8 and cover Section 5.11 in detail. 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. Sections 8.1 and 8.2 through Section 8.2.8 provide a nice introduction to image compression.

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

#### *1.4. ONE SEMESTER GRADUATE COURSE (WITH STUDENT HAVING BACKGROUND IN DIP)*<sup>5</sup>

defined in Section 1.1. We recommend coverage of Sections 9.1 through 9.4, and some of the algorithms in Section 9.5.

### **1.4 One Semester Graduate Course (with Student Having Background in DIP)**

Some programs have an undergraduate course in image processing as a prerequisite to a graduate course on the subject, in which case the course can be biased toward the latter chapters. In this case, a good deal of Chapters 2 and 3 is review, with the exception of Section 3.8, which deals with fuzzy image processing. Depending on what is covered in the undergraduate course, many of the sections in Chapter 4 will be review as well. For Chapter 5 we recommend the same level of coverage as outlined in the previous section.

In Chapter 6 we add full-color image processing (Sections 6.4 through 6.7). Chapters 7 and 8 are covered as outlined 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 through 10.4. In Chapter 11 we typically cover Sections 11.1 through 11.4.

### **1.5 Two Semester Graduate Course (No Background in DIP)**

In a two-semester course it is possible to cover material in all twelve chapters of the book. The key in organizing the syllabus is the background the students bring to the class. For example, in an electrical and computer engineering curriculum graduate students have strong background in frequency domain processing, so Chapter 4 can be covered much quicker than would be the case in which the students are from, say, a computer science program. The important aspect of a full year course is exposure to the material in all chapters, even when some topics in each chapter are not covered.

### **1.6 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. Because 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., letter size in the U.S. or A4 in Europe), stapled with three or more staples on the left margin to form a booklet, or bound using clear plastic standard binding products.  
1.2 One Semester Senior Course

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.

## 1.7 The Book Web Site

The companion web site

[www.prenhall.com/gonzalezwoods](http://www.prenhall.com/gonzalezwoods)

(or its mirror site)

[www.imageprocessingplace.com](http://www.imageprocessingplace.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 hand-outs 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.



## Chapter 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,

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

which gives  $x = 0.085d$ . 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  $580 \times 580$  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  $s = [(1.5\text{mm})/1,159] = 1.3 \times 10^{-6}$  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.

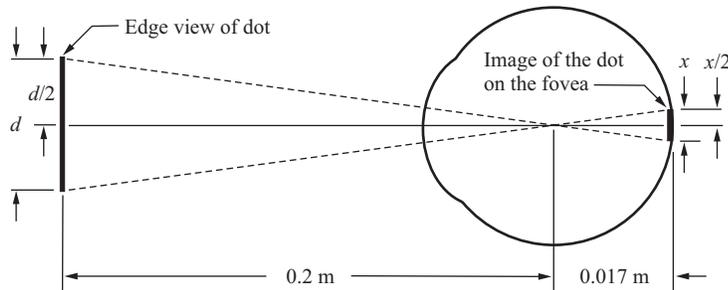


Figure P2.1

### Problem 2.3

The solution is

$$\begin{aligned}\lambda &= c/v \\ &= 2.998 \times 10^8 (\text{m/s}) / 60 (1/\text{s}) \\ &= 4.997 \times 10^6 \text{ m} = 4997 \text{ Km.}\end{aligned}$$

### 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 then would be 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 2.9

(a) The total amount of data (including the start and stop bit) in an 8-bit,  $1024 \times 1024$  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 3000K this time goes down to about 3.5 sec.

### 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.

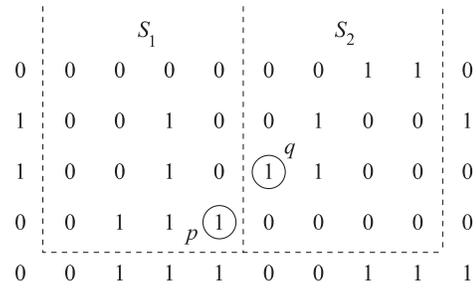


Figure P2.11

### Problem 2.12

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

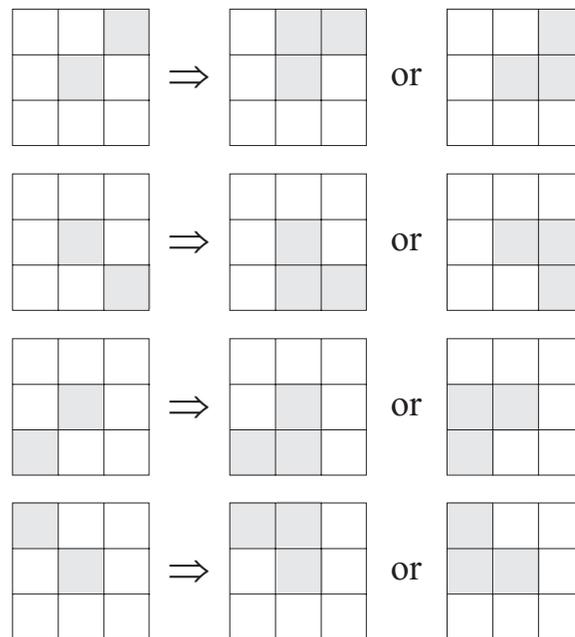


Figure P2.12

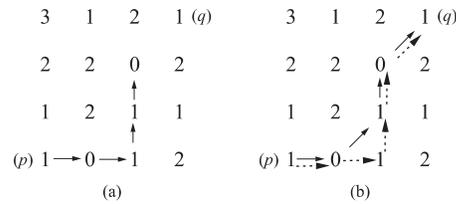


Figure P.2.15

### Problem 2.15

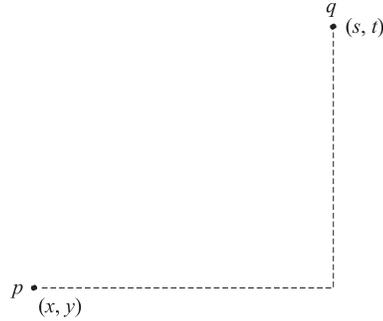
(a) When  $V = \{0, 1\}$ , 4-path does not exist between  $p$  and  $q$  because it is impossible to get from  $p$  to  $q$  by traveling along points that are both 4-adjacent and also have values from  $V$ . Figure P.2.15(a) shows this condition; it is not possible to get to  $q$ . The shortest 8-path is shown in Fig. P.2.15(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.

### 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. P.2.16, 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. (2.5-2). (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).

### Problem 2.18

With reference to Eq. (2.6-1), 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 pixel-by-pixel sum of the elements in  $S_1$  and  $S_2$ , as explained in Section 2.6.1. 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) multiply the pixels in each of the subimage areas by the constants shown, (2) add



**Figure P2.16**

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. 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 2.20

From Eq. (2.6-5), at any point  $(x, y)$ ,

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

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 image, so  $E\{f_i\} = f$ . Also, it is given that the noise has zero mean, so  $E\{\eta_i\} = 0$ . Thus, it follows that  $E\{\bar{g}\} = f$ , which proves the validity of Eq. (2.6-6).

To prove the validity of Eq. (2.6-7) consider the preceding equation again:

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

It is known from random-variable theory that the variance of the sum of uncorrelated random variables is the sum of the variances of those variables (Papoulis [1991]). Because it is given that the elements of  $f$  are constant and the  $\eta_i$  are uncorrelated, then

$$\sigma_{\bar{g}}^2 = \sigma_f^2 + \frac{1}{K^2} [\sigma_{\eta_1}^2 + \sigma_{\eta_2}^2 + \cdots + \sigma_{\eta_K}^2].$$

The first term on the right side is 0 because the elements of  $f$  are constants. The various  $\sigma_{\eta_i}^2$  are simply samples of the noise, which has variance  $\sigma_{\eta}^2$ . Thus,  $\sigma_{\eta_i}^2 = \sigma_{\eta}^2$  and we have

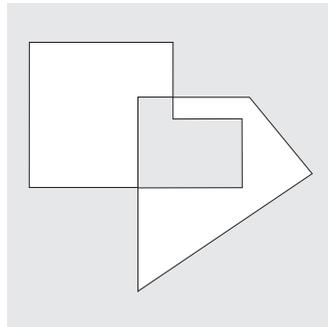
$$\sigma_{\bar{g}}^2 = \frac{K}{K^2} \sigma_{\eta}^2 = \frac{1}{K} \sigma_{\eta}^2$$

which proves the validity of Eq. (2.6-7).

## Problem 2.22

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 simple 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 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. 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,



$$(A \cap B) \cup (A \cup B)^c$$

**Figure P2.23**

comparing the differences between them makes no sense. Often, special markings are manufactured into the product for mechanical or image-based alignment

Controlled illumination (note that “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 each 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. Another (sometimes complementary) approach is to implement image processing techniques (e.g., image averaging) to reduce noise.

Obviously there are a number of 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 2.23

(a) The answer is shown in Fig. P2.23.

### Problem 2.26

From Eq. (2.6-27) and the definition of separable kernels,

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

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. (2.6-33), the 2-D Fourier transform is given by

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

It is easily verified that the Fourier transform kernel is separable (Problem 2.25), 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  $x = 0, 1, \dots, M-1$ .

