MODULE 3

5 Image Restoration and Reconstruction

> Things which we see are not by themselves what we see. . . .
> It remains completely unknown to us what the objects may be by themselves and apart from the receptivity of our senses. We know nothing but our manner of perceiving them.
>
> *Immanuel Kant*

Preview

As in image enhancement, the principal goal of restoration techniques is to improve an image in some predefined sense. Although there are areas of overlap, image enhancement is largely a subjective process, while image restoration is for the most part an objective process. Restoration attempts to recover an image that has been degraded by using a priori knowledge of the degradation phenomenon. Thus, restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image.

This approach usually involves formulating a criterion of goodness that will yield an optimal estimate of the desired result. By contrast, enhancement techniques basically are heuristic procedures designed to manipulate an image in order to take advantage of the psychophysical aspects of the human visual system. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

The material developed in this chapter is strictly introductory. We consider the restoration problem only from the point where a degraded, *digital* image is given; thus we consider topics dealing with sensor, digitizer, and display degradations only superficially. These subjects, although of importance in the overall treatment of image restoration applications, are beyond the scope of the present discussion.

5.2 Noise Models

The principal sources of noise in digital images arise during image acquisition and/or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition, and by the quality of the sensing elements themselves. For instance, in acquiring images with a CCD camera, light levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Images are corrupted during transmission principally due to interference in the channel used for transmission. For example, an image transmitted using a wireless network might be corrupted as a result of lightning or other atmospheric disturbance.

5.2.1 Spatial and Frequency Properties of Noise

Relevant to our discussion are parameters that define the spatial characteristics of noise, and whether the noise is correlated with the image. Frequency properties refer to the frequency content of noise in the Fourier sense (i.e., as opposed to frequencies of the electromagnetic spectrum). For example, when the Fourier spectrum of noise is constant, the noise usually is called *white noise*. This terminology is a carryover from the physical properties of white light, which contains nearly all frequencies in the visible spectrum in equal proportions. From the discussion in Chapter 4, it is not difficult to show that the Fourier spectrum of a function containing all frequencies in equal proportions is a constant.

With the exception of spatially periodic noise (Section 5.2.3), we assume in this chapter that noise is independent of spatial coordinates, and that it is uncorrelated with respect to the image itself (that is, there is no correlation between pixel values and the values of noise components). Although these assumptions are at least partially invalid in some applications (quantum-limited imaging, such as in X-ray and nuclear-medicine imaging, is a good example), the complexities of dealing with spatially dependent and correlated noise are beyond the scope of our discussion.

5.2.2 Some Important Noise Probability Density Functions

Consult the book Web site for a brief review of probability theory.

Based on the assumptions in the previous section, the *spatial* noise descriptor with which we shall be concerned is the statistical behavior of the intensity values in the noise component of the model in Fig. 5.1. These may be considered random variables, characterized by a probability density function (PDF). The following are among the most common PDFs found in image processing applications.

Gaussian noise

Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (also called *normal*) noise models are used frequently in practice. In fact, this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best.

The PDF of a Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2} \tag{5.2-1}$$

where z represents intensity, $\bar{z}$ is the mean[†] (average) value of z, and σ is its standard deviation. The standard deviation squared, σ^2, is called the *variance* of z. A plot of this function is shown in Fig. 5.2(a). When z is described by Eq. (5.2-1), approximately 70% of its values will be in the range $[(\bar{z} - \sigma), (\bar{z} + \sigma)]$, and about 95% will be in the range $[(\bar{z} - 2\sigma), (\bar{z} + 2\sigma)]$.

Rayleigh noise

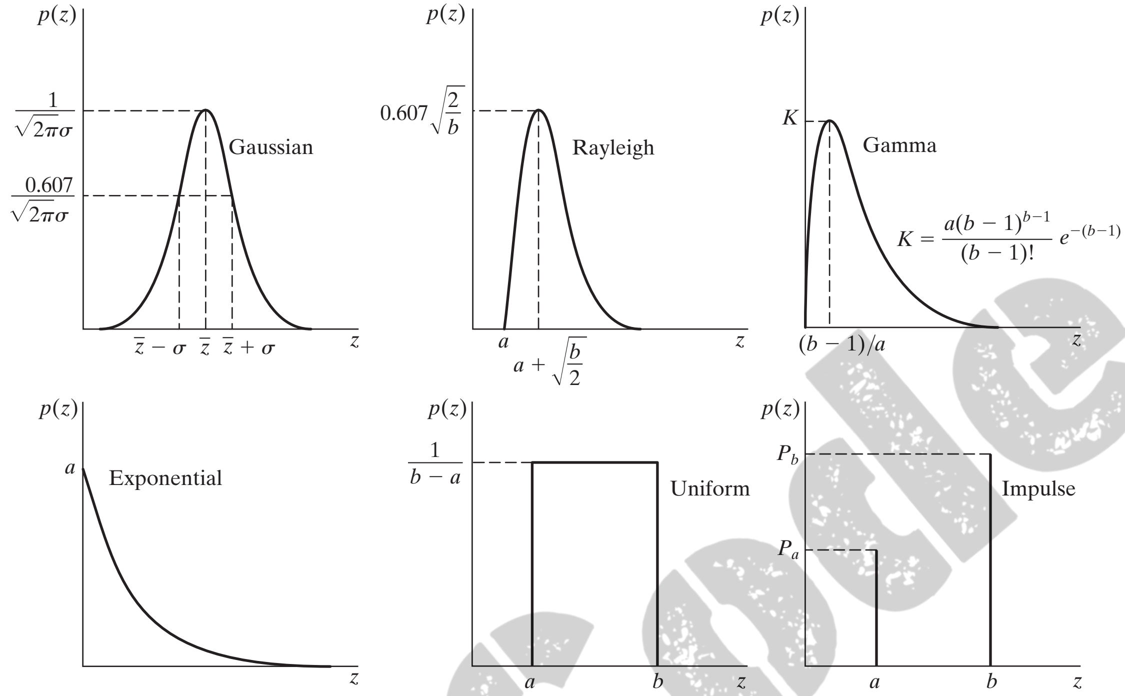
The PDF of Rayleigh noise is given by

$$p(z) = \begin{cases} \dfrac{2}{b}(z - a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \tag{5.2-2}$$

The mean and variance of this density are given by

$$\bar{z} = a + \sqrt{\pi b/4} \tag{5.2-3}$$

[†] We use $\bar{z}$ instead of m to denote the mean in this section to avoid confusion when we use m and n later to denote neighborhood size.



a b c
d e f

FIGURE 5.2 Some important probability density functions.

and

$$\sigma^2 = \frac{b(4 - \pi)}{4} \qquad (5.2\text{-}4)$$

Figure 5.2(b) shows a plot of the Rayleigh density. Note the displacement from the origin and the fact that the basic shape of this density is skewed to the right. The Rayleigh density can be quite useful for approximating skewed histograms.

Erlang (gamma) noise

The PDF of Erlang noise is given by

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \qquad (5.2\text{-}5)$$

where the parameters are such that $a > 0$, b is a positive integer, and "!" indicates factorial. The mean and variance of this density are given by

$$\bar{z} = \frac{b}{a} \qquad (5.2\text{-}6)$$

and

$$\sigma^2 = \frac{b}{a^2} \qquad (5.2\text{-}7)$$

Figure 5.2(c) shows a plot of this density. Although Eq. (5.2-5) often is referred to as the *gamma density*, strictly speaking this is correct only when the denominator is the gamma function, $\Gamma(b)$. When the denominator is as shown, the density is more appropriately called the *Erlang density*.

Exponential noise

The PDF of *exponential* noise is given by

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \tag{5.2-8}$$

where $a > 0$. The mean and variance of this density function are

$$\bar{z} = \frac{1}{a} \tag{5.2-9}$$

and

$$\sigma^2 = \frac{1}{a^2} \tag{5.2-10}$$

Note that this PDF is a special case of the Erlang PDF, with $b = 1$. Figure 5.2(d) shows a plot of this density function.

Uniform noise

The PDF of *uniform* noise is given by

$$p(z) = \begin{cases} \dfrac{1}{b - a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \tag{5.2-11}$$

The mean of this density function is given by

$$\bar{z} = \frac{a + b}{2} \tag{5.2-12}$$

and its variance by

$$\sigma^2 = \frac{(b - a)^2}{12} \tag{5.2-13}$$

Figure 5.2(e) shows a plot of the uniform density.

Impulse (salt-and-pepper) noise

The PDF of (*bipolar*) *impulse* noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \tag{5.2-14}$$

If $b > a$, intensity b will appear as a light dot in the image. Conversely, level a will appear like a dark dot. If either P_a or P_b is zero, the impulse noise is called *unipolar*. If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules randomly distributed over the image. For this reason, bipolar impulse noise also is called *salt-and-pepper* noise. *Data-drop-out* and *spike* noise also are terms used to refer to this type of noise. We use the terms *impulse* or *salt-and-pepper* noise interchangeably.

Noise impulses can be negative or positive. Scaling usually is part of the image digitizing process. Because impulse corruption usually is large compared with the strength of the image signal, impulse noise generally is digitized as extreme (pure black or white) values in an image. Thus, the assumption usually is that a and b are "saturated" values, in the sense that they are equal to the minimum and maximum allowed values in the digitized image. As a result, negative impulses appear as black (pepper) points in an image. For the same reason, positive impulses appear as white (salt) noise. For an 8-bit image this means typically that $a = 0$ (black) and $b = 255$ (white). Figure 5.2(f) shows the PDF of impulse noise.

As a group, the preceding PDFs provide useful tools for modeling a broad range of noise corruption situations found in practice. For example, Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and/or high temperature. The Rayleigh density is helpful in characterizing noise phenomena in range imaging. The exponential and gamma densities find application in laser imaging. Impulse noise is found in situations where quick transients, such as faulty switching, take place during imaging, as mentioned in the previous paragraph. The uniform density is perhaps the least descriptive of practical situations. However, the uniform density is quite useful as the basis for numerous random number generators that are used in simulations (Peebles [1993] and Gonzalez, Woods, and Eddins [2004]).

EXAMPLE 5.1: Noisy images and their histograms.

■ Figure 5.3 shows a test pattern well suited for illustrating the noise models just discussed. This is a suitable pattern to use because it is composed of simple, constant areas that span the gray scale from black to near white in only three increments. This facilitates visual analysis of the characteristics of the various noise components added to the image.

Figure 5.4 shows the test pattern after addition of the six types of noise discussed thus far in this section. Shown below each image is the histogram computed directly from that image. The parameters of the noise were chosen in each case so that the histogram corresponding to the three intensity levels in the test pattern would start to merge. This made the noise quite visible, without obscuring the basic structure of the underlying image.

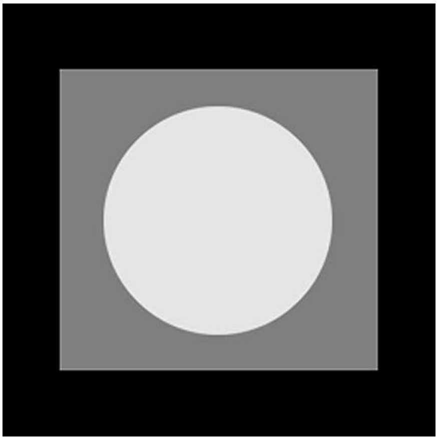
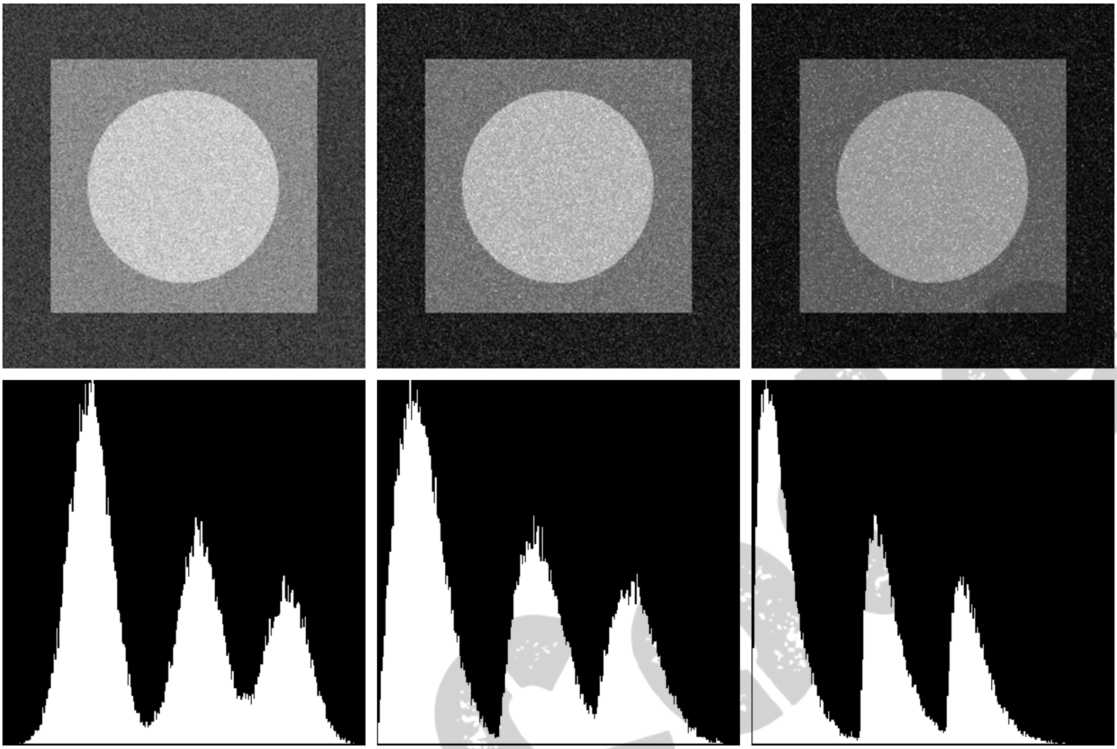


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.



Gaussian

Rayleigh

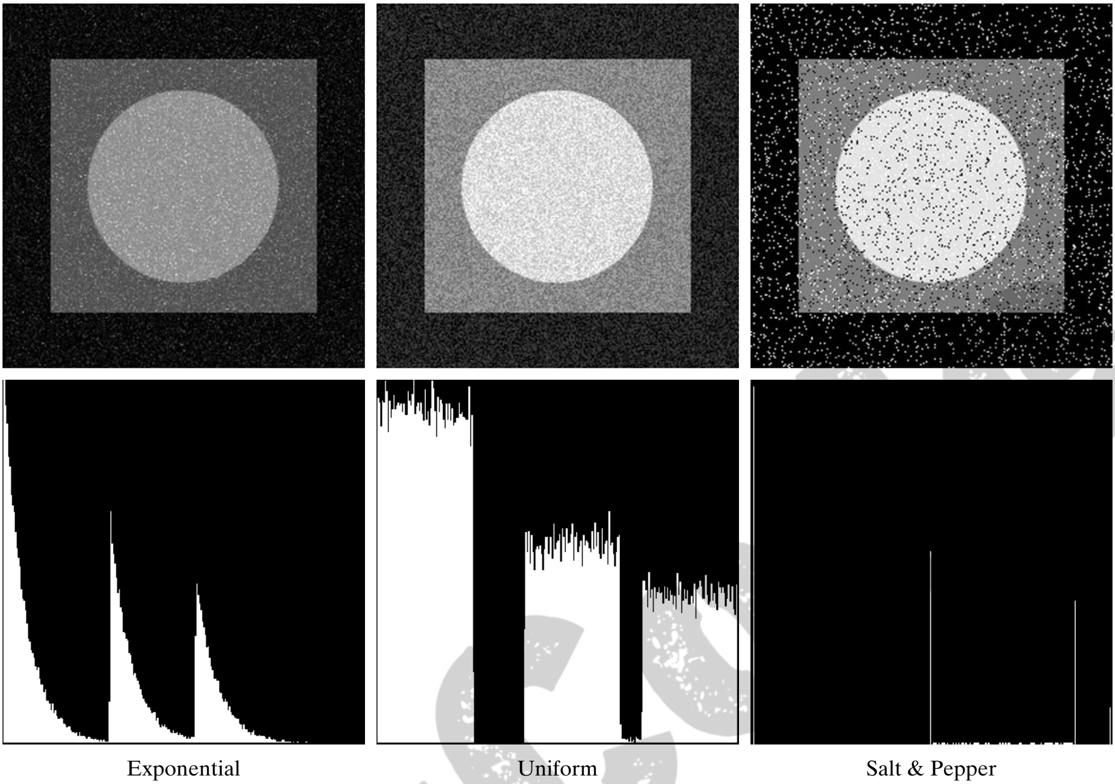
Gamma

FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

We see a close correspondence in comparing the histograms in Fig. 5.4 with the PDFs in Fig. 5.2. The histogram for the salt-and-pepper example has an extra peak at the white end of the intensity scale because the noise components were pure black and white, and the lightest component of the test pattern (the circle) is light gray. With the exception of slightly different overall intensity, it is difficult to differentiate visually between the first five images in Fig. 5.4, even though their histograms are significantly different. The salt-and-pepper appearance of the image corrupted by impulse noise is the only one that is visually indicative of the type of noise causing the degradation. ■

5.2.3 Periodic Noise

Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition. This is the only type of spatially dependent noise that will be considered in this chapter. As discussed in Section 5.4, periodic noise can be reduced significantly via frequency domain filtering. For example, consider the image in Fig. 5.5(a). This image is severely corrupted by (spatial) sinusoidal noise of various frequencies. The Fourier transform of a pure


Exponential

Uniform

Salt & Pepper

g h i
j k l

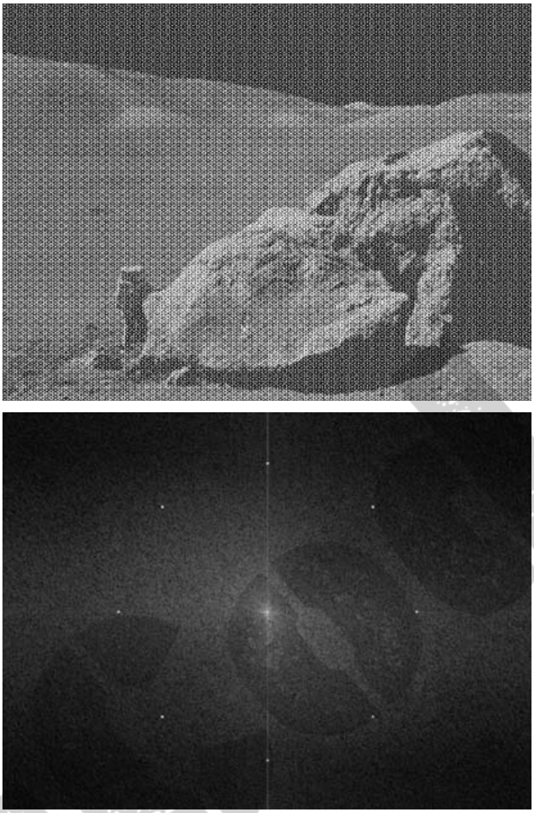
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt-and-pepper noise to the image in Fig. 5.3.

sinusoid is a pair of conjugate impulses[†] located at the conjugate frequencies of the sine wave (Table 4.3). Thus, if the amplitude of a sine wave in the spatial domain is strong enough, we would expect to see in the spectrum of the image a pair of impulses for each sine wave in the image. As shown in Fig. 5.5(b), this is indeed the case, with the impulses appearing in an approximate circle because the frequency values in this particular case are so arranged. We will have much more to say in Section 5.4 about this and other examples of periodic noise.

5.2.4 Estimation of Noise Parameters

The parameters of periodic noise typically are estimated by inspection of the Fourier spectrum of the image. As noted in the previous section, periodic noise tends to produce frequency spikes that often can be detected even by visual analysis. Another approach is to attempt to infer the periodicity of noise components directly from the image, but this is possible only in simplistic cases.

[†] Be careful not to confuse the term *impulse* in the frequency domain with the use of the same term in impulse noise.



a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

Automated analysis is possible in situations in which the noise spikes are either exceptionally pronounced, or when knowledge is available about the general location of the frequency components of the interference.

The parameters of noise PDFs may be known partially from sensor specifications, but it is often necessary to estimate them for a particular imaging arrangement. If the imaging system is available, one simple way to study the characteristics of system noise is to capture a set of images of "flat" environments. For example, in the case of an optical sensor, this is as simple as imaging a solid gray board that is illuminated uniformly. The resulting images typically are good indicators of system noise.

When only images already generated by a sensor are available, frequently it is possible to estimate the parameters of the PDF from small patches of reasonably constant background intensity. For example, the vertical strips (of 150×20 pixels) shown in Fig. 5.6 were cropped from the Gaussian, Rayleigh, and uniform images in Fig. 5.4. The histograms shown were calculated using image data from these small strips. The histograms in Fig. 5.4 that correspond to the histograms in Fig. 5.6 are the ones in the middle of the group of three in

Figs. 5.4(d), (e), and (k). We see that the shapes of these histograms correspond quite closely to the shapes of the histograms in Fig. 5.6. Their heights are different due to scaling, but the shapes are unmistakably similar.

The simplest use of the data from the image strips is for calculating the mean and variance of intensity levels. Consider a strip (subimage) denoted by S, and let $p_S(z_i)$, $i = 0, 1, 2, \dots, L - 1$, denote the probability estimates (normalized histogram values) of the intensities of the pixels in S, where L is the number of possible intensities in the entire image (e.g., 256 for an 8-bit image). As in Chapter 3, we estimate the mean and variance of the pixels in S as follows:

$$\bar{z} = \sum_{i=0}^{L-1} z_i p_S(z_i) \tag{5.2-15}$$

and

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p_S(z_i) \tag{5.2-16}$$

The shape of the histogram identifies the closest PDF match. If the shape is approximately Gaussian, then the mean and variance are all we need because the Gaussian PDF is completely specified by these two parameters. For the other shapes discussed in Section 5.2.2, we use the mean and variance to solve for the parameters a and b. Impulse noise is handled differently because the estimate needed is of the actual probability of occurrence of white and black pixels. Obtaining this estimate requires that both black and white pixels be visible, so a midgray, relatively constant area is needed in the image in order to be able to compute a histogram. The heights of the peaks corresponding to black and white pixels are the estimates of P_a and P_b in Eq. (5.2-14).

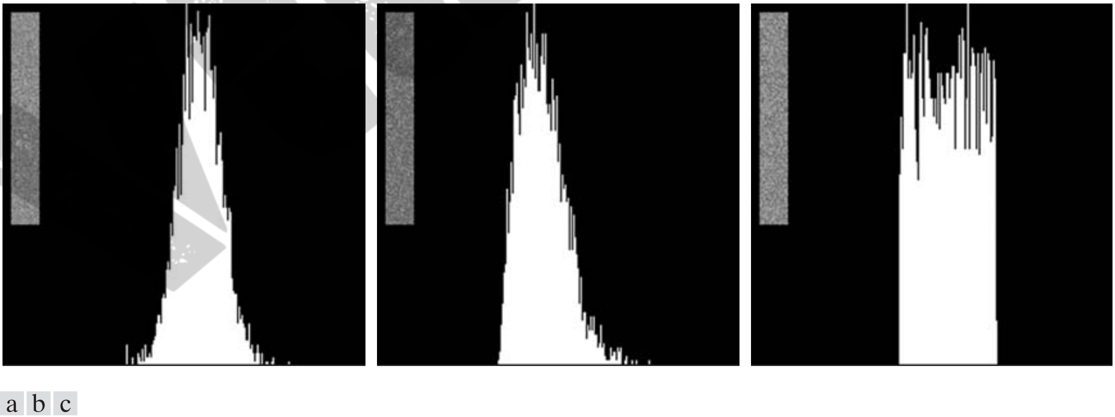


FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

5.3 Restoration in the Presence of Noise Only—Spatial Filtering

When the only degradation present in an image is noise, Eqs. (5.1-1) and (5.1-2) become

$$g(x, y) = f(x, y) + \eta(x, y) \tag{5.3-1}$$

and

$$G(u, v) = F(u, v) + N(u, v) \tag{5.3-2}$$

The noise terms are unknown, so subtracting them from $g(x, y)$ or $G(u, v)$ is not a realistic option. In the case of periodic noise, it usually is possible to estimate $N(u, v)$ from the spectrum of $G(u, v)$, as noted in Section 5.2.3. In this case $N(u, v)$ can be subtracted from $G(u, v)$ to obtain an estimate of the original image. In general, however, this type of knowledge is the exception, rather than the rule.

Spatial filtering is the method of choice in situations when only additive random noise is present. Spatial filtering is discussed in detail in Chapter 3. With the exception of the nature of the computation performed by a specific filter, the mechanics for implementing all the filters that follow are exactly as discussed in Sections 3.4 through 3.6.

5.3.1 Mean Filters

In this section we discuss briefly the noise-reduction capabilities of the spatial filters introduced in Section 3.5 and develop several other filters whose performance is in many cases superior to the filters discussed in that section.

Arithmetic mean filter

We assume that m and n are odd integers.

This is the simplest of the mean filters. Let S_{xy} represent the set of coordinates in a rectangular subimage window (neighborhood) of size $m \times n$, centered at point (x, y). The arithmetic mean filter computes the average value of the corrupted image $g(x, y)$ in the area defined by S_{xy}. The value of the restored image $\hat{f}$ at point (x, y) is simply the arithmetic mean computed using the pixels in the region defined by S_{xy}. In other words,

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t) \tag{5.3-3}$$

This operation can be implemented using a spatial filter of size $m \times n$ in which all coefficients have value $1/mn$. A mean filter smooths local variations in an image, and noise is reduced as a result of blurring.

Geometric mean filter

An image restored using a *geometric mean* filter is given by the expression

$$\hat{f}(x, y) = \left[\prod_{(s,t)\in S_{xy}} g(s, t) \right]^{\frac{1}{mn}} \tag{5.3-4}$$

Here, each restored pixel is given by the product of the pixels in the subimage window, raised to the power $1/mn$. As shown in Example 5.2, a geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process.

Harmonic mean filter

The *harmonic mean* filtering operation is given by the expression

$$\hat{f}(x, y) = \frac{mn}{\displaystyle\sum_{(s,t)\in S_{xy}} \frac{1}{g(s, t)}} \tag{5.3-5}$$

The harmonic mean filter works well for salt noise, but fails for pepper noise. It does well also with other types of noise like Gaussian noise.

Contraharmonic mean filter

The *contraharmonic* mean filter yields a restored image based on the expression

$$\hat{f}(x, y) = \frac{\displaystyle\sum_{(s,t)\in S_{xy}} g(s, t)^{Q+1}}{\displaystyle\sum_{(s,t)\in S_{xy}} g(s, t)^{Q}} \tag{5.3-6}$$

where Q is called the *order* of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise. For positive values of Q, the filter eliminates pepper noise. For negative values of Q it eliminates salt noise. It cannot do both simultaneously. Note that the contraharmonic filter reduces to the arithmetic mean filter if $Q = 0$, and to the harmonic mean filter if $Q = -1$.

EXAMPLE 5.2: Illustration of mean filters.

■ Figure 5.7(a) shows an 8-bit X-ray image of a circuit board, and Fig. 5.7(b) shows the same image, but corrupted with additive Gaussian noise of zero mean and variance of 400. For this type of image this is a significant level of noise. Figures 5.7(c) and (d) show, respectively, the result of filtering the noisy

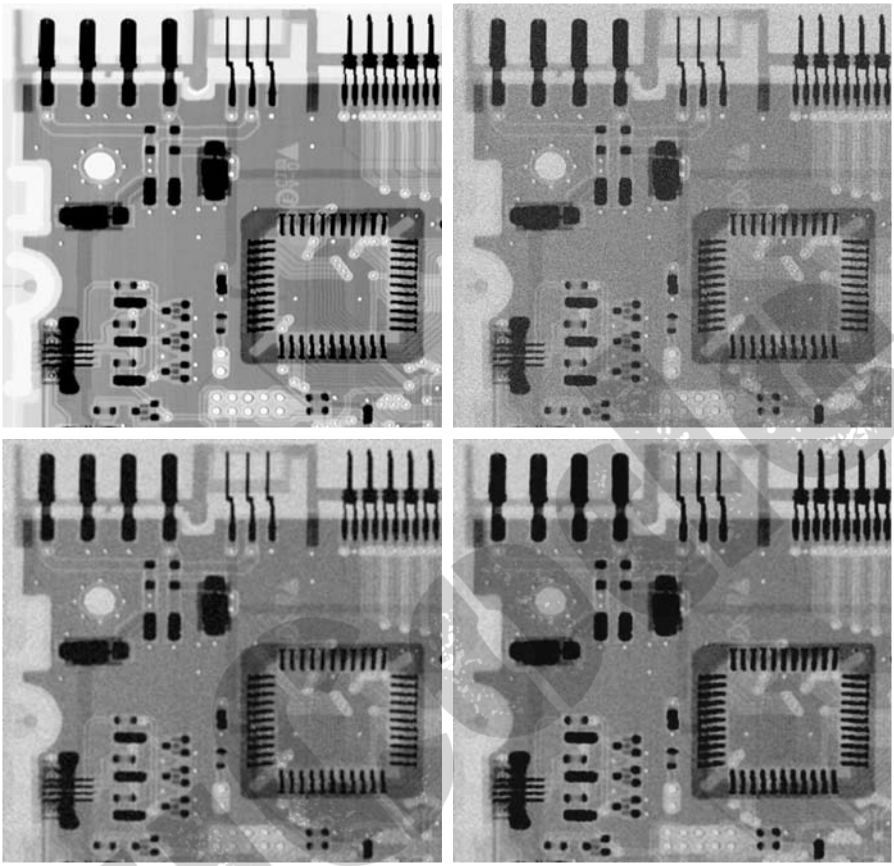
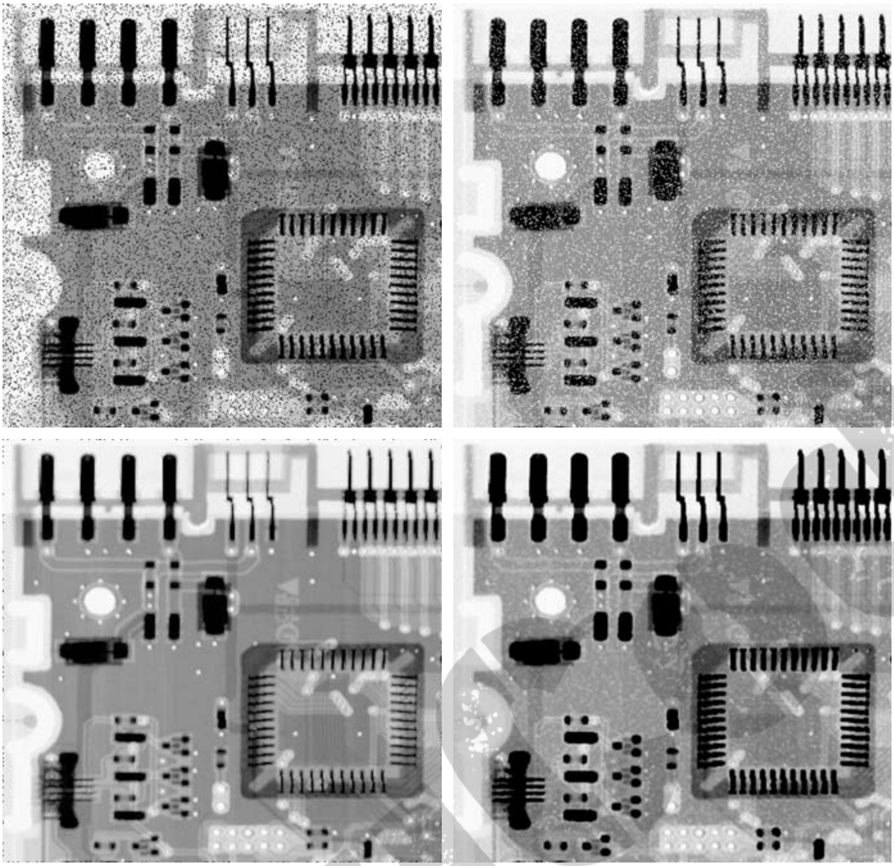


FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

image with an arithmetic mean filter of size 3 × 3 and a geometric mean filter of the same size. Although both filters did a reasonable job of attenuating the contribution due to noise, the geometric mean filter did not blur the image as much as the arithmetic filter. For instance, the connector fingers at the top of the image are sharper in Fig. 5.7(d) than in (c). The same is true in other parts of the image.

Figure 5.8(a) shows the same circuit image, but corrupted now by pepper noise with probability of 0.1. Similarly, Fig. 5.8(b) shows the image corrupted by salt noise with the same probability. Figure 5.8(c) shows the result of filtering Fig. 5.8(a) using a contraharmonic mean filter with $Q = 1.5$, and Fig. 5.8(d) shows the result of filtering Fig. 5.8(b) with $Q = -1.5$. Both filters did a good job in reducing the effect of the noise. The positive-order filter did a better job of cleaning the background, at the expense of slightly thinning and blurring the dark areas. The opposite was true of the negative-order filter.



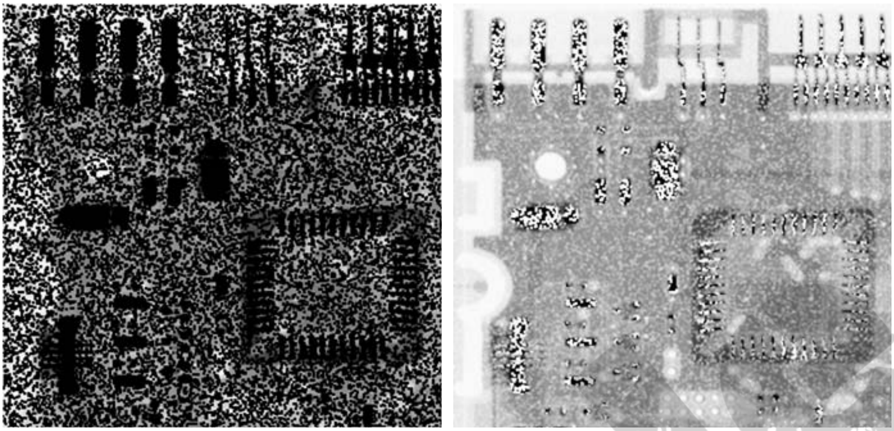
a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

In general, the arithmetic and geometric mean filters (particularly the latter) are well suited for random noise like Gaussian or uniform noise. The contraharmonic filter is well suited for impulse noise, but it has the disadvantage that it must be known whether the noise is dark or light in order to select the proper sign for Q. The results of choosing the wrong sign for Q can be disastrous, as Fig. 5.9 shows. Some of the filters discussed in the following sections eliminate this shortcoming. ■

5.3.2 Order-Statistic Filters

Order-statistic filters were introduced in Section 3.5.2. We now expand the discussion in that section and introduce some additional order-statistic filters. As noted in Section 3.5.2, order-statistic filters are spatial filters whose response is based on ordering (ranking) the values of the pixels contained in the image area encompassed by the filter. The ranking result determines the response of the filter.



a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

Median filter

The best-known order-statistic filter is the *median filter*, which, as its name implies, replaces the value of a pixel by the median of the intensity levels in the neighborhood of that pixel:

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s, t)\} \tag{5.3-7}$$

The value of the pixel at (x, y) is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. In fact, as Example 5.3 below shows, the median filter yields excellent results for images corrupted by this type of noise. Computation of the median and implementation of this filter are discussed in Section 3.5.2.

Max and min filters

See the second margin note in Section 10.3.5 regarding percentiles.

Although the median filter is by far the order-statistic filter most used in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but you will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100th percentile results in the so-called *max filter*, given by

$$\hat{f}(x, y) = \max_{(s,t)\in S_{xy}}\{g(s, t)\} \tag{5.3-8}$$

This filter is useful for finding the brightest points in an image. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area S_{xy}.

The 0th percentile filter is the *min filter*:

$$\hat{f}(x, y) = \min_{(s,t)\in S_{xy}} \{g(s, t)\} \tag{5.3-9}$$

This filter is useful for finding the darkest points in an image. Also, it reduces salt noise as a result of the min operation.

Midpoint filter

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter:

$$\hat{f}(x, y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{g(s, t)\} + \min_{(s,t)\in S_{xy}} \{g(s, t)\}\right] \tag{5.3-10}$$

Note that this filter combines order statistics and averaging. It works best for randomly distributed noise, like Gaussian or uniform noise.

Alpha-trimmed mean filter

Suppose that we delete the $d/2$ lowest and the $d/2$ highest intensity values of $g(s, t)$ in the neighborhood S_{xy}. Let $g_r(s, t)$ represent the remaining $mn - d$ pixels. A filter formed by averaging these remaining pixels is called an *alpha-trimmed mean* filter:

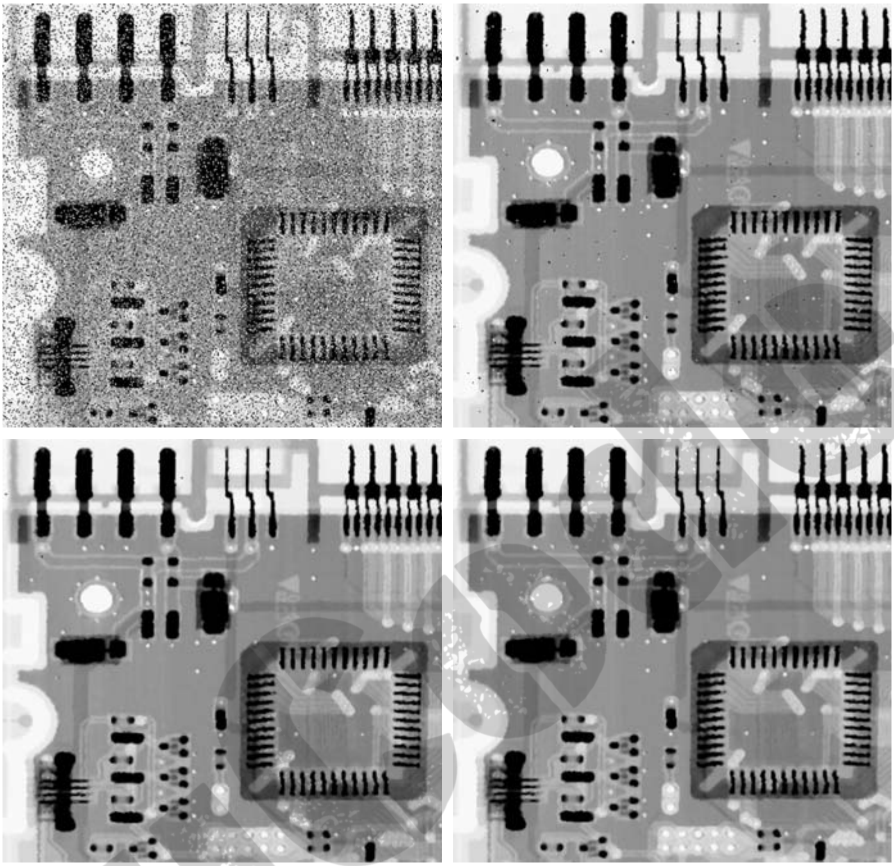
$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,\, t)\in S_{xy}} g_r(s, t) \tag{5.3-11}$$

where the value of d can range from 0 to $mn - 1$. When $d = 0$, the alpha-trimmed filter reduces to the arithmetic mean filter discussed in the previous section. If we choose $d = mn - 1$, the filter becomes a median filter. For other values of d, the alpha-trimmed filter is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

EXAMPLE 5.3: Illustration of order-statistic filters.

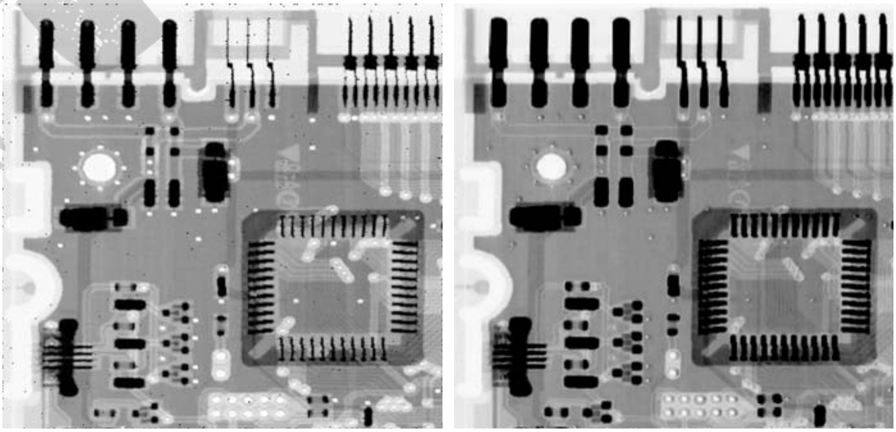
■ Figure 5.10(a) shows the circuit board image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. Figure 5.10(b) shows the result of median filtering with a filter of size 3 × 3. The improvement over Fig. 5.10(a) is significant, but several noise points still are visible. A second pass [on the image in Fig. 5.10(b)] with the median filter removed most of these points, leaving only few, barely visible noise points. These were removed with a third pass of the filter. These results are good examples of the power of median filtering in handling impulse-like additive noise. Keep in mind that repeated passes of a median filter will blur the image, so it is desirable to keep the number of passes as low as possible.

Figure 5.11(a) shows the result of applying the max filter to the pepper noise image of Fig. 5.8(a). The filter did a reasonable job of removing the pepper noise, but we note that it also removed (set to a light intensity level) some dark pixels from the borders of the dark objects. Figure 5.11(b) shows the result of applying the min filter to the image in Fig. 5.8(b). In this case, the min filter did a better job than the max filter on noise removal, but it removed some white points around the border of light objects. These made the light objects smaller and



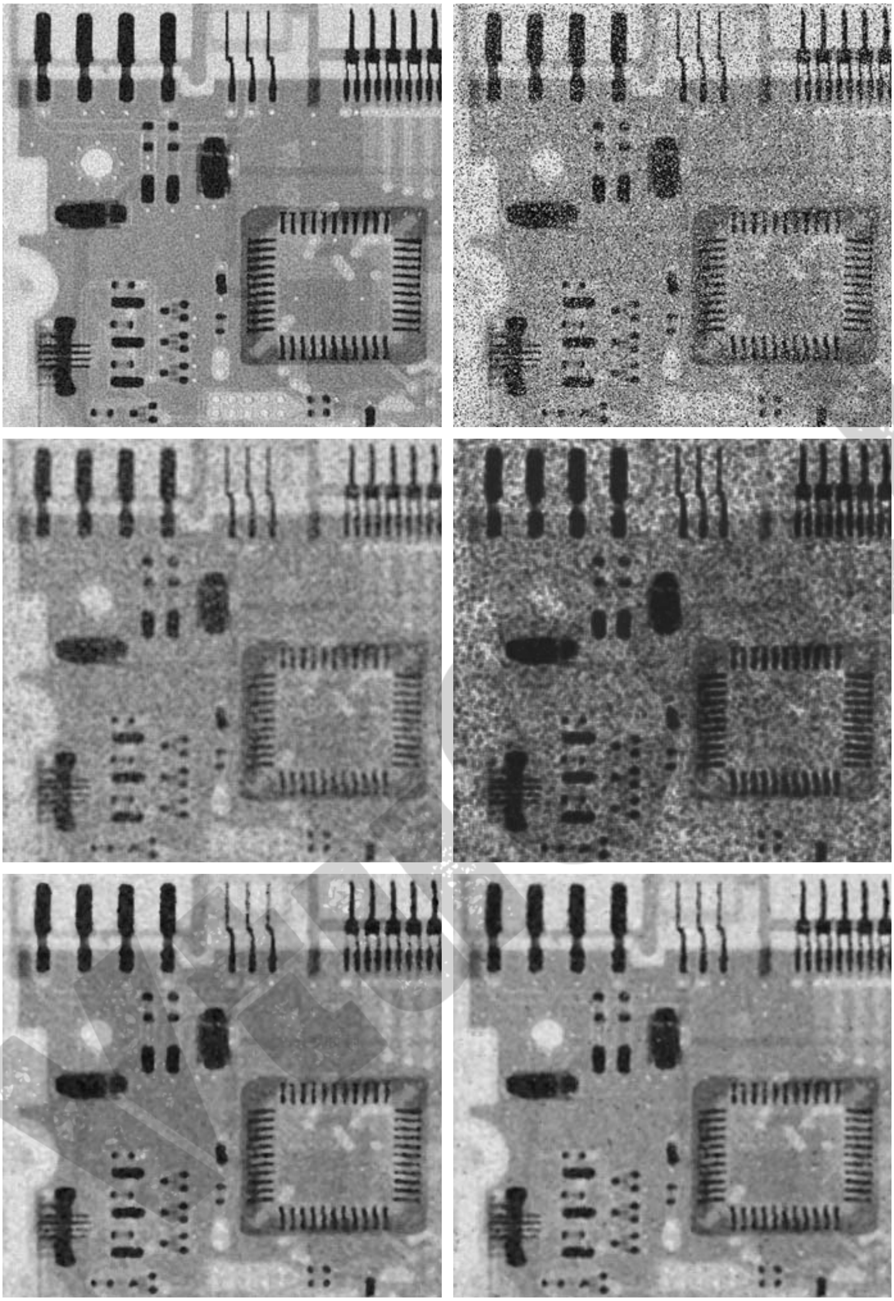
a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3 × 3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.



a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image (b) filtered with a 5 × 5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

some of the dark objects larger (like the connector fingers in the top of the image) because white points around these objects were set to a dark level.

The alpha-trimmed filter is illustrated next. Figure 5.12(a) shows the circuit board image corrupted this time by additive, uniform noise of variance 800 and zero mean. This is a high level of noise corruption that is made worse by further addition of salt-and-pepper noise with $P_a = P_b = 0.1$, as Fig. 5.12(b) shows. The high level of noise in this image warrants use of larger filters. Figures 5.12(c) through (f) show the results obtained using arithmetic mean, geometric mean, median, and alpha-trimmed mean (with $d = 5$) filters of size 5×5. As expected, the arithmetic and geometric mean filters (especially the latter) did not do well because of the presence of impulse noise. The median and alpha-trimmed filters performed much better, with the alpha-trimmed filter giving slightly better noise reduction. Note, for example, that the fourth connector finger from the top left is slightly smoother in the alpha-trimmed result. This is not unexpected because, for a high value of d, the alpha-trimmed filter approaches the performance of the median filter, but still retains some smoothing capabilities. ■

5.3.3 Adaptive Filters

Once selected, the filters discussed thus far are applied to an image without regard for how image characteristics vary from one point to another. In this section we take a look at two *adaptive* filters whose behavior changes based on statistical characteristics of the image inside the filter region defined by the $m \times n$ rectangular window S_{xy}. As the following discussion shows, adaptive filters are capable of performance superior to that of the filters discussed thus far. The price paid for improved filtering power is an increase in filter complexity. Keep in mind that we still are dealing with the case in which the degraded image is equal to the original image plus noise. No other types of degradations are being considered yet.

Adaptive, local noise reduction filter

The simplest statistical measures of a random variable are its mean and variance. These are reasonable parameters on which to base an adaptive filter because they are quantities closely related to the appearance of an image. The mean gives a measure of average intensity in the region over which the mean is computed, and the variance gives a measure of contrast in that region.

Our filter is to operate on a local region, S_{xy}. The response of the filter at any point (x, y) on which the region is centered is to be based on four quantities: (a) $g(x, y)$, the value of the noisy image at (x, y); (b) σ_η^2, the variance of the noise corrupting $f(x, y)$ to form $g(x, y)$; (c) m_L, the local mean of the pixels in S_{xy}; and (d) σ_L^2, the local variance of the pixels in S_{xy}. We want the behavior of the filter to be as follows:

1. If σ_η^2 is zero, the filter should return simply the value of $g(x, y)$. This is the trivial, zero-noise case in which $g(x, y)$ is equal to $f(x, y)$.
2. If the local variance is high relative to σ_η^2, the filter should return a value close to $g(x, y)$. A high local variance typically is associated with edges, and these should be preserved.
3. If the two variances are equal, we want the filter to return the arithmetic mean value of the pixels in S_{xy}. This condition occurs when the local area has the same properties as the overall image, and local noise is to be reduced simply by averaging.

An adaptive expression for obtaining $\hat{f}(x, y)$ based on these assumptions may be written as

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x, y) - m_L\right] \tag{5.3-12}$$

The only quantity that needs to be known or estimated is the variance of the overall noise, σ_η^2. The other parameters are computed from the pixels in S_{xy} at each location (x, y) on which the filter window is centered. A tacit assumption in Eq. (5.3-12) is that $\sigma_\eta^2 \leq \sigma_L^2$. The noise in our model is additive and position independent, so this is a reasonable assumption to make because S_{xy} is a subset of $g(x, y)$. However, we seldom have exact knowledge of σ_η^2. Therefore, it is possible for this condition to be violated in practice. For that reason, a test should be built into an implementation of Eq. (5.3-12) so that the ratio is set to 1 if the condition $\sigma_\eta^2 > \sigma_L^2$ occurs. This makes this filter nonlinear. However, it prevents nonsensical results (i.e., negative intensity levels, depending on the value of m_L) due to a potential lack of knowledge about the variance of the image noise. Another approach is to allow the negative values to occur, and then rescale the intensity values at the end. The result then would be a loss of dynamic range in the image.

EXAMPLE 5.4: Illustration of adaptive, local noise-reduction filtering.

■ Figure 5.13(a) shows the circuit-board image, corrupted this time by additive Gaussian noise of zero mean and a variance of 1000. This is a significant level of noise corruption, but it makes an ideal test bed on which to compare relative filter performance. Figure 5.13(b) is the result of processing the noisy image with an arithmetic mean filter of size 7 × 7. The noise was smoothed out, but at the cost of significant blurring in the image. Similar comments are applicable to Fig. 5.13(c), which shows the result of processing the noisy image with a geometric mean filter, also of size 7 × 7. The differences between these two filtered images are analogous to those we discussed in Example 5.2; only the degree of blurring is different.

Figure 5.13(d) shows the result of using the adaptive filter of Eq. (5.3-12) with $\sigma_\eta^2 = 1000$. The improvements in this result compared with the two previous filters are significant. In terms of overall noise reduction, the adaptive filter achieved results similar to the arithmetic and geometric mean filters. However, the image filtered with the adaptive filter is much sharper. For example, the connector fingers at the top of the image are significantly sharper in Fig. 5.13(d). Other features, such as holes and the eight legs of the dark component on the lower left-hand side of the image, are much clearer in Fig. 5.13(d). These results are typical of what can be achieved with an adaptive filter. As mentioned earlier, the price paid for the improved performance is additional filter complexity.

The preceding results used a value for σ_η^2 that matched the variance of the noise exactly. If this quantity is not known and an estimate is used that is too low, the algorithm will return an image that closely resembles the original because the corrections will be smaller than they should be. Estimates that are too high

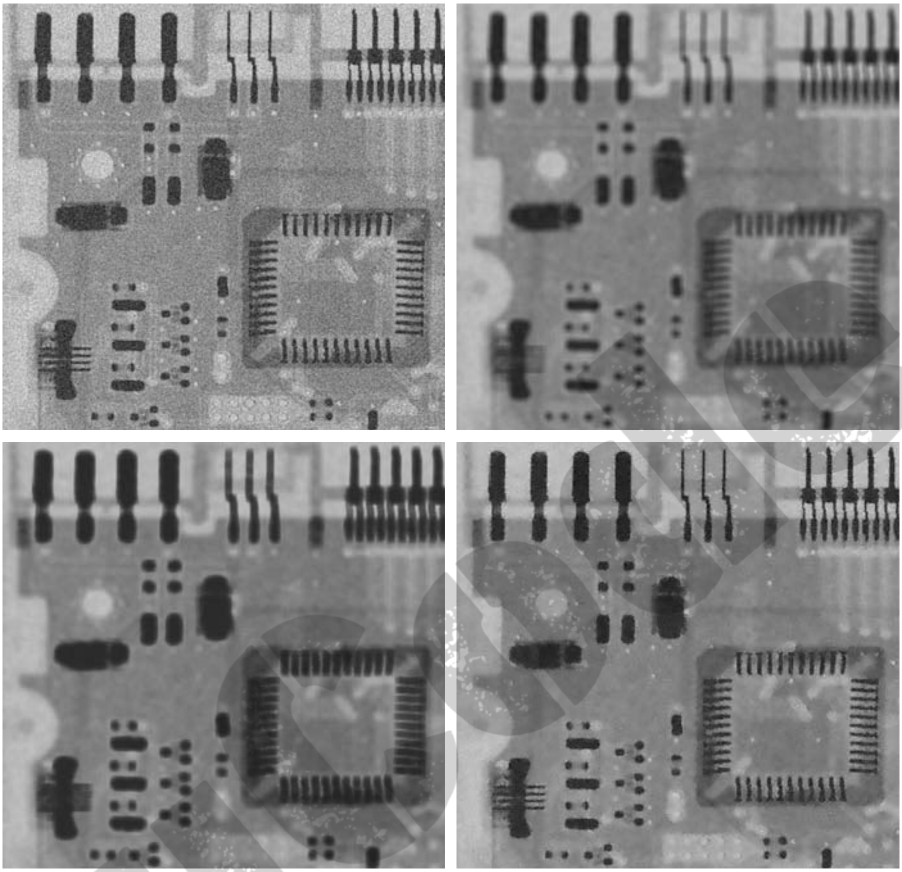


FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.

will cause the ratio of the variances to be clipped at 1.0, and the algorithm will subtract the mean from the image more frequently than it would normally. If negative values are allowed and the image is rescaled at the end, the result will be a loss of dynamic range, as mentioned previously. ■

Adaptive median filter

The median filter discussed in Section 5.3.2 performs well if the spatial density of the impulse noise is not large (as a rule of thumb, P_a and P_b less than 0.2). It is shown in this section that adaptive median filtering can handle impulse noise with probabilities larger than these. An additional benefit of the adaptive median filter is that it seeks to preserve detail while smoothing nonimpulse noise, something that the "traditional" median filter does not do. As in all the filters discussed in the preceding sections, the adaptive median filter also works in a rectangular window area S_{xy}. Unlike those filters, however, the adaptive median filter changes (increases) the size of S_{xy} during filter operation, depending on certain conditions listed in this section. Keep in mind that the output of the filter is a single value used to replace the value of the pixel at (x, y), the point on which the window S_{xy} is centered at a given time.

Consider the following notation:

$$\begin{aligned} z_{\min} &= \text{minimum intensity value in } S_{xy} \\ z_{\max} &= \text{maximum intensity value in } S_{xy} \\ z_{\text{med}} &= \text{median of intensity values in } S_{xy} \\ z_{xy} &= \text{intensity value at coordinates } (x, y) \\ S_{\max} &= \text{maximum allowed size of } S_{xy} \end{aligned}$$

The adaptive median-filtering algorithm works in two stages, denoted stage A and stage B, as follows:

```
Stage A:    $A1 = z_{\text{med}} - z_{\min}$
            $A2 = z_{\text{med}} - z_{\max}$
            If $A1 > 0$ AND $A2 < 0$,  go to stage $B$
            Else increase the window size
            If window size $\leq S_{\max}$ repeat stage $A$
            Else output $z_{\text{med}}$

Stage B:    $B1 = z_{xy} - z_{\min}$
            $B2 = z_{xy} - z_{\max}$
            If $B1 > 0$ AND $B2 < 0$,  output $z_{xy}$
            Else output $z_{\text{med}}$
```

The key to understanding the mechanics of this algorithm is to keep in mind that it has three main purposes: to remove salt-and-pepper (impulse) noise, to provide smoothing of other noise that may not be impulsive, and to reduce distortion, such as excessive thinning or thickening of object boundaries. The values $z_{\min}$ and $z_{\max}$ are considered statistically by the algorithm to be "impulse-like" noise components, even if these are not the lowest and highest possible pixel values in the image.

With these observations in mind, we see that the purpose of stage A is to determine if the median filter output, z_{med}, is an impulse (black *or* white) or not. If the condition $z_{\min} < z_{\text{med}} < z_{\max}$ holds, then z_{med} cannot be an impulse for the reason mentioned in the previous paragraph. In this case, we go to stage B and test to see if the point in the center of the window, z_{xy}, is itself an impulse (recall that z_{xy} is the point being processed). If the condition $B1 > 0$ AND $B2 < 0$ is true, then $z_{\min} < z_{xy} < z_{\max}$, and z_{xy} cannot be an impulse for the same reason that z_{med} was not. In this case, the algorithm outputs the unchanged pixel value, z_{xy}. By not changing these "intermediate-level" points, distortion is reduced in the image. If the condition $B1 > 0$ AND $B2 < 0$ is false, then either $z_{xy} = z_{\min}$ or $z_{xy} = z_{\max}$. In either case, the value of the pixel is an extreme value and the algorithm outputs the median value z_{med}, which we know from stage A is not a noise impulse. The last step is what the standard median filter does. The problem is that the standard median filter replaces every point in the image by the median of the corresponding neighborhood. This causes unnecessary loss of detail.

Continuing with the explanation, suppose that stage A *does* find an impulse (i.e., it fails the test that would cause it to branch to stage B). The algorithm then increases the size of the window and repeats stage A. This looping continues until the algorithm either finds a median value that is not an impulse (and branches to stage B), or the maximum window size is reached. If the maximum window size is reached, the algorithm returns the value of z_{med}. Note that there is no guarantee that this value is not an impulse. The smaller the noise probabilities P_a and/or P_b are, or the larger $S_{\max}$ is allowed to be, the less likely it is that a premature exit condition will occur. This is plausible. As the density of the impulses increases, it stands to reason that we would need a larger window to "clean up" the noise spikes.

Every time the algorithm outputs a value, the window S_{xy} is moved to the next location in the image. The algorithm then is reinitialized and applied to the pixels in the new location. As indicated in Problem 3.18, the median value can be updated iteratively using only the new pixels, thus reducing computational load.

EXAMPLE 5.5: Illustration of adaptive median filtering.

■ Figure 5.14(a) shows the circuit-board image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$, which is 2.5 times the noise level used in Fig. 5.10(a). Here the noise level is high enough to obscure most of the detail in the image. As a basis for comparison, the image was filtered first using the smallest median filter required to remove most visible traces of impulse noise. A 7×7 median filter was required to do this, and the result is shown in Fig. 5.14(b). Although the noise was effectively removed, the filter caused significant loss of detail in the image. For instance, some of the connector fingers at the top of the image appear distorted or broken. Other image details are similarly distorted.

Figure 5.14(c) shows the result of using the adaptive median filter with $S_{\max} = 7$. Noise removal performance was similar to the median filter. However, the adaptive filter did a better job of preserving sharpness and detail. The connector fingers are less distorted, and some other features that were either obscured or distorted beyond recognition by the median filter appear sharper and better defined in Fig. 5.14(c). Two notable examples are the feed-through small white holes throughout the board, and the dark component with eight legs in the bottom, left quadrant of the image.

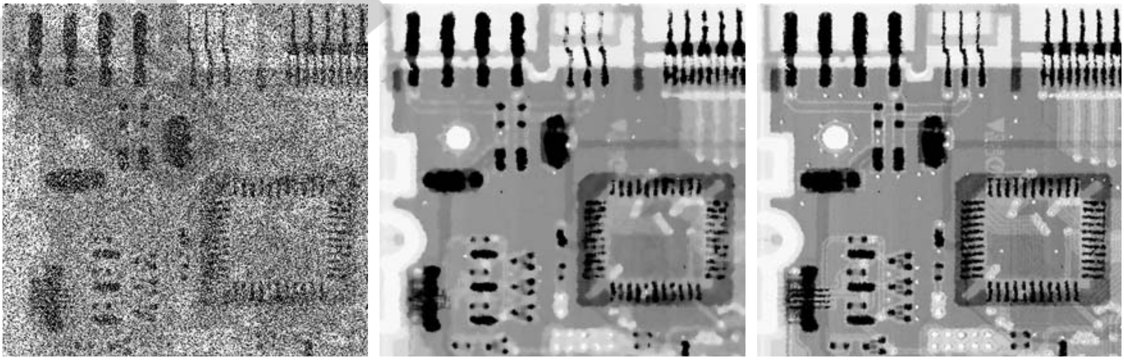


FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$.

Considering the high level of noise in Fig. 5.14(a), the adaptive algorithm performed quite well. The choice of maximum allowed window size depends on the application, but a reasonable starting value can be estimated by experimenting with various sizes of the standard median filter first. This will establish a visual baseline regarding expectations on the performance of the adaptive algorithm. ■

5.4 Periodic Noise Reduction by Frequency Domain Filtering

Periodic noise can be analyzed and filtered quite effectively using frequency domain techniques. The basic idea is that periodic noise appears as concentrated bursts of energy in the Fourier transform, at locations corresponding to the frequencies of the periodic interference. The approach is to use a selective filter (see Section 4.10) to isolate the noise. The three types of selective filters (bandreject, bandpass, and notch, introduced in Section 4.10) are used in Sections 5.4.1 through 5.4.3 for basic periodic noise reduction. We also develop an optimum notch filtering approach in Section 5.4.4.

5.4.1 Bandreject Filters

The transfer functions of ideal, Butterworth, and Gaussian bandreject filters, introduced in Section 4.10.1, are summarized in Table 4.6. Figure 5.15 shows perspective plots of these filters, and the following example illustrates using a bandreject filter for reducing the effects of periodic noise.

EXAMPLE 5.6: Use of bandreject filtering for periodic noise removal.

■ One of the principal applications of bandreject filtering is for noise removal in applications where the general location of the noise component(s) in the frequency domain is approximately known. A good example is an image corrupted by additive periodic noise that can be approximated as two-dimensional sinusoidal functions. It is not difficult to show that the Fourier transform of a sine consists of two impulses that are mirror images of each other about the origin of the transform. Their locations are given in Table 4.3. The impulses are both imaginary (the real part of the Fourier transform of a sine is zero) and are complex conjugates of each other. We will have more to say about this topic in Sections 5.4.3 and 5.4.4. Our purpose at the moment is to illustrate bandreject filtering.

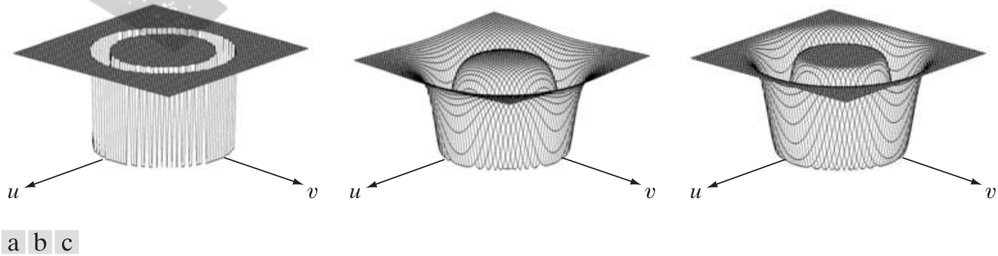


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

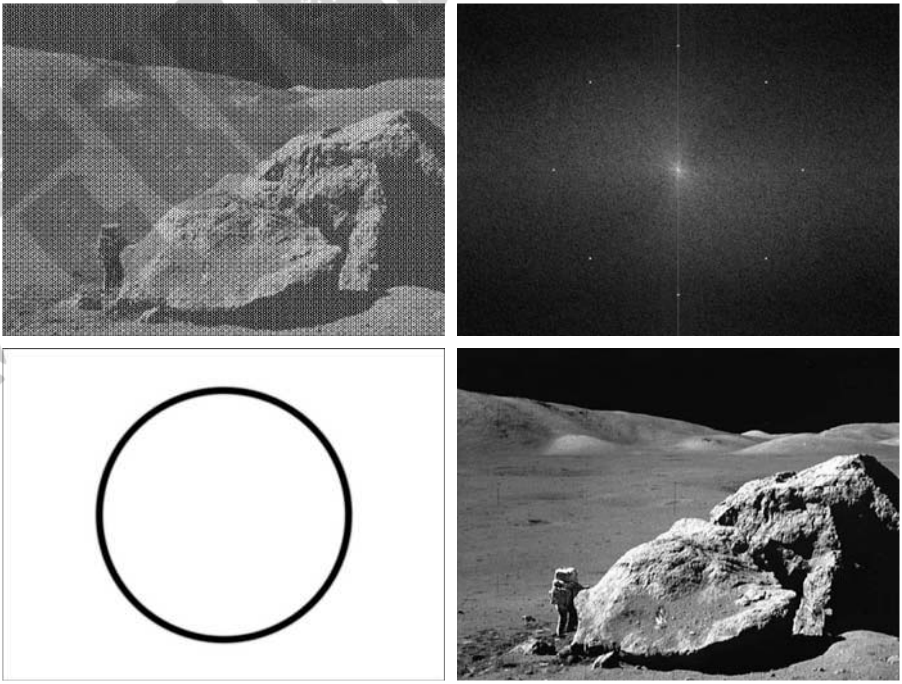
Figure 5.16(a), which is the same as Fig. 5.5(a), shows an image heavily corrupted by sinusoidal noise of various frequencies. The noise components are easily seen as symmetric pairs of bright dots in the Fourier spectrum shown in Fig. 5.16(b). In this example, the components lie on an approximate circle about the origin of the transform, so a circularly symmetric bandreject filter is a good choice. Figure 5.16(c) shows a Butterworth bandreject filter of order 4, with the appropriate radius and width to enclose completely the noise impulses. Since it is desirable in general to remove as little as possible from the transform, sharp, narrow filters are common in bandreject filtering. The result of filtering Fig. 5.16(a) with this filter is shown in Fig. 5.16(d). The improvement is quite evident. Even small details and textures were restored effectively by this simple filtering approach. It is worth noting also that it would not be possible to get equivalent results by a direct spatial domain filtering approach using small convolution masks. ■

5.4.2 Bandpass Filters

A *bandpass* filter performs the opposite operation of a bandreject filter. We showed in Section 4.10.1 how the transfer function $H_{\mathrm{BP}}(u, v)$ of a bandpass filter is obtained from a corresponding bandreject filter with transfer function $H_{\mathrm{BR}}(u, v)$ by using the equation

$$H_{\mathrm{BP}}(u, v) = 1 - H_{\mathrm{BR}}(u, v) \tag{5.4-1}$$

It is left as an exercise (Problem 5.12) to derive expressions for the bandpass filters corresponding to the bandreject equations in Table 4.6.



a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

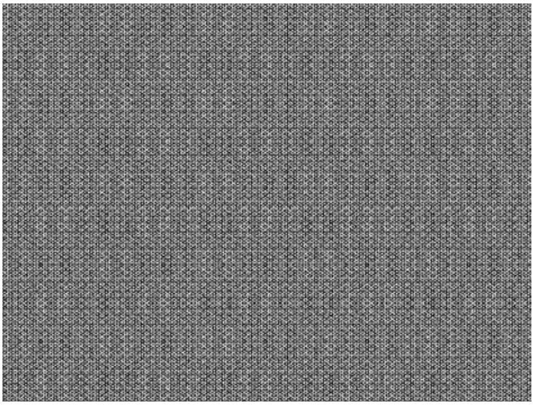


FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.

EXAMPLE 5.7: Bandpass filtering for extracting noise patterns.

■ Performing straight bandpass filtering on an image is not a common procedure because it generally removes too much image detail. However, bandpass filtering is quite useful in isolating the effects on an image caused by selected frequency bands. This is illustrated in Fig. 5.17. This image was generated by (1) using Eq. (5.4-1) to obtain the bandpass filter corresponding to the bandreject filter used in Fig. 5.16; and (2) taking the inverse transform of the bandpass-filtered transform. Most image detail was lost, but the information that remains is most useful, as it is clear that the noise pattern recovered using this method is quite close to the noise that corrupted the image in Fig. 5.16(a). In other words, bandpass filtering helped isolate the noise pattern. This is a useful result because it simplifies analysis of the noise, reasonably independently of image content. ■

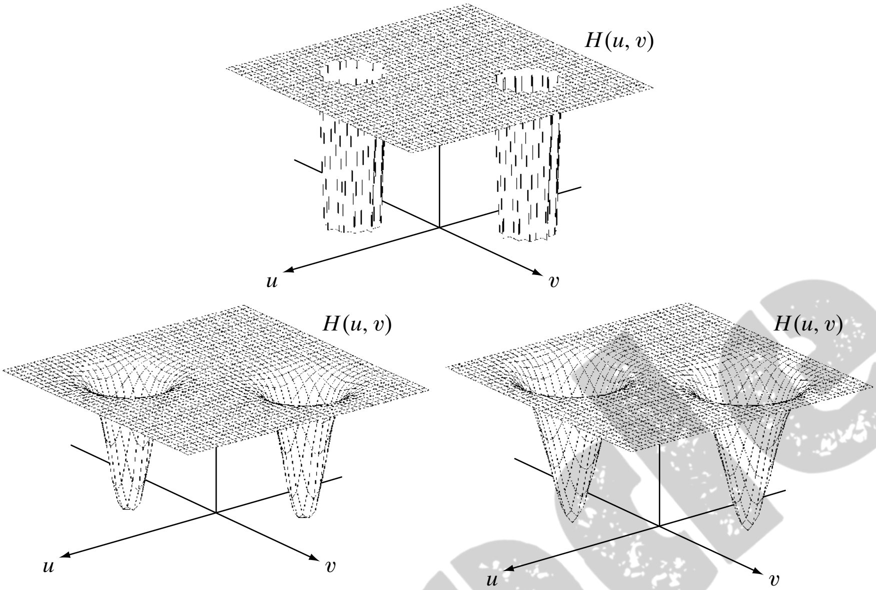
5.4.3 Notch Filters

A *notch* filter rejects (or passes) frequencies in predefined neighborhoods about a center frequency. Equations for notch filtering are detailed in Section 4.10.2. Figure 5.18 shows 3-D plots of ideal, Butterworth, and Gaussian notch (reject) filters. Due to the symmetry of the Fourier transform, notch filters must appear in symmetric pairs about the origin in order to obtain meaningful results. The one exception to this rule is if the notch filter is located at the origin, in which case it appears by itself. Although we show only one pair for illustrative purposes, the number of pairs of notch filters that can be implemented is arbitrary. The shape of the notch areas also can be arbitrary (e.g., rectangular).

As explained in Section 4.10.2, we can obtain notch filters that *pass*, rather than suppress, the frequencies contained in the notch areas. Since these filters perform exactly the opposite function as the notch reject filters, their transfer functions are given by

$$H_{\text{NP}}(u, v) = 1 - H_{\text{NR}}(u, v) \tag{5.4-2}$$

where $H_{\text{NP}}(u, v)$ is the transfer function of the notch pass filter corresponding to the notch reject filter with transfer function $H_{\text{NR}}(u, v)$.



a
b c

FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

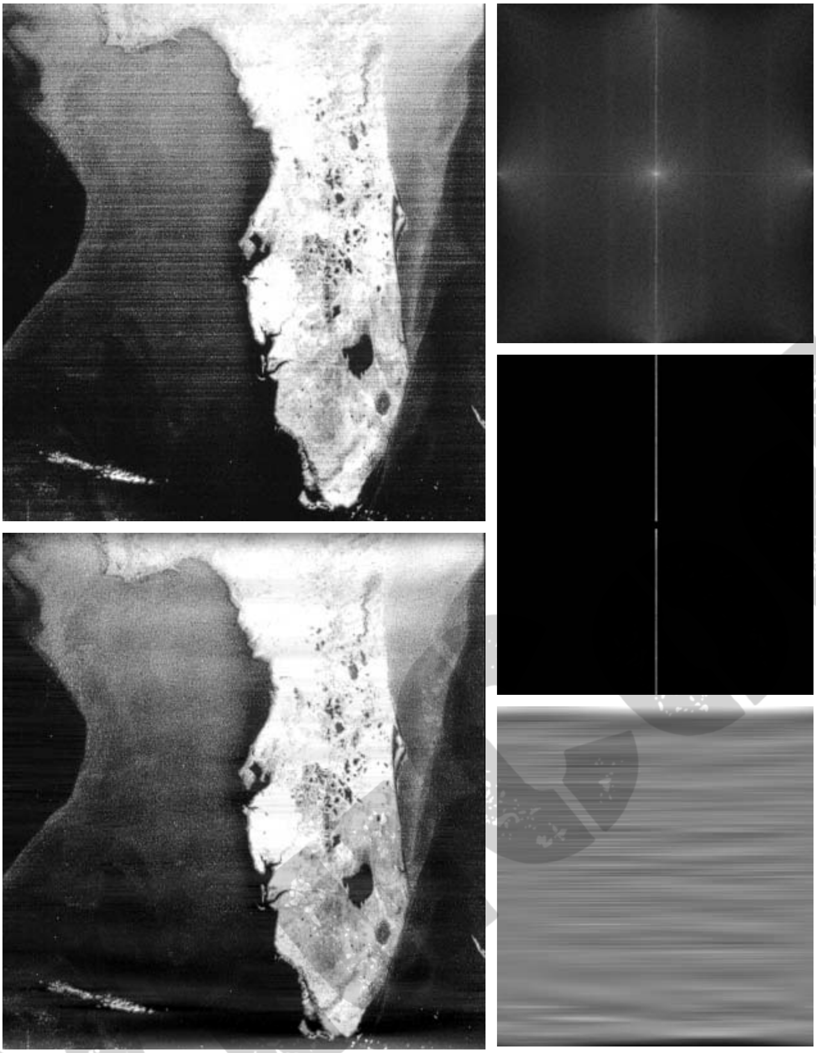
EXAMPLE 5.8: Removal of periodic noise by notch filtering.

■ Figure 5.19(a) shows the same image as Fig. 4.51(a). The notch filtering approach that follows reduces the noise in this image, without introducing the appreciable blurring we saw in Section 4.8.4. Unless blurring is desirable for reasons we discussed in that section, notch filtering is preferable if a suitable filter can be found.

Just by looking at the nearly horizontal lines of the noise pattern in Fig. 5.19(a), we expect its contribution in the frequency domain to be concentrated along the vertical axis. However, the noise is not dominant enough to have a clear pattern along this axis, as is evident from the spectrum shown in Fig. 5.19(b). We can get an idea of what the noise contribution looks like by constructing a simple ideal notch pass filter along the vertical axis of the Fourier transform, as shown in Fig. 5.19(c). The spatial representation of the noise pattern (inverse transform of the notch-pass–filtered result) is shown in Fig. 5.19(d). This noise pattern corresponds closely to the pattern in Fig. 5.19(a). Having thus constructed a suitable notch pass filter that isolates the noise to a reasonable degree, we can obtain the corresponding notch reject filter from Eq. (5.4-2). The result of processing the image with the notch reject filter is shown in Fig. 5.19(e). This image contains significantly fewer visible noise scan lines than Fig. 5.19(a). ■

5.4.4 Optimum Notch Filtering

Figure 5.20(a), another example of periodic image degradation, shows a digital image of the Martian terrain taken by the *Mariner 6* spacecraft. The interference pattern is somewhat similar to the one in Fig. 5.16(a), but the former pattern is considerably more subtle and, consequently, harder to detect in the frequency plane. Figure 5.20(b) shows the Fourier spectrum of the image in



a b
c
e d

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines. (b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

question. The starlike components were caused by the interference, and several pairs of components are present, indicating that the pattern contains more than just one sinusoidal component.

When several interference components are present, the methods discussed in the preceding sections are not always acceptable because they may remove too much image information in the filtering process (a highly undesirable feature when images are unique and/or expensive to acquire). In addition, the interference components generally are not single-frequency bursts. Instead, they tend to have broad skirts that carry information about the interference pattern. These skirts are not always easily detectable from the normal transform background. Alternative filtering methods that reduce the effect of

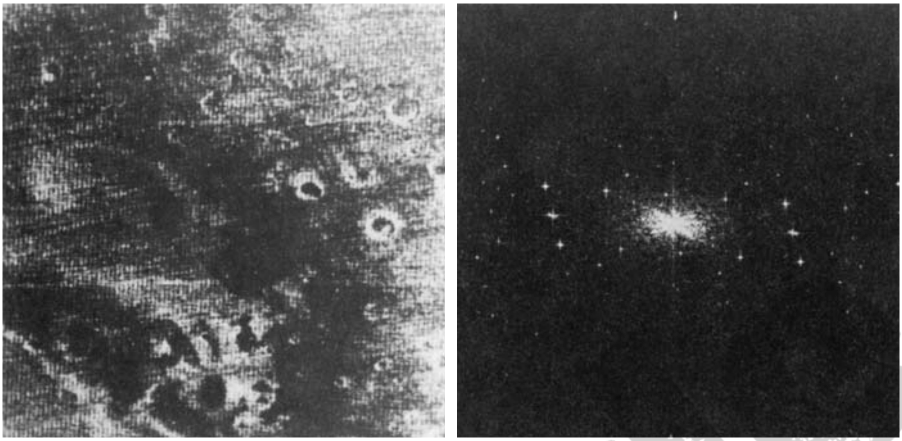


FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6.* (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

these degradations are quite useful in many applications. The method discussed here is optimum, in the sense that it minimizes local variances of the restored estimate $\hat{f}(x, y)$.

The procedure consists of first isolating the principal contributions of the interference pattern and then subtracting a variable, weighted portion of the pattern from the corrupted image. Although we develop the procedure in the context of a specific application, the basic approach is quite general and can be applied to other restoration tasks in which multiple periodic interference is a problem.

The first step is to extract the principal frequency components of the interference pattern. As before, this can be done by placing a notch pass filter, $H_{\text{NP}}(u, v)$, at the location of each spike. If the filter is constructed to pass only components associated with the interference pattern, then the Fourier transform of the interference noise pattern is given by the expression

$$N(u, v) = H_{\text{NP}}(u, v)G(u, v) \tag{5.4-3}$$

where, as usual, $G(u, v)$, denotes the Fourier transform of the corrupted image.

Formation of $H_{\text{NP}}(u, v)$ requires considerable judgment about what is or is not an interference spike. For this reason, the notch pass filter generally is constructed interactively by observing the spectrum of $G(u, v)$ on a display. After a particular filter has been selected, the corresponding pattern in the spatial domain is obtained from the expression

$$\eta(x, y) = \mathfrak{F}^{-1}\{H_{\text{NP}}(u, v)G(u, v)\} \tag{5.4-4}$$

Because the corrupted image is assumed to be formed by the addition of the uncorrupted image $f(x, y)$ and the interference, if $\eta(x, y)$ were known completely, subtracting the pattern from $g(x, y)$ to obtain $f(x, y)$ would be a simple matter. The problem, of course, is that this filtering procedure usually yields only an approximation of the true pattern. The effect of components not present in the estimate of $\eta(x, y)$ can be minimized instead by subtracting from $g(x, y)$ a *weighted* portion of $\eta(x, y)$ to obtain an estimate of $f(x, y)$:

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y) \tag{5.4-5}$$

where, as before, $\hat{f}(x, y)$ is the estimate of $f(x, y)$ and $w(x, y)$ is to be determined. The function $w(x, y)$ is called a *weighting* or *modulation* function, and the objective of the procedure is to select this function so that the result is optimized in some meaningful way. One approach is to select $w(x, y)$ so that the variance of the estimate $\hat{f}(x, y)$ is minimized over a specified neighborhood of every point (x, y).

Consider a neighborhood of size $(2a + 1)$ by $(2b + 1)$ about a point (x, y). The "local" variance of $\hat{f}(x, y)$ at coordinates (x, y) can be estimated from the samples, as follows:

$$\sigma^2(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left[\hat{f}(x + s, y + t) - \overline{\hat{f}}(x, y)\right]^2 \tag{5.4-6}$$

where $\overline{\hat{f}}(x, y)$ is the average value of $\hat{f}$ in the neighborhood; that is,

$$\overline{\hat{f}}(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \hat{f}(x + s, y + t) \tag{5.4-7}$$

Points on or near the edge of the image can be treated by considering partial neighborhoods or by padding the border with 0s.

Substituting Eq. (5.4-5) into Eq. (5.4-6) yields

$$\begin{aligned}\sigma^2(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \Big\{&[g(x + s, y + t) \\ &- w(x + s, y + t)\eta(x + s, y + t)] \\ &- [\overline{g}(x, y) - \overline{w(x, y)\eta(x, y)}]\Big\}^2\end{aligned} \tag{5.4-8}$$

Assuming that $w(x, y)$ remains essentially constant over the neighborhood gives the approximation

$$w(x + s, y + t) = w(x, y) \tag{5.4-9}$$

for $-a \leq s \leq a$ and $-b \leq t \leq b$. This assumption also results in the expression

$$\overline{w(x, y)\eta(x, y)} = w(x, y)\overline{\eta}(x, y) \tag{5.4-10}$$

in the neighborhood. With these approximations, Eq. (5.4-8) becomes

$$\begin{aligned}\sigma^2(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \Big\{&[g(x + s, y + t) \\ &- w(x, y)\eta(x + s, y + t)] \\ &- [\overline{g}(x, y) - w(x, y)\overline{\eta}(x, y)]\Big\}^2\end{aligned} \tag{5.4-11}$$

To minimize $\sigma^2(x, y)$, we solve

$$\frac{\partial \sigma^2(x, y)}{\partial w(x, y)} = 0 \tag{5.4-12}$$

for $w(x, y)$. The result is

$$w(x, y) = \frac{\overline{g(x, y)\eta(x, y)} - \overline{g}(x, y)\overline{\eta}(x, y)}{\overline{\eta^2}(x, y) - \overline{\eta}^2(x, y)} \tag{5.4-13}$$

To obtain the restored image $\hat{f}(x, y)$, we compute $w(x, y)$ from Eq. (5.4-13) and then use Eq. (5.4-5). As $w(x, y)$ is assumed to be constant in a neighborhood, computing this function for every value of x and y in the image is unnecessary. Instead, $w(x, y)$ is computed for *one* point in each nonoverlapping neighborhood (preferably the center point) and then used to process all the image points contained in that neighborhood.

EXAMPLE 5.9: Illustration of optimum notch filtering.

■ Figures 5.21 through 5.23 show the result of applying the preceding technique to the image in Fig. 5.20(a). This image is of size 512 × 512 pixels, and a neighborhood with $a = b = 15$ was selected. Figure 5.21 shows the Fourier spectrum of the corrupted image. The origin was not shifted to the center of the frequency plane in this particular case, so $u = v = 0$ is at the top left corner of the transform image in Fig. 5.21. Figure 5.22(a) shows the spectrum of $N(u, v)$, where only the noise spikes are present. Figure 5.22(b) shows the interference pattern $\eta(x, y)$ obtained by taking the inverse Fourier transform of $N(u, v)$. Note the similarity between this pattern and the structure of the noise present in Fig. 5.20(a). Finally, Fig. 5.23 shows the processed image obtained by using Eq. (5.4-5). The periodic interference was removed for all practical purposes. ■

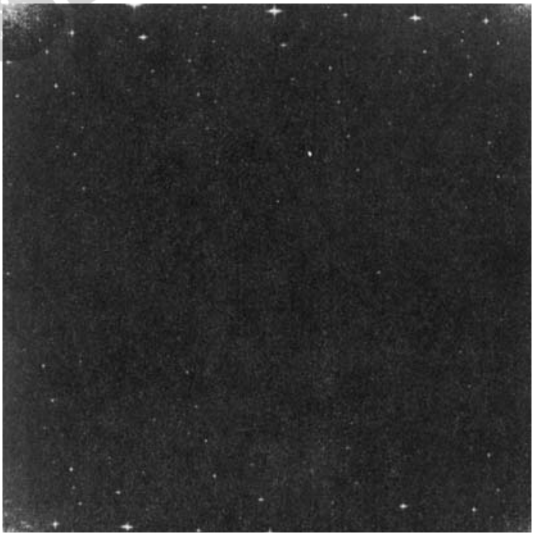


FIGURE 5.21 Fourier spectrum (without shifting) of the image shown in Fig. 5.20(a). (Courtesy of NASA.)

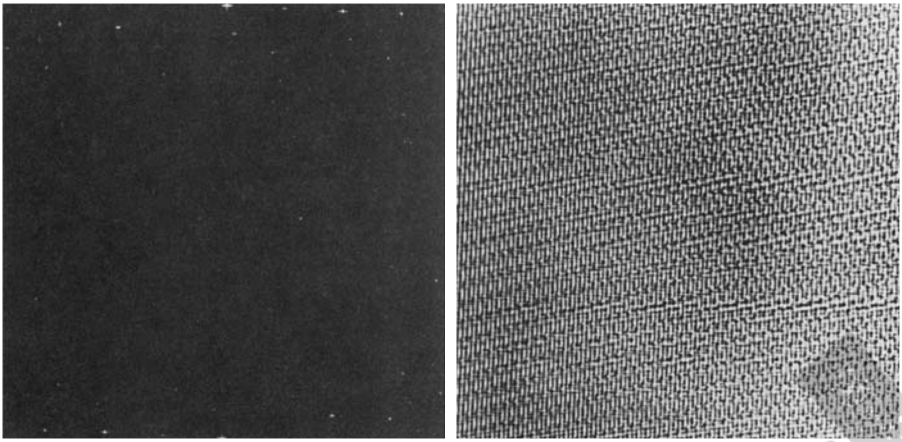


FIGURE 5.22 (a) Fourier spectrum of $N(u, v)$, and (b) corresponding noise interference pattern $\eta(x, y)$. (Courtesy of NASA.)

5.5 Linear, Position-Invariant Degradations

The input-output relationship in Fig. 5.1 before the restoration stage is expressed as

$$g(x, y) = H[f(x, y)] + \eta(x, y) \tag{5.5-1}$$

For the moment, let us assume that $\eta(x, y) = 0$ so that $g(x, y) = H[f(x, y)]$. Based on the discussion in Section 2.6.2, H is *linear* if

$$H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)] \tag{5.5-2}$$

where a and b are scalars and $f_1(x, y)$ and $f_2(x, y)$ are any two input images.

If $a = b = 1$, Eq. (5.5-2) becomes

Consult the book Web site for a brief review of linear system theory.

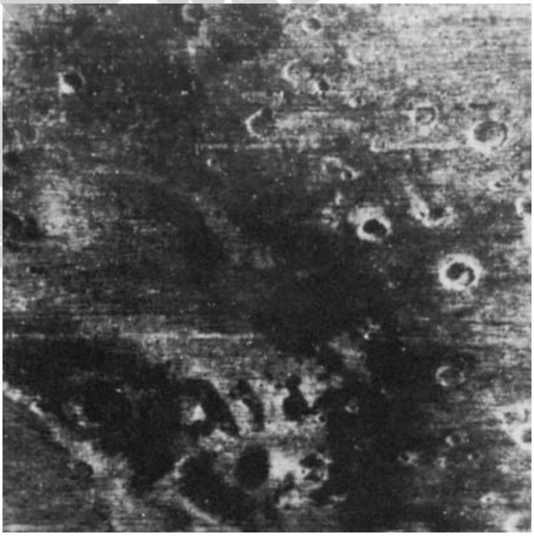


FIGURE 5.23 Processed image. (Courtesy of NASA.)

$$H[f_1(x, y) + f_2(x, y)] = H[f_1(x, y)] + H[f_2(x, y)] \tag{5.5-3}$$

which is called the property of *additivity*. This property simply says that, if H is a linear operator, the response to a sum of two inputs is equal to the sum of the two responses.

With $f_2(x, y) = 0$, Eq. (5.5-2) becomes

$$H[af_1(x, y)] = aH[f_1(x, y)] \tag{5.5-4}$$

which is called the property of *homogeneity*. It says that the response to a constant multiple of any input is equal to the response to that input multiplied by the same constant. Thus a linear operator possesses both the property of additivity and the property of homogeneity.

An operator having the input-output relationship $g(x, y) = H[f(x, y)]$ is said to be *position* (or *space*) *invariant* if

$$H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta) \tag{5.5-5}$$

for any $f(x, y)$ and any α and β. This definition indicates that the response at any point in the image depends only on the *value* of the input at that point, not on its *position*.

See the footnote in page 369 regarding continuous and discrete variables.

With a slight (but equivalent) change in notation in the definition of the impulse in Eq. (4.5-3), $f(x, y)$ can be expressed as:

$$f(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)\delta(x - \alpha, y - \beta)\, d\alpha\, d\beta \tag{5.5-6}$$

Assume again for a moment that $\eta(x, y) = 0$. Then, substitution of Eq. (5.5-6) into Eq. (5.5-1) results in the expression

$$g(x, y) = H[f(x, y)] = H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)\delta(x - \alpha, y - \beta)\, d\alpha\, d\beta\right] \tag{5.5-7}$$

If H is a linear operator and we extend the additivity property to integrals, then

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H[f(\alpha, \beta)\delta(x - \alpha, y - \beta)]\, d\alpha\, d\beta \tag{5.5-8}$$

Because $f(\alpha, \beta)$ is independent of x and y, and using the homogeneity property, it follows that

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)H[\delta(x - \alpha, y - \beta)]\, d\alpha\, d\beta \tag{5.5-9}$$

The term

$$h(x, \alpha, y, \beta) = H[\delta(x - \alpha, y - \beta)] \tag{5.5-10}$$

is called the *impulse response* of H. In other words, if $\eta(x, y) = 0$ in Eq. (5.5-1), then $h(x, \alpha, y, \beta)$ is the response of H to an impulse at coordinates (x, y). In optics, the impulse becomes a point of light and $h(x, \alpha, y, \beta)$ is commonly referred to as the *point spread function* (PSF). This name arises from the fact that all physical optical systems blur (spread) a point of light to some degree, with the amount of blurring being determined by the quality of the optical components.

Substituting Eq. (5.5-10) into Eq. (5.5-9) yields the expression

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta)\, d\alpha\, d\beta \tag{5.5-11}$$

which is called the *superposition* (or *Fredholm*) *integral of the first kind*. This expression is a fundamental result that is at the core of linear system theory. It states that if the response of H to an impulse is known, the response to *any* input $f(\alpha, \beta)$ can be calculated by means of Eq. (5.5-11). In other words, a linear system H is completely characterized by its impulse response.

If H is position invariant, then, from Eq. (5.5-5),

$$H[\delta(x - \alpha, y - \beta)] = h(x - \alpha, y - \beta) \tag{5.5-12}$$

Equation (5.5-11) reduces in this case to

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta)\, d\alpha\, d\beta \tag{5.5-13}$$

This expression is the *convolution integral* introduced for one variable in Eq. (4.2-20) and extended to 2-D in Problem 4.11. This integral tells us that knowing the impulse response of a linear system allows us to compute its response, g, to *any* input f. The result is simply the convolution of the impulse response and the input function.

In the presence of additive noise, the expression of the linear degradation model [Eq. (5.5-11)] becomes

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta)\, d\alpha\, d\beta + \eta(x, y) \tag{5.5-14}$$

If H is position invariant, Eq. (5.5-14) becomes

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta)\, d\alpha\, d\beta + \eta(x, y) \tag{5.5-15}$$

The values of the noise term $\eta(x, y)$ are random, and are assumed to be independent of position. Using the familiar notation for convolution, we can write Eq. (5.5-15) as

$$g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \tag{5.5-16}$$

or, based on the convolution theorem (see Section 4.6.6), we can express it in the frequency domain as

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \tag{5.5-17}$$

These two expressions agree with Eqs. (5.1-1) and (5.1-2). Keep in mind that, for discrete quantities, all products are term by term. For example, term ij of $H(u, v)F(u, v)$ is the product of term ij of $H(u, v)$ and term ij of $F(u, v)$.

In summary, the preceding discussion indicates that a linear, spatially-invariant degradation system with additive noise can be modeled in the spatial domain as the convolution of the degradation (point spread) function with an image, followed by the addition of noise. Based on the convolution theorem, the same process can be expressed in the frequency domain as the product of the transforms of the image and degradation, followed by the addition of the transform of the noise. When working in the frequency domain, we make use of an FFT algorithm, as discussed in Section 4.11. Keep in mind also the need for function padding in the implementation of discrete Fourier transforms, as outlined in Section 4.6.6.

Many types of degradations can be approximated by linear, position-invariant processes. The advantage of this approach is that the extensive tools of linear system theory then become available for the solution of image restoration problems. Nonlinear and position-dependent techniques, although more general (and usually more accurate), introduce difficulties that often have no known solution or are very difficult to solve computationally. This chapter focuses on linear, space-invariant restoration techniques. Because degradations are modeled as being the result of convolution, and restoration seeks to find filters that apply the process in reverse, the term *image deconvolution* is used frequently to signify linear image restoration. Similarly, the filters used in the restoration process often are called *deconvolution filters.*

5.6 Estimating the Degradation Function

There are three principal ways to estimate the degradation function for use in image restoration: (1) observation, (2) experimentation, and (3) mathematical modeling. These methods are discussed in the following sections. The process of restoring an image by using a degradation function that has been estimated in some way sometimes is called *blind deconvolution,* due to the fact that the true degradation function is seldom known completely.

5.6.1 Estimation by Image Observation

Suppose that we are given a degraded image without any knowledge about the degradation function H. Based on the assumption that the image was degraded by a linear, position-invariant process, one way to estimate H is to gather information from the image itself. For example, if the image is blurred, we can look at a small rectangular section of the image containing sample structures, like part of an object and the background. In order to reduce the effect of noise, we would look for an area in which the signal content is strong (e.g., an area of high contrast). The next step would be to process the subimage to arrive at a result that is as unblurred as possible. For example, we can do this by sharpening the subimage with a sharpening filter and even by processing small areas by hand.

Let the observed subimage be denoted by $g_s(x, y)$, and let the processed subimage (which in reality is our estimate of the original image in that area) be denoted by $\hat{f}_s(x, y)$. Then, assuming that the effect of noise is negligible because of our choice of a strong-signal area, it follows from Eq. (5.5-17) that

$$H_s(u, v) = \frac{G_s(u, v)}{\hat{F}_s(u, v)} \tag{5.6-1}$$

From the characteristics of this function, we then deduce the complete degradation function $H(u, v)$ based on our assumption of position invariance. For example, suppose that a radial plot of $H_s(u, v)$ has the approximate shape of a Gaussian curve. We can use that information to construct a function $H(u, v)$ on a larger scale, but having the same basic shape. We then use $H(u, v)$ in one of the restoration approaches to be discussed in the following sections. Clearly, this is a laborious process used only in very specific circumstances such as, for example, restoring an old photograph of historical value.

5.6.2 Estimation by Experimentation

If equipment similar to the equipment used to acquire the degraded image is available, it is possible in principle to obtain an accurate estimate of the degradation. Images similar to the degraded image can be acquired with various system settings until they are degraded as closely as possible to the image we wish to restore. Then the idea is to obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the same system settings. As noted in Section 5.5, a linear, space-invariant system is characterized completely by its impulse response.

An impulse is simulated by a bright dot of light, as bright as possible to reduce the effect of noise to negligible values. Then, recalling that the Fourier transform of an impulse is a constant, it follows from Eq. (5.5-17) that

$$H(u, v) = \frac{G(u, v)}{A} \tag{5.6-2}$$

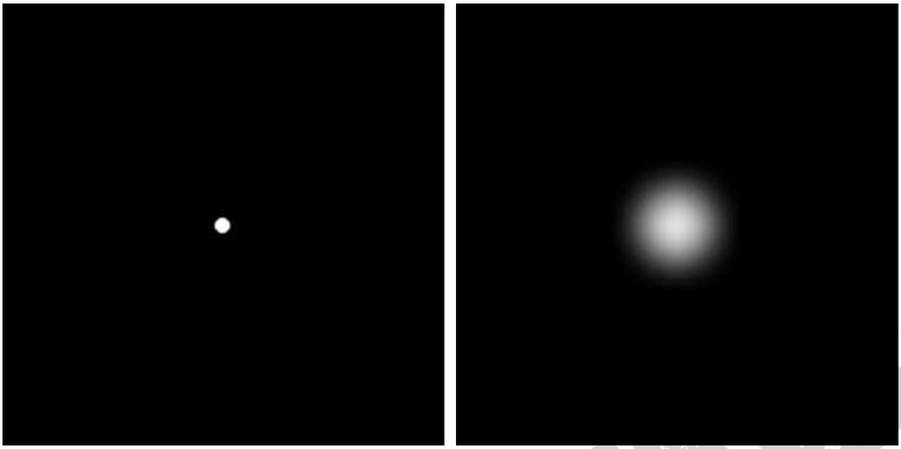
where, as before, $G(u, v)$ is the Fourier transform of the observed image and A is a constant describing the strength of the impulse. Figure 5.24 shows an example.

5.6.3 Estimation by Modeling

Degradation modeling has been used for many years because of the insight it affords into the image restoration problem. In some cases, the model can even take into account environmental conditions that cause degradations. For example, a degradation model proposed by Hufnagel and Stanley [1964] is based on the physical characteristics of atmospheric turbulence. This model has a familiar form:

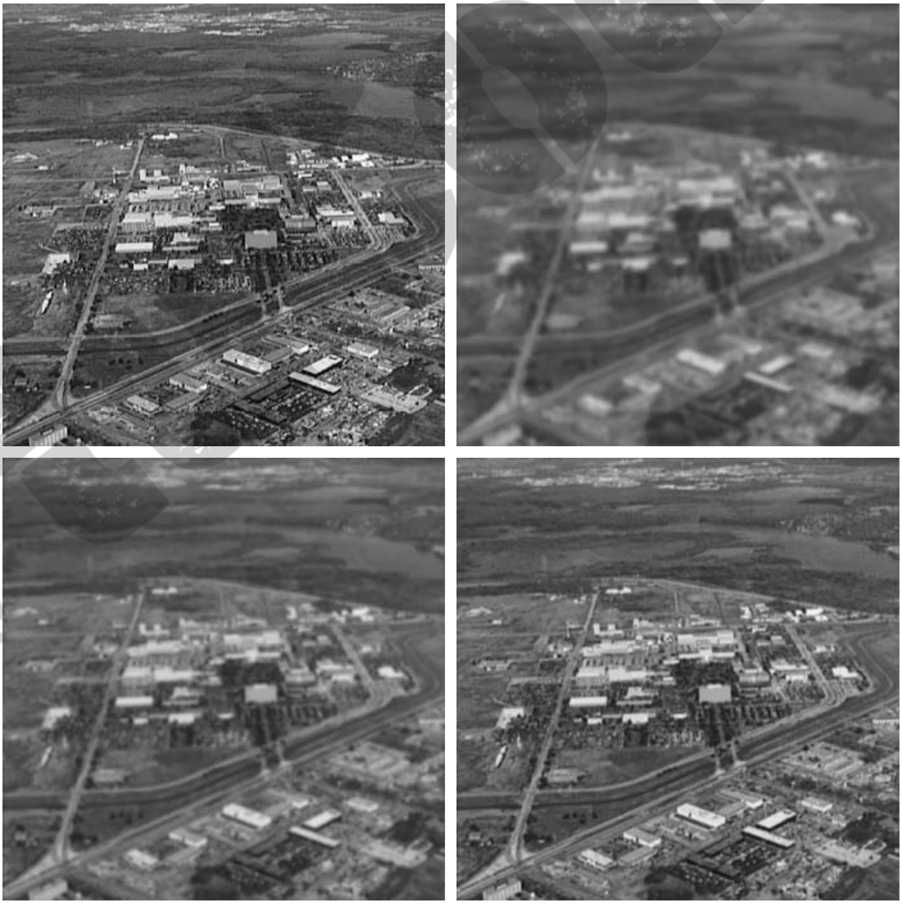
$$H(u, v) = e^{-k(u^2+v^2)^{5/6}} \tag{5.6-3}$$

where k is a constant that depends on the nature of the turbulence. With the exception of the 5/6 power on the exponent, this equation has the same form as the Gaussian lowpass filter discussed in Section 4.8.3. In fact, the Gaussian LPF is used sometimes to model mild, uniform blurring. Figure 5.25 shows examples



a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.



a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

obtained by simulating blurring an image using Eq. (5.6-3) with values $k = 0.0025$ (severe turbulence), $k = 0.001$ (mild turbulence), and $k = 0.00025$ (low turbulence). All images are of size 480×480 pixels.

Another major approach in modeling is to derive a mathematical model starting from basic principles. We illustrate this procedure by treating in some detail the case in which an image has been blurred by uniform linear motion between the image and the sensor during image acquisition. Suppose that an image $f(x, y)$ undergoes planar motion and that $x_0(t)$ and $y_0(t)$ are the time-varying components of motion in the x- and y-directions, respectively. The total exposure at any point of the recording medium (say, film or digital memory) is obtained by integrating the instantaneous exposure over the time interval during which the imaging system shutter is open.

Assuming that shutter opening and closing takes place instantaneously, and that the optical imaging process is perfect, isolates the effect of image motion. Then, if T is the duration of the exposure, it follows that

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)]\, dt \tag{5.6-4}$$

where $g(x, y)$ is the blurred image.

From Eq. (4.5-7), the Fourier transform of Eq. (5.6-4) is

$$\begin{aligned} G(u, v) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(ux+vy)}\, dx\, dy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f[x - x_0(t), y - y_0(t)]\, dt\right] e^{-j2\pi(ux+vy)}\, dx\, dy \end{aligned} \tag{5.6-5}$$

Reversing the order of integration allows Eq. (5.6-5) to be expressed in the form

$$G(u, v) = \int_0^T \left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)] e^{-j2\pi(ux+vy)}\, dx\, dy\right] dt \tag{5.6-6}$$

The term inside the outer brackets is the Fourier transform of the displaced function $f[x - x_0(t), y - y_0(t)]$. Using Eq. (4.6-4) then yields the expression

$$\begin{aligned} G(u, v) &= \int_0^T F(u, v) e^{-j2\pi[ux_0(t)+vy_0(t)]}\, dt \\ &= F(u, v)\int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\, dt \end{aligned} \tag{5.6-7}$$

where the last step follows from the fact that $F(u, v)$ is independent of t.

By defining

$$H(u, v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\, dt \tag{5.6-8}$$

Eq. (5.6-7) can be expressed in the familiar form

$$G(u, v) = H(u, v)F(u, v) \tag{5.6-9}$$

If the motion variables $x_0(t)$ and $y_0(t)$ are known, the transfer function $H(u, v)$ can be obtained directly from Eq. (5.6-8). As an illustration, suppose that the image in question undergoes uniform linear motion in the x-direction only, at a rate given by $x_0(t) = at/T$. When $t = T$, the image has been displaced by a total distance a. With $y_0(t) = 0$, Eq. (5.6-8) yields

$$\begin{aligned} H(u, v) &= \int_0^T e^{-j2\pi u x_0(t)}\, dt \\ &= \int_0^T e^{-j2\pi uat/T}\, dt \\ &= \frac{T}{\pi ua} \sin(\pi ua) e^{-j\pi ua} \end{aligned} \tag{5.6-10}$$

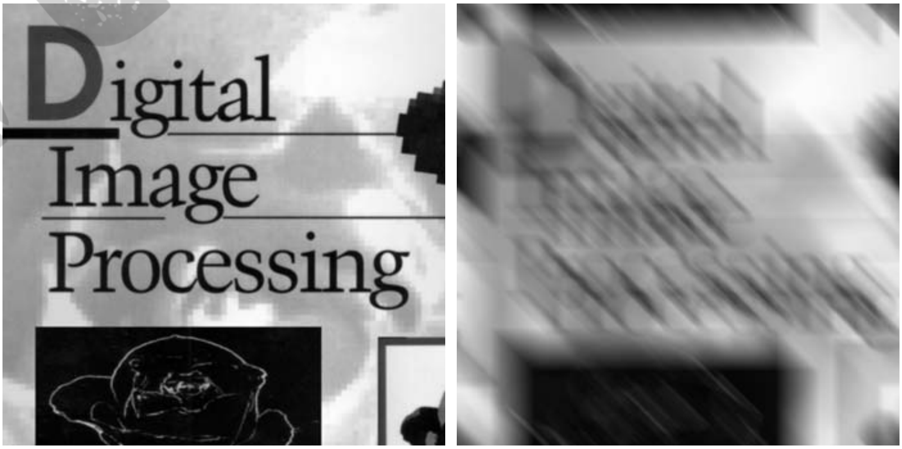
Observe that H vanishes at values of u given by $u = n/a$, where n is an integer. If we allow the y-component to vary as well, with the motion given by $y_0 = bt/T$, then the degradation function becomes

As explained at the end of Table 4.3, we sample Eg (5.6-11) in u and v to generate a discrete filter.

$$H(u, v) = \frac{T}{\pi(ua + vb)} \sin[\pi(ua + vb)] e^{-j\pi(ua+vb)} \tag{5.6-11}$$

EXAMPLE 5.10: Image blurring due to motion.

■ Figure 5.26(b) is an image blurred by computing the Fourier transform of the image in Fig. 5.26(a), multiplying the transform by $H(u, v)$ from Eq. (5.6-11), and taking the inverse transform. The images are of size 688 × 688 pixels, and the parameters used in Eq. (5.6-11) were $a = b = 0.1$ and $T = 1$. As discussed in Sections 5.8 and 5.9, recovery of the original image from its blurred counterpart presents some interesting challenges, particularly when noise is present in the degraded image. ■



a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

5.7 Inverse Filtering

The material in this section is our first step in studying restoration of images degraded by a degradation function H, which is given or obtained by a method such as those discussed in the previous section. The simplest approach to restoration is direct inverse filtering, where we compute an estimate, $\hat{F}(u, v)$, of the transform of the original image simply by dividing the transform of the degraded image, $G(u, v)$, by the degradation function:

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} \tag{5.7-1}$$

The division is an array operation, as defined in Section 2.6.1 and in connection with Eq. (5.5-17). Substituting the right side of Eq. (5.1-2) for $G(u, v)$ in Eq. (5.7-1) yields

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)} \tag{5.7-2}$$

This is an interesting expression. It tells us that even if we know the degradation function we cannot recover the undegraded image [the inverse Fourier transform of $F(u, v)$] exactly because $N(u, v)$ is not known. There is more bad news. If the degradation function has zero or very small values, then the ratio $N(u, v)/H(u, v)$ could easily dominate the estimate $\hat{F}(u, v)$. This, in fact, is frequently the case, as will be demonstrated shortly.

One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin. From the discussion of Eq. (4.6-21) we know that $H(0, 0)$ is usually the highest value of $H(u, v)$ in the frequency domain. Thus, by limiting the analysis to frequencies near the origin, we reduce the probability of encountering zero values. This approach is illustrated in the following example.

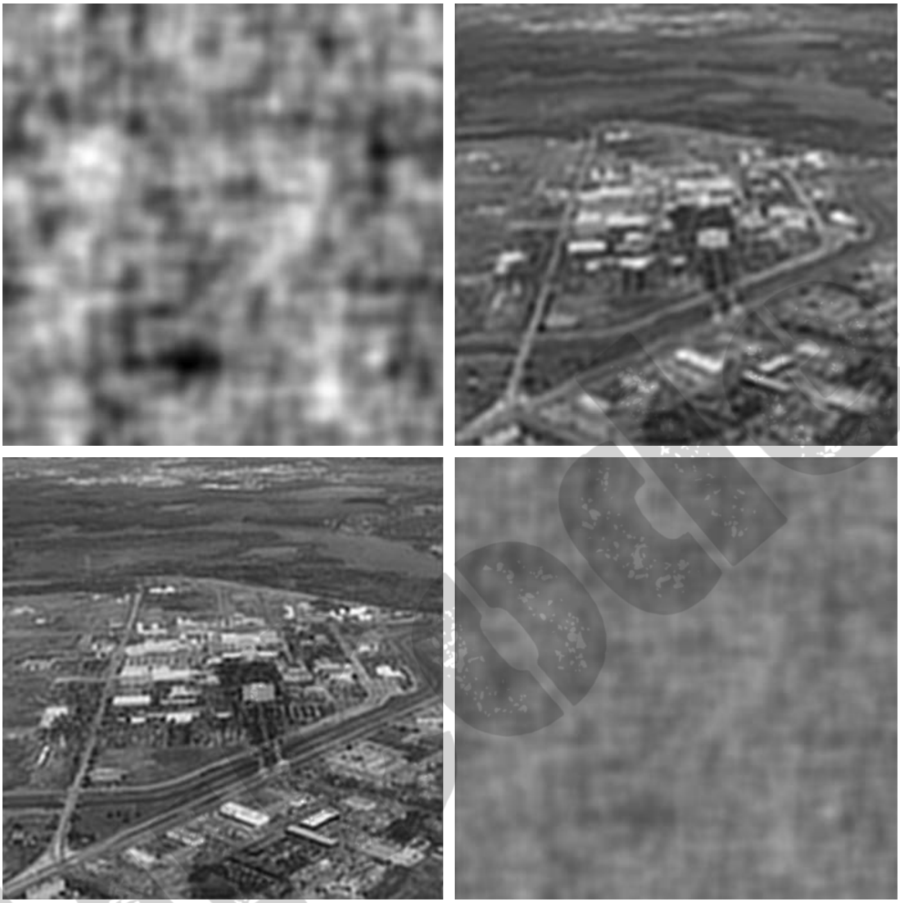
EXAMPLE 5.11: Inverse filtering.

■ The image in Fig. 5.25(b) was inverse filtered with Eq. (5.7-1) using the exact inverse of the degradation function that generated that image. That is, the degradation function used was

$$H(u, v) = e^{-k[(u-M/2)^2+(v-N/2)^2]^{5/6}}$$

with $k = 0.0025$. The $M/2$ and $N/2$ constants are offset values; they center the function so that it will correspond with the centered Fourier transform, as discussed on numerous occasions in the previous chapter. In this case, $M = N = 480$. We know that a Gaussian-shape function has no zeros, so that will not be a concern here. However, in spite of this, the degradation values became so small that the result of full inverse filtering [Fig. 5.27(a)] is useless. The reasons for this poor result are as discussed in connection with Eq. (5.7-2).

Figures 5.27(b) through (d) show the results of cutting off values of the ratio $G(u, v)/H(u, v)$ outside a radius of 40, 70, and 85, respectively. The cut off was implemented by applying to the ratio a Butterworth lowpass function of order 10. This provided a sharp (but smooth) transition at the



a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with *H* cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

desired radius. Radii near 70 yielded the best visual results [Fig. 5.27(c)]. Radius values below that tended toward blurred images, as illustrated in Fig. 5.27(b), which was obtained using a radius of 40. Values above 70 started to produce degraded images, as illustrated in Fig. 5.27(d), which was obtained using a radius of 85. The image content is almost visible in this image behind a "curtain" of noise, but the noise definitely dominates the result. Further increases in radius values produced images that looked more and more like Fig. 5.27(a). ■

The results in the preceding example are illustrative of the poor performance of direct inverse filtering in general. The basic theme of the three sections that follow is how to improve on direct inverse filtering.

5.8 Minimum Mean Square Error (Wiener) Filtering

The inverse filtering approach discussed in the previous section makes no explicit provision for handling noise. In this section, we discuss an approach that incorporates both the degradation function and statistical characteristics of noise into the restoration process. The method is founded on considering images and noise as random variables, and the objective is to find an estimate $\hat{f}$ of the uncorrupted image f such that the mean square error between them is minimized. This error measure is given by

Note that entire images are being considered random variables, as discussed at the end of Section 2.6.8.

$$e^2 = E\{(f - \hat{f})^2\} \tag{5.8-1}$$

where $E\{\cdot\}$ is the expected value of the argument. It is assumed that the noise and the image are uncorrelated; that one or the other has zero mean; and that the intensity levels in the estimate are a linear function of the levels in the degraded image. Based on these conditions, the minimum of the error function in Eq. (5.8-1) is given in the frequency domain by the expression

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v)\\ &= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v) \qquad (5.8\text{-}2)\\ &= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)\end{aligned}$$

where we used the fact that the product of a complex quantity with its conjugate is equal to the magnitude of the complex quantity squared. This result is known as the *Wiener filter*, after N. Wiener [1942], who first proposed the concept in the year shown. The filter, which consists of the terms inside the brackets, also is commonly referred to as the *minimum mean square error filter* or the *least square error filter*. We include references at the end of the chapter to sources containing detailed derivations of the Wiener filter. Note from the first line in Eq. (5.8-2) that the Wiener filter does not have the same problem as the inverse filter with zeros in the degradation function, unless the entire denominator is zero for the same value(s) of u and v.

The terms in Eq. (5.8-2) are as follows:

$H(u,v)$ = degradation function
$H^*(u,v)$ = complex conjugate of $H(u,v)$
$|H(u,v)|^2 = H^*(u,v)H(u,v)$
$S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise [see Eq. (4.6–18)][†]
$S_f(u,v) = |F(u,v)|^2$ = power spectrum of the undegraded image

[†] The term $|N(u,v)|^2$ also is referred to as the *autocorrelation* of the noise. This terminology comes from the correlation theorem (first line of entry 7 in Table 4.3). When the two functions are the same, correlation becomes *autocorrelation* and the right side of that entry becomes $N^*(u,v)N(u,v)$, which is equal to $|N(u,v)|^2$. Similar comments apply to $|F(u,v)|^2$, which is the autocorrelation of the image. We discuss correlation in more detail in Chapter 12.

As before, $H(u, v)$ is the transform of the degradation function and $G(u, v)$ is the transform of the degraded image. The restored image in the spatial domain is given by the inverse Fourier transform of the frequency-domain estimate $\hat{F}(u, v)$. Note that if the noise is zero, then the noise power spectrum vanishes and the Wiener filter reduces to the inverse filter.

A number of useful measures are based on the power spectra of noise and of the undegraded image. One of the most important is the *signal-to-noise ratio*, approximated using frequency domain quantities such as

$$\text{SNR} = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u, v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u, v)|^2} \tag{5.8-3}$$

This ratio gives a measure of the level of information bearing signal power (i.e., of the original, undegraded image) to the level of noise power. Images with low noise tend to have a high SNR and, conversely, the same image with a higher level of noise has a lower SNR. This ratio by itself is of limited value, but it is an important metric used in characterizing the performance of restoration algorithms.

The *mean square error* given in statistical form in Eq. (5.8-1) can be approximated also in terms a summation involving the original and restored images:

$$\text{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x, y) - \hat{f}(x, y)]^2 \tag{5.8-4}$$

In fact, if one considers the restored image to be "signal" and the difference between this image and the original to be noise, we can define a signal-to-noise ratio in the spatial domain as

$$\text{SNR} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x, y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x, y) - \hat{f}(x, y)]^2} \tag{5.8-5}$$

The closer f and $\hat{f}$ are, the larger this ratio will be. Sometimes the square root of these measures is used instead, in which case they are referred to as the *root-mean-square-signal-to-noise ratio* and the *root-mean-square-error*, respectively. As we have mentioned several times before, keep in mind that quantitative metrics do not necessarily relate well to perceived image quality.

When we are dealing with spectrally white noise, the spectrum $|N(u, v)|^2$ is a constant, which simplifies things considerably. However, the power spectrum of the undegraded image seldom is known. An approach used frequently when these quantities are not known or cannot be estimated is to approximate Eq. (5.8-2) by the expression

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right]G(u,v) \qquad (5.8\text{-}6)$$

where K is a specified constant that is added to all terms of $|H(u,v)|^2$. The following examples illustrate the use of this expression.

EXAMPLE 5.12: Comparison of inverse and Wiener filtering.

■ Figure 5.28 illustrates the advantage of Wiener filtering over direct inverse filtering. Figure 5.28(a) is the full inverse-filtered result from Fig. 5.27(a). Similarly, Fig. 5.28(b) is the radially limited inverse filter result of Fig, 5.27(c). These images are duplicated here for convenience in making comparisons. Figure 5.28(c) shows the result obtained using Eq. (5.8-6) with the degradation function used in Example 5.11. The value of K was chosen interactively to yield the best visual results. The advantage of Wiener filtering over the direct inverse approach is evident in this example. By comparing Figs. 5.25(a) and 5.28(c), we see that the Wiener filter yielded a result very close in appearance to the original image. ■

EXAMPLE 5.13: Further comparisons of Wiener filtering.

■ The first row of Fig. 5.29 shows, from left to right, the blurred image of Fig. 5.26(b) heavily corrupted by additive Gaussian noise of zero mean and variance of 650; the result of direct inverse filtering; and the result of Wiener filtering. The Wiener filter of Eq. (5.8-6) was used, with $H(u,v)$ from Example 5.10, and with K chosen interactively to give the best possible visual result. As expected, the inverse filter produced an unusable image. Note that the noise in the inverse filtered image is so strong that its structure is in the direction of the *deblurring* filter. The Wiener filter result is by no means perfect, but it does give us a hint as to image content. With some difficulty, the text is readable.

The second row of Fig. 5.29 shows the same sequence, but with the level of noise variance reduced by one order of magnitude. This reduction had little effect on the inverse filter, but the Wiener results are considerably improved. The text

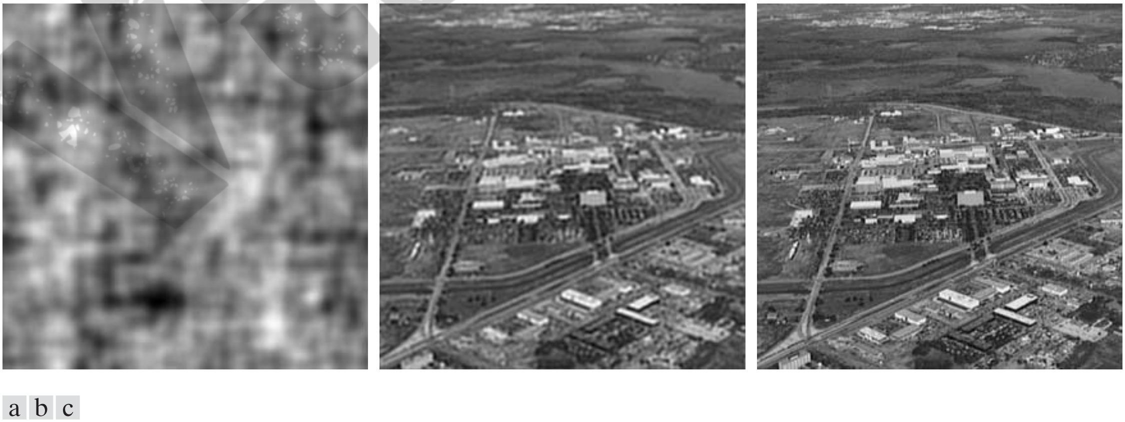


FIGURE 5.28 Comparison of inverse and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

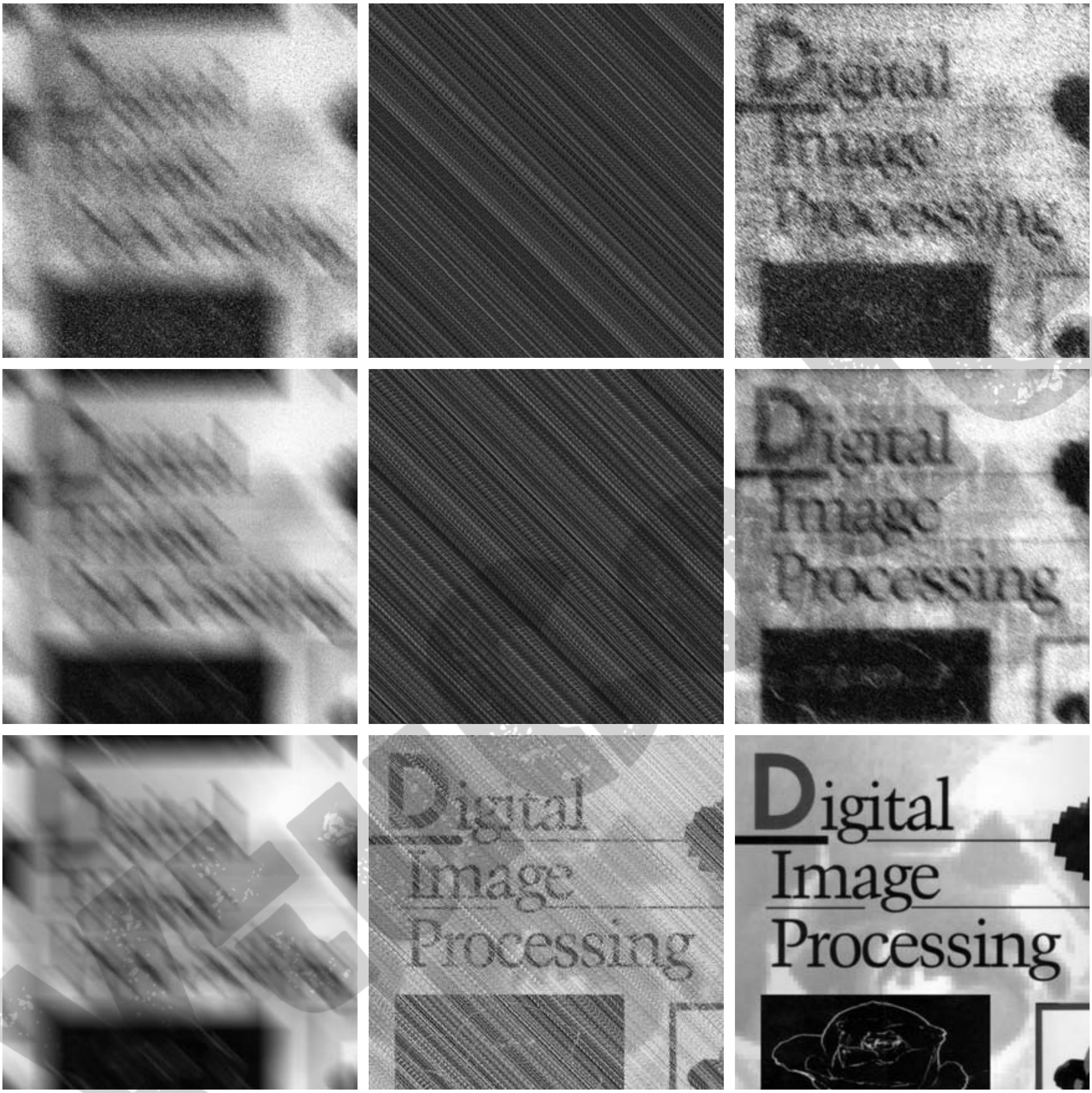


FIGURE 5.29 (a) 8-bit image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

now is much easier to read. In the third row of Fig. 5.29, the noise variance has decreased more than five orders of magnitude from the first row. In fact, image 5.29(g) has no visible noise. The inverse filter result is interesting in this case. The noise is still quite visible, but the text can be seen through a "curtain" of noise. This is a good example of the comments made regarding Eq. (5.7-2). In other words, as is evident in Fig. 5.29(h), the inverse filter was quite capable of essentially eliminating the blur in the image. However, the noise still dominates the result. If we could "look" behind the noise in Figs. 5.29(b) and (e), the characters also would show little blurring. The Wiener filter result in Fig. 5.29(i) is excellent, being quite close visually to the original image in Fig. 5.26(a). These types of results are representative of what is possible with Wiener filtering, as long as a reasonable estimate of the degradation function is available. ■

5.9 Constrained Least Squares Filtering

The problem of having to know something about the degradation function H is common to all methods discussed in this chapter. However, the Wiener filter presents an additional difficulty: The power spectra of the undegraded image and noise must be known. We showed in the previous section that it is possible to achieve excellent results using the approximation given in Eq. (5.8-6). However, a constant estimate of the ratio of the power spectra is not always a suitable solution.

The method discussed in this section requires knowledge of only the mean and variance of the noise. As discussed in Section 5.2.4, these parameters usually can be calculated from a given degraded image, so this is an important advantage. Another difference is that the Wiener filter is based on minimizing a statistical criterion and, as such, it is optimal in an average sense. The algorithm presented in this section has the notable feature that it yields an optimal result for *each* image to which it is applied. Of course, it is important to keep in mind that these optimality criteria, while satisfying from a theoretical point of view, are not related to the dynamics of visual perception. As a result, the choice of one algorithm over the other will almost always be determined (at least partially) by the perceived visual quality of the resulting images.

By using the definition of convolution given in Eq. (4.6-23), and as explained in Section 2.6.6, we can express Eq. (5.5-16) in vector-matrix form:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta} \tag{5.9-1}$$

For example, suppose that $g(x, y)$ is of size $M \times N$. Then we can form the first N elements of the vector $\mathbf{g}$ by using the image elements in first row of $g(x, y)$, the next N elements from the second row, and so on. The resulting vector will have dimensions $MN \times 1$. These are also the dimensions of $\mathbf{f}$ and $\boldsymbol{\eta}$, as these vectors are formed in the same manner. The matrix $\mathbf{H}$ then has dimensions $MN \times MN$. Its elements are given by the elements of the convolution given in Eq. (4.6-23).

Consult the book Web site for a brief review of vectors and matrices.

It would be reasonable to arrive at the conclusion that the restoration problem can now be reduced to simple matrix manipulations. Unfortunately, this is not the case. For instance, suppose that we are working with images of medium size; say $M = N = 512$. Then the vectors in Eq. (5.9-1) would be of dimension

262,144 × 1, and matrix **H** would be of dimensions 262,144 × 262,144. Manipulating vectors and matrices of such sizes is not a trivial task. The problem is complicated further by the fact **H** is highly sensitive to noise (after the experiences we had with the effect of noise in the previous two sections, this should not be a surprise). However, formulating the restoration problem in matrix form does facilitate derivation of restoration techniques.

Consult the Tutorials section in the book Web site for an entire chapter devoted to the topic of algebraic techniques for image restoration.

Although we do not fully derive the method of constrained least squares that we are about to present, this method has its roots in a matrix formulation. We give references at the end of the chapter to sources where derivations are covered in detail. Central to the method is the issue of the sensitivity of **H** to noise. One way to alleviate the noise sensitivity problem is to base optimality of restoration on a measure of smoothness, such as the second derivative of an image (our old friend the Laplacian). To be meaningful, the restoration must be constrained by the parameters of the problems at hand. Thus, what is desired is to find the minimum of a criterion function, C, defined as

$$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\nabla^2 f(x, y)\right]^2 \tag{5.9-2}$$

subject to the constraint

$$\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\|^2 = \|\boldsymbol{\eta}\|^2 \tag{5.9-3}$$

where $\|\mathbf{w}\|^2 \triangleq \mathbf{w}^T\mathbf{w}$ is the Euclidean vector norm,[†] and $\hat{\mathbf{f}}$ is the estimate of the undegraded image. The Laplacian operator ∇^2 is defined in Eq. (3.6-3).

The frequency domain solution to this optimization problem is given by the expression

$$\hat{F}(u, v) = \left[\frac{H^*(u, v)}{|H(u, v)|^2 + \gamma|P(u, v)|^2}\right]G(u, v) \tag{5.9-4}$$

where γ is a parameter that must be adjusted so that the constraint in Eq. (5.9-3) is satisfied, and $P(u, v)$ is the Fourier transform of the function

$$p(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{5.9-5}$$

We recognize this function as the Laplacian operator introduced in Section 3.6.2. As noted earlier, it is important to keep in mind that $p(x, y)$, as well as all other relevant spatial domain functions, must be properly padded with zeros prior to computing their Fourier transforms for use in Eq. (5.9-4), as discussed in Section 4.6.6. Note that Eq. (5.9-4) reduces to inverse filtering if γ is zero.

[†] Recall that, for a vector **w** with n components, $\mathbf{w}^T\mathbf{w} = \sum_{k=1}^{n} w_k^2$, where w_k is the kth component of **w**.

EXAMPLE 5.14: Comparison of Wiener and constrained least squares filtering.

■ Figure 5.30 shows the result of processing Figs. 5.29(a), (d), and (g) with constrained least squares filters, in which the values of γ were selected manually to yield the best visual results. This is the same procedure we used to generate the Wiener filtered results in Fig. 5.29(c), (f), and (i). By comparing the constrained least squares and Wiener results, it is noted that the former yielded slightly better results for the high- and medium-noise cases, with both filters generating essentially equal results for the low-noise case. It is not unexpected that the constrained least squares filter would outperform the Wiener filter when selecting the parameters manually for better visual results. The parameter γ in Eq. (5.9-4) is a scalar, while the value of K in Eq. (5.8-6) is an approximation to the ratio of two unknown frequency domain functions; this ratio seldom is constant. Thus, it stands to reason that a result based on manually selecting γ would be a more accurate estimate of the undegraded image. ■

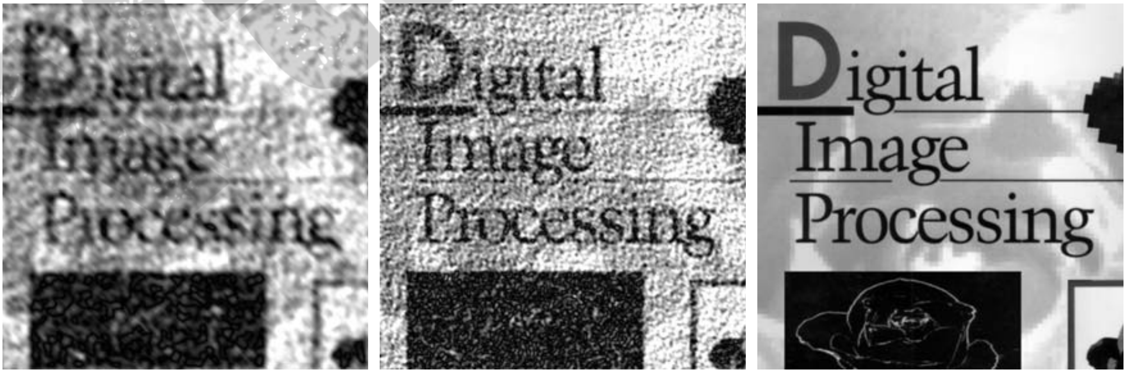
As shown in the preceding example, it is possible to adjust the parameter γ interactively until acceptable results are achieved. If we are interested in optimality, however, then the parameter γ must be adjusted so that the constraint in Eq. (5.9-3) is satisfied. A procedure for computing γ by iteration is as follows.

Define a "residual" vector **r** as

$$\mathbf{r} = \mathbf{g} - \mathbf{H}\hat{\mathbf{f}} \tag{5.9-6}$$

Since, from the solution in Eq. (5.9-4), $\hat{F}(u, v)$ (and by implication $\hat{\mathbf{f}}$) is a function of γ, then **r** also is a function of this parameter. It can be shown (Hunt [1973]) that

$$\begin{aligned}\phi(\gamma) &= \mathbf{r}^T\mathbf{r} \\ &= \|\mathbf{r}\|^2\end{aligned} \tag{5.9-7}$$



a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

is a monotonically increasing function of γ. What we want to do is adjust γ so that

$$\|\mathbf{r}\|^2 = \|\boldsymbol{\eta}\|^2 \pm a \tag{5.9-8}$$

where a is an accuracy factor. In view of Eq. (5.9-6), if $\|\mathbf{r}\|^2 = \|\boldsymbol{\eta}\|^2$, the constraint in Eq. (5.9-3) will be strictly satisfied.

Because $\phi(\gamma)$ is monotonic, finding the desired value of γ is not difficult. One approach is to

1. Specify an initial value of γ.
2. Compute $\|\mathbf{r}\|^2$.
3. Stop if Eq. (5.9-8) is satisfied; otherwise return to step 2 after increasing γ if $\|\mathbf{r}\|^2 < \|\boldsymbol{\eta}\|^2 - a$ or decreasing γ if $\|\mathbf{r}\|^2 > \|\boldsymbol{\eta}\|^2 + a$. Use the new value of γ in Eq. (5.9-4) to recompute the optimum estimate $\hat{F}(u, v)$.

Other procedures, such as a Newton–Raphson algorithm, can be used to improve the speed of convergence.

In order to use this algorithm, we need the quantities $\|\mathbf{r}\|^2$ and $\|\boldsymbol{\eta}\|^2$. To compute $\|\mathbf{r}\|^2$, we note from Eq. (5.9-6) that

$$R(u, v) = G(u, v) - H(u, v)\hat{F}(u, v) \tag{5.9-9}$$

from which we obtain $r(x, y)$ by computing the inverse transform of $R(u, v)$. Then

$$\|\mathbf{r}\|^2 = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} r^2(x, y) \tag{5.9-10}$$

Computation of $\|\boldsymbol{\eta}\|^2$ leads to an interesting result. First, consider the variance of the noise over the entire image, which we estimate by the sample-average method, as discussed in Section 3.3.4:

$$\sigma_\eta^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\eta(x, y) - m_\eta]^2 \tag{5.9-11}$$

where

$$m_\eta = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\eta(x, y) \tag{5.9-12}$$

is the sample mean. With reference to the *form* of Eq. (5.9-10), we note that the double summation in Eq. (5.9-11) is equal to $\|\boldsymbol{\eta}\|^2$. This gives us the expression

$$\|\boldsymbol{\eta}\|^2 = MN[\sigma_\eta^2 + m_\eta^2] \tag{5.9-13}$$

This is a most useful result. It tells us that we can implement an optimum restoration algorithm by having knowledge of only the mean and variance of the noise. These quantities are not difficult to estimate (Section 5.2.4), assuming that the noise and image intensity values are not correlated. This is a basic assumption of all the methods discussed in this chapter.

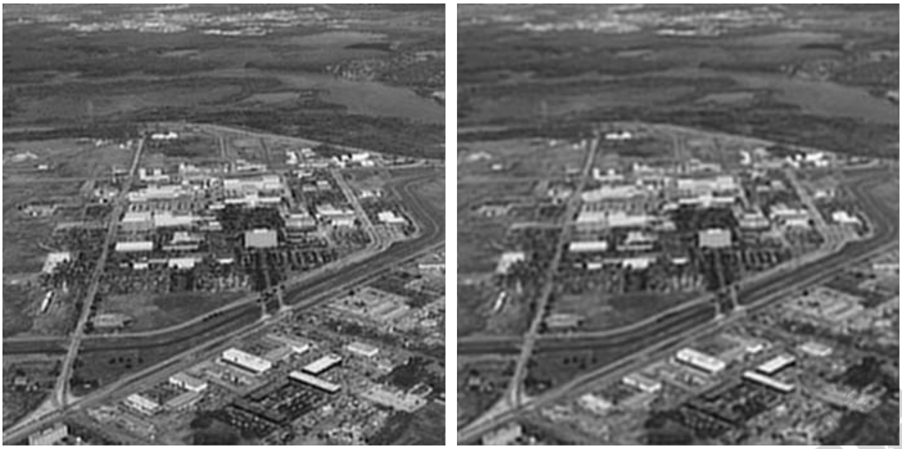


FIGURE 5.31 (a) Iteratively determined constrained least squares restoration of Fig. 5.16(b), using correct noise parameters. (b) Result obtained with wrong noise parameters.

EXAMPLE 5.15: Iterative estimation of the optimum constrained least squares filter.

■ Figure 5.31(a) shows the result obtained by using the algorithm just described to estimate the optimum filter for restoring Fig. 5.25(b). The initial value used for γ was 10^{-5}, the correction factor for adjusting γ was 10^{-6}, and the value for a was 0.25. The noise parameters specified were the same used to generate Fig. 5.25(a): a noise variance of 10^{-5}, and zero mean. The restored result is almost as good as Fig. 5.28(c), which was obtained by Wiener filtering with K manually specified for best visual results. Figure 5.31(b) shows what can happen if the wrong estimate of noise parameters are used. In this case, the noise variance specified was 10^{-2} and the mean was left at a value of 0. The result in this case is considerably more blurred. ■

As stated at the beginning of this section, it is important to keep in mind that optimum restoration in the sense of constrained least squares does not necessarily imply "best" in the visual sense. Depending on the nature and magnitude of the degradation and noise, the other parameters in the algorithm for iteratively determining the optimum estimate also play a role in the final result. In general, automatically determined restoration filters yield inferior results to manual adjustment of filter parameters. This is particularly true of the constrained least squares filter, which is completely specified by a single, scalar parameter.