ERRORS

1. Lens Aberration (Hubble) סטיה העדשה

2. Diffraction of electromagnetic waves at aperture stop of the lens

3. Defocusing למקד

4. Motions and vibrations of the camera תנועות ותנודות של המצלמה
   telephoto lenses
   motion of the object, blur טשטוש - תנועות ותנודות של הגוף

5. Row jittering
   synchronization of frame buffer

6. Bad transmission along video lines שידור רע לאורך קווי וידיו
   Echoes הדים

7. Electronic interference הפרעות אלקטרוניות
   fixed and moving patterns

8. Noisy digital circuits רעש במעגלים דיגיטליים
   individual bits flip - random errors שגיאות אקראיות

9. Atmospheric turbulence מערבולת האטמוספירה

will find references at the end of this chapter to explore further. However, not all of the math can be eliminated.

The example of the Hubble Space Telescope is an opportune one, since it is an ideal way to introduce a technique for characterizing the distortion inherent in an image. A star, when viewed through a telescope, should be seen as a perfect point of light. Ideally, all of the light energy of the star would be focused on a single pixel. In practice this is not so; the distortions of the atmosphere and the telescope optics will yield a slightly blurred image in which the central pixel is brightest, and a small region around it is also brighter than the background. The distortions that have been inflicted on the point image of the star are reflected in the shape and intensity distribution of the star's image. All stars (for a reasonable optical system) will have the same distortions imposed upon them; indeed, all points on the image have been replaced by these small blobs, and the sum of all of the blobs is the sampled image.

The effect that an image acquisition system has on a perfect point source is called the *point spread function* (PSF). The sample image has been produced by convolving the PSF with the perfect image, so that the same blur exists at all points. Figure 6.1 shows a diagrammatic view of how distortion and noise have been applied to the original image to give the sampled, observed image. To obtain the perfect image given the sampled one is the goal of restoration, and it is not generally possible. We therefore wish to improve the image as much as possible, and the PSF tells us what has been done to the image. The ideal solution is to *deconvolve* the image and the PSF, but this can only be done approximately and at some significant expense.

This discussion assumes that the PSF is the same at all points of the image, in

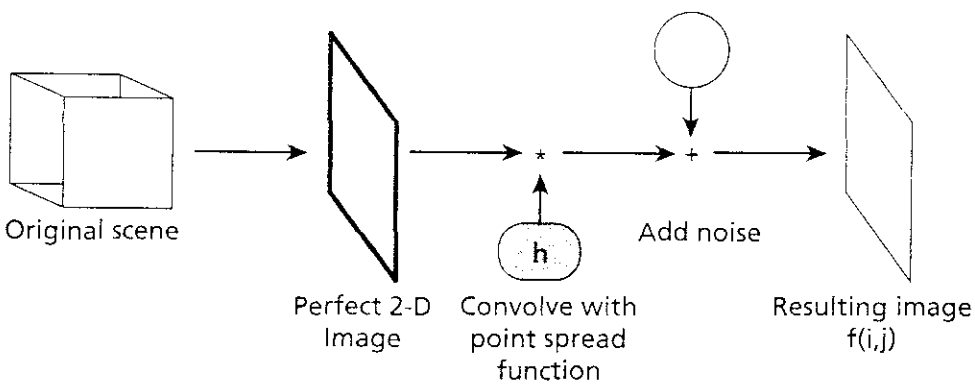


Figure 6.1 One model of how a perfect image becomes distorted by imperfect (real) acquisition systems.

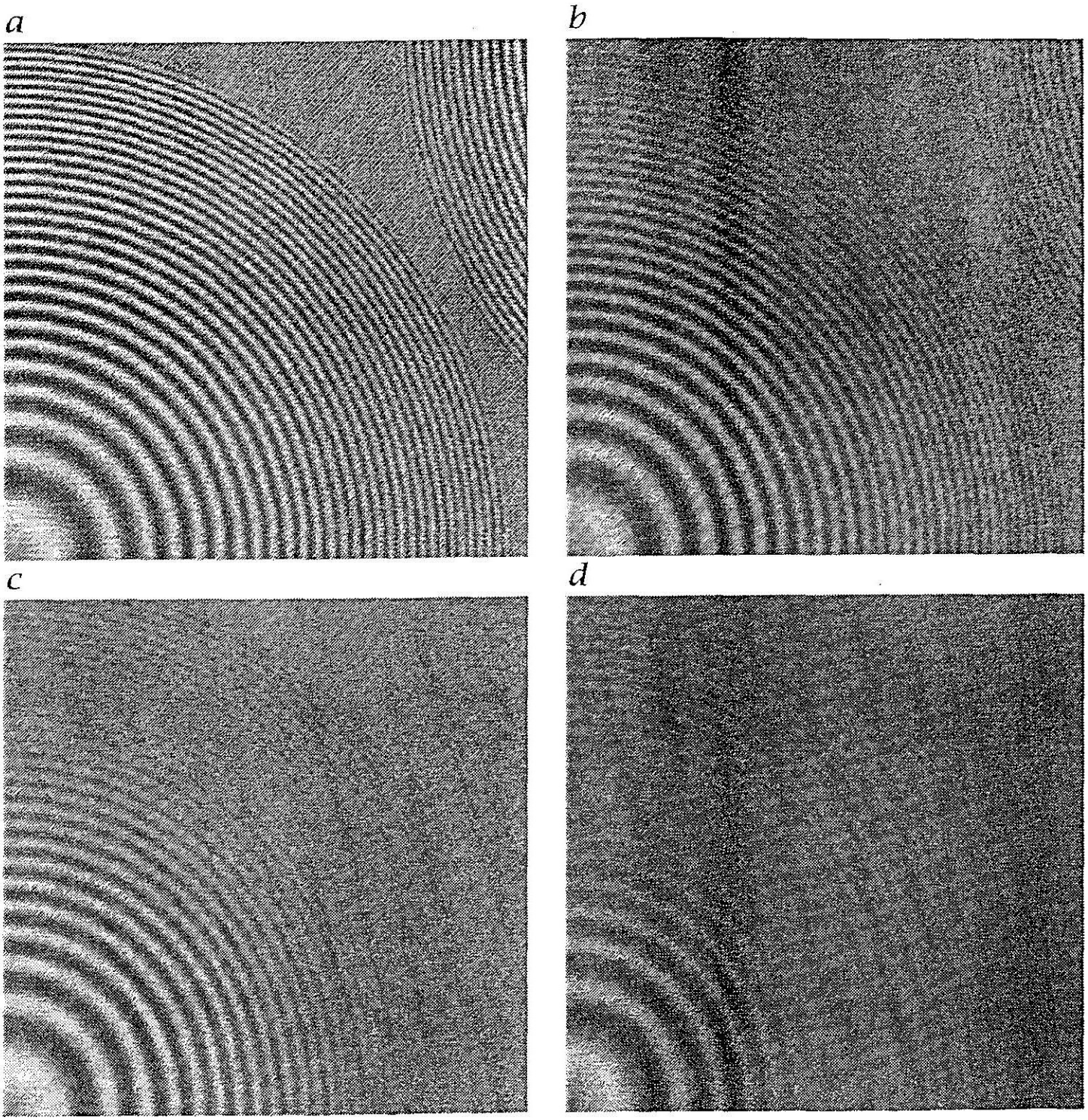


Figure 9.2: A sequence of images of the ring test pattern with increasing degree of defocusing. After focusing (image **a***), the camera was moved step by step toward the test pattern from* **b** *to* **d** *(**exercise 9.1**).*

What is a Point Spread Function?

The PSF of an optical system is the irradiance distribution that results from a single point source in object space. A telescope forming an image of a distant star is a good example: the star is so far away that for all practical purposes it can be considered a point.

Although the source may be a point, the image is not. There are two main reasons. First, aberrations in the optical system will spread the image over a finite area. Second, diffraction effects will also spread the image, even in a system that has no aberrations.

optical transfer function

The **optical transfer function** (**OTF**) of an imaging system (camera, video, system, microscope etc.) is the true measure of resolution (image sharpness) that the system is capable of. The common practice of defining resolution in terms of pixel count is not meaningful, as it is the overall OTF of the complete system, including lens and anti-aliasing filter as well as other factors, that defines true performance. In the most common applications (cameras and video systems) it is the Modulation Transfer Function (the magnitude of the OTF), that is most relevant, although the phase component can have a secondary effect. While resolution, as commonly used with reference to camera systems, describes only the number of pixels in an image, and hence the potential to show fine detail, the transfer function describes the ability of adjacent pixels to change from black to white in response to patterns of varying spatial frequency, and hence the actual capability to show fine detail, whether with full or reduced contrast. An image reproduced with an optical transfer function that 'rolls off' at high spatial frequencies will appear 'blurred' in everyday language. **Modulation Transfer Function** or **MTF** (the OTF magnitude with phase ignored) is roughly the equivalent of frequency response in an audio system, and can be represented by a graph of light amplitude (brightness) versus spatial frequency (cycles per picture width).

FFT Demo

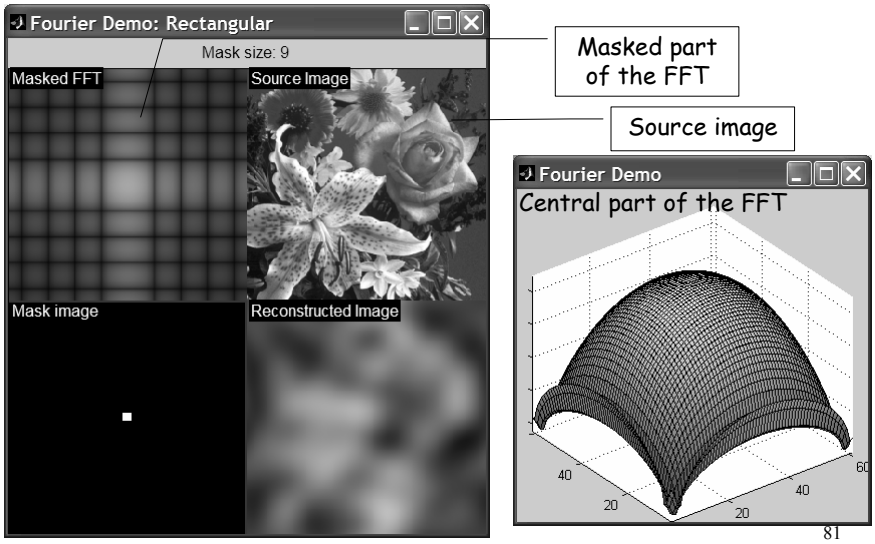
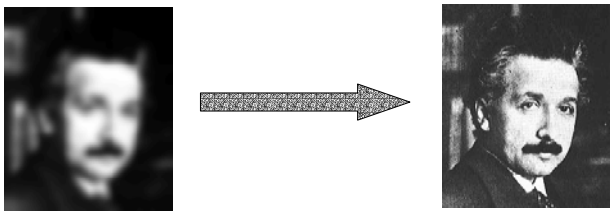


Image Deblurring (Deconvolution)

Problem: how to undo blurring & recover the sharp image?



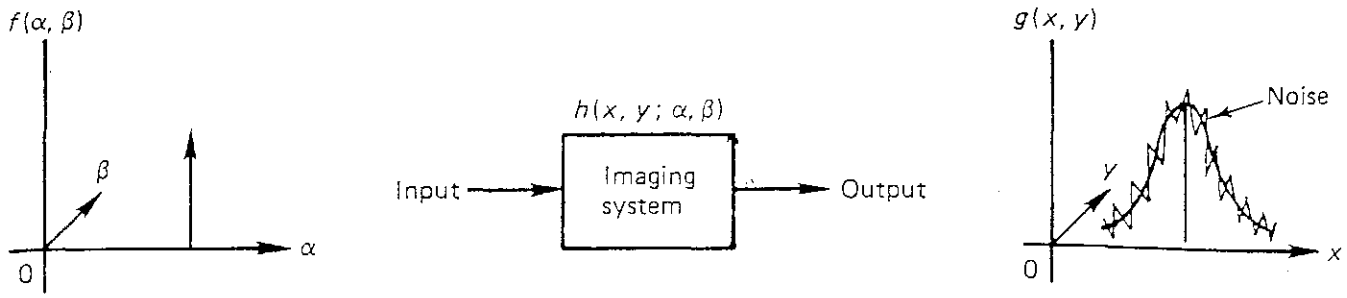


Figure 1.7 Blurring due to an imaging system. Given the noisy and blurred image the image restoration problem is to find an estimate of the input image $f(x, y)$.

1.4 IMAGE RESTORATION

Image restoration refers to removal or minimization of known degradations in an image. This includes deblurring of images degraded by the limitations of a sensor or its environment, noise filtering, and correction of geometric distortion or nonlinearities due to sensors. Figure 1.7 shows a typical situation in image restoration. The image of a point source is blurred and degraded due to noise by an imaging system. If the imaging sytem is linear, the image of an object can be expressed as

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y; \alpha, \beta) f(\alpha, \beta)\, d\alpha\, d\beta + \eta(x, y) \tag{1.1}$$

where $\eta(x, y)$ is the additive noise function, $f(\alpha, \beta)$ is the object, $g(x, y)$ is the image, and $h(x, y; \alpha, \beta)$ is called the *point spread function* (PSF). A typical image restoration problem is to find an estimate of $f(\alpha, \beta)$ given the PSF, the blurred image, and the statistical properties of the noise process.

A fundamental result in filtering theory used commonly for image restoration is called the *Wiener filter*. This filter gives the best linear mean square estimate of the object from the observations. It can be implemented in frequency domain via the fast unitary transforms, in spatial domain by two-dimensional recursive techniques similar to Kalman filtering, or by FIR nonrecursive filters (see Fig. 8.15). It can also be implemented as a semirecursive filter that employs a unitary transform in one of the dimensions and a recursive filter in the other.

Several other image restoration methods such as least squares, constrained least squares, and spline interpolation methods can be shown to belong to the class of Wiener filtering algorithms. Other methods such as maximum likelihood, maximum entropy, and maximum a posteriori are nonlinear techniques that require iterative solutions. These and other algorithms useful in image restoration are discussed in Chapter 8.

A typical imaging system consists of an image formation system, a detector, and a recorder. For example, an electro-optical system such as the television camera contains an optical system that focuses an image on a photoelectric device, which is scanned for transmission or recording of the image. Similarly, an ordinary camera uses a lens to form an image that is detected and recorded on a photosensitive film. A general model for such systems (Fig. 8.3) can be expressed as

$$v(x, y) = g[w(x, y)] + \eta(x, y) \tag{8.1}$$

$$w(x, y) = \int\int_{-\infty}^{\infty} h(x, y; x', y')u(x', y')\,dx'\,dy' \tag{8.2}$$

$$\eta(x, y) = f[g(w(x, y))]\eta_1(x, y) + \eta_2(x, y) \tag{8.3}$$

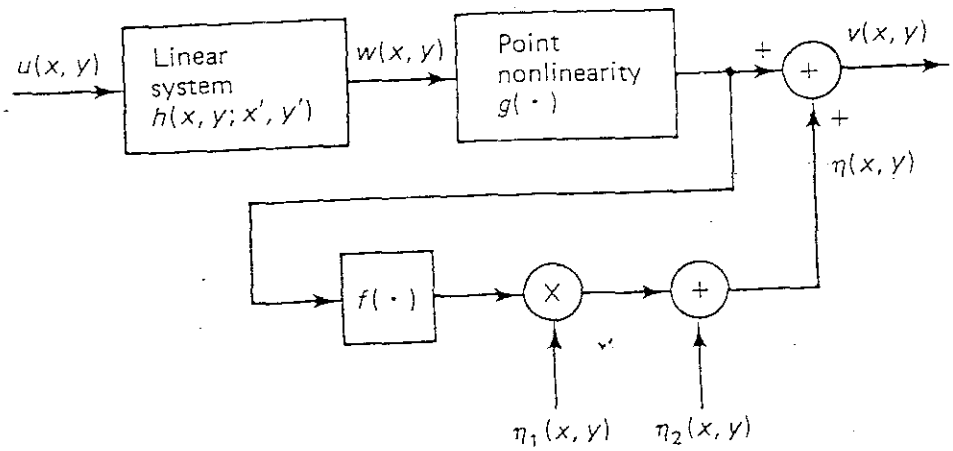


Figure 8.3 Image observation model.

where $u(x, y)$ represents the object (also called the *original image*), and $v(x, y)$ is the observed image. The image formation process can often be modeled by the linear system of (8.2), where $h(x, y; x', y')$ is its impulse response. For space invariant systems, we can write

$$h(x, y; x', y') = h(x - x', y - y'; 0, 0) \triangleq h(x - x', y - y') \tag{8.4}$$

The functions $f(\cdot)$ and $g(\cdot)$ are generally nonlinear and represent the characteristics of the detector/recording mechanisms. The term $\eta(x, y)$ represents the additive noise, which has an image-dependent random component $f[g(w)]\eta_1$ and an image-independent random component η_2.

ble 8.1 lists impulse response models for several spatially invariant systems. ffraction-limited coherent systems have the effect of being ideal low-pass filters. r an incoherent system, this means band-limitedness and a frequency response tained by convolving the coherent transfer function (CTF) with itself (Fig. 8.4). egradations due to phase distortion in the CTF are called *aberrations* and manifest emselves as distortions in the pass-band of the incoherent optical transfer function)TF). For example, a severely out-of-focus lens with rectangular aperture causes aberration in the OTF, as shown in Fig. 8.4.

Motion blur occurs when there is relative motion between the object and the mera during exposure. Atmospheric turbulence is due to random variations in the fractive index of the medium between the object and the imaging system. Such egradations occur in the imaging of astronomical objects. Image blurring also curs in image acquisition by scanners in which the image pixels are integrated er the scanning aperture. Examples of this can be found in image acquisition by dar, beam-forming arrays, and conventional image display systems using tele-

TABLE 8.1 Examples of Spatially Invariant Models

Type of system	Impulse response $h(x, y)$	Frequency response $H(\xi_1, \xi_2)$
Diffraction limited, coherent (with rectangular aperture)	$ab\ \mathrm{sinc}(ax)\ \mathrm{sinc}(by)$	$\mathrm{rect}\left(\frac{\xi_1}{a}, \frac{\xi_2}{b}\right)$
Diffraction limited, incoherent (with rectangular aperture)	$\mathrm{sinc}^2(ax)\ \mathrm{sinc}^2(by)$	$\mathrm{tri}\left(\frac{\xi_1}{a}, \frac{\xi_2}{b}\right)$
Horizontal motion	$\frac{1}{\alpha_0}\ \mathrm{rect}\left(\frac{x}{\alpha_0} - \frac{1}{2}\right)\delta(y)$	$e^{-j\pi\xi_1\alpha_0}\ \mathrm{sinc}(\xi_1\alpha_0)$
Atmospheric turbulence	$\exp\{-\pi\alpha^2(x^2 + y^2)\}$	$\frac{1}{\alpha^2}\ \exp\left[\frac{-\pi(\xi_1^2 + \xi_2^2)}{\alpha^2}\right]$
Rectangular scanning aperture	$\mathrm{rect}\left(\frac{x}{\alpha}, \frac{y}{\beta}\right)$	$\alpha\beta\ \mathrm{sinc}(\alpha\xi_1)\ \mathrm{sinc}(\beta\xi_2)$
CCD interactions	$\sum_{k,l=-1}^{1}\sum \alpha_{k,l}\delta(x - k\Delta, y - l\Delta)$	$\sum_{k,l=-1}^{1}\sum \alpha_{k,l}e^{-j2\pi\Delta(\xi_1 k + \xi_2 l)}$

Figure 3.4-3 Typical Blur Mask Coefficients

0	0	0	0	0
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0
0	0	0	0	0

a. Horizontal PSF mask with uniform blur.

0	0	1	0	0
0	0	1	0	0
0	0	4	0	0
0	0	1	0	0
0	0	1	0	0

b. Vertical PSF mask with center-weighting.

1	0	0	0	0
0	2	0	0	0
0	0	4	0	0
0	0	0	2	0
0	0	0	0	1

c. Diagonal PSF mask with gaussian distribution.

0	1	1	1	0
1	2	4	2	1
1	4	8	4	1
1	2	4	2	1
0	1	1	1	0

d. Circular PSF mask with gaussian distribution.

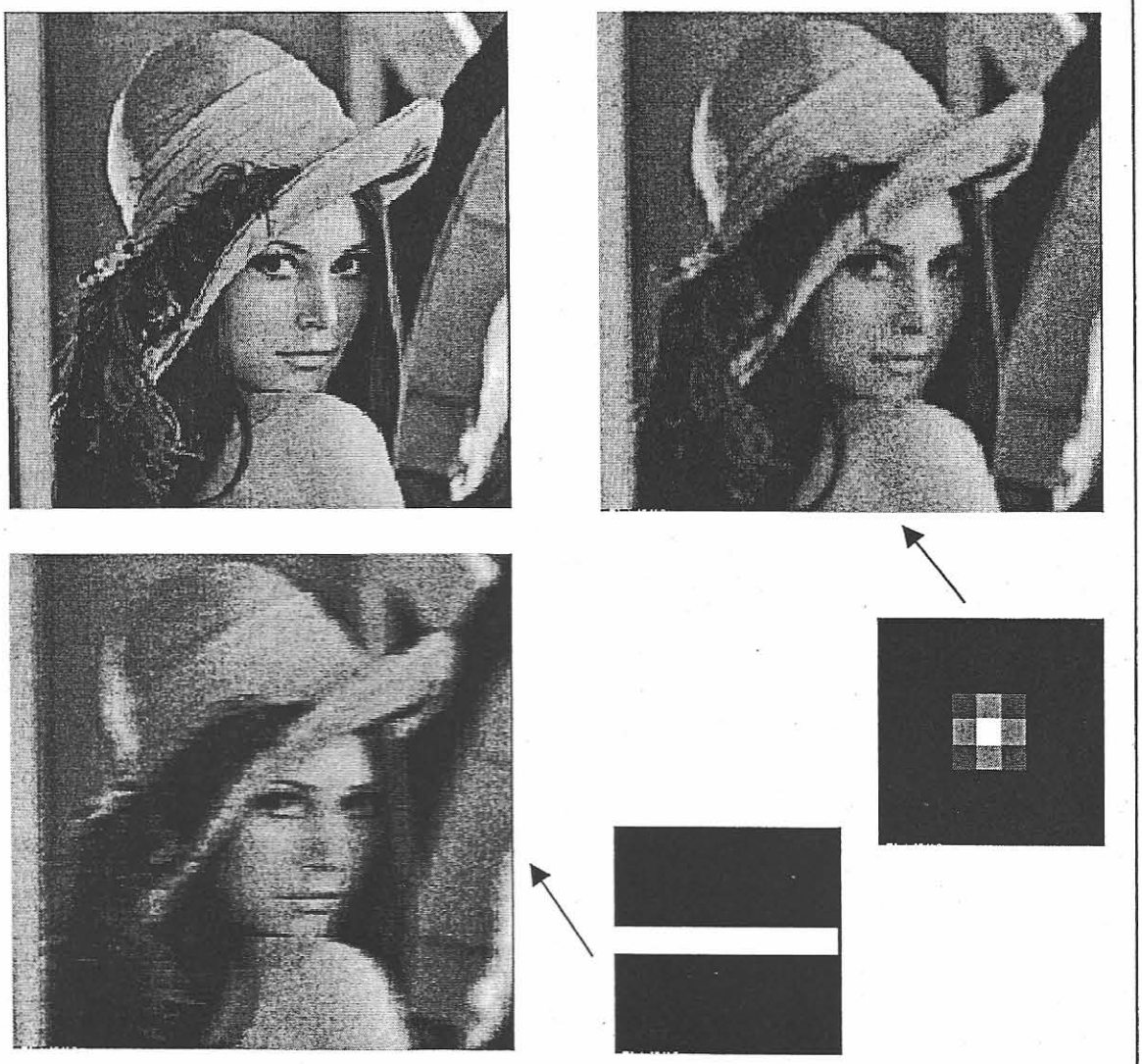
3.4.1 Inverse Filter

The inverse filter uses the foregoing model, with the added assumption of r noise ($N(u, v) = 0$). If this is the case, the Fourier transform of the degraded image is

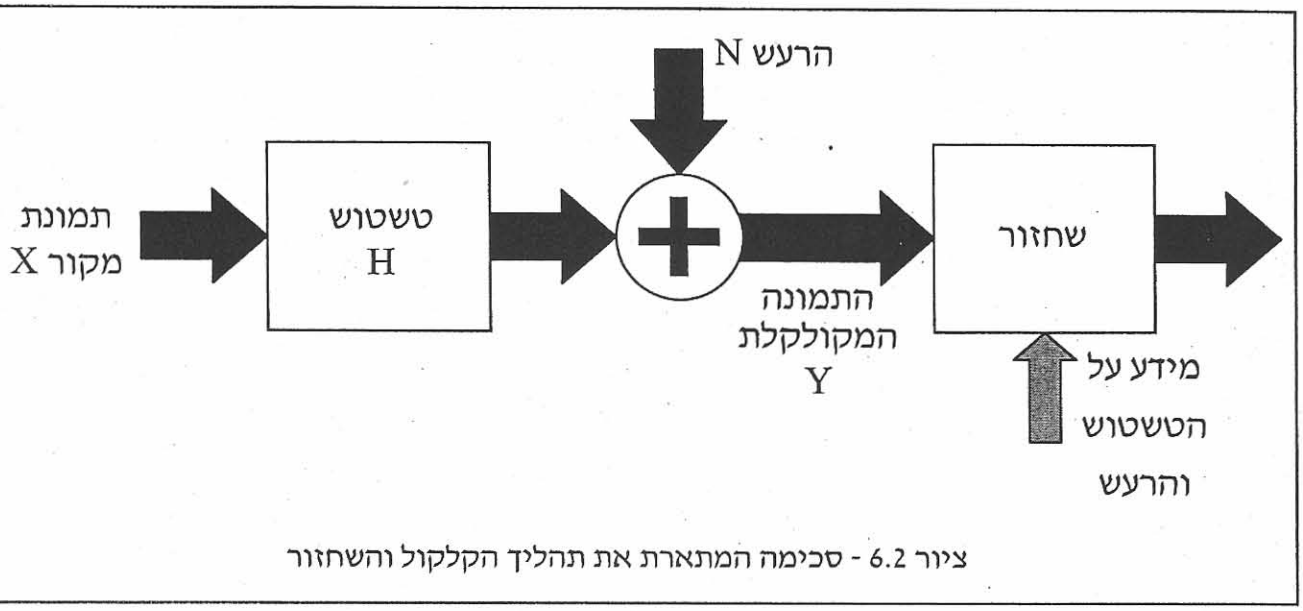
$$D(u, v) = H(u, v)I(u, v) + 0$$

So the Fourier transform of the original image can be found as follows:

$$y = x * h + n$$
$$y = Hx + N$$



ציור 6.1 - תמונת מקור וגרסאות מקולקלות שלה ע"י טשטוש גאוסי (למעלה בצד ימין) וטשטוש כתוצאה מתנועה (שמאל למטה). בשני המקרים הוסף רעש גאוסי לבן בעל שונות $\sigma_N = 5$



ציור 6.2 - סכימה המתארת את תהליך הקלקול והשחזור

3.4.1 Inverse Filter

The inverse filter uses the foregoing model, with the added assumption of no noise ($N(u, v) = 0$). If this is the case, the Fourier transform of the degraded image is

$$D(u, v) = H(u, v)I(u, v) + 0$$

So the Fourier transform of the original image can be found as follows:

$$I(u, v) = \frac{D(u, v)}{H(u, v)} = D(u, v)\frac{1}{H(u, v)}$$

To find the original image, we take the inverse Fourier transform of $I(u, v)$:

$$I(r, c) = F^{-1}[I(u, v)] = F^{-1}\left[\frac{D(u, v)}{H(u, v)}\right] = F^{-1}\left[D(u, v)\frac{1}{H(u, v)}\right]$$

where $F^{-1}[\ ]$ represents the inverse Fourier transform.

The equation implies that the original, undegraded image can be obtained by multiplying the Fourier transform of the degraded image $D(u, v)$ by $1/H(u, v)$ and then inverse Fourier transforming the result. Thus, the restoration filter applied is $1/H(u, v)$, the *inverse filter*. Note that this inversion is a point-by-point inversion, *not* a matrix inversion.

EXAMPLE 3 – 2

$$H(u, v) = \begin{bmatrix} 50 & 50 & 25 \\ 20 & 20 & 20 \\ 20 & 35 & 22 \end{bmatrix} \qquad \frac{1}{H(u, v)} = \begin{bmatrix} \frac{1}{50} & \frac{1}{50} & \frac{1}{25} \\ \frac{1}{20} & \frac{1}{20} & \frac{1}{20} \\ \frac{1}{20} & \frac{1}{35} & \frac{1}{22} \end{bmatrix}$$

To find $1/H(u, v)$, we take each term separately and divide it into 1.

Unfortunately, in practice, complications arise when this technique is applied. If any points in $H(u, v)$ are zero, we face a mathematical dilemma—division by zero. If the assumption of no noise is correct, then the degraded image transform $D(u, v)$ will also have corresponding zeros, and we are left with an indeterminate ratio, 0/0. If the assumption is incorrect, and the image has been corrupted by additive noise, then the zeros will not coincide, and the image restored by the inverse filter will be obscured by the contribution of the noise terms. This can be seen by considering the following equation:

$$D(u, v) = H(u, v)I(u, v) + N(u, v)$$

Then, when we apply the inverse filter, we obtain

$$\hat{I}(u, v) = \frac{D(u, v)}{H(u, v)} = \frac{H(u, v)I(u, v)}{H(u, v)} + \frac{N(u, v)}{H(u, v)}$$
$$= I(u, v) + \frac{N(u, v)}{H(u, v)}$$

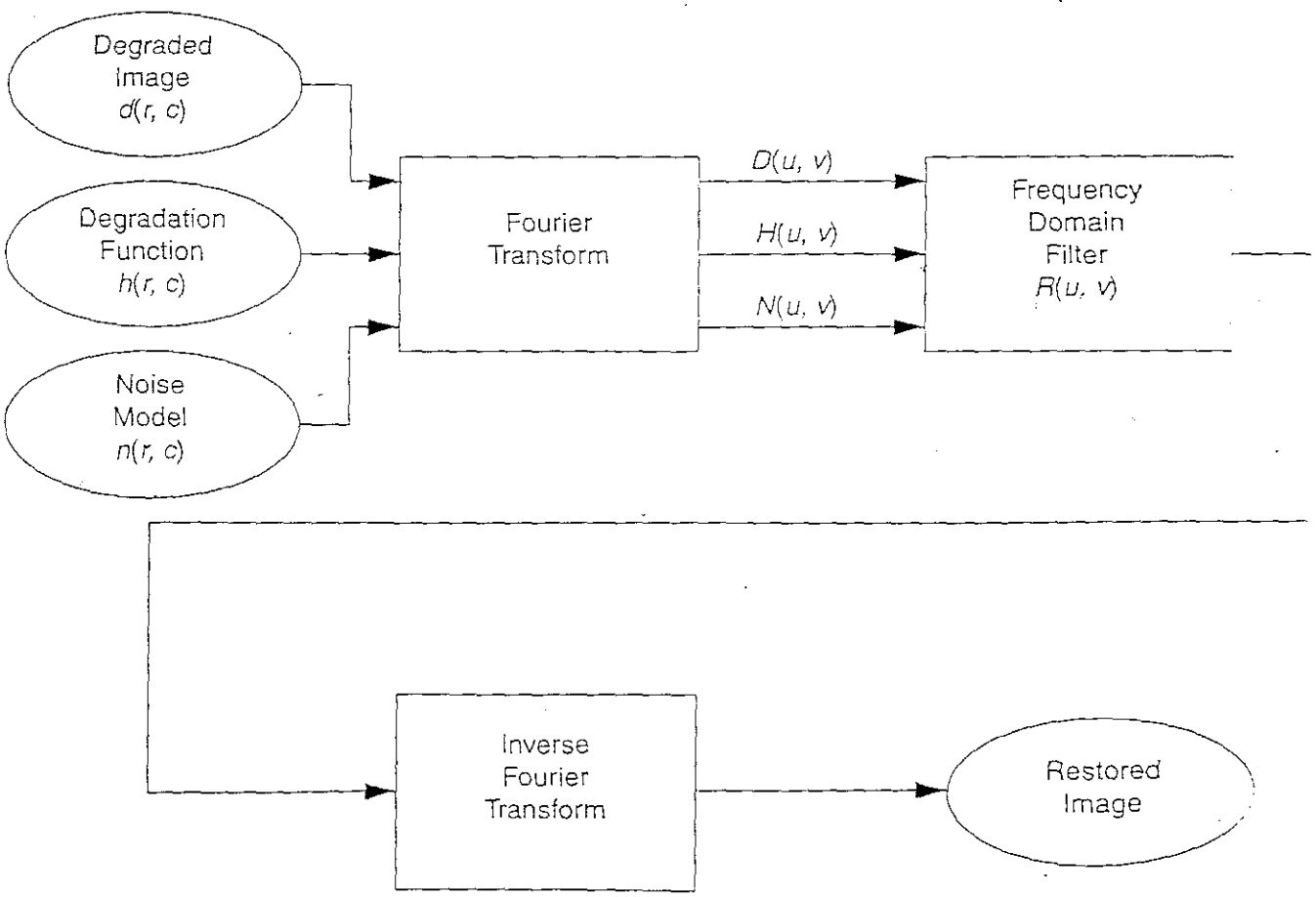
As the values in $H(u, v)$ become very small, the second term becomes very large, and it overshadows the $I(u, v)$ term, which corresponds to the original image we are trying to recover.

Frequency domain filtering operates by using the Fourier transform representation of images. This representation consists of information about the spatial frequency content of the image, also referred to as the spectrum of the image. In Figure 3.4-1 is the general model for frequency domain filtering. The Fourier transform is performed on three spatial domain functions: 1) the degraded image, $d(r, c)$, 2) the degradation function, $h(r, c)$, and 3) the noise model, $n(r, c)$. Next, the frequency domain filter is applied to the Fourier transform outputs, $N(u, v)$, $D(u, v)$, and $H(u, v)$. The output of the filter operation undergoes an inverse Fourier transform to give the restored image.

The specific models used for $h(r, c)$ and $n(r, c)$ depend on the application and, in practice, often must be estimated. In some cases, they may not be explicitly required (as in the previous section on spatial filters for noise removal, where the $h(r, c)$ was assumed unnecessary). The typical noise models are provided in Section 3.2. The degradation function can be experimentally determined in various ways.

Another name for the degradation function is the point spread function (PSF). The *point spread function* (the 2-D equivalent of the impulse response) describes what happens to a single point of light when it passes through a system. The PSF for a *linear, shift invariant system* completely describes the system. This makes it easy to find the degradation function for a system, *if the system is available and the conditions under which the image was acquired have not changed*—all we need to do is to

re 3.4-1 Frequency Domain Filtering



Pseudoinverse Filter

The pseudoinverse filter is a stabilized version of the inverse filter. For a linear shift invariant system with frequency response $H(\omega_1, \omega_2)$, the pseudoinverse filter is defined as

$$H^{-}(\omega_1, \omega_2) = \begin{cases} \dfrac{1}{H(\omega_1, \omega_2)}, & H \neq 0 \\ 0, & H = 0 \end{cases} \tag{8.26}$$

Here, $H^{-}(\omega_1, \omega_2)$ is also called the *generalized inverse* of $H(\omega_1, \omega_2)$, in analogy with the definition of the generalized inverse of matrices. In practice, $H^{-}(\omega_1, \omega_2)$ is set to zero whenever $|H|$ is less than a suitably chosen positive quantity ϵ.

Example 8.3

Figure 8.10 shows a blurred image simulated digitally as the output of a noiseless linear system. Therefore, $W(\omega_1, \omega_2) = H(\omega_1, \omega_2)U(\omega_1, \omega_2)$. The inverse filtered image is obtained as $\hat{U}(\omega_1 \omega_2) \triangleq W(\omega_1, \omega_2)/H(\omega_1, \omega_2)$. In the presence of additive noise, the inverse filter output can be written as

$$\hat{U} = \frac{W}{H} + \frac{N}{H} = U + \frac{N}{H} \tag{8.27}$$

where $N(\omega_1, \omega_2)$ is the noise term. Even if N is small, N/H can assume large values resulting in amplification of noise in the filtered image. This is shown in Fig. 8.10c, where the small amount of noise introduced by computer round-off errors has been amplified by the inverse filter. Pseudoinverse filtering reduces this effect (Fig. 8.10d).

From (9.49), a system that recovers $f(n_1, n_2)$ from $g(n_1, n_2)$ is an inverse filter, shown in Figure 9.17.

The inverse filter in Figure 9.17 tends to be very sensitive to noise. When $B(\omega_1, \omega_2)$ is very small, $1/B(\omega_1, \omega_2)$ is very large, and small noise in the frequency regions where $1/B(\omega_1, \omega_2)$ is very large may be greatly emphasized. One method of lessening the noise sensitivity problem is to limit the frequency response $1/B(\omega_1, \omega_2)$ to some threshold γ as follows:

$$H(\omega_1, \omega_2) = \begin{cases} \dfrac{1}{B(\omega_1, \omega_2)}, & \text{if } \dfrac{1}{|B(\omega_1, \omega_2)|} < \gamma \\ \gamma \dfrac{|B(\omega_1, \omega_2)|}{B(\omega_1, \omega_2)}, & \text{otherwise.} \end{cases} \tag{9.50}$$

The inverse filter $1/B(\omega_1, \omega_2)$ and its variation in (9.50) can be implemented in a variety of ways. We may design a filter whose frequency response approximates the desired one, using the filter design techniques discussed in Chapters 4 and 5, and then convolve the blurred image with the designed filter. Alternatively, we can implement the system, using DFTs and inverse DFTs in a manner analogous to the Wiener filter implementation discussed in Section 9.2.1.

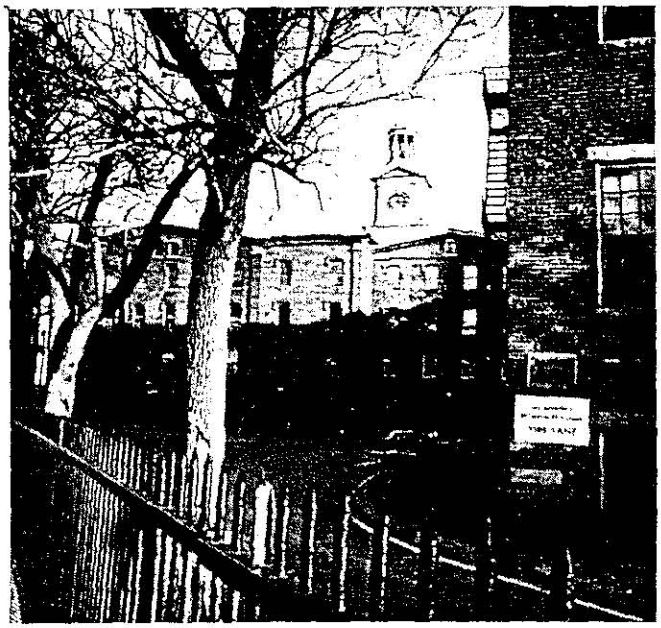
(a) Original image

(b) Blurred image

(c) Inverse filtered

(d) Pseudo-inverse filtered

Figure 8.10 Inverse and pseudo-inverse filtered images.



(a)

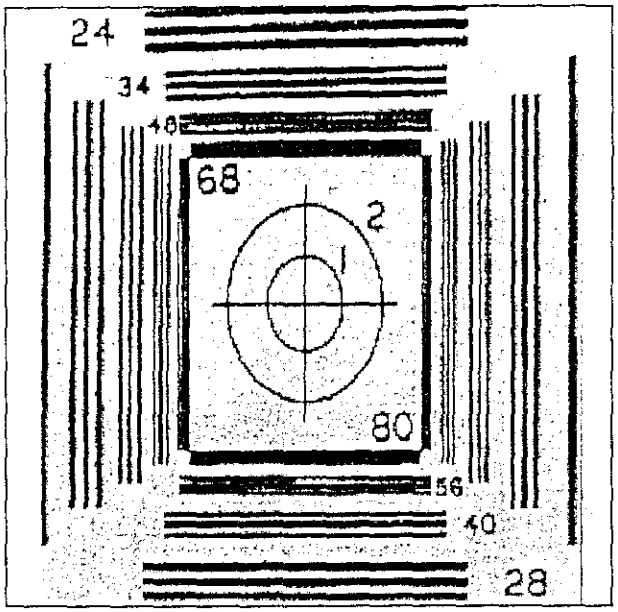
(b)

(c)

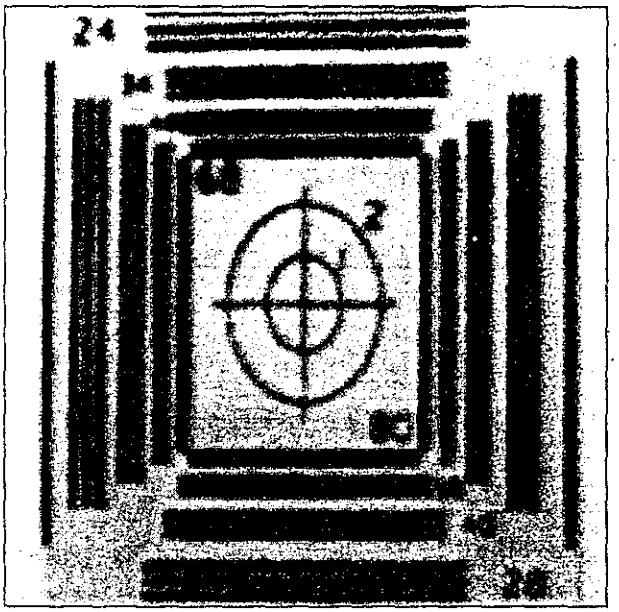
Figure 9.18 (a) Original image of 512 × 512 pixels; (b) image blurred by a Gaussian-shaped point spread function; (c) result of inverse filtering.

One method to deal with this problem is to limit the restoration to a specific radius about the origin in the spectrum, called the restoration cutoff frequency. For spectral components beyond this radius, we can set the filter gain to 0 ($\hat{I}(u, v) = 0$). This is the equivalent of an ideal lowpass filter, which may result in blurring and ringing. In practice, the selection of the cutoff frequency must be experimentally determined and is highly application specific. In Figure 3.4-4 we see the result of applying the inverse filter to an image blurred by an 11×11 gaussian convolution mask. Here

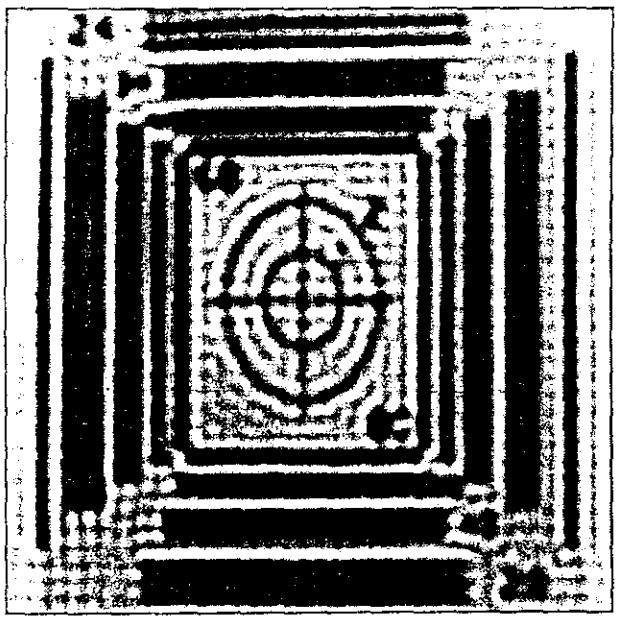
Figure 3.4-4 Inverse Filter



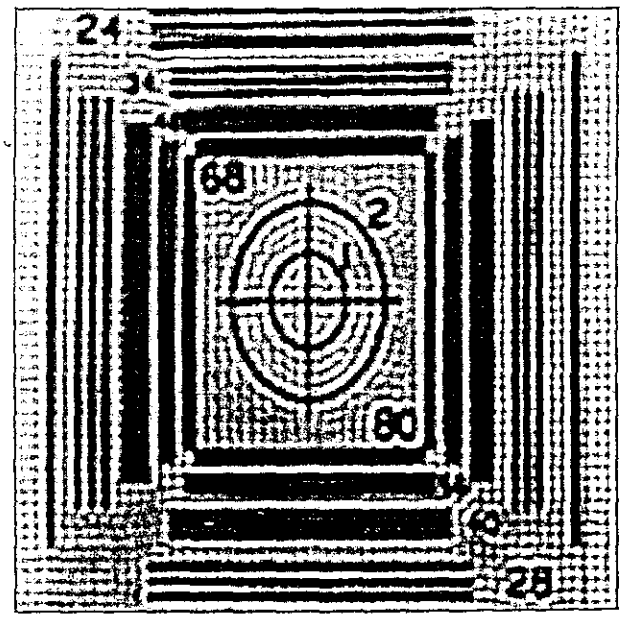
a. Original image.



b. Image blurred with an 11×11 gaussian convolution mask.

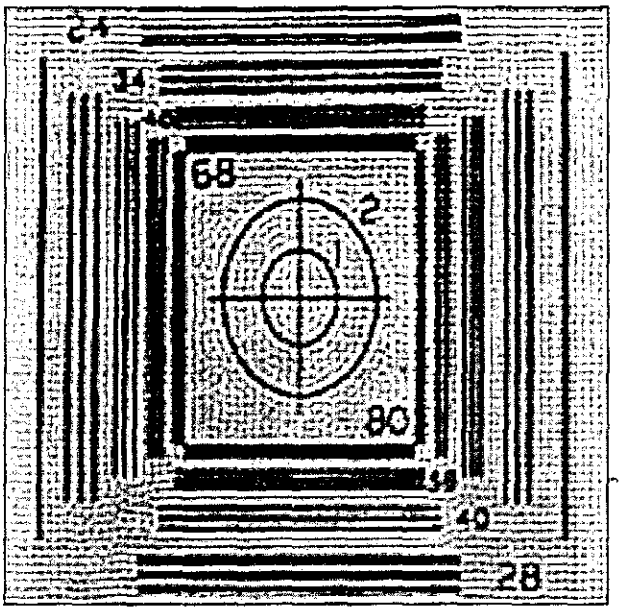


c. Inverse filter, with cutoff frequency = 40, histogram stretched with 3% low and high clipping to show detail.

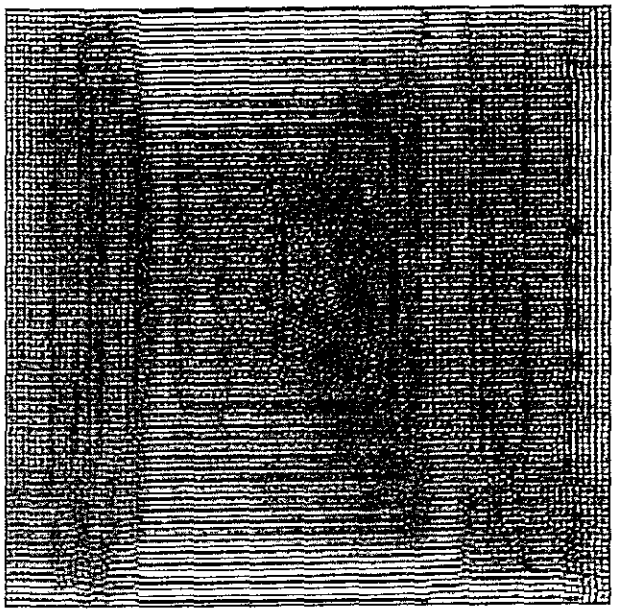


d. Inverse filter, with cutoff frequency = 60, histogram stretched.

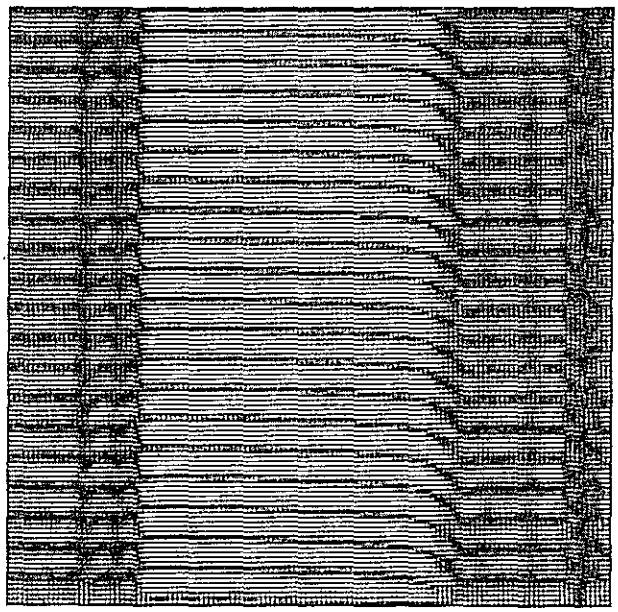
igure 3.4-4 (Continued)



e. Inverse filter, with cutoff frequency = 80, histogram stretched.



f. Inverse filter, with cutoff frequency = 100, histogram stretched.



g. Inverse filter, with cutoff frequency = 120, histogram stretched.

we see that selection of a cutoff frequency that is too low may provide poor results, and with a cutoff frequency too high the resulting image is overwhelmed by noise effects.

With some types of degradation, the function $H(u, v)$ falls off quickly as we move away from the origin in the spectrum. In this case we may want to set the filter gain to 1 for frequencies beyond the restoration cutoff. Another possibility is to model a Butterworth filter, or something between the extremes of setting the gain to 0 or 1. In practice a similar result can be achieved by limiting the gain of the filter to some maximum value.

the following discussion we consider the problem of restoring an image that has been blurred by uniform linear motion. We singled out this problem because of its practical implications and also because it lends itself well to an analytical formulation. Solution of the uniform blurring case also demonstrates how zeros of $H(u, v)$ can be handled computationally. These considerations are important, because they often arise in practice in other contexts of image restoration by inverse filtering.

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) is obtained in this case by integrating the instantaneous exposure over the time interval during which the 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

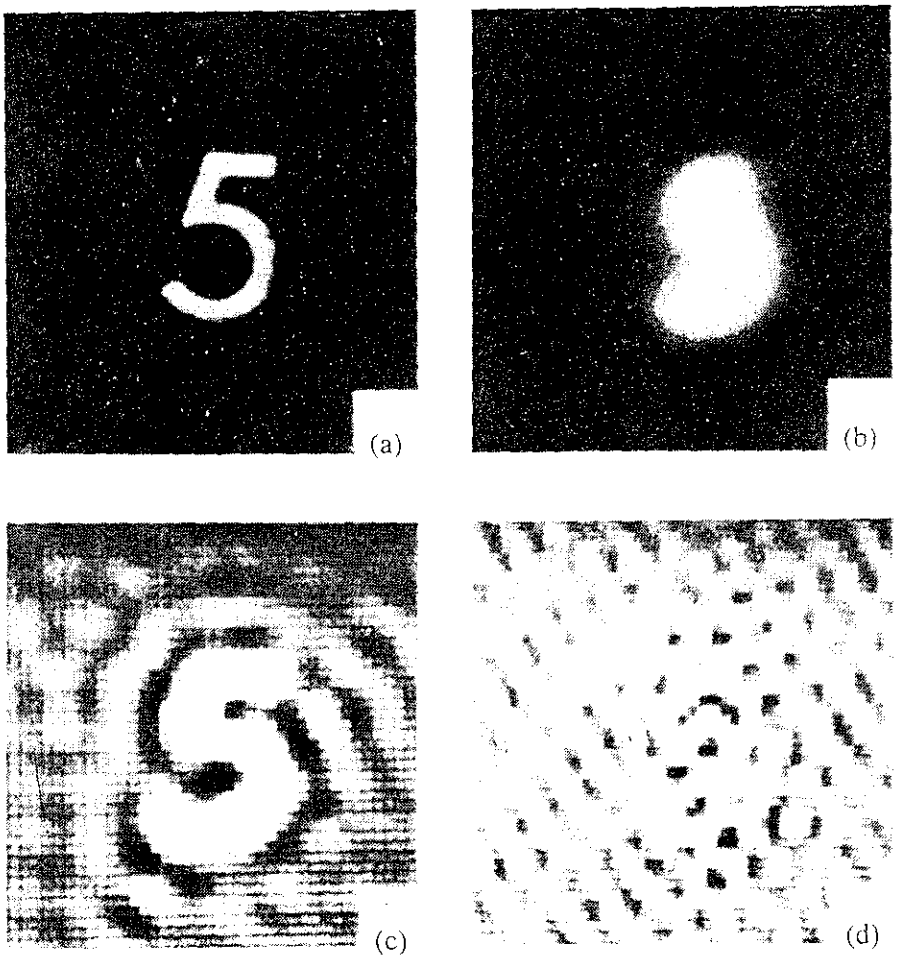
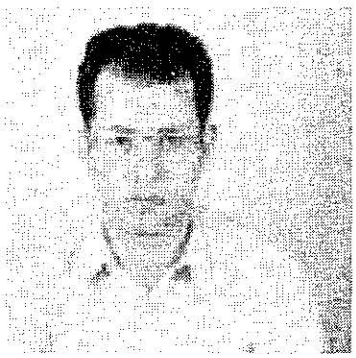
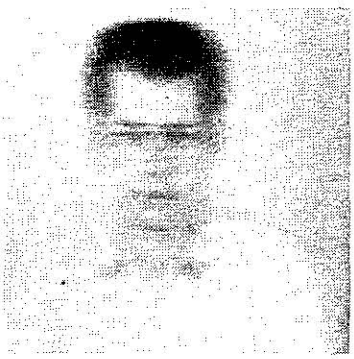


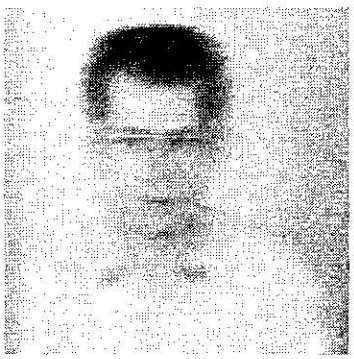
Figure 5.3 *Example of image restoration by inverse filtering: (a) original image $f(x, y)$; (b) degraded (blurred) image $g(x, y)$; (c) result of restoration by considering a neighborhood about the origin of the uv plane that does not include excessively small values of $H(u, v)$; (d) result of using a larger neighborhood in which this condition does not hold. (From McGlamery [1967].)*



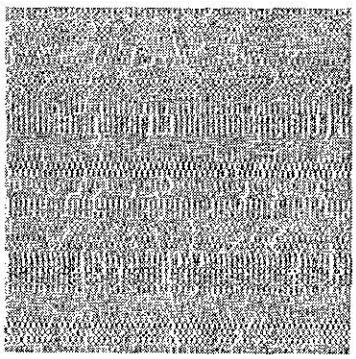
(a) Original image



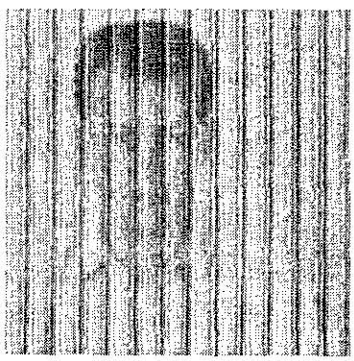
(b) Realistic blurring



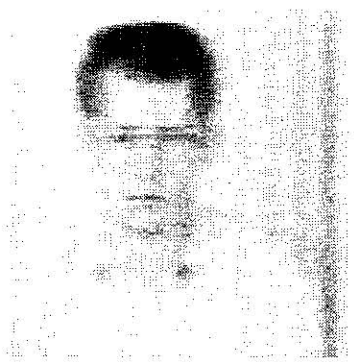
(c) Blurring with cylindrical boundary condition



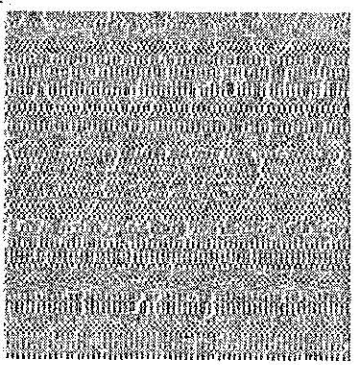
(d) Inverse filtering of (b)



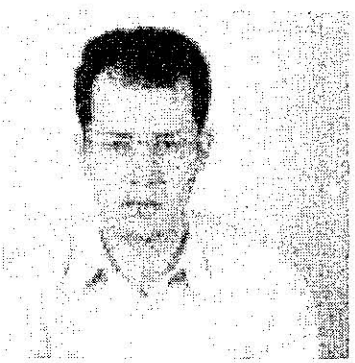
(e) Inverse filtering of (b) omitting division by 0



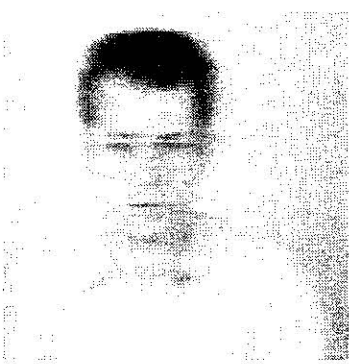
(f) Inverse filtering of (b) omitting division with terms beyond the first 0



(g) Inverse filtering of (c)

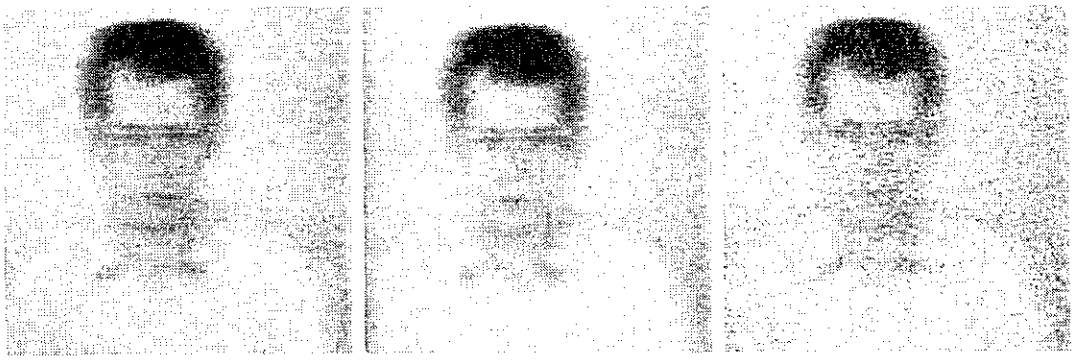


(h) Inverse filtering of (c) omitting division by 0



(i) Inverse filtering of (c) omitting division with terms beyond the first 0

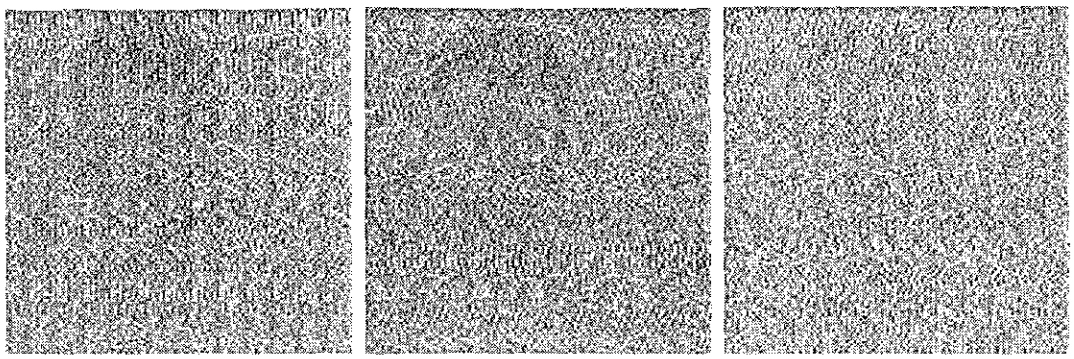
Figure 6.7: Image restoration with inverse filtering.



(a) Realistic blurring with added Gaussian noise ($\sigma = 10$)

(b) Blurring using cylindrical boundary with added Gaussian noise ($\sigma = 10$)

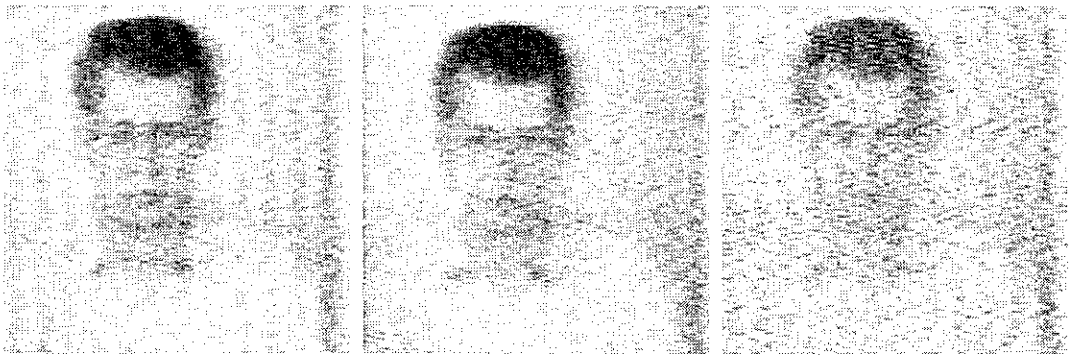
(c) Realistic blurring with added Gaussian noise ($\sigma = 20$)



(d) Inverse filtering of (a), omitting division by 0

(e) Inverse filtering of (b), omitting division by 0

(f) Inverse filtering of (c), omitting division by 0



(g) Inverse filtering of (a), but omitting division with terms beyond the first 0

(h) Inverse filtering of (b), but omitting division with terms beyond the first 0

(i) Inverse filtering of (c), but omitting division with terms beyond the first 0

Figure 6.8: Image restoration with inverse filtering in the presence of noise.

Wiener Filters

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

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

def: impulse response of H = point spread function (PSF)

$$h(x,\alpha,y,\beta) = H[\delta(x-\alpha, y-\beta)]$$

to find f given g is Fredholm integral of the 1st kind

position invariant

$$g(x,y) = \iint f(\alpha,\beta)\, h(x-\alpha, y-\beta)\,d\alpha\,d\beta + \eta(x,y)$$

(noise)

Fourier transform ⇓

$$G(u,v) = H(u,v)\,F(u,v) + N(u,v)$$

H = degradation function

How to find h or H

1. look at subimage with little noise

   $g_s(x,y)$ = observed subimage

   $\hat{f}_s(x,y)$ = estimate of original subimage

   $$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$ position invariant

So $H(u,v)$ has same shape

2. experiment with effect on small dot (5.24 - Gonzalez)

   $$H(u,v) = \frac{G(u,v)}{A}$$ F(impulse) = const

3. physical model

atmospheric turbulence $\rightarrow H(u,v) = e^{-k(u^2+v^2)^{5/6}}$

3.2.2 Mean and Variance

The two basic parameters that describe a random variable are its *mean value* (also known as the *expectation value*) and its *variance*. The mean value μ is defined as

$$\mu = \langle g \rangle = \int_{-\infty}^{\infty} p(g) g \mathrm{d}g, \qquad \mu = \sum_{q=0}^{Q-1} p_q g_q. \tag{3.2}$$

The computation of the expectation value is denoted — as in *quantum mechanics* — by a pair of angle brackets: $\langle \cdots \rangle$.

The mean can also be determined without knowing the probability density function explictly by averaging an infinite number of measurements:

$$\mu = \langle g \rangle = \lim_{P \to \infty} \frac{1}{P} \sum_{p=1}^{P} g_p \tag{3.3}$$

Since we cannot take an infinite number of measurements, the determination of the mean by (3.3) remains an estimate with a residual uncertainty that depends on the form of the probability density function, i. e., the type of the random process.

The *variance* $\sigma^2 = \langle (g - \langle g \rangle)^2 \rangle$ is a measure of the extent to which the measured values deviate from the mean value:

$$\sigma^2 = \int_{-\infty}^{\infty} p(g)(g - \langle g \rangle)^2 \mathrm{d}g, \qquad \sigma^2 = \sum_{q=0}^{Q-1} p_q (g_q - \langle g \rangle)^2. \tag{3.4}$$

3.2.3 Central Moments

The probability density function can be characterized in more detail by quantities similar to the variance, the *moments* of nth order $m_n = \langle (g - \langle g \rangle)^n \rangle$:

$$m_n = \int_{-\infty}^{\infty} p(g)(g - \langle g \rangle)^n \mathrm{d}, \qquad m_n = \sum_{q=0}^{Q-1} p_q (g_q - \langle g \rangle)^n. \tag{3.5}$$

The first central moment is — by definition — zero. The second moment corresponds to the variance. The third moment, the *skewness*, is a measure for the asymmetry of the probability density function around the mean value. If this function is symmetrical with respect to the mean value, the third and all higher-order odd moments vanish.

3.2.4 Normal and Binomial Distribution

The probability density function depends on the nature of the underlying process. Many processes with continuous random variables can be adequately described by the *normal* or *Gaussian* probability distribution

$$p(g) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(g - \langle g \rangle)^2}{2\sigma^2} \right). \tag{3.6}$$

3.3.2 Correlations and Covariances

Now we can relate the gray values at two different positions with each other. One measure for the correlation of the gray values is the expectation value for the product of the gray values at the two positions, the *autocorrelation function*

$$\begin{aligned} R_{gg}(m,n;m',n') &= \langle G_{mn}G_{m'n'}\rangle \\ &= \sum_{q=0}^{Q-1}\sum_{q'=0}^{Q-1} g_q g_{q'} p(q,q';m,n;m',n'). \end{aligned} \tag{3.14}$$

The probability density function has six parameters and tells us the probability that we simultaneously measure the gray value q at the point (m,n) and q' at the point (m',n'). The autocorrelation function is four-dimensional. Therefore this general statistics is hardly ever used. Things become easier if the statistics does not explicitly depend on the position of the pixel. Such a random field is called *homogeneous*. The mean value is then constant over the whole image,

$$\langle G\rangle = \text{const.} \tag{3.15}$$

and the autocorrelation function becomes *shift invariant*:

$$\begin{aligned} R_{gg}(m+k,n+l;m'+k,n'+l) &= \\ &= R_{gg}(m,n;m',n') \\ &= R_{gg}(m-m',n-n';0,0) \\ &= R_{gg}(0,0;m'-m,n'-n). \end{aligned} \tag{3.16}$$

The last two identities are obtained when we set $(k,l) = (-m',-n')$ and $(k,l) = (-m,-n)$. Since the autocorrelation function depends only on the distance between points, it reduces from a four- to a two-dimensional function. Fortunately, many stochastic processes are homogeneous. A deterministic image which additively contains *zero-mean noise*,

$$G' = G + R, \qquad \langle G'\rangle = G, \tag{3.17}$$

is not a homogeneous field, because the mean is a spatially varying quantity. By subtraction of the mean, however, we obtain a homogeneous random field. Some processes show *multiplicative noise*. Multiplicative noise can be converted to additive noise by taking the logarithm of the gray values.

The autocorrelation function for a homogeneous random field takes a much simpler form, since it depends only on the distance between the pixels:

$$R_{gg}(m,n) = \frac{1}{MN}\sum_{m'=0}^{M-1}\sum_{n'=0}^{N-1} G_{m'n'}G_{m'+m,n'+n}. \tag{3.18}$$

This expression includes spatial averaging. For a general homogeneous random field it is not certain that spatial averaging leads to the same mean as the ensemble mean. A random field that meets this criterion is called an *ergodic random field*. Another difficulty concerns indexing. As soon as $(m,n) \neq (0,0)$, the indices run over the range of the matrix. We then have to consider the periodic extension of the matrix, as discussed in Sect. 2.3.5. This is known as *cyclic autocorrelation*.

As many processes consist of a deterministic and a zero-mean random process, it is helpful first to subtract the mean and then to calculate the correlation:

$$C_{gg}(m,n) = \frac{1}{MN}\sum_{m'=0}^{M-1}\sum_{n'=0}^{N-1}(G_{m'n'} - \langle G_{m'n'}\rangle) \\ (G_{m'+m,n'+n} - \langle G_{m'+m,n'+n}\rangle). \tag{3.19}$$

This function is called the *autocovariance*. The autocovariance for zero-shift ($(m,n) = (0,0)$) is equal to the variance.

Now we illustrate the meaning of the autocorrelation function with some examples. First we consider an image containing only zero-mean homogeneous noise. The fluctuations at the individual pixels should be independent of each other. Autocorrelation (and autocovariance) then vanishes except for zero shift, i.e., for a zero pixel distance. For zero shift, the autocovariance is equal to the variance of the noise. This means that the autocorrelation is unequal to zero if the fluctuations at neighboring pixels are not independent. If the autocorrelation gradually decreases with the distance of the pixels, the pixels become more and more statistically independent. We can then define a characteristic length scale over which the gray values at the pixels are correlated to each other. In this sense the autocorrelation function is a description of the interrelation between the gray values of neighboring pixels.

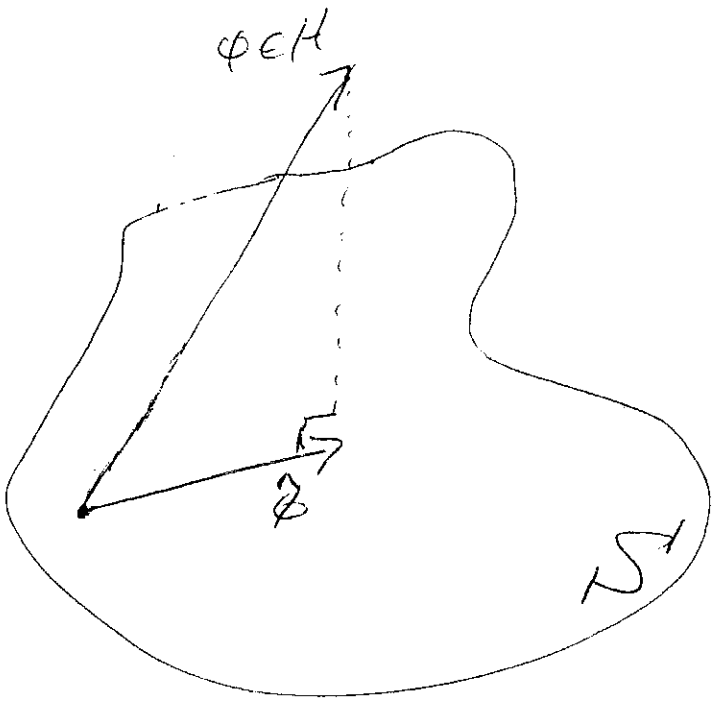
In a similar manner as we correlate one image with itself, we can correlate images from two different homogeneous stochastic processes $\boldsymbol{G}$ and $\boldsymbol{H}$. By analogy to (3.18) and (3.19), the *cross-correlation* function and the *cross covariance* are defined as

$$R_{gh}(k,l) = \frac{1}{MN}\sum_{m'=0}^{M-1}\sum_{n'=0}^{N-1} G_{m'n'}H_{m'+m,n'+n} \tag{3.20}$$

$$C_{gh}(m,n) = \frac{1}{MN}\sum_{m'=0}^{M-1}\sum_{n'=0}^{N-1}(G_{m'n'} - \langle G_{m'n'}\rangle) \\ (H_{m'+m,n'+n} - \langle H_{m'+m,n'+n}\rangle). \tag{3.21}$$

The cross-correlation operation is very similar to *convolution* (Table 2.2, Sect. 4.2). The only difference is the sign of the indices (m,n) in the second term.

$(\phi - \hat{\phi}, v) = 0 \qquad v \in S$



orthogonality principle

Approximation Theory

Given a Hilbert space H with an inner product (,) we define

$$d(\varphi_1, \varphi_2) = ||\varphi_1 - \varphi_2|| = \sqrt{(\varphi_1 - \varphi_2, \varphi_1 - \varphi_2)}$$

Given a subspace $S \subset H$ with a basis $(\psi_1, \psi_2, , ..., \psi_n)$
Any $\widehat{\varphi} \in S$ can be written as $\widehat{\varphi} = \sum_{i=1}^{n} \alpha_i \psi_i$.

Problem: Given $\varphi \in H$ find $\widehat{\varphi} \in S$ so that $||\varphi - \widehat{\varphi}||$ is a minimum

Solution: Choose φ so that

$$(\varphi - \widehat{\varphi}, v) = 0 \quad \text{for every } v \in S$$

Problem

Given $g(x,y)$ find $\hat{f}(x,y)$ so that

$$\varepsilon^2 = E\left\{\left[f(x,y) - \hat{f}(x,y)\right]^2\right\} \quad \text{is minimum}$$

Solution

$$\hat{f}(x,y) = E\left\{ f(x,y) \mid g(x',y') \text{ for all } (x',y') \in R^2 \right\}$$

TRUE but USELESS !!

REVISED PROBLEM

Given $g(x,y)$ find $\hat{f}(x,y)$ linear in g so that ε^2 is minimized

$$\hat{f}(x,y) = \iint g(x',y')\, m(x,y,x',y')\, dx'\, dy'$$

or using the cross-correlation

$$\begin{aligned} R_{fg}(x,y,\widetilde{x},\widetilde{y}) &= E\left[\widehat{f}(x,y)g(\widetilde{x},\widetilde{y})\right] \\ R_{gg}(x\acute{},y\acute{},\widetilde{x},\widetilde{y}) &= E\left[g(x\acute{},y\acute{})g(\widetilde{x},\widetilde{y})\right] \end{aligned}$$

$$R_{fg}(x,y,\widetilde{x},\widetilde{y}) = \iint\limits_{\text{image}} R_{gg}(x\acute{},y\acute{},\widetilde{x},\widetilde{y})m(x,x\acute{};y,y\acute{})dx\acute{}dy\acute{}$$

This is still very difficult so we assume that all the statistics are homogeneous and invariant. So

$$\begin{aligned} R_{fg}(x,y,\widetilde{x},\widetilde{y}) &= R_{fg}(x-\widetilde{x},y-\widetilde{y}) \\ R_{gg}(x\acute{},y\acute{},\widetilde{x},\widetilde{y}) &= R_{gg}(x\acute{}-\widetilde{x},y-\widetilde{y}) \\ R_{nn}(x\acute{},y\acute{},\widetilde{x},\widetilde{y}) &= R_{nn}(x\acute{}-\widetilde{x},y-\widetilde{y}) \\ m(x,x\acute{};y,y\acute{}) &= m(x-x\acute{}y-y\acute{}) \end{aligned}$$

Then

$$\begin{aligned} R_{fg}(x-\widetilde{x},y-\widetilde{y}) &= \iint\limits_{\text{image}} R_{gg}(x\acute{}-\widetilde{x},y\acute{}-\widetilde{y})m(x-x\acute{},y-y\acute{})dx\acute{}dy\acute{} \\ &= R_{gg}(x\acute{}-\widetilde{x},y\acute{}-\widetilde{y}) * m(x,y) \end{aligned}$$

Fourier transform yields

$$S_{fg}(u,v) = S_{gg}(u,v)H(u,v)$$

but from (*) we have

$$g(x,y) = \iint h(x,x\acute{};y,y\acute{})f(x\acute{},y\acute{})dx\acute{}dy\acute{}+n(x,y) = \iint h(x-x\acute{},y-y\acute{})f(x\acute{},y\acute{})dx\acute{}dy\acute{}+n(x,y)$$

and in Fourier space

$$G = HF + N$$

Wiener Filter

Notation:
$f(x,y)$ = "original" picture
$g(x,y)$= "actual" picture
$\widehat{f}(x,y)$ = estimate to $f(x,y)$

Assumption: the noise is additive. So

$$(*) \quad g(x,y) = \iint h(x,x';y,y')f(x',y')dx'dy' + n(x,y)$$

We **assume** $h(x,x';y,y')$ is known (?)
Assumption:

1. $E\left[n(x,y)\right] = 0$
2. $\left.\begin{array}{l} E\left[f(x,y)f(x',y')\right] = R_{ff}(x,y,x',y') \\ E\left[n(x,y)n(x',y')\right] = R_{nn}(x,y,x',y') \end{array}\right\}$ known

Problem:
Given $g(x,y)$ find $\widehat{f}(x,y)$ so that

$$\varepsilon^2 = E\left[\left(f(x,y) - \widehat{f}(x,y)\right)^2\right] \text{ is a minimum}$$

Solution:

To get a meaningful solution we need to assume that $\widehat{f}(x,y)$ is a linear function of $g(x,y)$.

$$\widehat{f}(x,y) = \iint m(x,x';y,y')g(x',y')dx'dy'$$

We need to calculate $m(x,x';y,y')$ as a function of g and h
We wish that

$$E\left[\left(\widehat{f}(x,y) - \iint m(x,x';y,y')g(x',y')dx'dy'\right)^2\right] \text{ be a minimum}$$

Using the orthogonality theorem we have

$$E\left[\left(\widehat{f}(x,y) - \iint m(x,x';y,y')g(x',y')dx'dy', g(\widetilde{x},\widetilde{y})\right)\right] = 0 \quad \text{all } \widetilde{x},\widetilde{y}$$

Interchanging the order of the integrals

$$E\left[\widehat{f}(x,y)g(\widetilde{x},\widetilde{y})\right] = \iint E\left[m(x,x';y,y')g(x',y')g(\widetilde{x},\widetilde{y})dx'dy'\right]$$

So we wish

$$E\left\{ \left[f(x,y) - \iint g(x',y')\, m(x,y,x',y')\, dx'dy' \right]^2 \right\} = \min$$

Use orthogonality theorem

$$E\left\{ \left(f - \iint g\, m\, dx'dy',\ g(\tilde{x},\tilde{y}) \right) \right\} = 0 \qquad \text{all } \tilde{x},\tilde{y}$$

or interchanging order of integrals

$$E\{ f(x,y)\, g(\tilde{x},\tilde{y}) \} = \iint E\{ g(\tilde{x},\tilde{y})\, g(x',y') \}\, m(x,y,x',y'$$

or

$$R_{fg}(x,y;\tilde{x},\tilde{y}) = \iint_{\text{image}} R_{gg}(\tilde{x},\tilde{y},x',y')\, m(x,y;x',y'$$
$$dx';dy'$$

This is still difficult.
So assume all statistics are homogeneous & invariant

i.e. $R_{ff}(x,y;x',y') = R_{ff}(x-x', y-y')$

$R_{nn}(x,y;x',y') = R_{nn}(x-x', y-y')$

and $h(x,x',y,y') = h(x-x', y-y')$

$\Downarrow$

$R_{fg}(x,y,x',y') = R_{fg}(x-x', y-y')$

So we have

$$R_{fg}(x-\tilde{x}, y-\tilde{y}) = \iint R_{gg}(x'-\tilde{x}, y'-\tilde{y})\, m(x-x', y-y')\, dx'$$

or

$$R_{fg}(x-\tilde{x}, y-\tilde{y}) = R_{gg}(x-\tilde{x}, y-\tilde{y}) * m(x,y)$$

(convolution)

Fourier Transform

$\Downarrow$

$$S_{fg}(u,v) = S_{gg}(u,v)\, M(u,v)$$

but

$$g(x,y) = \iint h(x-x', y-y') f(x',y') + n(x,y)$$

$$\Downarrow$$

$$S_{fg}(u,v) = S_{ff}(u,v)\, H^*(u,v)$$

&

$$S_{gg}(u,v) = S_{ff}(u,v)\,|H(u,v)|^2 + S_{nn}(u,v)$$

So

$$M(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(u,v)} = \frac{S_{ff}(u,v)H^*(u,v)}{S_{ff}(u,v)\,|H(u,v)|^2 + S_{nn}(u,v}$$

$$= \frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{S_{nn}(u,v)}{S_{ff}(u,v)}}$$

Def: $\dfrac{S_{nn}(u,v)}{S_{ff}(u,v)} = \dfrac{N}{S}$ = noise to signal ratio at (u,v)

So

$\dfrac{N}{S} \sim 0 \implies M(u,v) = \dfrac{H^*(u,v)}{|H(u,v)|^2}$ inverse filter

So optimal solution emphesizes regions where $\frac{N}{S}$ is small

Wiener Filter

$$M(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_{nn}(u,v)}{S_{ff}(u,v)}}$$

H = Fourier transform of "blurring

S_{nn} = Fourier transform of R_{nn} (noise)

S_{ff} = Fourier transform of R_{ff} (power spectrum)

$$R_{ff} = E\left\{ ff^t \right\}$$

$$\left(R_{ff}\right)_{ij} = E\left\{ f_i f_j^t \right\}$$

$$R_{nn} = E\left\{ nn^t \right\}$$

R_{ff} , R_{nn} are real and symmetric

Wiener Filter

$$M(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_{nn}(u,v)}{S_{ff}(u,v)}}$$

H = Fourier transform of "blurring

S_{nn} = Fourier transform of R_{nn} (noise)

S_{ff} = Fourier transform of R_{ff} (power spectrum)

$$R_{ff} = E\left\{ ff^t \right\}$$

$$\left(R_{ff}\right)_{ij} = E\left\{ f_i f_j^t \right\}$$

$$R_{nn} = E\left\{ nn^t \right\}$$

R_{ff} , R_{nn} are real and symmetric

Wiener Filter

minimize $e^2 = E\left\{\left(f - \tilde{f}\right)^2\right\}$

$$F(u,v) = \left\{\frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_{nn}(u,v)}{S_{ff}(u,v)}}\right\} G(u,v)$$

H = degradation

$S_{nn} = |N(u,v)|^2$ power spectrum of noise

$S_{ff} = |F(u,v)|^2$ power spectrum of **undegraded** image

Sometimes

$$\hat{F}(u,v) = \left\{\frac{H^*(u,v)}{|H(u,v)|^2 + K}\right\} G(u,v)$$

Inverse:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

Minimization of equation (4.88) is easy if the estimate $\hat{f}$ is a linear combination of the values in the image g; the estimate $\hat{f}$ is then close (but not necessarily equal) to the theoretical optimum. The estimate is equal to the theoretical optimum only if the stochastic processes describing images f, g, and the noise ν are homogeneous, and their probability density is Gaussian [Andrews and Hunt 77]. These conditions are not usually fulfilled for typical images.

Denote the Fourier transform of the Wiener filter by H_W. Then, the estimate $\hat{F}$ of the Fourier transform F of the original image f can be obtained as

$$\hat{F}(u,v) = H_W(u,v)\ G(u,v) \tag{4.89}$$

The function H_W is not derived here, but may be found elsewhere [Papoulis 65, Rosenfeld and Kak 82, Bates and McDonnell 86, Gonzalez and Woods 92]. The result is

$$H_W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + [S_{\nu\nu}(u,v)/S_{ff}(u,v)]} \tag{4.90}$$

where H is the transform function of the degradation, $*$ denotes complex conjugate, $S_{\nu\nu}$ is the spectral density of the noise, and S_{ff} is the spectral density of the undegraded image.

If Wiener filtration is used, the nature of degradation H and statistical parameters of the noise need to be known. Wiener filtration theory solves the problem of optimal a posteriori linear mean square estimates—all statistics (for example, power spectrum) should be available in advance. Note the term $S_{ff}(u,v)$ in equation (4.90), which represents the spectrum of the undegraded image. This information may be difficult to obtain considering the goal of image restoration, to determine the undegraded image.

Note that the ideal inverse filter is a special case of the Wiener filter in which noise is absent, i.e., $S_{\nu\nu} = 0$.

Restoration is illustrated in Figures 4.29 and 4.30. Figure 4.29a shows an image that was degraded by 5 pixels motion in the direction of the x axis, and Figure 4.29b shows the result of restoration where Wiener filtration was used. Figure 4.30a shows an image degraded by wrong focus and Figure 4.30b is the result of restoration using Wiener filtration.

Despite its unquestionable power, Wiener filtration suffers several substantial limitations. First, the criterion of optimality is based on minimum mean square error and weights all errors equally, a mathematically fully acceptable criterion that unfortunately does not perform well if an image is restored for human viewing. The reason is that humans perceive the restoration errors more seriously in constant-gray-level areas and in bright regions, while they are much less sensitive to errors located in dark regions and in high-gradient areas. Second, spatially variant degradations cannot be restored using the standard Wiener filtration approach, and these degradations are common. Third, most images are highly non-stationary, containing large homogeneous areas separated by high-contrast edges. Wiener filtration cannot handle non-stationary signals and noise. To deal with real-life image degradations, more sophisticated approaches may be needed. Examples include **power spectrum equalization** and **geometric mean filtration**. These and other specialized restoration techniques can be found in higher-level texts devoted to this topic; [Castleman 96] is well suited for such a purpose.

3.4.2 Wiener Filter

The Wiener filter, also called a minimum mean-square estimator (developed by Norbert Wiener in 1942), alleviates some of the difficulties inherent in inverse filtering by attempting to model the error in the restored image through the use of statistical methods. After the error is modeled, the average error is mathematically minimized, thus the term *minimum mean-square estimator*. The resulting equation is the Wiener filter:

$$R_W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \left[\dfrac{S_n(u, v)}{S_I(u, v)}\right]}$$

where $H^*(u, v)$ = complex conjugate of $H(u, v)$

$S_n(u, v) = |N(u, v)|^2$ = power spectrum of the noise

$S_I(u, v) = |I(u, v)|^2$ = power spectrum of the original image

This equation assumes a square image of size $N \times N$. The complex conjugate can be found by negating the imaginary part of a complex number. Other practical considerations are discussed in Section 3.4.6. Examining this equation will provide us with some understanding of how it works.

If we assume that the noise term $S_n(u, v)$ is zero, this equation reduces to an inverse filter since $|H(u, v)|^2 = H^*(u, v)H(u, v)$. As the noise term increases, the denominator of the Wiener filter increases, thus decreasing the value of $R_W(u, v)$. Thus, as the contribution of the noise increases, the filter gain decreases. This seems reasonable—in portions of the spectrum uncontaminated by noise we have an inverse filter, whereas in portions of the spectrum heavily corrupted by noise, the filter attenuates the signal, with the amount of attenuation being determined by the ratio of the noise spectrum to the uncorrupted image spectrum.

The Wiener filter is applied by multiplying it by the Fourier transform of the degraded image, and the restored image is obtained by taking the inverse Fourier transform of the result, as follows:

$$\hat{I}(r, c) = F^{-1}\left[\hat{I}(u, v)\right] = F^{-1}\left[R_W(u, v)D(u, v)\right]$$

Figure 3.4-5 compares the inverse filter and the Wiener filter. The filters are applied to images that have been blurred and then had various amounts of gaussian noise added. With small amounts of noise, the inverse filter works adequately, but when the noise level is increased, the Wiener filter results are obviously superior.

In practical applications the original, uncorrupted image is not typically available, so the power spectrum ratio is replaced by a parameter K whose optimal value must be experimentally determined:

$$R_W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K}$$

Making the K parameter a function of the frequency domain variables (u, v) may also provide some added benefits. Because the noise typically dominates at high frequencies,

Phase of the Wiener Filter. Equation (8.41) can be written as

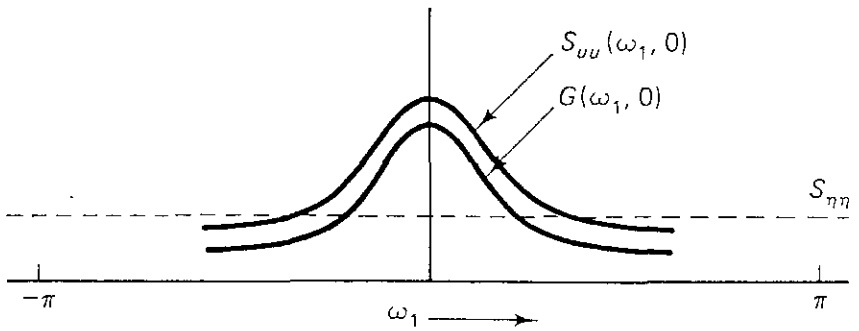
$$\left.\begin{aligned} G &= |G| e^{\,j\theta_G} \\ |G| &= \frac{|H| S_{uu}}{|H|^2 S_{uu} + S_{\eta\eta}} \\ \theta_G &= \theta_G(\omega_1, \omega_2) = \theta_{H^*} = -\theta_H = \theta_{H^{-1}} \end{aligned}\right\} \tag{8.46}$$

that is, *the phase of the Wiener filter is equal to the phase of the inverse filter* (in the frequency domain). Therefore, the Wiener filter or, equivalently, the mean square criterion of (8.28), does not compensate for phase distortions due to noise in the observations.

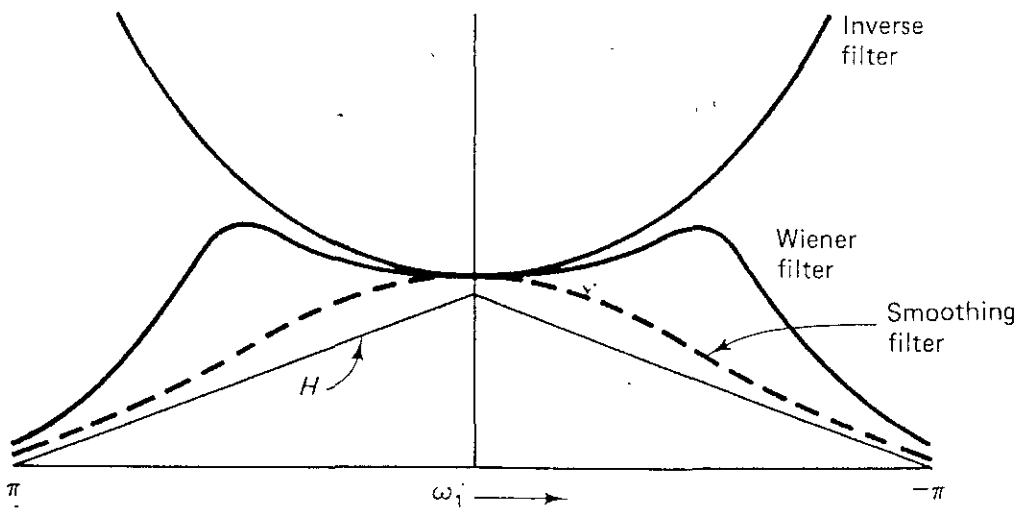
Wiener Smoothing Filter. In the absence of any blur, $H = 1$ and the Wiener filter becomes

$$G|_{H=1} = \frac{S_{uu}}{S_{uu} + S_{\eta\eta}} = \frac{S_{nr}}{S_{nr} + 1} \tag{8.47}$$

where $S_{nr} \triangleq S_{uu}/S_{\eta\eta}$ defines the signal-to-noise ratio at the frequencies (ω_1, ω_2). This is also called the *(Wiener) smoothing filter.* It is a zero-phase filter that depends only



(a) Noise smoothing ($H = 1$)



(b) Deblurring

Figure 8.11 Wiener filter characteristics.

DISCRETE WIENER FILTER

As before we find

$$M^{opt} = R_{ff} H^T (H R_{ff} H^T + R_{nn})^{-1}$$

$$\hat{f}^{opt} = R_{ff} H^T (H R_{ff} H^T + R_{nn})^{-1} g$$

Assume all matrices can be simultaneously diagonalized !

$$\Downarrow$$

$$F^{opt}(i) = \left[\frac{h(i)}{h^2(i) + \frac{\lambda_{nn}(i)}{\lambda_{ff}(i)}} \right] G(i)$$

i.e. component-wise processing

So Simultaneous diagonalization $\Longleftrightarrow$ spatially invariant

Def:

$$R(u_1, u_2, x_1, x_2) = Prob\left[u_1 = x_1 , u_2 = x_2 \right] \quad x_1, x_2 = 0, \ldots, L-1$$

$$= \frac{\text{number of pairs of pixels } u_1 = x_1 + u_2 = x_2}{\text{total number of pixels in region}}$$

on the signal-to-noise ratio S_{nr}. For frequencies where $S_{nr} \gg 1$, G becomes nearly equal to unity which means that all these frequency components are in the passband. When $S_{nr} \ll 1$, $G = S_{nr}$; that is, all frequency components where $S_{nr} \ll 1$, are attenuated in proportion to their signal-to-noise ratio. For images, S_{nr} is usually high at lower spatial frequencies. Therefore, the noise smoothing filter is a low-pass filter (see Fig. 8.11a).

Relation with Inverse Filtering. In the absence of noise, we set $S_{\eta\eta} = 0$ and the Wiener filter reduces to

$$G|_{S_{\eta\eta}=0} = \frac{H^* S_{uu}}{|H|^2 S_{uu}} = \frac{1}{H} \tag{8.48}$$

which is the inverse filter. On the other hand, taking the limit $S_{\eta\eta} \to 0$, we obtain

$$\lim_{S_{\eta\eta}\to 0} G = \begin{cases} \dfrac{1}{H}, & H \neq 0 \\ 0, & H = 0 \end{cases} = H^- \tag{8.49}$$

which is the pseudoinverse filter. Since the blurring process is usually a low-pass filter, the Wiener filter acts as a high-pass filter at low levels of noise.

Interpretation of Wiener Filter Frequency Response. When both noise and blur are present, the Wiener filter achieves a compromise between the low-pass noise smoothing filter and the high-pass inverse filter resulting in a band-pass filter (see Fig. 8.11b). Figure 8.12 shows Wiener filtering results for noisy blurred images. Observe that the deblurring effect of the Wiener filter diminishes rapidly as the noise level increases.

Wiener Filter for Diffraction Limited Systems. The Wiener filter for the continuous observation model, analogous to (8.39)

$$v(x, y) = \int\int_{-\infty}^{\infty} h(x - x', y - y')u(x', y')dx'dy' + \eta(x, y) \tag{8.50}$$

is given by

$$G(\xi_1, \xi_2) = \frac{S_{uu}(\xi_1, \xi_2)H^*(\xi_1, \xi_2)}{|H(\xi_1, \xi_2)|^2 S_{uu}(\xi_1, \xi_2) + S_{\eta\eta}(\xi_1, \xi_2)} \tag{8.51}$$

For a diffraction limited system, $H(\xi_1, \xi_2)$ will be zero outside a region, say $\mathcal{R}$, in the frequency plane. From (8.51), G will also be zero outside $\mathcal{R}$. Thus, *the Wiener filter cannot resolve beyond the diffraction limit.*

(a) Blurred with small noise

(b) Restored image (a)

(c) Blurred with increased noise

(d) Restored image (c)

Figure 8.12 Wiener filtering of noisy blurred images.

One Dimensional Motion

$$g(x,y) = \int_0^T f(x - x_0(t), y - y_0(t))dt$$

Fourier transform

$$\begin{aligned} G(m,n) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-2\pi i(mx+ny)}dxdy \\ &= \int_0^T \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-x_0, y-y_0)e^{-2\pi i(mx+ny)}dxdy\right)dt \\ &= \int_0^T F(u,v)e^{-2\pi i(mx_0+ny_0)}dt \end{aligned}$$

So the transfer function is

$$H(m,n) = \int_0^T e^{-2\pi i(mx_0(t)+ny_0(t))}dt$$

Assume $x_0 = \frac{\alpha t}{T}$ $y_0 = 0$. Then

$$H(m,n) = \int_0^T e^{-2\pi i m\frac{\alpha t}{T}}dt = Te^{-\pi i m\alpha}\frac{\sin(\pi m\alpha)}{\pi m\alpha}$$

Note that $H(m,n) = 0$ whenever $m\alpha$ is an integer.
Using sum instead of integral we get
i_T = total number of pixels recorded by the same cell of the camera
N = total number of pixels in a row of the image

$$H(m,n) = \frac{1}{i_T}\frac{\sin(\frac{nm}{N}i_T)}{\sin(\frac{nm}{N})}e^{-\pi i\frac{m}{N}\alpha}$$

Wiener Filter

$$M(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_{nn}(u,v)}{S_{ff}(u,v)}}$$

Note: $S_{ff}(u,v)$ is autocorrelation of the **original** picture

Assumption: Noise is white:

$$S_{nn}(u,v) = S_{nn}(0,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_{nn}(x,y)dxdy$$

can be found when there is no image $f(x,y) = 0$

Wierner-Khinchine Theorem

The spatial autocorrelation function of a random field f(x,y) is equal to the spectral density $|F(u,v)|^2$

example: blurred image

$$\begin{aligned} H(m,n) &= \frac{1}{i_T}\frac{\sin(\frac{nm}{N}i_T)}{\sin(\frac{nm}{N})}e^{-\pi i\frac{m}{N}\alpha} \\ |H(m,n)|^2 &= \frac{1}{i_T^2}\frac{\sin^2(\frac{nm}{N}i_T)}{\sin^2(\frac{nm}{N})} \end{aligned}$$

iener2

gorithm

`wiener2` estimates the local mean and variance around each pixel

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2)$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2$$

where η is the N-by-M local neighborhood of each pixel in the image `A`. `wiener2` then creates a pixel-wise Wiener filter using these estimates

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(a(n_1, n_2) - \mu)$$

where ν^2 is the noise variance. If the noise variance is not given, `wiener2` uses the average of all the local estimated variances.

e Also

`filter2`, `medfilt2`

ference

Lim, Jae S. *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1990. pp. 536-540.

If the restored image exhibits ringing introduced by the discrete Fourier transform used in the algorithm, it sometimes helps to use function `edgetaper` prior to calling `deconvwnr`. The syntax is

`edgetaper`

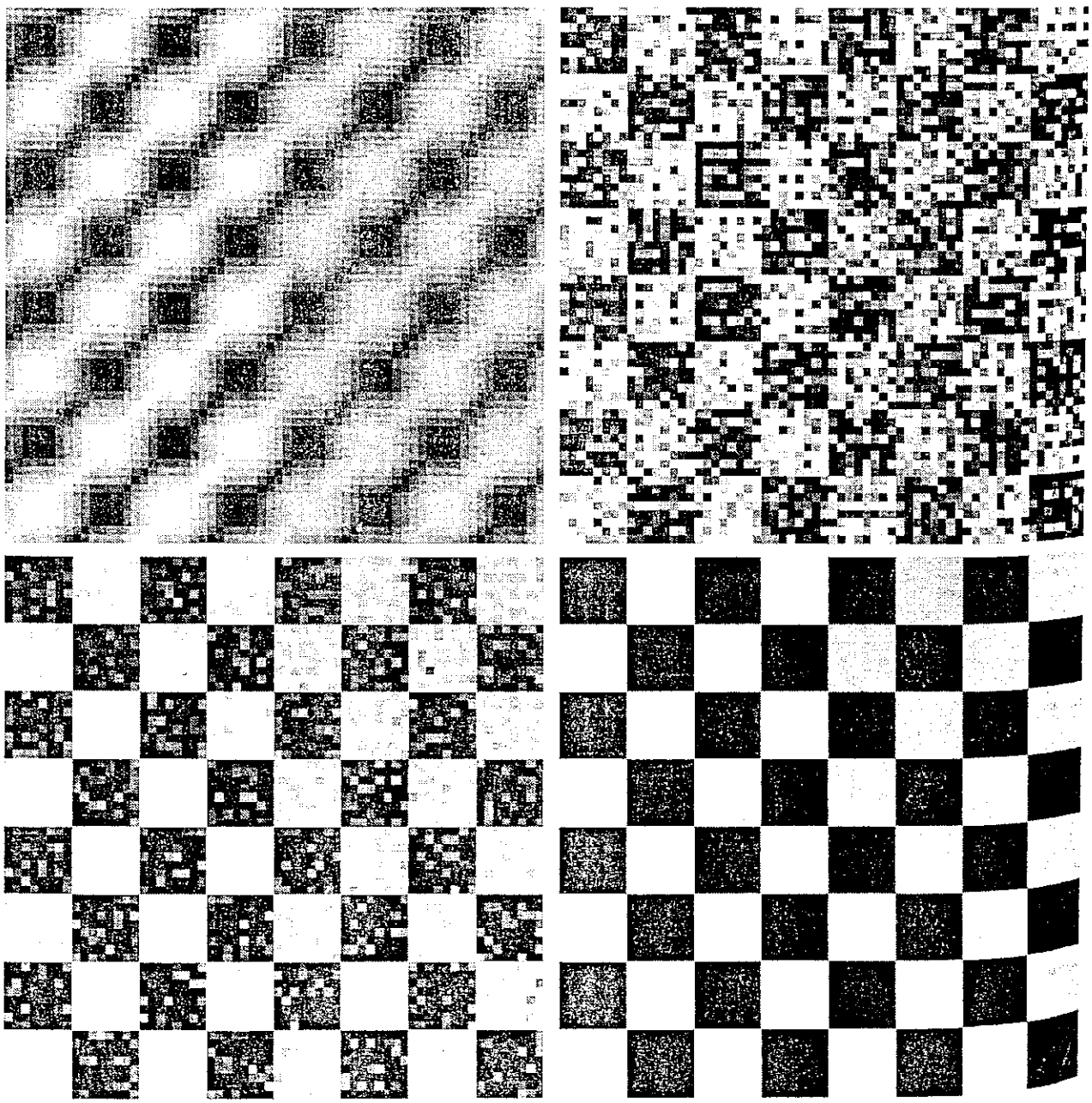
```
J = edgetaper(I, PSF)
```

This function blurs the edges of the input image, `I`, using the point spread function, `PSF`. The output image, `J`, is the weighted sum of `I` and its blurred version. The weighting array, determined by the autocorrelation function of `PSF`, makes `J` equal to `I` in its central region, and equal to the blurred version of `I` near the edges.

AMPLE 5.8: ig function onvwnr to ore a blurred, y image.

Figure 5.8(a) is the same as Fig. 5.7(d), and Fig. 5.8(b) was obtained using the command

```
>> fr1 = deconvwnr(g, PSF);
```



RE 5.8 Blurred, noisy ge. (b) Result verse ring. Result of ner filtering g a constant . (d) Result iener filtering g correlation tions.

Inverse filtering

Problem:
- Given g and "some" information about H & n
- find f

Solutions:
Inverse filtering is bad: (too much noise)

$$\hat{f} = \frac{g}{H} \quad \Rightarrow \quad \hat{f} = f + \frac{n}{H}$$

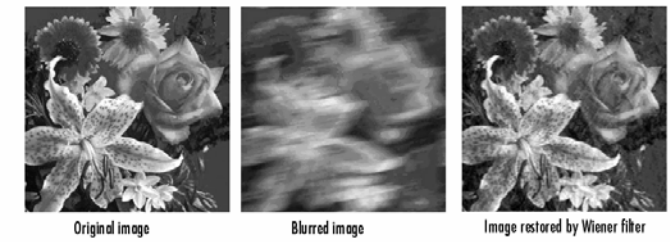
Better approach:
Minimize some error functional such as: $E\left\{(f-\hat{f})^2\right\}$
(the expectation value of the squared difference)

Deblurring with Wiener

```
% I is flowers.tif (croped)
LEN = 31;
THETA = 11;
PSF = fspecial('motion',LEN,THETA);
Blurred = imfilter(I,PSF,'circular','conv');

wnr1 = deconvwnr(Blurred,PSF);
```

Without noise, this is an inverse filter.

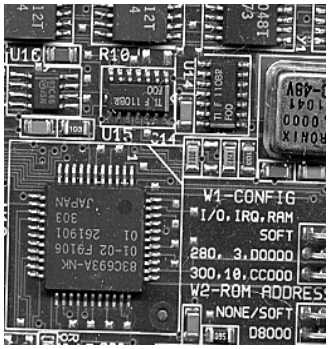



Dr. Yoram Tal

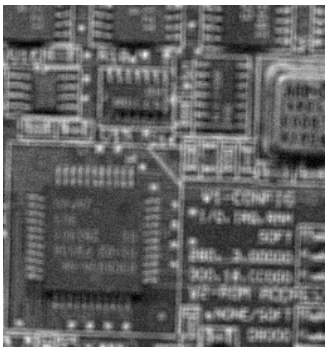

Additive Noise

```
PSF = fspecial('gaussian',5,5);
Blurred = imfilter(I,PSF,'symmetric','conv');
V = .002;
BlurredNoisy = imnoise(Blurred,'gaussian',0,V);
```

Source image



Blurred & noised image




Dr. Yoram Tal

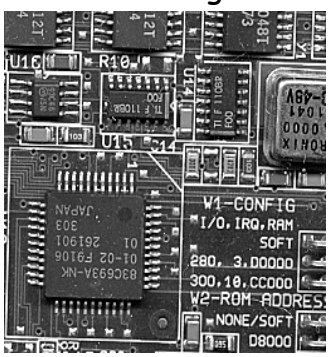

Deblurring noisy images

Lucy–Richardson deconvolution

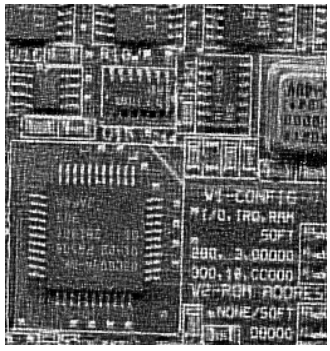
Number of iterations

```
luc1 = deconvlucy(BlurredNoisy,PSF,15);
```

Source image



deblurred image



Methods

deconvblind Restore image using blind deconvolution

deconvlucy Restore image using accelerated Richardson-Lucy algorithm.

deconvreg Restore image using Regularized filter

deconvwnr Restore image using Wiener filter

Deblurring Model

Image degradation can be approximately described by this equation

$$\mathbf{g} = \mathbf{H} \cdot \mathbf{f} + \mathbf{n}, \quad \text{where:}$$

g = The blurred image
H = The distortion operator, also called the point-spread function (PSF). This function, when convolved with the image, creates the distortion
f = The original undegraded image
n = Additive noise, introduced during image acquisition, that corrupts the image.

Constrained Matrix

$$\begin{aligned} g &= Hf + n \\ n &= g - Hf \end{aligned}$$

Assume statistics of the noise are known, eg $n^T n = \epsilon$

Let Lf be a functional of the picture e.g. discrete Laplacian. We wish to find

$$\begin{aligned} (Lf)^T (LF) &= \text{minimum} \\ (g - Hf)^T (g - Hf) &= \epsilon \end{aligned}$$

Use Lagrange multipliers

$$\begin{aligned} f &= \left[H^* H + \gamma L^* L\right]^{-1} H^* g \\ &= \frac{H^* g}{|H|^2 + \gamma L^* L} \end{aligned}$$

Note instead of $L = Laplacian$ take $L^* L = -Laplacian$.

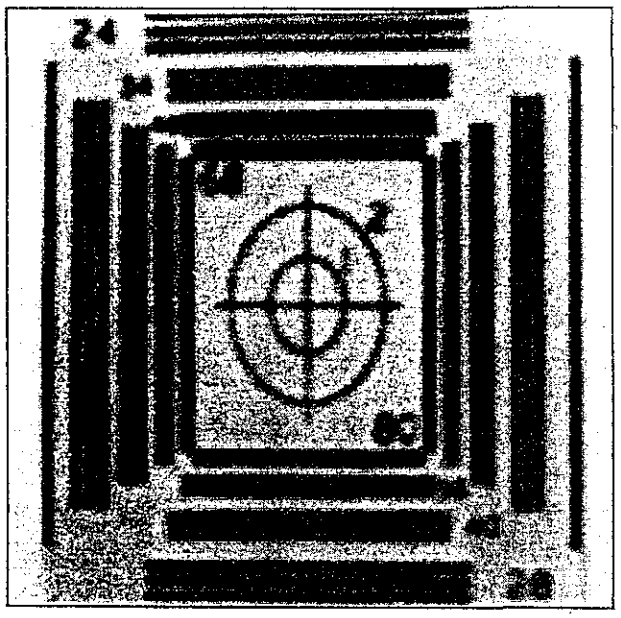
Looks similar to Wiener $L^* L = \frac{S_{nn}}{S_{ff}}$

it seems reasonable to have the value of K increase as the frequency increases, which will cause the filter to attenuate the signal at high frequencies. The following filter applies this idea.

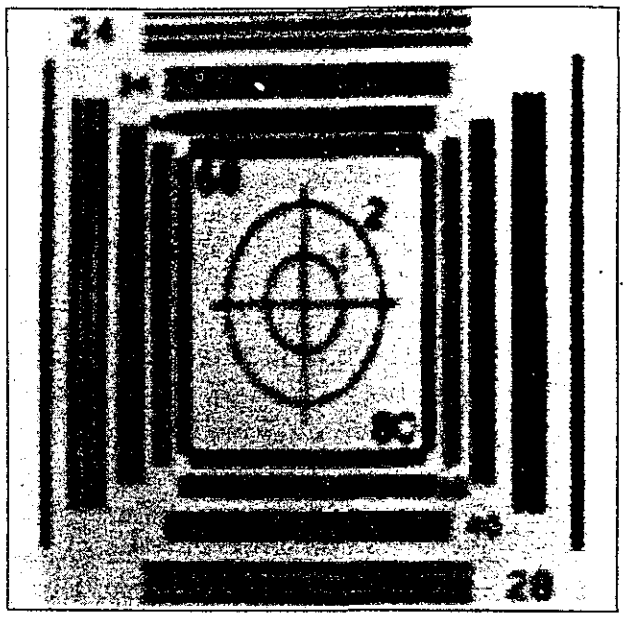
3.4.3 Constrained Least-Squares Filter

The constrained least-squares filter provides a filter that can eliminate some of the artifacts caused by other frequency domain filters. This is done by including a

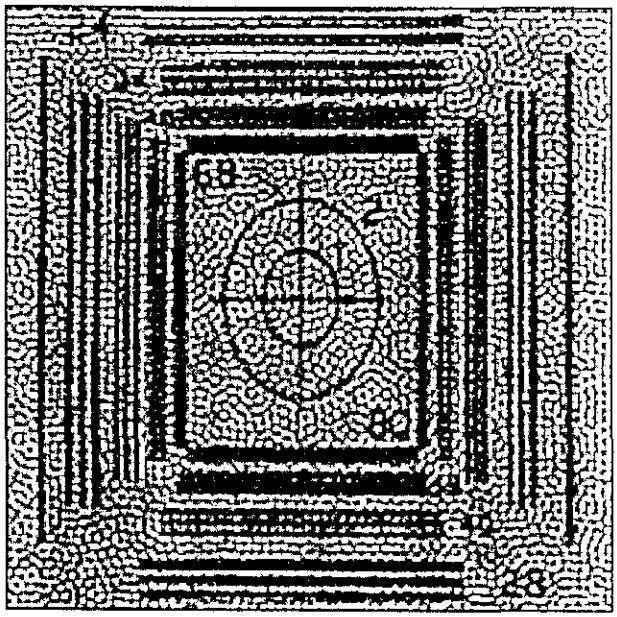
ure 3.4-5 Wiener Filter



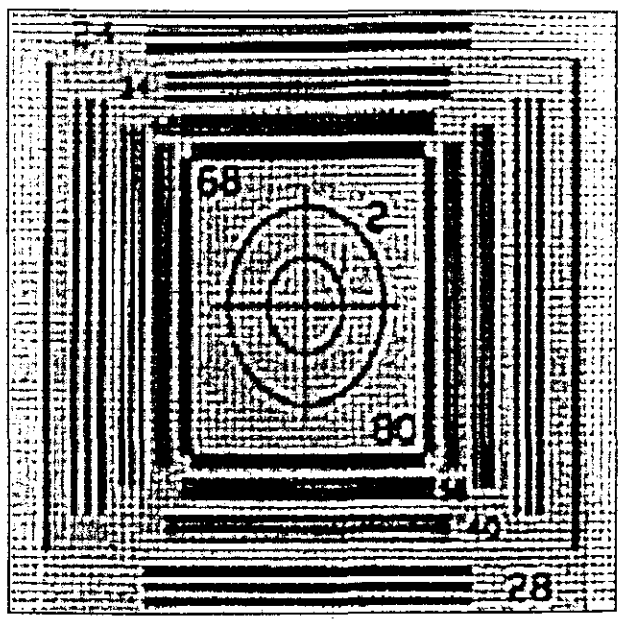
a. Image blurred with an 11 x 11 gaussian convolution mask.



b. Image with gaussian noise—variance = 5; mean = 0.

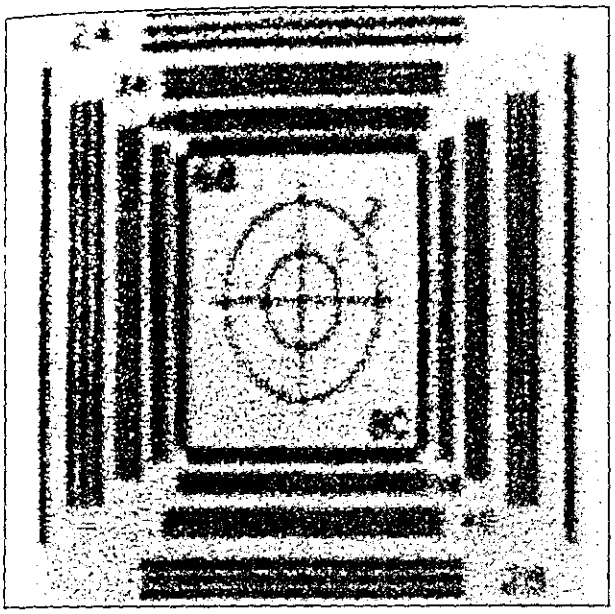


c. Inverse filter, with cutoff frequency = 80, histogram stretched with 3% low and high clipping to show detail.

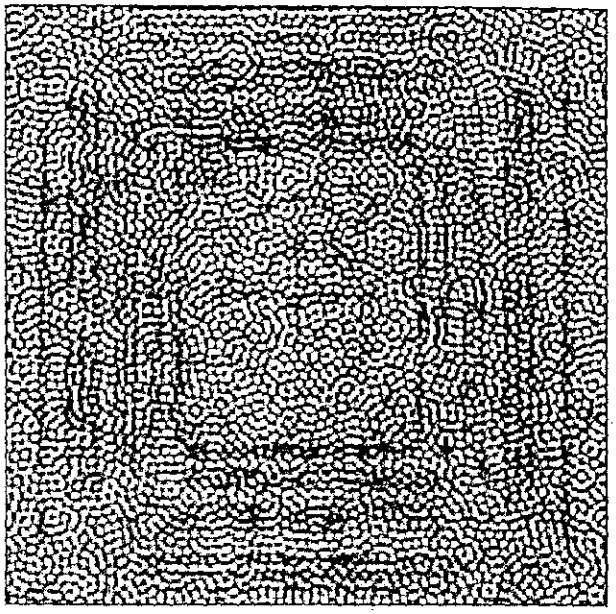


d. Wiener filter, with cutoff frequency = 80, histogram stretched.

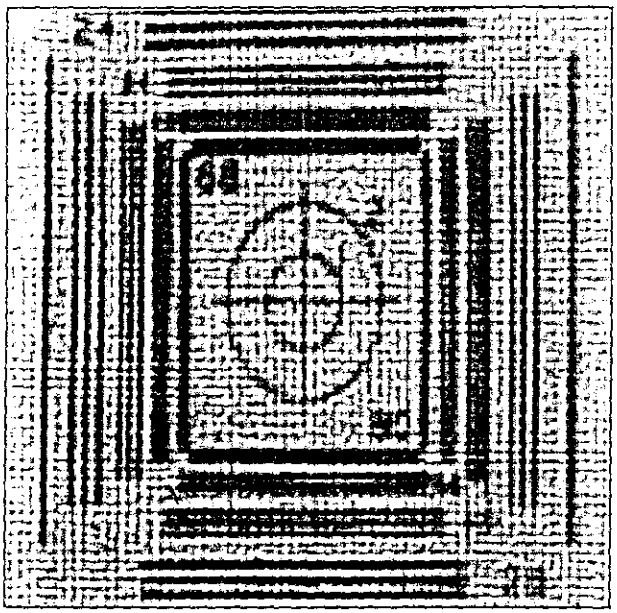
gure 3.4-5 (Continued)



e. Image with gaussian noise—variance = 200; mean = 0.



f. Inverse filter, with cutoff frequency = 80, histogram stretched.



g. Wiener filter, with cutoff frequency = 80, histogram stretched.

smoothing criterion in the filter derivation, so that the result will not have undesirable oscillations (these appear as "waves" in the image), as sometimes occurs with other frequency domain filters. The constrained least-squares filter is given by

$$R_{CLS}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma|P(u, v)|^2}$$

where γ = adjustment factor

$P(u, v)$ = the Fourier transform of smoothness criterion function

The adjustment factor's value is experimentally determined and is application dependent. A standard function to use for $p(r, c)$ (the inverse Fourier transform of $P(u, v)$) is the laplacian filter mask, as follows:

$$p(r, c) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

However, before $P(u, v)$ is calculated, the $p(r, c)$ function must be extended with zeros (zero-padded) to the same size as the image. Figure 3.4-6 shows the results of applying this filter.

The constrained least-squares filter is applied by multiplying it by the Fourier transform of the degraded image, and the restored image is obtained by taking the inverse Fourier transform of the result as follows:

$$\hat{I}(r, c) = F^{-1}\left[\hat{I}(u, v)\right] = F^{-1}\left[R_{CLS}(u, v)D(u, v)\right]$$

3.4.4 Geometric Mean Filters

The geometric mean filter equation provides us with a general form for many of the frequency domain restoration filters. It is defined as follows:

$$R_{GM}(u, v) = \left[\frac{H^*(u, v)}{|H(u, v)|^2}\right]^{\alpha} \left[\frac{H^*(u, v)}{|H(u, v)|^2 + \gamma\left[\frac{S_n(u, v)}{S_f(u, v)}\right]}\right]^{1-\alpha}$$

The terms are as previously defined, with γ and α being positive real constants. If $\alpha = 1/2$ and $\gamma = 1$, this filter is called a *power spectrum equalization filter* (also called a *homomorphic filter*). If $\alpha = 1/2$, then this filter is an average between the inverse filter and the Wiener filter, hence the term geometric mean, although it is standard to refer to the general form of the equation as *geometric mean filter(s)*.

The geometric mean filter is applied by multiplying it by the Fourier transform of the degraded image, and the restored image is obtained by taking the inverse Fourier transform of the result, as follows:

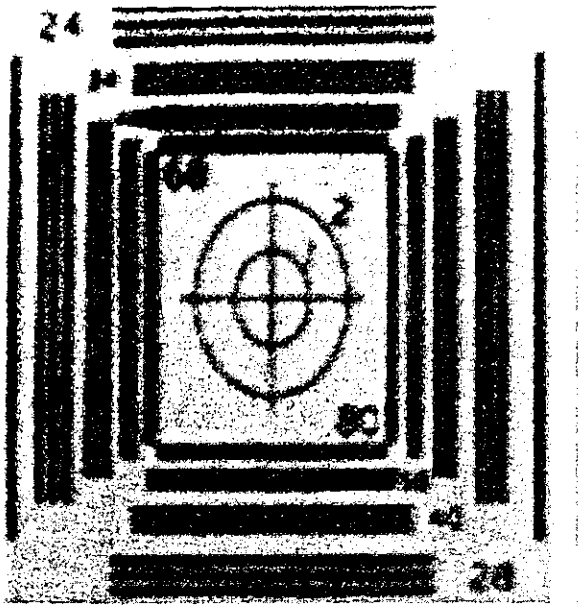
$$\hat{I}(r, c) = F^{-1}\left[\hat{I}(u, v)\right] = F^{-1}\left[R_{GM}(u, v)D(u, v)\right]$$

If $\alpha = 0$, this filter is called a *parametric Wiener filter*. The equation reduces to the Wiener filter equation, but with γ included as an adjustment parameter:

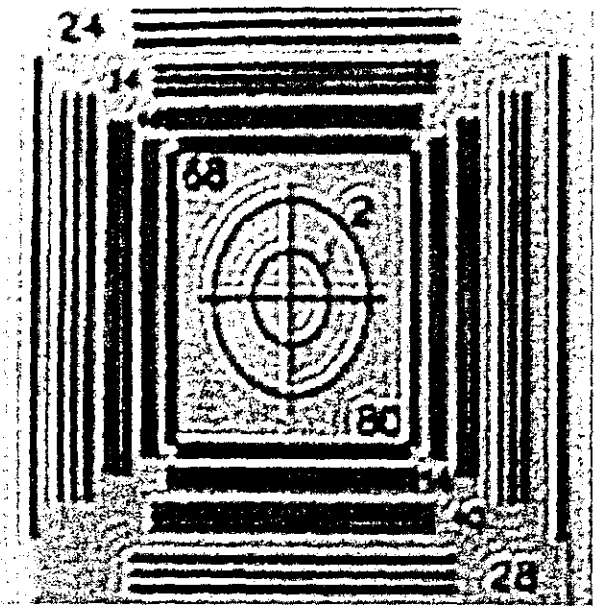
$$R_{PW}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma\left[\frac{S_n(u, v)}{S_f(u, v)}\right]}$$

When $\gamma = 1$, this filter becomes a standard Wiener filter, and when $\gamma = 0$, this filter becomes the inverse filter. As γ is adjusted, the results vary between these two filters, with larger values providing more of the Wiener filtering effect.

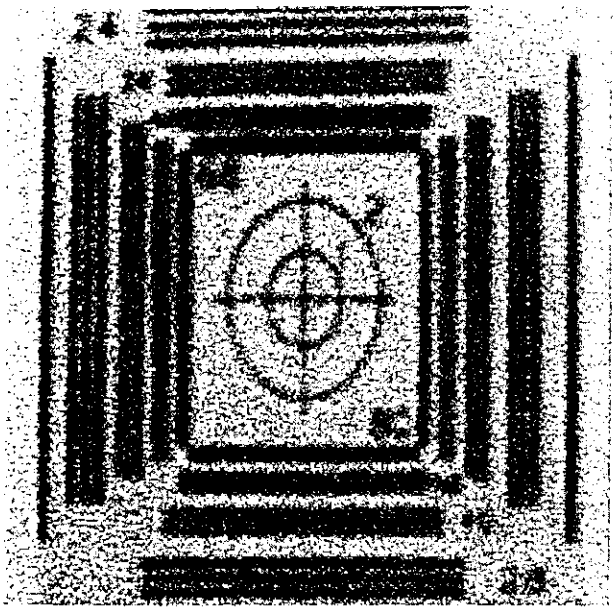
gure 3.4-6 Constrained Least-Squares Filter



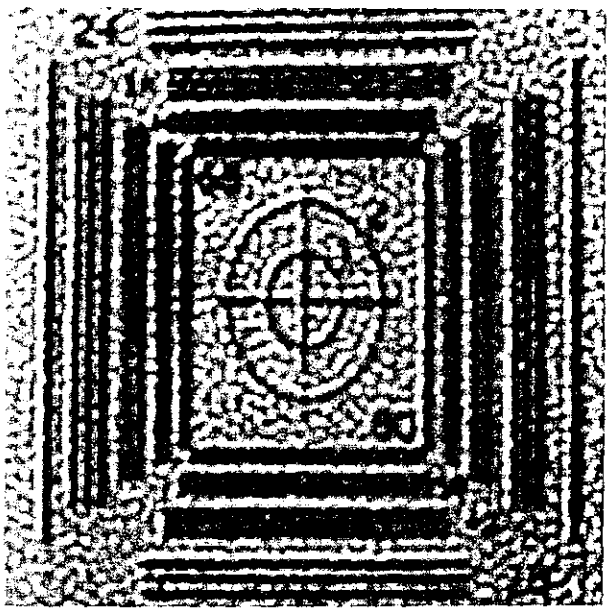
a. Blurred image with added gaussian noise, mean = 0, variance = 5.



b. Result of CLS filter on a.



c. Blurred image with added gaussian noise, mean = 0, variance = 200.



d. Result of CLS filter on c.

The parametric Wiener filter is applied by multiplying it by the Fourier trans form of the degraded image, and the restored image is obtained by taking the invers Fourier transform of the result as follows:

$$\hat{I}(r, c) = F^{-1}\left[\hat{I}(u, v)\right] = F^{-1}\left[R_{PW}(u, v)D(u, v)\right]$$

In general, the frequency domain filters work well for small amounts of blurri and moderate amounts of additive noise. The inverse filter is inadequate with t

a b c
d e f
g h i

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.

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.

For $m = 0$ we must remember to use:

$$F_1(0,n) = \frac{G_1(0,n)}{1+\Gamma} \quad for \quad 0 \leq n \leq N-1$$

$$F_2(0,n) = \frac{G_2(0,n)}{1+\Gamma} \quad for \quad 0 \leq n \leq N-1$$

If we Fourier transform back using functions $F_1(m,n)$ an $F_2(m,n)$ we obtain the restored image shown in Figure *6.9a. This image should be compared with images 6.7e and 6.7f, which are obtained by inverse filtering.*

The restoration of the noisy images of Figures *6.8a and 6.8c by Wiener filtering is shown in* Figures *6.9b and 6.9c. These images should be compared with Figures 6.8g and 6.8i respectively. In all cases Wiener filtering produces superior results.*

(a) Wiener filtering with $\Gamma = 0.01$ of image 6.7b (realistic blurring)

(b) Wiener filtering with $\Gamma = 0.5$ of image 6.8a (realistic blurring with additive Gaussian noise ($\sigma = 10$))

(c) Wiener filtering with $\Gamma = 1$ of image 6.8c (realistic blurring with additive Gaussian noise ($\sigma = 20$))

Figure 6.9: Image restoration with Wiener filtering.

If the degradation process is assumed linear, why don't we solve a system of linear equations to reverse its effect instead of invoking the convolution theorem?

Indeed, the system of linear equations we must invert is given in matrix form by equation (6.10), $\mathbf{g} = H\mathbf{f}$. However, we saw that it is more realistic to include in this equation an extra term representing noise (see equation (6.26)):

(6.75)

(a) Constraint matrix inversion with $\gamma = 0.001$ for image 6.7b (realistic blurring)

(b) Constraint matrix inversion with $\gamma = 0.05$ for image 6.8a (realistic blurring and Gaussian noise ($\sigma = 10$))

(c) Constraint matrix inversion with $\gamma = 0.1$ for image 6.8c (realistic blurring and Gaussian noise ($\sigma = 20$))

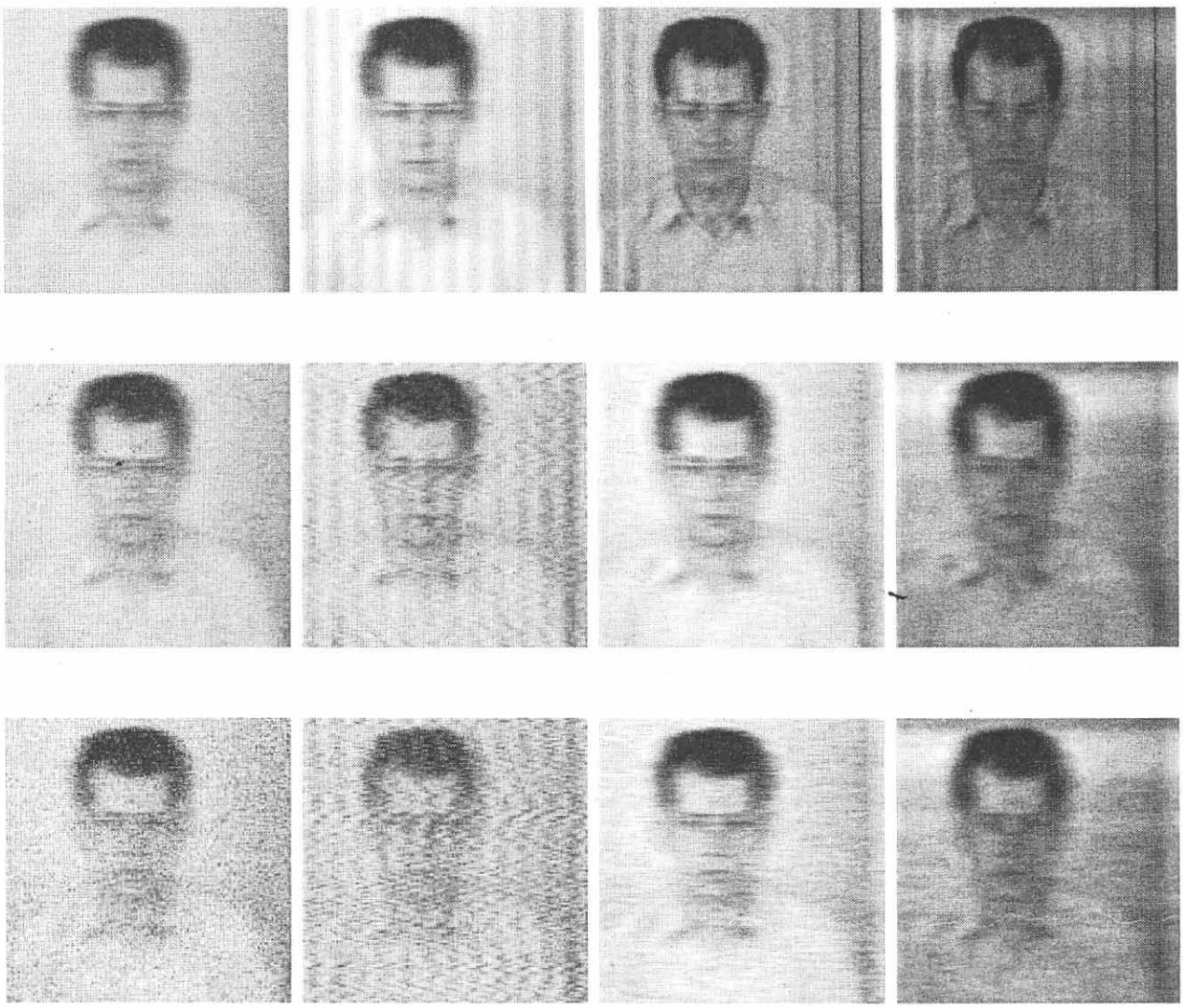
Figure 6.11: Restoration with constraint matrix inversion.

What is the "take home" message of this chapter?

This chapter explored some techniques used to correct (i.e. restore) the damaged values of an image. The problem of restoration requires some prior knowledge concerning the original uncorrupted signal or the imaged scene, and in that way differs from the image enhancement problem. Geometric restoration of an image requires knowledge of the correct location of some reference points.

Grey level restoration of an image requires knowledge of some statistical properties of the corrupting noise, the blurring process and the original image itself. Often, we bypass the requirement for knowing the statistical properties of the original image by imposing some spatial smoothness constraints on the solution, based on the heuristic that "the world is largely smooth". Having chosen the correct model for the degradation process and the uncorrupted image, we have then to solve the problem of recovering the original image values.

The full problem of image restoration is a very difficult one as it is non-linear. It can be solved with the help of local or global optimization approaches. However, simpler solutions can be found, in the form of convolution filters, if we make the assumption that the degradation process is shift invariant and if we restrict the domain of the sought solution to that of linear solutions only. *Figure* 6.12 summarizes the results obtained by the various restoration methods discussed, for an image blurred by motion along the horizontal axis.



(a) (b) (c) (d)

Figure 6.12: (a) Original images. (b) Restoration by inverse filtering with omission of all terms beyond the first zero of the filter transfer function. (c) Restoration by Wiener filtering. (d) Restoration by constraint matrix inversion.

Throughout this chapter, we will rely on a subjective comparison of original, degraded, and processed images by a human observer to illustrate the performance of each image restoration algorithm. In addition when the information is available, we will provide the normalized mean square error (NMSE) between the original image $f(n_1, n_2)$ and the degraded image $g(n_1, n_2)$, and that between the original image $f(n_1, n_2)$ and the processed image $p(n_1, n_2)$. The NMSE between $f(n_1, n_2)$ and $p(n_1, n_2)$ is defined by

$$\text{NMSE } [f(n_1, n_2), p(n_1, n_2)] = 100 \times \frac{\text{Var } [f(n_1, n_2) - p(n_1, n_2)]}{\text{Var } [f(n_1, n_2)]} \% \qquad (9.12)$$

where Var $[\cdot]$ is the variance. Using the variance ensures that the NMSE will not be affected by adding a bias to $p(n_1, n_2)$. The measure NMSE $[f(n_1, n_2), g(n_1, n_2)]$ is similarly defined. The SNR improvement due to processing is defined by

$$\text{SNR improvement} = 10 \log_{10} \frac{\text{NMSE } [f(n_1, n_2), g(n_1, n_2)]}{\text{NMSE } [f(n_1, n_2), p(n_1, n_2)]} \text{ dB}. \qquad (9.13)$$

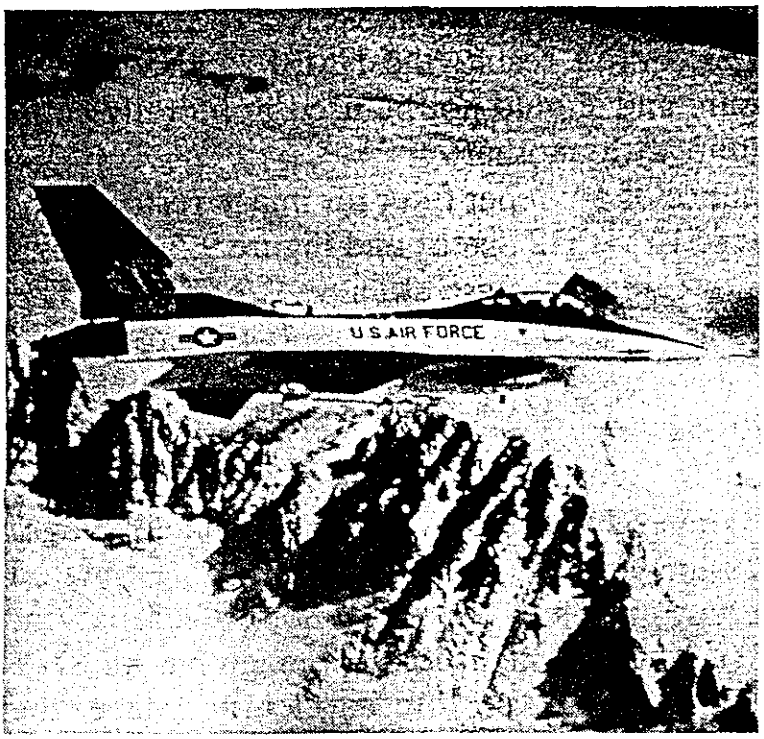
A human observing two images affected by the same type of degradation will generally judge the one with the smaller NMSE to be closer to the original. A very small NMSE generally can be taken to mean that the image is very close to the original. It is important to note, however, that the NMSE is just one of many possible objective measures and can be misleading. When images with different

types of degradation are compared, the one with the smallest NMSE will not necessarily seem closest to the original. As a result, the NMSE and SNR improvements are stated for reference only and should not be used in literally comparing the performance of one algorithm with another.

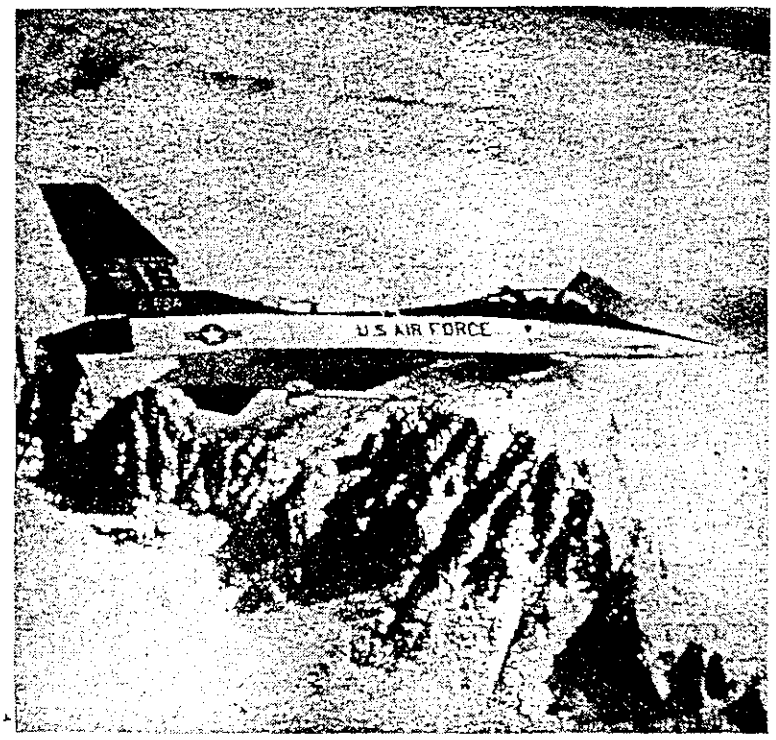
Figure 9.5 illustrates the performance of a Wiener filter for image restoration. Figure 9.5(a) shows an original image of 512 $\times$ 512 pixels, and Figure 9.5(b) shows the image degraded by zero-mean white Gaussian noise at an SNR of 7 dB. The SNR was defined in Chapter 8 as

$$\text{SNR in dB} = 10 \log_{10} \frac{\text{Var } [f(n_1, n_2)]}{\text{Var } [v(n_1, n_2)]}. \qquad (9.14)$$

Figure 9.5(c) shows the result of the Wiener filter applied to the degraded image. In the Wiener filter, $P_v(\omega_1, \omega_2)$ was assumed given and $P_f(\omega_1, \omega_2)$ was estimated by averaging $|F(\omega_1, \omega_2)|^2$ for ten different images. For white noise degradation, $P_v(\omega_1, \omega_2)$ is constant independent of (ω_1, ω_2). The processed image has an SNR improvement of 7.4 dB. As Figure 9.5 shows, Wiener filtering clearly reduces the background noise. This is also evidenced by the SNR improvement. However, it also blurs the image significantly. Many variations of Wiener filtering have been proposed to improve its performance. Some of these variations will be discussed in the next section.



(a)



(b)

(c)

Figure 9.5 (a) Original image of 512 × 512 pixels; (b) degraded image at SNR of 7 dB, with NMSE of 19.7%; (c) processed image by Wiener filtering, with NMSE of 3.6% and SNR improvement of 7.4 dB.

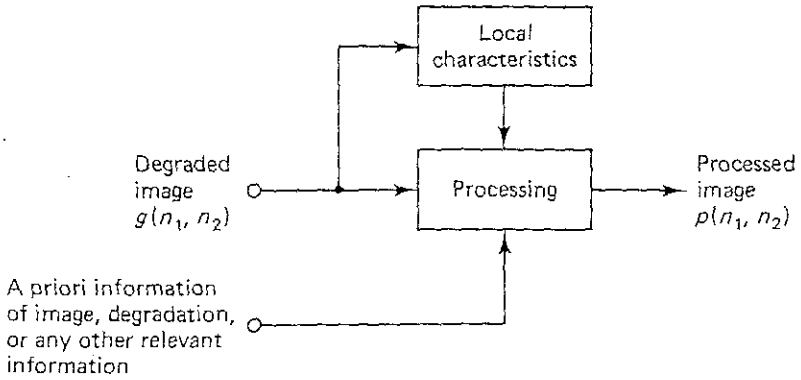


Figure 9.8 General adaptive image processing system.

as the lowpass filter cutoff frequency. Without a specific application context, only general statements can be made. In general, the more the available knowledge is used, the higher the resulting performance will be. If the available information is inaccurate, however, the system's performance may be degraded. In general, more sophisticated rules for adaptation are associated with subimage-by-subimage processing, while simple rules are associated with pixel-by-pixel processing for computational reasons.

When an adaptive image processing method is applied to the problem of restoring an image degraded by additive random noise, it is possible to reduce background noise without significant image blurring. In the next four sections, we discuss a few representative adaptive image restoration systems chosen from among the many proposed in the literature.

9.2.4 The Adaptive Wiener Filter

Most adaptive restoration algorithms for reducing additive noise in an image can be represented by the system in Figure 9.9. From the degraded image and prior knowledge, some measure of the local details of the noise-free image is determined. One such measure is the local variance. A space-variant* filter $h(n_1, n_2)$ which is a function of the local image details and of additional prior knowledge is then determined.

The space-variant filter is then applied to the degraded image in the local region from which the space-variant filter was designed. When the noise is wideband, the space-variant $h(n_1, n_2)$ is lowpass in character. In low-detail image regions such as uniform intensity regions, where noise is more visible than in high-detail regions, a large amount (low cutoff frequency) of lowpass filtering is performed to reduce as much noise as possible. Since little signal variation is present in low-detail regions, even a large amount of lowpass filtering does not significantly affect the signal component. In high-detail image regions such as edges, where a large signal component is present, only a small amount of lowpass filtering is

*For a space-variant filter, the filter coefficients change as a function of (n_1, n_2). For notational simplicity, we denote the filter coefficients by $h(n_1, n_2)$. It should be noted, however, that $h(n_1, n_2)$ changes as we process different parts of an image.

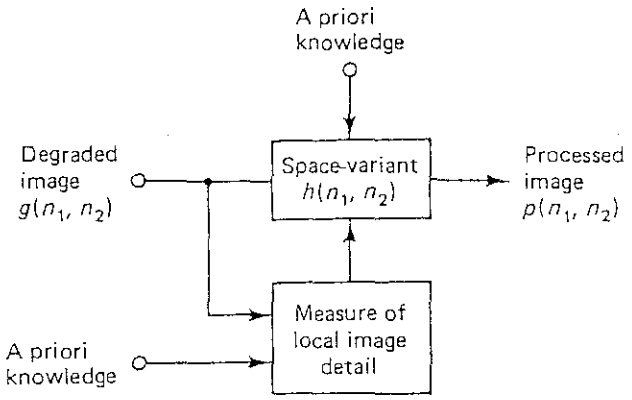


Figure 9.9 Typical adaptive image restoration system for additive noise reduction.

performed so as not to distort (blur) the signal component. This does not reduce much noise, but the same noise is less visible in the high-detail than in the low-detail regions.

A number of different algorithms can be developed, depending on which specific measure is used to represent local image details, how the space-variant $h(n_1, n_2)$ is determined as a function of the local image details, and what prior knowledge is available. One of the many possibilities is to adaptively design and implement the Wiener filter discussed in Section 9.2.1. As Figure 9.3 shows, the Wiener filter requires knowledge of the signal mean m_f, noise mean m_v, signal power spectrum $P_f(\omega_1, \omega_2)$, and noise power spectrum $P_v(\omega_1, \omega_2)$. Instead of assuming a fixed m_f, m_v, $P_f(\omega_1, \omega_2)$, and $P_v(\omega_1, \omega_2)$ for the entire image, they can be estimated locally. This approach will result in a space-variant Wiener filter. Even within this approach, many variations are possible, depending on how m_f, m_v, $P_f(\omega_1, \omega_2)$, and $P_v(\omega_1, \omega_2)$ are estimated locally and how the resulting space-variant Wiener filter is implemented. We will develop one specific algorithm to illustrate this approach.

We first assume that the additive noise $v(n_1, n_2)$ is zero mean and white with variance of σ_v^2. Its power spectrum $P_v(\omega_1, \omega_2)$ is then given by

$$P_v(\omega_1, \omega_2) = \sigma_v^2. \tag{9.21}$$

Consider a small local region in which the signal $f(n_1, n_2)$ is assumed stationary. Within the local region, the signal $f(n_1, n_2)$ is modeled by

$$f(n_1, n_2) = m_f + \sigma_f w(n_1, n_2) \tag{9.22}$$

where m_f and σ_f are the local mean and standard deviation of $f(n_1, n_2)$, and $w(n_1, n_2)$ is zero-mean white noise with unit variance.* There is some empirical evidence that (9.22) is a reasonable model for a typical image [Trussell and Hunt; Kuan et al. (1985)].

In (9.22), the signal $f(n_1, n_2)$ is modeled by a sum of a space-variant local

*The notation $w(n_1, n_2)$ is used to represent both a window function and white noise. Specifically which is meant will be clear from the context.

Bt Bayes's theorem

$$P(x|y) = \frac{P(y|x)P(x)}{\int P(y|x)P(x)dx}$$

$P(y|x)$ is the conditional probability of y given x.
Thus, $P(x)$ is the object distribution $f(x)$
$P(y|x)$ is the PSF cenetered at x i.e. $g(x, y)$
$P(y)$ is the degraded image or equivalentlu the convolution kernel $c(y)$
So we construct an iterative method

$$f_{i+1}(x) = \left(\int \frac{g(y,x)c(y)dy}{\int g(y,z)f_i(z)dz} \right) f_i(x)$$

or equivalently

$$f_{i+1}(x) = \left\{ \left[\frac{c(x)}{f_i(x) \otimes g(x)} \right] \otimes g(-x) \right\} f_i(x)$$

In this case $g(x)$ is known (non-blind convolution) and one iterates until $f_i(x)$ is known.

For **blind** convolution we replace this by two steps
1, Given the object n Lucy-Richardson iterations are done to find the PSF $g^{(k)}(x)$
2. The next $f(x)$ is found by using n Lucy-Richardson iterations

$$\begin{aligned} g^k_{i+1} &= \left\{ \left[\frac{c(x)}{g^k_i(x) \otimes f^{k-1}(x)} \right] \otimes f^{k-1}(-x) \right\} g^k_i \\ f^k_{i+1}(x) &= \left\{ \left[\frac{c(x)}{f^k_i(x) \otimes g^k(x)} \right] \otimes g^k(-x) \right\} f^k_i(x) \end{aligned}$$

$$g = H * f + n$$

Wiener Deconvolution – minimum least square error estimation

$$G(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{P_n(u,v)}{P_f(u,v)}}$$

Problems:

1. MSE is not physically relevant
   Human eye more tolerant of errors in dark areas and high gradient areas

2. Cannot handle spatially variant blurring PSF
3. Cannot handle nonstationary signals and noise
4. Depends on P_f, P_n Which are not usually known

Power Spectrum Equalization - Homomorphic filter

$$G(u,v) = \left(\frac{1}{|H(u,v)|^2 + \frac{P_n(u,v)}{P_f(u,v)}} \right)^{\frac{1}{2}}$$

Geometric Mean Filter

$$G(u,v) = \left[\frac{H^*(u,v)}{H(u,v)} \right]^{\alpha} \left(\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma \frac{P_n(u,v)}{P_f(u,v)}} \right)^{1-\alpha}$$

If $\alpha = 0$ Called the parametric Wiener filter

Constrained Least Square Restoration

$$\| g - Hf \|^2 = n$$

So consider Lagrange multiplier formulation

$$W(\hat{f}) = \| Q\hat{f} \|^2 + \lambda \left(\| g - H\hat{f} \|^2 - \| n \|^2 \right)$$

Set $\frac{\partial w}{\partial \hat{f}} = 0$ Then $\hat{f} = \frac{H^* g}{|H|^2 + \gamma |Q|^2}$ $\gamma = 1/\lambda$

Example: Q=Laplacian $Q(u,v) = -4\pi(u^2 + v^2)$

$$g = H * f + n$$

Wiener Deconvolution – minimum least square error estimation

fr = deconvwnr(g, PSF, NACORR, FACORR)

NACORR, FACORR are the autocorrelation functions of the noise and the undegraded picture (in picture space as matrix or scalar). This is the inverse FFT of the power spectrum.

Constrained Least Square Restoration

$$\hat{f} = \frac{H^* g}{|H|^2 + \gamma |Q|^2} \qquad \gamma = 1/\lambda$$

Example: Q=Laplacian $Q(u,v) = -4\pi(u^2 + v^2)$

Fr=deconvreg(g, PSF, NOISEPOWER, RANGE)

Lucy Richardson Restoration

Fr=deconvlucy(g, PSF, NUMIT, DAMPAR, WEIGHT)

Blind Deconcolution

Fr=deconvblind(g, INITPSF, NUMIT, DAMPAR, WEIGHT)

(a) (b)

Figure 4.29: *Restoration of motion blur using Wiener filtration. Courtesy P. Kohout, Criminalistic Institute, Prague.*

4.5 Summary

- **Image pre-processing**
  - Operations with images at the lowest level of abstraction—both input and output are intensity images—are called *pre-processing*.
  - The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing.
  - Four basic types of pre-processing methods exist:
    * Brightness transformations
    * Geometric transformations
    * Local neighborhood pre-processing
    * Image restoration
- **Pixel brightness transformations**
  - There are two classes of pixel brightness transformations:
    * *Brightness corrections*
    * *Gray-scale transformations*
  - Brightness corrections modify pixel brightness taking into account its original brightness and its position in the image.
  - Gray-scale transformations change brightness without regard to position in the image.
  - Frequently used brightness transformations include:

(a) (b)

Figure 4.30: *Restoration of wrong focus blur using Wiener filtration. Courtesy P. Kohout, Criminalistic Institute, Prague.*

  - * Brightness thresholding
  - * Histogram equalization
  - * Logarithmic gray-scale transforms
  - * Look-up table transforms
  - * Pseudo-color transforms
- The goal of histogram equalization is to create an image with equally distributed brightness levels over the whole brightness scale.

- **Geometric transformations**
  - Geometric transforms permit the elimination of the geometric distortions that occur when an image is captured.
  - A geometric transform typically consists of two basic steps:
    - * *Pixel co-ordinate transformation*
    - * *Brightness interpolation*
  - Pixel co-ordinate transformations map the co-ordinates of the input image pixel to a point in the output image; *affine* and *bilinear* transforms are frequently used.
  - The output point co-ordinates do not usually match the digital grid after the transform and interpolation is employed to determine brightnesses of output pixels; *nearest-neighbor*, *linear*, and *bi-cubic* interpolations are frequently used.

- **Local pre-processing**
  - Local pre-processing methods use a small neighborhood of a pixel in an input image to produce a new brightness value in the output image.
  - For the pre-processing goal, two groups are common: *smoothing* and *edge detection.*