Chapter 5  Image Restoration and Reconstruction

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 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.

The restoration approach usually involves formulating a criterion of goodness that will yield an optimal estimate of the desired result, while enhancement techniques are heuristic procedures to manipulate an image in order to take advantage of the human visual system.

Some restoration techniques are best formulated in the spatial domain, while others are better suited for the frequency domain.
5.1 A Model of the Image Degradation/Restoration Process

Figure 5.1 shows an image degradation/restoration process.

The degraded image in the spatial domain is given by

\[ g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \]  \hspace{1cm} (5.1-1)

where \( h(x, y) \) is the spatial representation of the degradation function and “\( \star \)” indicates convolution. Therefore, we can have the frequency domain representation of (5.1-1)

\[ G(u, v) = H(u, v)F(u, v) + N(u, v). \] \hspace{1cm} (5.1-2)

These two equations are the bases for most of the restoration material in Chapter 5.
5.2 Noise Models

The principal sources of noise in digital images arise during image acquisition and/or transmission.

Spatial and Frequency Properties of Noise

In the spatial domain, we are interested in the 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.

In general, we assume that noise is independent of spatial coordinates and it is uncorrelated with respect to the image itself.

Some Important Noise Probability Density Functions

Gaussian noise

Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (normal) noise models are used frequently in practice.

The probability density function (PDF) of a Gaussian random variable, \( z \), is given by

\[
p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}} \quad (5.2-1)
\]

where \( z \) represents intensity, \( \bar{z} \) is the mean (average) value of \( z \), and \( \sigma \) is its standard deviation.
Rayleigh noise

The probability density function of Rayleigh noise is given by

\[
p(z) = \begin{cases} 
  \frac{2}{b} (z - a) e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\
  0 & \text{for } z < a 
\end{cases}
\]  

(5.2-2)

The mean and variance of this density are given by

\[
\bar{z} = a + \frac{\sqrt{\pi b}}{4}
\]  

(5.2-3)

and

\[
\sigma^2 = \frac{b(4 - \pi)}{4}.
\]  

(5.2-4)
Erlang (gamma) noise

The probability density function of Erlang noise is given by

\[ p(z) = \begin{cases} \frac{a^b z^{b-1} e^{-az}}{(b-1)!} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \quad (5.2-5) \]

where \( a > 0 \) and \( b \) is a positive integer. The mean and variance of this density are given by

\[ \bar{z} = \frac{b}{a} \quad (5.2-6) \]

and

\[ \sigma^2 = \frac{b}{a^2}. \quad (5.2-7) \]

Exponential noise

The PDF of exponential noise is given by

\[ p(z) = \begin{cases} ae^{-az} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \quad (5.2-8) \]

where \( a > 0 \). The mean and variance of this density are given by

\[ \bar{z} = \frac{1}{a} \quad (5.2-9) \]

and

\[ \sigma^2 = \frac{1}{a^2}. \quad (5.2-10) \]
Uniform noise

The PDF of uniform noise is given by

\[
p(z) = \begin{cases} 
\frac{1}{b-a} & \text{for } a \leq z \leq b \\
0 & \text{otherwise}
\end{cases} \quad (5.2-11)
\]

The mean and variance of this density function are given by

\[
\bar{z} = \frac{a + b}{2} \quad (5.2-12)
\]

and

\[
\sigma^2 = \frac{(b - a)^2}{12}. \quad (5.2-13)
\]

Impulse (salt-and-pepper) noise

The PDF of impulse noise is given by

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

If \( b > a \), intensity \( b \) appears as a light dot in the image. Conversely, intensity \( a \) will appear like a dark dot.

If either \( P_a \) or \( P_b \) is zero, the impulse noise is called unipolar.

If neither \( P_a \) nor \( P_b \) is zero, and especially if they are approximately equal, the impulse noise values will resemble salt-and-pepper granules randomly distributed over the image.
Example 5.1: Noisy images and their histograms

Figure 5.3 shows a test pattern.

Figure 5.4 shows the test pattern after addition of the six types of noise. Shown below each image is the histogram computed directly from that image.
FIGURE 5.4 (Continued) Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.
Periodic Noise

Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition.

The periodic noise can be reduced significantly via frequency domain filtering, which will be discussed in Section 5.4.
Estimation of Noise Parameters

The parameters of periodic noise can be estimated by inspection of the Fourier spectrum of the image.

Periodic noise tends to produce frequency spikes, which are detectable even by visual analysis.

In simplistic cases, it is also possible to infer the periodicity of noise components directly from the image.

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

It is often necessary to estimate the noise probability density functions for a particular imaging arrangement.

When images already generated by a sensor are available, it may be possible to estimate the parameters of the probability density functions from small patches of reasonably constant background intensity.

\[ \text{Figure 5.6} \quad \text{Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.} \]

The vertical stripes shown in Figure 5.6 were cropped from (a), (b), and (h) of Figure 5.4.
The histograms shown in Figure 5.6 were calculated using image data from these small stripes. We can see that the shapes of these histograms correspond closely to the shapes shown in (d), (e), and (k) of Figure 5.4.

The simplest use of the data from the image strips is for calculating the mean and variance of intensity levels. Let $S$ denote a stripe and $p_S(z_i), i = 0, 1, 2, ..., L - 1$, denote the probability estimates of the intensities of the pixels in $S$, then the mean and variance of the pixels in $S$ are

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

and

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

The shape of the histogram identifies the closest probability density function match.

The Gaussian probability density function is completely specified by these two parameters.

For the other shapes discussed previously, we can use the mean and variance to solve the parameters $a$ and $b$.

Impulse noise is handled differently because the estimate needed is of the actual probability of occurrence of the white and black pixels.
5.3 Restoration in the Presence of Noise Only – Spatial Filtering

When the only degradation present in an image is noise,

\[ g(x, y) = h(x, y) \ast f(x, y) + \eta(x, y) \]  \hspace{1cm} (5.1-1)

and

\[ G(u, v) = H(u, v)F(u, v) + N(u, v) \]  \hspace{1cm} (5.1-2)

become

\[ g(x, y) = f(x, y) + \eta(x, y) \]  \hspace{1cm} (5.3-1)

and

\[ G(u, v) = F(u, v) + N(u, v) \]  \hspace{1cm} (5.3-2)

Since the noise terms are unknown, 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 \( G(u, v) \).

Mean Filters

Arithmetic mean filter

Let \( S_{xy} \) represent the set of coordinates in a subimage window of size \( m \times n \), centered at \( (x, y) \). The arithmetic mean filter computes the average value of the corrupted image \( g(x, y) \) in \( S_{xy} \).

The value of the restored image \( \hat{f} \) at point \( (x, y) \) is the arithmetic mean computed in the region \( S_{xy} \):

\[ \hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t). \]  \hspace{1cm} (5.3-3)
Geometric mean filter

Using a geometric mean filter, an image is restored by

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

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 filter is given by the expression

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

which works well for some types of noise like Gaussian noise and salt noise, but fails for pepper noise.

Contraharmonic mean filter

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

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

where \( Q \) is called the order of the filter.

The contraharmonic mean filter is well suited for reducing or eliminating the effects of salt-and-pepper noise.
For positive values of $Q$, it eliminates pepper noise.

For negative values of $Q$, it eliminates salt noise.

When $Q = 0$, the contraharmonic mean filter reduces to the arithmetic mean filter.

When $Q = -1$, the contraharmonic mean filter becomes the harmonic mean filter.

Example 5.2: Illustration of mean filters

Figure 5.7 (a) shows an 8-bit image, and Figure 5.7 (b) shows its corrupted version with additive Gaussian noise of zero mean and variance of 400.

Figure 5.7 (c) and Figure 5.7 (d) show the result of filtering the noisy image with a $3 \times 3$ arithmetic mean filter and a $3 \times 3$ geometric mean filter, respectively.
Figure 5.8 (a) and Figure 5.8 (b) show the images corrupted by 10% pepper noise and 10% salt noise, respectively.

Figure 5.8 (c) shows the result of filtering Figure 5.8 (a) using a contraharmonic mean filter with $Q = 1.5$.

Figure 5.8 (d) shows the result of filtering Figure 5.8 (b) using a contraharmonic mean filter with $Q = -1.5$.

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.

In general, the arithmetic and geometric mean filters are suited for random noise like Gaussian or uniform noise.
The contraharmonic mean filter is well suited for impulse noise, with the disadvantage that it must known whether the noise is dark or light in order to select \( Q \).

Figure 5.9 shows some results of choosing the wrong sign for \( Q \).
Order-Statistic Filters

As discussed in Chapter 3, 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.

**Median filter**

The best-known order-statistic filter is the median filter, which will replace the value of a pixel by the median of the intensity levels in the neighbourhood of that pixel:

\[
\hat{f}(x, y) = \text{median} \{ g(s, t) \}. \quad (5.3-7)
\]

For certain types of random noise, the median filters can provide excellent noise-reduction capabilities.

The median filters are particularly effective in the presence of both bipolar and unipolar impulse noise.

**Max and min filters**

The max and min filters are defined as

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

and

\[
\hat{f}(x, y) = \min_{(s, t) \in S_{xy}} \{ g(s, t) \}. \quad (5.3-9)
\]

The max filter is useful for finding the brightest points in an image, while the min filter can be used for finding the darkest points in an image.
Midpoint filter

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

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

The midpoint filter works best for random 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 \( S_{xy} \). Let \( g_r(s, t) \) represent the remaining \( mn - d \) pixels, an alpha-trimmed mean filter is given by

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

When \( d = 0 \), the alpha-trimmed mean filter is reduced to the arithmetic mean filter.

If \( d = mn - 1 \), the alpha-trimmed mean filter becomes a median filter.
Example 5.3: Illustration of order-statistic filters

Figure 5.10 (a) shows the 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 $3 \times 3$.

Figure 5.10 (c) and Figure 5.10 (d) show the result of applying the same filter on Figure 5.10 (b) and Figure 5.10 (c), respectively.

Figure 5.11 (a) shows the result of applying the max filter to the pepper noise image of Figure 5.8 (a).

Figure 5.11 (b) shows the result of applying the min filter to the image of Figure 5.8 (b).

The min filter did a better job on noise removal, but it removes some white points around the border of light objects.
The results of applying the alpha-trimmed filter are shown in Figure 5.12.

In Figure 5.12 (e), it should be $d = 6$. 
Adaptive Filters

Adaptive filters are capable of performance superior to that of the filters discussed thus far. However, the price paid for improved filtering power is an increase in filter complexity.

Adaptive, local noise reduction filter

The simplest statistical measures of a random variable are its mean and variance, which are reasonable parameters for an adaptive filter.

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.

The response of a filter, which operates on a local region $S_{xy}$, at any point $(x, y)$ is to be based on four quantities:

(a) $g(x, y)$, the value of the noisy image at $(x, y)$;
(b) $\sigma_{\eta}^2$, the variance of the noise corrupting $f(x, y)$ to form $g(x, y)$;
(c) $m_L$, the local mean of the pixels in $S_{xy}$;
(d) $\sigma_L^2$, the local variance of the pixels in $S_{xy}$.

We want to have the following behaviours for the filter:

1. If $\sigma_{\eta}^2$ is zero, the filter should just return the value of $g(x, y)$.
   This is the zero-noise case in which $g(x, y)$ is equal to $f(x, y)$.
2. If the local variance, $\sigma_L^2$, is high relative to $\sigma_{\eta}^2$, the filter should return a value close to $g(x, y)$. 
A high local variance typically is associated with edges, which 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.

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

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_n^2}{\sigma_L^2} [g(x,y) - m_L].$$

(5.3-12)

The only quantity needed to be estimated is the variance of the overall noise, $\sigma_n^2$, and other parameters can be computed from the pixels in $S_{xy}$.

A tacit assumption in (5.3-12) is $\sigma_n^2 \leq \sigma_L^2$, which is reasonable because $S_{xy}$ is a subset of $g(x,y)$. In practice, however, it is possible for this condition to be violated. So, a test should be performed in implementation so that the ratio is set to 1 if $\sigma_n^2 > \sigma_L^2$ occurs.
Example 5.4: Illustration of adaptive, local noise-reduction filtering

Figure 5.13 (a) shows an image corrupted by additive Gaussian noise of zero mean and a variance of 1000.

Figure 5.13 (b) is the result of applying an arithmetic mean filter of size $7 \times 7$ to Figure 5.13 (a).

Figure 5.13 (c) shows the result of applying a geometric mean filter of size $7 \times 7$ to Figure 5.13 (a).

Figure 5.13 (d) shows the result of using the adaptive filter

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma^2}{\sigma^2_L} [g(x,y) - m_L] (5.3-12)$$

with $\sigma^2 = 1000$. 

---

**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 $7 \times 7$. 

---
Adaptive median filter

The median filter discussed previously performs well if the spatial density of the impulse noise is not large ($P_a$ and $P_b$ are less than 0.2).

The adaptive median filtering can handle impulse noise with probabilities larger than these.

Unlike other filters, the adaptive median filter changes the size of $S_{xy}$ during operation, depending on certain conditions.

Consider the following notations:

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

The adaptive median filtering algorithm works in two stages:

**Stage A:**

A1 = $z_{\text{med}} - z_{\text{min}}$

A2 = $z_{\text{med}} - z_{\text{max}}$

If A1 > 0 AND A2 < 0, go to stage B

Else increase the window size

If window size $\leq S_{\text{max}}$ repeat stage A

Else output $z_{\text{med}}$

**Stage B:**

B1 = $z_{xy} - z_{\text{min}}$

B2 = $z_{xy} - z_{\text{max}}$

If B1 > 0 AND B2 < 0, output $z_{xy}$

Else output $z_{\text{med}}$
Keep in mind that this algorithm 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 the distortion of object boundaries.

**Example 5.5: Illustration of adaptive median filtering**

Figure 5.14 (a) shows an image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$.

Figure 5.14 (b) is the result of applying a $7 \times 7$ median filter. Although the noise was effectively removed, the filter caused significant loss of the detail in the image.

Figure 5.14 (c) shows the result of using the adaptive median filter with $S_{\text{max}} = 7$. It can be observed that with the similar noise removal performance, the adaptive median filter did a better job of preserving sharpness and detail.
5.4 Periodic Noise Reduction by Frequency Domain Filtering

Periodic noise can be analyzed and filtered effectively by using frequency domain techniques.

Bandreject Filters

Figure 5.15 shows perspective plots of ideal, Butterworth, and Gaussian bandreject filters,

![Figure 5.15](image)

which were discussed in Section 4.10.1 and are summarized in Table 4.6.

**TABLE 4.6**

Bandreject filters. $W$ is the width of the band, $D$ is the distance $D(u,v)$ from the center of the filter, $D_0$ is the cutoff frequency, and $n$ is the order of the Butterworth filter. We show $D$ instead of $D(u,v)$ to simplify the notation in the table.

<table>
<thead>
<tr>
<th>Ideal</th>
<th>Butterworth</th>
<th>Gaussian</th>
</tr>
</thead>
</table>
| $H(u,v) = \begin{cases} 
0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\
1 & \text{otherwise}
\end{cases}$ | $H(u,v) = \frac{1}{1 + \left[ \frac{DW}{D^2 - D_0^2} \right]^{2n}}$ | $H(u,v) = 1 - e^{-\frac{D^2 + D_0^2}{2W^2}}$ |

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.
Example 5.6: Use of Bandreject filtering for periodic noise removal

Figure 5.16 (a), which is the same as Figure 5.5 (a), shows an image corrupted by sinusoidal noise of various frequencies.

The noise components can be seen as symmetric pairs of bright dots in the Fourier spectrum shown in Figure 5.16 (b).

Since the component lie on an approximate circle about the origin of the transform, so a circularly symmetric bandreject filter is a good choice.

![Figure 5.16](image)

**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 5.16 (c) shows a Butterworth bandreject filter of order 4.

Figure 5.16 (d) shows the result of filtering Figure 5.16 (a) with the filter shown in Figure 5.16 (c).
Bandpass Filters

A bandpass filter performs the opposite operation of a bandreject filter.

The transfer function $H_{BP}(u,v)$ of a bandpass filter is obtained from a corresponding bandreject filter transfer function $H_{BR}(u,v)$ by using the equation

$$H_{BP}(u,v) = 1 - H_{BR}(u,v).$$ (5.4-1)

Performing straight bandpass filtering on an image is not a common procedure because it generally removes too much image detail. However, bandpass filtering is useful in isolating the effects on an image caused by selected frequency bands.

**Example 5.7: Bandpass filtering for extracting noise patterns**

The image shown in Figure 5.17 was generated by

1. using (5.4-1) to obtain the bandpass filter corresponding to the bandreject filter used in Figure 5.16;
2. taking the inverse transform of the bandpass-filtered transform.

![FIGURE 5.17](image.png)

Although most image detail was lost, the remained information shows the noise pattern, which is quite close to the noise that corrupted the image in Figure 5.16.
Notch Filters

A notch filter rejects/passes frequencies in predefined neighbourhoods about a center frequency. Figure 5.18 shows plots of ideal, Butterworth, and Gaussian notch (reject) filters.

![Figure 5.18](image)

Similar to (5.4-1), the transfer function $H_{NP}(u, v)$ of a notch pass filter is obtained from a corresponding notch reject filter transfer function, $H_{NR}(u, v)$, by using the equation

$$H_{NP}(u, v) = 1 - H_{NR}(u, v).$$  \hspace{1cm} (5.4-2)
Example 5.8: Removal of periodic noise by notch filtering

Figure 5.19 (a) shows the same image as Figure 4.51 (a).

Figure 5.19 (b) shows the spectrum of Figure 5.19 (a), in which the noise is not domain enough to have a clear pattern along the vertical axis.

Figure 5.19 (c) shows the notch pass filter applied on Figure 5.19 (b).

Figure 5.19 (d) shows the spatial representation of the noise pattern (inverse transform of the notch-pass-filtered result).

Figure 5.19 (e) shows the result of processing the image with the Figure 5.19 (d) shows.
Optimum Notch Filtering

Figure 5.20 shows another example of periodic image degradation.

When several interference components are present, the methods discussed previously are not always acceptable because they may remove too much image information in the filtering process.

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.

The first step can be done by placing a notch pass filter, $H_{NP}(u,v)$, at the location of each spike.

The Fourier transform of the interference noise pattern is given by the expression

$$N(u,v) = H_{NP}(u,v)G(u,v),$$

(5.4-3)

where $G(u,v)$ donates the Fourier transform of the corrupted image.
Since the formation of $H_{NP}(u, v)$ requires judgment about what is or is not an interference spike, 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

$$
\eta(x, y) = \mathcal{F}^{-1} \{ H_{NP}(u, v)G(u, v) \}.
$$

(5.4-4)

Since 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, to obtain $f(x, y)$ would be a simple matter

$$
f(x, y) = g(x, y) - \eta(x, y).
$$

However, the 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 by subtracting from $g(x, y)$ a weighted portion of $\eta(x, y)$ to obtain an estimate of $f(x, y)$:

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

(5.4-5)

where the function $w(x, y)$ is called a weighted or modulation function.

One approach is to select $w(x, y)$ so that the variance of the estimate $\hat{f}(x, y)$ is minimized over a specified neighbourhood of every point $(x, y)$. 
Consider a neighbourhood of size \((2a + 1) \times (2b + 1)\) about a point \((x, y)\). The local variance of \(\hat{f}(x, y)\) at \((x, y)\) can be estimated from samples, as

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

where \(\bar{f}(x, y)\) is the average value of \(\hat{f}(x, y)\) in the neighbourhood:

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

Then, we have

\[
\sigma^2(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left\{ \left[ g(x + s, y + t) - w(x + s, y + t) \eta(x + s, y + t) \right] \right. \\
- \left[ g(x, y) - w(x, y) \eta(x, y) \right] \right\}^2 \tag{5.4-8}
\]

Assuming that \(w(x, y)\) essentially remains constant over the neighbourhood,

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

for \(-a \leq s \leq a\) and \(-b \leq t \leq b\). It also leads to

\[
\bar{w}(x, y) \eta(x, y) = w(x, y) \eta(x, y) \tag{5.4-10}
\]

in the neighbourhood. Then, (5.4-8) becomes

\[
\sigma^2(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left\{ \left[ g(x + s, y + t) - w(x, y) \eta(x, y) \right] \right. \\
- \left[ g(x, y) - w(x, y) \eta(x, y) \right] \right\}^2 \tag{5.4-11}
\]
To minimize $\sigma^2(x, y)$, we solve

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

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

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

Since $w(x, y)$ is assumed to be constant in a neighbourhood, it is computed for one point in each neighbourhood and then used to process all of the image points in that neighbourhood.

**Example 5.9: Illustration of optimum notch filtering.**

Figure 5.21 through Figure 5.23 show the result of applying the preceding techniques to the image in Figure 5.20 (a).

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 case, so $u = v = 0$ is at the top left corner in Figure 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 Figure 5.22 (b) and the structure of the noise present in Figure 5.20 (a).

Figure 5.23 shows the processed image obtained by using

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

In Figure 5.23, the periodic interference has been removed.
5.5 Linear, Position-Invariant Degradations

The input-output relationship in Figure 5.1 before the restoration can be expressed as

\[ g(x, y) = H[f(x, y)] + \eta(x, y). \]  (5.5-1)

First, we assume that \( \eta(x, y) = 0 \) so that \( g(x, y) = H[f(x, y)] \). \( H \) is linear if

\[ H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)], \]  (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 \( a = b = 1 \), then (5.5-2) becomes

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

which is called the property of additivity.

If \( f_2(x, y) = 0 \), (5.5-2) becomes

\[ H[af_1(x, y)] = aH[f_1(x, y)], \]  (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.

An operator having the input-output relationship

\[ g(x, y) = H[f(x, y)] \]

is said to be position (space) invariant if

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

for any \( f(x, y) \) and any \( \alpha \) and \( \beta \). (5.5-5) 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.
With a slight change in notation in the definition of the impulse in

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) \delta(t - t_0, z - z_0) dt \, dz = f(t_0, z_0),
\] (4.5-3)

\(f(x, y)\) can be expressed as

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

Assuming \(\eta(x, y) = 0\), then substituting (5.5-6) into (5.5-1) we have

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

If \(H\) is a linear operator, then

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

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

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

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

where the term

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

is called the impulse response of \(H\).

In other words, if \(\eta(x, y) = 0\), then \(h(x, \alpha, y, \beta)\) is the response of \(H\) to an impulse at \((x, y)\).
Equation
\[ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta) \, d\alpha \, d\beta \]  
(5.5-11)

is called the superposition (or Fredholm) integral of the first kind, and is a fundamental result at the core of linear system theory.

If \( H \) is position invariant, from
\[ H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta), \]
(5.5-5)
we have
\[ H[\delta(x - \alpha, y - \beta)] = h(x - \alpha, y - \beta), \]
(5.5-12)

and (5.5-11) reduces to
\[ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta) \, d\alpha \, d\beta. \]
(5.5-13)

The expression (5.5-13) is the case of convolution integral
\[ f(t) \ast h(t) = \int_{-\infty}^{\infty} f(\tau)h(t - \tau) \, d\tau \]
(4.2-20)

being extended to 2-D.

Equation (5.5-13) tells us that knowing the impulse 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, (5.5-11) becomes
\[ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta) \, d\alpha \, d\beta + \eta(x, y). \]
(5.5-14)
If \( H \) is position invariant, it becomes

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

Assuming that the values of the random noise \( \eta(x, y) \) are independent of position, we have

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

Based on the convolution theorem, we can express (5.5-16) in the frequency domain as

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

In summary, a linear, spatially invariant degradation system with additive noise can be modeled in the spatial domain as the convolution of the degradation function with an image, followed by the additive of noise (as expressed in (5.5-16)).

The same process can be expressed in the frequency domain as stated in (5.5-17).
5.6 Estimating the Degradation Function

There are three principal ways to estimate the degradation function used in image restoration:

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.

In order to reduce the effect of noise, we would look for an area in which the signal content is strong.

Let the observed subimage be denoted by $g_s(x, y)$, and the processed subimage be denoted by $\hat{f}_s(x, y)$. Assuming that the effect of noise is negligible, it follows from

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$  \hspace{1cm} (5.5-17)

that

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

Then, we can have $H(u, v)$ based on our assumption of position invariant.

For example, suppose that a radial plot of $H_s(u, v)$ has the approximate shape of a Gaussian curve. Then we can construct a function $H(u, v)$ on a large scale, but having the same basic shape.

This estimation is a laborious process used in very specific circumstances.
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 on the degradation by imaging an impulse (small dot of light) using the same system settings.

An impulse is simulated by a bright dot of light, as bright as possible to reduce the effect of noise to negligible values. Since the Fourier transform of an impulse is a constant, it follows

\[ H(u, v) = \frac{G(u, v)}{A}. \]  \hspace{1cm} (5.6-2)

Figure 5.24 shows an example.
Estimation by Modeling

Degradation modeling has been used for years.

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 is based on the physical characteristics of atmospheric turbulence

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

where \( k \) is a constant that depends on the nature of the turbulence.

Figure 5.25 shows examples of using (5.6-3) with different values of \( k \).
A major approach in modeling is to derive a mathematical model starting from basic principles.

We show this procedure by a 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.

The total exposure at any point of the recording medium is obtained by integrating the instantaneous exposure over the time interval when the imaging system shutter is open.

If the \( T \) is the duration of the exposure, the blurred image \( g(x, y) \) is

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

From

\[
F(\mu, \nu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) e^{-j2\pi(\mu t + \nu z)} dt dz , \tag{4.5-7}
\]

the Fourier transform of (5.6-4) is

\[
G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(ux + vy)} dxdy \tag{5.6-5}
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \int_0^T f[x - x_0(t), y - y_0(t)] dt \right] e^{-j2\pi(ux + vy)} dxdy
\]

By reversing the order of integration,

\[
G(u, v) = \int_0^T \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)] e^{-j2\pi(ux + vy)} dxdy \right] dt
\]
Since the term inside the outer brackets is the Fourier transform of the displaced function \( f[x - x_0(t), y - y_0(t)] \), we have

\[
G(u, v) = \int_0^T F(u, v) e^{-j2\pi[ux_0(t) + vy_0(t)]} dt
\]

\[
= F(u, v) \int_0^T e^{-j2\pi[ux_0(t) + vy_0(t)]} dt
\]

(5.6-7)

By defining

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

(5.6-8)

we can rewrite (5.6-7) in the familiar form

\[
G(u, v) = H(u, v)F(u, v)
\]

(5.6-9)

Example:

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 \( y_0(t) = 0 \), (5.6-8) yields

\[
H(u, v) = \int_0^T e^{-j2\pi ux_0(t)} dt
\]

\[
= \int_0^T e^{-j2\pi uat / T} dt
\]

\[
= \frac{T}{\pi ua} \sin(\pi ua) e^{-j\pi ua}
\]

(5.6-10)

If we allow the \( y \)-component to vary as well, with the motion given by \( y_0(t) = bt / T \), the degradation function becomes

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

(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 Figure 5.26 (a), multiplying the transform by $H(u,v)$ from (5.6-11).

The parameters used in (5.6-11) were $a = b = 0.1$ and $T = 1$. 
5.7 Inverse Filtering

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 by

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

(5.7-1)

Substituting the right side of

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

(5.1-2)

in (5.7-1) yields

\[
\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}.
\]  

(5.7-2)

The bad news is that we cannot recover the undegraded image exactly because \( N(u, v) \) is not known.

More bad news is that if the degradation function \( H(u, v) \) has zero or very small values, so the second term of (5.7-2) could easily dominate the estimate of \( \hat{F}(u, v) \).

One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin. As discussed earlier, we know that \( H(0,0) \) is usually the highest value of \( H(u, v) \) in the frequency domain.
Example 5.11: Inverse filtering

The image in Figure 5.25 (b) was inverse filtered with

\[ \hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \]  \hspace{1cm} (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 \( k = 0.0025 \). In this case, \( M = N = 480 \).

Although a Gaussian-shape function has no zeros and it is not a concern here, the degradation values become so small that the result of full inverse filtering shown in Figure 5.27 (a) is useless.
Figure 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.

Values above 70 started to produce degraded images, and further increases in radius values would produce images that looked more and more like Figure 5.27 (a).

The results in Example 5.11 show the poor performance of direct inverse filtering in general.
5.8 Minimum Mean Square Error (Wiener) Filtering

Here we discuss an approach that incorporates both the degradation function and statistical characteristics of noise into the restoration process.

Considering images and noise as random variables, the objective is to find an estimate \( \hat{f} \) of the uncorrupted image \( f \) such that the mean square error between them is minimized.

The error measure is given by

\[
e^2 = E \{ (f - \hat{f})^2 \} \quad (5.8-1)
\]

where \( E \{\cdot\} \) is the expected value of the argument.

By assuming that

1. the noise and the image are uncorrelated;
2. one or the other has zero mean;
3. the intensity levels in the estimate are a linear function of the levels in the degraded image.

Then, the minimum of the error function in (5.8-1) is given in the frequency domain by the expression

\[
\hat{F}(u, v) = \left[ \frac{H^*(u, v)S_f(u, v)}{S_f(u, v)|H(u, v)|^2 + S_\eta(u, v)} \right] G(u, v)
\]

\[
= \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v) \quad (5.8-2)
\]

\[
= \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v)
\]
The terms in (5.8-2) are as follows:

\[ \hat{F}(u,v) \] is the frequency domain estimate

\[ G(u,v) \] is the transform of the degraded image

\[ H(u,v) \] is the transform of the degradation function

\[ H^*(u,v) \] is complex conjugate of \( H(u,v) \)

\[ |H(u,v)|^2 = H^*(u,v)H(u,v) \]

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

\[ S_f(u,v) = |F(u,v)|^2 \] = power spectrum of the undegraded image

This result is known as the Wiener filter, which also is commonly referred to as the minimum mean square error filter or the least square error filter.

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 \( v \).

If the noise is zero, then the Wiener filter reduces to the inverse filter.

One of the most important measures is the signal-to-noise ratio, approximated using frequency domain quantities such as

\[
SNR = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F(u,v)|^2}{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |N(u,v)|^2} \tag{5.8-3}
\]
The mean square error given in statistical form in (5.8-1) can be approximated also in terms a summation involving the original and restored images:

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

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

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

The closer \( f \) and \( \hat{f} \) are, the larger this ratio will be.

If we are dealing with white noise, the spectrum \( |N(u, v)|^2 \) is a constant, which simplifies things considerably. However, \( |F(u, v)|^2 \) is usually unknown.

An approach is used frequently when these quantities are not known or cannot be estimated:

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

where \( K \) is a specified constant that is added to all terms of \( |H(u, v)|^2 \).

Note: White noise is a random signal (or process) with a flat power spectral density. In other words, the signal contains equal power within a fixed bandwidth at any center frequency.
Example 5.12: Comparison of inverse and Wiener filtering

Figure 5.28 shows the advantage of Wiener filtering over direct inverse filtering.

![Figure 5.28](image)

**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 5.28 (a)** is the full inverse-filtered result from **Figure 5.27 (a)**.

**Figure 5.28 (b)** is the radially limited inverse result of **Figure 5.27 (c)**.

**Figure 5.28 (c)** shows the result obtained using

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

with the degradation function

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

used in **Example 5.11**. The value of \( K \) was chosen interactively to yield the best visual result.

By comparing **Figure 5.25 (a)** and **Figure 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

![Image showing different filtering results]

**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.

5.9 Constrained Least Squares Filtering (Optional)
5.10 Geometric Mean Filter

It is possible to generalize the Wiener filter slightly to the so-called geometric mean filter:

\[
\hat{F}(u, v) = \left[ \left| H(u, v) \right|^2 \right]^{\alpha} \left[ \frac{H^*(u, v)}{\left| H(u, v) \right|^2 + \beta \left[ \frac{S_n(u, v)}{S_f(u, v)} \right]} \right]^{1-\alpha} G(u, v),
\]

where \( \alpha \) and \( \beta \) are positive real constants.

If \( \alpha = 1 \), this filter reduces to the inverse filter.

If \( \alpha = 0 \), the filter becomes the so-called parametric Wiener filter, which reduces to the standard Wiener filter when \( \beta = 1 \).

If \( \alpha = 1/2 \), this filter becomes a product of the two quantities raised to the same power, which is the definition of the geometric mean. When \( \beta = 1 \), the filter is also commonly referred to as the spectrum equalization filter.

With \( \beta = 1 \), as \( \alpha \) decreases below \( 1/2 \), the filter performance will tend more toward to the inverse filter; as \( \alpha \) increases above \( 1/2 \), the filter will behave more like the Wiener filter.
5.11 Image Reconstruction from Projections

In this section, we will examine the problem of reconstructing an image from a series of projections, with a focus on X-ray computed tomography (CT), which is one of the principal applications of digital image processing in medicine.

Introduction

Consider Figure 5.32 (a), which consists of a single object on a uniform background.

Suppose that we pass a thin, flat beam of X-rays from left to right, and assume that the energy of the beam is absorbed more by the object than by the background. Using a strip of X-ray absorption detectors on the other side will yield the signal, whose amplitude (intensity) is proportional to absorption.

The approach is to project the 1-D signal back across the direction from which the beam came, as Figure 5.32 (b) shows. This approach is called backprojection.
We certainly cannot determine a single object or a multitude of objects along the path of the beam by a single project.

If we rotate the position of the source-detector pair by 90° and repeat the previous procedure, we will get a backprojection image shown in Figure 5.32 (d). Adding this result to Figure 5.32 (b) will result an image illustrated in Figure 5.32 (e).

We should be able to learn more about the shape of the object in question by taking more views in the same manner, as shown in Figure 5.33.

As the number of projections increases, the strength of non-intersecting backprojects deceases relative to the strength of regions in which multiple backprojects intersect.

Figure 5.33 (f) shows the result formed from 32 projections.

The reconstructed image seems to be a reasonably good approximation to the shape of the original object. However, the image is blurred by a “halo” effect, which shows a “star” in Figure 5.33 (e). As the number of views increases, the shape of the “halo” becomes circular, as shown in Figure 5.33 (e).
Blurring in CT reconstruction is an important issue and will be addressed in later discussion.

Since the projections 180° apart are mirror images of each other, we only need to consider angle increments halfway around a circle in order to generate all the projects required for reconstruction.

Example 5.16: Backprojection of a simple planar region containing two objects

**FIGURE 5.34** (a) A region with two objects. (b)–(d) Reconstruction using 1, 2, and 4 backprojections 45° apart. (e) Reconstruction with 32 backprojections 5.625° apart. (f) Reconstruction with 64 backprojections 2.8125° apart.
Principles of Computed Tomography (CT)

The theoretical foundation of CT dates back to Johann Radon, a mathematician from Vienna who derived a method in 1907 for projecting a 2-D object along parallel rays as part of his work on line integrals. The method is referred as the Radon transform now.

Allan M. Cormack, a physicist at Tufts University, partially “rediscovered” these concepts and applied them to CT. Cormack published his initial findings in 1963 and 1964. He provided the mathematical formulae needed for the reconstruction and built a CT prototype to show his ideas.

Working independently, electrical engineer Godfrey N. Hounsfield and his colleagues at EMI in London formulated a similar solution and built the first medical CT machine.

Cormack and Hounsfield shared the 1979 Nobel Prize in Medicine for their contributions to medical tomography.

Figure 5.35 shows the first four generations of CT scanners.
The fifth-generation (G5) CT scanners eliminate all mechanical motion by employing electron beams controlled electromagnetically.

The sixth-generation (G6) CT scanners rotate the source-detector pair continuously through $360^\circ$, while the patient is moved at a constant speed along the axis perpendicular to the scan.

The seventh-generation (G7) CT scanners (also called multislice CT scanners) use parallel banks of detectors to collect volumetric CT data simultaneously.

**Projections and the Radon Transform**

A straight line in Cartesian coordinates can be described either by its slope-intercept form

$$y = ax + b,$$

or, as in Figure 5.36, by its normal representation

$$x \cos \theta + y \sin \theta = \rho. \quad (5.11-1)$$

Figure 5.36 Normal representation of a straight line.
The projection of a parallel-ray beam may be modeled by a set of such lines, as shown in Figure 5.37.

![Figure 5.37 Geometry of a parallel-ray beam.](image)

An arbitrary point in the projection signal is given by the raysum along the line

\[ x \cos \theta_k + y \sin \theta_k = \rho_j. \]

In the case of continuous, the raysum is a line integral, given by

\[ g(\rho_j, \theta_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta_k + y \sin \theta_k - \rho_j) \, dx \, dy \quad (5.11-2) \]

Recall the properties of the impulse, \( \delta \), the right side of (5.11-2) is zero unless the argument of \( \delta \) is zero. It indicates that the integral is computed only along the line \( x \cos \theta_k + y \sin \theta_k = \rho_j \).
If we consider all values of $\rho$ and $\theta$, (5.11-2) generalizes

$$g(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) \, dx \, dy. \quad (5.11-3)$$

The equation (5.11-3) gives the projection of $f(x, y)$ along an arbitrary line in the $xy$-plane, is called the Radon transform.

The Radon transform is the cornerstone of reconstruction from projections, with CT being its principle application in the field of image processing.

In the discrete case, (5.11-3) becomes

$$g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho), \quad (5.11-4)$$

where $x$, $y$, $\rho$, and $\theta$ are now discrete variables.

If we fix $\theta$ and allow $\rho$ to vary, (5.11-4) simply sums the pixels of $f(x, y)$ along the line defined by specified values of these two parameters.

Incrementing through all values of $\rho$ required to span the image (with $\theta$ fixed) yields one projection. Changing $\theta$ and repeating the same procedure will yield another projection.
Example 5.17: Using the Radon transform to obtain the projection of a circular region.

We want to obtain the Radon transform for the projection of the circular object

\[
f(x, y) = \begin{cases} 
A & x^2 + y^2 \leq r^2 \\
0 & \text{otherwise}
\end{cases},
\]

where \( A \) is a constant and \( r \) is the radius of the object. The circular object is shown in Figure 5.38 (a).

Since the object is circularly symmetric, its projections are the same for all angles, so all we need is to obtain the projection for \( \theta = 0^\circ \). From (5.11-3), we get

\[
g(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(x - \rho) dx dy \\
= \int_{-\infty}^{\infty} f(\rho, y) dy
\]

This is a line integral along the line \( L(\rho, 0) \).
Note that \( g(\rho, \theta) = 0 \) when \(|\rho| > r\). When \(|\rho| \leq r\), the integral is evaluated from \( y = -\sqrt{r^2 - \rho^2} \) to \( y = \sqrt{r^2 - \rho^2} \).

Therefore,

\[
g(\rho, \theta) = \int_{-\sqrt{r^2 - \rho^2}}^{\sqrt{r^2 - \rho^2}} f(\rho, y) \, dy = \int_{-\sqrt{r^2 - \rho^2}}^{\sqrt{r^2 - \rho^2}} A \, dy
\]

It yields

\[
g(\rho, \theta) = g(\rho) = \begin{cases} 
2A\sqrt{r^2 - \rho^2} & |\rho| \leq r \\
0 & \text{otherwise}
\end{cases}
\]

Figure 5.38 (b) shows the result.

\( g(\rho, \theta) = g(\rho) \) indicates that \( g \) is independent of \( \theta \) because the object is symmetric about the origin.
When the Radon transform, \( g(\rho, \theta) \), is displayed as an image with \( \rho \) and \( \theta \) as rectilinear coordinates, the result is called a sinogram, similar in concept to displaying the Fourier spectrum. Like the Fourier spectrum, a sinogram contains the data necessary to reconstruct \( f(x, y) \).

![Figure 5.39](image_url)

**Figure 5.39** Two images and their sinograms (Radon transforms). Each row of a sinogram is a projection along the corresponding angle on the vertical axis. Image (c) is called the Shepp-Logan phantom. In its original form, the contrast of the phantom is quite low. It is shown enhanced here to facilitate viewing.

**Figure 5.39** (b) is the sinogram of the rectangle shown in **Figure 5.39** (a).

**Figure 5.39** (c) shows an image of the Shepp-Logan phantom, a widely used synthetic image designed to simulate the absorption of major areas of the brain. The sinogram of **Figure 5.39** (c) is shown in **Figure 5.39** (d).
To obtain a formal expression for a back-projected image from Radon transform, referring to Figure 5.37, we begin with a single point, \( g(\rho_j, \theta_k) \), of the complete projection, \( g(\rho, \theta_k) \), for a fixed value of rotation, \( \theta_k \).

Forming part of an image by back-projecting this single point is simply to copy the line \( L(\rho_j, \theta_k) \) onto the image, where the value of each point in that line is \( g(\rho_j, \theta_k) \). Repeating this process of all values of \( \rho_j \) in the projected signal results

\[
\tilde{f}_{\theta_k}(x, y) = g(\rho, \theta_k) = g(x \cos \theta_k + y \sin \theta_k, \theta_k).
\]

This equation holds for an arbitrary value of \( \theta_k \), therefore, we can write in general that the image formed from a single backprojection obtained at an angle \( \theta \) is given by

\[
f_\theta(x, y) = g(x \cos \theta + y \sin \theta, \theta). \tag{5.11-5}
\]

We form the final image by integrating over all the back-projected images

\[
f(x, y) = \int_0^\pi f_\theta(x, y) d\theta \tag{5.11-6}
\]

In the discrete case, the integral becomes a sum of all back-projected images:

\[
f(x, y) = \sum_{\theta=0}^\pi f_\theta(x, y) \tag{5.11-7}
\]

For example, if 0.5° increments are being used, the summation is from 0 to 179.5°.

A back-projected image formed in this manner is referred to as a laminogram, which is only an approximation to the image from which the projections were generated.
Example 5.18: Obtaining back-projected images from sinograms

Equation

$$f(x, y) = \sum_{\theta=0}^{\pi} f_{\theta}(x, y)$$  \hspace{1cm} (5.11-7)

was used to generate the back-projected images in Figure 5.32 through Figure 5.34 from projections obtained with

$$g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho).$$  \hspace{1cm} (5.11-4)

These equations were also used to generate Figure 5.40 (a) and Figure 5.40 (b), which show the back-projected images corresponding to the sinograms in Figure 5.39 (b) and Figure 5.39 (d).

Note that there is a significant amount of blurring shown in Figure 5.40 (a) and (b). It is obvious that a straight use of Equations (5.11-4) and (5.11-7) will not yield acceptable results.
The Fourier-Slice Theorem

The relationship relating the 1-D Fourier transform of a projection and the 2-D Fourier transform of the region from which the projection was obtained is the basis for reconstruction methods capable of dealing with the blurring problem.

The 1-D Fourier transform of a projection with respect to $\rho$ is

$$G(\omega, \theta) = \int_{-\infty}^{\infty} g(\rho, \theta) e^{-j2\pi\rho \rho} d\rho \quad (5.11-8)$$

where $\omega$ is the frequency variable, and this expression is for a given value of $\theta$.

Substituting

$$g(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy \quad (5.11-3)$$

for $g(\rho, \theta)$ results the expression

$$G(\omega, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\rho \rho} dx dy d\rho$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \left[ \int_{-\infty}^{\infty} \delta( x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\rho \rho} d\rho \right] dx dy$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi( x \cos \theta + y \sin \theta) \rho} dx dy \quad (5.11-9)$$

By letting $u = \omega \cos \theta$ and $v = \omega \sin \theta$, (5.11-9) becomes

$$G(\omega, \theta) = \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux + vy) \rho} dx dy \right]_{u=\omega \cos \theta; v=\omega \sin \theta} \quad (5.11-10)$$

We recognize (5.11-10) as the 2-D Fourier transform of $f(x, y)$ evaluated at the values of $u$ and $v$ indicated.
Equation (5.11-10) leads to

\[ G(\omega, \theta) = \left[ F(u, v) \right]_{u = \omega \cos \theta; v = \omega \sin \theta} = F(\omega \cos \theta, \omega \sin \theta), \quad (5.11-11) \]

which is known as the Fourier-slice theorem (or the projection-slice theorem).

The Fourier-slice theorem states that the Fourier transform of a projection is a slice of the 2-D Fourier transform of the region from which the projection was obtained.

This terminology can be explained with Figure 5.41.

As Figure 5.41 shows, the 1-D Fourier transform of an arbitrary projection is obtained by extracting the values of \( F(u, v) \) along a line oriented at the same angle as the angle used in generating the projection.

In principle, we could obtain \( f(x, y) \) simply by obtaining the inverse Fourier transform \( F'(u, v) \), though it is expensive computationally with the involvement of inverting a 2-D transform.
Reconstruction Using Parallel-Beam Filtered Backprojections

Regarding to the blurred results, fortunately, there is a simple solution based on filtering the projections before computing the backprojections.

Recall

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

the 2-D inverse Fourier transform of \( F(u, v) \) is

\[
    f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux + vy)} du dv .
\]

As in (5.11-10) and (5.11-11), letting \( u = \omega \cos \theta \) and \( v = \omega \sin \theta \), we can express (5.11-12) in polar coordinates:

\[
    f(x, y) = \int_{0}^{2\pi} \int_{0}^{\infty} F(\omega \cos \theta, \omega \sin \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta \quad (5.11-13)
\]

Then, using the Fourier-slice theorem, we have

\[
    f(x, y) = \int_{0}^{2\pi} \int_{0}^{\infty} G(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta .
\]

Using the fact that \( G(\omega, \theta + \pi) = G(-\omega, \theta) \), we can express (5.11-14) as

\[
    f(x, y) = \int_{0}^{2\pi} \int_{-\infty}^{\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} d\omega d\theta .
\]

In terms of integration with respect to \( |\omega| \), the term \( x \cos \theta + y \sin \theta \) is a constant, which is recognized as \( \rho \). Thus, (5.11-15) can be written as

\[
    f(x, y) = \int_{0}^{\pi} \left[ \int_{-\infty}^{\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega \rho} d\omega \right] d\theta .
\]
Recall

\[ f(t) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t} d\mu, \quad (4.2-17) \]

the inner expression in (5.11-16) is a 1-D inverse Fourier transform with the added term \(|\omega|\).

Based on the discussion in Section 4.7, \(|\omega|\) is a one-dimensional filter function.

\[ |\omega| \]

is not integrable, because its amplitude extends to \(\pm\infty\) in both directions, so the inverse Fourier transform is undefined.

In practice, the approach is to window the ramp so it becomes zero outside of defined frequency interval, as shown in Figure 5.42 (a).

**Figure 5.42**

(a) Frequency domain plot of the filter \(|\omega|\) after band-limiting it with a box filter. (b) Spatial domain representation. (c) Hamming windowing function. (d) Windowed ramp filter, formed as the product of (a) and (c). (e) Spatial representation of the product (note the decrease in ringing).

Figure 5.42 (b) shows its spatial domain representation, obtained by computing its inverse Fourier transform. The resulting windowed filter exhibits noticeable ringing in the spatial domain. As discussed in Chapter 4, windowing with a smooth function will help in this situation.
An M-point discrete window function used frequently for implementation with the 1-D FFT is given by

\[
h(\omega) = \begin{cases} 
  c + (c - 1) \cos \frac{2\pi \omega}{M - 1} & 0 \leq \omega \leq (M - 1) \\
  0 & \text{otherwise}
\end{cases} \quad (5.11-17)
\]

When \( c = 0.54 \), this function is called the Hamming window.

Figure 5.42 (c) is a plot of the Hamming window, and Figure 5.42 (d) shows the product of this window and the band-limited ramp filter shown in Figure 5.42 (a).

Figure 5.42 (e) shows the representation of the product in the spatial domain, obtained by computing the inverse FFT.

Comparing Figure 5.42 (e) and Figure 5.42 (b), we can find that ringing was reduced in the window ramp.

On the other hand, because the width of the central lobe in Figure 5.42 (e) is slightly wider than that of Figure 5.42 (b), we would expect backprojections based on a Hamming window to have less ringing but be slightly more blurred.

Recalling

\[
G(\omega, \theta) = \int_{-\infty}^{\infty} g(\rho, \theta) e^{-j2\pi\omega \rho} d\rho \quad (5.11-8)
\]

that \( G(\omega, \theta) \) is the 1-D Fourier transform of \( g(\rho, \theta) \), which is a single projection obtained at a fixed angle, \( \theta \).
Equation

\[ f(x, y) = \int_0^\pi \left[ \int_{-\infty}^{\infty} |\omega| G(\omega, \theta) e^{j2\pi\rho\omega} d\omega \right] \rho = x \cos \theta + y \sin \theta \ d\theta \]  

(5.11-16)

states that the complete, back-projected image \( f(x, y) \) is obtained as follows:

1. Compute the 1-D Fourier transform of each projection.
2. Multiply each Fourier transform by the filter function \( |\omega| \), which has been multiplied by a suitable (e.g., Hamming) window.
3. Obtain the inverse 1-D Fourier transform of each resulting filtered transform.
4. Integrate (sum) all the 1-D inverse transform from Step 3.

This image reconstruction approach is called filtered backprojection.

In practice, because the data are discrete, all frequency domain computations are carried out using a 1-D FFT algorithm, and filtering is implemented using the same basic procedure explained in Chapter 4 for 2-D functions.
Example 5.19: Image reconstruction using filtered backprojections

Figure 5.43 (a) shows the rectangle reconstructed using a ramp filter. The most vivid feature of this result is the absence of any visually detectable blurring. However, ringing is present, visible as faint lines, especially around the corners of the rectangle. Figure 5.43 (c) can show these lines in the zoomed section.

Using a Hamming window on the ramp filter helped considerably with the ringing problem, at the expense of slight blurring, as Figure 5.43 (b) and Figure 5.43 (d) show.
The reconstructed phantom images shown in Figure 5.44 are from using the un-windowed ramp filter and a Hamming window on the ramp filter.

Since the phantom image does not have transitions that are sharp and prominent as the rectangle, so ringing is imperceptible in this case, though result shown in Figure 5.44 (b) is a slightly smooth than that of Figure 5.44 (a).

The discussion has been based on obtaining filtered backprojections via an FFT implementation. However, from the convolution theorem introduced in Chapter 4, we know that the equivalent results can be obtained using spatial convolution.

Note that the term inside the brackets in

\[ f(x, y) = \int_0^{\pi} \left[ \int_{-\infty}^{\infty} |\omega| G(\omega, \theta) e^{j2\pi \omega \rho} d\omega \right]_{\rho = x \cos \theta + y \sin \theta} d\theta \quad (5.11-16) \]

is the inverse Fourier transform of the product of two frequency domain functions. According to the convolution theorem, they are equal to the convolution of the spatial representations (inverse Fourier transform) of these two functions.

Let \( s(\rho) \) denote the inverse Fourier transform of \(|\omega|\), we can write (5.11-16) as
The last two lines of (5.11-18) say the same thing: Individual backprojections at an angle $\theta$ can be obtained by convolving the corresponding projection, $g(\rho, \theta)$, and the inverse Fourier transform of the ramp filter, $s(\rho)$.

With the exception of round off differences in computation, the results of using convolution will be identical to the results using FFT.

In general, convolution turns out to be more computationally efficient and is used in most of modern CT systems, while Fourier transform plays a central role in theoretical formulations and algorithm development.
5.9 Constrained Least Squares Filtering

Other than having to know something about the degradation function $H$, the Wiener filter presents an additional difficulty: the power spectra of the undegraded image and noise, $S_f(u,v)$ and $S_\eta(u,v)$, must be known.

Although it is possible to achieve excellent results using the approximation given in

$$\hat{F}(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} G(u,v), \quad (5.8-6)$$

a constant estimate is not always a suitable solution.

By using the definition of 2-D circular convolution

$$f(x,y) \circledast h(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m,y-n), \quad (4.6-23)$$

we can express

$$g(x,y) = h(x,y) \circledast f(x,y) + \eta(x,y) \quad (5.5-16)$$

in vector-matrix form:

$$g = Hf + \eta. \quad (5.9-1)$$

Suppose that $g(x,y)$ is of size $M \times N$, we can form the first $N$ elements of the vector $g$ by using the image elements in first row of $g(x,y)$, the next first $N$ elements the second row, and so on.

The resulting vector will have dimensions $MN \times 1$. These are also the dimensions of $f$ and $\eta$. The matrix $H$ then has dimensions $MN \times MN$. 
Although the **restoration** problem seems to have been reduced to simple matrix manipulations, due to the huge sizes of matrices, manipulating vectors and matrices is not a trivial task.

However, formulating the **restoration** problem in matrix form does facilitate derivation of restoration techniques.

The method of **constrained least squares** has its roots in a matrix formulation.

Central to the method is the issue of the sensitivity of $H$ to noise. One way to lessen the noise sensitivity problem is to base optimality of restoration on a measure of smoothness, such as the second derivative of an image.

To find the minimum of a criterion function, $C$,

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

subject to the constraint

$$
\| g - H\hat{f} \|^2 = \| \eta \|^2, \quad (5.9-2)
$$

where $\| w \|^2 \triangleq w^T w$ is the Euclidean vector norm, and $\hat{f}$ is the estimate of the undegraded image. The **Laplacian operator** $\nabla^2$ is defined in

$$
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \quad (3.6-3)
$$

The **frequency domain** solution to this optimization problem is
\[
\hat{F}(u, v) = \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma |P(u, v)|^2} \right] G(u, v), \quad (5.9-4)
\]

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

\[
p(x, y) = \begin{bmatrix}
0 & -1 & 0 \\
-1 & -4 & -1 \\
0 & -1 & 0
\end{bmatrix}. \quad (5.9-5)
\]

Example 5.14: Comparison of Wiener and constrained least squares filtering

It is possible to adjust the parameter \( \gamma \) until acceptable results are achieved.

A procedure for computing \( \gamma \) by iteration:

Define a “residual” vector \( r \) until as

\[
r = g - H\hat{f}. \quad (5.9-6)
\]

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.
According to (5.9-4), $\hat{F}(u,v)$ (and by implication $\hat{f}$) is a function of $\gamma$, then $r$ also is a function of this parameter. It can be shown that
\[
\phi(\gamma) = r^T r = \|r\|^2 \tag{5.9-7}
\]
is a monotonically increasing function of $\gamma$. What we want to do is to adjust $\gamma$ so that
\[
\|r\|^2 = \|\eta\|^2 \pm a, \tag{5.9-8}
\]
where $a$ is an accuracy factor. If $\|r\|^2 = \|\eta\|^2$, the constraint in
\[
\|g - H\hat{f}\|^2 = \|\eta\|^2 \tag{5.9-3}
\]
will be strictly satisfied.

One approach to find $\gamma$ is

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

Other procedures, such as a Newton-Raphson algorithm, can improve the speed of convergence.

In order to use the abovementioned algorithm, we need the quantities $\|r\|^2$ and $\|\eta\|^2$. 
Note from
\[ r = g - H\hat{f}, \quad (5.9-6) \]
we have
\[ R(u,v) = G(u,v) - H(u,v)\hat{F}(u,v). \quad (5.9-9) \]

Then we can obtain \( r(x,y) \) by computing the inverse transform of \( R(u,v) \), and
\[
\|r\|^2 = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} r^2(x,y). \quad (5.9-10)
\]

To compute \( \|\eta\|^2 \), first, we consider the variance of the noise over the entire image, which we estimate by the sample-average method (discussed in Chapter 3):
\[
\sigma_{\eta}^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[ \eta(x,y) - m_{\eta} \right]^2, \quad (5.9-11)
\]
where
\[
m_{\eta} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \eta(x,y) \quad (5.9-12)
\]
is the sample mean.

Referring to the form of (5.9-10), the double summation in (5.9-11) is equal to \( \|\eta\|^2 \). This gives us the expression
\[
\|\eta\|^2 = MN \left[ \sigma_{\eta}^2 + m_{\eta}^2 \right], \quad (5.9-13)
\]
which tells us that we can implement an optimum restoration algorithm by having knowledge of only the mean and variance of the noise.
Example 5.15: Iterative estimation of the optimum constrained least squares filter.

![Images](image1.jpg) ![Images](image2.jpg)

**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.

Note: Fig. 5.16 (b) should be Fig. 5.25 (b).

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.