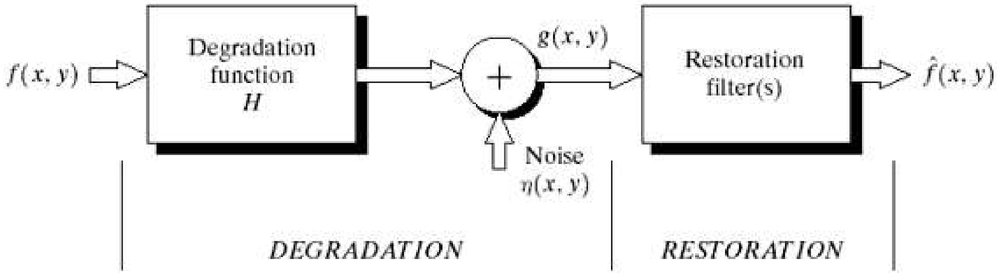


FIGURE 5.1 A model of the image degradation/restoration process.

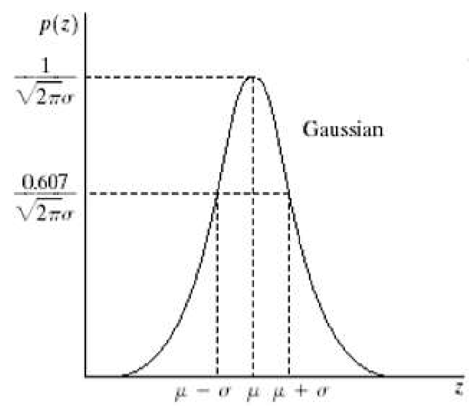
$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

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

Gaussian Noise

Gaussian or Normal (mathematical tractability in both the spatial and frequency domains)

$$p(z;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

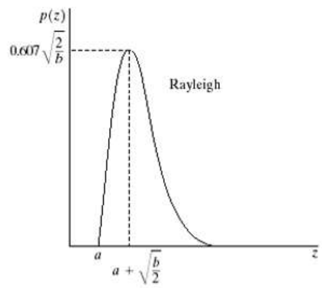


where z represents the grey level, μ is the mean or average value of z and σ^2 is called the variance of z

Rayleigh Noise

The PDF of Rayleigh noise is given by

$$p(z) = \begin{cases} \frac{2}{b} e^{-(z-a)^2/b} & for\ z \geq a \\ 0 & for\ z \prec a \end{cases}$$



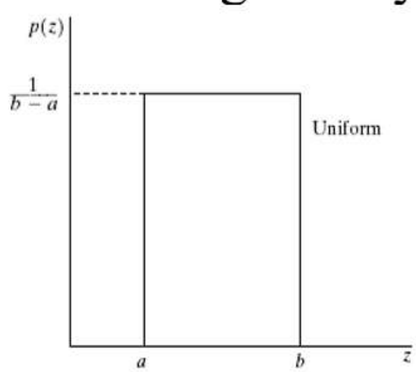
the mean and variance of this density are given by

$$\mu = a + \sqrt{\pi b/4} \qquad \sigma^2 = \frac{b(4-\pi)}{4}$$

Uniform Noise

The PDF of the uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & if\ a \leq z \leq b \\ 0 & otherwise \end{cases}$$



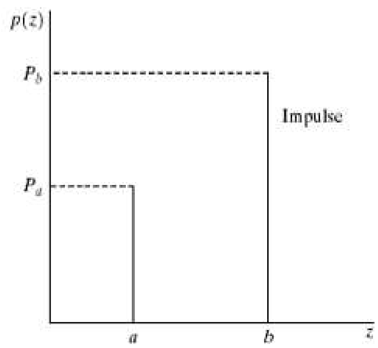
the mean and variance of this density are given by

$$\mu = \frac{a+b}{2} \qquad \sigma^2 = \frac{(b-a)^2}{12}$$

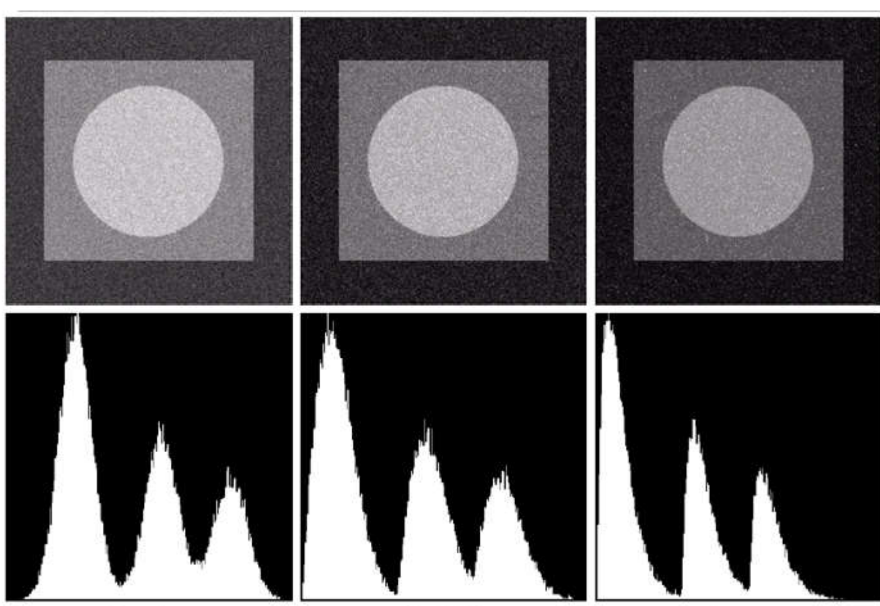
Impulse (Salt & Pepper) Nois

The PDF of bi-polar impulse noise is given by

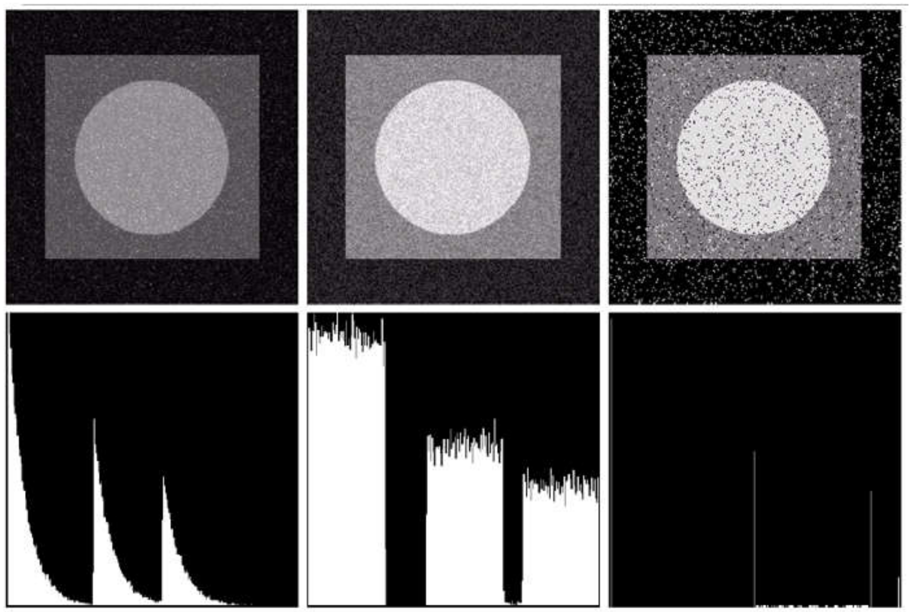
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$



If $b >> a$, grey-level b will appear as a light dot in the image; while grey-level a will appear as a dark dot. If either P_a or P_b is zero, the impulse noise is called *unipolar*. If neither is zero and both are equal, the noise values will resemble salt and pepper granules randomly distributed over the image.



Gaussian Rayleigh Gamma



Exponential Uniform Salt & Pepper

Estimation of Noise Parameters

- Periodic noise:
  - inspection in the frequency domain, or
  - from image direcly (in simple cases only)
- Non-periodic noise:
  - noise PDF parameters may be known partially from sensor specifications, or
  - can be estimated from relatively flat image areas. Mean and variance are estimated as follows:

$$\mu = \sum_{z_i \in S} z_i p(z_i) \quad and \quad \sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 p(z_i)$$

where the z_i's are the gray-level values of the pixels in S and $p(z_i)$ are the corresponding normalized histogram values.

Restoration in the presence of noise (only spatial filtering)

1.mean filter

Arithmetic mean filter:

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

This operation can be implemented using a convolution mask in which all coefficients have value $1/mn$. As discussed in Section 3.6.1, a mean filter simply smoothes local variations in an image. Noise is reduced as a result of blurring.

Geometric mean filter:

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

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

Comparing both

filter of size 3×3 and a geometric mean filter of the same size. Although both filters did a reasonable job of attenuating the contribution due to noise, the geometric mean filter did not blur the image as much as the arithmetic filter. For instance,

Harmonic mean filter

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

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

Contraharmonic mean filter:

4. Contraharmonic mean filter

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

where Q is called the filter order.

- **Effects**: well suited for salt&pepper noise:
  - Q>0, it eliminates pepper noise, and
  - Q<0, it removes salt noise, but
  - it cannot eliminate both types of noises simultaneously!
- **Question**: what happens when $Q=0$ and $Q=-1$?

Arithmetic mean and Harmonic mean, respectively!

Note that:

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

2.Order-statistics filter:

Median filter

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

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

The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. In fact, as Ex-

Max and min filter:

the median filter represents the 50th percentile of a ranked set of numbers, but max filter represents the 100th percentile and min filter uses the 0th percentile.

Max:

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

Min:

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

Midpoint filter:

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

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

Order Statistics Filters

4. Alpha-trimmed mean filter

We remove *d/2* lowest and *d/2* highest samples from the local input set (g_r represents the remaining samples) and average the remaining samples:

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

- Combines order statistics and averaging,
- Works well for combination of noises, like Gaussian and salt&pepper noises.

Question: what happen when *d=0* and *d/2=(mn-1)/2*?

Arithmetic mean and median, respectively!

Adaptive filter:

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

where $\hat{f}$ is the filter output, g(x,y) is the degraded image to be restored, σ_η^2 is the variance of the noise, σ_L^2 is the local variance of the pixels in a local neighborhood S_{xy} , and m_L is the local mean of the pixels in S_{xy}. **Discuss the behavior** of the filter (a) when there is no noise, (b) when the local area contains important details, e.g. edges, and (c) when the local area is smooth.

Answer:

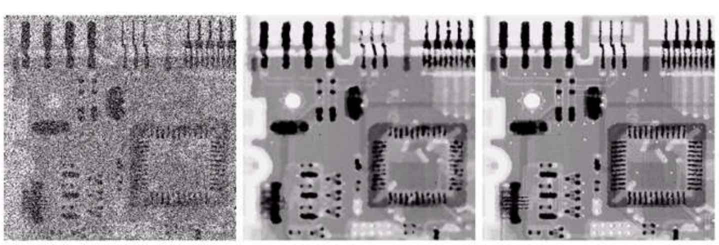
(a) If σ_η^2 is zero, the filter returns the value of g(x,y). This is the trivial zero-noise case in which f(x,y)=g(x,y).

(b) if the local variance is high relative to σ_η^2, the filter returns a value close to g(x,y). A high local variance is typically associated with edges and these should be preserved.

(c) When the local area is smooth, the two variances would be equal and the filter returns the arithmetic mean of the pixels.

Adaptive median:

Image Restoration: adaptive median



a b c

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

The median filter can be adapted in several ways, e.g. selecting a direction for filtering inside a fixed window or varying the window size, according to some local statistics.

Periodic noise reduction by frequency domain filtering:

1.bandreject filters:

5.4.1 Bandreject Filters

Bandreject filters remove or attenuate a band of frequencies about the origin of the Fourier transform. An ideal bandreject filter is given by the expression

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v) < D_0-\dfrac{W}{2}\\ 0 & \text{if } D_0-\dfrac{W}{2}\le D(u,v)\le D_0+\dfrac{W}{2}\\ 1 & \text{if } D(u,v) > D_0+\dfrac{W}{2}\end{cases} \tag{5.4-1}$$

where $D(u, v)$ is the distance from the origin of the centered frequency rectangle, as given in Eq. (4.3-3), W is the width of the band, and D_0 is its radial center.

Similarly, a Butterworth bandreject filter of order n is given by the expression

$$H(u,v)=\frac{1}{1+\left[\dfrac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}} \tag{5.4-2}$$

and a Gaussian bandreject filter is given by

$$H(u,v)=1-e^{-\frac{1}{2}\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^2} \tag{5.4-3}$$

Figure 5.15 shows perspective plots of these three filters.

One of the principal applications of bandreject filtering is for noise removal in applications where the general location of the noise component(s) in the frequency domain is approximately known. A good example is an image corrupted

2.bandpass filters:

A *bandpass* filter performs the opposite operation of a bandreject filter. In Sec-

$$H_{bp}(u,v)=1-H_{br}(u,v).$$

It removes too much image detail.

3.notch filters:

A *notch* filter rejects (or passes) frequencies in predefined neighborhoods about a center frequency. Figure 5.18 shows 3-D plots of ideal, Butterworth, and Gauss-

The transfer function of an ideal notch reject filter of radius D_0, with centers at (u_0, v_0) and, by symmetry, at $(-u_0, -v_0)$, is

$$H(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) \le D_0 \quad \text{or} \quad D_2(u,v) \le D_0 \\ 1 & \text{otherwise} \end{cases} \tag{5.4-5}$$

where

$$D_1(u,v) = \left[(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2\right]^{1/2} \tag{5.4-6}$$

$$D_2(u,v) = \left[(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2\right]^{1/2} \tag{5.4-7}$$

The transfer function of a Butterworth notch reject filter of order n is given by

$$H(u,v) = \frac{1}{1 + \left[\dfrac{D_0^2}{D_1(u,v)D_2(u,v)}\right]^n} \tag{5.4-8}$$

where $D_1(u, v)$ and $D_2(u, v)$ are given in Eqs. (5.4-6) and (5.4-7), respectively. A Gaussian notch reject filter has the form

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]} \tag{5.4-9}$$

Optimum notch filtering:

Read it again, I don't understand it at all

when several periodic interference components are present in transform spectrum, the notch filtering method may not work because they may remove too many much image information in the filteringprocess. We have to use optimum method, In this sense that it minimizes local variances of the restored estimate $\hat{f}$(x,y).

The procedure consists of first isolating the principal contributions of the interference pattern and then subtracting a variable, weighted portion of the pattern from the corrupted image. Although we develop the procedure in the context

Linear-position-invariant degradation:

Let's also assume that H is position (or space) invariant:

$$H[f(x-a, y-b)] = g(x-a, y-b)$$

for any $f(x,y)$ and a and b, i.e. the response at any point in the image depends only on the value of the image at that point, not on its position.

Estimating the degradation function:

Estimation by image observation:

Assuming no noise, we look at parts of the image, e.g. background or part of an object, and try to deblur it.

Denote by $g_s(x,y)$ the observed sub-image and $f_s(x,y)$ the constructed sub-image, then

$$H_s(u,v) = \frac{G_s(u,v)}{F_s(u,v)}$$

from the characteristics of this function, one can deduce the complete transfer function $H(u,v)$.

Estimation by experimentation:

Assuming that similar equipment to the ones used in acquisition exist, then one can obtain an accurate estimation of the degradation.

Idea: try to get the impulse response of the degradation by imaging an impulse using the same system settings. Then

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

where A is the FT of an impulse.

Estimation by modeling:

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

where k is constant that depends on the nature of the turbulence.

Note that this is almost a Gaussian! In fact, Gaussian LPF has been used to model mild, uniform blurring.

Another major approach in modeling is to derive a mathematical model starting from basic principles. We illustrate this procedure by treating in some detail the case in which an image has been blurred by uniform linear motion between the image and the sensor during image acquisition. Suppose that an image

Inverse filtering:

inversing filtering:

The simplest approach to restoration is direct inverse filtering. assume that we have estimated H. In the absence of any information concerning noise, we get

مربوط به

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

here we have some problems. even we know H, we can not ~~restore~~ recover f(x,y) because we dont know N(u,v). also, if the H(u,v) is zero or very small values then the ratio $\frac{N(u,v)}{H(u,v)}$ could easily dominate the estimate $\hat{F}(u,v)$.

One approach can get around this is to limit the filter frequencies to values near the origin. e.g. pseudo-inverse filtering

Wiener filter:

Remark:

Inverse and pseudo-inverse filtering reverse the effects of system only; but can do nothing about the random noise in the signal.

Alternative Solution: Wiener Filtering

Wiener filtering has been successfully used to filter images corrupted by noise and blurring. The idea of Wiener filtering is to find the "best" estimate of the true input *u(m,n)* from the observed image *v(m,n)* by modeling the input and output images as random sequences.

"best" in the mean square error sense.

The optimization problem is then stated as: find the estimate of *u(m,n)* which minimizes

$$\sigma_e^2 = E\{[u(m,n) - \hat{u}(m,n)]^2\}$$

- Problems:
  - this is a nonlinear function, and
  - the conditional probability function needed to compute it is very difficult to calculate.
- Cure:
  - Restrict the estimate to be linear, i.e., find *g(.,.;.,.)* in

$$\hat{u}(m,n) = \sum \sum g(m,n;k,l)v(k,l)$$

To minimize:

$$\sigma_e^2 = E\{[u(m,n) - \hat{u}(m,n)]^2\}$$

Interpretation of the Wiener Filter

- In the absence of blur, H=1, the Wiener filter becomes a smoothing filter (work it out)

- When there is no noise, the Wiener filter reduces to the inverse filter (work it out)

- When both noise and blur are present, the Wiener filter seeks a compromise between: lowpass noise smoothing, and highpass deblurring; the result is a bandpass filter. However, deblurring decreases rapidly as the noise power increases. Experiment with this!