IMAGE RESTORATION AND RECONSTRUCTION

1. **Model of the Image Degradation/Restoration Process,**
2. **Noise Models, Restoration in the Presence of Noise Only—Spatial Filtering**
3. **Inverse Fitering**

   (Weiner Filters)

4. Periodic Noise Reduction Using Frequency Domain Filtering,

Image Degradation/Restoration Process

In digital Image acquistion there are problems in Acquiring Good quality Images is called Degradation Phenomenon.

1. Imaging Sensors: Quality of Sensing devices
2. Environmental conditions: Lighting levels
3. Image transmission: Losing quality when transmitted over Network
4. Wrong adjustment of Apertures

} Noise

OUTPUT: Image quality is reduced or Corrupted.

While displaying such low quality image we have to enhance the degraded image upto acceptable quality is called Image Resotration.

Since we know the sources of low quality images are due to the problems in Hardware, Environment, Transmission Errors we can some times predict (or) estimate the amount of degradation and restore them.

In this chapter we discuss more about:

- **How to understand better about such degradation in Acquiring images/ Transmitted images**
- **Enhance or Restore such images**

NOISE Models of Digital Images

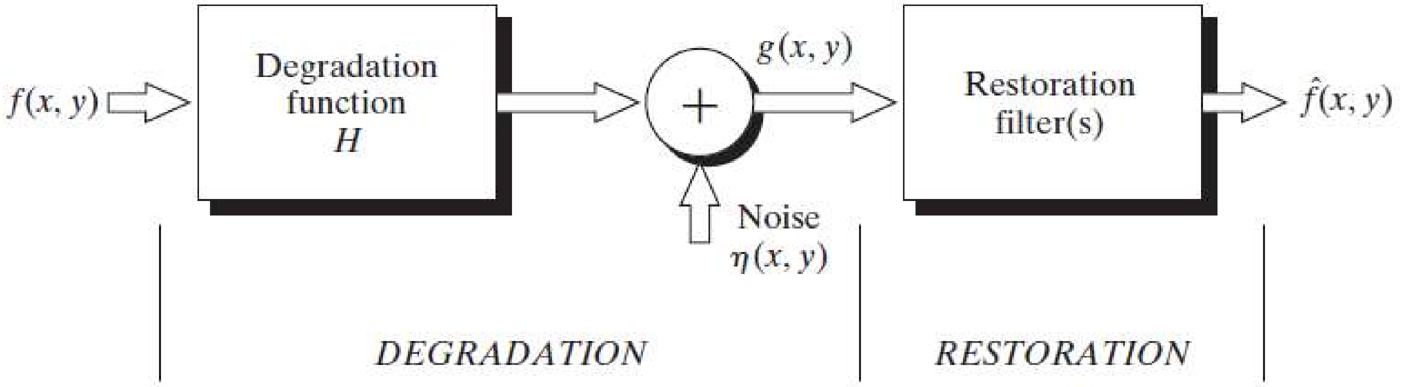


FIGURE 5.1 A model of the image degradation/restoration process.

Spatial Domain:
g(x,y) = f(x,y) * h(x,y) + ή(x,y)
f*h →f convolve with h

f(x,y): Input Image
Degradation stage:
H: Degradation Function that reduces quality of the image f(x,y)
Noise: **ή(x,y**) [Read **ή** as neu]: Noise Function that added
g(x,y): Total Low Grade or Corrupted Image
Restoration Filter(s): Filters that can apply on Images
Objective of Restoration stage is:
$\hat{f}$(x,y): f-cap of (x,y) : Estimated Image or restored Image

Frequency Domain:
G(u,v) = F(u,v) H(u,v) + ή(u,v)
Convolve → Multiplication

If we Know the Degradation function and Noise that added to input Image, we can find corresponding Restoration Filter to estiamate Image restoration.

Digital Image Noise

Noises are invariably present in Digital Image acquisiton due to various factors which we don’t have control some of them are given below:

1. Imaging Sensors: Degradation of Quality of Sensing devices
2. Environmental conditions: Lighting levels & Sensing temperatures
3. Image transmission: Losing quality when transmitted over Network
4. Wrong adjustment of Apertures

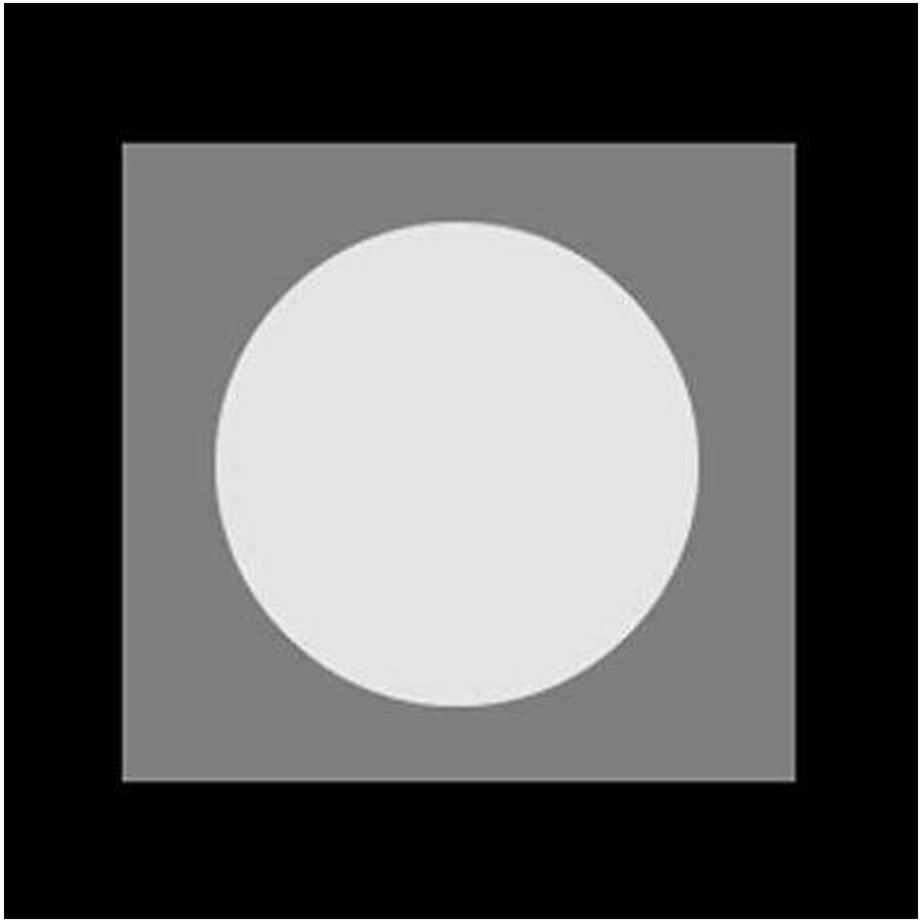
Noise

To Understand Image Noise we understand Spatial and Frequency properties of Noise

We have some hidden pixels intensity in the original image, if we want to find what are such pixels contributing to that, how will you find?

Image Histograms?
Gives Probability of Intensity values.
Lets understand the PDF of Image Noise

Types of Noises and their understanding using Histograms and PDF (Probability Density Function)



Let us consider a Digital Image having limited number of Intensities. a test pattern well suited for illustrating the noise models just discussed.

This is a suitable pattern to use because it is composed of simple, constant areas that span the gray scale from black to near white in only three increments.

This facilitates visual analysis of the characteristics of the various noise components added to the image.

Subsequently add different types of Noise and analyse their effect on Output Image.

Types of Noises in Images (Spatial Domain)

- Gaussian Noise
- Reyleigh Noise
- Erlang Noise
- Exponential Noise
- Uniform Noise
- Impulse Noise

Noise Probability Density Function (PDF):

Gaussian Noise:

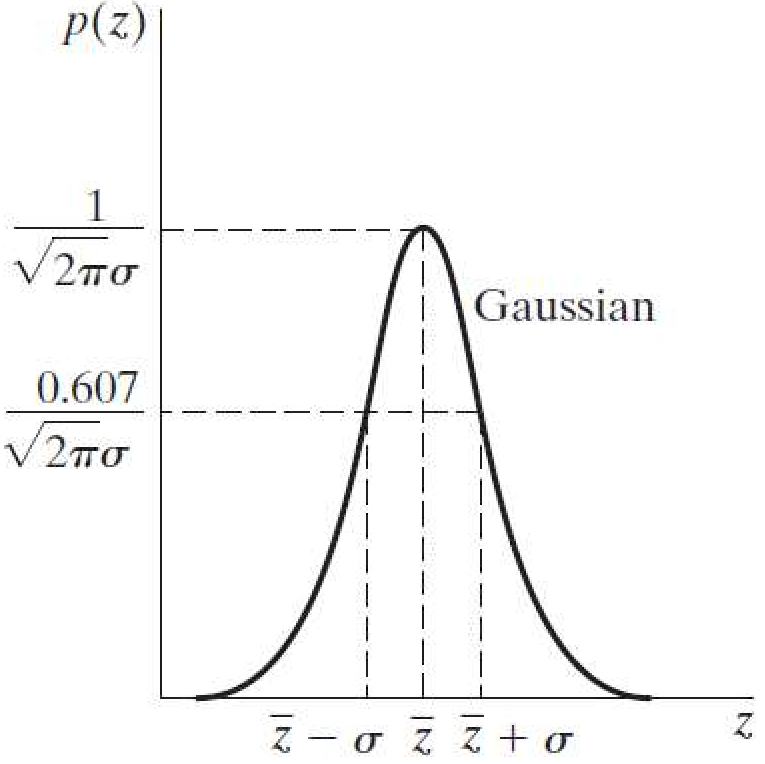
The PDF of a Gaussian random variable, z, is given by

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

where z represents intensity, ž is the mean (average) value of *z*, ***σ** (sigma) is its standard* deviation. The standard deviation squared, **σ^2** is called the *variance of z.*

A plot of this function is shown in Fig. When z is described by Eq. (5.2-1), approximately 70% of Noise causing Pixel values will be in the range [(ž - ***σ***), (ž + ***σ***)], and about 95% will be in the range [(z - 2 ***σ***), (z + 2 ***σ***)].

Graph Right shows is plot using the Eq 5.2-1.

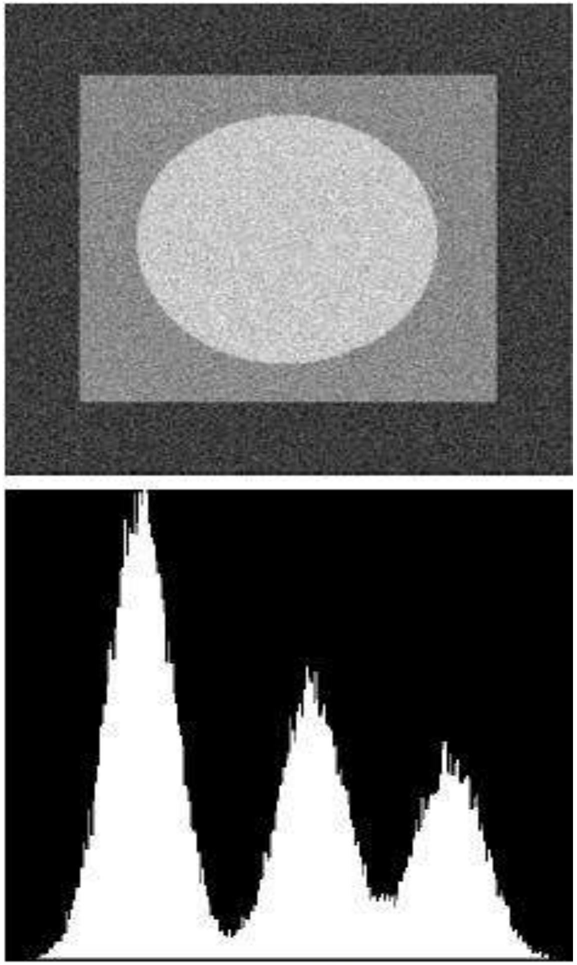


Gaussian Noise PDF:

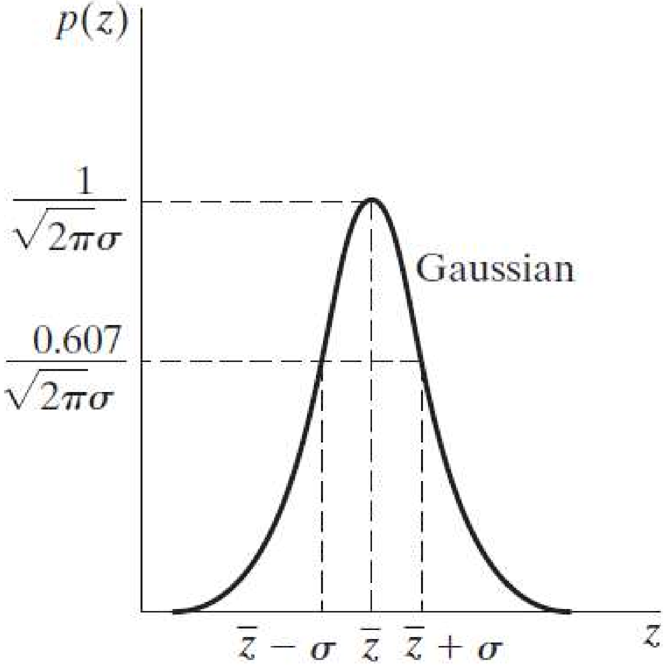
Gaussian Noise:

Lets add the Gaussian Noise to the Input Image shown in the SLide 5, see the output below along with the Histogram PDF function. Right the Gaussian PDF studied in the previous Slide. **Do they Look Similar?**

Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and/or high temperature



Gaussian



Reyleigh Noise PDF:

Reyleigh Noise:

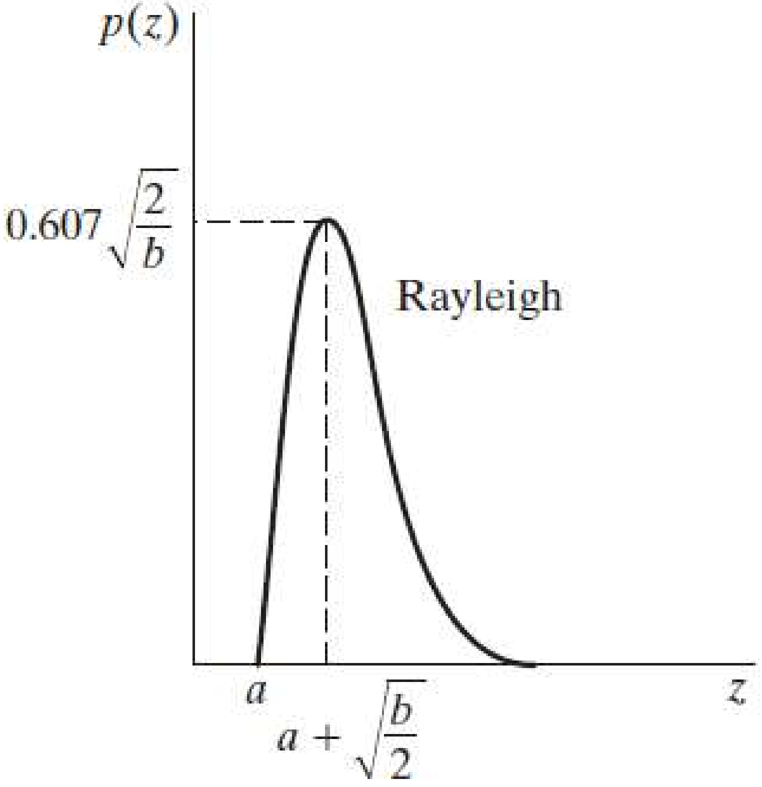
The PDF of a Reyleigh random variable, z, is given by

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

where a,b are constants. The mean and variance of this density are given by

$$\bar{z} = a + \sqrt{\pi b/4} \quad (5.2\text{-}3)$$

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



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

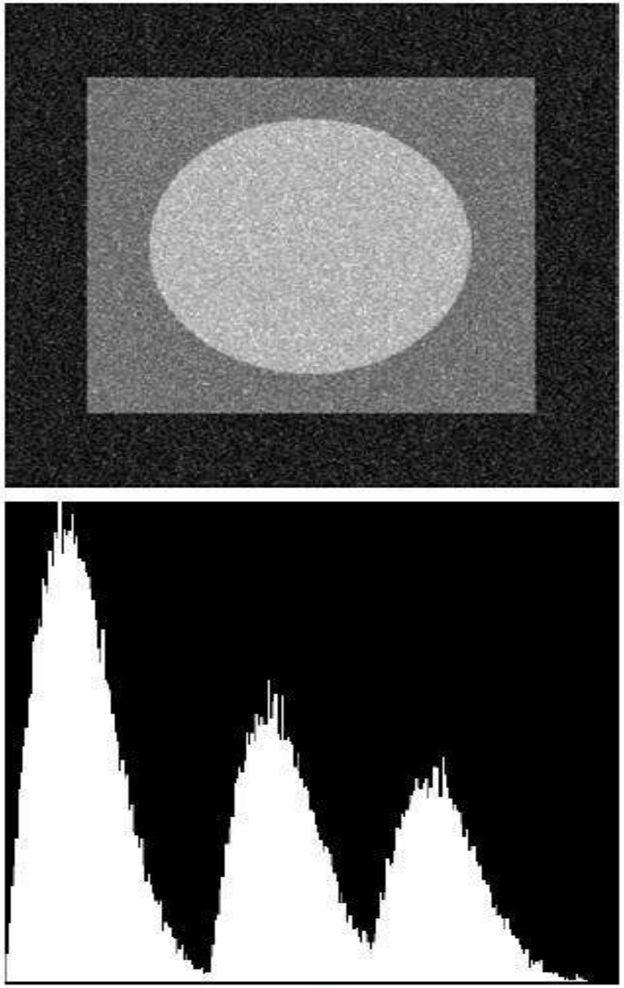
Graph Right shows is plot using the Eq 5.2-2.

Reyleigh Noise PDF:

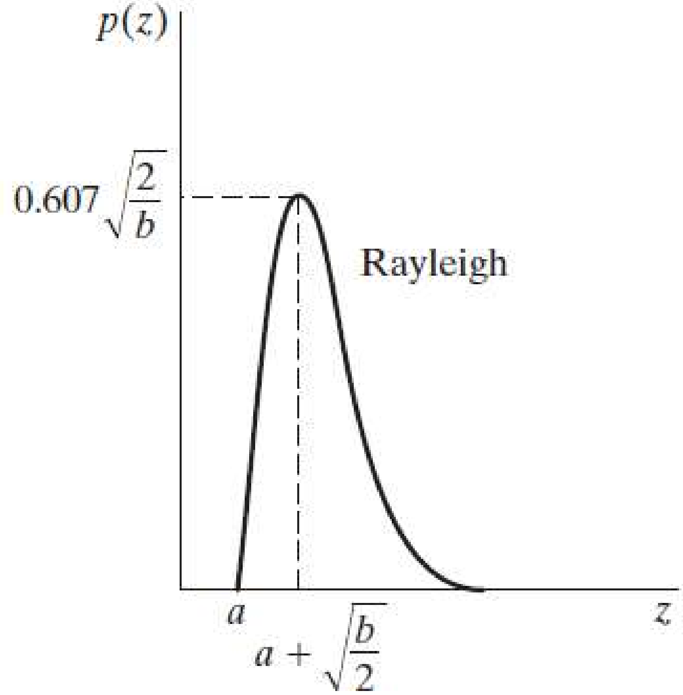
Reyleigh Noise:

Lets add the Releigh Noise to the Input Image shown in the SLide 5, see the output below along with the Histogram PDF function. Right the Reyleigh PDF studied in the previous Slide. **Do they Look Similar?**

The Rayleigh density is helpful in characterizing noise phenomena in range imaging.



Rayleigh



Erlang/Gamma Noise PDF:

Erlang Noise: (Gamma Noise)

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

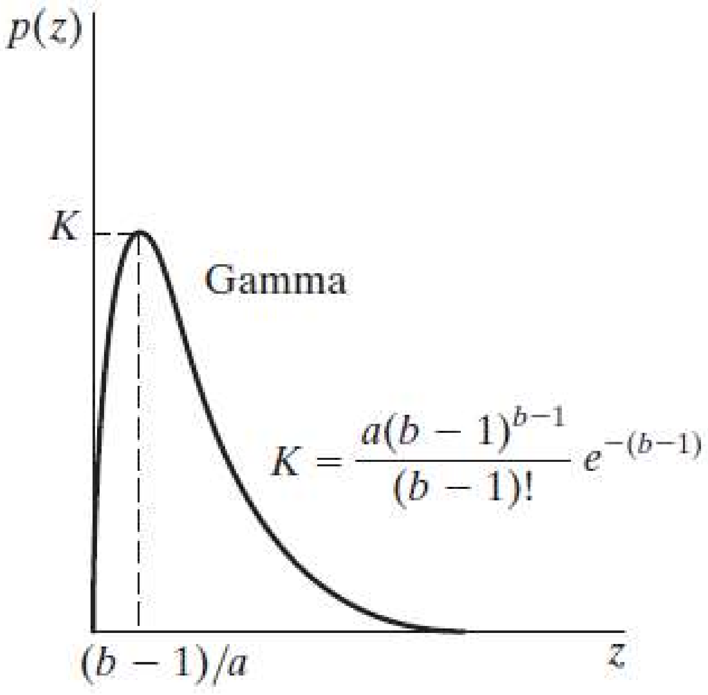
$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \qquad (5.2\text{-}5)$$

where the parameters are such that a > 0, b is Positive Integer. “!” indicates factorial.

The mean and variance of this density are given by

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

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



Graph Right shows is plot of Erlang Density using t Eq 5.2-5.

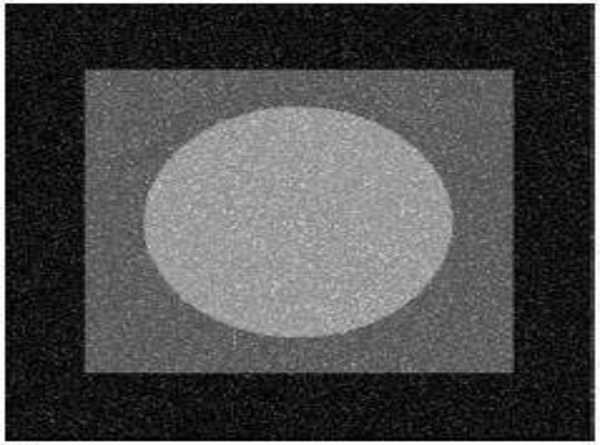
Erlang/Gamma Noise PDF:

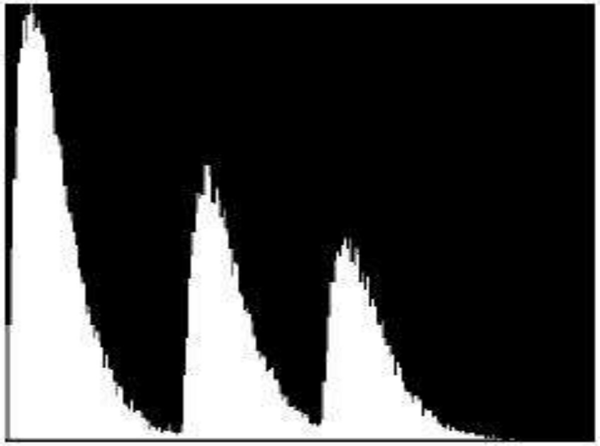
Erlang Noise:

Lets add the Erlang Noise to the Input Image shown in the SLide 5, see the output below along with the Histogram PDF function. Right the Erlang PDF studied in the previous Slide.

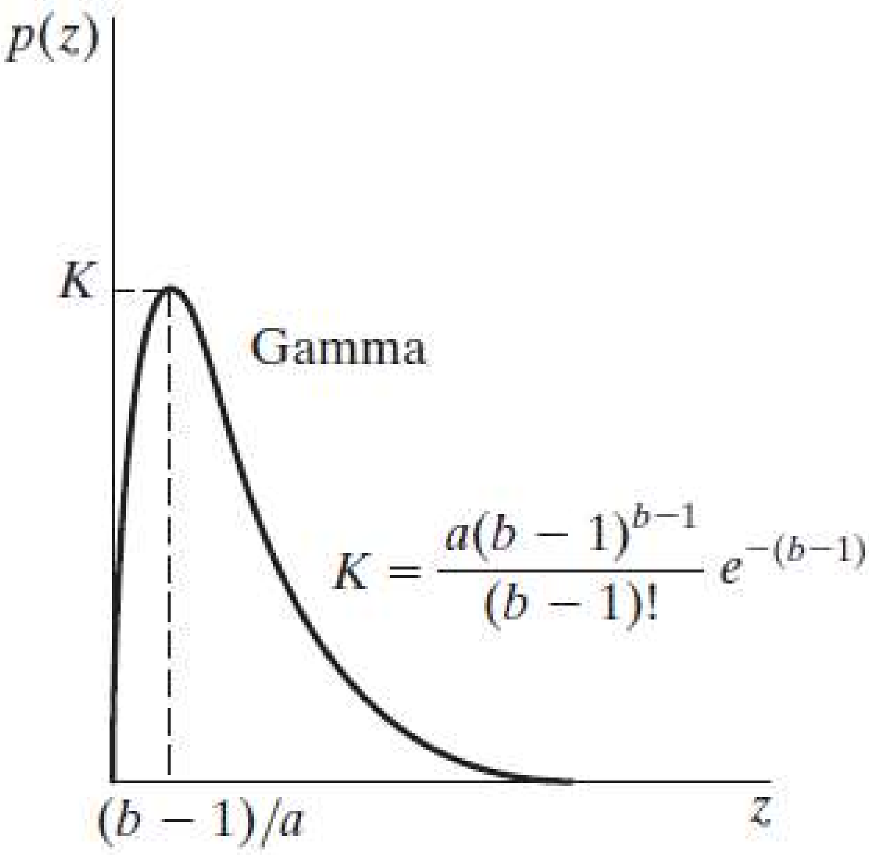
Do they Look Similar?

The gamma densities find application in laser imaging.





Gamma



Exponential Noise PDF:

Exponential noise

The PDF of *exponential* noise is given by

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

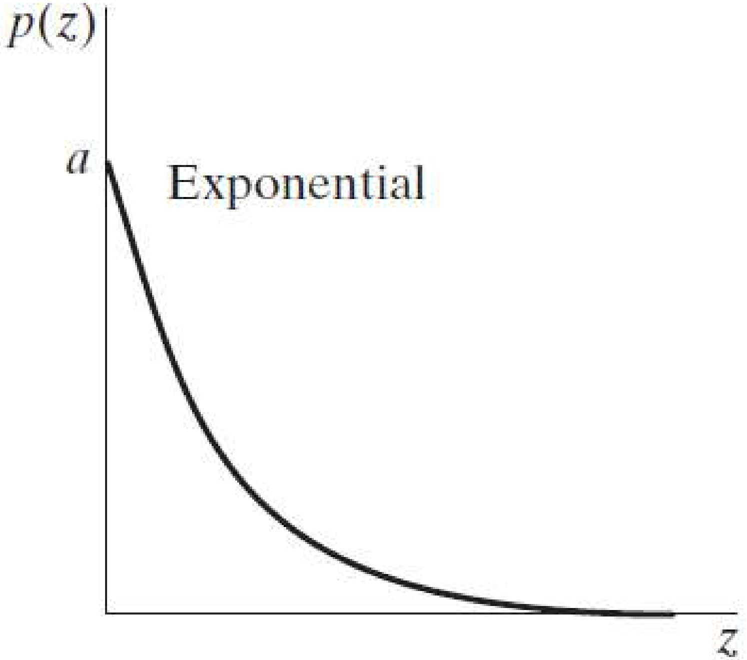
where $a > 0$. The mean and variance of this density function are

$$\bar{z} = \frac{1}{a} \qquad (5.2\text{-}9)$$

and

$$\sigma^2 = \frac{1}{a^2} \qquad (5.2\text{-}10)$$

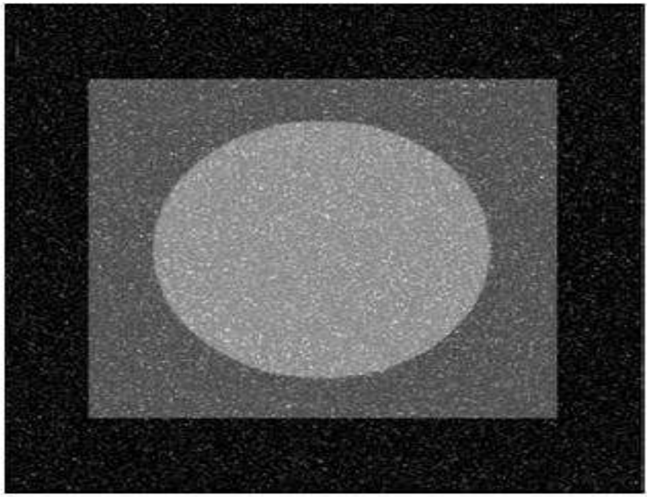
Note that this PDF is a special case of the Erlang PDF, with $b = 1$. Figure 5.2(d) shows a plot of this density function.



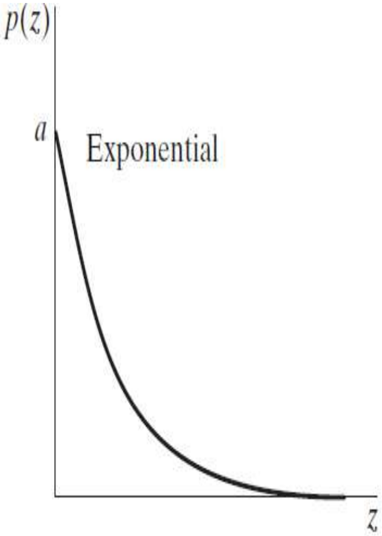
Exponential Noise PDF:

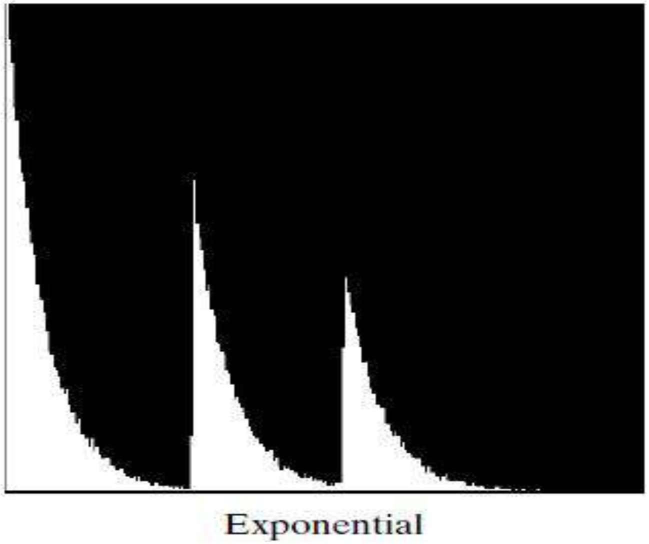
Exponential Noise:

Lets add the Exponential Noise to the Input Image shown in the SLide 5, see the output below along with the Histogram PDF function. Right the Exponential PDF studied in the previous Slide. **Do they Look Similar?**

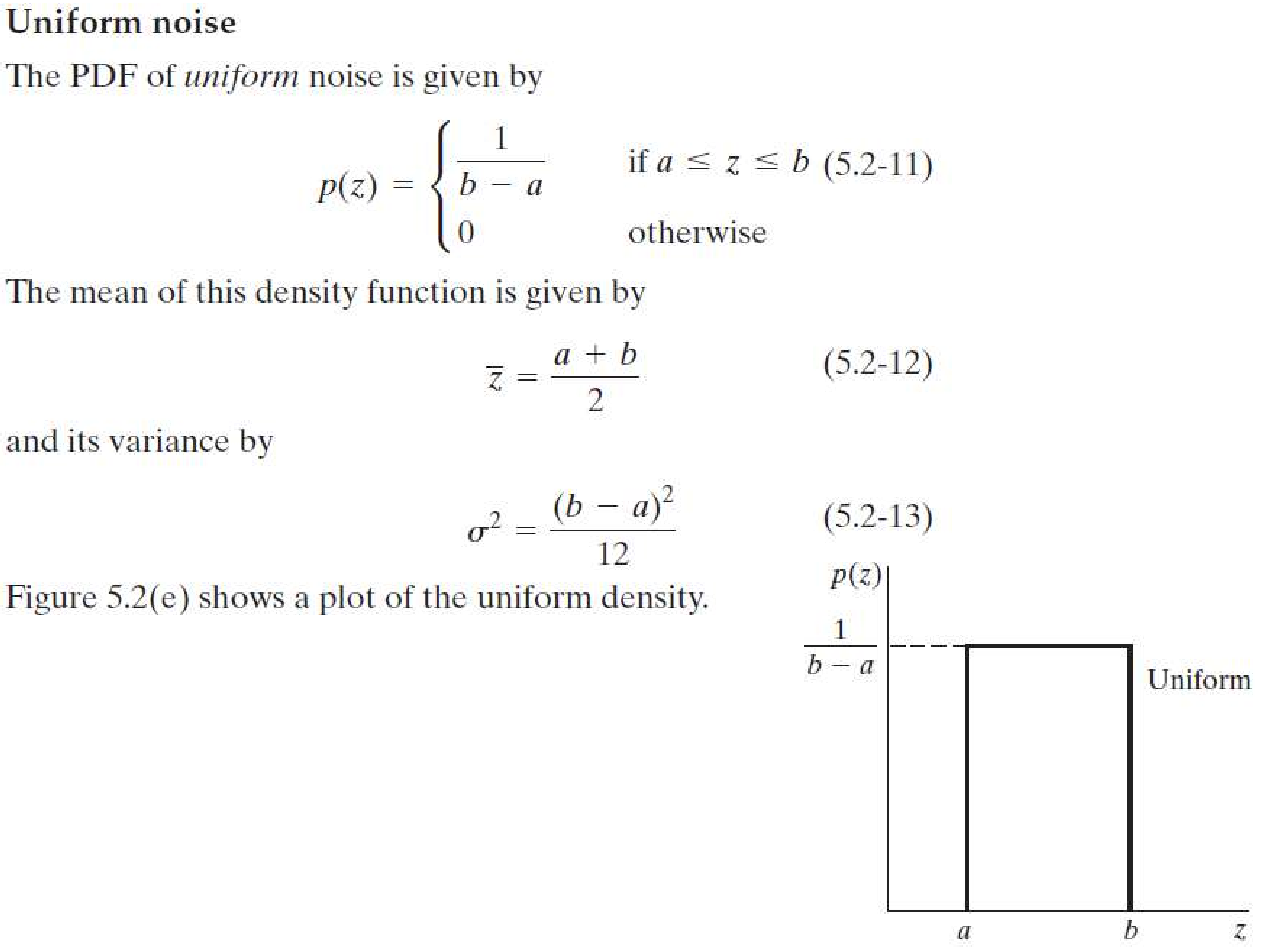


The exponential and gamma densities find application in laser imaging.





Uniform Noise PDF:



Uniform noise

The PDF of *uniform* noise is given by

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

The mean of this density function is given by

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

and its variance by

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

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

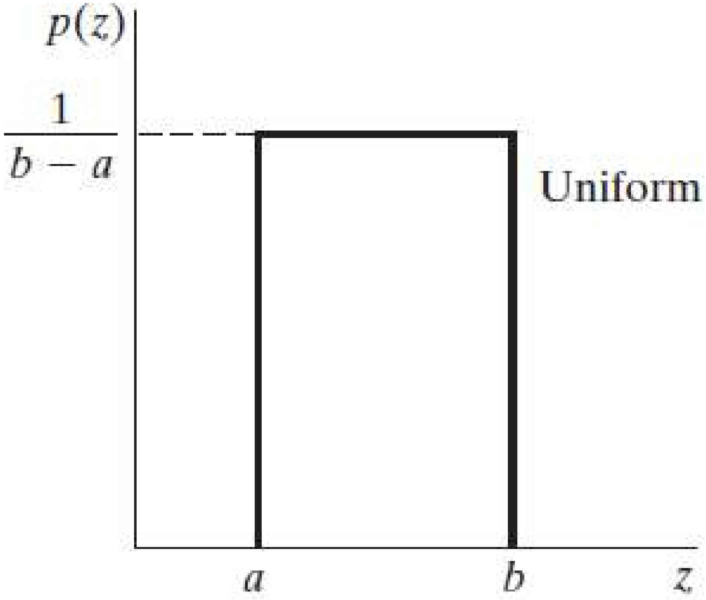
$p(z)$
$\frac{1}{b - a}$
Uniform
a
b
z

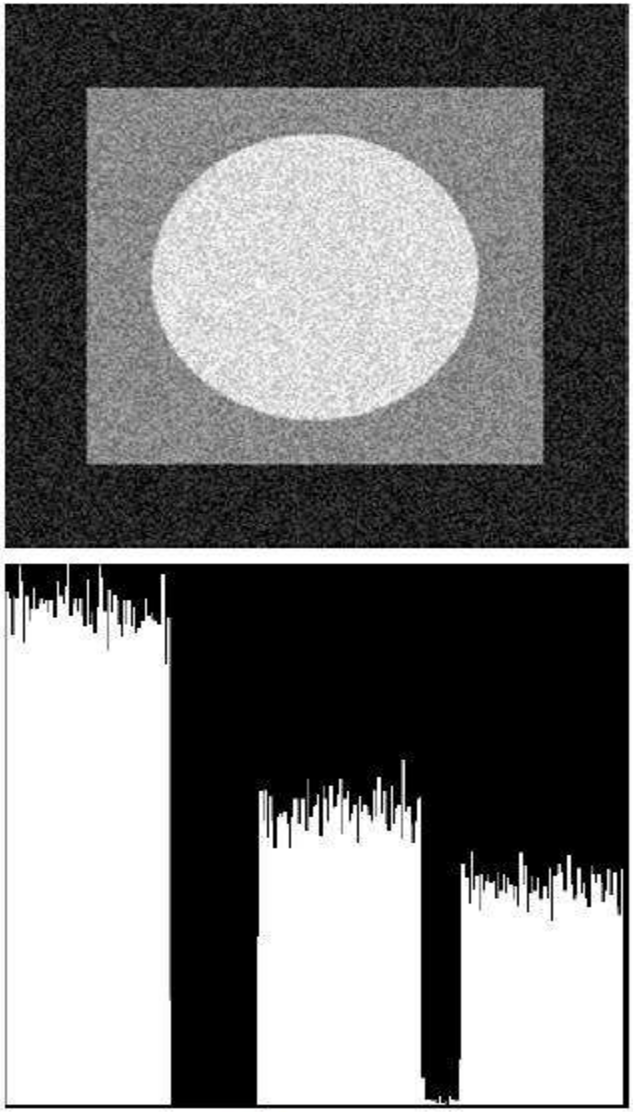
Uniform Noise PDF:

Uniform Noise:

Lets add the Uniform Noise to the Input Image shown in the SLide 5, see the output below along with the Histogram PDF function. Right the Uniform PDF studied in the previous Slide. **Do they Look Similar?**

the uniform density is quite useful as the basis for numerous random number generators that are used in simulations





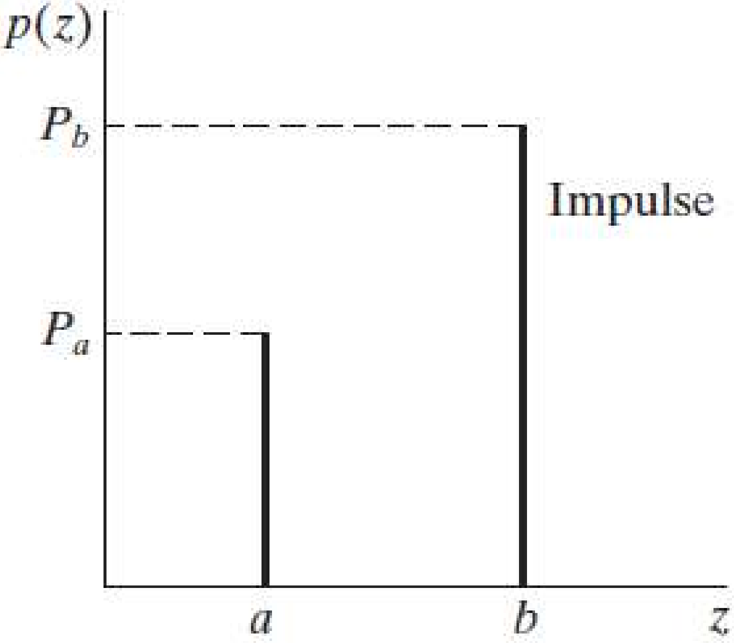
Uniform

Impulse Noise PDF:

Impulse (salt-and-pepper) noise

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

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



If a>b intensity *b will appear as a light dot in the image. Conversely, level a will* appear like a dark dot.

If either P_a or P_b is zero, the impulse noise is called *unipolar.*

If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules randomly distributed over the image. For this reason, bipolar impulse noise also is called *salt & pepper noise.*

Impulse noise is found in situations where quick transients, such as faulty switching, take place during imaging,

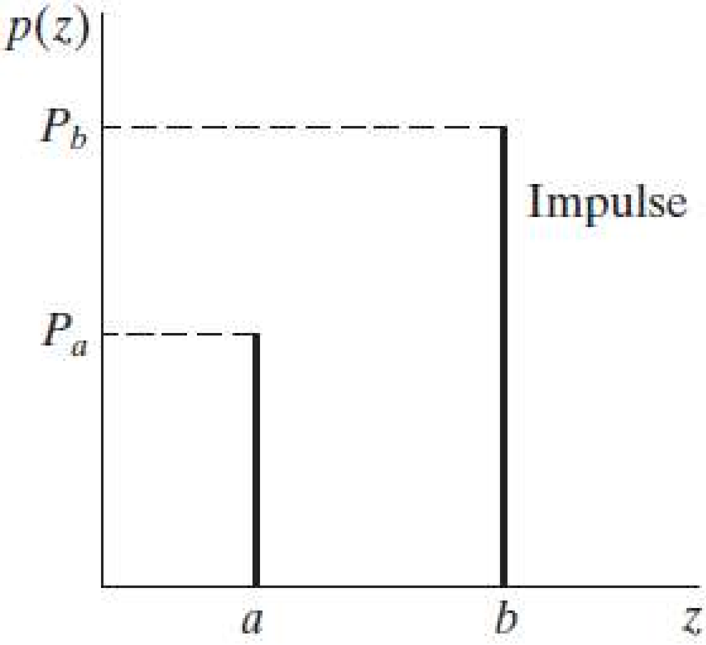
Impulse Noise

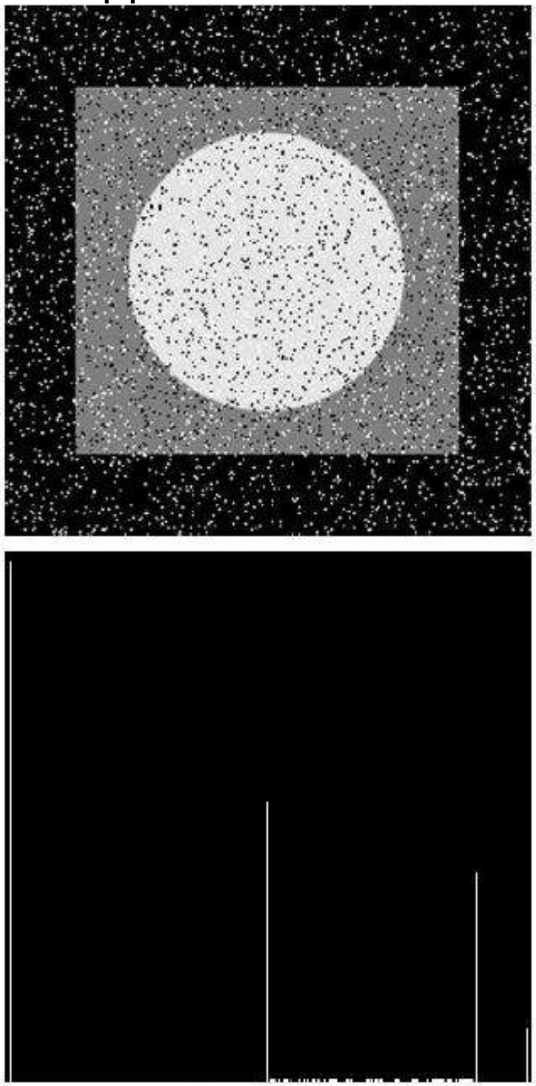
Data-drop-out and spike noise also are terms used to refer to this type of noise. We use the terms *impulse or salt-and-pepper noise interchangeably.*
a and b are "saturated" values, in the sense that they are equal to the minimum and maximum allowed values in the digitized image. For an 8-bit image this means a=0 and b = 255 typically that (black) and (white). As a result, negative impulses appear as black (pepper) points in an image. For the same reason, positive impulses appear as white (salt) noise.

Salt & Pepper Noise PDF:

Salt & Pepper Noise:

Lets add the Salt & Pepper Noise to the Input Image shown in the SLide 5, see the output below along with the Histogram PDF function. Right the Salt & Pepper PDF studied in the previous Slide. **Do they Look Similar?**



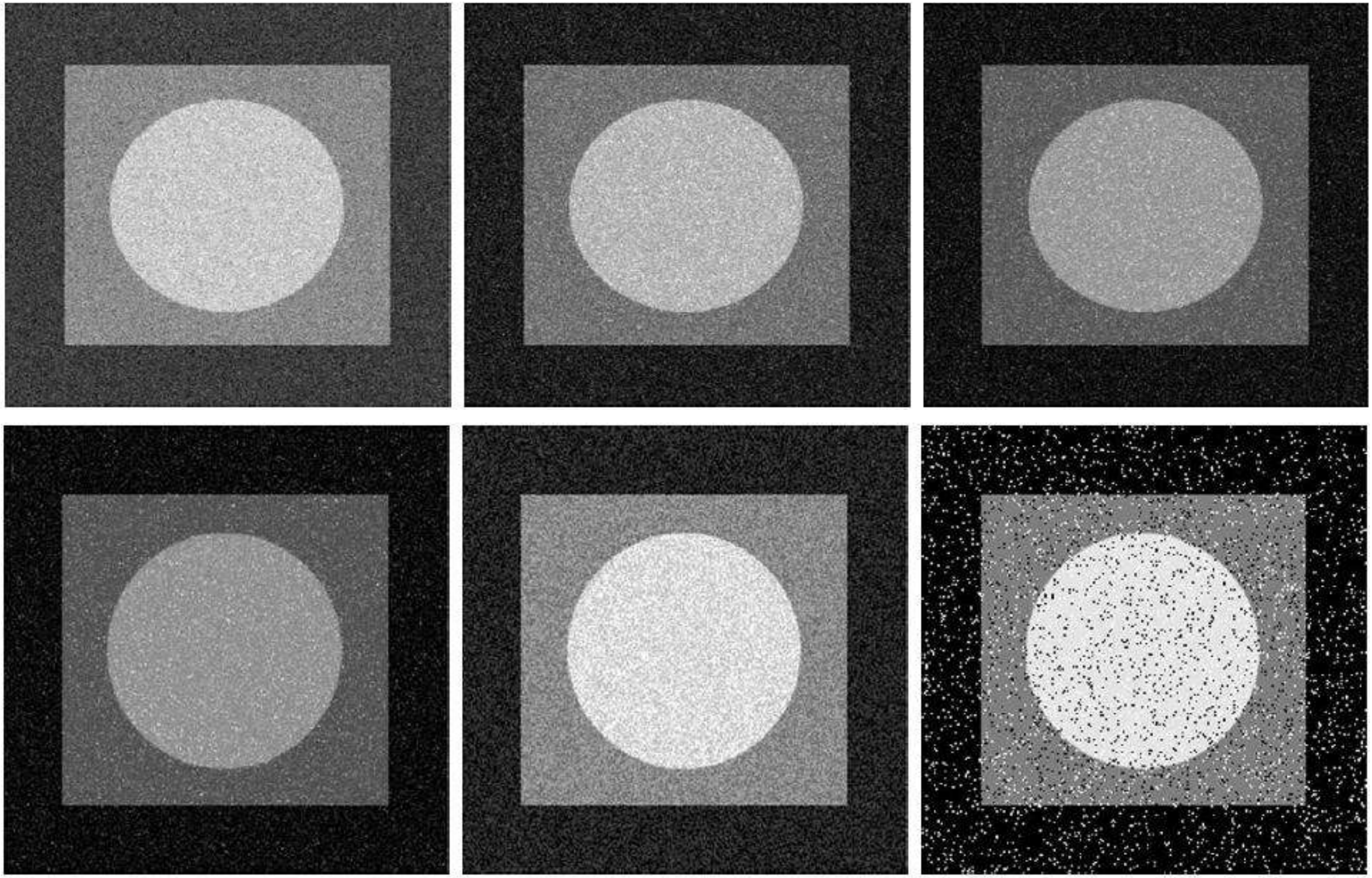


Salt & Pepper

Further analysis of Noise types on Images

What type of Noise is caused the output imgaes degradation below can you guess?

It is difficult to guess so only way to find is apply all types of De-noising techniques and see which is yields best resutls



Periodic Noise (Which can be solved in Frequency Domain)

Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition (Spatial dependant noise). Periodic noise can be reduced significantly via frequency domain filtering. This image is severely corrupted by **sinusoidal noise of various frequencies**.

The Fourier transform of a pure sinusoid is a pair of conjugate impulses located at the conjugate frequencies of the sine wave. Thus, if the amplitude of a sine wave in the spatial domain is strong enough, we would expect to see in the spectrum of the image a pair of impulses for each sine wave in the image. As shown in Fig. 5.5(b), this is indeed the case, with the impulses appearing in an approximate circle because the frequency values in this particular case are so arranged.

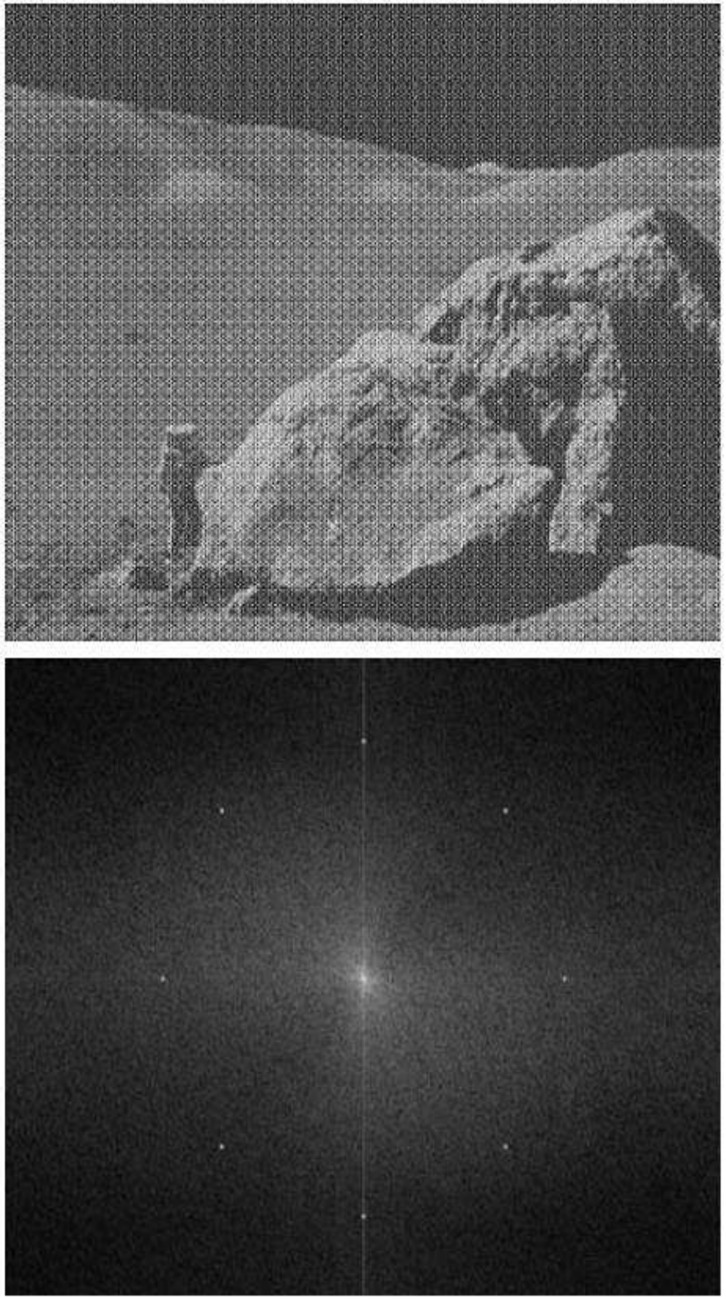


FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

Estimation of Noise Parameters

If you know the imaging arrangements where you capture Images , how to get Noise Parameters?

If you know sensor specifications then you get approximately from them.
Else?

If the imaging system is available, but sensor documents not available?
one simple way to study the characteristics of system noise is to capture a set of images of “flat” environments. For example, in the case of an optical sensor, this is as simple as imaging a solid gray board that is illuminated uniformly. The resulting images typically are good indicators of system noise.

If you have only Images but don’t have any information about imaging arrangements then?

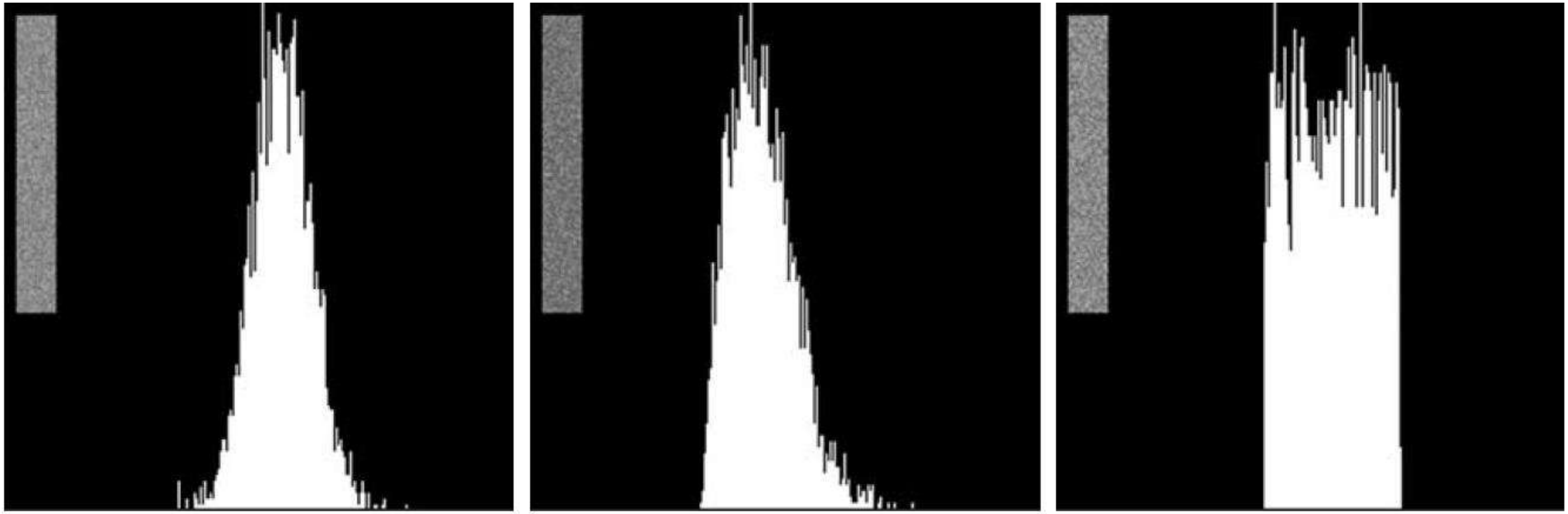
- ➢Take Local histogram of the Input Image patch (sub-image)
- ➢Compare the Local histogram with Global histogram, if they match then you know what type of Noise is majorly influenced.
- ➢ Estimate the Noise parameters of the PDF from small patches of reasonably constant background intensity

Estimation of Noise Parameters

For example, the vertical strips (of 140X20 pixels) shown in below Fig. were cropped from the Gaussian, Rayleigh, and uniform images. The histograms shown were calculated using image data from these small strips.

What is your Analysis?

The shapes of these histograms correspond quite closely to the shapes of the histograms of known Noise Types? Gaussian, Reyleigh, Uniform.



Estimation of Noise Parameters

$$\overline{z} = \sum_{i=0}^{L-1} z_i p_S(z_i) \qquad (5.2\text{-}15)$$

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

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

The shape of the histogram identifies the closest PDF match.

Gaussian type of Noise:

If the shape is approximately Gaussian, then the mean and variance are all we need because the Gaussian PDF is completely specified by these two parameters.

For the other shapes of Noise: use the mean and variance to solve for the parameters a and b.

For Impulse Noise: the estimate needed is of the actual probability of occurrence of white and black pixels. Obtaining this estimate requires that both black and white pixels be visible, so a midgray, relatively constant area is needed in the image in order to be able to compute a histogram. The heights of the peaks corresponding to black and white pixels are the estimates of P_a and P_b

Restoration in the presense of Noise (only spatial Filtering)

When the only degradation present in an image is noise Only then Old Degraded euations of image are given as:

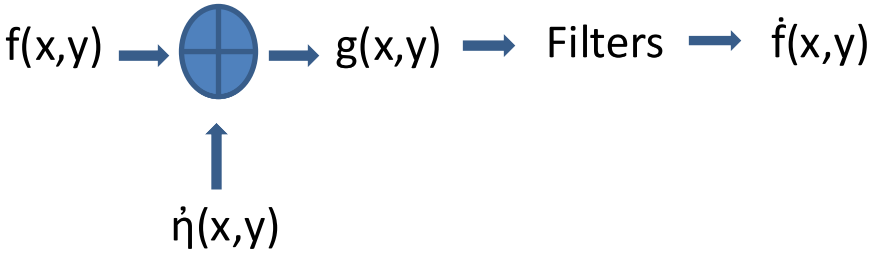
$g(x, y) = f(x, y) + \acute{\eta}(x, y)$ ← There is no degraded Function but has only Noise.

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

IN REALITY THIS IS NOT CASE

The noise terms are unknown, so subtracting them from g(x, y) or G(u, v) is not a realistic option.

In the case of periodic noise, it usually is possible to estimate from the spectrum of G(u, v), In this case N(u, v) can be subtracted from G(u, v) to obtain an estimate of the original image. In general, however, this type of knowledge is the exception, rather than the rule.



Restoration in the presense of Noise using Spatial Filters

- Mean Filters
  - Arithmatic Mean Filters
  - Geomatric Mean Filters
  - Harmonic Mean Filters
  - Contra Harmonic Mean Filters
- Order-Statistics Filters
  - Median Filters
  - Max – Min Filters
  - Mid Point Filters
  - Alpha Trimmed Mean Filters
- Adaptive Filters
  - Adaptive, local noise Filters
  - Adaptive median Filters

Arithmatic Mean Filter

TYPE OF NOISE: Spatial filtering is the method of choice in situations when only additive random noise is present.

PROCEDURE: This is the simplest of the mean filters. Let S_{xy} represent the set of coordinates in a rectangular subimage window (neighborhood) of size m x n, centered at point (*x, y). The arithmetic mean filter computes the average value of the corrupted* image in the area defined by S_{xy}. The value of the restored image ḟ at point (*x, y) is simply the arithmetic mean computed using the pixels in the region* defined by S_{xy}.

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

WHY IT WORKS??

A mean filter smooths local variations in an image, and noise is reduced as a result of blurring

Geometric Filters

TYPE OF NOISE:?

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

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

PROCEDURE: 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.

WHY IT WORKS??

Image comparison

a b
c d

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

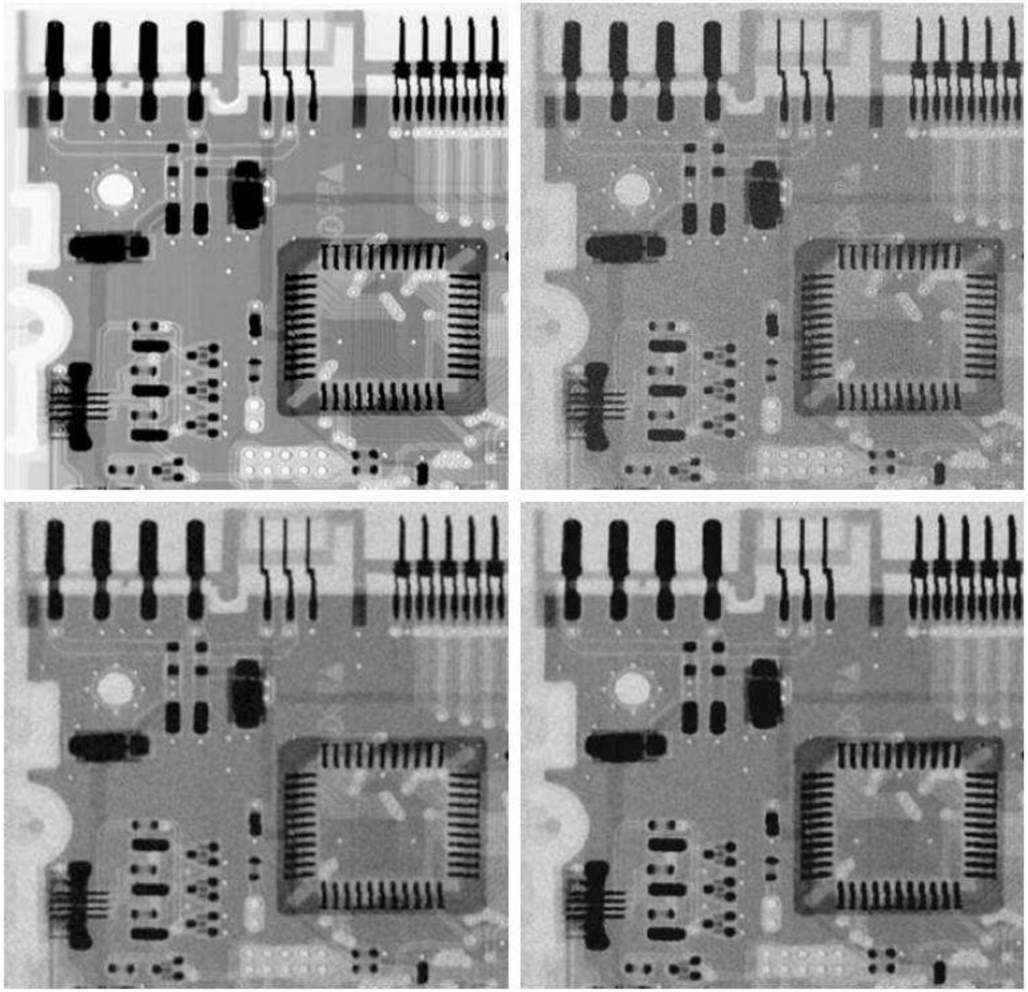


Figure 5.7(a) shows an 8-bit X-ray image of a circuit board, and Fig. 5.7(b) shows the same image, but corrupted with additive Gaussian noise of zero mean and variance of 400. For this type of image this is a significant level of noise. Figures 5.7(c) and (d) show, respectively, the result of filtering the noisy image with an arithmetic mean filter of size 3x3 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.

Harmonic Filters

Harmonic mean filter

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

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

The harmonic mean filter works well for salt noise, but fails for pepper noise.

It does well also with other types of noise like Gaussian noise.

Contraharmonic mean filter

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

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

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

Images Restoration comparison between Salt noise, pepper noise using harmonic Filters

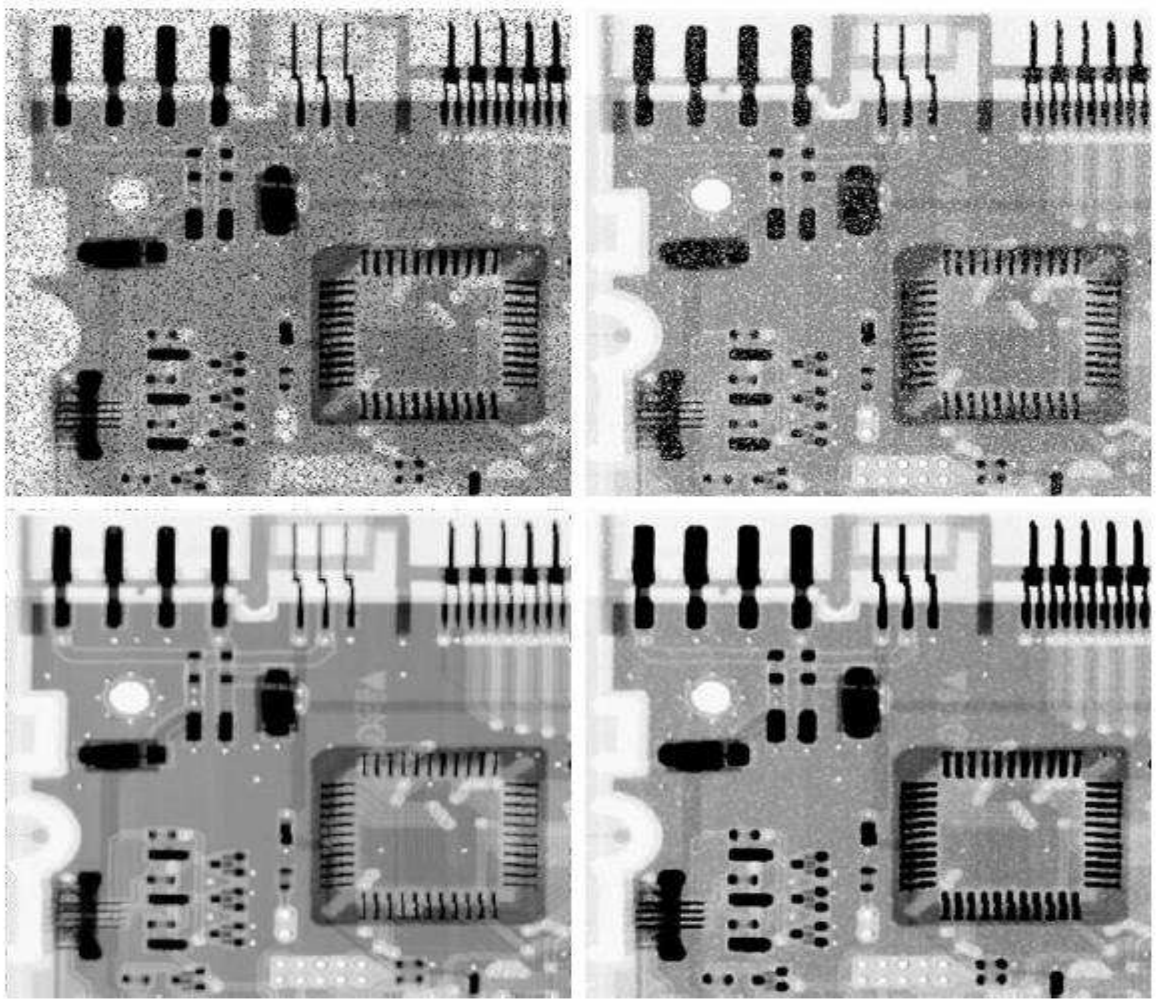


FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contra-harmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

Both filters did a good job in reducing the effect of the noise.The positive-order filter did a better job of cleaning the background, at the expense of slightly thinning and blurring the dark areas.The opposite was true of the negativeorder filter.

Order Statistics Filter (Spatial Filter)

order-statistic filters are spatial filters whose response is based on ordering (ranking) the values of the pixels contained in the image area encompassed by the filter. The ranking result determines the response of the filter.

- Median Filters
- Max – Min Filters
- Mid Point Filters
- Alpha Trimmed Mean Filters

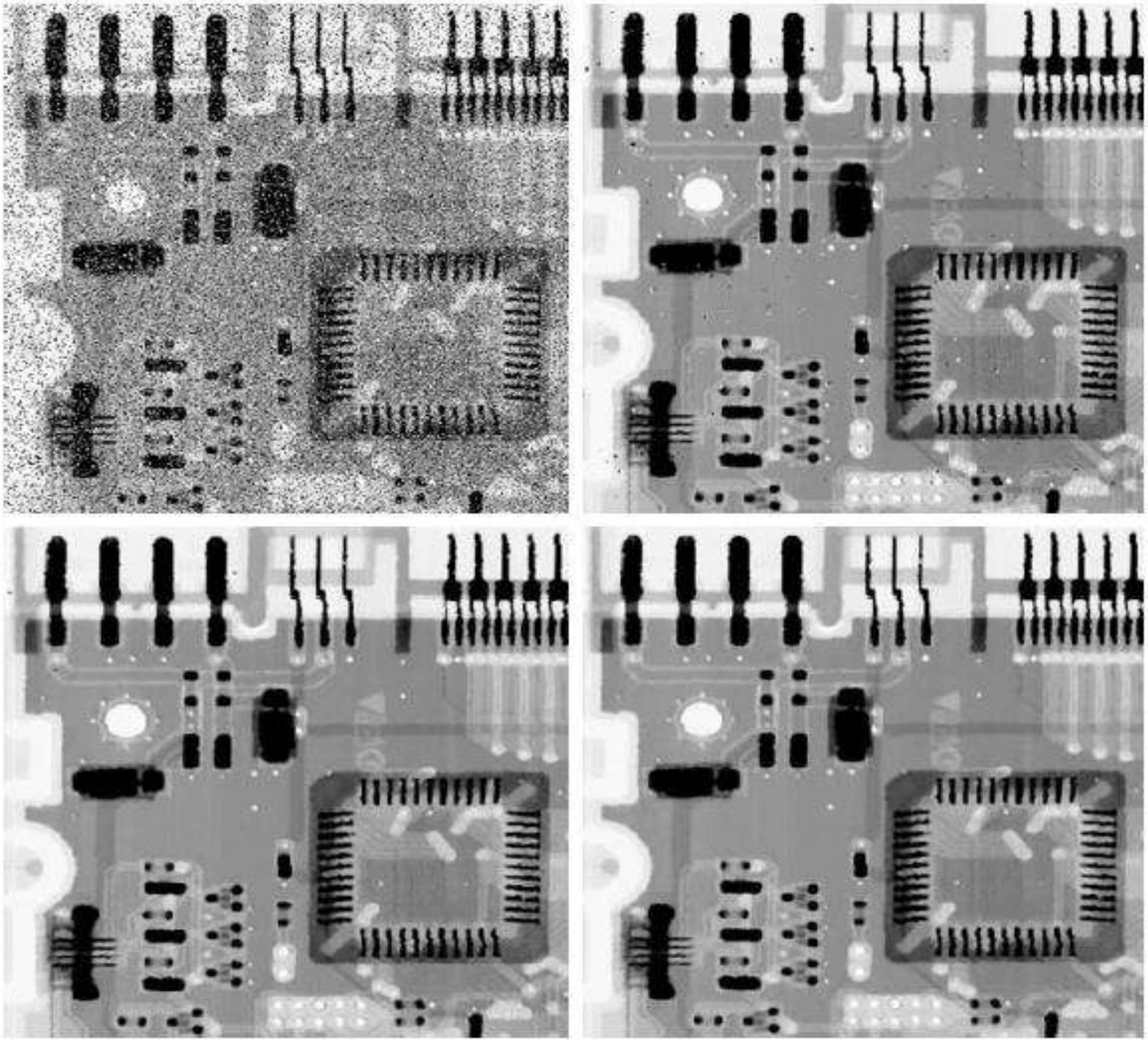
Median filter

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

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

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

Image restoration using Median Filters



a b
c d

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

Order statistics: max-min Filters

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

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

This filter is useful for finding the brightest points in an image.

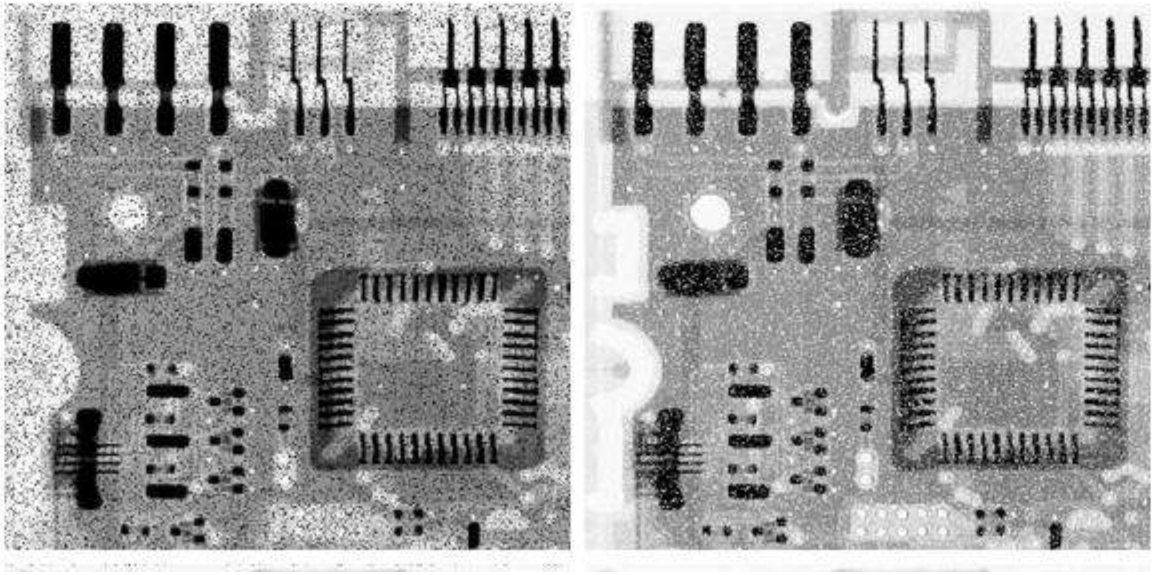
Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area Sxy.

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

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

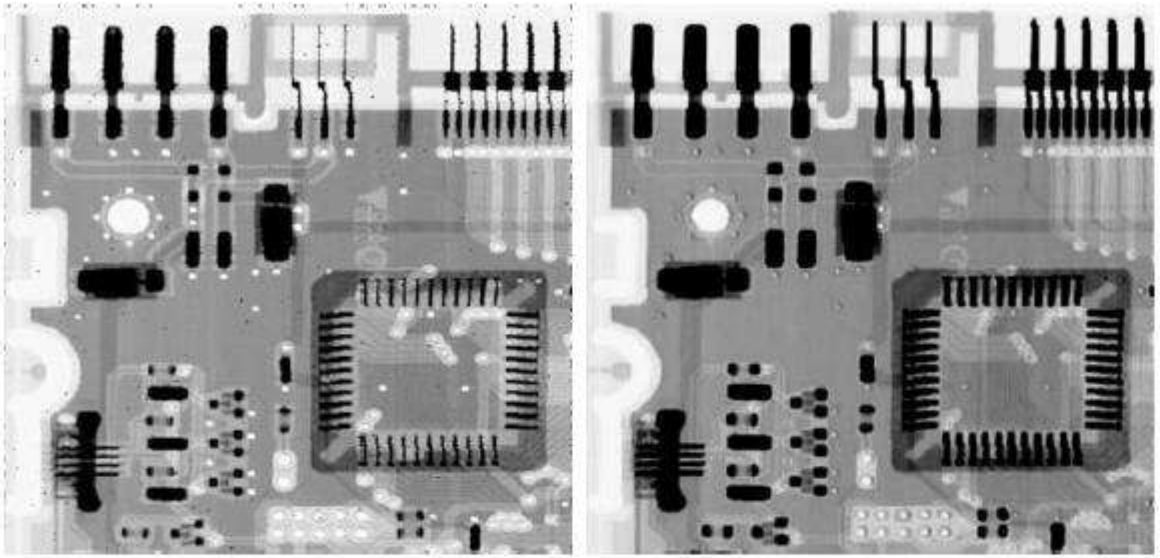
This filter is useful for finding the darkest points in an image. Also, it reduces salt noise as a result of the min operation.

Imagre restoration using max-min filters



a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability.



a b

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

Mid Point Filters

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

Note that this filter combines order statistics and averaging.

It works best for randomly distributed noise, like Gaussian or uniform noise.

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

Alpha-trimmed mean filter

Alpha trimmed Filters

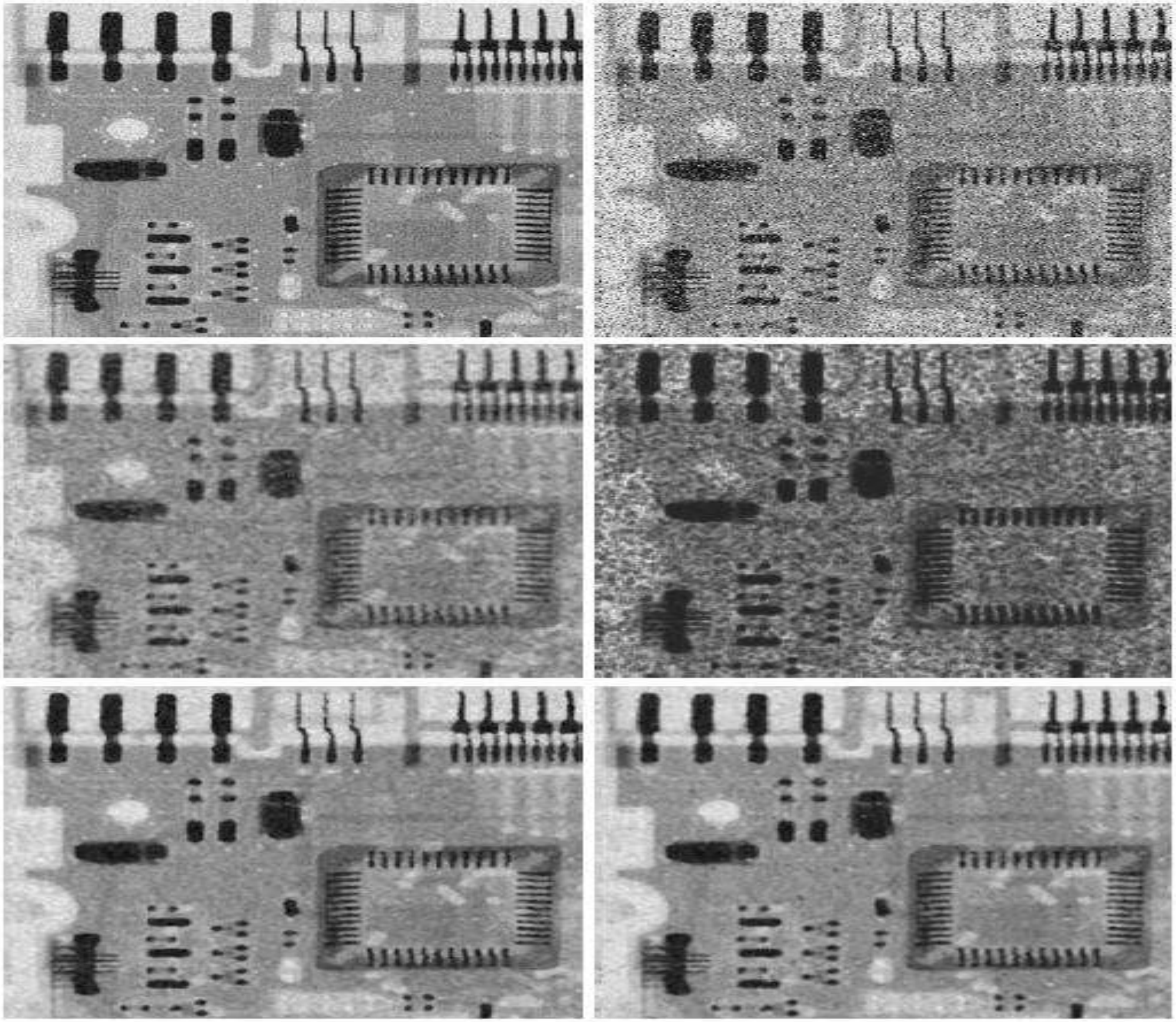
Suppose that we delete the d/2 lowest and the d/2 highest intensity values of g(s, t) in the neighborhood Sxy. Let $g_r(s, t)$ represent the remaining (mn – d) pixels.
A filter formed by averaging these remaining pixels is called an *alpha trimmed mean filter:*

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

where the value of *d can range from 0 to* mn - 1.
When d = 0, *the alphatrimmed* filter reduces to the arithmetic mean filter discussed in the previous
section.
If we choose d = mn - 1, the filter becomes a median filter.

For other values of *d, the alpha-trimmed filter is useful in situations involving multiple* types of noise, such as a combination of salt-and-pepper and Gaussian noise.

Image restoration using Means filters and Median Filter and Alpha trimmed Filter



a b
c d
e f

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

Adaptive Filters

Adaptive filters whose behavior changes based on statistical characteristics of the image.

Filters behavior changes based on statistical characteristics of the image inside the filter region defined by the (m x n) rectangular window $S_{xy.}$
Adaptive filters are capable of performance superior to that of the filters discussed thus far.
The price paid for improved filtering power is an increase in filter complexity.
What is the Mean of Intensity and Variance of intensity means?

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.

Adaptive Local noise reduction filters

The response of the filter at any point (*x, y) on which the region is centered is to be based on four quantities:*

(a) g(x, y), the value of the noisy image at (*x, y);*

(b) $\sigma^2_n$the Global variance of the noise corrupting f(x, y) to form g(x, y);

(c) m_L The local mean of the pixels in S_{xy}. and

*(d) σ^2_L*The local variance of the pixels in S_{xy}.

The behavior of the filter to be as follows:

1. If σ^2_n is zero, the filter should return simply the value of g(x, y)**. This is the** trivial, zero-noise case in which g(x, y) is equal to f(x, y).

2. If the local variance is high relative to the σ^2_n , filter should return a value close to *g(x, y).A high local variance typically is associated with edges, and* these should be preserved.

3. If the two variances are equal, we want the filter to return the arithmetic mean value of the pixels in Sxy. This condition occurs when the local area has the same properties as the overall image, and local noise is to be reduced simply by averaging.

An adaptive expression for obtaining based on these assumptions may be written as:

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

Adaptive Mean Filters

Adaptive median filtering can handle impulse noise with probabilities p_a and P_b larger than 0.2.

Preserve detail while smoothing nonimpulse noise

Increases the size of window S_{xy} during filter operation, depending on certain conditions listed in this section.

The output of the filter is a single value used to replace the value of the pixel at (x, y), the point on which the window S_{xy} is centered at a given time

Adaptive Mean Filters Algorithm

Consider the following notation:

z_{min} = minimum intensity value in S_{xy}

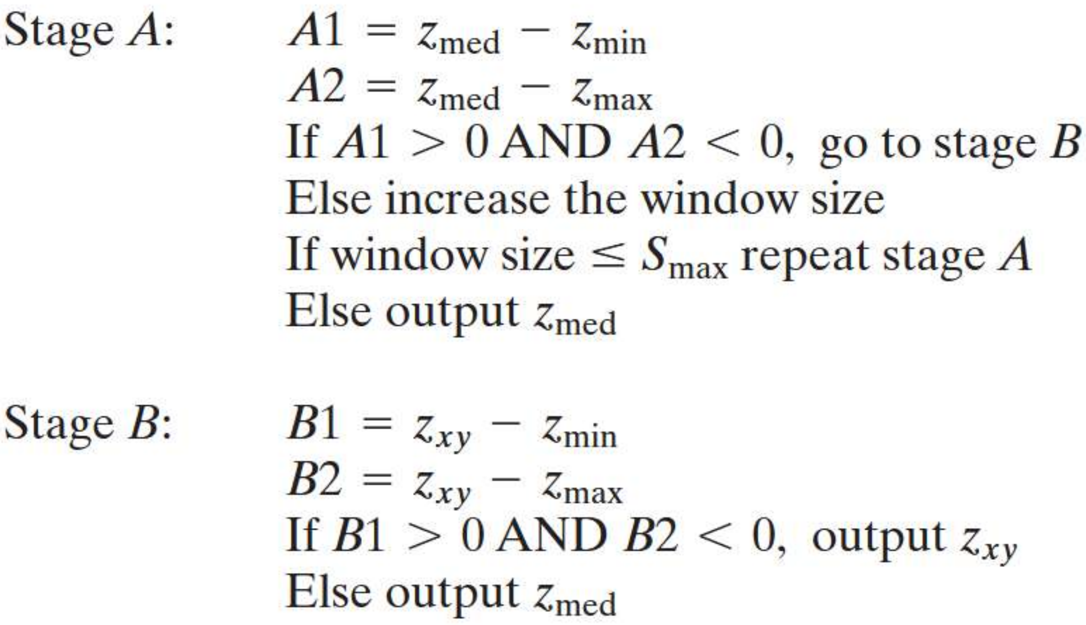
z_{max} = maximum intensity value in S_{xy}

S_{max} = maximum allowed size of S_{xy}

z_{xy} = intensity value at coordinates (x, y)

z_{med} = median of intensity values in S_{xy}

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



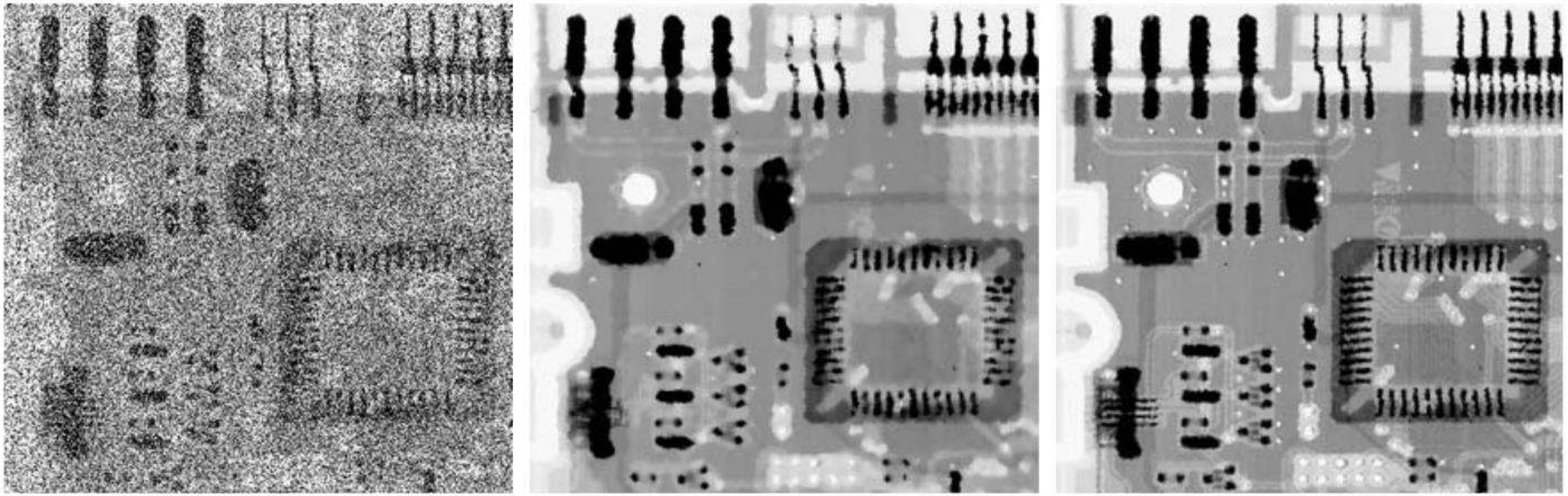
```
Stage $A$:    $A1 = z_{\text{med}} - z_{\text{min}}$
              $A2 = z_{\text{med}} - z_{\text{max}}$
              If $A1 > 0$ AND $A2 < 0$, go to stage $B$
              Else increase the window size
              If window size $\leq S_{\text{max}}$ repeat stage $A$
              Else output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

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 $z_{\text{med}}$
```

Adaptive Mean Filters Algorithm

the purpose of stage A is to determine if the median filter output, is an impulse (black or white) or not.

In stage B test to see if the point in the center of the window, is itself an impulse.

Suppose that stage A does find an impulse (i.e., it fails the test that would cause it to branch to stage B). The algorithm then increases the size of the window and repeats stage A. This looping continues until the algorithm either finds a median value that is not an impulse (and branches to stage B), or the maximum window size is reached.



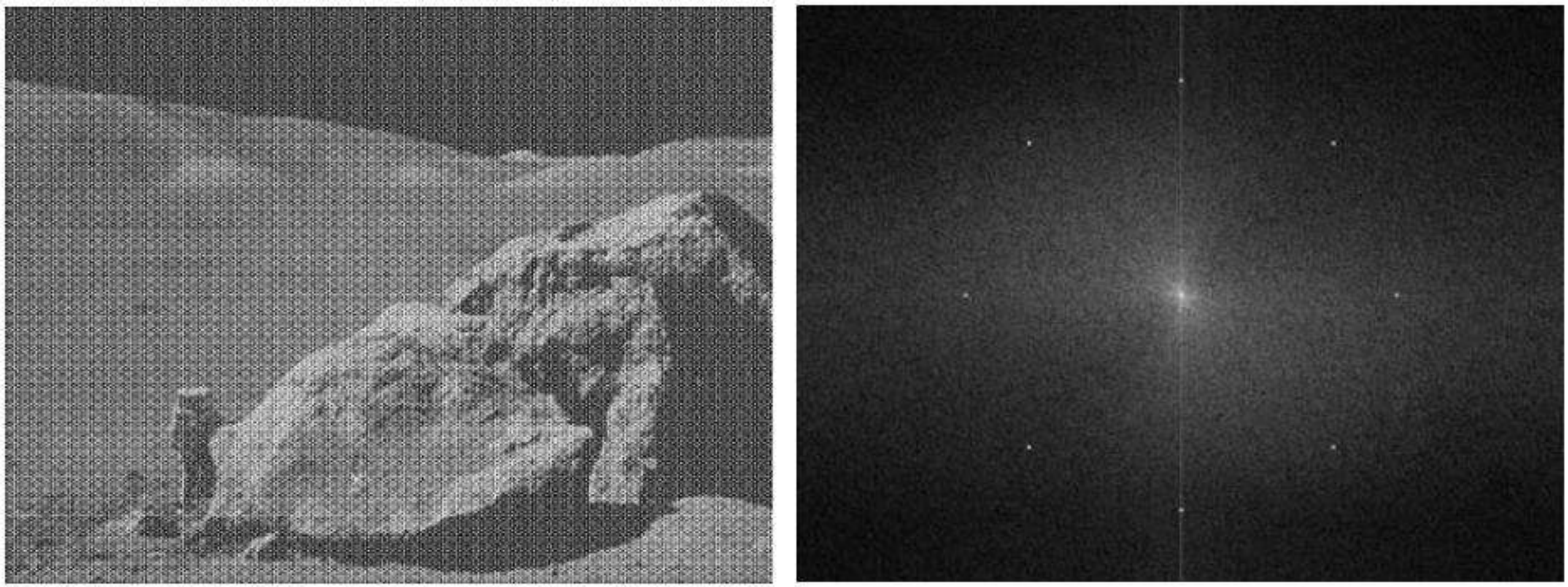
a b c

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 $S_{\max} = 7$.

Periodic Noise

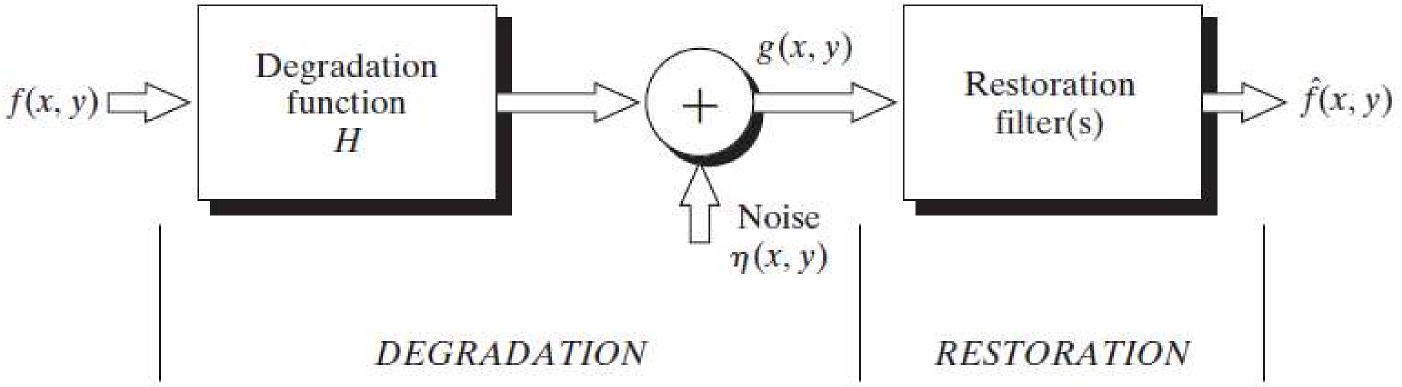
Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition. periodic noise can be reduced significantly via frequency domain filtering. consider the image in Fig. 5.5(a). This image is severely corrupted by (spatial) sinusoidal noise of various frequencies. The Fourier transform of a pure sinusoid is a pair of conjugate impulses† located at the conjugate frequencies of the sine wave. Thus, if the amplitude of a sine wave in the spatial domain is strong enough, we would expect to see in the spectrum of the image a pair of impulses for each sine wave in the image. As shown in Fig. 5.5(b), this is indeed the case, with the impulses appearing in an approximate circle because the frequency values in this particular case are so arranged.

Spectrum on Right shows that the Each pair of conjugate impulses corresponds to each sine wave



NOISE Models of Digital Photos

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



Spatial Domain:
g(x,y) = f(x,y) * h(x,y) + η̇(x,y)
f*h →f convolve with h

f(x,y): Input Image
Degradation stage:
H: Degradation Function that reduces quality of the image f(x,y)
Noise: **η̇(x,y**) [Read **η̇** as neu]: Noise Function that added
g(x,y): Total Low Grade or Corrupted Image
Restoration Filter(s): Filters that can apply on Images
Objective of Restoration stage is:
$\hat{f}$(x,y): f-cap of (x,y) : Estimated Image or restored Image

Frequency Domain:
G(u,v) = F(u,v) H(u,v) + η̇(u,v)
Convolve → Multiplication

If we Know the Degradation function and Noise that added to input Image, we can find corresponding Restoration Filter to estiamate Image restoration.

Image/Photos degradation causes

- Atmospheric turbulances
- Cosmic turbulances
- Aperture
- Hand shake while capturing image/photo

Image restoration

- Blind Deconvolution
- Estimation by Experimention

Basic Terminologies

De-convolution: Restoration seeks to find filters that apply the Convolution process in reverse called De-convolution.

Image Deconvolution: linear image restoration process is called Image deconvolution.

The filters used in the restoration process often are called deconvolution filters.

How to estimate the Input Image Degradation?

Three principal ways to estimate the degradation function for use in image restoration:

(1) observation,

(2) experimentation, and

(3) mathematical modeling.

Blind Deconvolution

The process of restoring an image by using a degradation function that has been estimated in some ways some times called “Blind DeConvolution” because we don’t know the degradation function that cause input image to degrade.

All the above are fall in the same category.

1.) Estimation of Degradation function H() by Image Observation

Intution behind this process:

Based on the assumption that the image was degraded by a linear, position-invariant process, one way to estimate H is to gather information from the image itself.

For example: If the image is blurred, we can look at a small rectangular section of the image containing sample structures, like part of an object and the background. In order to reduce the effect of noise, we would look for an area in which the signal content is strong (e.g., an area of high contrast). The next step would be to process the subimage to arrive at a result that is as unblurred as possible.

We can do this by sharpening the subimage with a sharpening filter and even by processing small areas by hand.

1.) Estimation of Degradation function H() by Image Observation

Step1: Look for areas of strong signal content

Step2: Let the observed subimage be denoted by G_s (x, y)

Simple gray levels of object and background

Let $G_s(x,y)$ is denoted as sub image and

Let constructed Sub image be denoted as $f_s(x,y)$.

Assuming the effect of Noise is negligible (Noise ~=0) because of our choice of strong signal area

Estimate the Original Image in the windows $\dot{F}_s(u,v)$ is given below formula:

$H_s(u,v) = \frac{G_s(u,v)}{\dot{F}_s(u,v)}$

known

unknown

For example, suppose that a radial plot of Hs (u, v) has the approximate shape of a Gaussian curve.We can [illegible]mation to construct a function H(u, v) on a larger scale, but having the same basic shape.

Once we have H then we can construct the $\dot{F}_s(u,v)$ to restore the Image.

LABORIOUS PROCESS

2. Estimation of Degraded Function H() by Experimentation

Intution behind the process:

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 degraded imae can be acquired with various system settings unit they are degraded as closed as possible to the image to which we restore.

Then the idea is to obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the same system settings.

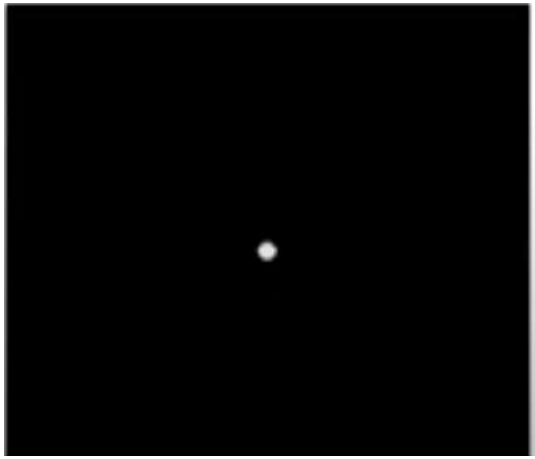
An impulse is simulated by a bright dot of light, as bright as possible to reduce the effect of noise to negligible values.

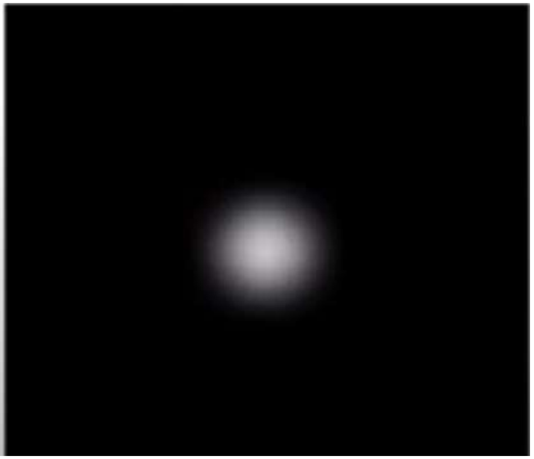
Esitmating the degradation by experimentation

- If the Image acquisition system is ready
- Obtain the Fourier transform of impulse response is a Constant (A).
- H(u,v) = $\frac{G(u,v)}{A}$

where, G(u, v) is the Fourier transform of the observed image and A is a constant describing the strength of the impulse.

- Degradation H(u,v) estimation by Impulse characterization.
- Fig:a) An impulse of light (shown magnified).
- Fig:b) Imaged (degraded)





3. Estimating the Degraded Function H() by Modeling

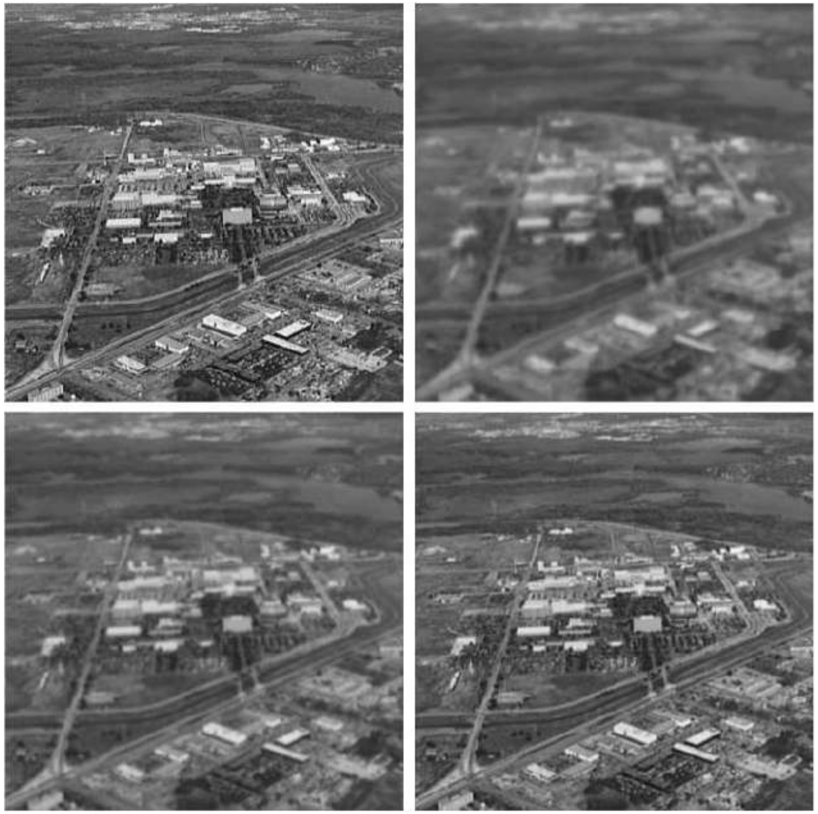
- Degradation modelling has been used because of the insight it affords into the image restoration problem
- In some cases, the model can seen take into account environmental conditions that cause degradation
- Ex: Atomospheric turbulence model which can degrade the Images acquired, this degradation estimatin is given below

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

where *k* is a constant that depends on the nature of the turbulence. With the 5/6th power on the exponent

Esitmation by modeling

Original image is given this image restored by changing values of K for different values given below. We can see different blur images based on values of k. ORIGINAL Image has a blur with k=0.00025.

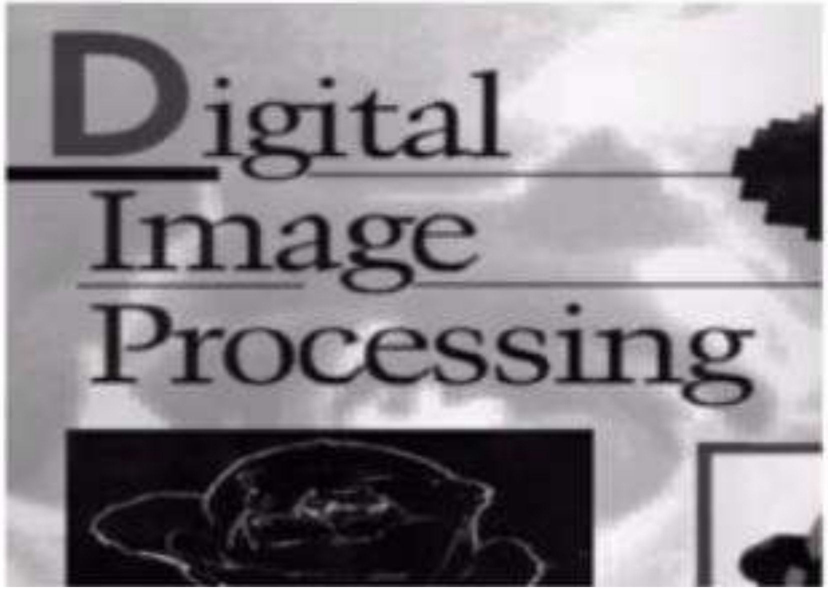


a b
c d

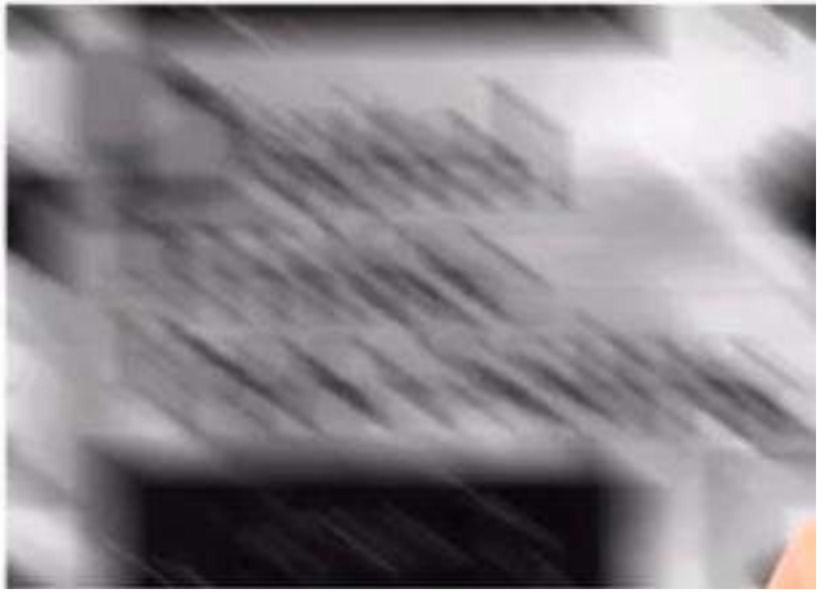
FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

Can you estimate Blur image and restore Blur to Readable image?

Original



Apply motion model



Image/Photo Blur estimation by modelling using Planar motion

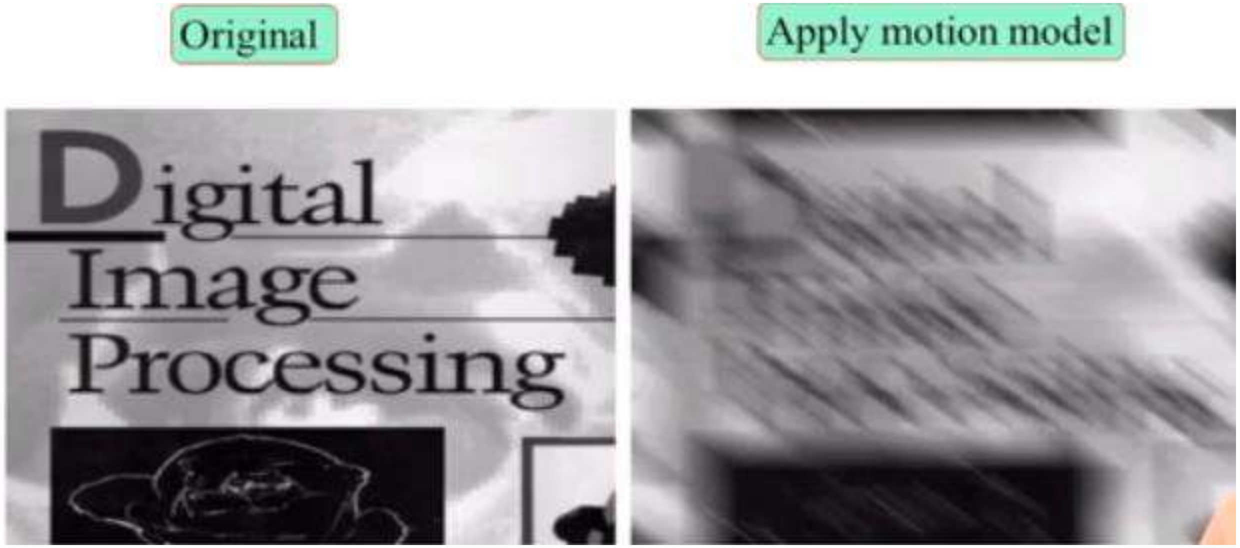
Photo/Image has been blurred by uniform linear motion between the image and the sensor during image acquisition. Suppose that an image f(x,y) undergoes planar motion and that $x_0(t)$ and $y_0(t)$ are the time varying components of motion in the x- and y-directions, respectively. The total exposure at any point of the recording medium (say, film or digital memory) is obtained by integrating the instantaneous exposure over the time interval during which the imaging system shutter is open.

Assuming that shutter opening and closing takes place instantaneously, and that the optical imaging process is perfect, isolates the effect of image motion.

Estimation by modeling (Planar motion)

Then, if T is the duration of the exposure, it follows that Motion of the Image given by:

g(x,y) = $\int_0^T f\big(x - x_0(t) - y - y_0(t)\big)dt$ where g(x, y) is the blurred image.



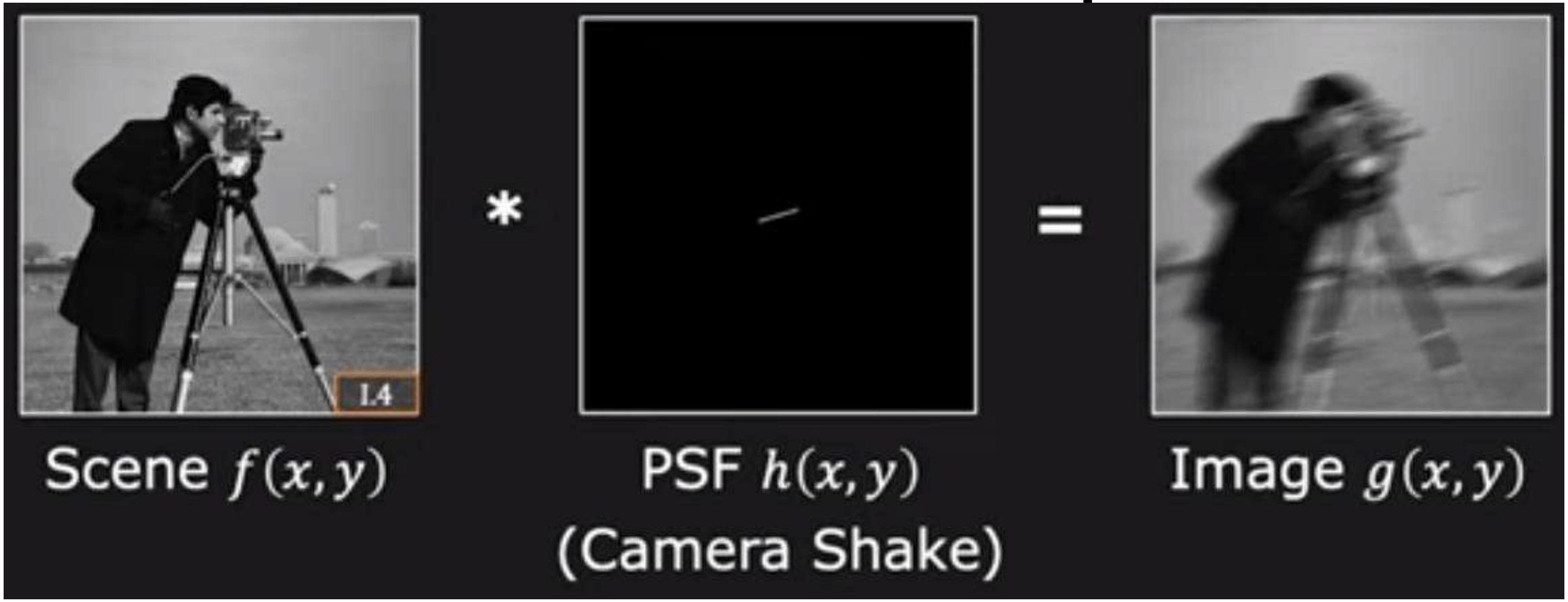
Deconvolution Image processing (Motion Deblur)

Example of Motion Blur: You are holding a camera and press button and a little bit of shake this can cause image is convolve with motion blur.

Example of Motion Blur: You are holding a camera while riding a Bike and press Photo capture button and Bike motion can cause image is convolve with motion blur.

The process of recovering this motion blur is called Deconvolving and this is much easier to do in Fourier domain.

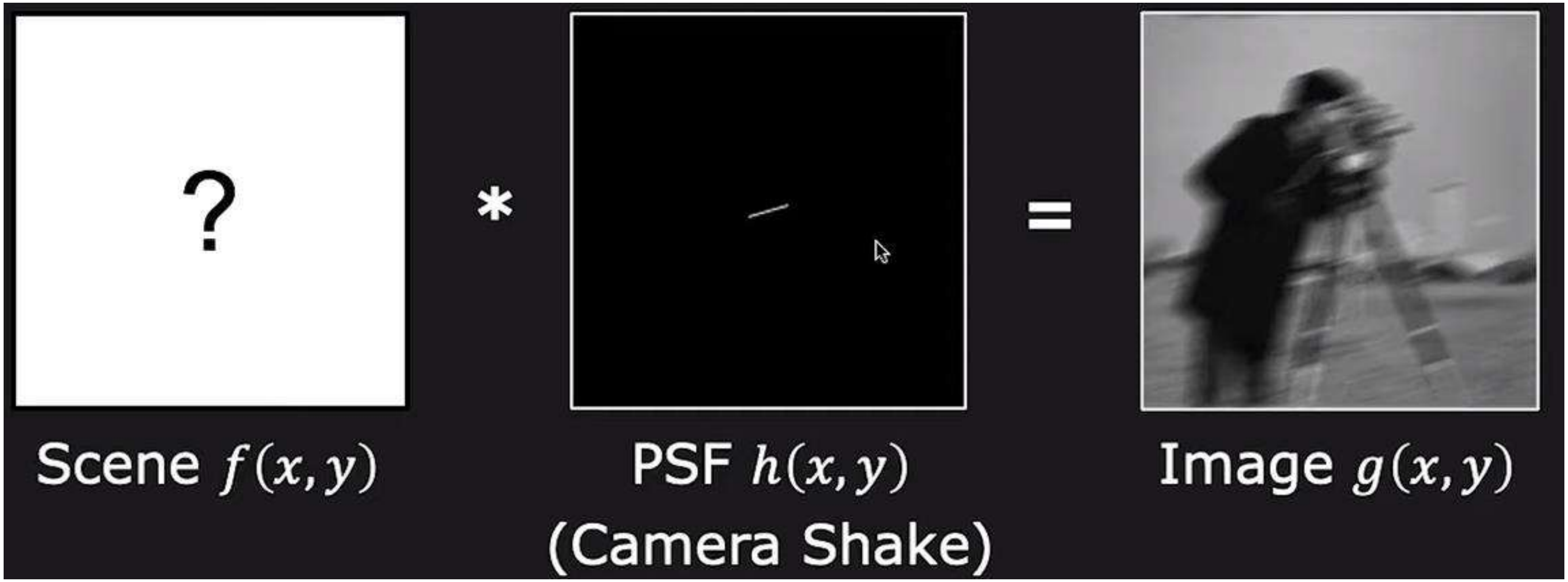
Motion Blur example



You have an Ideal Scene f(x,y) because of motion blur it is convolve with PSF (Point Spread Function) h(x,y).
Output is Smeared Image g(x,y).

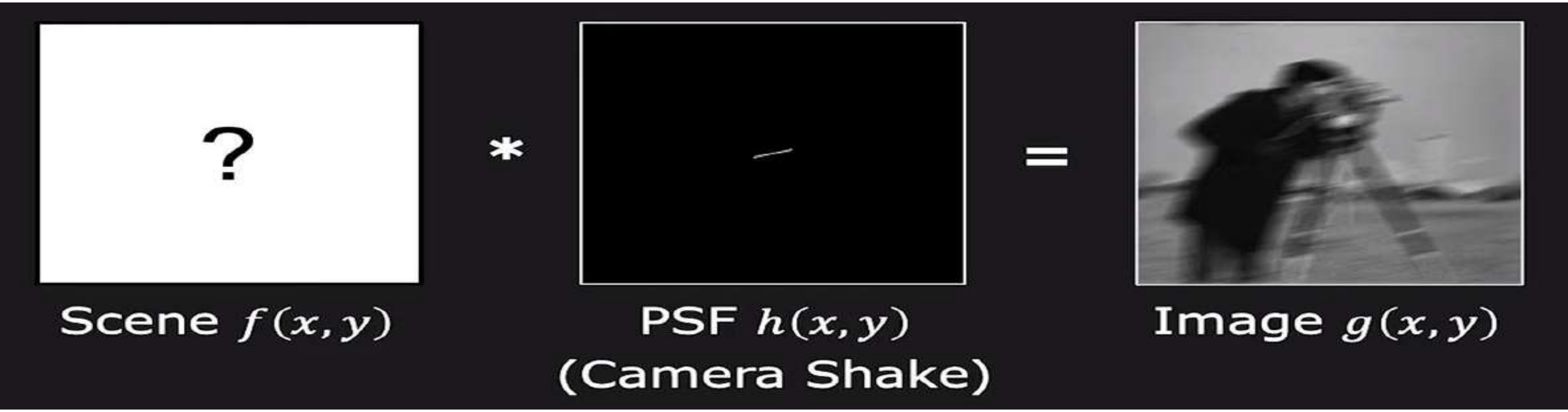
This Motion blur is given as
f(x,y) * h(x,y) = g(x,y) → This is capture image but what you want want is f(x,y)

Problem statement (Motion Blur)

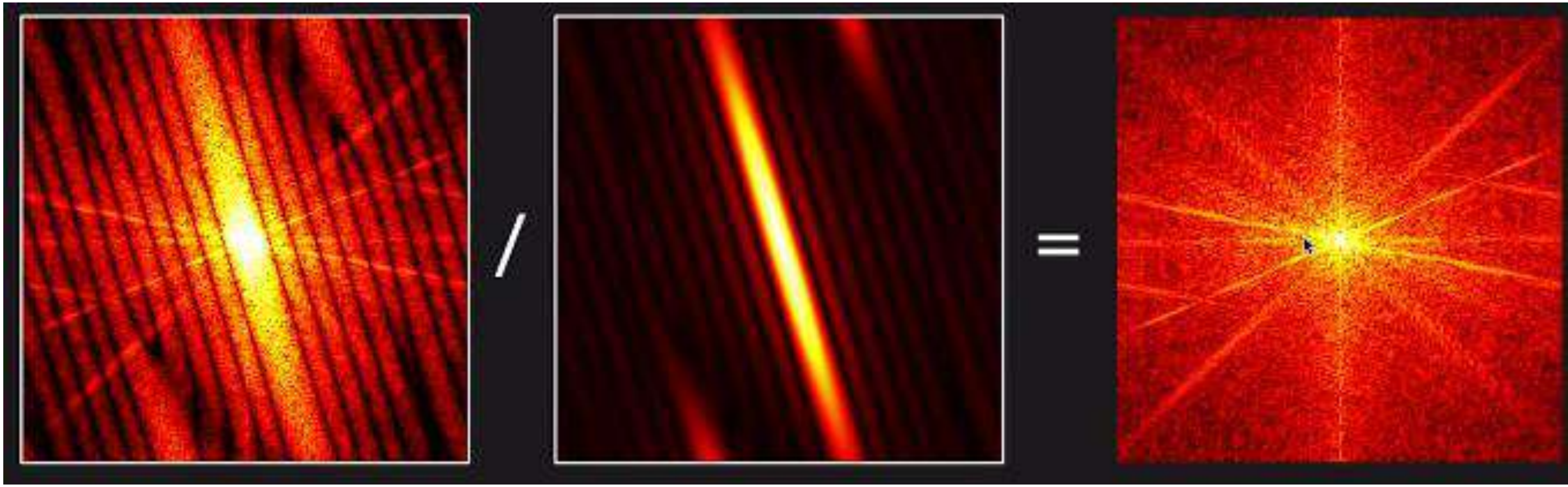


Given captured Image g(x,y) and PSF h(x,y) can we estimate the Actual image f(x,y).
WHERE DOES THIS PSF() Point Spread Fuction comes from?
In you Digital Cameras/Mobile Phones a Small Sensor IMU is embedded in it which has Accelero meters, gyroscope in it so with these when you capture image using camera and the camera motion is captured during click of button given to us. Use that to create PSF.
So we have PSF h(x,y) and g(x,y) with us.

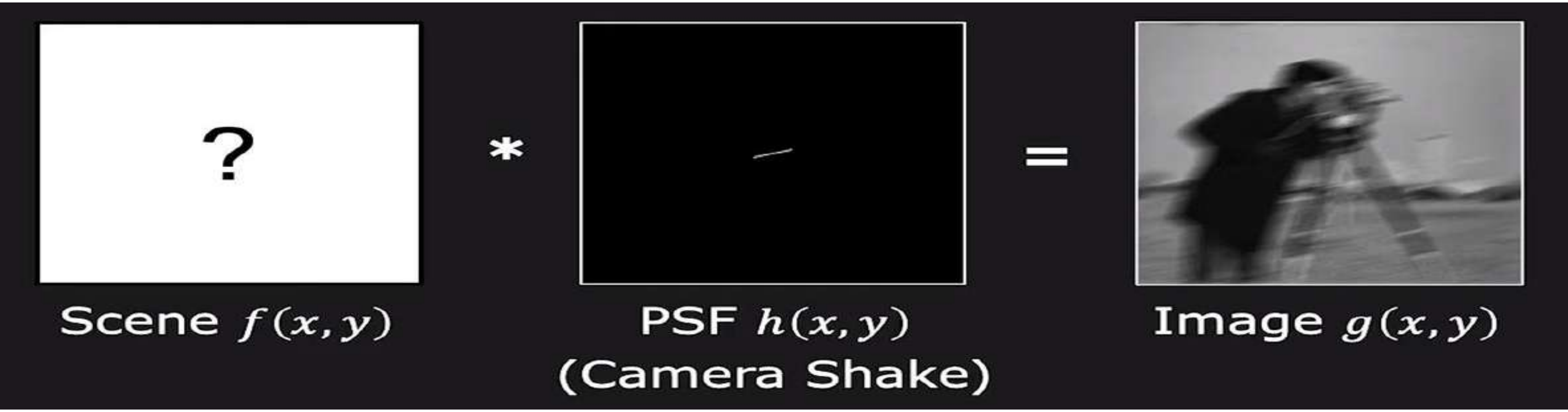
Motion Blur: Deconvolution



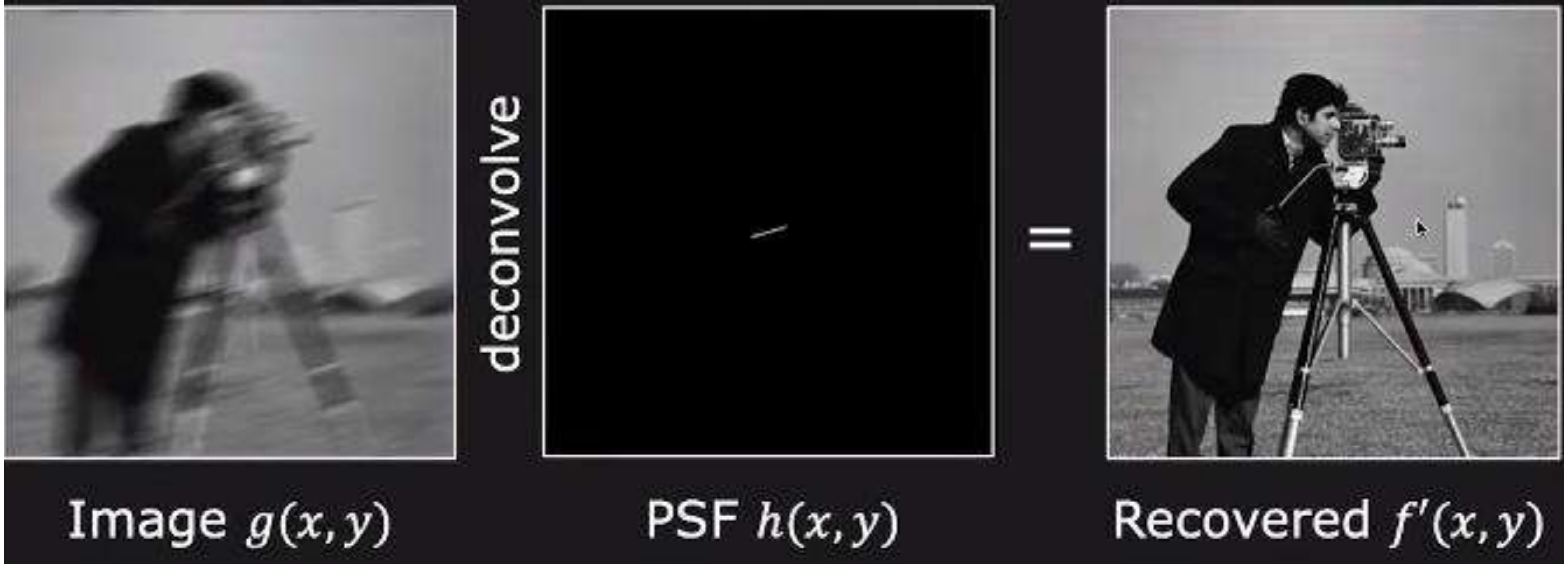
$\dot{F}(u,v) = \frac{1}{H(u,v)}G(u,v)$ ➔ Inverse Fourier Transform ➔ $\dot{f}(x,y)$



Motion Blur: Deconvolution



$\dot{F}(u,v) = \frac{1}{H(u,v)}G(u,v)$ ➔ Inverse Fourier Transform ➔ $\dot{f}(x,y)$



How to remove both Noise N(x,y) and Degradations H(x,y) in input images

Need:

In reality images are disturbed by both Degradation function and Noise. Methods that handles both are discussed from now on.

- The process of removing blurs and noise is known as Deconvolution or Inverse Filtering
- Assumptions:
  - Blur is characterized by PSF or the Impulse response of the Sytems
  - Blurs are Linear and output of Imaging system is the convolution of impulse responses and input Image.

Inverse Filtering

Degradation model in Fourier Transform

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

Lets Assume Noise is N(u,v) = 0

Therefore F(u,v) = $\frac{1}{H(u,v)}$G(u,v)

But in reality restored image is given below

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

If we don’t know Noise N(u,v) or

If Degradation function is small/very Small then we can’t restore a proper image because expression becomes Infinity.

How will you over come this?

Inverse Filtering

Inverse Filter acts as:

High pass Filter causing bluring and Noise

H(u,v) = ~0 or Closer to ZERO

Therefore $\frac{1}{H(u,v)}$ approach to INFINITY

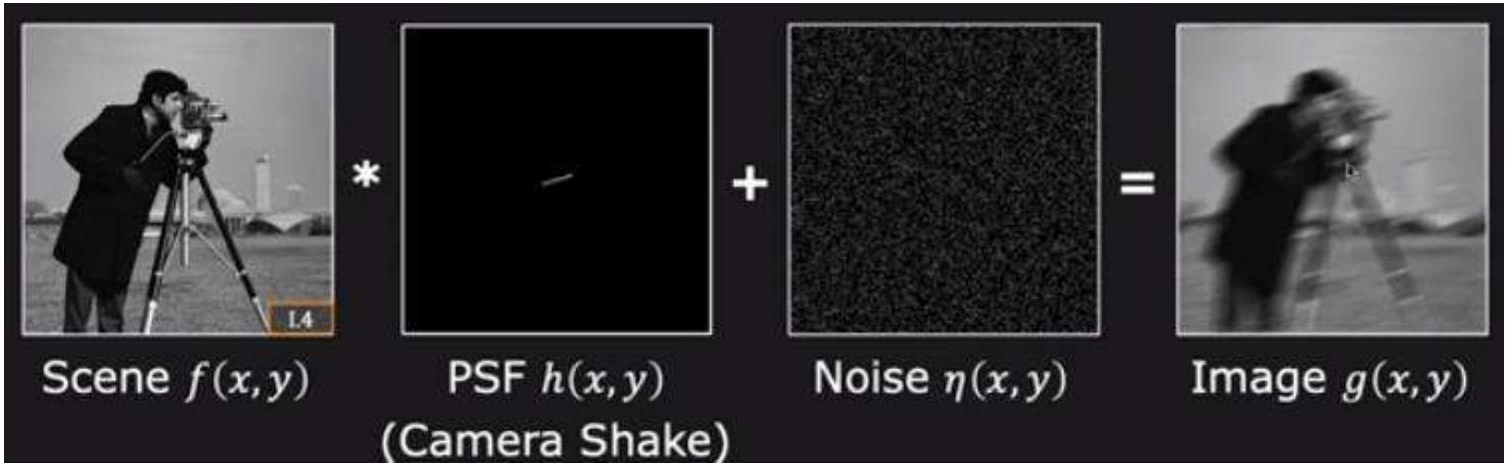
To design the Inverse Filtering we can write as Transfer function

$$\left\{ \hat{H}(u,v) = \begin{cases} \frac{1}{H(u,v)}, & if\ |H(u,v)| >= \mathcal{E} \\ 1\ , & if |H(u,v)| < \ \mathcal{E} \end{cases} \right.$$

Ɛ: This is Threshold Value

This is used in Mitigation in ZERO in Degradation Images

Disadvantages of Inverse Filtering



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

1. Real time: Noise gets added to Images captured in Digital cameras/Mobile Phones
2. $H(u,v) = 0$, $\dot{F}(u,v) = \infty$ → Image is not recoverable
3. Motion blur $H(u,v)$ is Low pass filter

   For High frequencies (u,v) :

   * Noise $N(u,v)$ in $G(u,v)$ is High
   * Filtern $H(u,v) \approx 0$

   Noise in G(u,v) is Amplified

We need some kind of Noise Suppression During Deconvolution , that Brings us WIENR FILTERS

(Image Restoration: WIENER FILTERS [Minimum Mean Square Error Filters]

Intution behind this process

Images/Photos denoising before applying Degradation

Challenges in image Restoration :

Zero Knowledge about Noise

Partial knowledge about Noise Filter

Full Knowledge about Noise

An image which has noise we are aware partially about them, in such cases we use Weiner Filter

(Image Restoration using WIENER FILTERS

N. Wiener proposed the concept of Wiener Filter in 1942

Known as Minimum Mean Square Error Filter

Or

Least square error Filter

(Image Restoration) WIENER FILTERS

Approach:

Wiener Filter incorporates both the degradation Function and statistical characteristics of Noise into the Restoration process

Objective is to find the Estimate *f*` of the uncorrupted image *f* such that Mean square error between them is Minimized

Wiener Filter in Frequency Domain

Restores the Image but also removes noise by smoothing.

The Minimized Error is given by :

e^2= E[$f(x, y)$. Ḟ$(x, y)^2$] where E{*} is expected value of the argument.

To estimate error the correctional Metrices of f and n are required. Let us assume the R$_f$ and R$_n$ are the corresponding matrices of f (input image) and n (noise) respectively.

These correlational metrices are denoted in terms of expectation operation as:

R$_f$ = E{f. f^T} , R $_n$ = E{n. n^T}

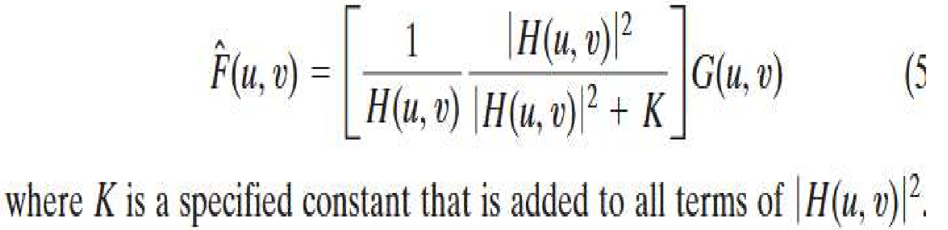
The use of Unconstrained form and substitution of Q+Q = R$_f$$^{-n}$ R^n give the following estimation:

Ḟ = $(H^T * H +$ ƔR$_f$$^{-n}$ $R^n)$ H^{-1}G

G -> Degraded Image

H -> Degradation Function

Ɣ$(gammma)$ = $\frac{1}{\alpha}$ where α is Lagrange Multiplier



$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right]G(u,v) \qquad (5$$

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

Weiner Filter (Frequency Domain)

The Filter in Frequency Domain corresponds to the mentioned Equation:

$$F(u,v) = \frac{1}{H(u,v)}\left[\frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma\left|\frac{s_n(u,v)}{s_f(u,v)}\right|}\right].\ G(u,v)$$

Where $s_n(u,v)$: Spectral power density Noise

$s_f(u,v)$: Spectral power Density of Signal

Weiner Filter (Frequency Domain)

In Some cases when $\frac{s_{n}(u,v)}{s_{f}(u,v)}$ = K (CONSTANT)

The Filter in Frequency Domain corresponds to the mentioned Equation:

F(u,v) = $\frac{1}{H(u,v)}\left[\frac{|H(u,v)|^2}{|H(u,v)|^2+ \gamma\, k}\right]$. G(u,v)

Where $k = Low\ frequency\ aspect\ of\ Filter$

If k = 0 : It will behave as inverse Filter

Value of Ɣ (gamma) should be selected carefully

If Ɣ = 0 then Filter becomes Inverse Filter

If Ɣ = 1 then Filter becomes Wiener Filter.

Other Values it becomes parametric Filter

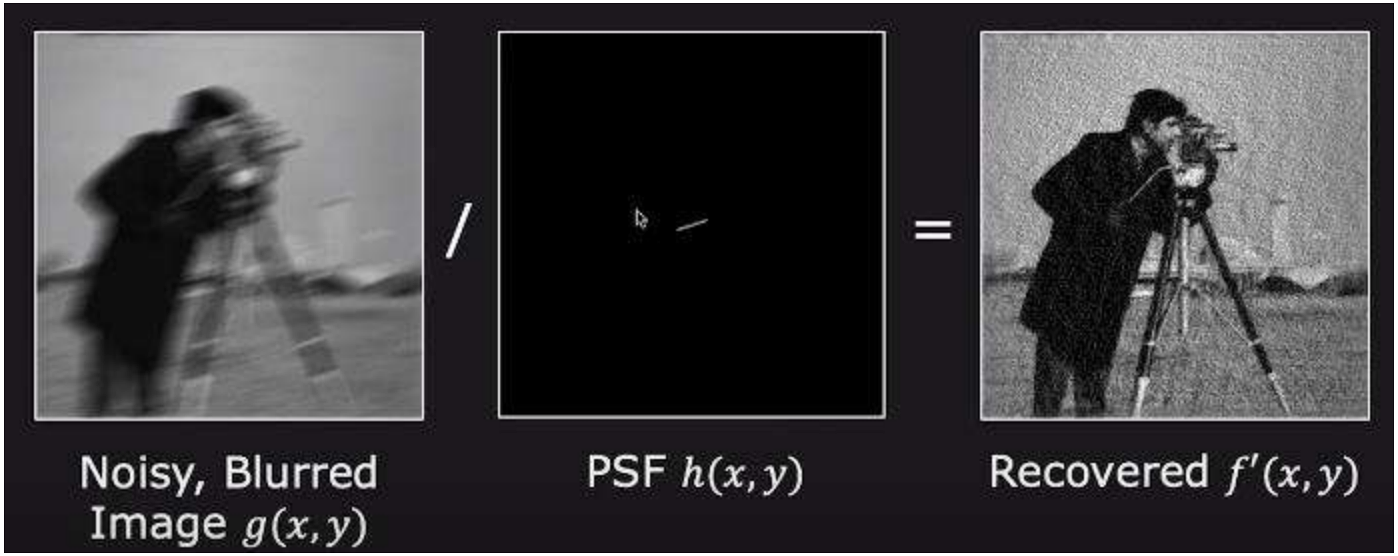
Weiner Filter (Frequency Domain)

In Some cases when $\frac{s_{n(u,v)}}{s_f(u,v)}$ = K (CONSTANT)

The Filter in Frequency Domain corresponds to the mentioned Equation:

F(u,v) = $\frac{1}{H(u,v)}\left[\frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma\, k}\right]$. G(u,v)

Where $k = 0.002$ keeping this as constant we can recover some images as shown below:

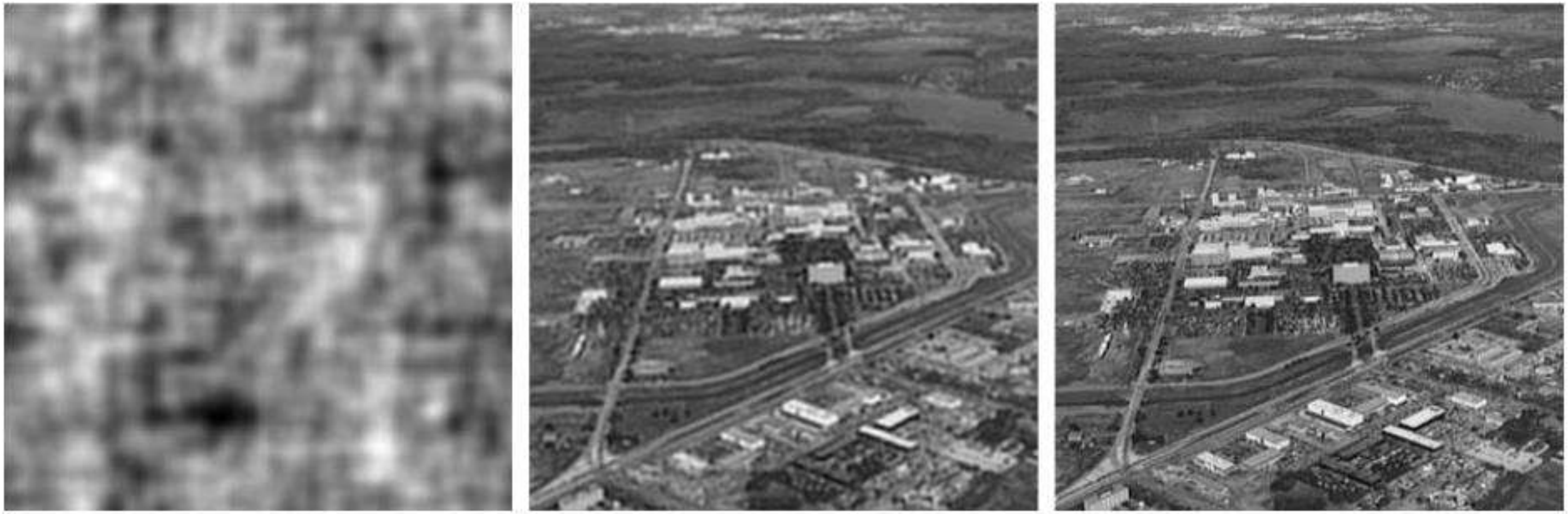


(Image Restoration) WIENER FILTERS

Use of this Filter:

- Wiener Filter Removes additive Noise and inverts the Blurring simultaneously
- The Wiener filter is optimal interms of the Mean square error
- Minimizes the overall mean square error in the process of inverse Filtering and Noise smooting
- The Wiener filter is a Linear estimation of Original Image.

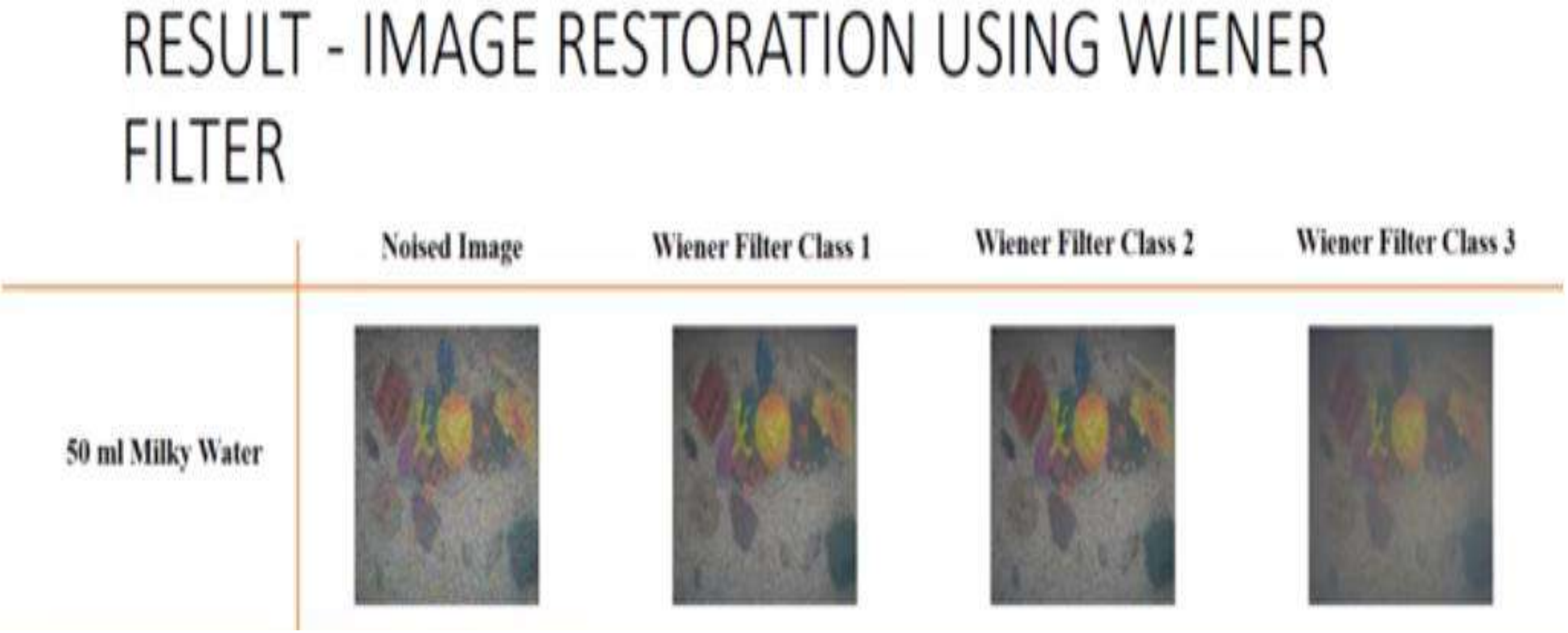
Image restoration results comparision



a b c

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.

Under water Image Restoration using Wiener Filters



RESULT - IMAGE RESTORATION USING WIENER FILTER

	Noised Image	Wiener Filter Class 1	Wiener Filter Class 2	Wiener Filter Class 3
50 ml Milky Water				

Disadvantage of Wiener Filter

- When we dont have information on the Power spectra the Wiener Filter is not optimal

Constrained Least square filter

Extension of Wiener Filter

Try to enforce the constraint to represent some degree of smoothness so that resultant image is smooth and noise free.

Laplacian or Second order derivative of Image

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

Problem becomes minimization of (Q).

Constrained Least square filter

The approximation of Second order derivative is possible with Convolving the Mask

P(x,y) = $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Let P(u,v) is Fourier Transform of Matrix

Now the Minimization of Second order derivative of image = (based on Constraint)

$\|g - H\dot{f}\|^2 = \|n\|^2$

➔ $\|Q\dot{f}\| = \dot{f}^T Q^T Q\dot{f}$

This Solution leads to Transpose function T^F in Frequency Domain:

$\dot{f}(u,v) = \left\{\frac{H(u,v)}{|H(u,v)|^2 + \gamma|P(u,v)|^2}\right\} G(u,v)$

$\gamma = Tune\ the\ degree\ of\ smooth\,ness.$

Let the residual difference be $r_s = H - g\dot{f}$

Then $\|r_s\|^2 = \|n\|^2 \pm c\ (This\ is\ \ for\ Optimatl\ Filter)$

Where $\|n\|^2 = MN({s_n}^2 + {m_n}^2)$

M,N = Image resolution

s_n= Mean

m_n = Variance

Constrained Least square Filter

For obtaining the Optimal filter the parameter (γ) tune the parameter

1. Specify the initial value of (γ)
2. Compute $\dot{\mathrm{f}}$ and $\|r\|^2$
3. Check whether R(u,v) = G(u,v) – H(u,v) $\dot{\mathrm{f}}$(u,v)

        if yes, then STOP

    if $\|r\|^2 < \|n\|^2$, increase the value of γ

    else if $\|r\|^2 > \|n\|^2 + \mathrm{a}$ then decrease the value of (γ)

Periodic Noise reduction by Frequency Domain Filtering

Periodic noise can be analyzed and filtered quite effectively using frequency domain techniques. Periodic noise appears as concentrated bursts of energy in the Fourier transform, at locations corresponding to the frequencies of the periodic interference.

The approach is to use a selective filter to isolate the noise.

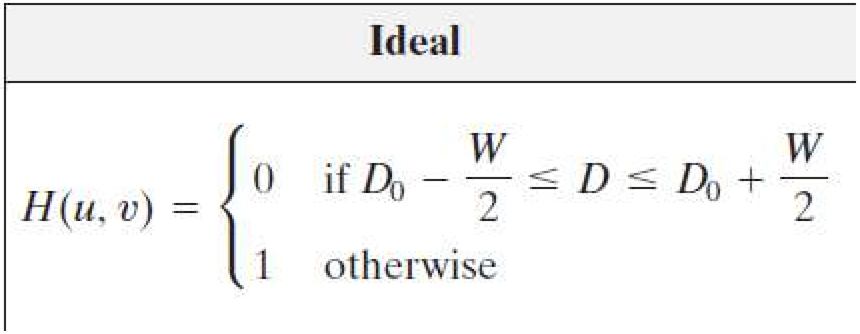
The three types of selective filters:
Bandreject,
Bandpass, and
Notch Filters.

Bandreject Filters: Reduce Periodic Noise

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.

Good example is an image corrupted by additive periodic noise that can be approximated as two-dimensional sinusoidal functions. It can be shown that the Fourier transform of a sine consists of two impulses that are mirror images of each other about the origin of the transform.

Ideal Bandreject Filter:



Ideal
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$

Where ***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
We show D instead of D(u,v) to simplify the notation in the table.

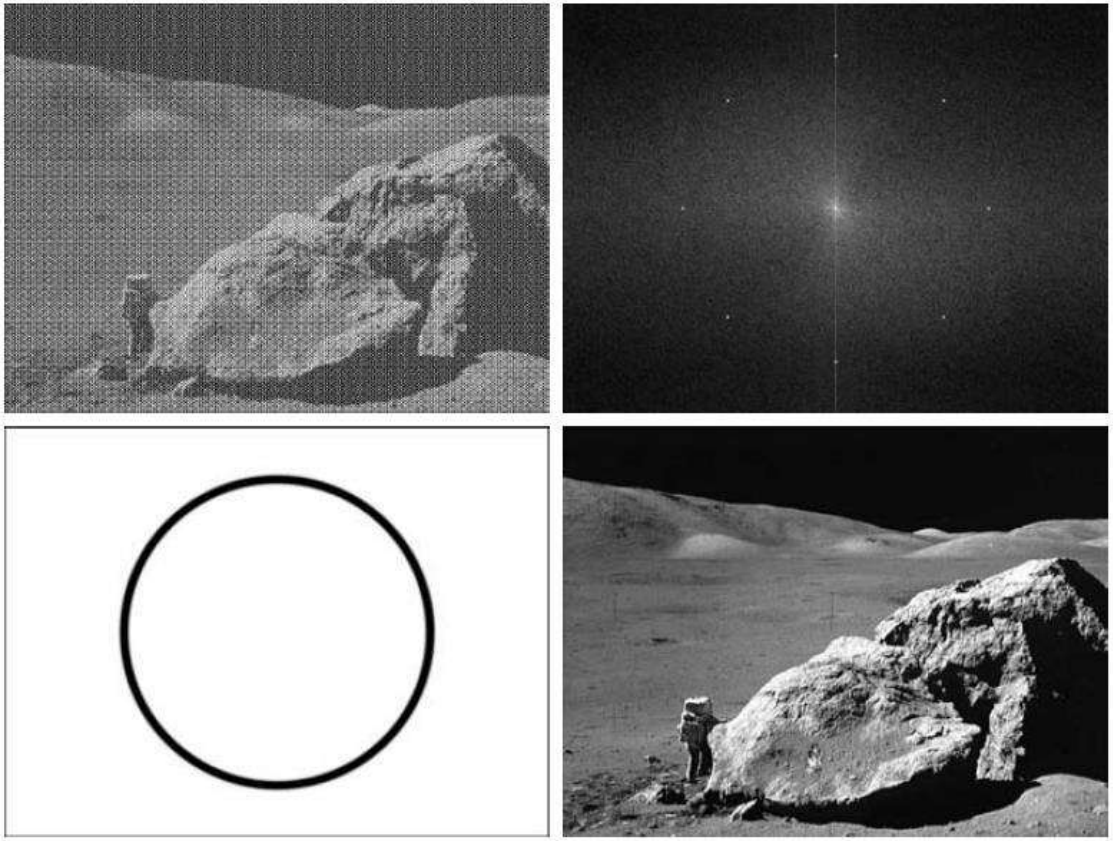
Butterworth Bandreject Filter

Butterworth Bandreject Filter equation is given below:

Butterworth

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

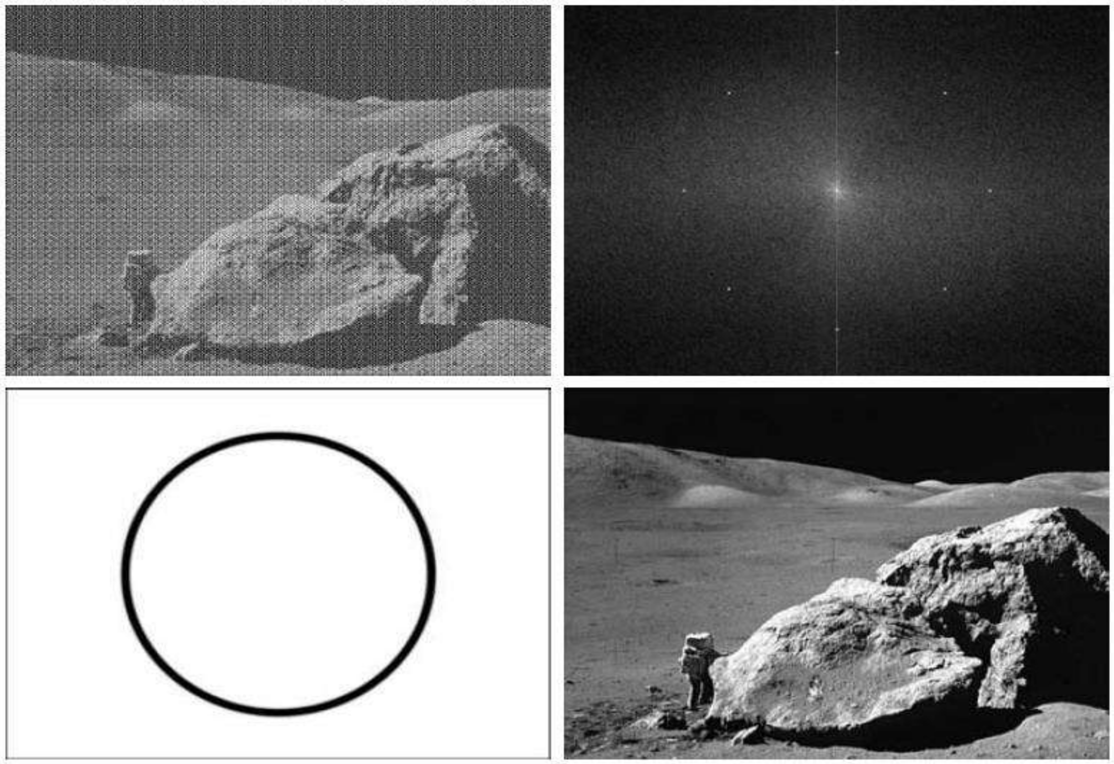
Where ***W*** *is the width of the band,*
D *is the distance D(u,v) from the center of the filter,*
D₀ *is the* cutoff frequency, and
n *is the order of the Butterworth filter.*
We show D instead of D(u,v) to simplify the notation in the table.



a b
c d

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

Butterworth Bandreject Filter



a b
c d

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

shows an image heavily corrupted by sinusoidal noise of various frequencies.The noise components are easily seen as symmetric pairs of bright dots in the Fourier spectrum shown in Fig. 5.16(b). In this example, the components lie on an approximate circle about
the origin of the transform, so a circularly symmetric bandreject filter is a good
choice.

5.16(c) shows a Butterworth bandreject filter of order 4, with the appropriate radius and width to enclose completely the noise impulses. Since it is desirable in general to remove as little as possible from the transform, sharp, narrow filters are common in bandreject filtering.

Even small details and textures were restored effectively by this simple filtering approach. It is worth noting also that it would not be possible to get equivalent results by a direct spatial domain filtering approach using small convolution masks.

Gaussian Bandreject Filter

Gaussian Bandrejct Filter is given as:

Gaussian

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

Where ***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
We show D instead of D(u,v) to simplify the notation in the table.

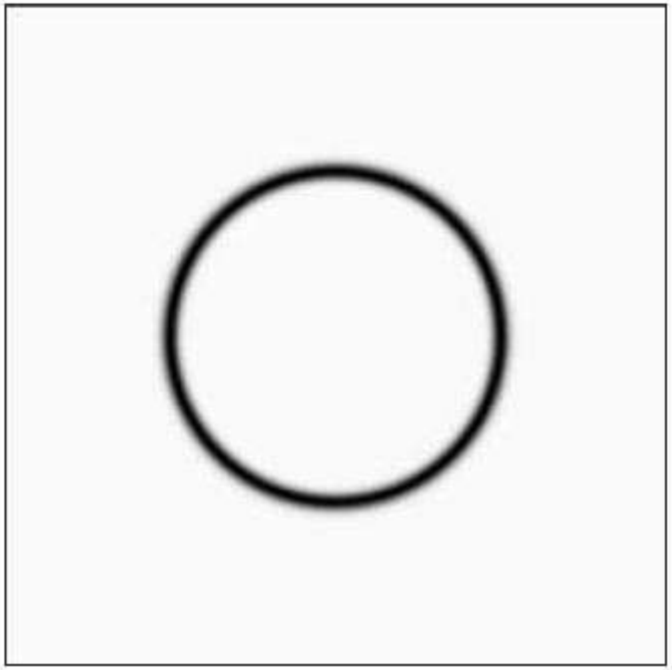
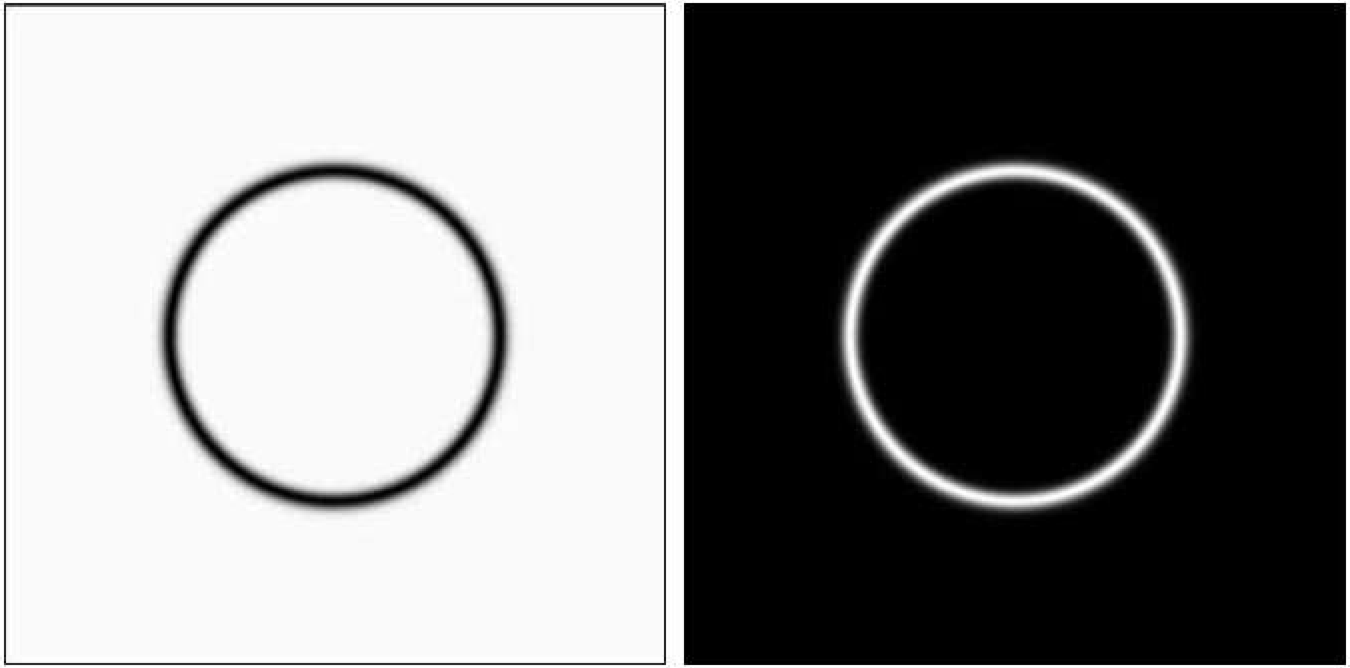


FIGURE 4.63
(a) Bandreject
Gaussian filter.

Bandpass Filters

A *bandpass filter performs the opposite operation of a bandreject filter.*
A bandpass filter is obtained from a bandreject filter in the same manner that we obtained a highpass filter from a lowpass filter: The Transfer Function

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

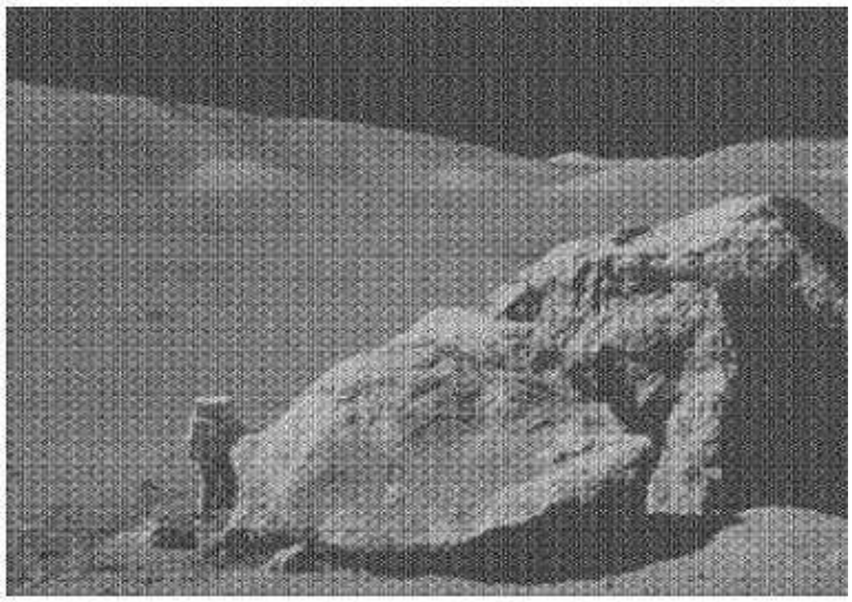


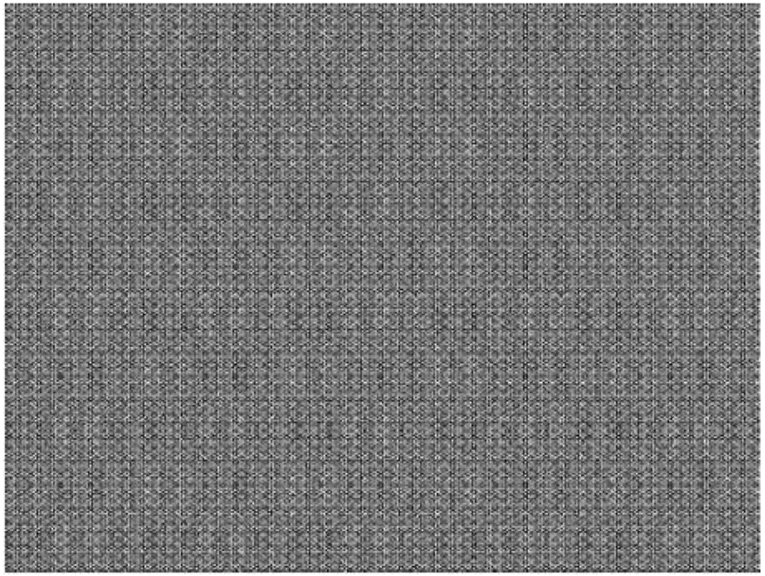
a b

FIGURE 4.63 (a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

Bandpass filtering is quite useful in isolating the effects on an image caused by selected frequency bands.

Bandpass Filter: To isolate Noise in the images





Sinusoidal Noise Image and Right is Noise only isolated and captured in the Image.

Steps to isolate the Noise from This image:

(1) Use Eq (1) using Eq. (5.4-1) to obtain the bandpass filter corresponding to the bandreject filter used in Fig. 5.16; and

(2) Taking the inverse transform of the bandpass-filtered transform. Most image detail was lost, but the information that remains is most useful, as it is clear that the noise pattern recovered using this method is quite close to the noise that corrupted the image in Fig. 5.16(a).

Summary of Steps for Fourier Transform

Summary of Steps for Filtering in the Frequency Domain

The material in the previous two sections can be summarized as follows:

1. Given an input image (*x, y) of size* M * N, ***obtain the padding parameters***
P and Q from Eqs. (4.6-31) and (4.6-32). Typically, we select P = 2M and Q = 2N.

2. Form a padded image, of size P * Q **by appending the necessary** number of zeros to f(*x, y).*

3. Multiply $f_p(x, y)$ **by** $(-1)^{x+y}$ **to center its transform.**

4. Compute the DFT, F(*u, v), of the image from step 3.*

5. Generate a real, symmetric filter function,*H(u, v), of size* P * Q ***with center***
at coordinates (*P/2,Q/2).† Form the product* G(u, v) = H(u, v)F(u, v) using array multiplication; that is, G(i, k) = H(i, k)F(i, k).

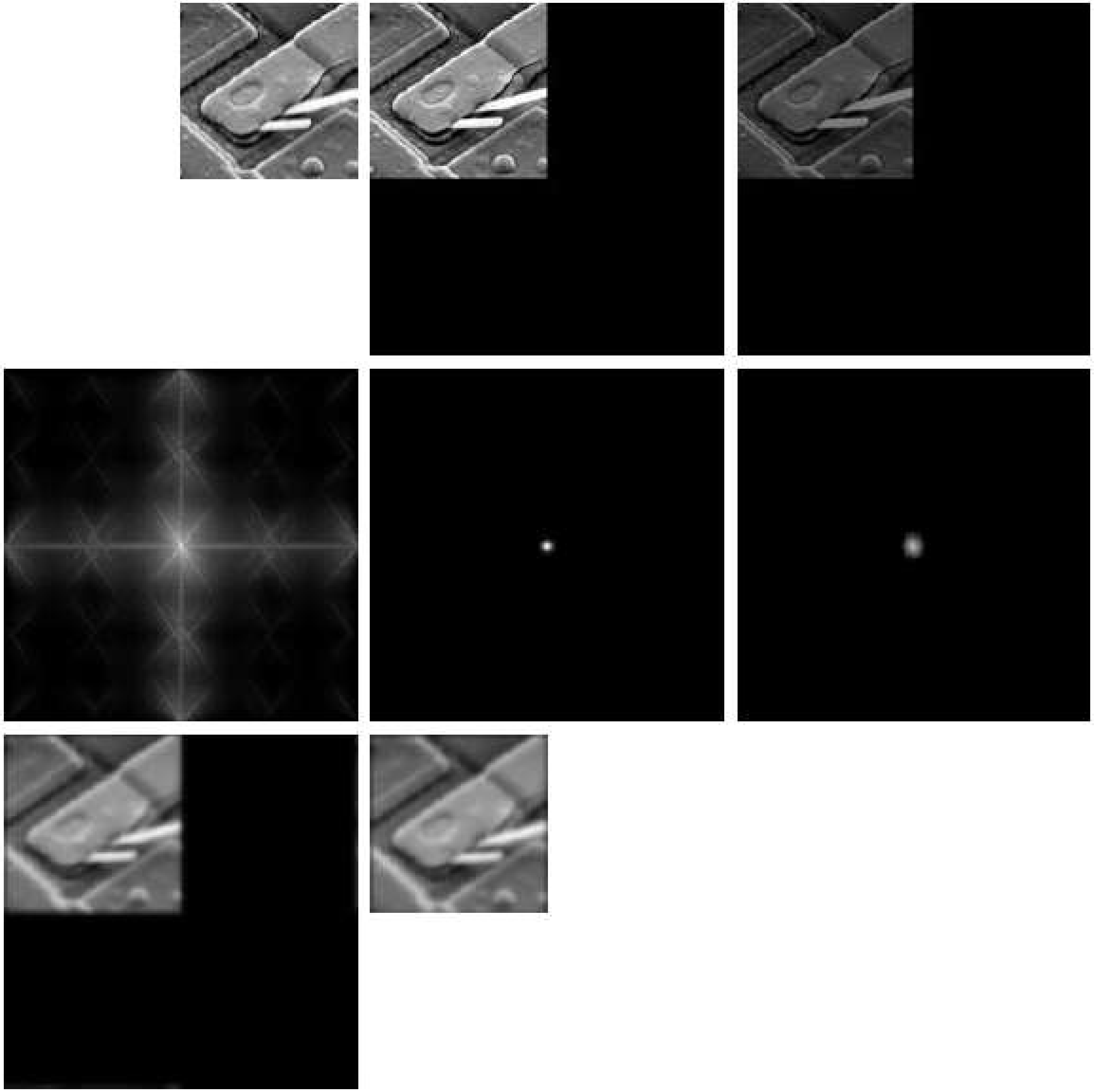
6. Obtain the processed image:

$$g_p(x, y) = \{\text{real}\,[\mathfrak{I}^{-1}[G(u, v)]]\}\,(-1)^{x+y}$$

where the real part is selected in order to ignore parasitic complex components resulting from computational inaccuracies, and the subscript *p indicates* that we are dealing with padded arrays.

7. Obtain the final processed result, (*x, y), by extracting* M * N ***the region*** from the top, left quadrant of $g_p(x, y)$.

Figure 4.36 illustrates the preceding steps.The legend in the figure explains the source of each image. If it were enlarged, Fig. 4.36(c) would show black dots interleaved in the image because negative intensities are clipped to 0 for display. Note in Fig. 4.36(h) the characteristic dark border exhibited by lowpass filtered images processed using zero padding.



a b c
d e f
g h

FIGURE 4.36 (a) An $M \times N$ image, f. (b) Padded image, f_p of size $P \times Q$. (c) Result of multiplying f_p by $(-1)^{x+y}$. (d) Spectrum of F_p. (e) Centered Gaussian lowpass filter, H, of size $P \times Q$. (f) Spectrum of the product HF_p. (g) g_p, the product of $(-1)^{x+y}$ and the real part of the IDFT of HF_p. (h) Final result, g, obtained by cropping the first M rows and N columns of g_p.

Notch Filters

A Notch filter rejects (or passes) frequencies in predefined neighborhoods about the center of frequency rectangle.

Zero-phase-shift filters must be symmetric about the origin, so a notch with center at (u_0, v_0) must have a corresponding notch at location ($-u_0$, $-v_0$).

Notch reject filters are constructed as products of highpass filters whose centers have been translated to the centers of the notches.

The general form of Notch Reject Filter is given below Equation :

$$H_{\mathrm{NR}}(u, v) = \prod_{k=1}^{Q} H_k(u, v) H_{-k}(u, v)$$

where $H_k(u, v)$ and $H_{-k}(u, v)$are highpass filters whose centers are at (u_k, v_k) and ($-u_k$, $-v_k$), respectively. These centers are specified with respect to the center of the frequency rectangle, (*M/2, N/2). The distance computations for* each filter are thus carried out using the expressions given Below Equations:

$$D_k(u, v) = \left[(u - M/2 - u_k)^2 + (v - N/2 - v_k)^2\right]^{1/2} \qquad (4.10\text{-}3)$$

$$D_{-k}(u, v) = \left[(u - M/2 + u_k)^2 + (v - N/2 + v_k)^2\right]^{1/2} \qquad (4.10\text{-}4)$$

Butterworth Notch Reject Filter

The following is a Butterworth notch reject filter of order *n, containing* three notch pairs:

$$H_{\mathrm{NR}}(u, v) = \prod_{k=1}^{3}\left[\frac{1}{1 + [D_{0k}/D_k(u, v)]^{2n}}\right]\left[\frac{1}{1 + [D_{0k}/D_{-k}(u, v)]^{2n}}\right] \quad (4.10\text{-}5)$$

where D_k and $D_{-k}$and are given by Eqs. (4.10-3) and (4.10-4).The constant D_{0k} is the same for each pair of notches, but it can be different for different pairs. Other notch reject filters are constructed in the same manner, depending on the highpass filter chosen. As with the filters discussed earlier, a Butterworth *notch pass filter* is obtained from a notch reject filter using the expression

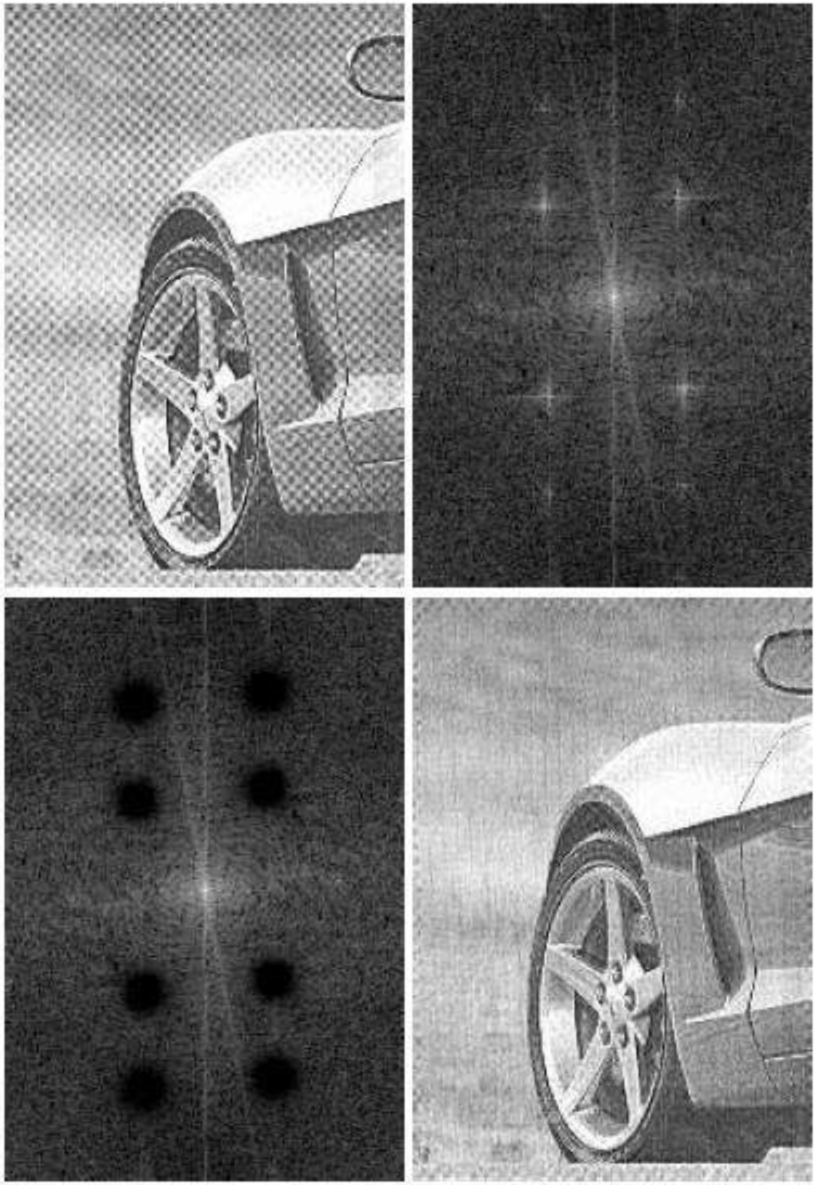
$$H_{\mathrm{NP}}(u, v) = 1 - H_{\mathrm{NR}}(u, v) \quad (4.10\text{-}6)$$

Application of Notch Filters

One of the principal applications of notch filtering is for selectively modifying local regions of the DFT. (Discrete Fourier Transformed)
(Passes/Reject frequencies in predefined Neighborhood Regions)
This type of processing typically is done interactively, working directly on DFTs obtained without padding.
The advantages of working interactively with actual DFTs (as opposed to having to "translate" from padded to actual frequency values) outweigh any wraparound errors that may result from not using padding in the filtering process.

Application of Notch Filter: Removing Moire patterns in input Image

Removing periodic noise Example1



a b
c d

FIGURE 4.64 (a) Sampled newspaper image showing a moiré pattern. (b) Spectrum. (c) Butterworth notch reject filter multiplied by the Fourier transform. (d) Filtered image.

Figure 4.64(a) is the scanned newspaper image from Fig. 4.21, showing a prominent moiré pattern, and Fig. 4.64(b) is its spectrum.

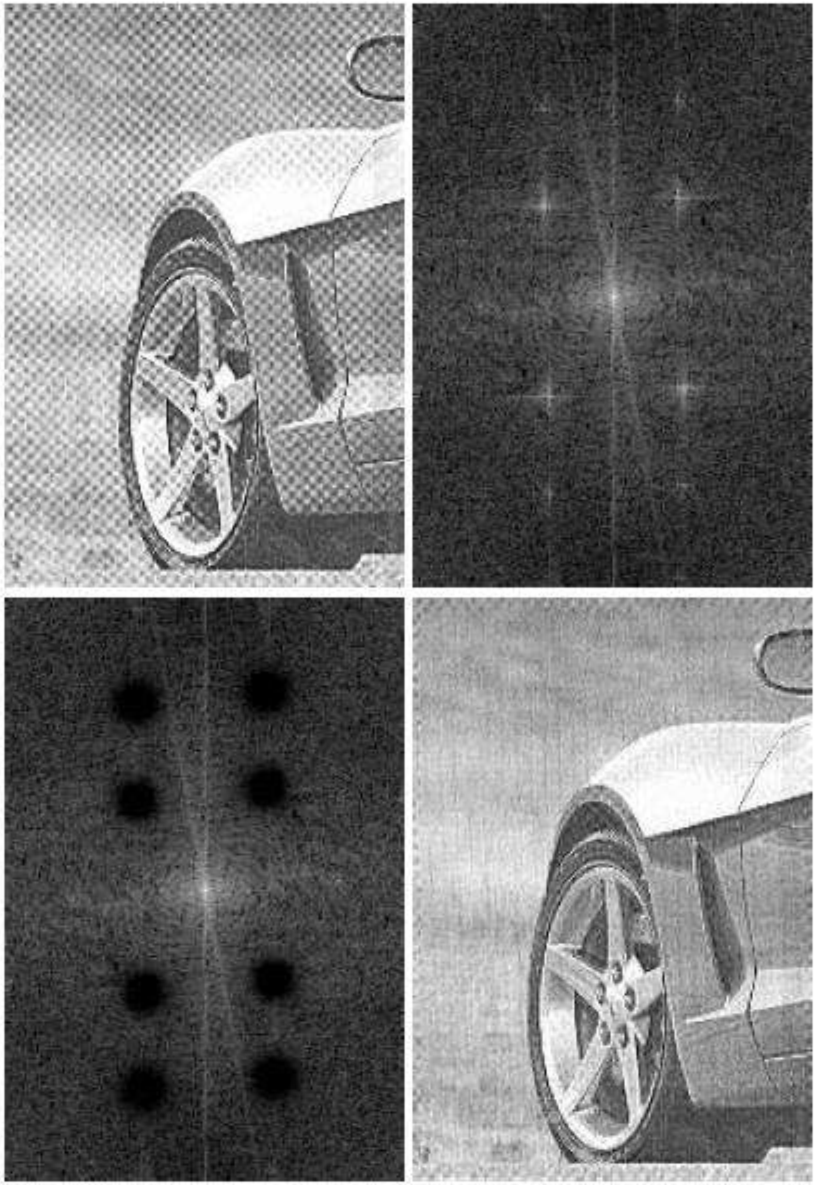
We know that the Fourier transform of a pure sine, which is a periodic function, is a pair of conjugate symmetric impulses. The symmetric “impulse-like” bursts in Fig. 4.64(b) are a result of the near periodicity of the moiré pattern.

We can attenuate these bursts by using notch filtering.

Figure 4.64(c) shows the result of multiplying the DFT of Fig. 4.64(a) by a Butterworth notch reject filter with D_0 = 3 & n = 4 and for all notch pairs.

The value of the radius was selected (by visual inspection of the spectrum) to encompass the energy bursts completely, and the value of *n was selected to give* notches with mildly sharp transitions.

Application of Notch Reject Filter: Removing Moire patterns in input Image (**CONTD………**)



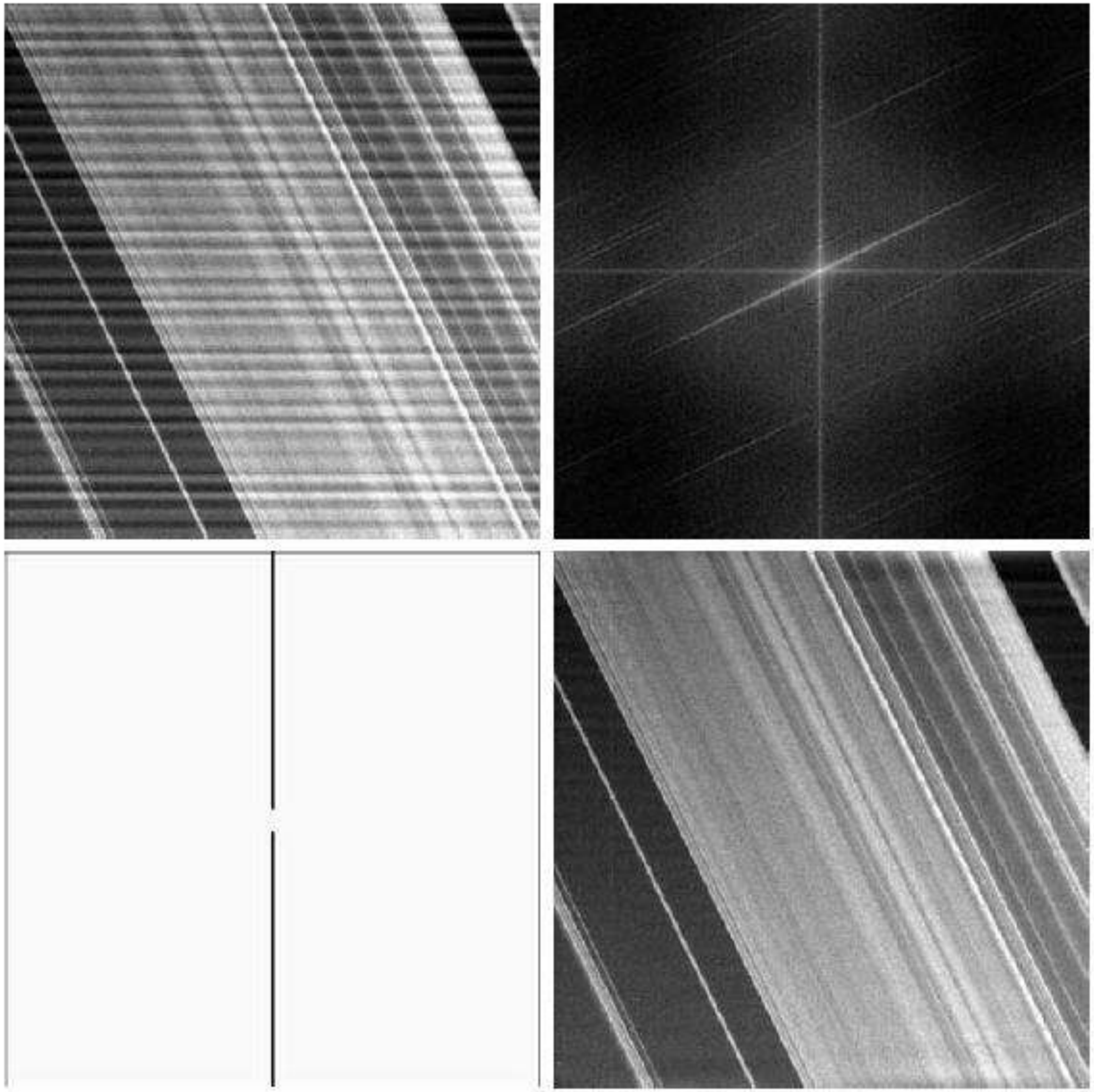
a b
c d

FIGURE 4.64
(a) Sampled newspaper image showing a moiré pattern.
(b) Spectrum.
(c) Butterworth notch reject filter multiplied by the Fourier transform.
(d) Filtered image.

The locations of the center of the notches were determined interactively from the spectrum.

Figure 4.64(d) shows the result obtained with this filter using the procedure outlined in Section 4.7.3. The improvement is significant, considering the low resolution and degradation of the original image.

Application of Notch Filters: Removing periodic noise Example2



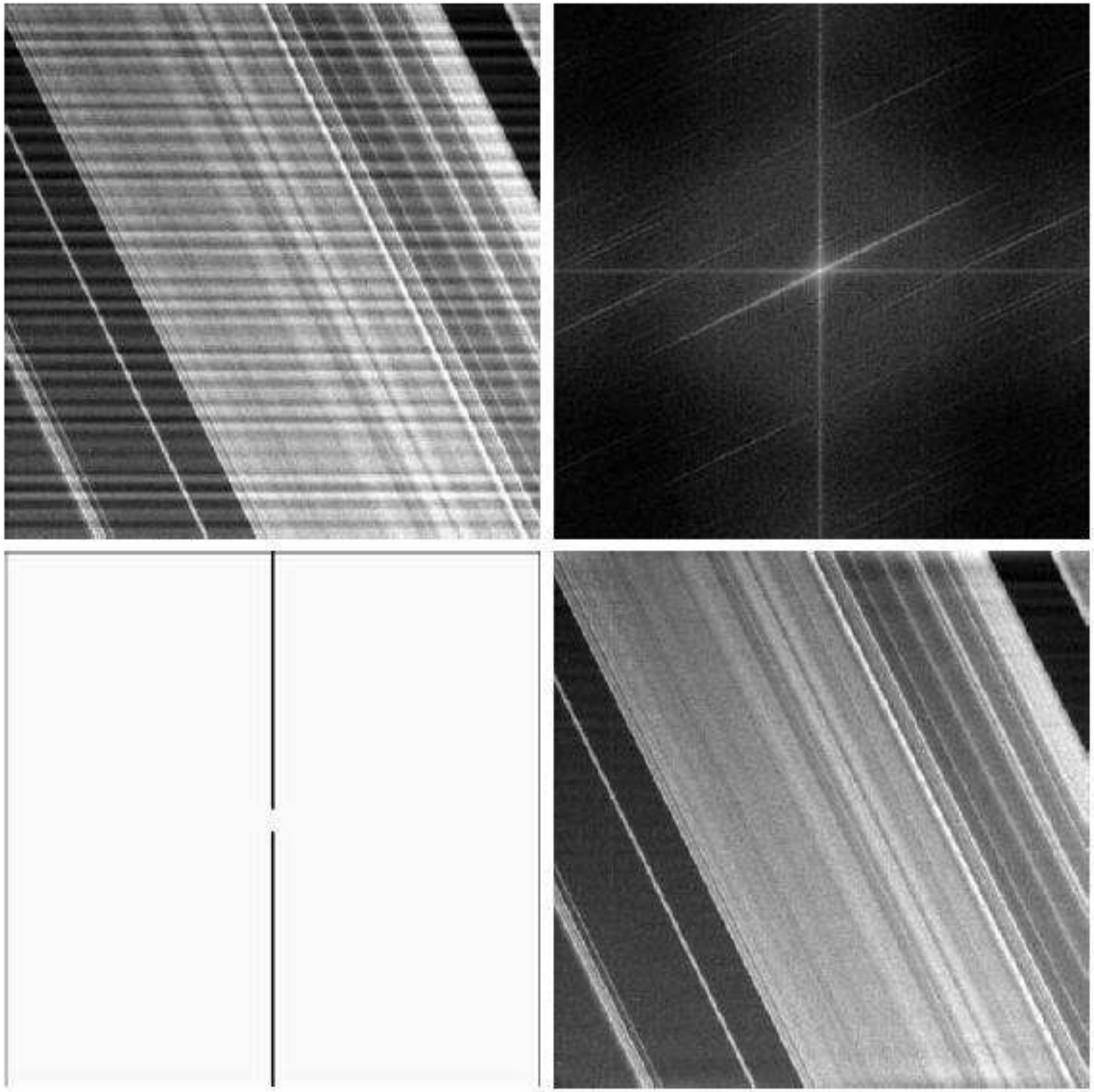
a b
c d

FIGURE 4.65 (a) 674 × 674 image of the Saturn rings showing nearly periodic interference. (b) Spectrum: The bursts of energy in the vertical axis near the origin correspond to the interference pattern. (c) A vertical notch reject filter. (d) Result of filtering. The thin black border in (c) was added for clarity; it is not part of the data. (Original image courtesy of Dr. Robert A. West, NASA/JPL.)

Figure 4.65(a) shows an image of part of the rings surrounding the planet Saturn. This image was captured by Satellite *Cassini*. The vertical sinusoidal pattern was caused by an AC signal superimposed on the camera video signal just prior to digitizing the image. This was an unexpected problem that corrupted some images from the mission.

Fortunately, this type of interference is fairly easy to correct by postprocessing. One approach is to use notch filtering.

Application of Notch Reject Filters: Removing periodic noise Example2



a b
c d

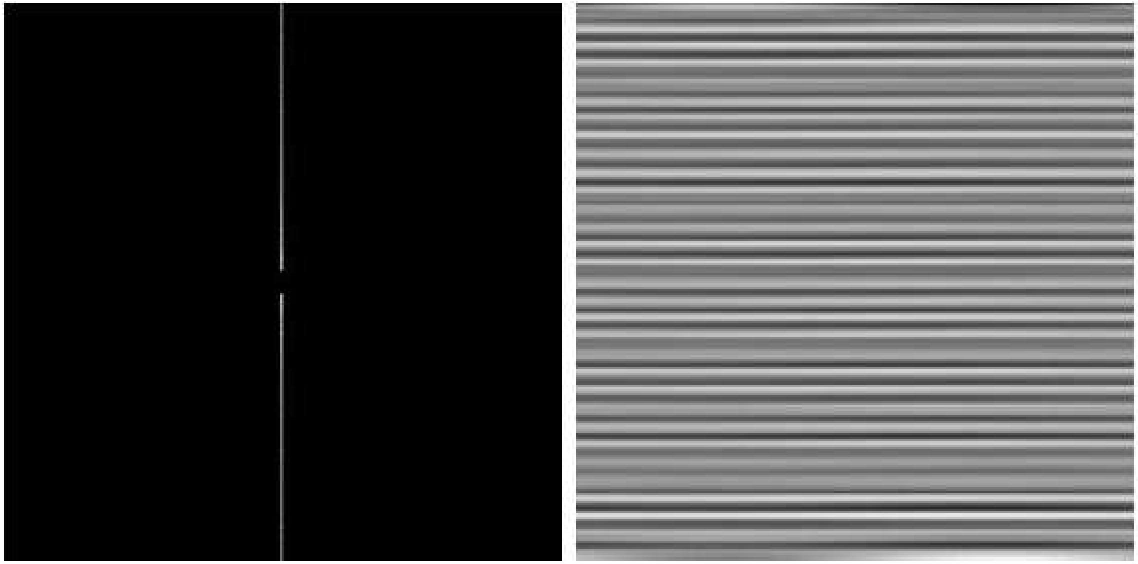
FIGURE 4.65 (a) 674 × 674 image of the Saturn rings showing nearly periodic interference. (b) Spectrum: The bursts of energy in the vertical axis near the origin correspond to the interference pattern. (c) A vertical notch reject filter. (d) Result of filtering. The thin black border in (c) was added for clarity; it is not part of the data. (Original image courtesy of Dr. Robert A. West, NASA/JPL.)

Figure 4.65(b) shows the DFT spectrum. Careful analysis of the vertical axis reveals a series of small bursts of energy which correspond to the nearly sinusoidal interference. A simple approach is to use a narrow notch rectangle filter starting with the lowest frequency burst and extending for the remaining of the vertical axis. Figure 4.65(c) shows such a filter (white represents 1 and black 0). Figure 4.65(d) shows the result of filtering the corrupted image with this filter. This result is a significant improvement over the original image.

Application of Notch Pass Filters: Isolating the Periodic Noise

a b

FIGURE 4.66 (a) Result (spectrum) of applying a notch pass filter to the DFT of Fig. 4.65(a). (b) Spatial pattern obtained by computing the IDFT of (a).

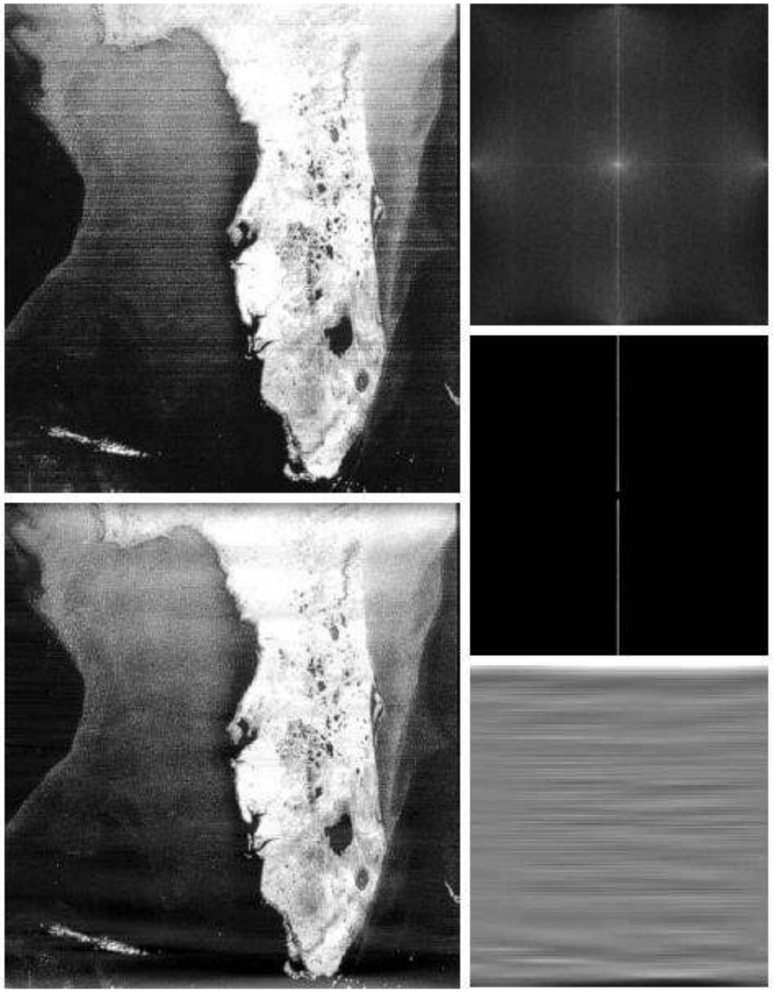


We isolated the frequencies in the vertical axis using a notch *pass version of* the same filter [Fig. 4.66(a)].Then, as Fig. 4.66(b) shows, the IDFT of these frequencies yielded the spatial interference pattern itself.

Notch Filters (Pass)/(Reject) With out blurring of the Input Image. [Compare with Image smoothing using Frequency domain Filters]

What is your observation?

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



a b
c
e d

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

st by looking at the nearly horizontal lines of the noise pattern in Fig. 5.19(a), we expect its contribution in the frequency domain to be concentrated along the vertical axis. However, the noise is not dominant enough to have a clear pattern along this axis, as is evident from the spectrum shown in Fig. 5.19(b).We can get an idea of what the noise contribution looks like by constructing a simple ideal notch pass filter along the vertical axis of the Fourier transform, as shown in Fig. 5.19(c).The spatial representation of the noise pattern (inverse transform of the notch-pass–filtered result) is shown in Fig. 5.19(d).This noise pattern corresponds closely to the pattern in Fig. 5.19(a).

we can obtain the corresponding notch reject filter from Eq. (5.4-2). The result of processing the image with the notch reject filter is shown in Fig. 5.19(e). This is output Image.