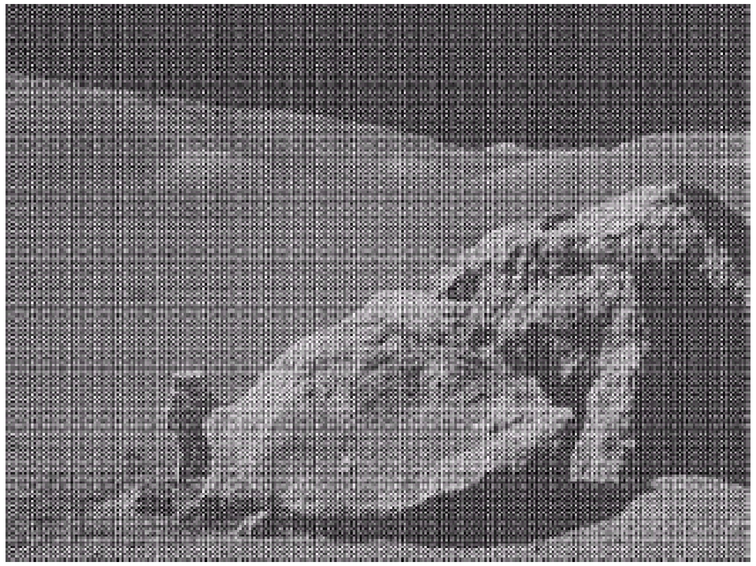
Image Processing and Pattern Recognition (IPPR)

Chapter 4:Image Restoration



Basanta Joshi, PhD

Asst. Prof., Depart of Electronics and Computer Engineering

Member, Laboratory for ICT Research and Development (LICT)

Institute of Engineering

basanta@ioe.edu.np

http://www.basantajoshi.com.np

https://scholar.google.com/citations?user=iocLiGcAAAAJ

https://www.researchgate.net/profile/Basanta_Joshi2

Contents

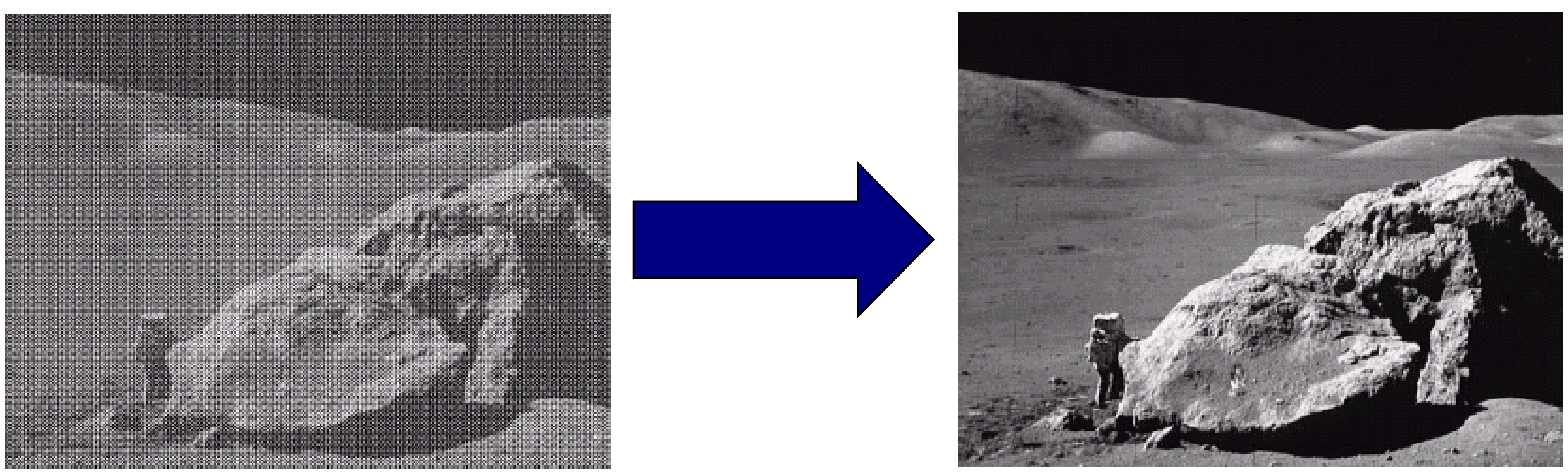
In this lecture we will look at image restoration techniques used for noise removal

- What is image restoration?
- Noise and images
- Noise models
- Noise removal using spatial domain filtering
- Periodic noise
- Noise removal using frequency domain filtering

What is Image Restoration?

Image restoration attempts to restore images that have been degraded

- Identify the degradation process and attempt to reverse it
- Similar to image enhancement, but more objective



Noise and Images

The sources of noise in digital images arise during image acquisition (digitization) and transmission

- Imaging sensors can be affected by ambient conditions
- Interference can be added
  to an image during transmission

Noise and Images

Image restoration

- Restoration is an objective process that attempts to recover an image that has been degraded
  - A priori knowledge of the degradation phenomenon
  - Restoration techniques generally oriented toward modeling the degradation
  - Application of the inverse process to “recover” the original image
  - Involves formulating some criterion (criteria) of “goodness” that is used to measure the desired result

Noise and Images

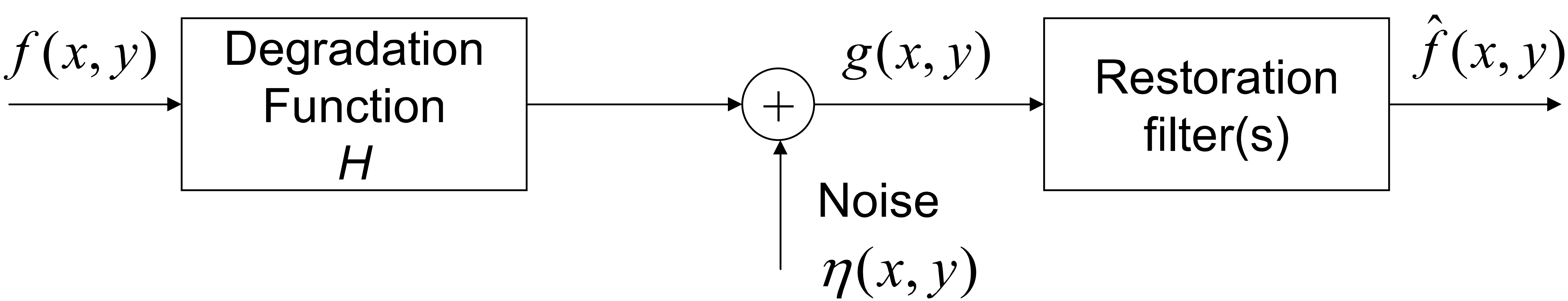
Image restoration (continued)

- Removal of blur by a deblurring function is an example restoration technique
- We will consider the problem only from where a degraded *digital* image is given
  - Degradation source will not be considered here
- Restoration techniques may be formulated in the
  - Frequency domain
  - Spatial domain

Noise and Images

Image degradation/restoration process

- Given $g(x,y)$, some knowledge about H, and some knowledge about the noise term, the objective is to produce an estimate of the original image.
  - The more that is known about H and the noise term the closer the estimate can be
- Various types of restoration filters are used to accomplish this.



Noise and Images

Image degradation/restoration process

- If *H* is a linear, position invariant process, then the degraded image can be described as the convolution of *h* and *f* with an added noise term:

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

- *h*(*x*,*y*) is the spatial domain representation of the degradation function.
- In the frequency domain, the representation is:

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

- Each term in this expression is the Fourier transform of the of the corresponding terms in the equation above.

Noise Model

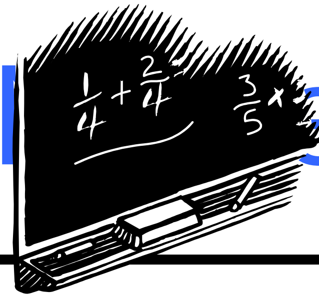
We can consider a noisy image to be modelled as follows:

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

where *f(x, y)* is the original image pixel, $\eta(x, y)$ is the noise term and *g(x, y)* is the resulting noisy pixel

If we can estimate the model the noise in an image is based on this will help us to figure out how to restore the image

Image Corruption Example



Original Image ***x***

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	51	52	52	56	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
148	154	157	160	163	167	170
151	155	159	162	165	169	172

Image f (x, y)

y

Noisy Image ***x***

Image f (x, y)

y

Noise models

- Common sources of noise
  - Acquisition
    - Environmental conditions (heat, light), imaging sensor quality
  - Transmission
    - Noise in transmission channel
- Spatial and frequency properties of noise
  - Frequency properties of noise refer to the frequency content of noise in the Fourier sense
  - For example, if the Fourier spectrum of the noise is constant, the noise is usually called *white noise*
    - A carry over from the fact that white light contains nearly all frequencies in the visible spectrum in basically equal proportions
  - Excepting spatially periodic noise, we will assume that noise is independent of spatial coordinates and uncorrelated to the image

Noise Models

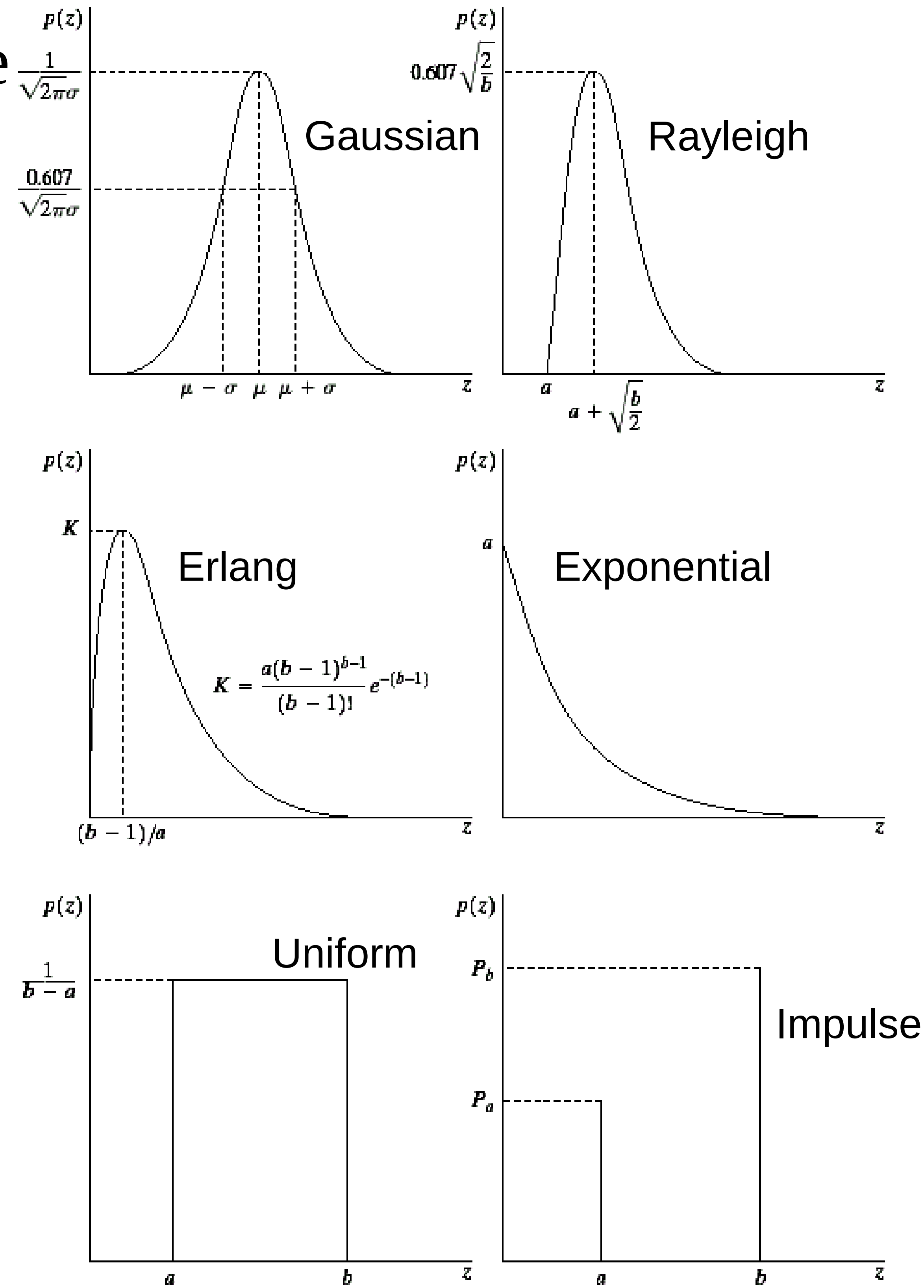
Noise probability density functions

- With respect to the *spatial* noise term, we will be concerned with the statistical behavior of the intensity values.
- May be treated as random variables characterized by a probability density function (PDF)
- Common PDFs used will describe:
  - Gaussian noise
  - Rayleigh noise
  - Erlang (Gamma) noise
  - Exponential noise
  - Uniform noise
  - Impulse (salt-and-pepper) noise

Noise Models

There are many different models for the image
noise term $\eta(x, y)$:

- Gaussian
  - Most common model
- Rayleigh
- Erlang
- Exponential
- Uniform
- Impulse
  - *Salt and pepper* noise



Noise Models

Gaussian noise

- Gaussian (normal) noise models are simple to consider
- The PDF of a Gaussian random variable, *z*, is given to the right as:
- In this case, approximately 70% of the values of *z* will be within within one standard deviation
- Approximately 95% of the values of *z* will be within within two standard deviations

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

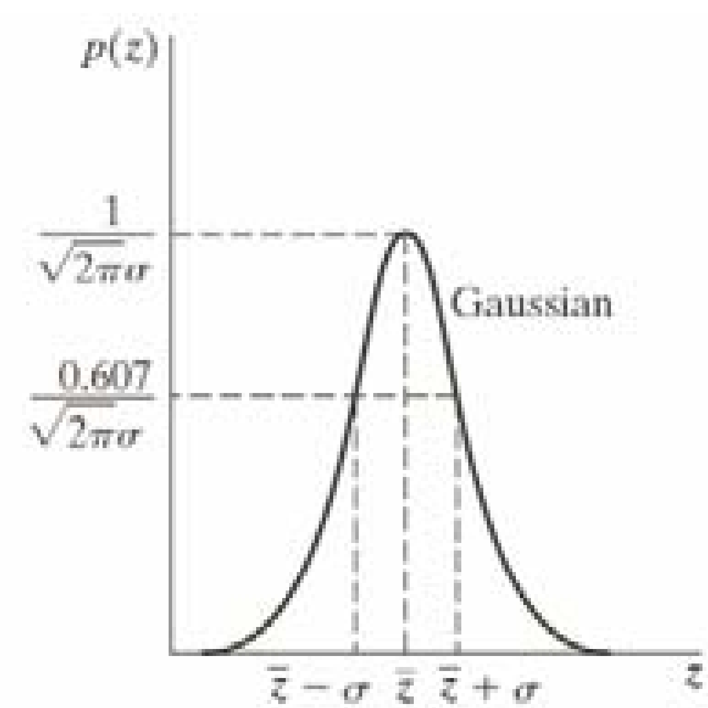
where

z represents intensity

$\bar{z}$ represents the mean (average) value of z

σ is the standard deviation

σ^2 is the variance of z



Noise Models

Rayleigh noise

- The PDF of Rayleigh noise is given as:

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

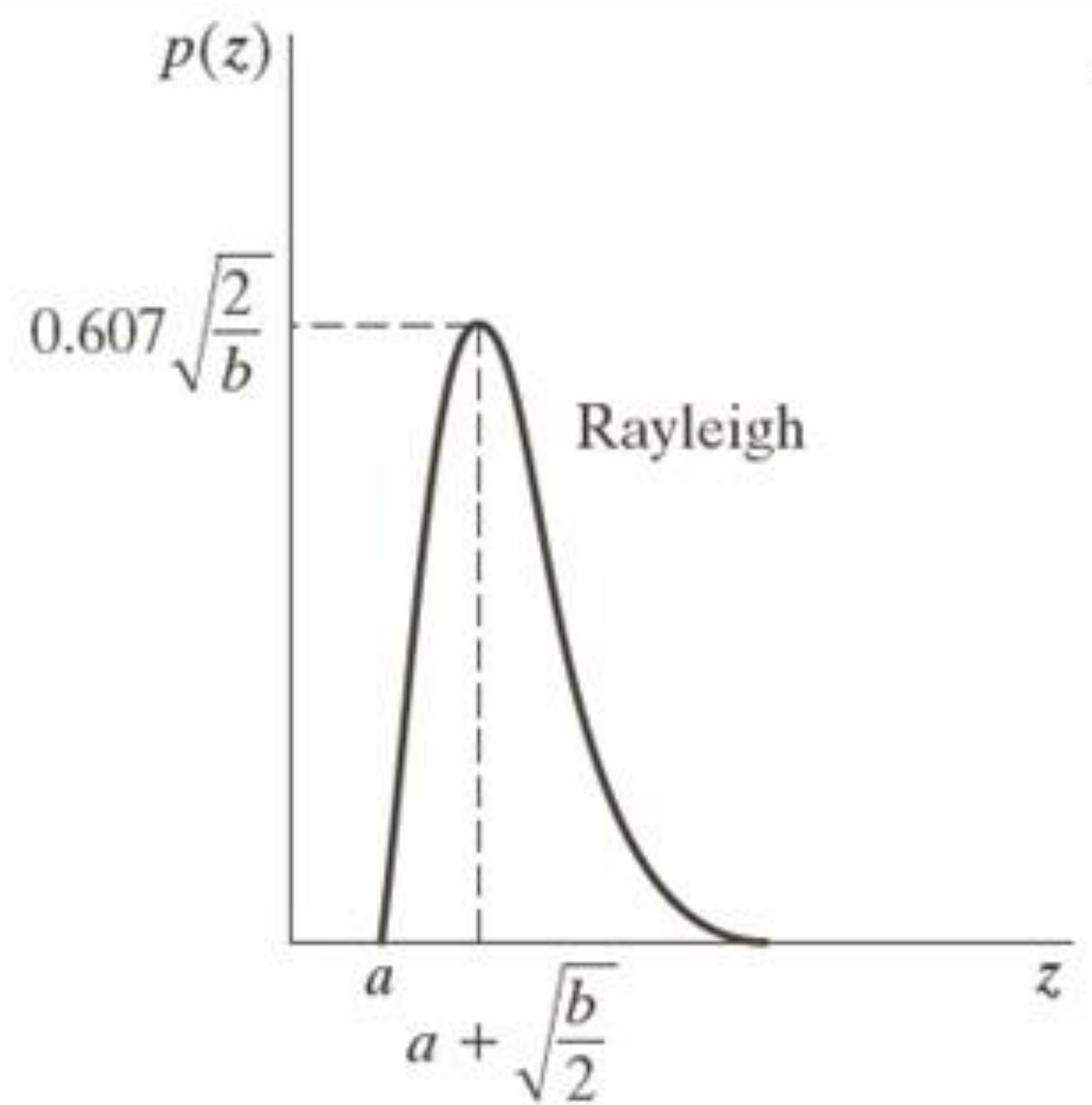
where

z represents intensity

$$\bar{z} = a + \sqrt{\pi b / 4}$$

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

- Note the displacement, by *a*, from the origin
- The basic shape of this PDF is skewed to the right
  – Can be useful in approximating skewed histograms



Noise Models

Erlang (Gamma) noise

- The PDF of Erlang noise is given as:

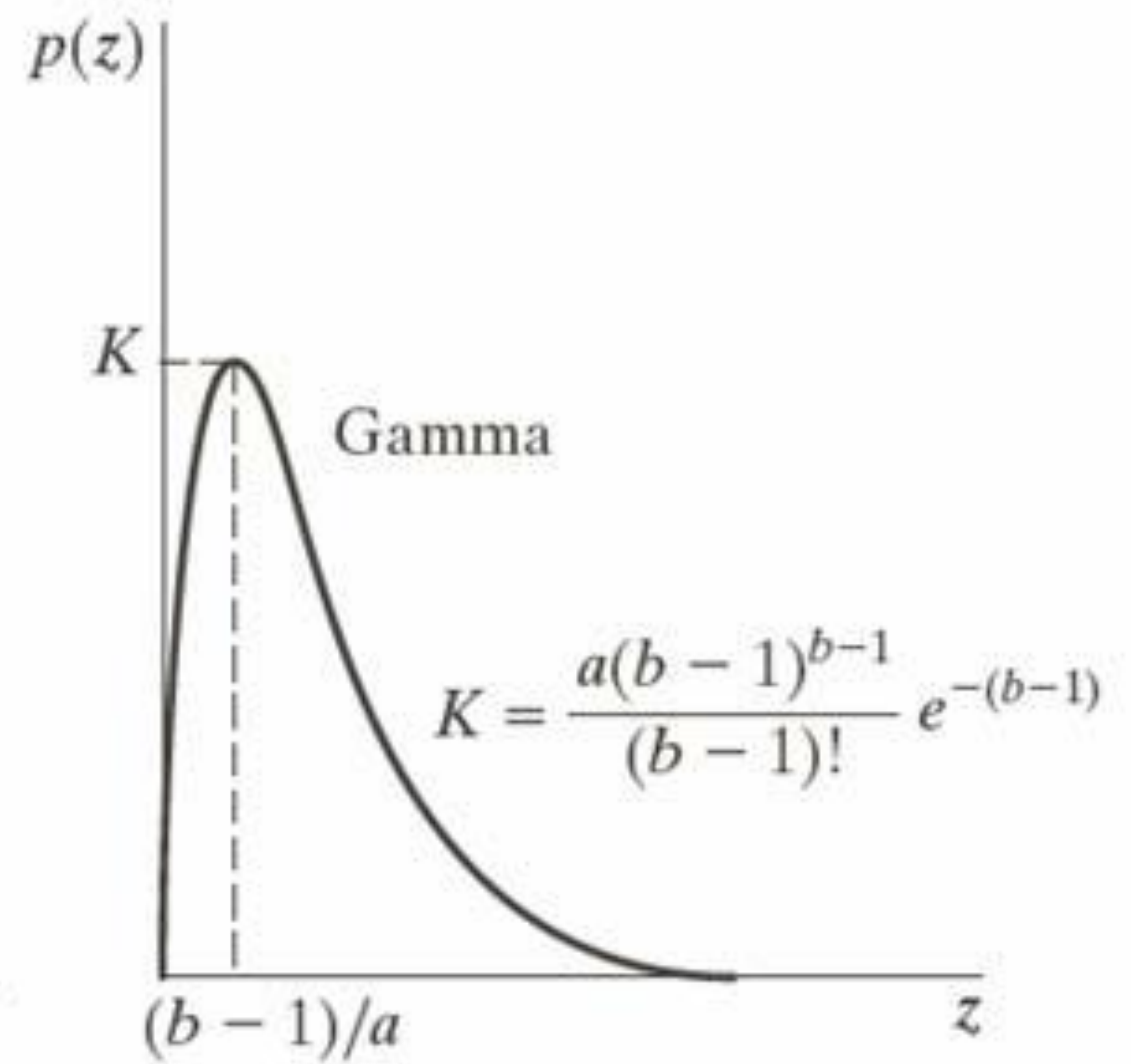
$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \ge 0 \\ 0 & \text{for } z < 0 \end{cases}$$

$$\textit{where}$$

$$z \text{ represents intensity}$$

$$\bar{z} = b/a$$

$$\sigma^2 = b/a^2$$



- $a > 0$, b is a positive integer

Noise Models

Exponential noise

- The PDF of exponential noise is given as:

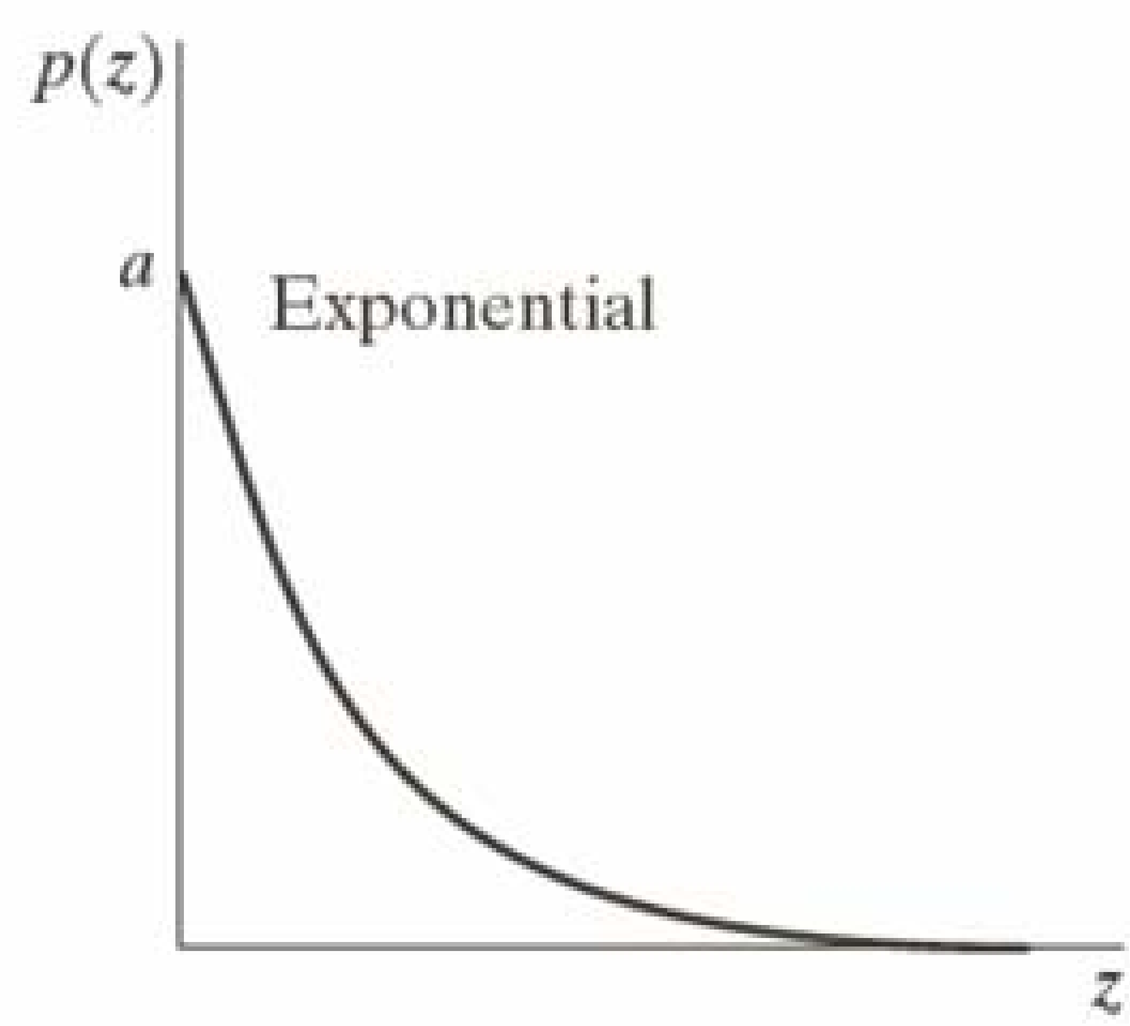
$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

$where$

z represents intensity

$$\bar{z} = 1/a$$

$$\sigma^2 = 1/a^2$$



- *a > 0*
- This PDF is a special case of the Erlang PDF with *b*=1

Noise Models

Uniform noise

- The PDF of uniform noise is given as:

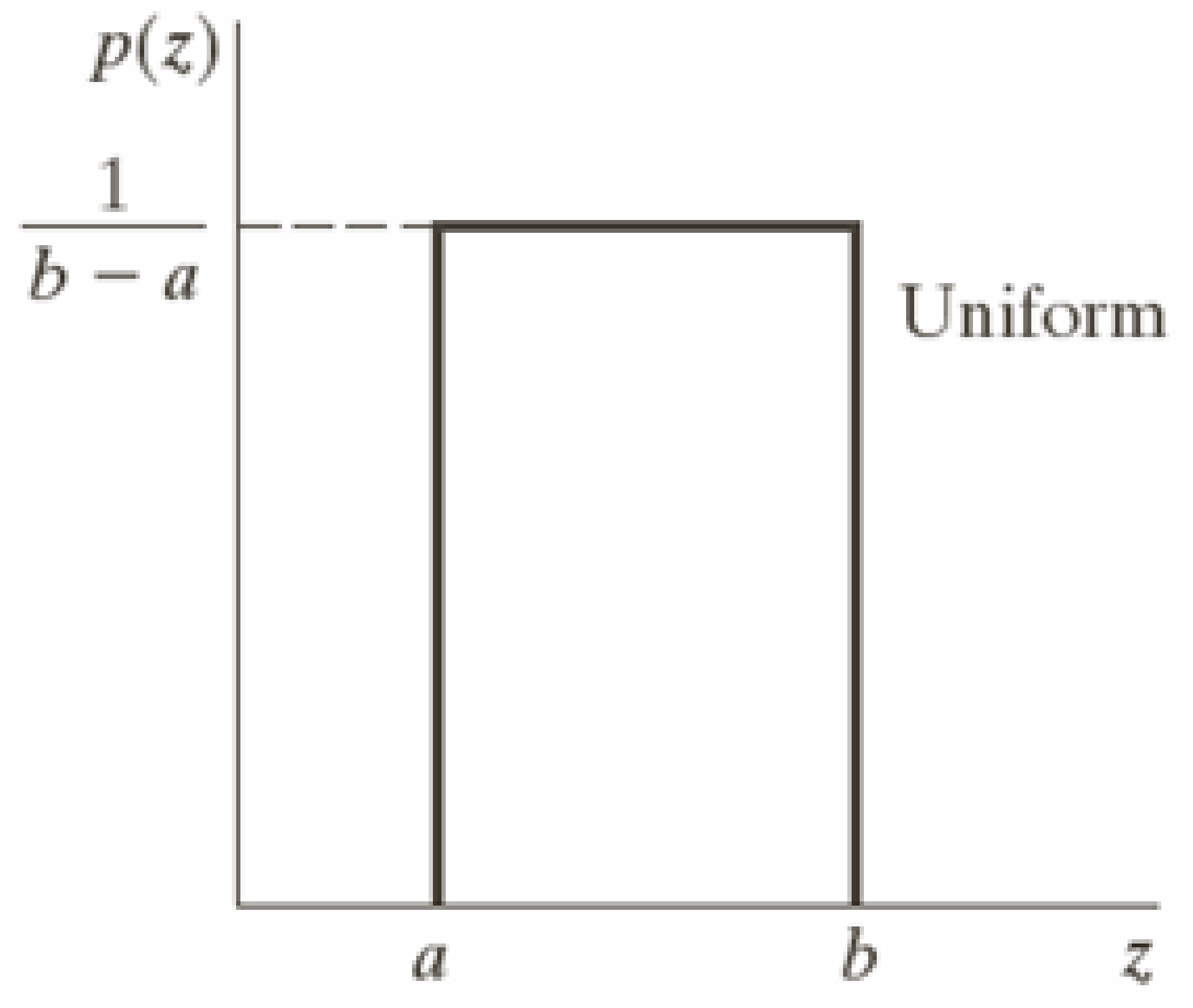
$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

where

z represents intensity

$$\bar{z} = \frac{a+b}{2}$$

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



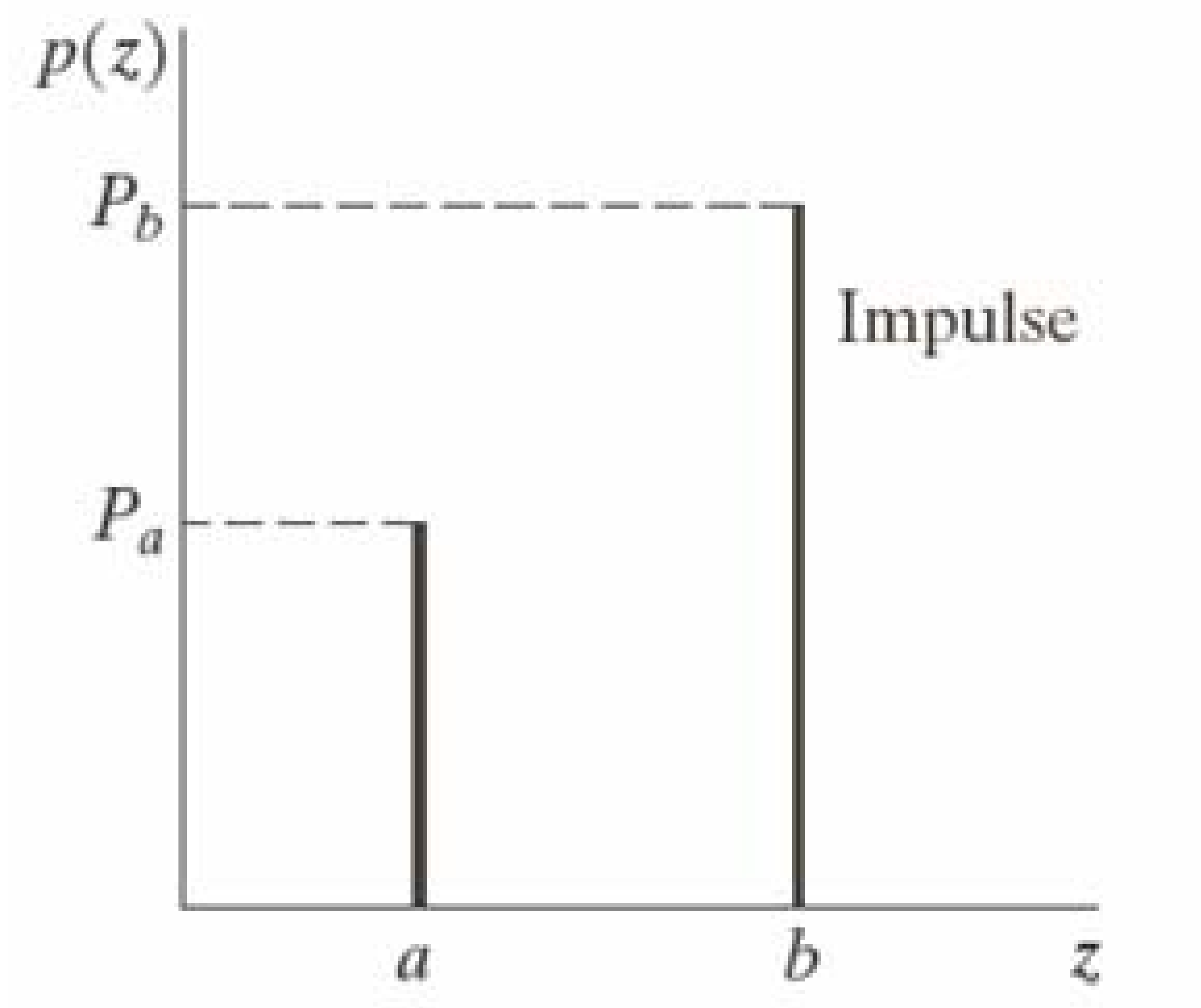
Noise Models

Impulse (salt-and-pepper) noise

- The PDF of (bipolar) impulse noise is given as:

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

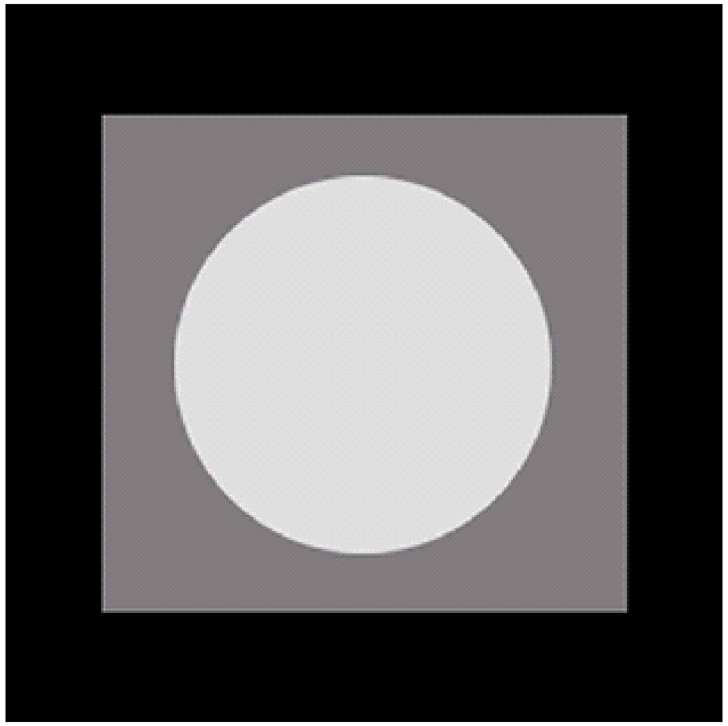
- If *b*>*a* then any pixel with intensity *b* will appear as a light dot in the image
- Pixels with intensity *a* will appear as a dark dot



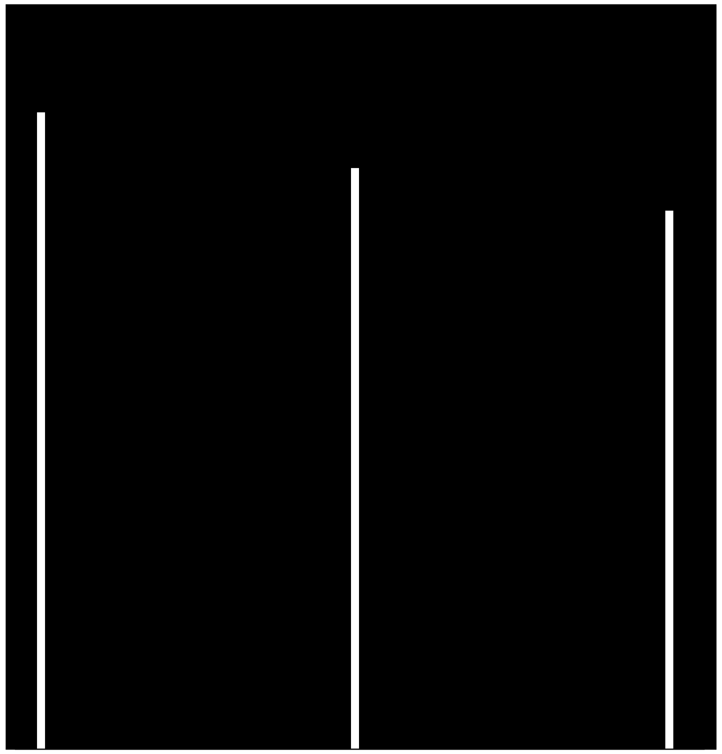
Noise Example

The test pattern to the right is ideal for demonstrating the addition of noise

The following slides will show the result of adding noise based on various models to this image

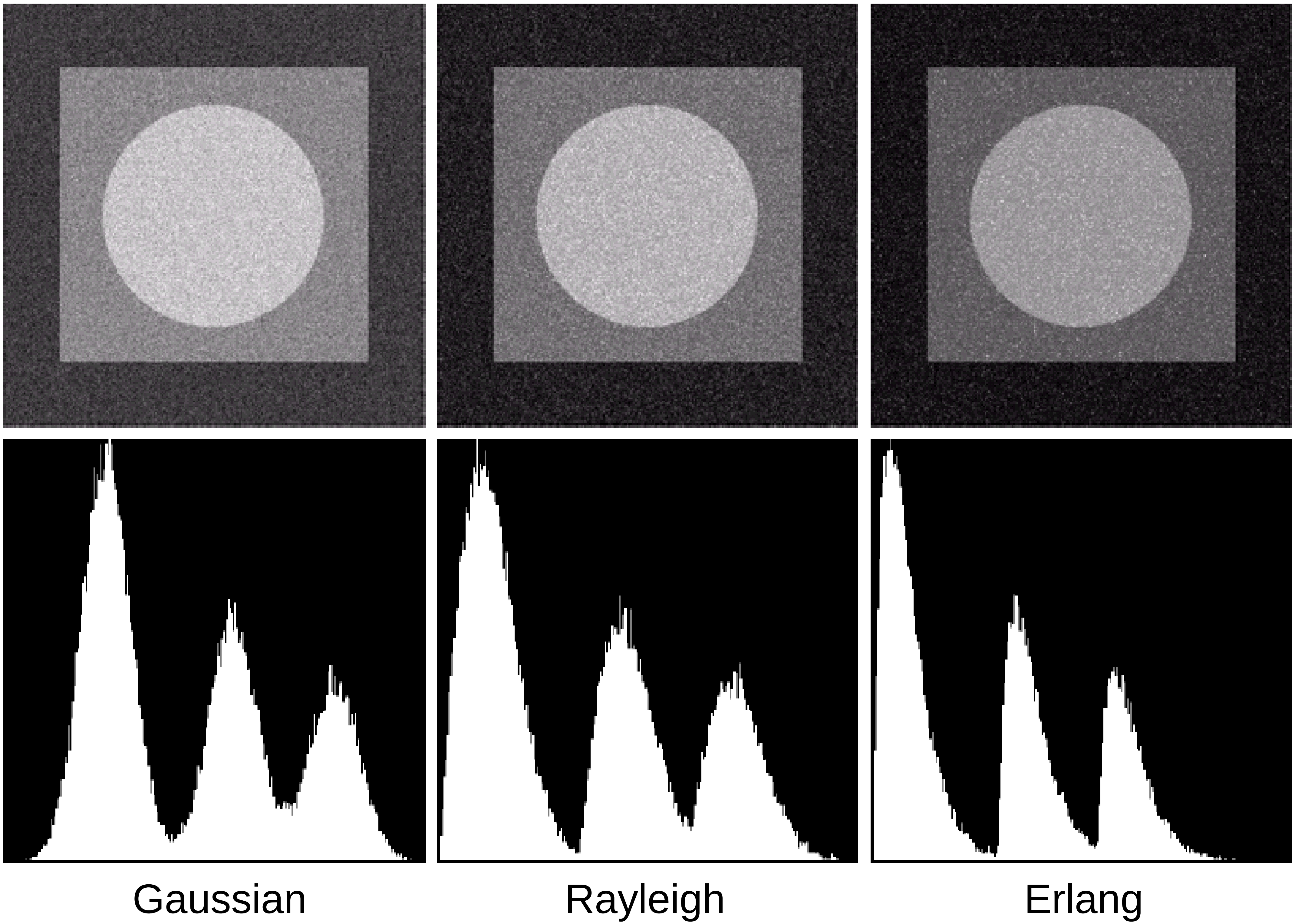


Image

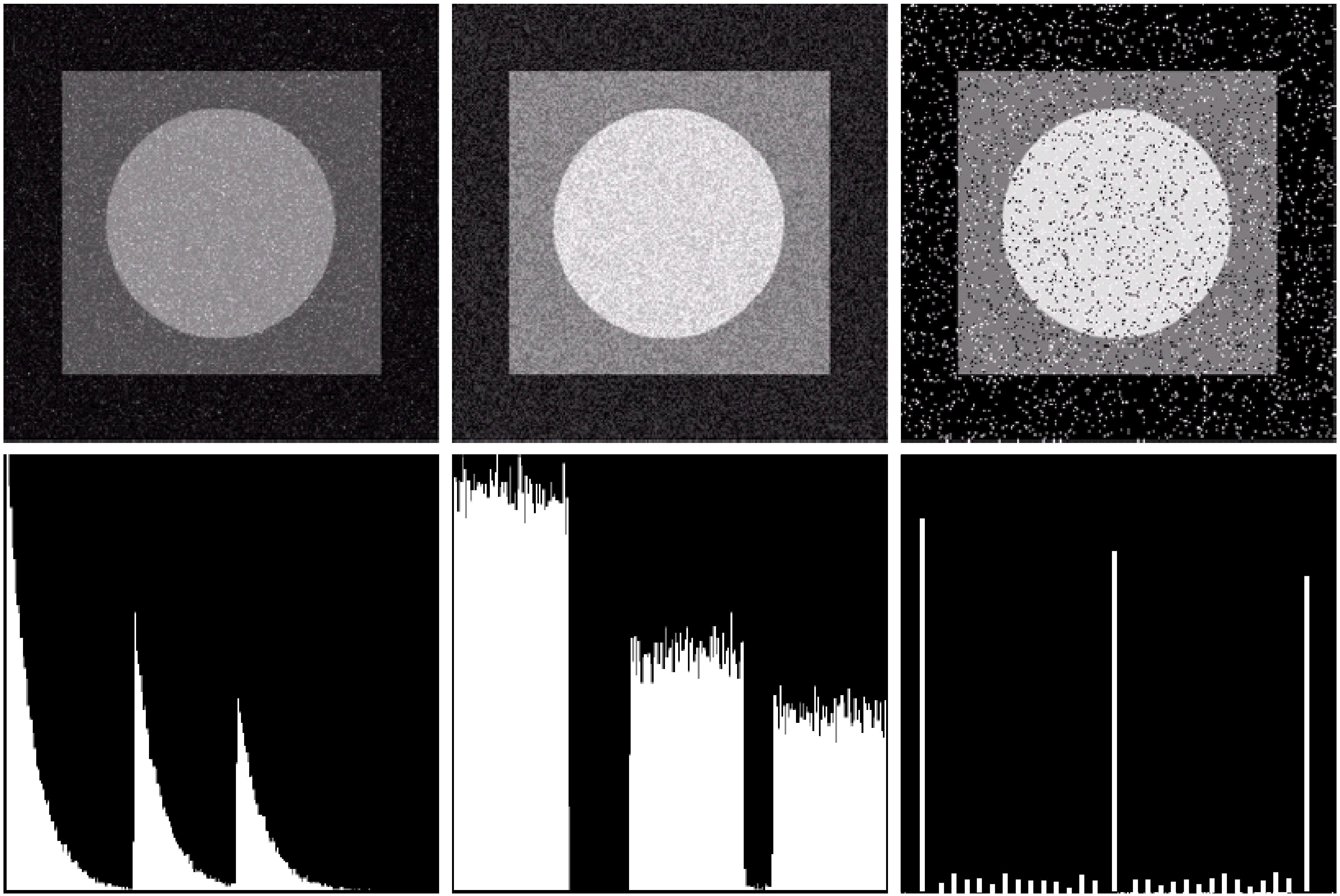


Histogram

Noise Example (cont…)



Noise Example (cont…)



Exponential Uniform Impulse

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Noise Models

Estimation of noise parameters (continued)

- If only a set of images already generated by a sensor are available, estimate the PDF function of the noise from small strips of reasonably constant background intensity
- Consider a subimage (*S*) and let
  $p_s(z_i)$, i=0,1,2, L-1
- denote the probability estimates of the intensities of the pixels in *S*.
- *L* is the number of possible intensities in the image
- The mean and the variance of the pixels in *S* are given by:

$$\bar{z} = \sum_{i=0}^{L-1} z_i p_s(z_i) \text{ and } \sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p_s(z_i)$$

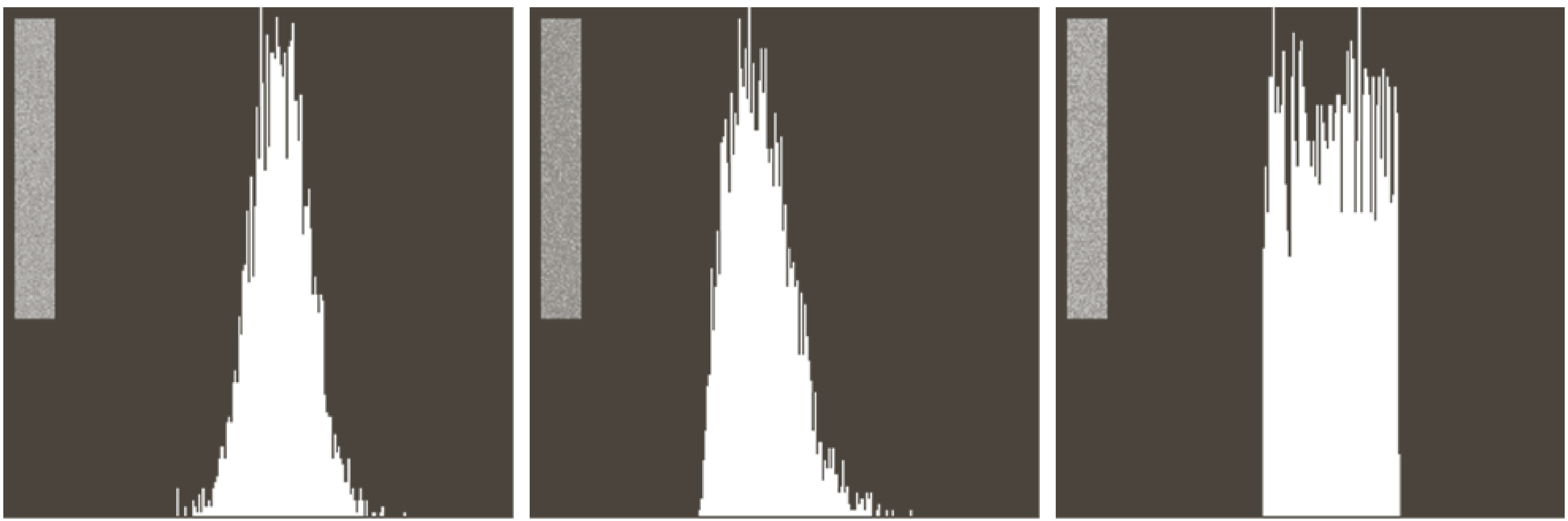
Noise Models

Estimation of noise parameters (continued)

- The shape of the noise histogram identifies the closest PDF match
  - If the shape is Gaussian, then the mean and variance are all that is needed to construct a model for the noise (i.e. the mean and the variance completely define the Gaussian PDF)
  - If the shape is Rayleigh, then the Rayleigh shape parameters (*a* and *b*) can be calculated using the mean and variance
  - If the noise is impulse, then a constant (with the exception of the noise) area of the image is needed to calculate P_a and P_b probabilities for the impulse PDF

Noise Models

Histograms from noisy strips of an area of an image



a b c

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

Filtering to Remove Noise

We can use spatial filters of different kinds to remove different kinds of noise

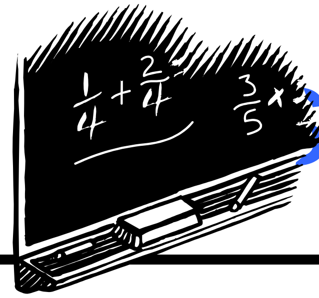
The *arithmetic mean* filter is a very simple one and is calculated as follows:

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

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

This is implemented as the simple smoothing filter

Blurs the image to remove noise

Noise Removal Example

Original Image

x

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
148	154	157	160	163	167	170
151	155	159	162	165	169	172

Image f (x, y)

y

Filtered Image

x

Image f (x, y)

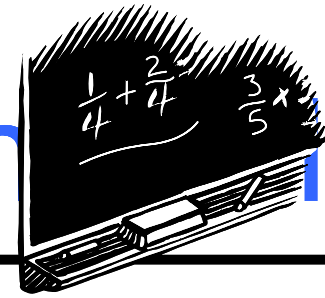
y

Other Means

There are different kinds of mean filters all of which exhibit slightly different behaviour:

- Geometric Mean
- Harmonic Mean
- Contraharmonic Mean

Other Means (cont…)

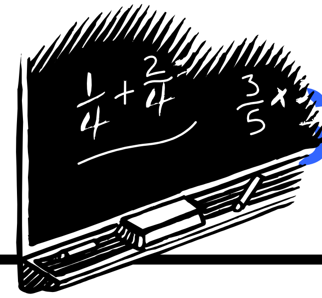


There are other variants on the mean which can give different performance

Geometric Mean:

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

Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail



se Removal Example

Original Image

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
148	154	157	160	163	167	170
151	155	159	162	165	169	172

x *y*

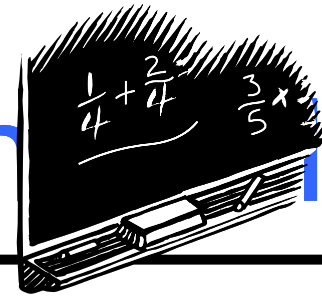
Image f (x, y)

Filtered Image

x *y*

Image f (x, y)

Other Means (cont…)



Harmonic Mean:

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

Works well for salt noise, but fails for pepper noise

Also does well for other kinds of noise such as Gaussian noise

Image Corruption Example

Original Image x

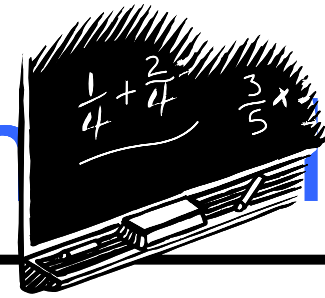
54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

Image f (x, y) y

Filtered Image x

Image f (x, y) y

Other Means (cont…)



Contraharmonic Mean:

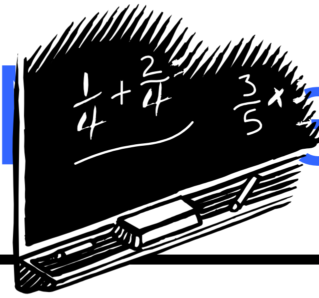
$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Q is the *order* of the filter and adjusting its value changes the filter's behaviour

Positive values of Q eliminate pepper noise

Negative values of Q eliminate salt noise

Image Corruption Example



Original Image

x →, *y* ↓

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

Image f (x, y)

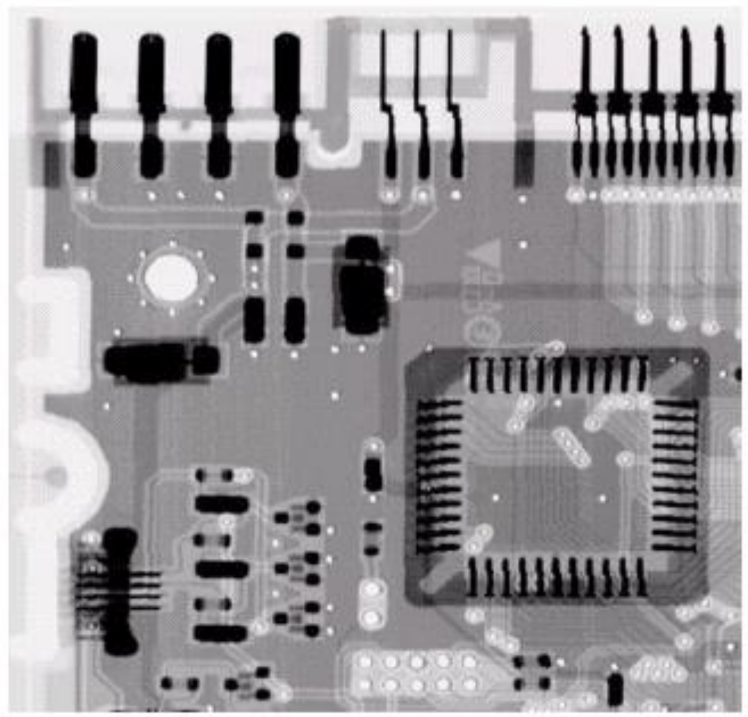
Filtered Image

x →, *y* ↓

Image f (x, y)

Noise Removal Examples

Original Image



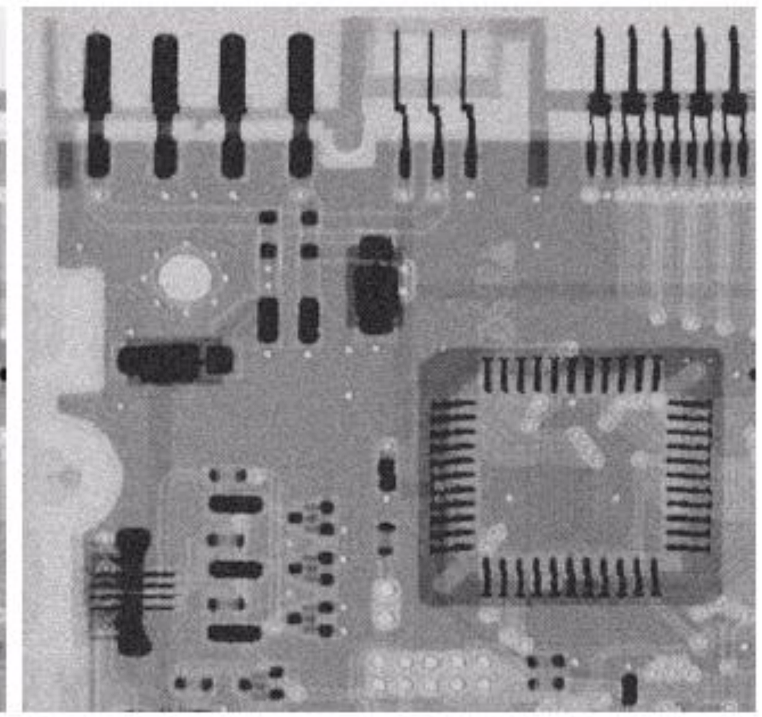
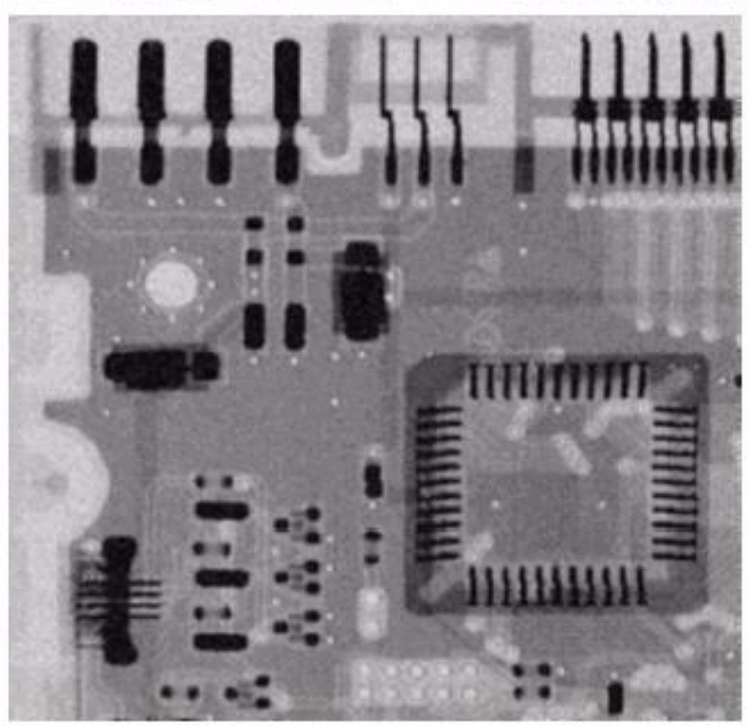
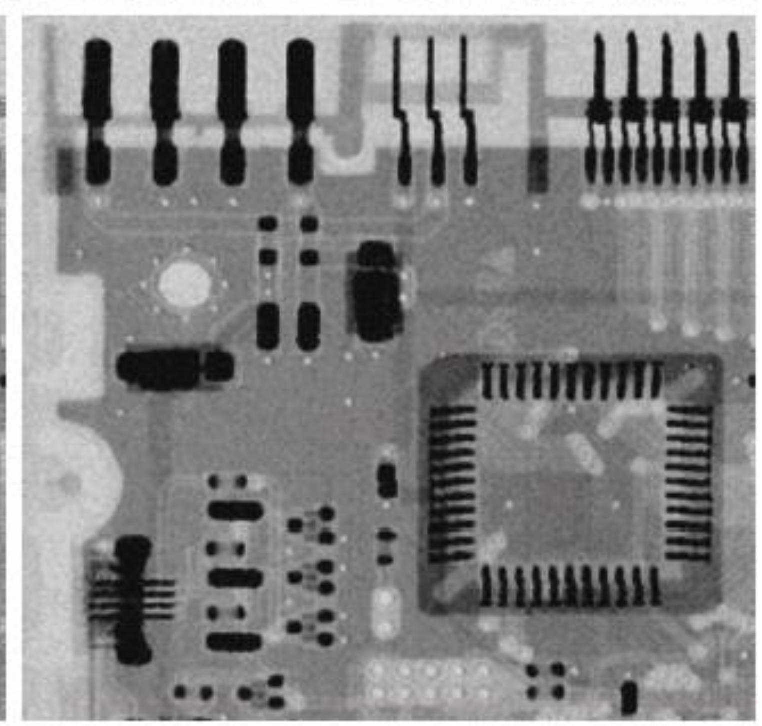


Image Corrupted By Gaussian Noise

After A 3*3 Arithmetic Mean Filter



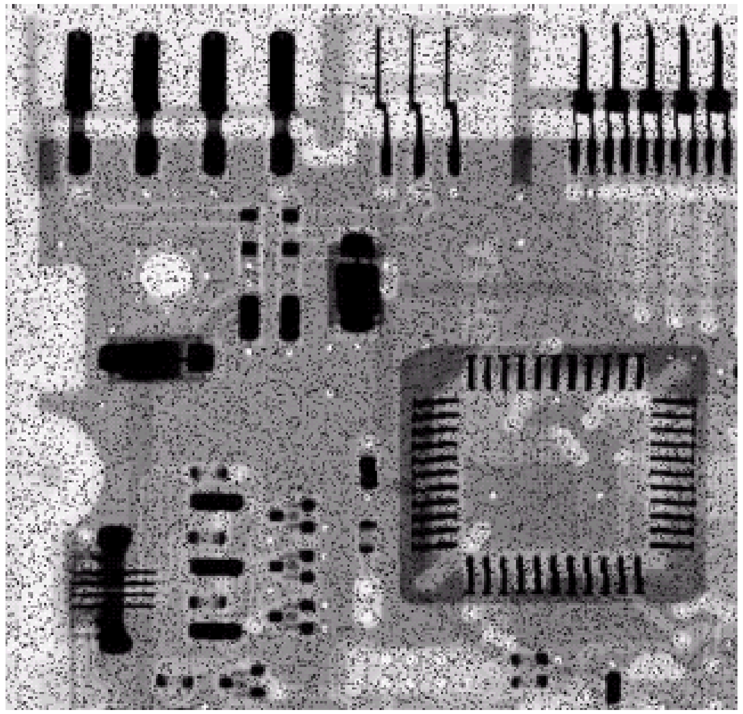


After A 3*3 Geometric Mean Filter

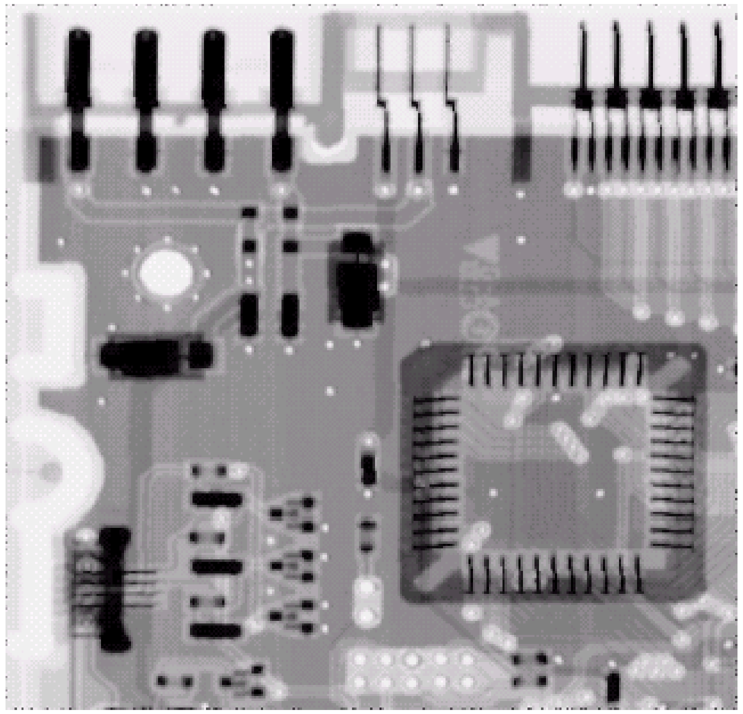
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



Result of Filtering Above With 3*3 Contraharmonic Q=1.5



Noise Removal Examples (cont…)

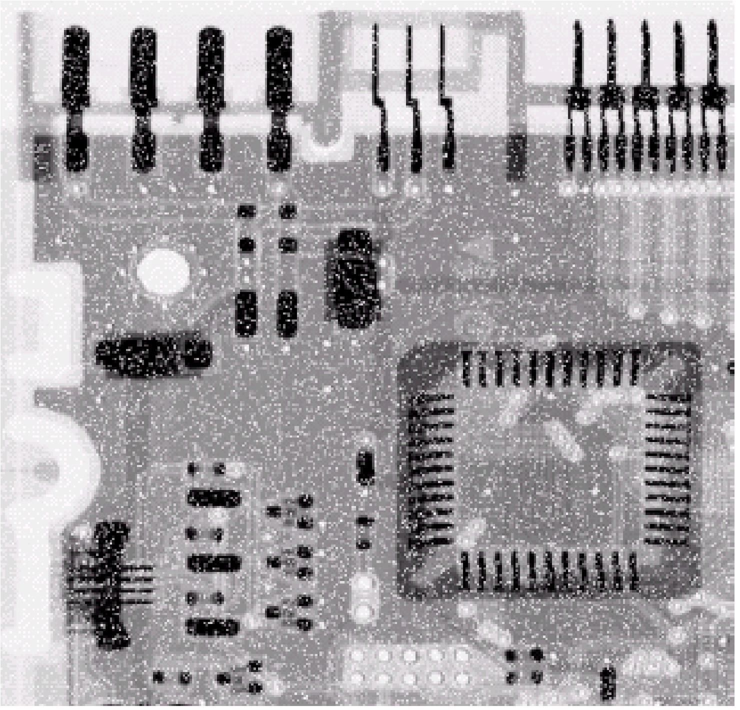
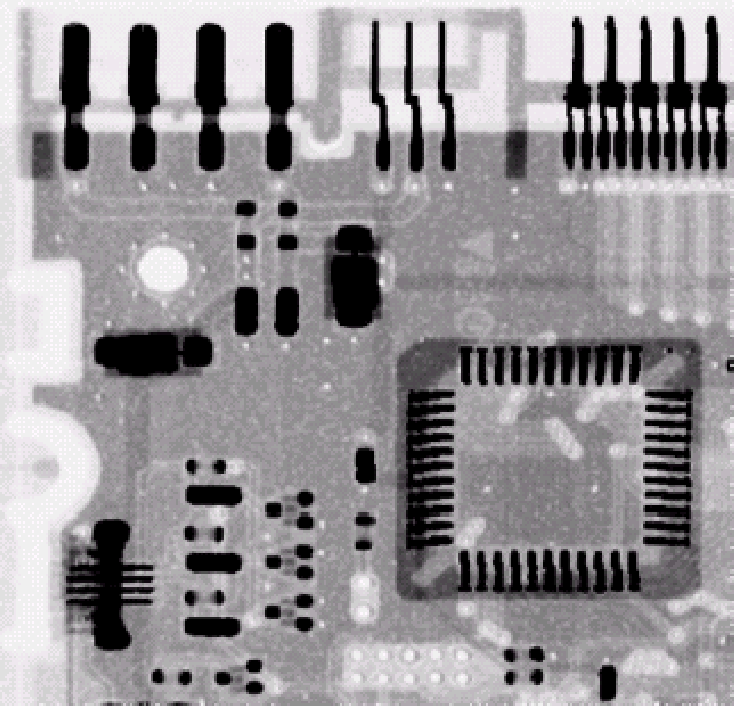


Image Corrupted By Salt Noise

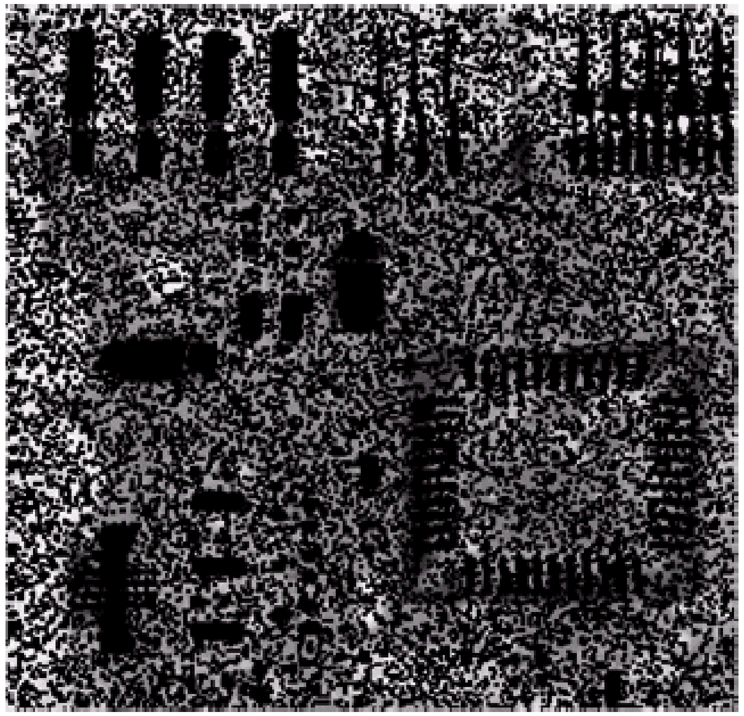


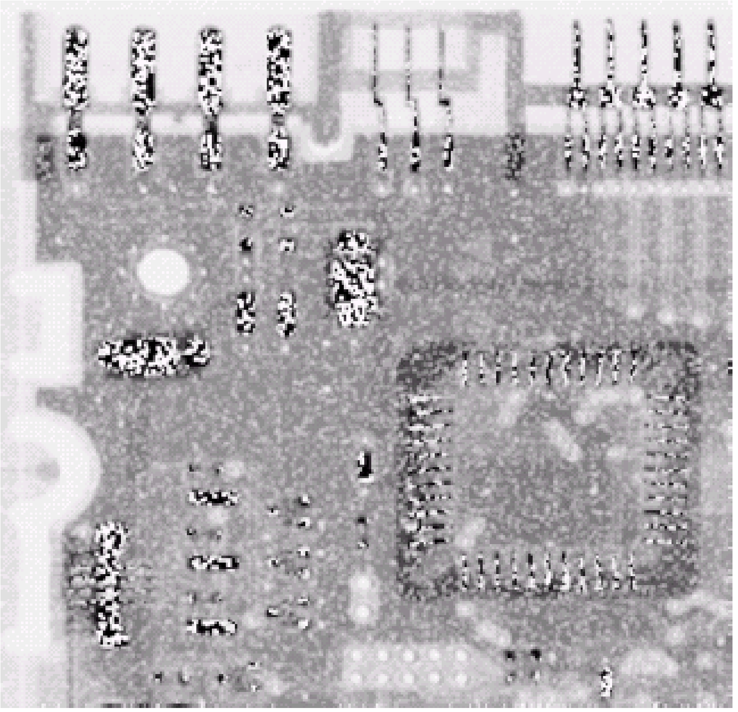
Result of Filtering Above With 3*3 Contraharmonic Q=-1.5

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Contraharmonic Filter: Here Be Dragons

Choosing the wrong value for Q when using the contraharmonic filter can have drastic results





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Order Statistics Filters

Spatial filters that are based on ordering the pixel values that make up the neighbourhood operated on by the filter

Useful spatial filters include

- Median filter
- Max and min filter
- Midpoint filter
- Alpha trimmed mean filter

Median Filter

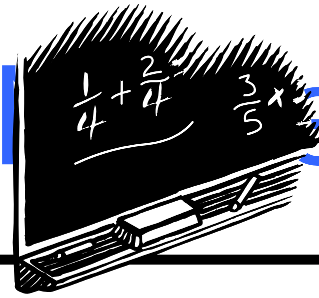
Median Filter:

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

Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters

Particularly good when salt and pepper noise is present

Image Corruption Example



Original Image

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

x *y* ***Image f (x, y)***

Filtered Image

x *y* ***Image f (x, y)***

Max and Min Filter

Max Filter:

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

Min Filter:

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

Max filter is good for pepper noise and min is good for salt noise

Image Corruption Example

Original Image

x

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

Image f (x, y)

y

Filtered Image

x

Image f (x, y)

y

Midpoint Filter

Midpoint Filter:

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

Good for random Gaussian and uniform noise

Image Corruption Example

Original Image ***x***

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

Image f (x, y)

y

Filtered Image ***x***

Image f (x, y)

y

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

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

We can delete the *d/2* lowest and *d/2* highest grey levels

So $g_r(s, t)$ represents the remaining $mn - d$ pixels

Alpha-Trimmed Mean Filter

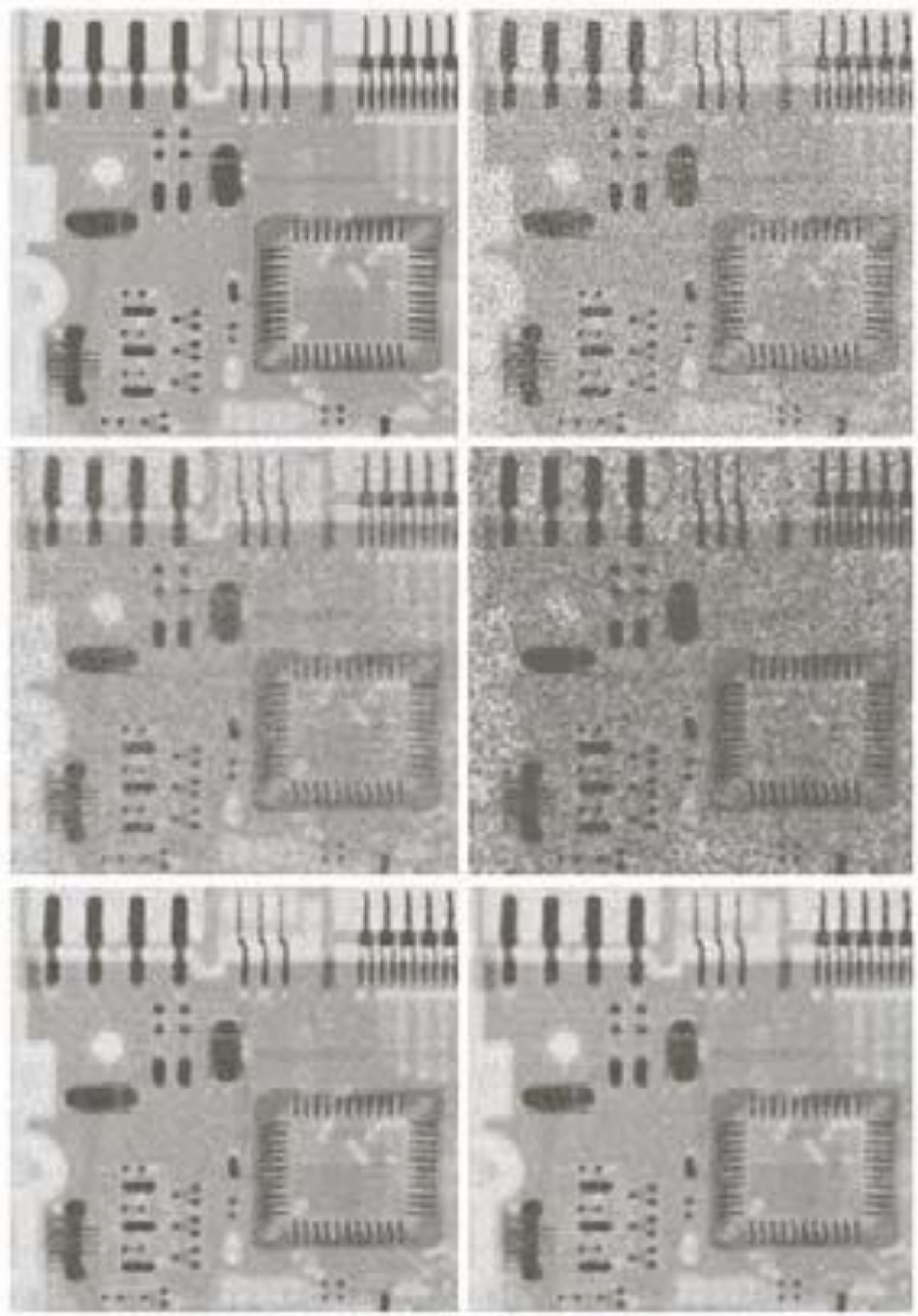
Alpha-trimmed mean filter

- If we delete the *d/2* lowest and the *d/2* highest intensity values from a neighborhood $g(s,t)$ of size $m*n$ and let $g_r(s,t)$ represent the remaining $mn-d$ pixels, the average of the remaining pixels is called an *alpha-trimmed mean* filter and is given by:

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

- d can vary from 0 to mn-1
- If d=0 the filter becomes the arithmetic mean filter
- If d=mn-1, the filter reduces to a median filter

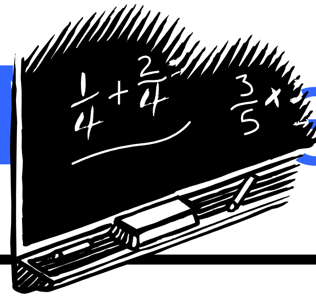
Alpha-Trimmed Mean Filter



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 $d = 5$.

Image Corruption Example



Original Image *x*

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

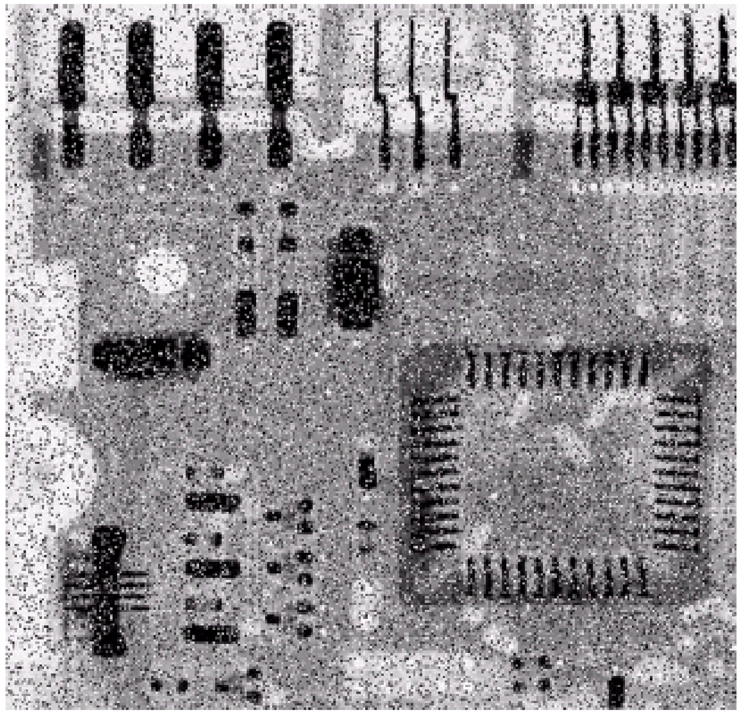
y **_Image f (x, y)_**

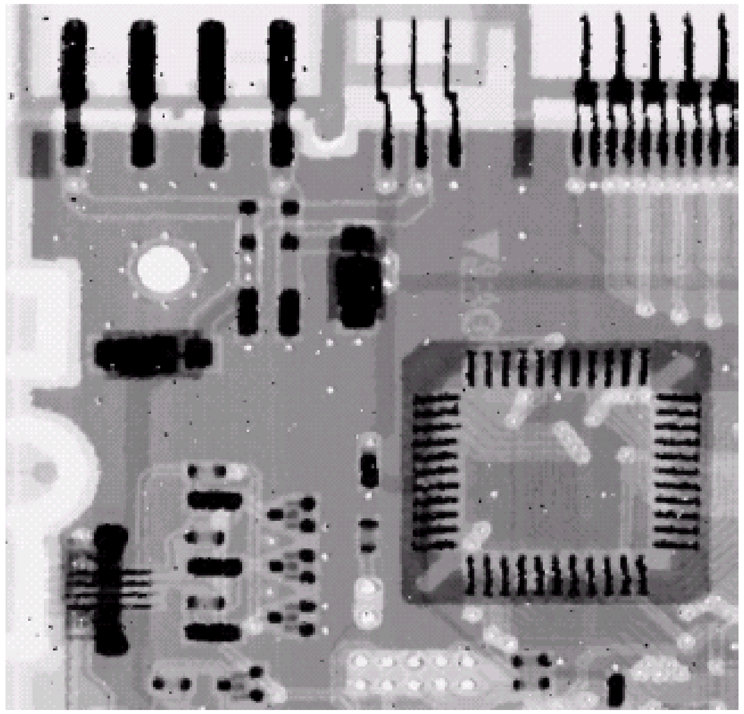
Filtered Image *x*

y **_Image f (x, y)_**

Noise Removal Examples

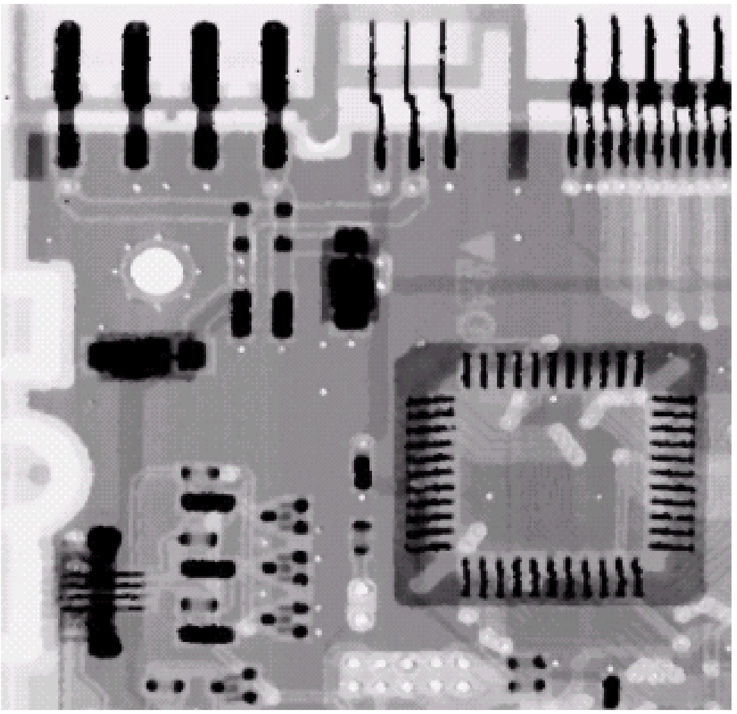
Image Corrupted By Salt And Pepper Noise

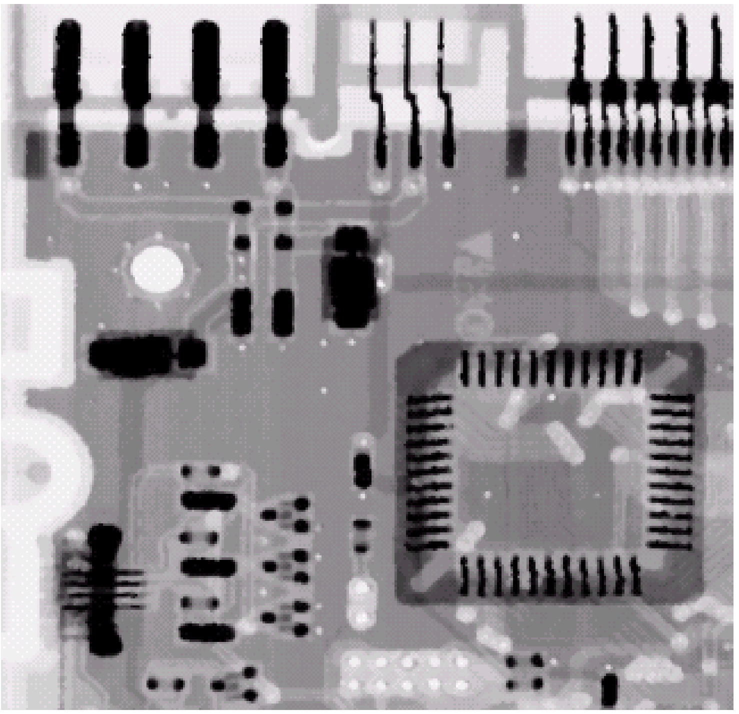




Result of 1 Pass With A 3*3 Median Filter

Result of 2 Passes With A 3*3 Median Filter



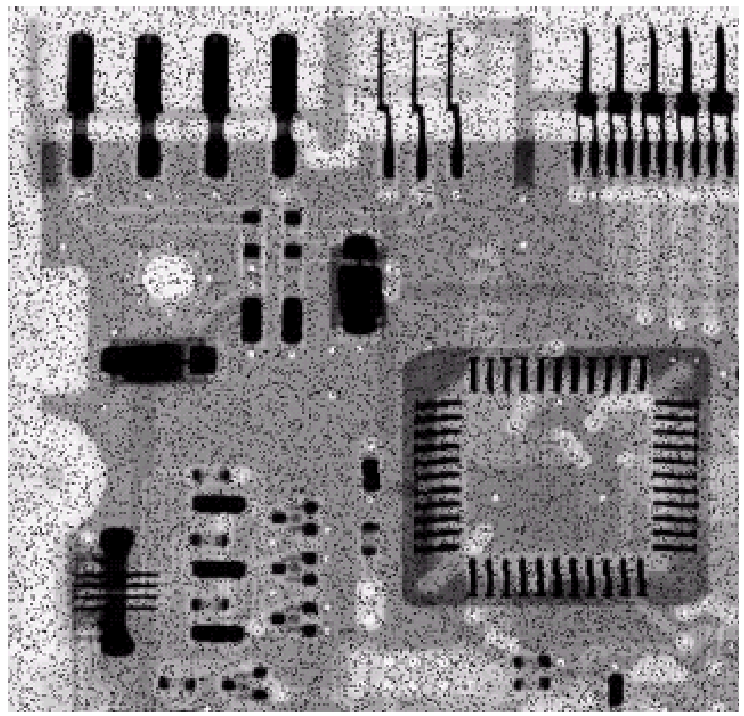


Result of 3 Passes With A 3*3 Median Filter

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



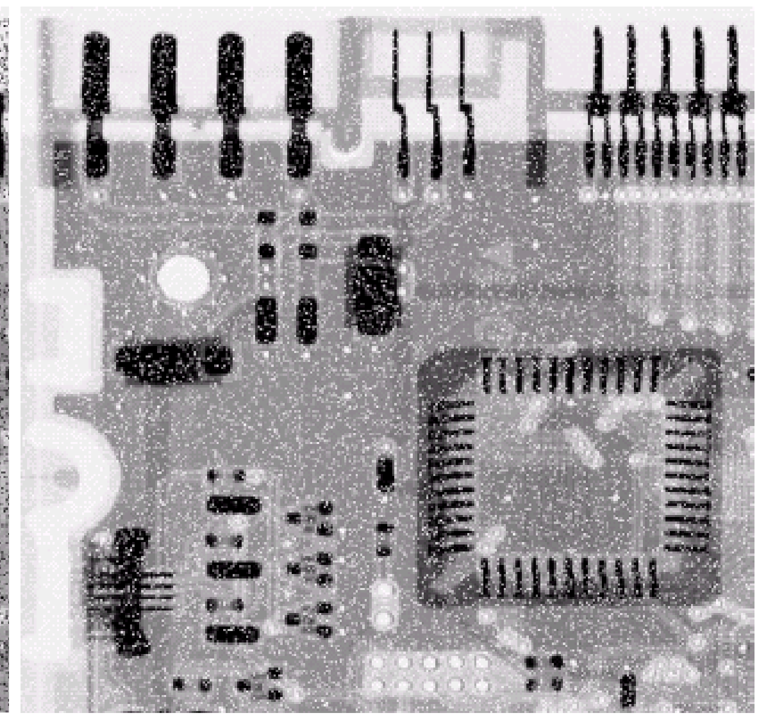
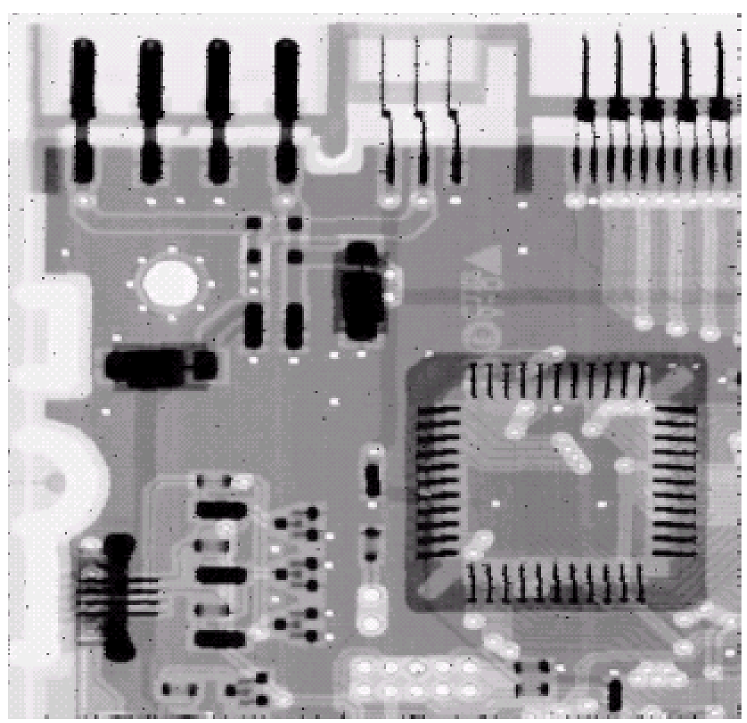
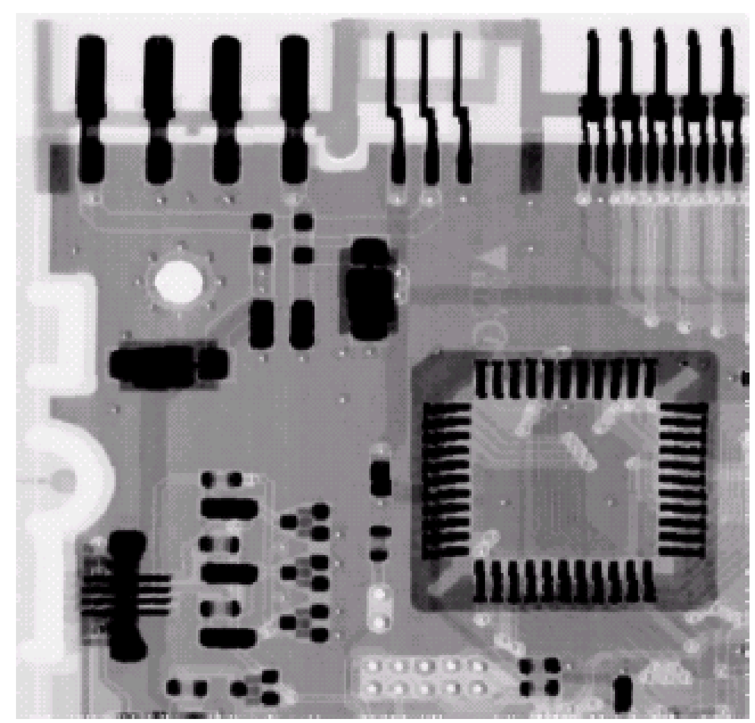


Image Corrupted By Salt Noise

Result Of Filtering Above With A 3*3 Max Filter





Result Of Filtering Above With A 3*3 Min Filter

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Noise Removal Examples (cont…)

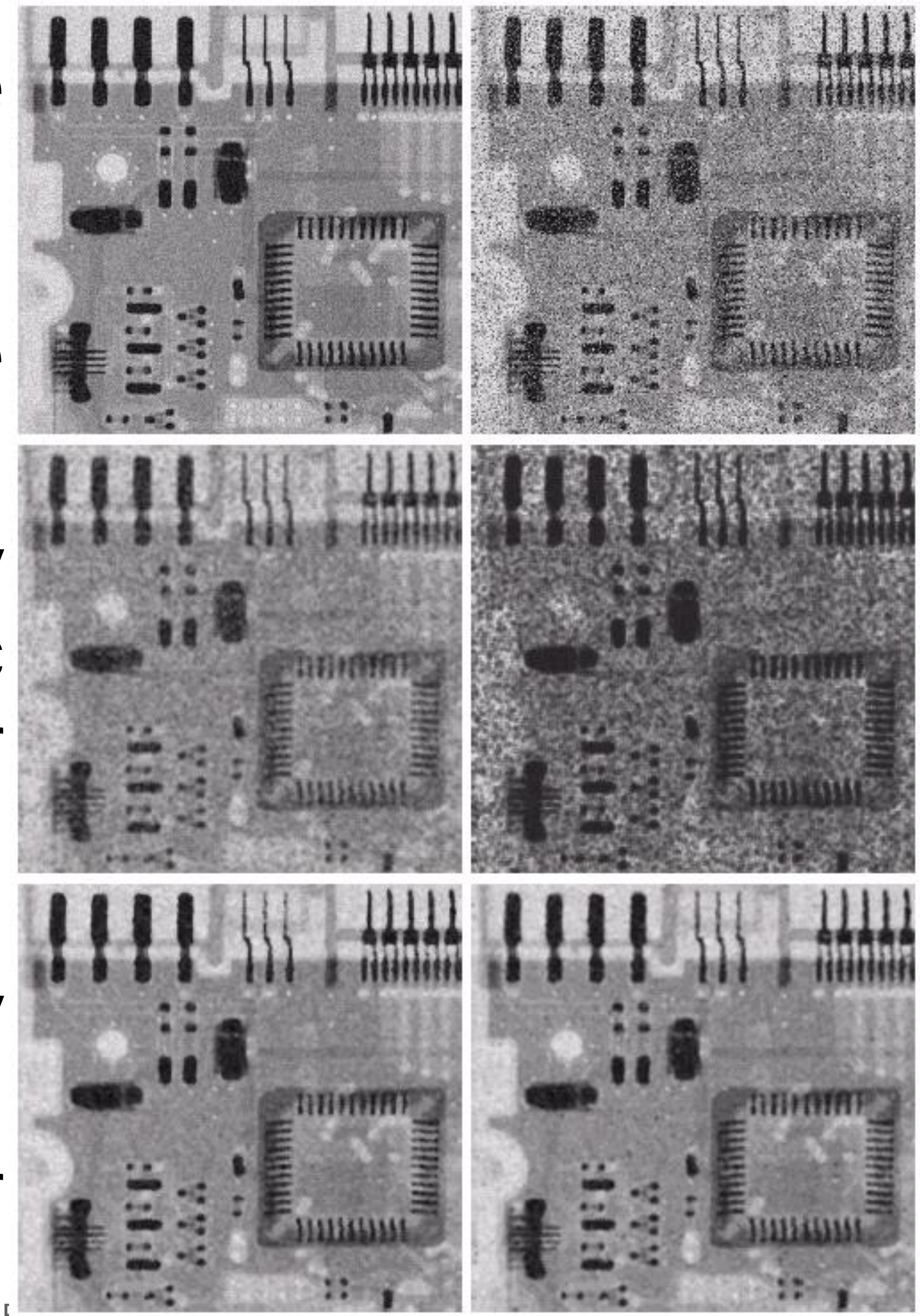


Image Corrupted By Uniform Noise

Image Further Corrupted By Salt and Pepper Noise

Filtered By 5*5 Arithmetic Mean Filter

Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Median Filter

Filtered By 5*5 Alpha-Trimmed Mean Filter

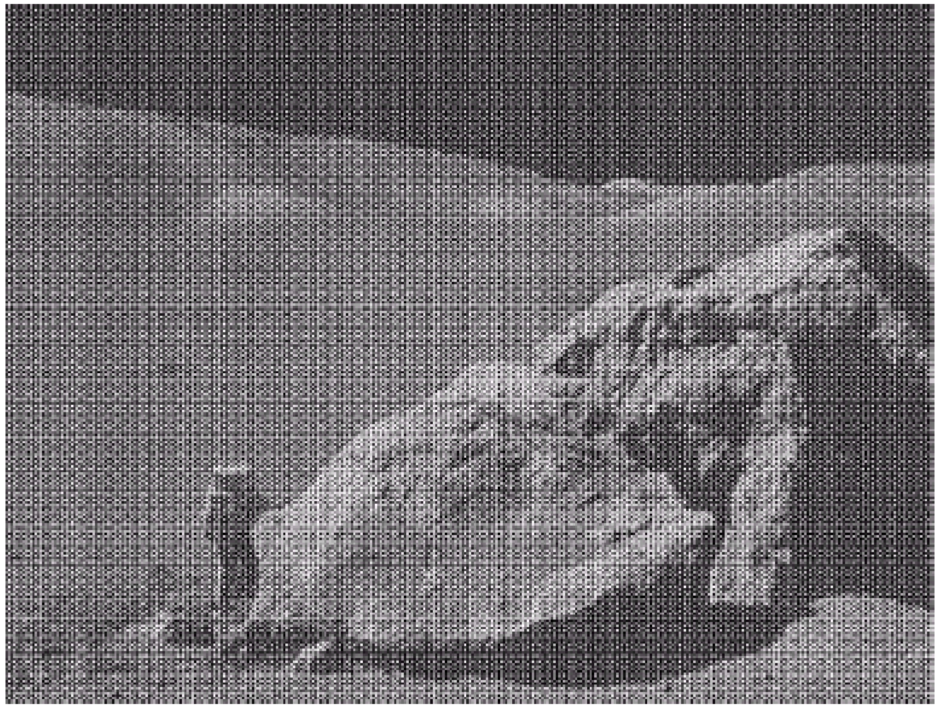
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

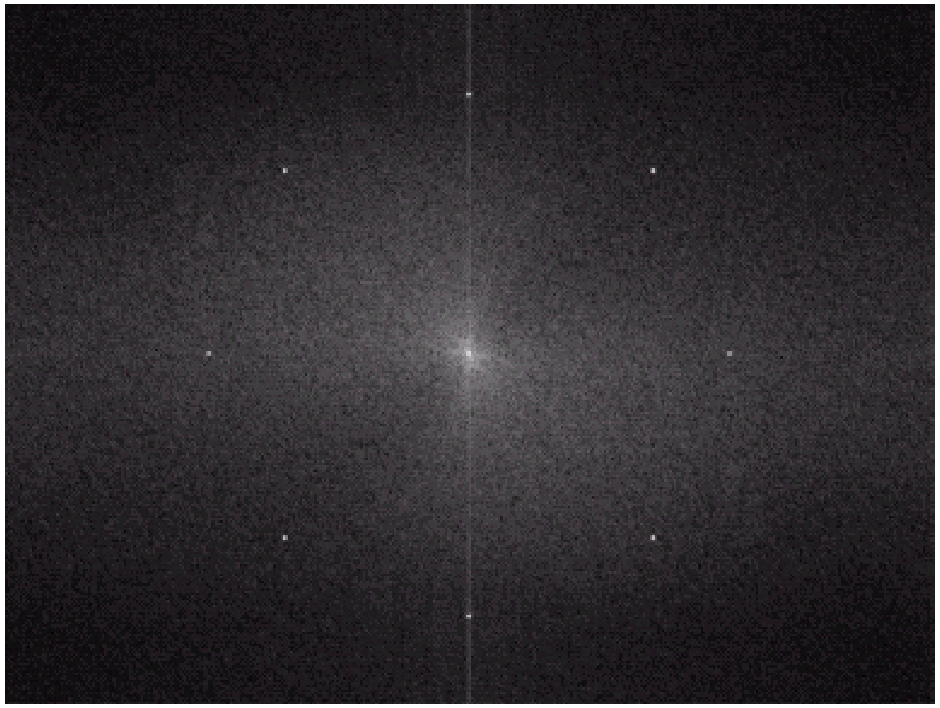
Periodic Noise

Typically arises due to electrical or electromagnetic interference

Gives rise to regular noise patterns in an image

Frequency domain techniques in the Fourier domain are most effective at removing periodic noise





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Band Reject Filters

Removing periodic noise form an image involves removing a particular range of frequencies from that image

Band reject filters can be used for this purpose

An ideal band reject filter is given as follows:

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

Band Reject Filters

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

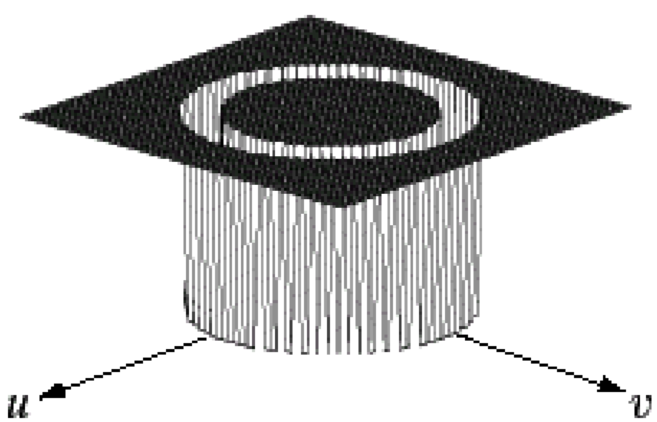
$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)W}{D^2(u, v) - D_0^2} \right]^{2n}} \tag{5.4-2}$$

and a Gaussian bandreject filter is given by

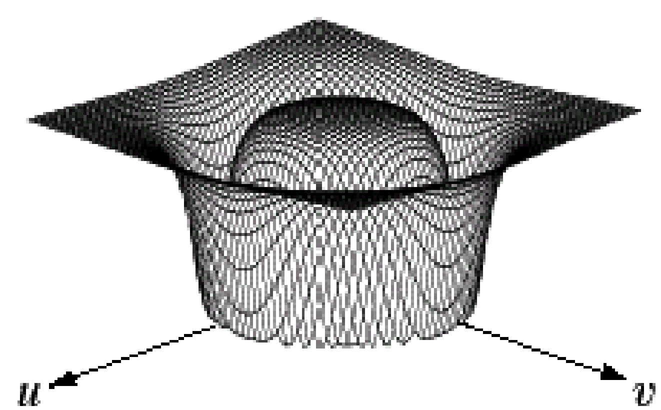
$$H(u, v) = 1 - e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)W}\right]^2}. \tag{5.4-3}$$

Band Reject Filters (cont…)

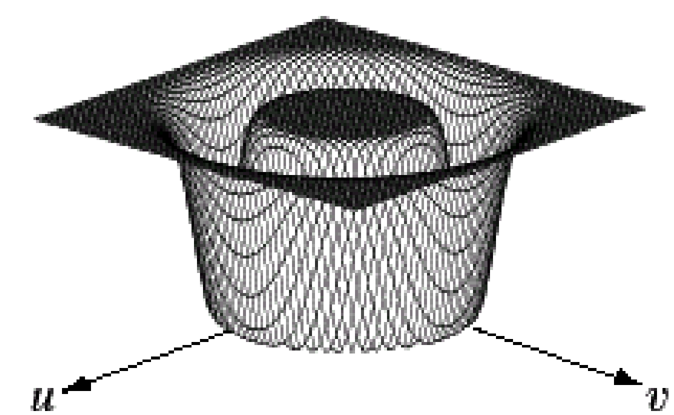
The ideal band reject filter is shown below, along with Butterworth and Gaussian versions of the filter



Ideal Band Reject Filter



Butterworth Band Reject Filter (of order 1)

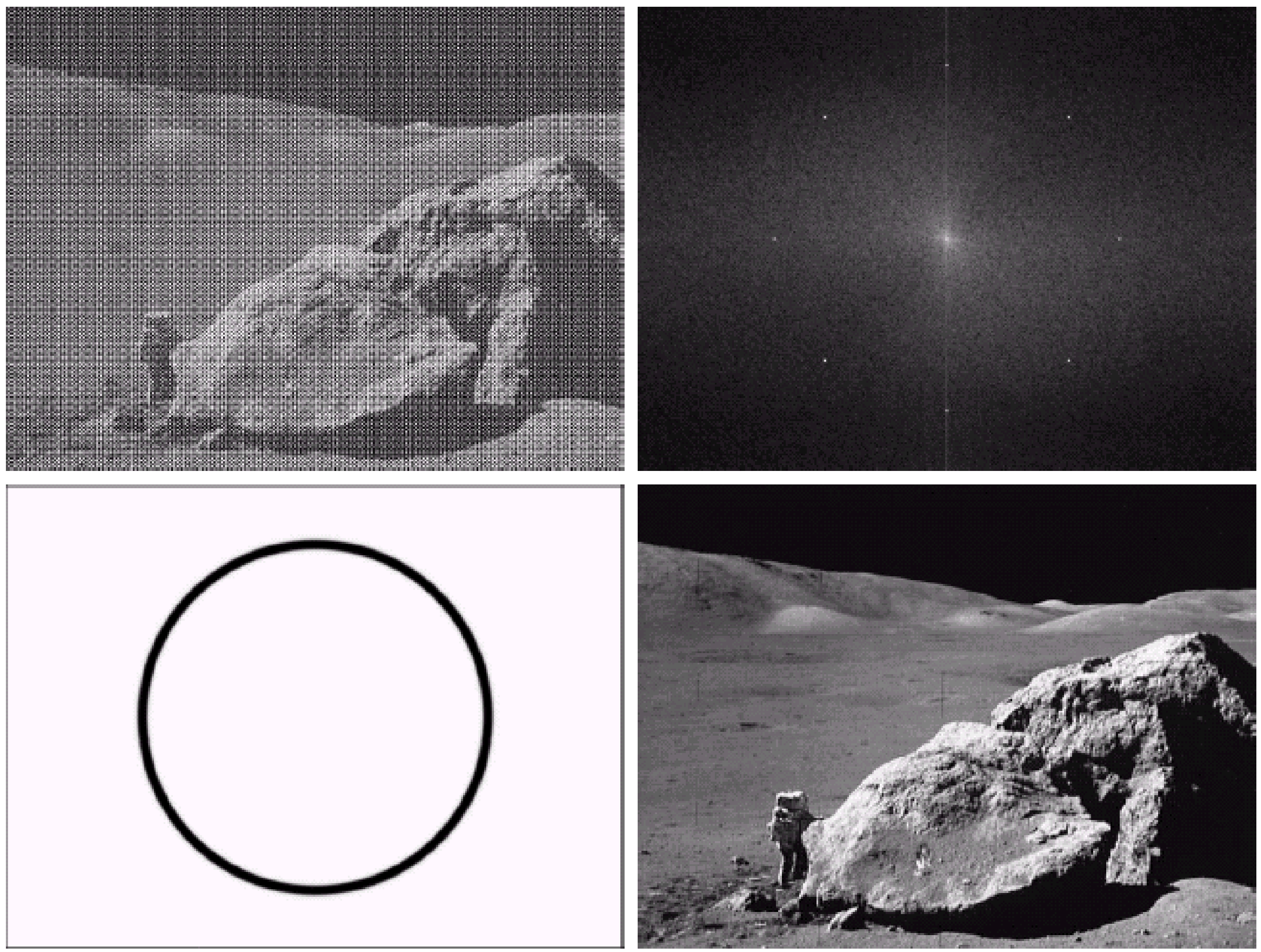


Gaussian Band Reject Filter

Band Reject Filter Example

Image corrupted by sinusoidal noise

Fourier spectrum of corrupted image



Butterworth band reject filter

Filtered image

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Summary

In this lecture we will look at image restoration for noise removal

Restoration is slightly more objective than enhancement

Spatial domain techniques are particularly useful for removing random noise

Frequency domain techniques are particularly useful for removing periodic noise

Adaptive Filters

The filters discussed so far are applied to an entire image without any regard for how image characteristics vary from one point to another

The behaviour of **adaptive filters** changes depending on the characteristics of the image inside the filter region

We will take a look at the **adaptive median filter**

Adaptive Filters

Adaptive filters

- All filters considered thus far are applied to an image without regard for how image characteristics may vary from one point to another in the image
- An *adaptive* filter is one whose behavior can change based on statistical characteristics of an area within the image
  - This is typically the $m*n$ filter region in the $S_{x,y}$ window
- Generally provides superior performance at the cost of increased filter complexity

Adaptive Filters

Adaptive, local noise reduction filter

- The *mean* and *variance* are reasonable parameters upon which to base a simple adaptive filter
  - They are closely related to image properties
  - The *mean* gives the average intensity over a region
  - The *variance* gives a measure of the contrast in a region
- A simple filter will operate on a local region $S_{x,y}$ with the response at any point (x,y) base on four quantities:
  - The value of the noisy image at (x,y): $\boldsymbol{g(x,y)}$
  - The variance of the noise corrupting $f(x,y)$ to form $g(x,y)$: σ^2_η
  - The local mean of the pixels in $S_{x,y}$: m_L
  - The local variance of the pixels in $S_{x,y}$: σ^2_L

Adaptive Filters

Adaptive, local noise reduction filter algorithm

- If $\sigma^2_\eta=0$, return the value $g(x,y)$
  - This is the zero-noise case where $g(x,y)= f(x,y)$
- If the local variance (σ^2_L) is high relative to σ^2_η, return a value close to $g(x,y)$
  - A high local variance is generally associated with image features (*i.e.* an edge, etc.) and should be preserved
- If $\sigma^2_L = \sigma^2_\eta$, return the arithmetic mean of the pixels in $S_{x,y}$
  - This occurs if the local area has the same properties as the overall image. Local noise is reduced by averaging.

Adaptive Filters

Adaptive, local noise reduction filter equation

- An adaptive expression may be written as:

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

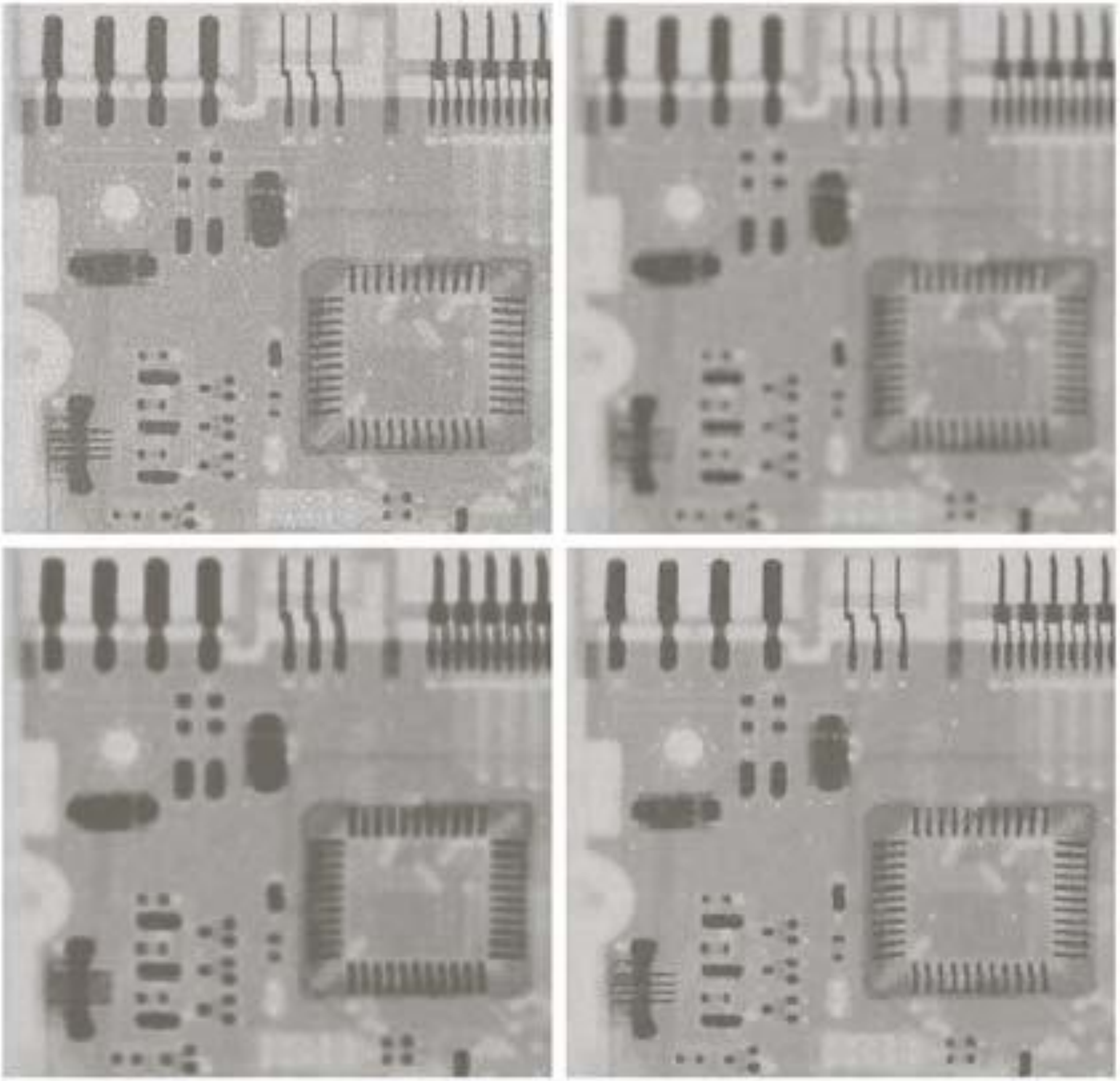
- The only quantity that must be known is σ^2_η
- Everything else can be computed from $S_{x,y}$
- An assumption here is that $\sigma^2_\eta \leq \sigma^2_L$
  - This is generally reasonable given that the noise we are considering is additive and position independent
  - If this is not true then a simple test could set the ratio of the variances to one if $\sigma^2_\eta > \sigma^2_L$

Adaptive Filters

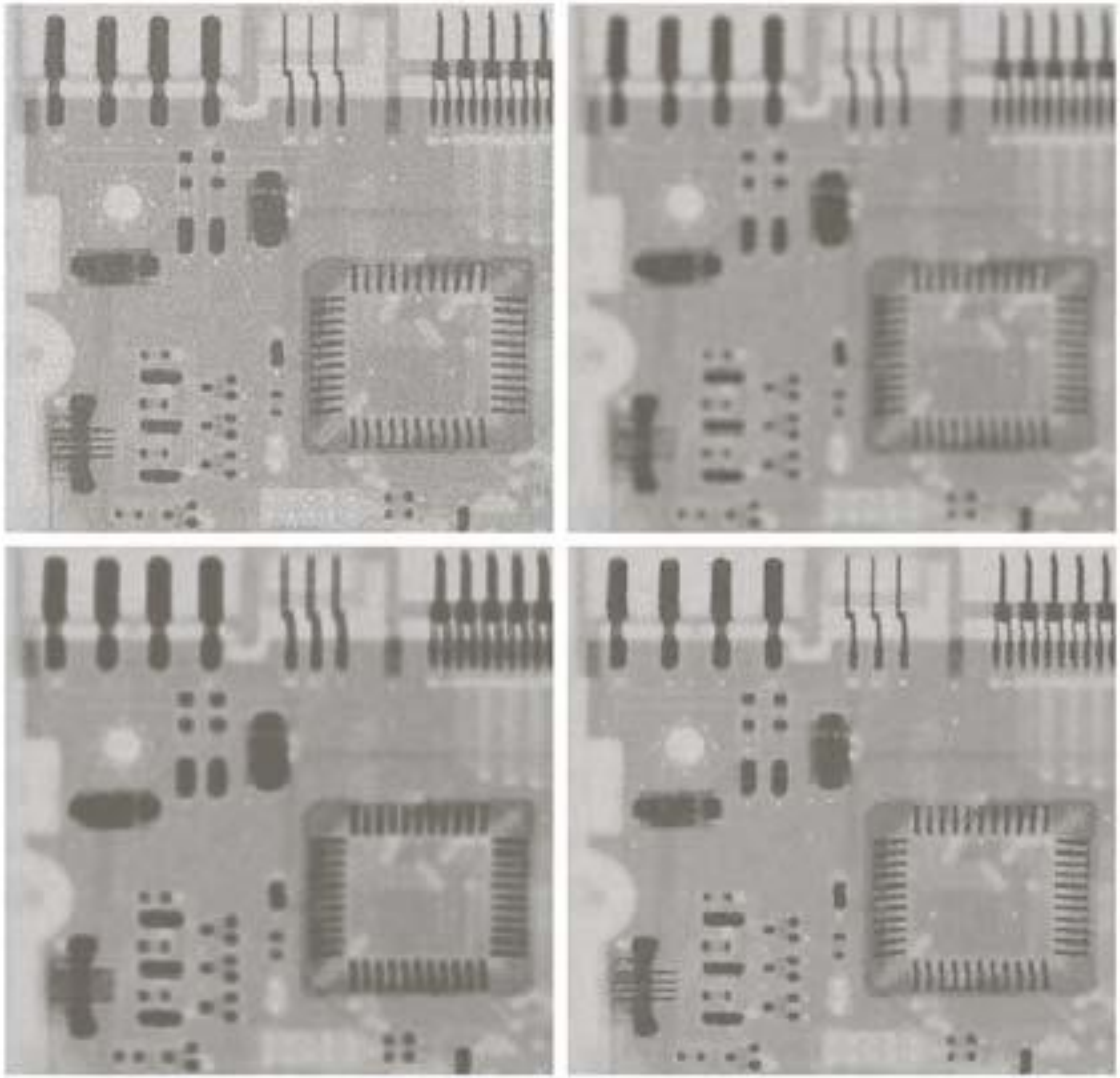
Adaptive, local noise reduction filter example

a b
c d

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



Adaptive Filters



a b
c d

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

Adaptive Median Filtering

Adaptive median filter

- A median filter works well in the spectral density of the impulse noise is not large
  - A P_a and P_b less than 0.2 is a good general rule of thumb
- An adaptive median filter can handle noise with probabilities greater than these
- An additional benefit is that the adaptive median filter attempts to preserve detail while smoothing the impulse noise
- The adaptive median filter works in a rectangular window area $S_{x,y}$
  - The size of $S_{x,y}$ is not fixed
- The output of the filter is a single value that will be used to replace the center value of $S_{x,y}$

Adaptive Median Filtering

The median filter performs relatively well on impulse noise as long as the spatial density of the impulse noise is not large

The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise

The key insight in the adaptive median filter is that the filter size changes depending on the characteristics of the image

Adaptive Median Filtering (cont…)

Remember that filtering looks at each original pixel image in turn and generates a new filtered pixel

First examine the following notation:

- z_{min} = minimum grey level in S_{xy}
- z_{max} = maximum grey level in S_{xy}
- z_{med} = median of grey levels in S_{xy}
- z_{xy} = grey level at coordinates (x, y)
- S_{max} =maximum allowed size of S_{xy}

Adaptive Median Filtering (cont…)

Level A: $A1 = z_{med} - z_{min}$

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

If $A1 > 0$ and $A2 < 0$, Go to level B

Else increase the window size

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

Else output z_{med}

Level B: $B1 = z_{xy} - z_{min}$

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

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

Else output z_{med}

Adaptive Median Filtering (cont…)

The key to understanding the algorithm is to remember that the adaptive median filter has three purposes:

- Remove impulse noise
- Provide smoothing of other noise
- Reduce distortion

Adaptive Filtering Example

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

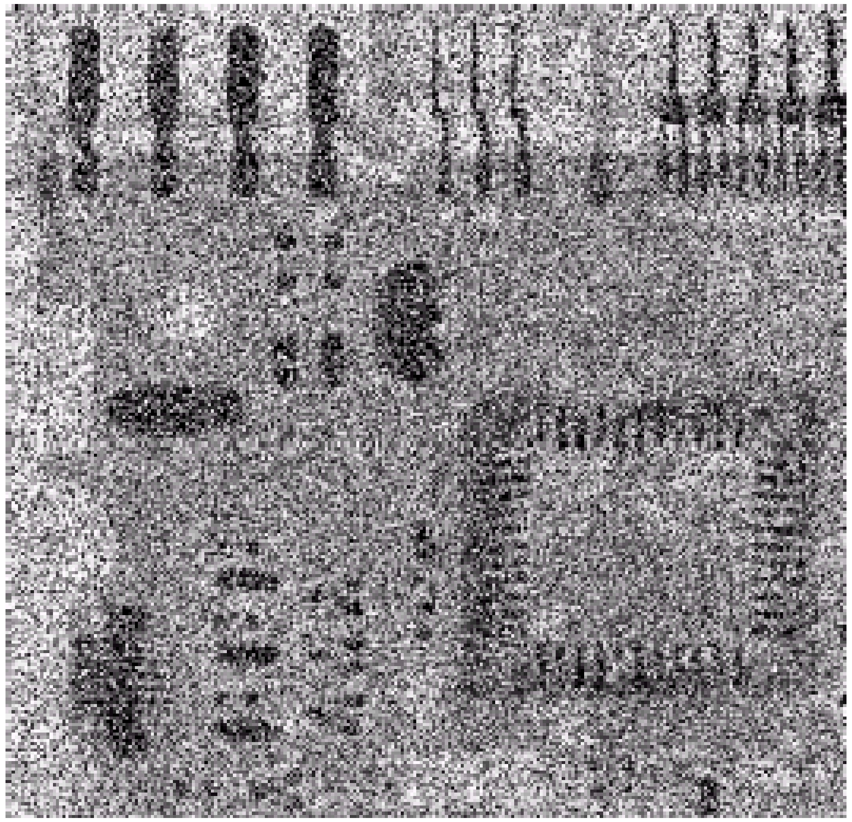
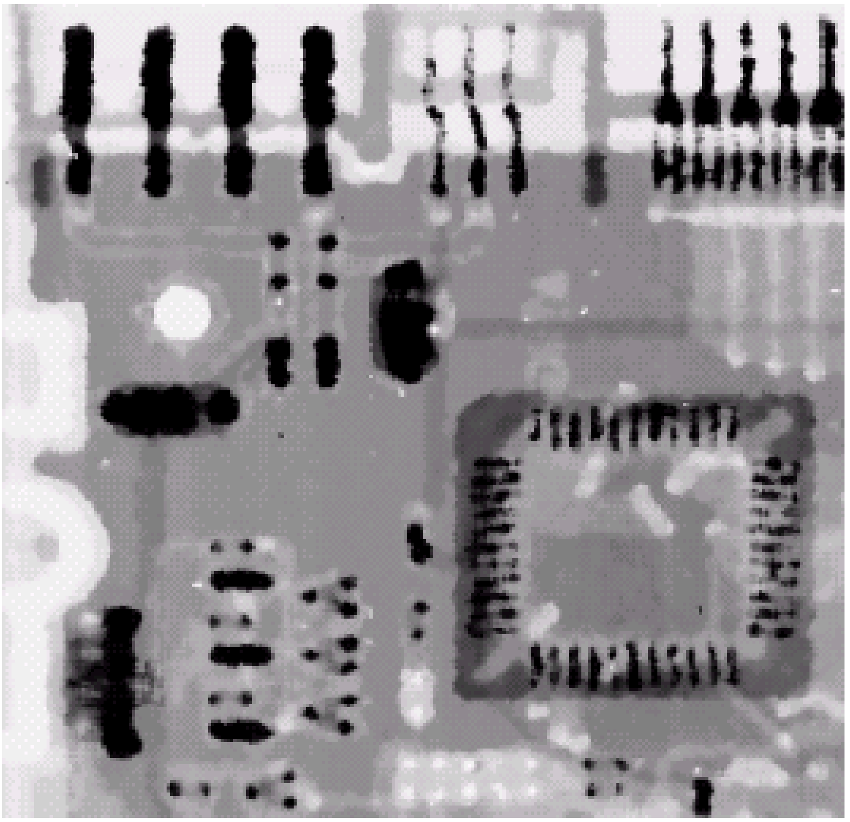
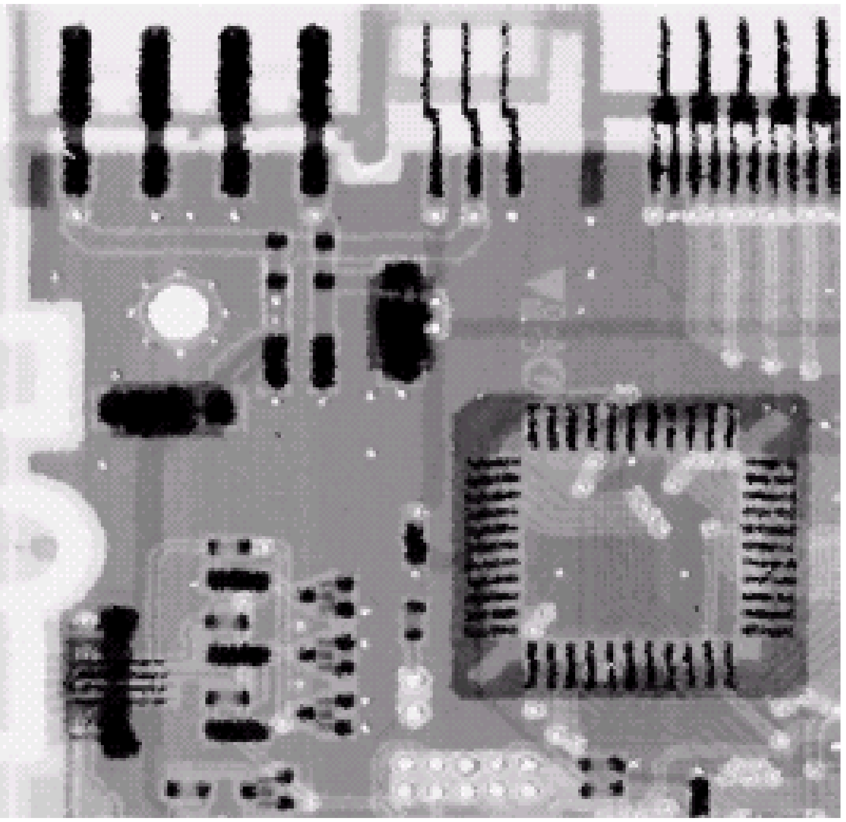


Image corrupted by salt and pepper noise with probabilities $P_a = P_b = 0.25$

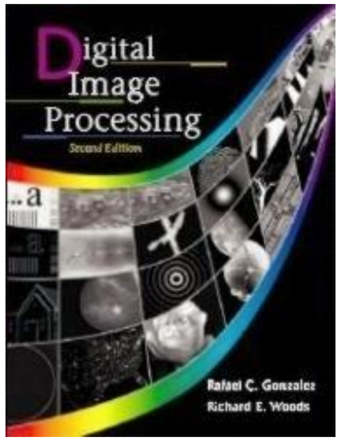


Result of filtering with a 7 * 7 median filter



Result of adaptive median filtering with i = 7

References



"Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002

–"Fundamentals of Digital Image Processing" Anil K. Jain, 1989

–Image Processing and Pattern Recognition Slides of Dr. Sanjeeb Prasad Panday

–https://www.slideshare.net/nidhalelabbadi/image-processing-zooming-and-shrinking

Thank you !!!