Digital Image Processing

Image Restoration

Contents

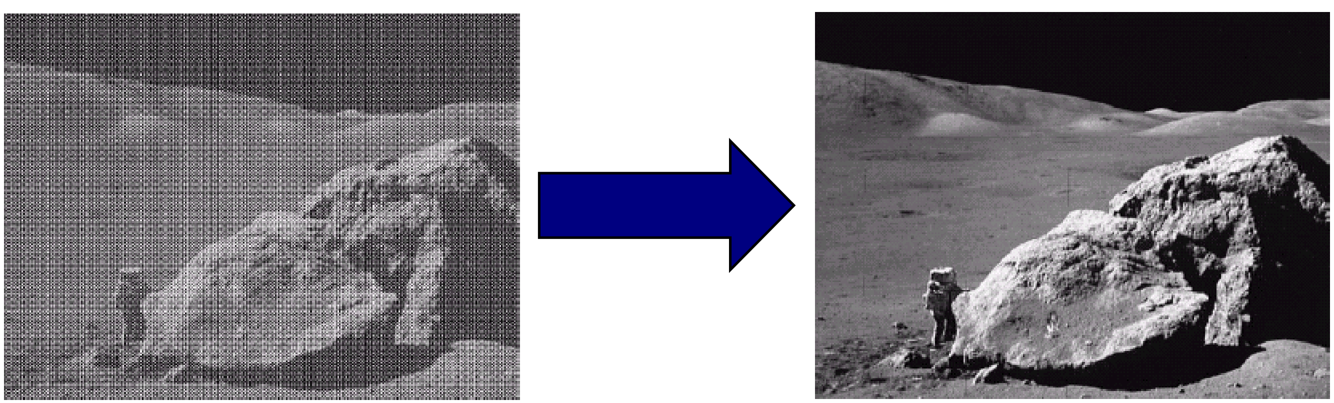
In this lecture we will look at image restoration techniques used for noise removal

– 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 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

Filtering to Remove Noise

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

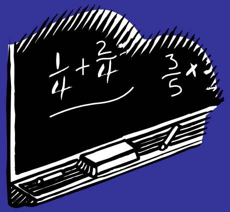
The *arithmetic mean* filter is a very simple one and is calculated as follows:

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

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

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

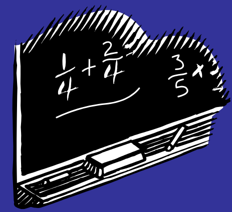
Image f (x, y)

y

Filtered Image ***x***

Image f (x, y)

y



Other Means (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

Other Means (cont…)

Contraharmonic Mean:

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

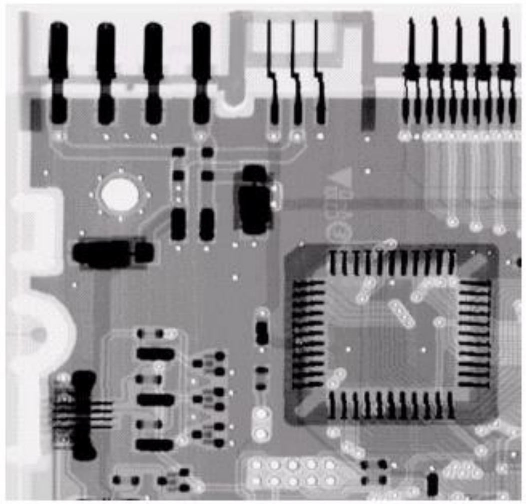
Q is the *order* of the filter and adjusting its value changes the filter’s behaviour

Positive values of Q eliminate pepper noise

Negative values of Q eliminate salt noise

Noise Removal Examples

Original Image



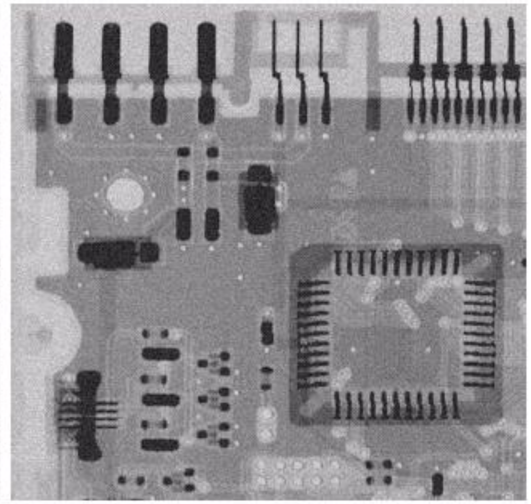
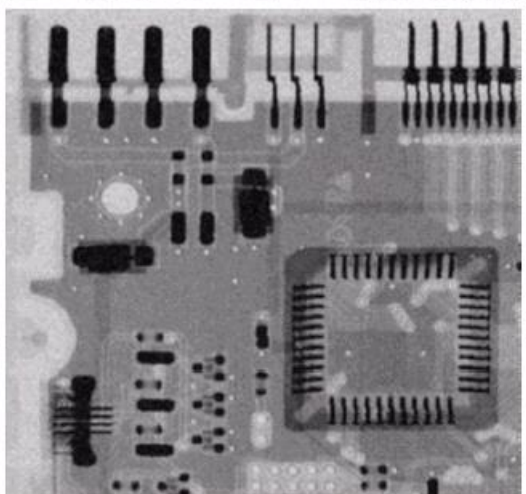
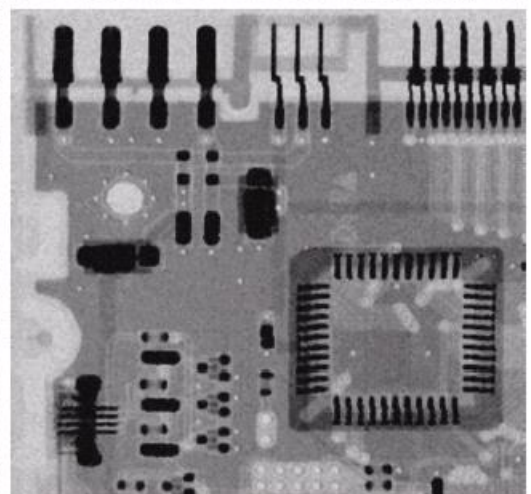


Image Corrupted By Gaussian Noise

After A 3*3 Arithmetic Mean Filter

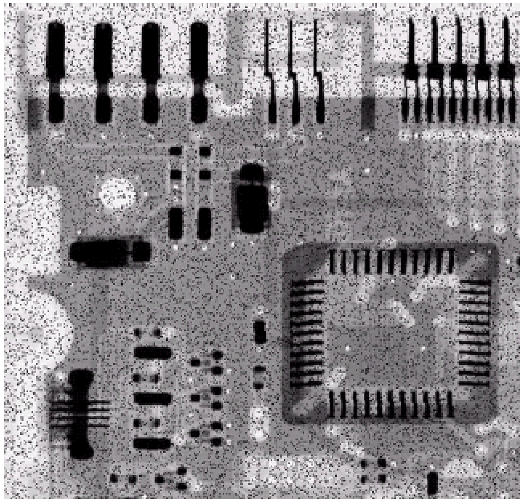




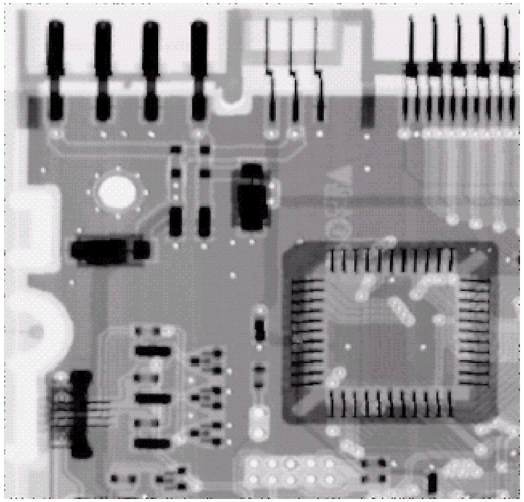
After A 3*3 Geometric Mean Filter

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



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



Noise Removal Examples (cont…)

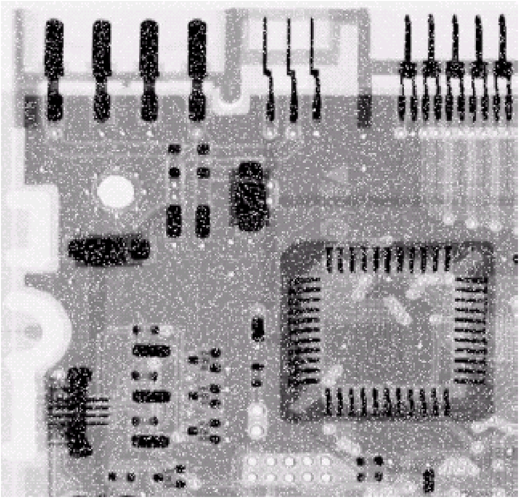
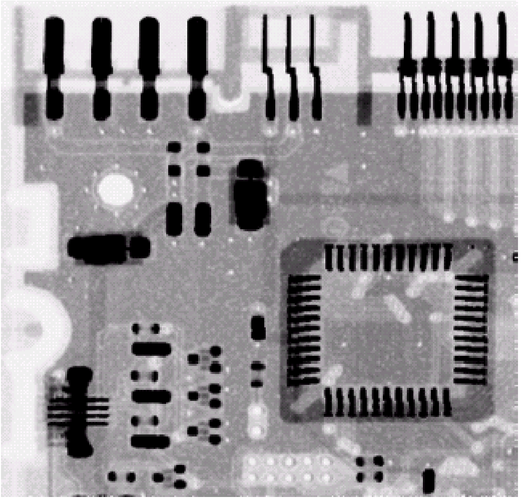


Image
Corrupted
By Salt
Noise

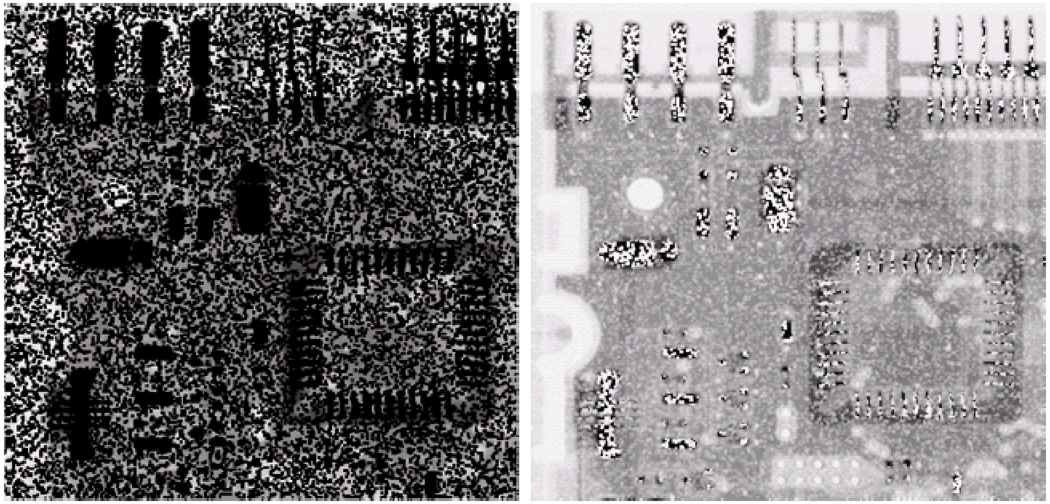


Result of
Filtering Above
With 3*3
Contraharmonic
Q=-1.5

Contraharmonic Filter: Here Be Dragons

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

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



Degradation Model

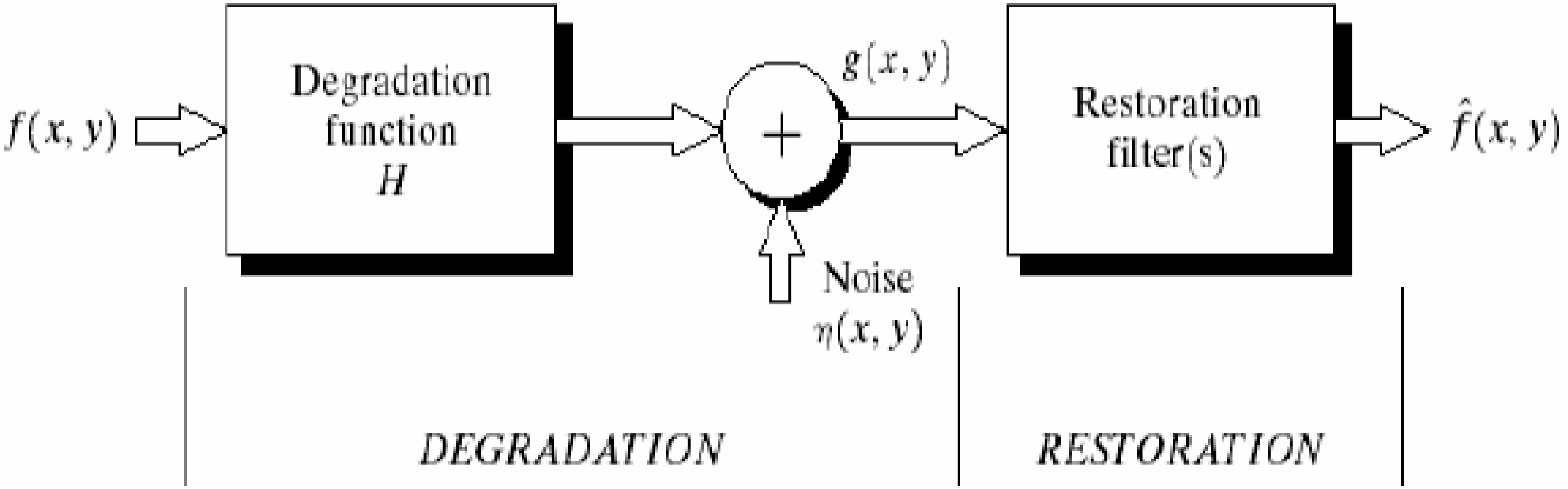
- Given $g(x,y)$, some knowledge about the degradation function *H*, and some knowledge about the additive noise term $\eta(x,y)$ -- objective : to obtain an estimate $\hat{f}(x,y)$ of the original image

Spatial domain

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

Frequency domain

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



Noise Models

- Images are often degraded by random noise
- Noise can occur during image capture, transmission or processing, and may be dependent on or independent of image content
- Noise is usually described by its probabilistic characteristics
- White noise -- constant power spectrum (its intensity does not decrease with increasing frequency)

Noise Models

- It is frequently applied as a crude approximation of image noise in most cases
- The advantage is that it simplifies the calculations
- Noises
  - Gaussian (Normal) noise
  - Rayleigh noise
  - Gamma noise
  - Exponential noise
  - Uniform noise
  - Impulse noise

Gaussian (Normal) noise

- A very good approximation of noise that occurs in many practical cases
- Probability density of the random variable is given by the Gaussian function

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

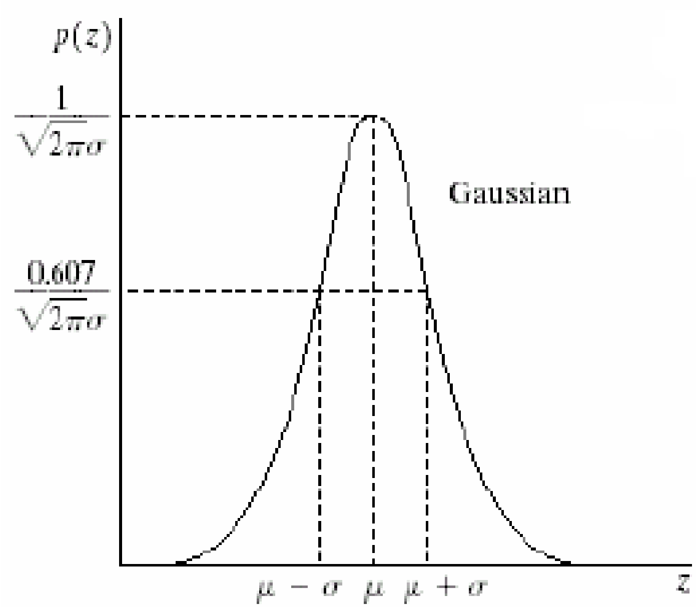
z: gray level (Gaussian random variable)

μ : the mean of avg. value of z

σ : standard deviation

70% of its values: $[(\mu-\sigma),(\mu-\sigma)]$

95% of its values: $[(\mu-2\sigma),(\mu-2\sigma)]$



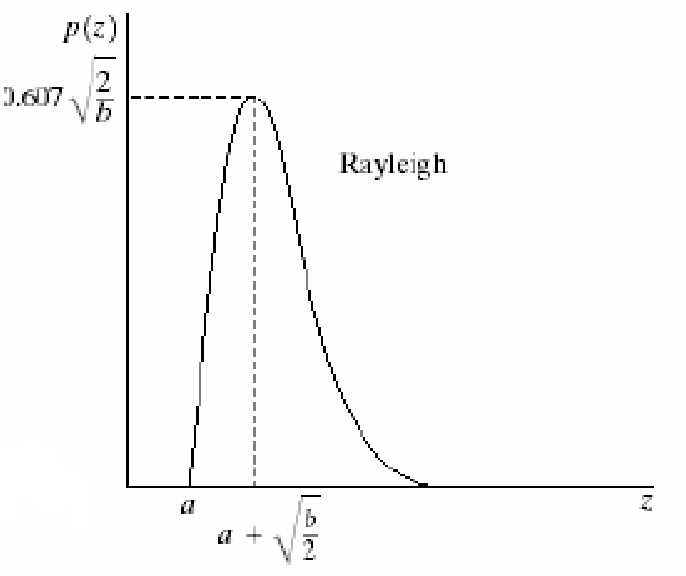
Rayleigh noise

- Note
  - The displacement from the origin (Gaussian)
  - The shape is skewed to the right

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

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

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



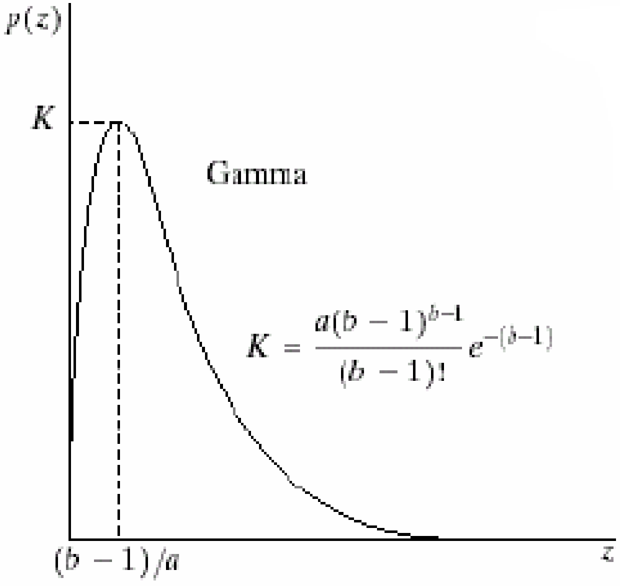
Gamma noise

- a > 0, b : positive integer
- Laser imaging

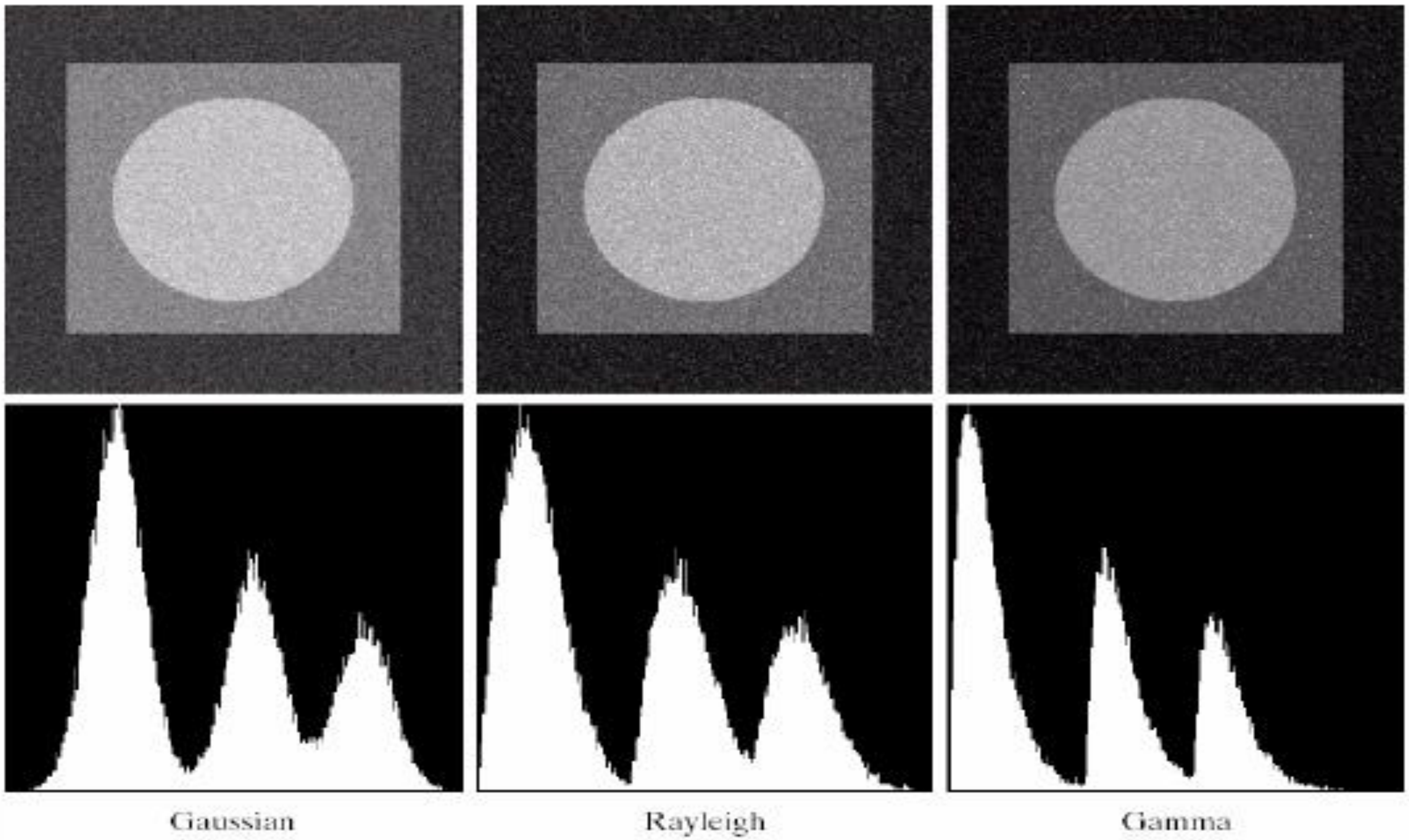
$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

$$\mu = \frac{b}{a}$$

$$\sigma^2 = \frac{b}{a^2}$$



Images & histograms resulting from adding noises



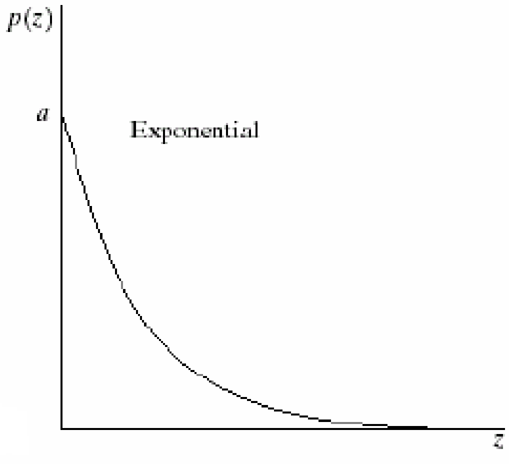
Exponential noise & Uniform noise

- Exponential noise

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

$$\mu = \frac{1}{a}$$

$$\sigma^2 = \frac{1}{a^2}$$

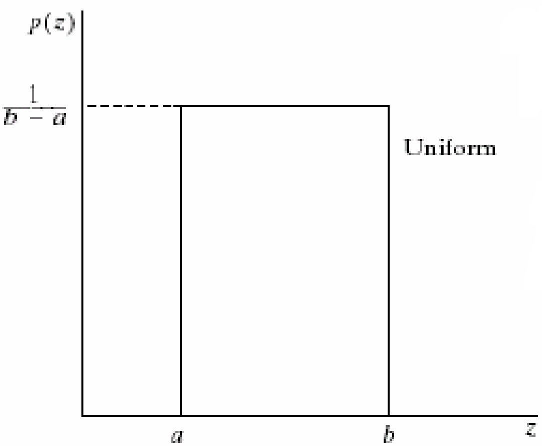


-

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

$$\mu = \frac{a+b}{2}$$

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



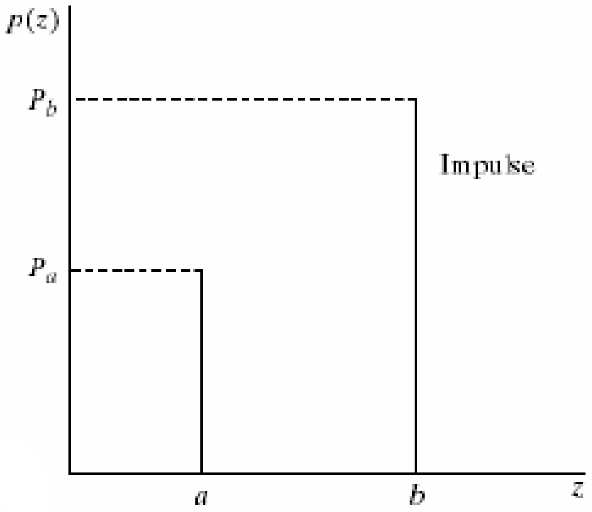
Impulse noise (Salt and Pepper Noise)

- Appearance is randomly scattered white (salt) or black (pepper) pixels over the image

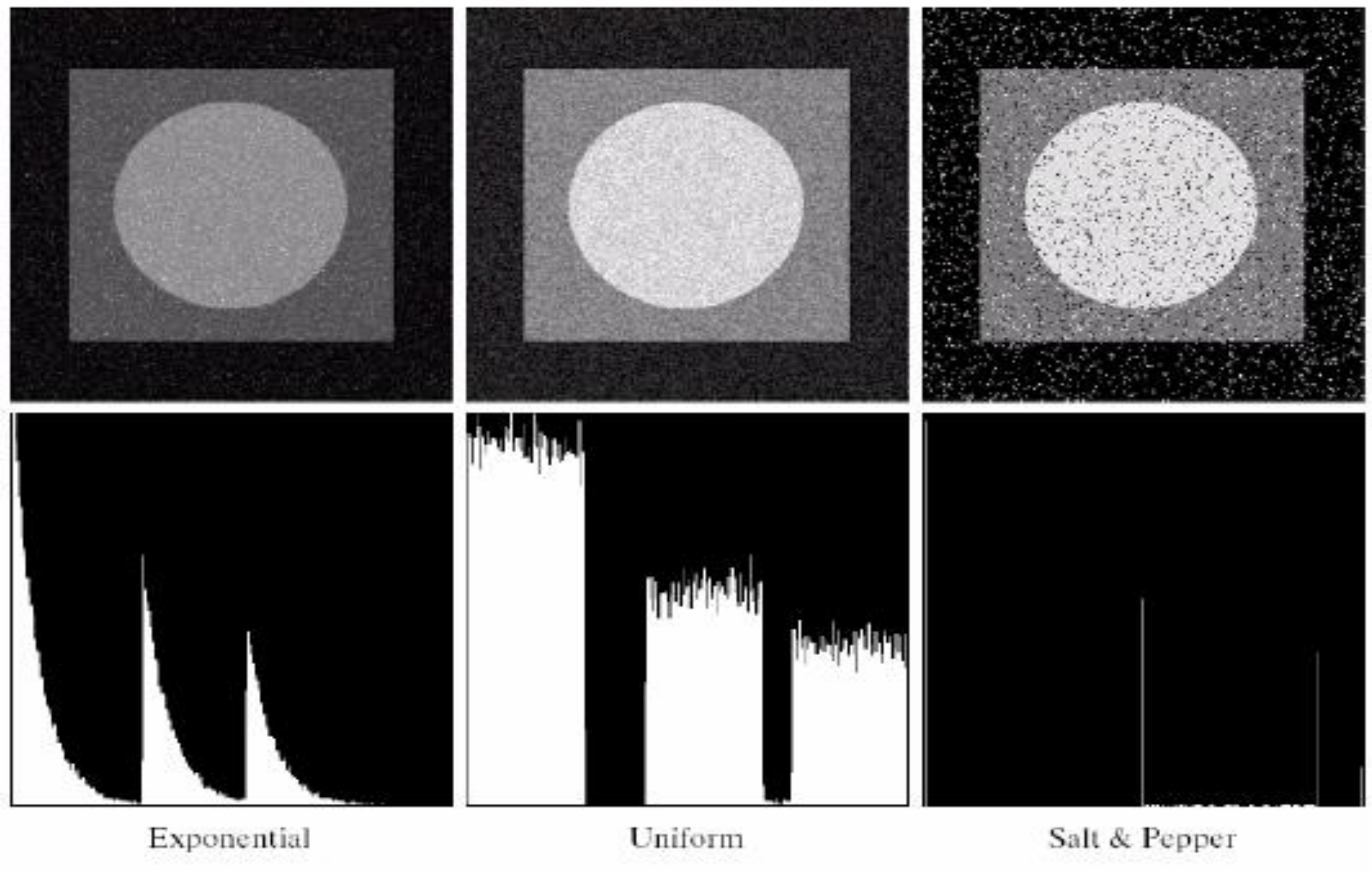
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ P_c & \text{otherwise} \end{cases}$$

Bipolar: $P_a \neq 0$ and $P_b \neq 0$

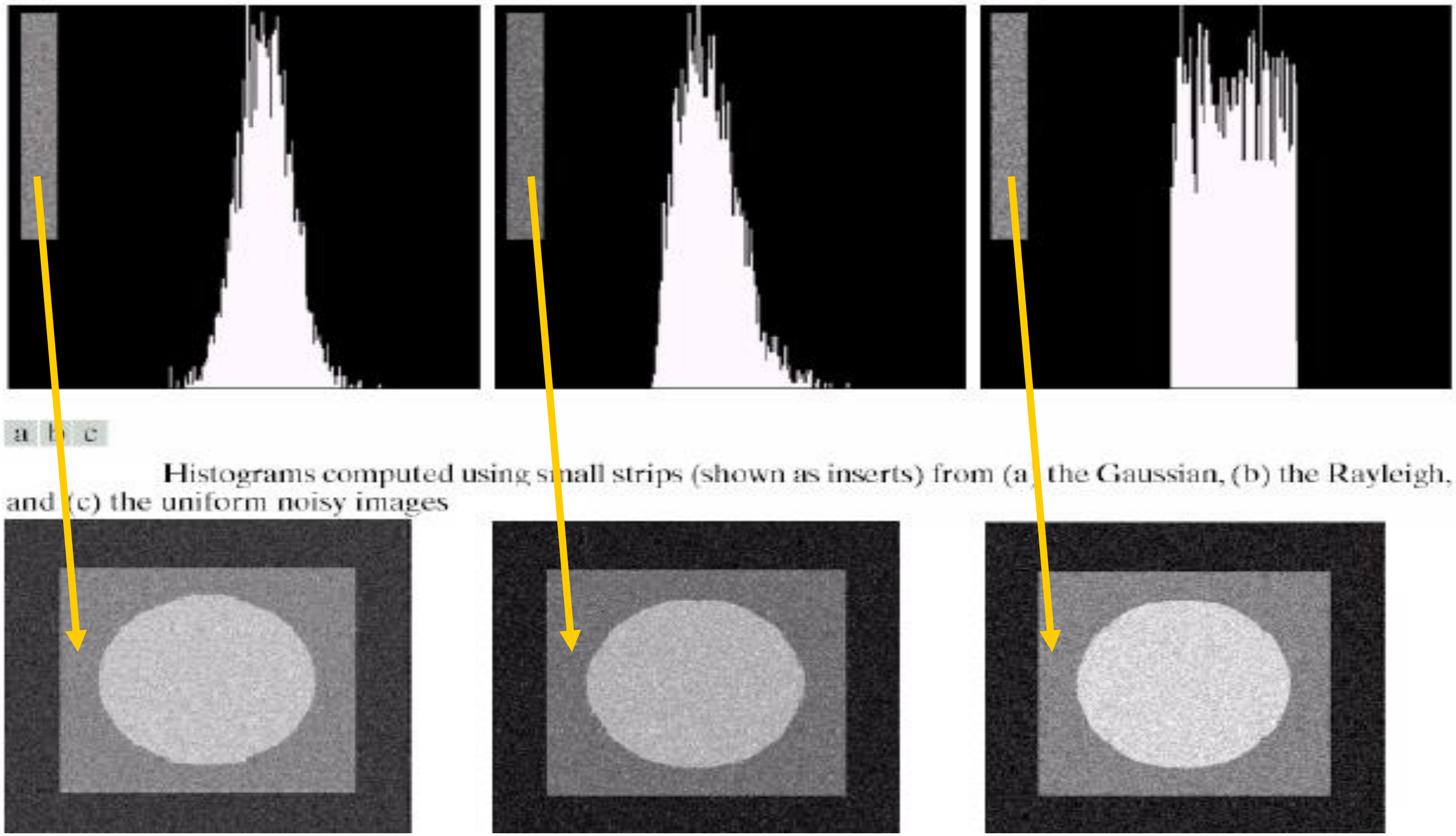
Unipolar (salt-and-pepper): $P_a = 0$ or $P_b = 0$



Images & histograms resulting from adding noises



Estimation of Noise Parameters (I)



a b c
Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images

Estimation of Noise Parameters (Ⅱ)

- The shape of the histogram identifies the closest PDF match
  - get mean & variance of the gray levels
  - use mean & variance to solve for the parameters a & b
    - Gaussian noise : mean & variance only
    - Impulse noise : the actual probability of occurrence of white & black pixels are needed

Restoration in the Presence of Noise Only – Spatial Filtering

- The only degradation present in an image is noise
  - Noise : unknown $\longrightarrow$ cannot be subtracted from image or Fourier spectrum

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

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

    - exception : periodic noise
- Spatial filtering is the method of choice in situations when only additive noise is present
  - Enhancement & restoration become almost indistinguishable disciplines in this particular case

Mean filters (I)

- Arithmetic :
  - average value of the corrupted image *g(s,t)* in the area defined by mask S of size *m* x *n*
  - The kernel contains coefficients of value *1/mn*

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

  - Smoothing local variations; noise reduction as a result of blurring

Mean filters (Ⅱ)

- Geometric :
  - Smoothing is comparable to arithmetic mean
  - Tend to lose less image detail

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

- What are the drawbacks with mean filtering?
  - A single pixel with a very unrepresentative value can significantly affect the mean value of all the pixels in its neighborhood
  - When the filter neighborhood straddles an edge, the filter will interpolate new values for pixels on the edge and so will blur that edge. This may be a problem if sharp edges are required in the output

Order-Statistics Filter (I)

- The response is based on ordering the pixels contained in the image area encompassed by the filter
- There are several variations:
  - Median filter : $\hat{f}(x, y) = \underset{(a,b)\subseteq S}{\text{median}} \{g(x-a, y-b)\}$
  - Max filter : reduce pepper noise
    $$\hat{f}(x, y) = \max_{(a,b)\subseteq S} \{g(x-a, y-b)\}$$
  - Min filter : reduce salt noise
    $$\hat{f}(x, y) = \min_{(a,b)\subseteq S} \{g(x-a, y-b)\}$$

Order-Statistics Filter (Ⅱ)

- Midpoint filter : works best for Gaussian or uniform noise
  - Order statistics + averaging

$$\hat{f}(x,y)=\frac{1}{2}\left[\max_{(a,b)\subseteq S}\{g(x-a,y-b)\}+\min_{(a,b)\subseteq S}\{g(x-a,y-b)\}\right]$$

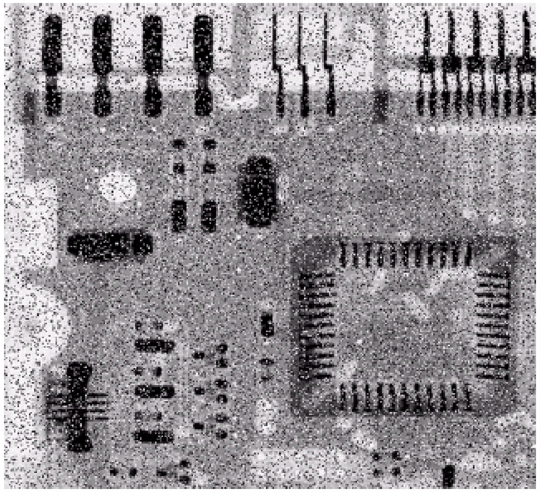
- Alpha-trimmed mean filter :
  - Delete the d/2 lowest and d/2 highest gray-level values of *g (x-a,y-b)*
  - Let $g_r(x,y)$ be the sum of the remaining pixels
  - Useful in situations involving multiple types of noise

$$\hat{f}(x,y)=\frac{g_r(x,y)}{mn-d}$$

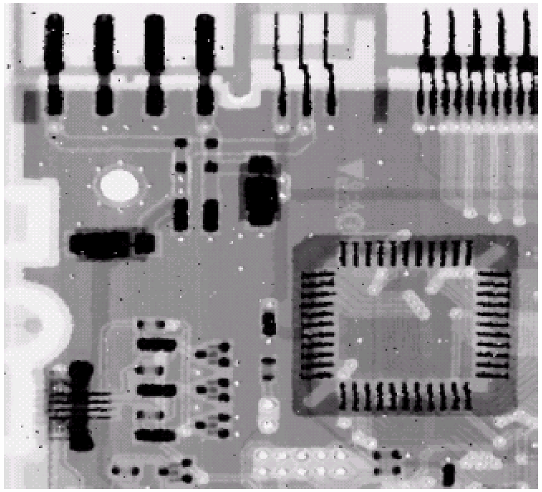
Demo1

- Repeated application of the median filter

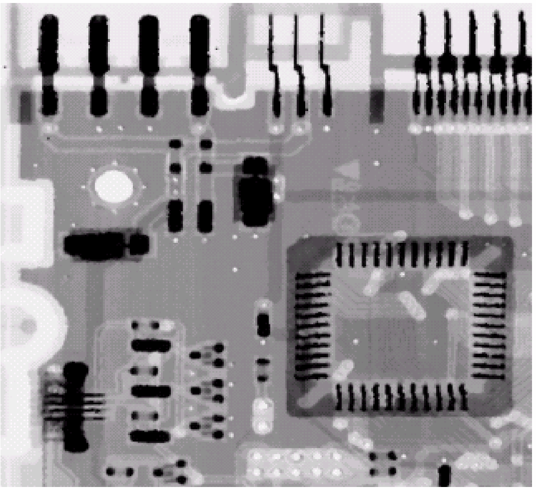
Corrupted by pepper-and-salt noise



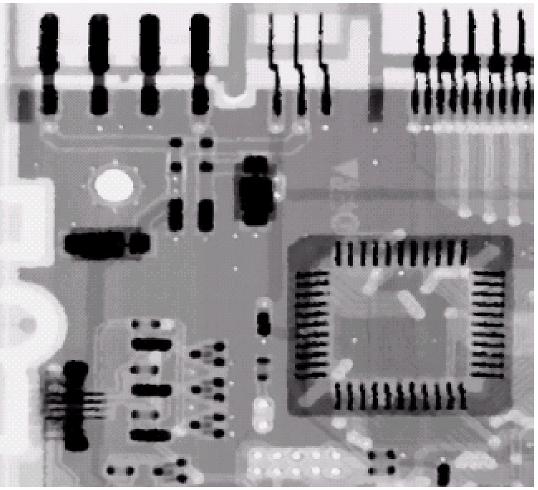
1st time



2nd time



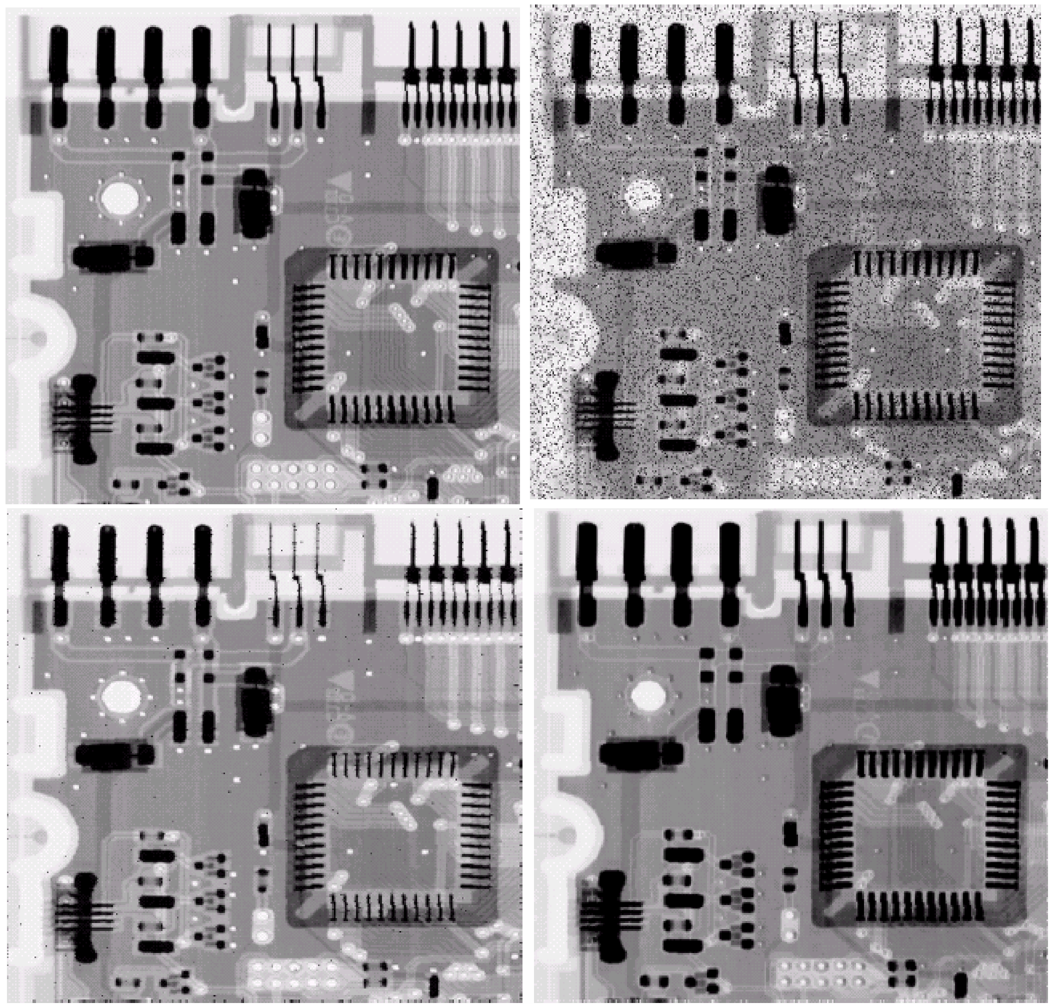
3rd time



Demo2

original

Corrupted by
pepper noise



Max filter

Min filter

Order-Statistics Filter: Drawback

- Relatively expensive and complex to compute. To find the median it is necessary to sort all the values in the neighborhood into numerical order and this is relatively slow, even with fast sorting algorithms such as *quicksort*
- Possible remedies?
  - When the neighborhood window is slid across the image, many of the pixels in the window are the same from one step to the next, and the relative ordering of these with each other will obviously not have changed

Adaptive Filtering (I)

- Changing the behavior according to the values of the grayscales under the mask

$$\underbrace{m_f}_{\text{Mean under the mask}} + \frac{\overbrace{\sigma_f^2}^{}}{\underbrace{\sigma_f^2}_{\text{Variance under the mask}} + \underbrace{\sigma_g^2}_{\text{Variance of the image}}}\left(\underbrace{g}_{\text{Current grayscale}} - m_f\right)$$

Adaptive Filtering (Ⅱ)

- If σ_f^2 is high, then the fraction is close to 1; the output is close to the original value g
  - High σ_f^2 implies significant detail, such as edges
- If the local variance is low, such as the m_f background, the fraction is close to 0; the output is close to

$$m_f + \frac{\sigma_f^2}{\sigma_f^2 + \sigma_g^2}\left(g - m_f\right)$$

Adaptive Filtering: Variation

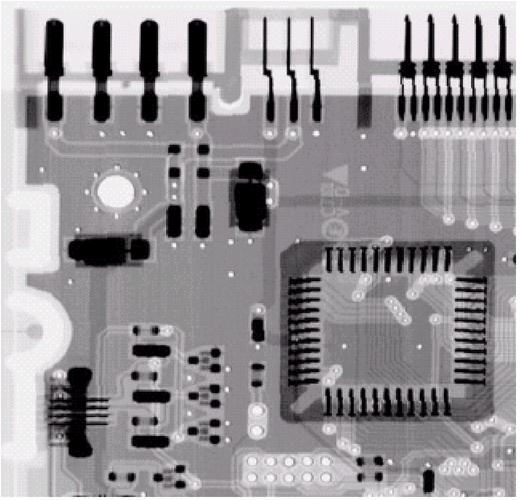
- σ_g^2 is often unknown, so is taken as the mean of all values of σ_f^2 over the entire image
- In practice, we adopt the slight variant :

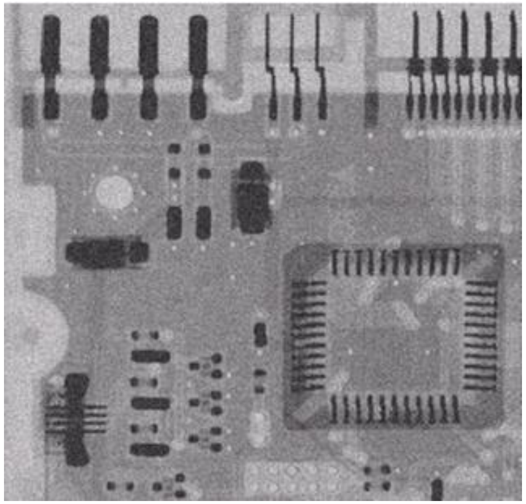
$$m_f + \frac{\max\{0, \sigma_f^2 - \sigma_g^2\}}{\max\{\sigma_f^2, \sigma_g^2\}}(g - m_f)$$

- 3 purposes :
  - Remove salt-and-pepper noise
  - Smooth other noise that are not be impulsive
  - Reduce distortion(e.g., excessive thinning/thickening of object boundaries)

Demo (7x7 mask)

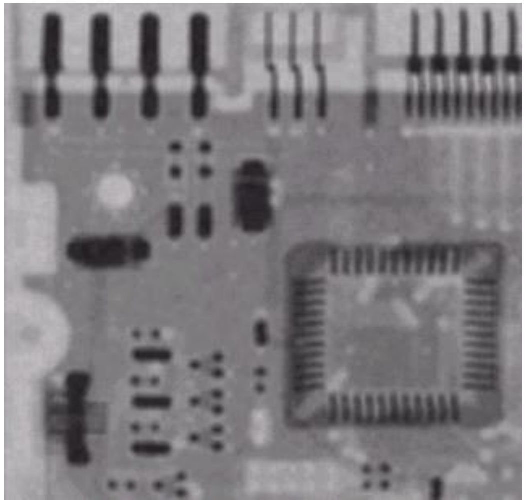
original

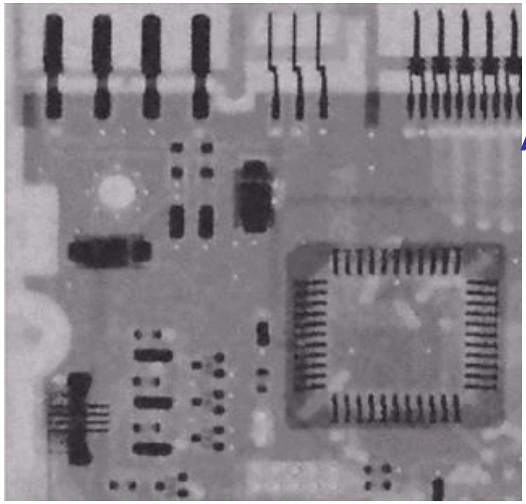




Corrupted by Gaussian noise with variance=1000

Mean filter





Adaptive filtering

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 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 $\leq S_{max}$ repeat 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

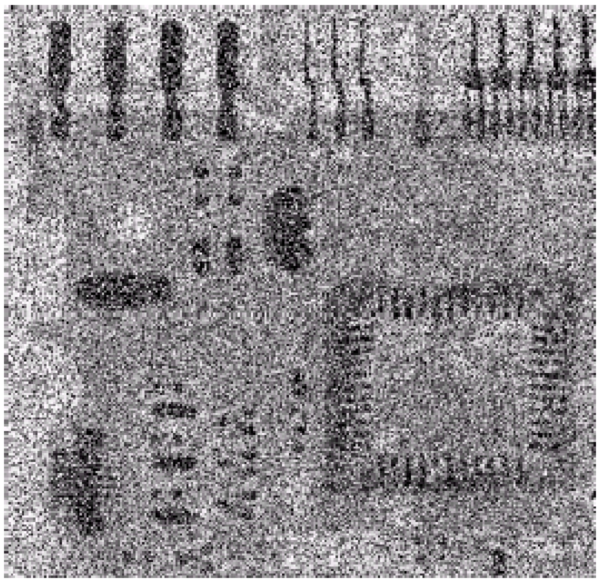
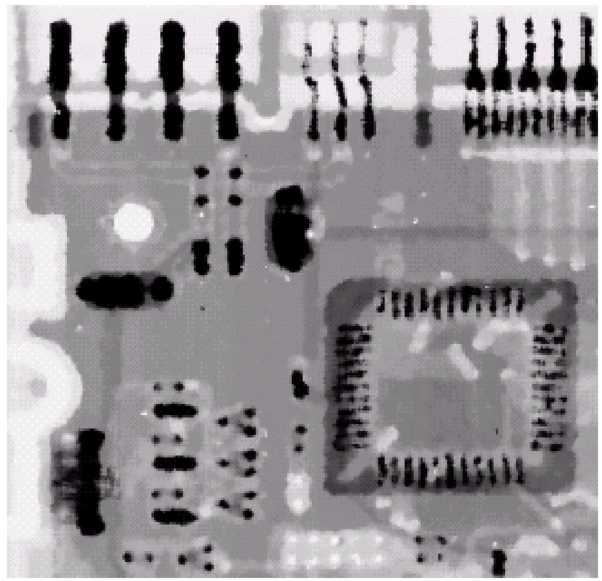
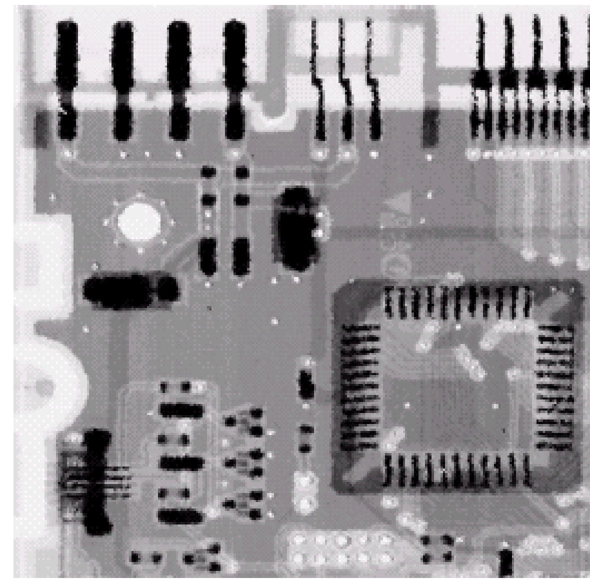


Image corrupted by salt and pepper noise with probabilities $P_a = P_b$=0.25



Result of filtering with a 7 * 7 median filter



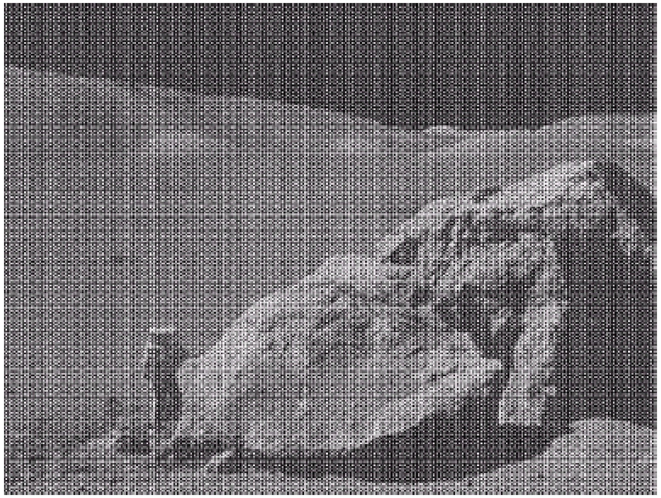
Result of adaptive median filtering with i = 7

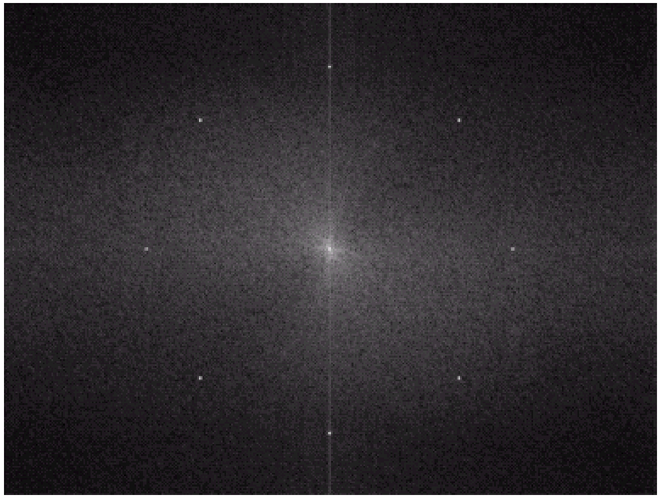
Periodic Noise

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





Band Reject Filters

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

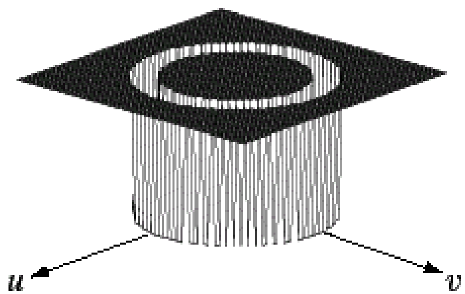
Band reject filters can be used for this purpose

An ideal band reject filter is given as follows:

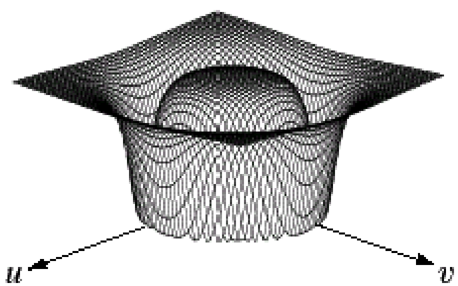
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \leq D(u,v) \leq D_0 + \frac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \frac{W}{2} \end{cases}$$

Band Reject Filters (cont…)

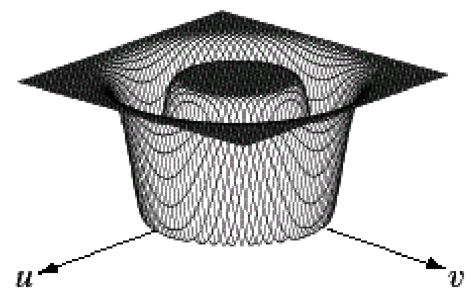
The ideal band reject filter is shown below, along with Butterworth and Gaussian versions of the filter



Ideal Band Reject Filter



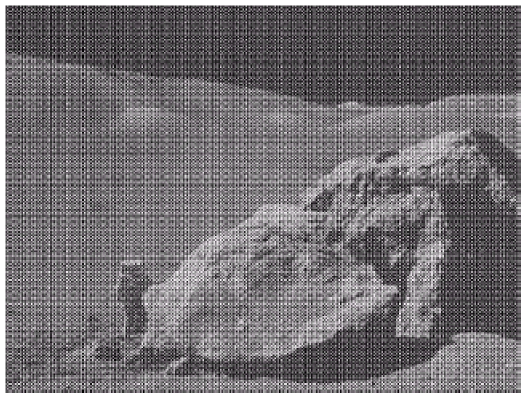
Butterworth Band Reject Filter (of order 1)



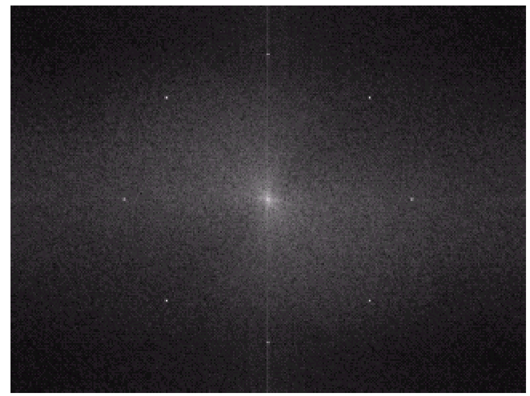
Gaussian Band Reject Filter

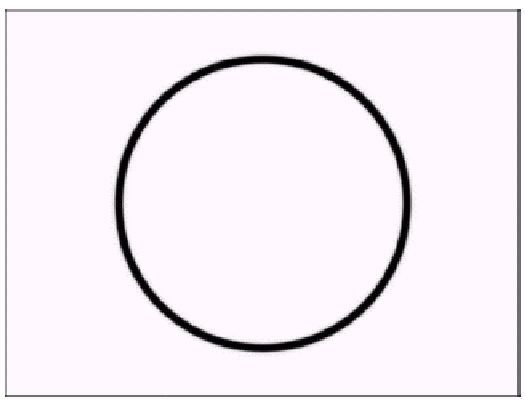
Band Reject Filter Example

Image corrupted by sinusoidal noise

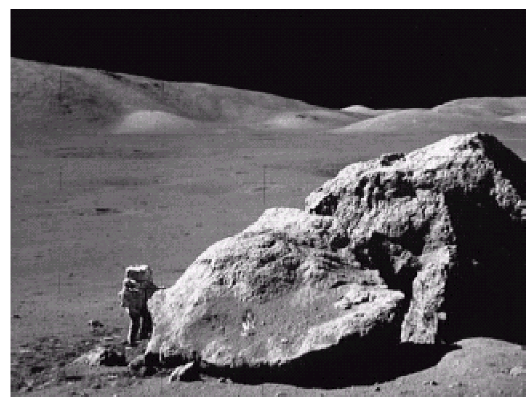


Fourier spectrum of corrupted image





Butterworth band reject filter



Filtered image

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

More into Degradation

Filter (degradation function)

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

Degraded image

Original image (what we're after)

- Knowing G & H, how to obtain F?
- Two methods:
  - Inverse filtering
  - Wiener filtering

Inverse Filtering

- The simplest approach to restoration

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

Noise – random function

- $N(u,v)$ is a random function whose Fourier transform is unknown: we cannot recover the undegraded image even if we know $H(u,v)$
- Problem: if $H(u,v)$ approaches 0, $N(u,v)/H(u,v)$ dominate the estimate $\hat{F}(u,v)$
  - – solution: limit the analysis to frequencies near the origin

- There are two similar approaches:
  - Low-pass filtering with filter L(u,v):

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

  - Thresholding (using only filter frequencies near the origin)

$$F(u,v)=\begin{cases}\dfrac{G(u,v)}{H(u,v)} & \text{if } |D(u,v)|\le d\\ G(u,v) & \text{if } |D(u,v)|> d\end{cases}$$

D(u,v) being the distance from the center

The Poor Performance of Direct Inverse Filtering

a b
c d

(a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

(b)-(d): Butterworth filter

Inverse Filtering: Weaknesses

- Inverse filtering is not robust enough
  - Doesn't explicitly handle the noise
- It is easily corrupted by the random noise

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

$$=> F(u,v) = \frac{G(u,v) - N(u,v)}{H(u,v)}$$

- The noise can completely dominate the output

Wiener Filtering

- What measure can we use to say whether our restoration has done a good job?
- Given the original image ***f*** and the restored version ***r***, we would like ***r*** to be as close to ***f*** as possible
- One possible measure is the sum-squared-differences

$$\sum \left(f_{i,j} - r_{i,j} \right)^2$$

- Wiener filtering: minimum mean square error:

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

Specified constant

Comparison of Inverse & Wiener Filtering

a b c

Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering. (b) Radially limited inverse filter result. (c) Wiener filter result.

- Column 1:
  Blurred image with additive Gaussian noise of variances 650, 65 and 0.0065
- Column 2:
  Inverse filtering
- Column 3:
  Wiener filtering