IMAGE RESTORATION

Aziz M. Qaroush

Birzeit University

First Semester 2015/2016

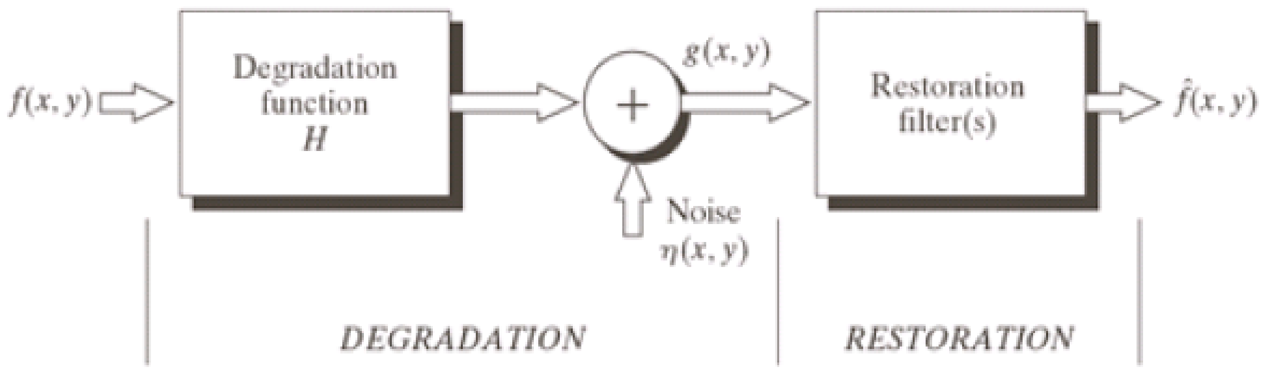
Outline

- Introduction
- Degradation/Restoration Model
- Noise Models
- Restoration in the Presence of Noise only – Spatial Filtering
  - Mean Filters
  - Order-statistic Filters
  - Adaptive Filters
- Periodic Noise Reduction Using Frequency Filtering
- Estimation of Degradation Function
- Inverse Filtering
- Minimum Mean Error Filtering

Introduction

- The principle goal of image restoration is similar to that of image enhancement in obtaining an improved image according to some criterion
- *Image enhancement is subjective* in the sense that the used techniques are heuristic and applied to take advantage of the psychophysical aspects of the human visual system
- *Image restoration is objective* in the sense that a the degradation introduced into the image is known based a priori knowledge of the degradation phenomena
- Restoration can be performed in spatial or frequency domains. Spatial treatment is applicable only when the degradation is additive noise

A Degradation /Restoration Model



- The primary objective of restoration is to obtain an estimate of the original image based on the knowledge of the degradation and noise functions
- The degraded image g(x,y) can be modeled as

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

$$Or$$

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

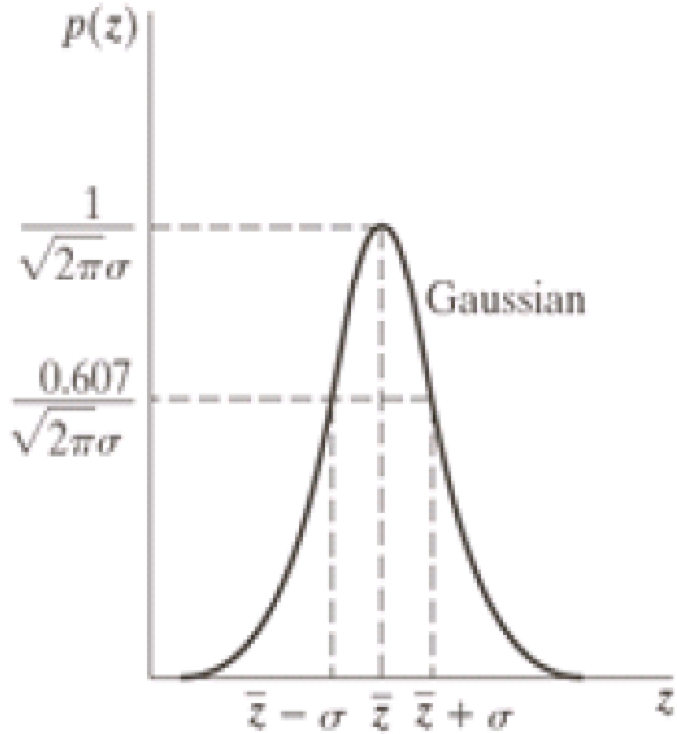
Noise Models

- Noise is introduced into a digital image during image acquisition and/or transmission
- Noise introduced during acquisition is dependant on the quality of sensing elements and the environmental conditions (Light levels and temperature)
- Image corruption during transmission is primarily due to channel interference (lightning and atmospheric disturbances)
- In our discussion, we assume that the noise introduced into the image is a random variable that is independent (uncorrelated) of the spatial coordinates and pixel values

Noise Models

- **Gaussian Noise**

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-z_0)^2/2\sigma^2}$$



z_0 is the mean value

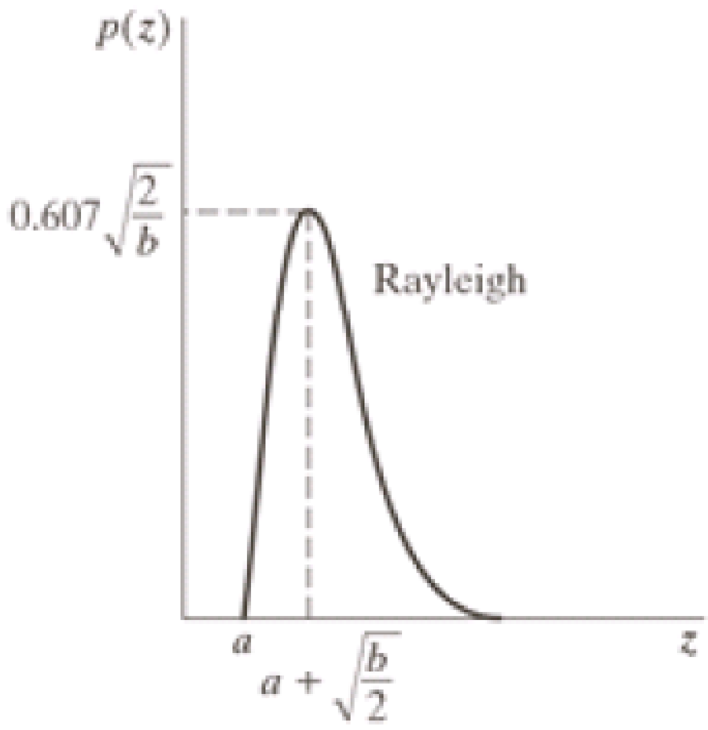
σ^2 is the variance

- It is useful in characterizing noise of electronic circuits and sensors

Noise Models

- **Rayleigh Noise**

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



- The mean is

$$z_0 = a + \sqrt{\pi b / 4}$$

- The variance

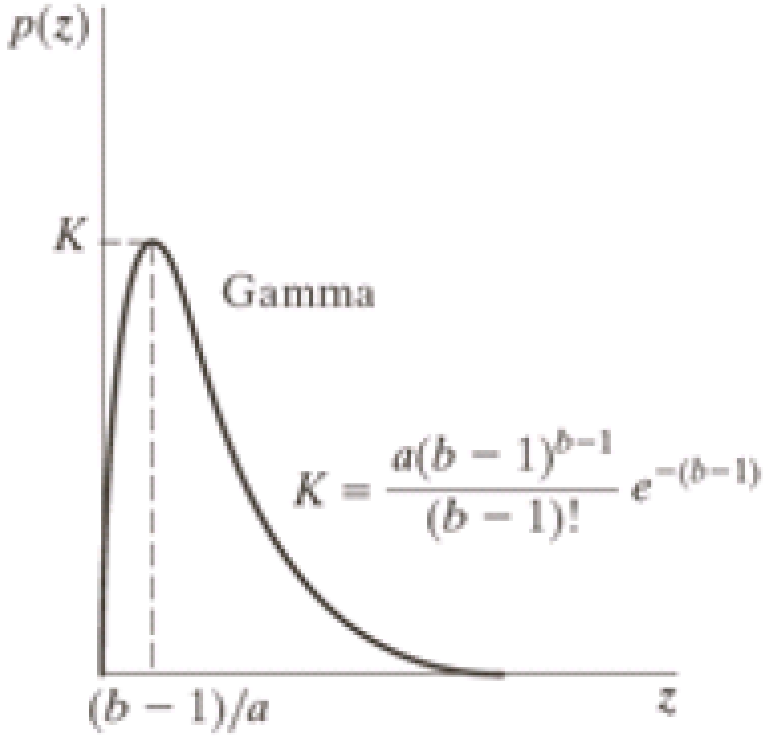
$$\sigma^2 = \frac{b(4-\pi)}{4}$$

- It is useful in characterizing noise in range imaging
- It is useful in approximating skewed histograms

Noise Models

- **Erlang (Gamma) Noise**

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



- The mean is

$$z_0 = \frac{b}{a}$$

- The variance

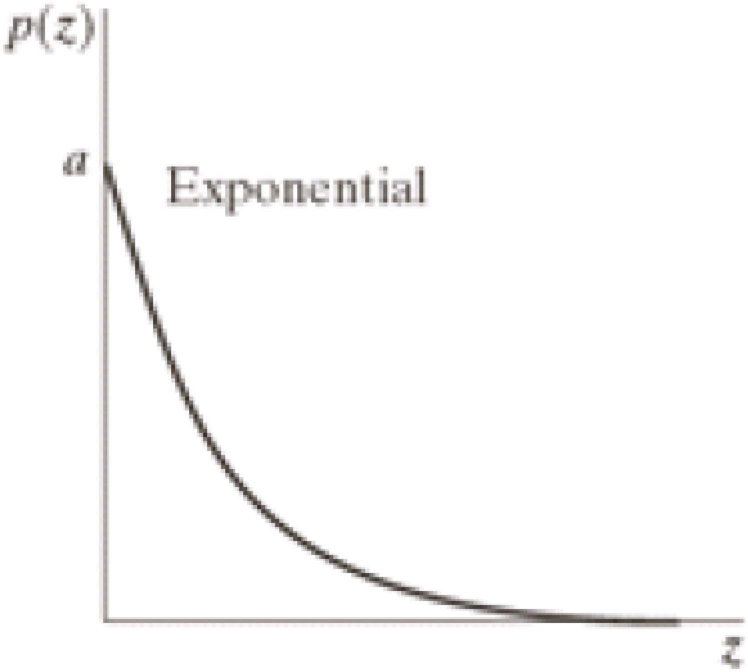
$$\sigma^2 = \frac{b}{a^2}$$

- a>0 and b is a positive integer
- It is useful in characterizing noise in laser imaging

Noise Models

- **Exponential Noise**

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



- The mean is

$$z_0 = \frac{1}{a}$$

- The variance

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

- It is useful in characterizing noise in laser imaging

Noise Models

- **Uniform Noise**

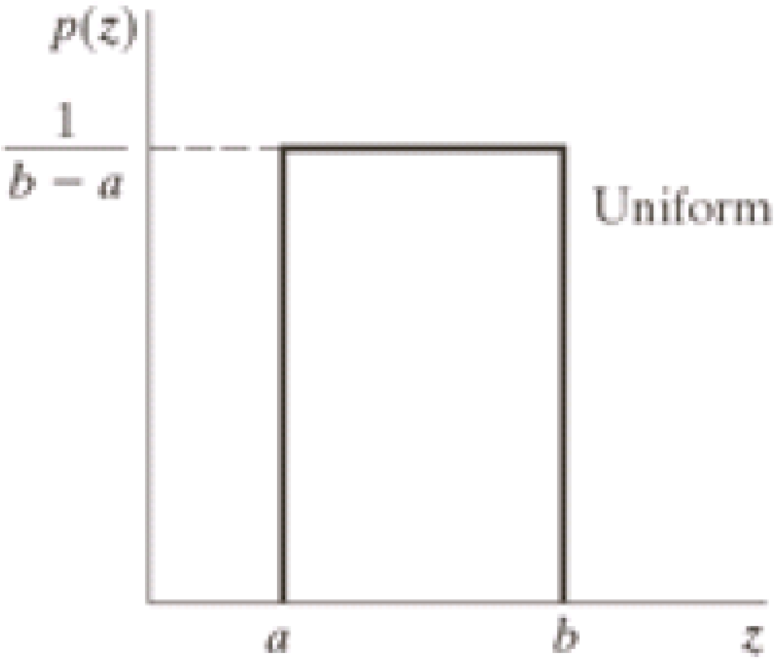
$$p(z) = \begin{cases} \dfrac{1}{b-1}, & a \le z \le b \\ 0, & otherwise \end{cases}$$

- The mean is

$$z_0 = \frac{b-a}{2}$$

- The variance

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



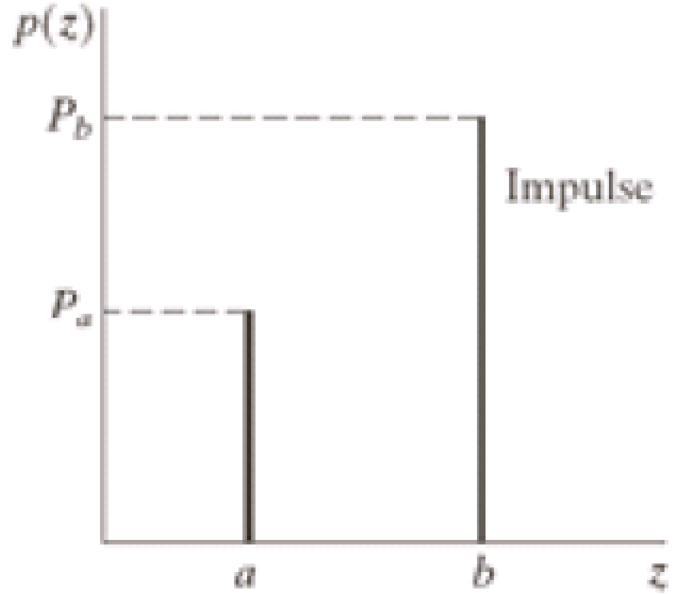
- It is the least descriptive of all types

Noise Models

- **Impulse (salt-and-pepper) Noise**

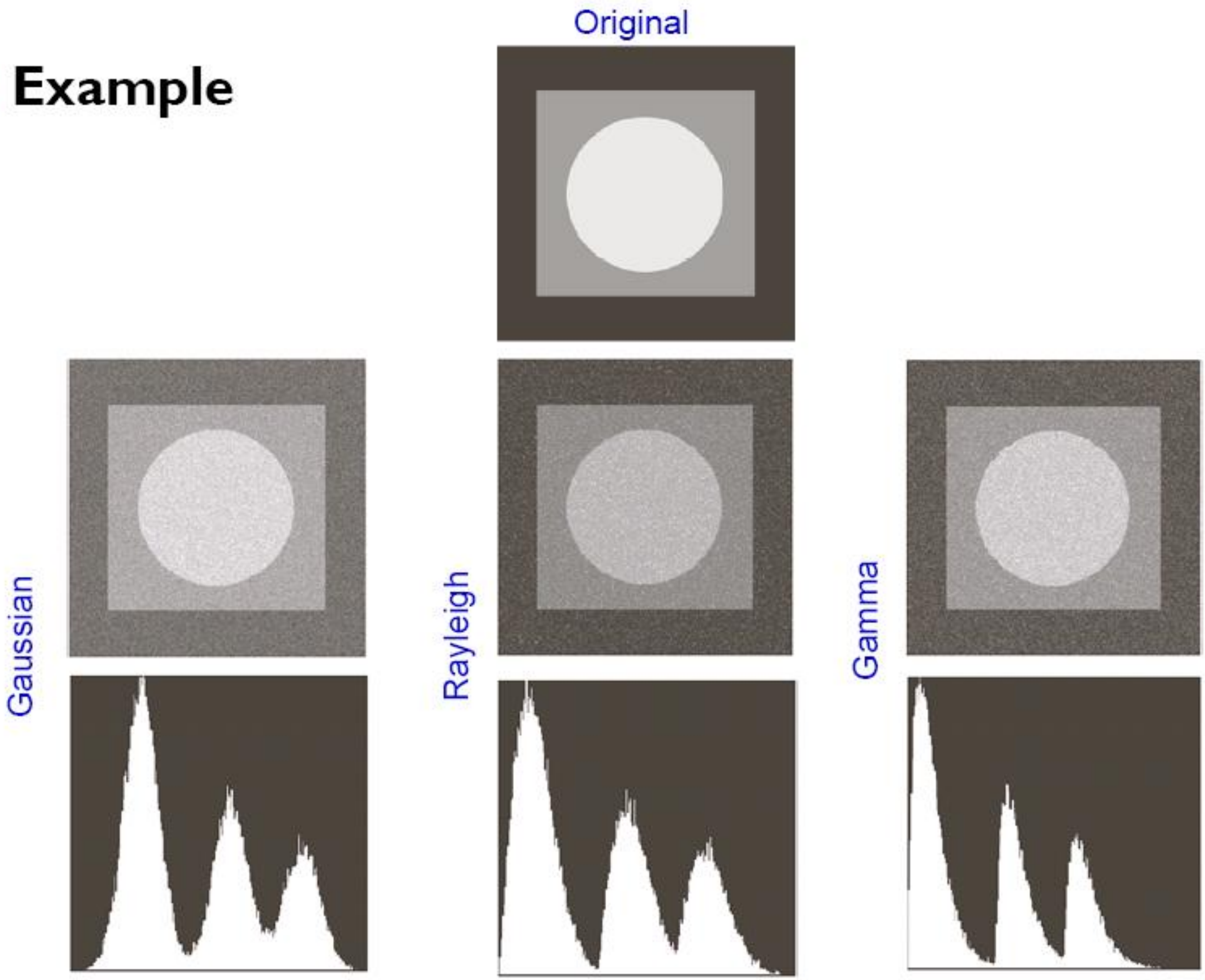
$$p(z) = \begin{cases} P_a, & z = a \\ P_b, & z = b \\ 0, & otherwise \end{cases}$$

- Usually represented as black and white dots in the image
- It appears in in situations with quick transitions such as faulty switching



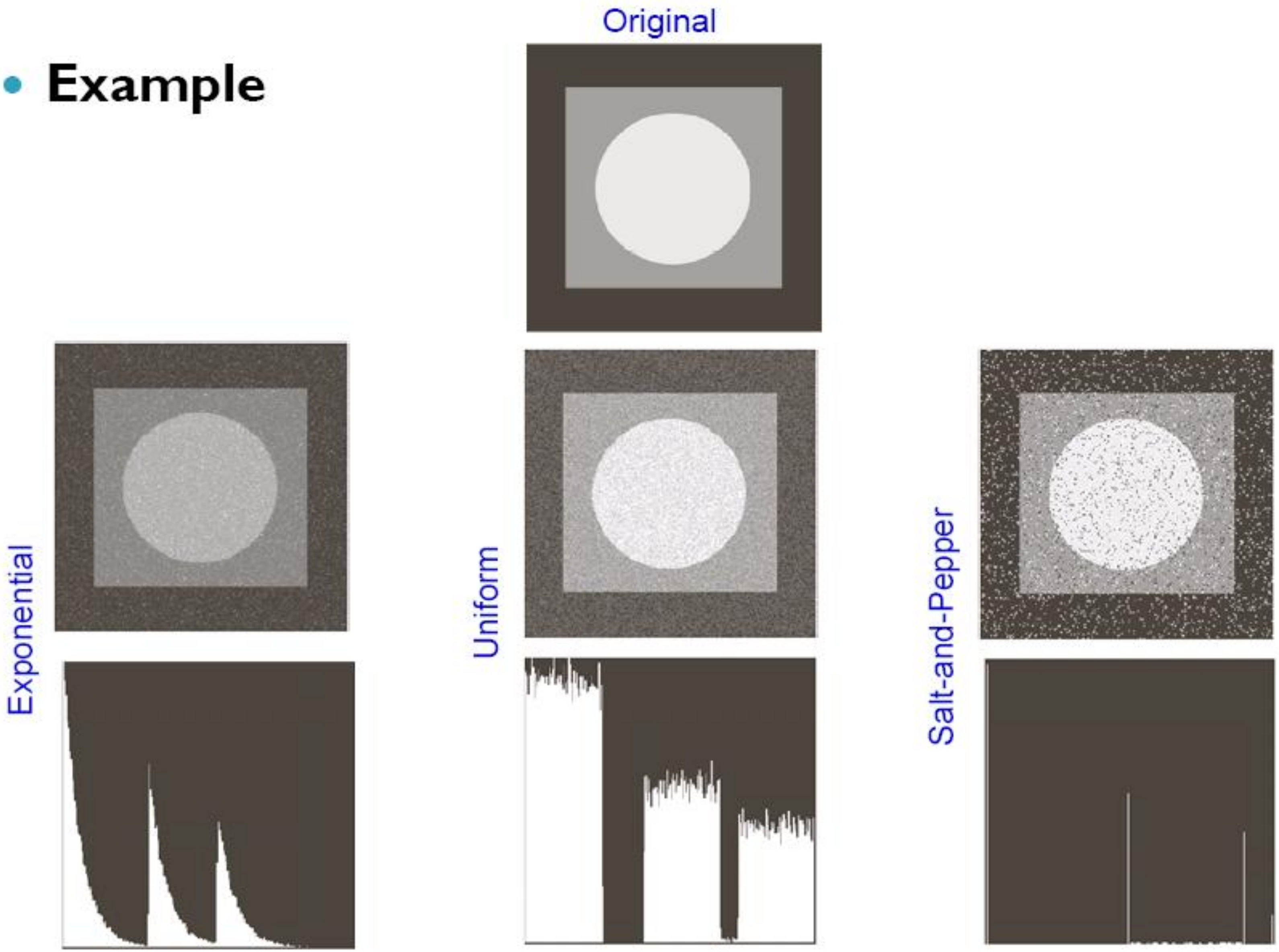
Noise Models

- **Example**



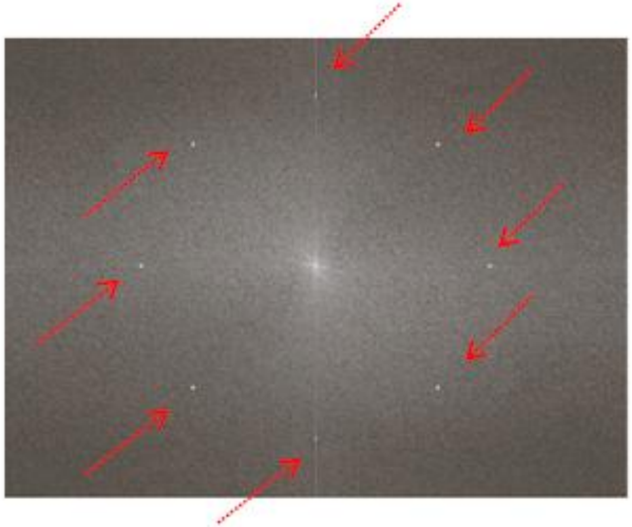
Noise Models

- **Example**



Noise Models

- **Periodic Noise**
  - Periodic noise is usually added to the image from electrical and electromechanical interference during image acquisition
  - It is the only type of spatially dependent noise that we will discuss
  - Periodic noise can be greatly reduced using frequency domain filtering



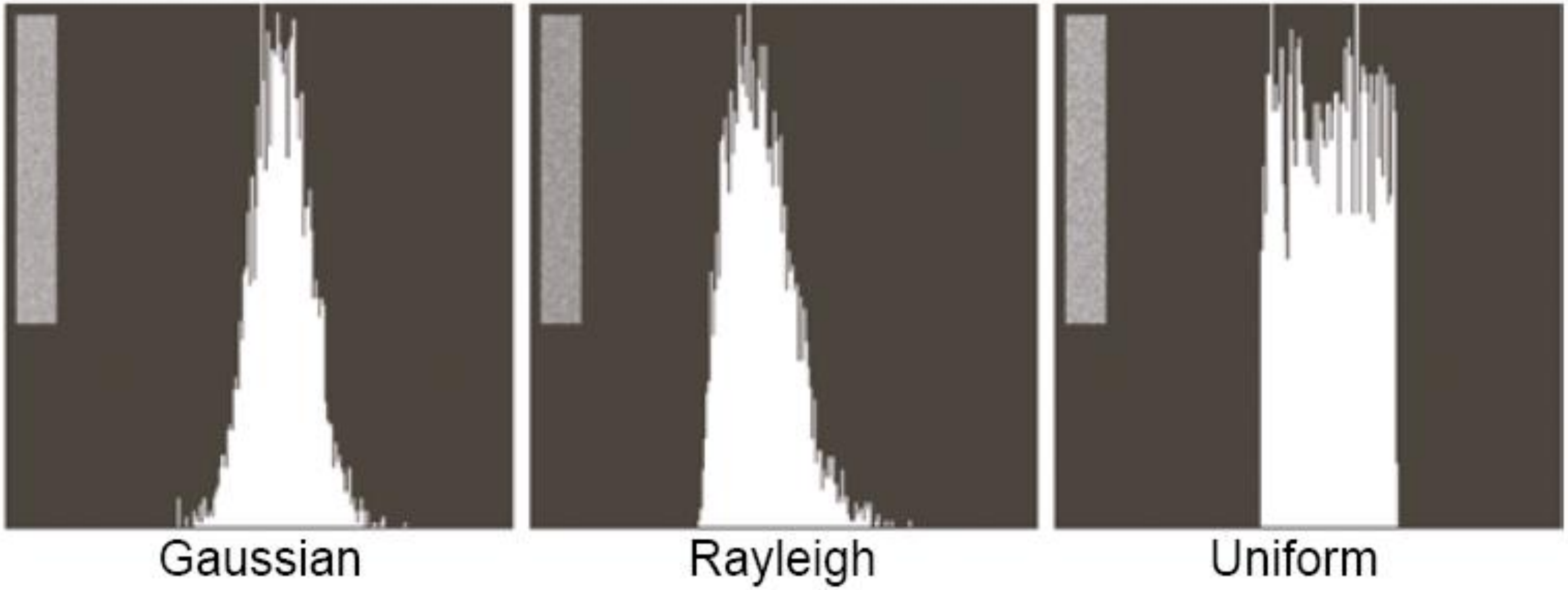
Noise Models

- **Estimating Noise Parameters**
  - Periodic noise parameters can be estimated by
    - Inspection of the Fourier transform of the image since periodic noise tend to produce spikes that can be easily detected visually
    - Infer periodicity of noise components from the image (difficult)
  - The parameters of noise PDFs may be
    - Known partially from sensor specifications
    - Estimated from acquired images by examining the histograms of small patches of reasonably constant background intensity.
    - The approximate shape of the histogram determines the type of noise and we can compute the mean and variance by

$$\bar{z} = \sum_{i=0}^{L-1} zi \times p_s(z_i)$$

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

- **Estimating Noise Parameters – Example**



- Histograms of reasonably constant intensity region extracted from images corrupted by Gaussian, exponential, and uniform

Restoration in the Presence of Noise Only

- If noise is the only degradation present, the degradation model becomes

$$g(x,y) = f(x,y) + \eta(x,y)$$
$$Or$$
$$G(u,v) = F(u,v) + N(u,v)$$

- Restoration can be simply done by subtraction if the noise values are known!
- Alternatively, we use spatial filtering when only additive noise is present
- In what follows, we explore new types of spatial filters. Filtering using the new filters is done in the same way discussed in Chapter 3

Restoration in the Presence of Noise Only

- **Mean Filters**
  - For a rectangular subimage window S_{xy} of size mxn that is centered at pixel (x,y), we define the following mean filters
    - **Arithmetic Mean Filter**

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

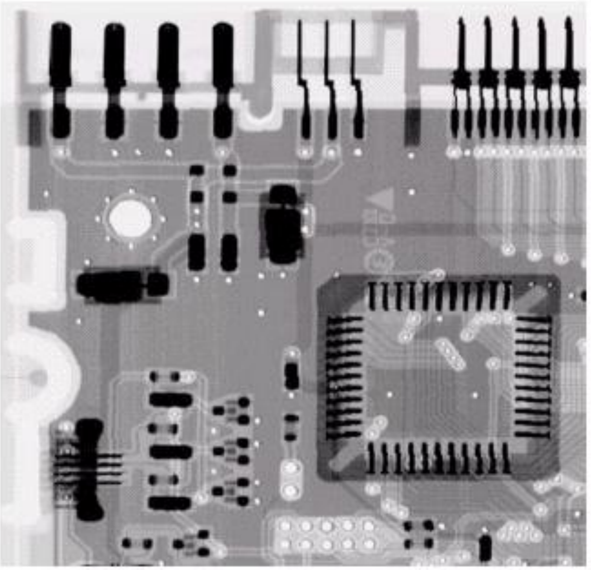
      - It's capable of reducing noise levels, but it smoothes local variations in an image
    - **Geometric Mean Filter**

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

      - Achieves similar smoothing results as the arithmetic mean filter, but tends to lose less details in the image

Restoration in the Presence of Noise Only

Original image



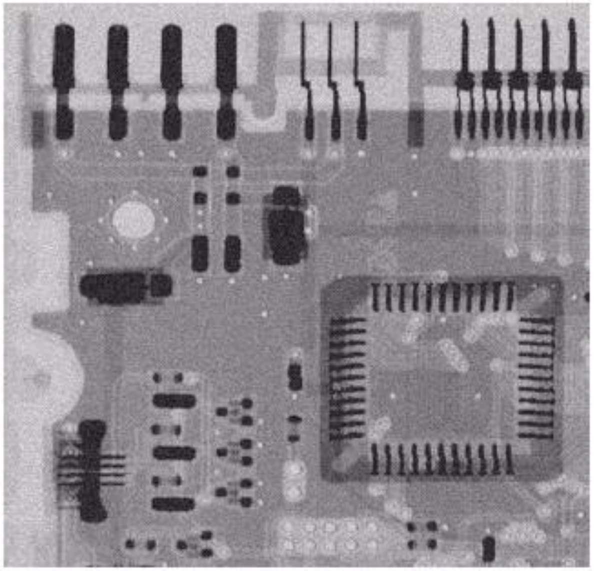
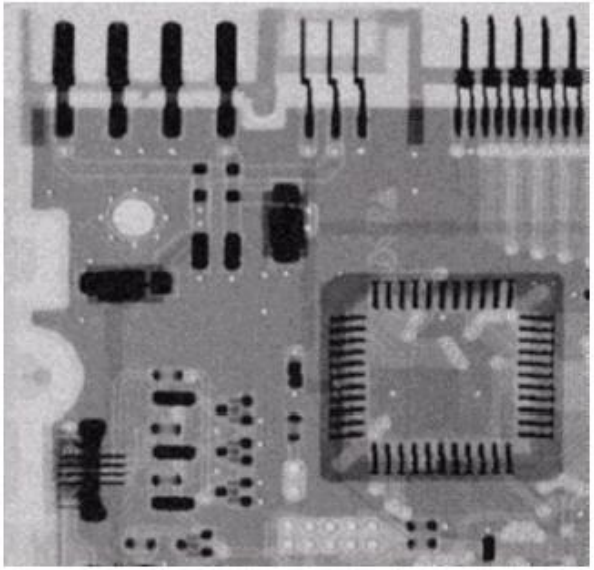


Image corrupted by AWGN

Image obtained using a 3x3 arithmetic mean filter



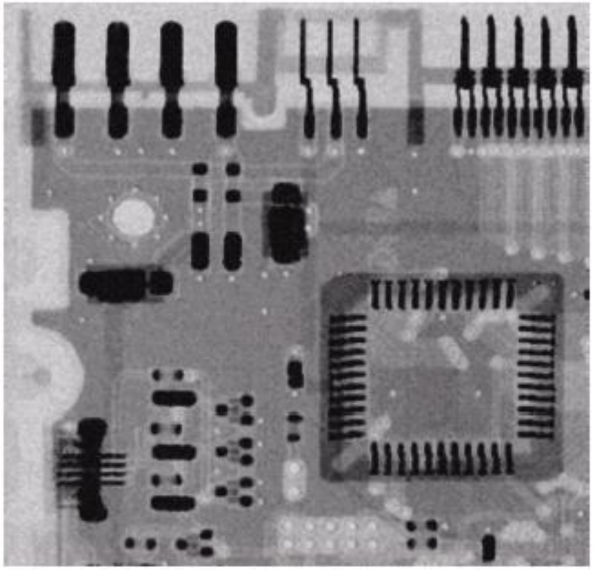


Image obtained using a 3x3 geometric mean filter

Restoration in the Presence of Noise Only

- **Mean Filters**
  - **Harmonic Mean Filter**

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

    - The harmonic mean filter works well for salt noise, but fails for pepper noise. It does well also for other types of noise
  - **Contraharmonic Mean Filter**

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

    - Q is the order of the filter
    - Well suited reducing or virtually eliminating salt-and-pepper noise. Positive Q eliminates pepper noise while positive Q eliminates salt noise
    - It is important to select the proper sign of the filter

Restoration in the Presence of Noise Only

Image corrupted by pepper noise with prob. = 0.1

Image corrupted by salt noise with prob. = 0.1

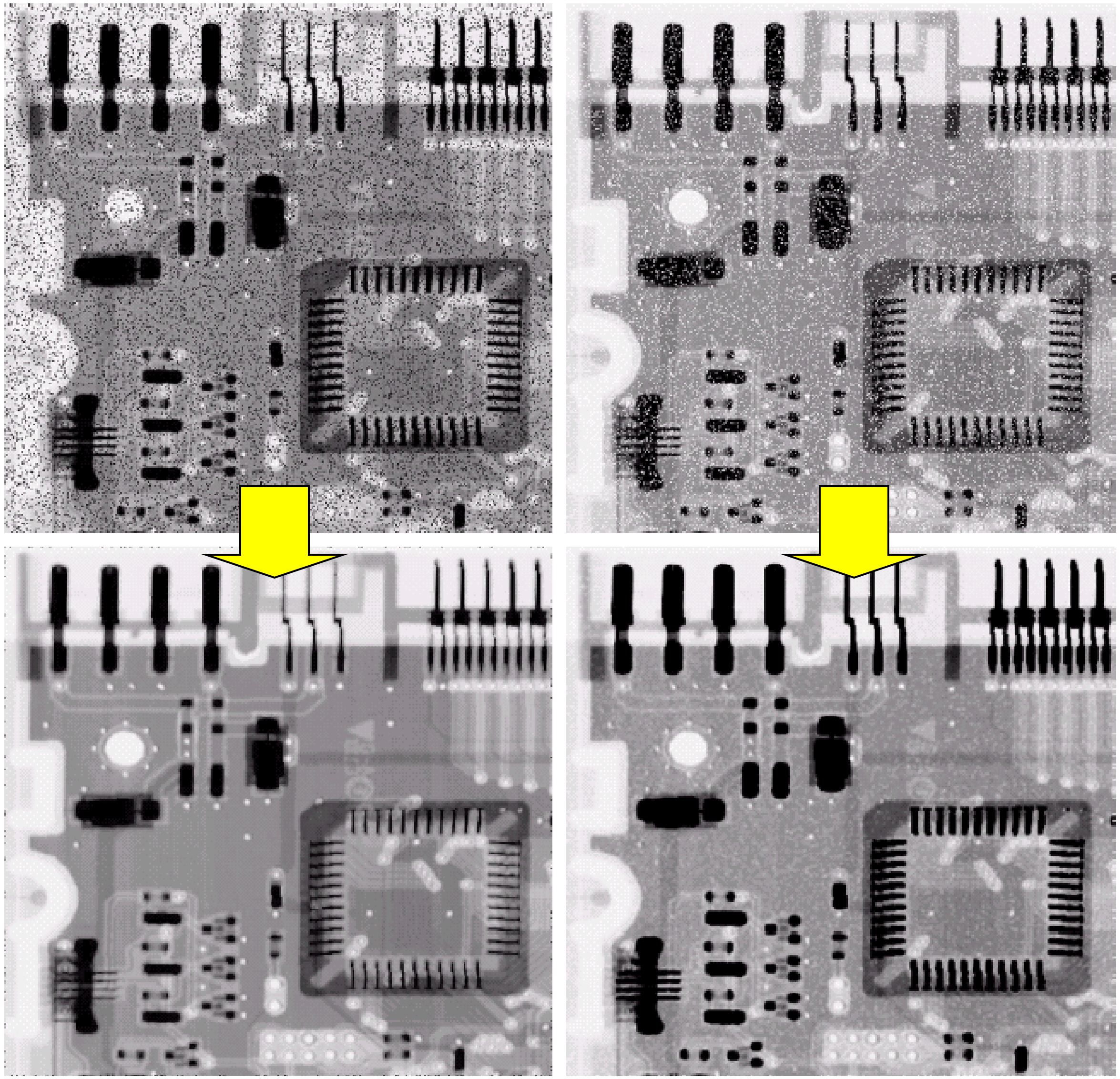


Image obtained using a 3x3 contra-harmonic mean filter With $Q = 1.5$

Image obtained using a 3x3 contra-harmonic mean filter With $Q=-1.5$

Restoration in the Presence of Noise Only

Image corrupted by pepper noise with prob. = 0.1

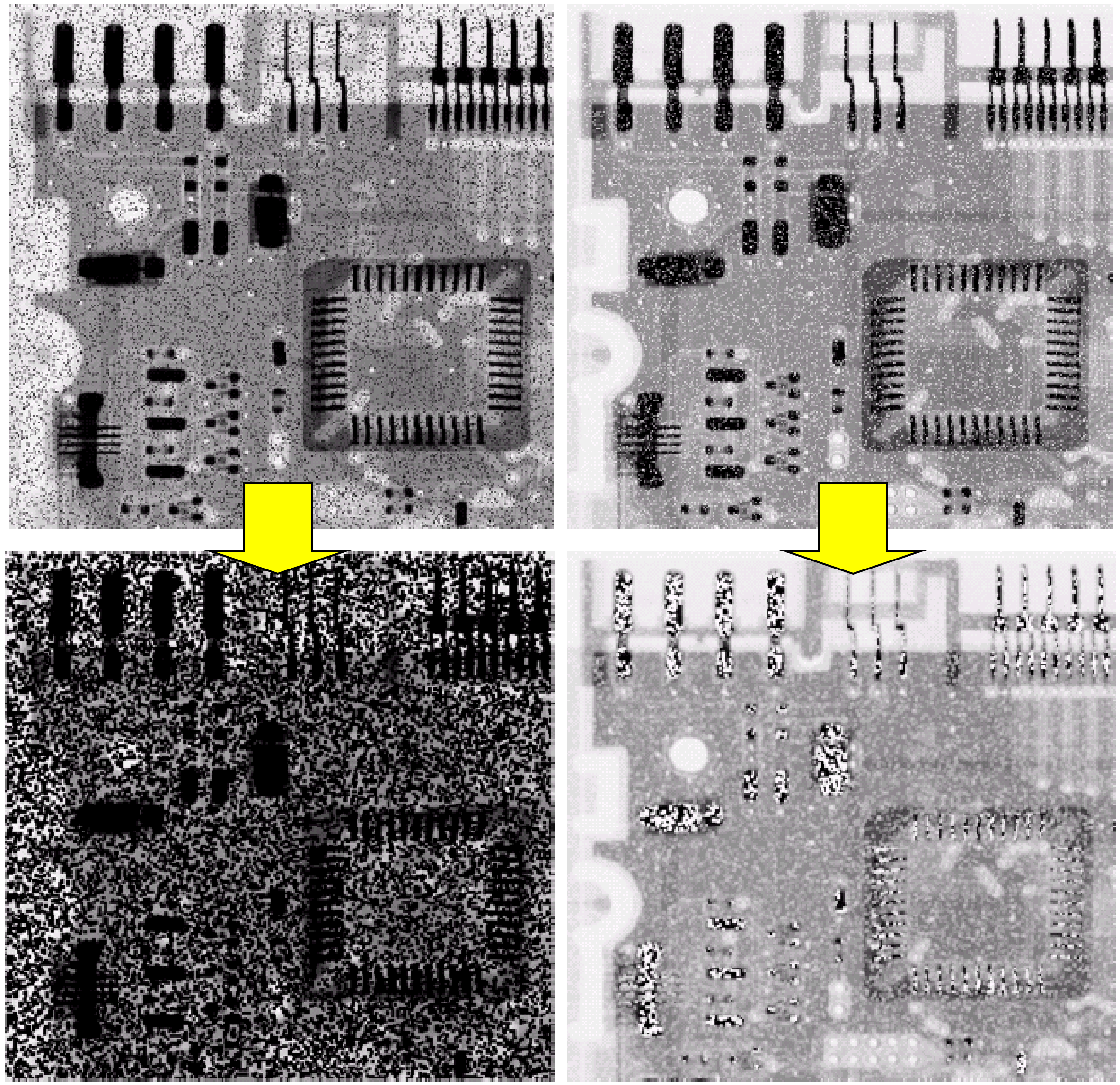


Image corrupted by salt noise with prob. = 0.1

Image obtained using a 3x3 contra-harmonic mean filter With Q=-1.5

Image obtained using a 3x3 contra-harmonic mean filter With Q=1.5

Restoration in the Presence of Noise Only

- **Order-Statistic Filters**

  Order-statistic filters are spatial filters whose response is based on ordering of the values of the pixels contained the image area under the filter mask

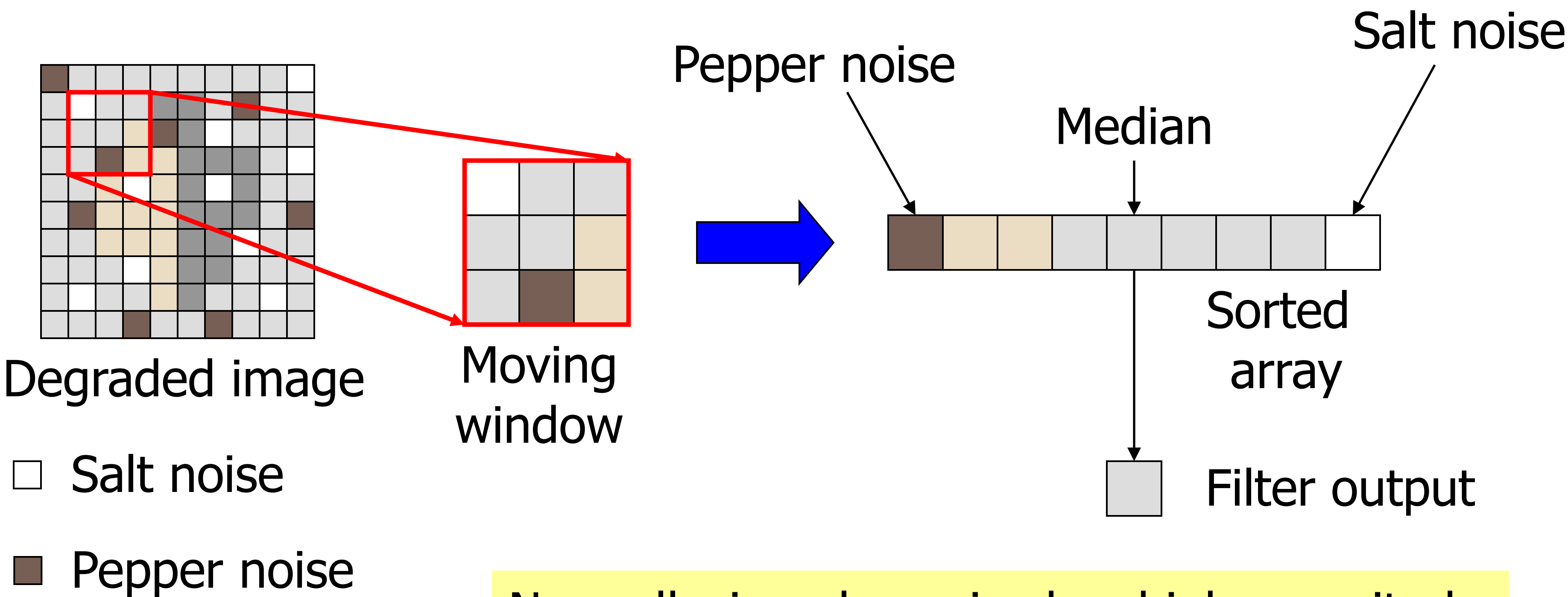
  - **Median Filter**
    - Replaces the value of the pixel by the median of the intensity levels in the neighborhood of the pixel

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

    - It's capable of reducing noise levels with considerably less blurring than linear smoothing filters
    - It's particularly effective on unipolar or bipolar impulsive noise

Restoration in the Presence of Noise Only

A median filter is good for removing impulse, isolated noise

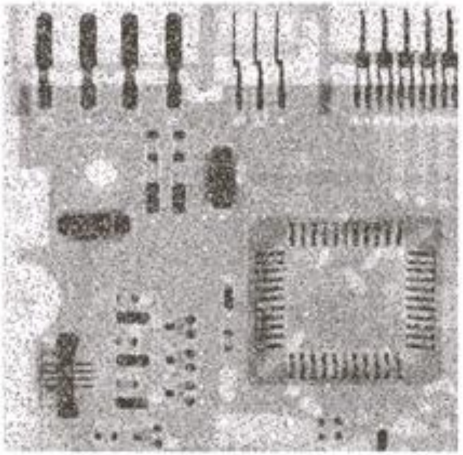


Normally, impulse noise has high magnitude and is isolated. When we sort pixels in the moving window, noise pixels are usually at the ends of the array.

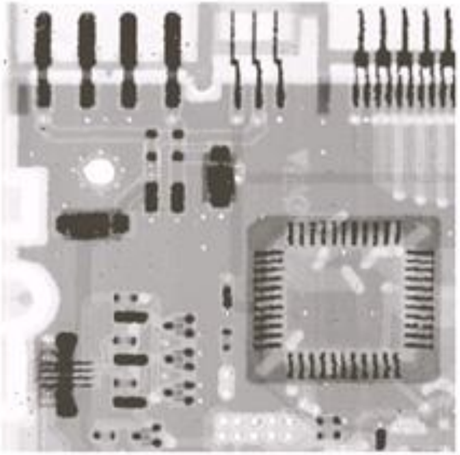
Therefore, it's rare that the noise pixel will be a median value.

- **Recursive median filtering**

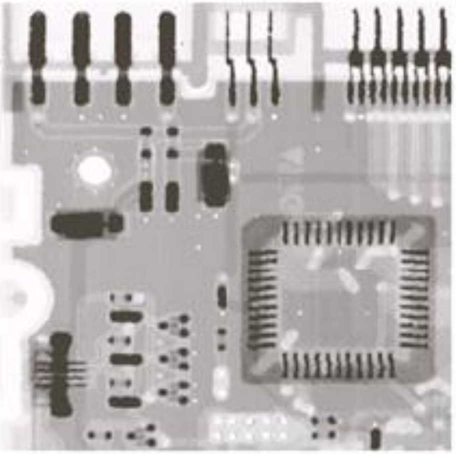
Image corrupted with pepper noise



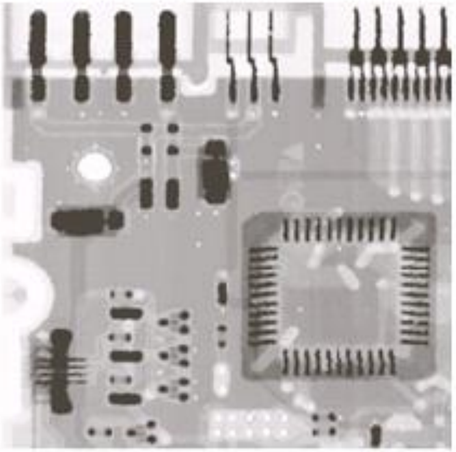
3x3 Median filtering 1st pass



3x3 Median filtering 2nd pass



3x3 Median filtering 3rd pass



Recursive median filtering is perfect for impulse noise. However, recursion filtering may lead to blurring

Restoration in the Presence of Noise Only

- **Order-Statistic Filters**
  - **Maximum Filter**
    - Replaces the value of the pixel by the maximum of the intensity levels in the neighborhood of the pixel

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

    - It is useful in finding the brightest points in the image and in reducing pepper noise
  - **Minimum Filter**
    - Replaces the value of the pixel by the minimum of the intensity levels in the neighborhood of the pixel

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

    - It is useful in finding the darkest points in the image and in reducing salt noise

Restoration in the Presence of Noise Only

- **Min and Max Filtering**

Image corrupted with pepper noise

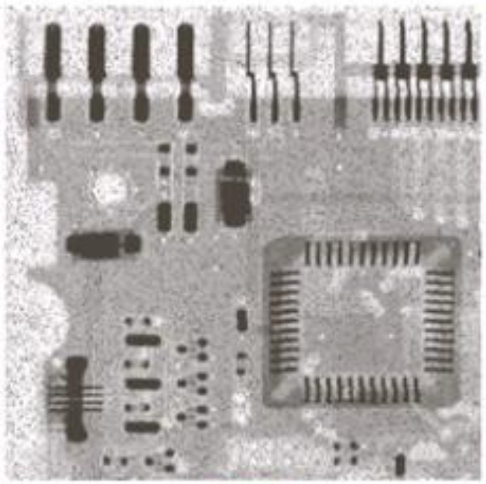


Image corrupted with white noise

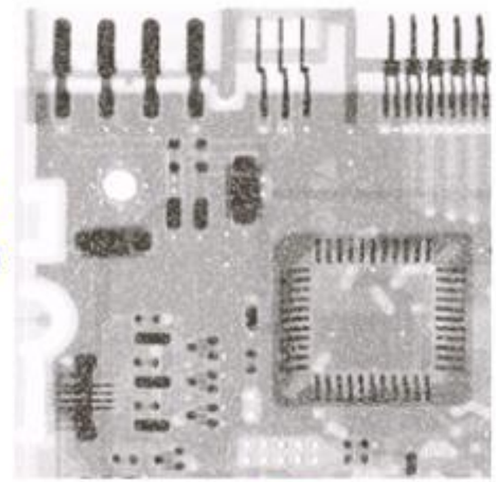


Image filtered with 3x3 max filter

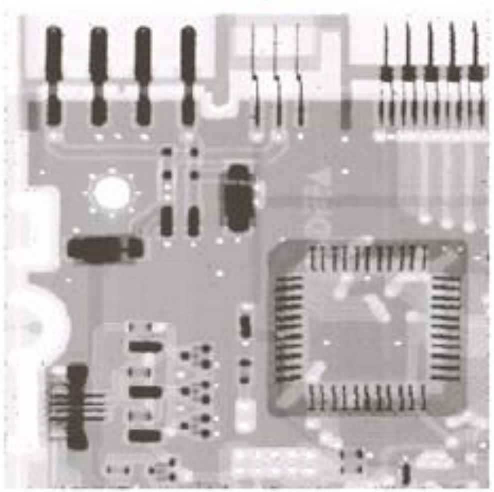
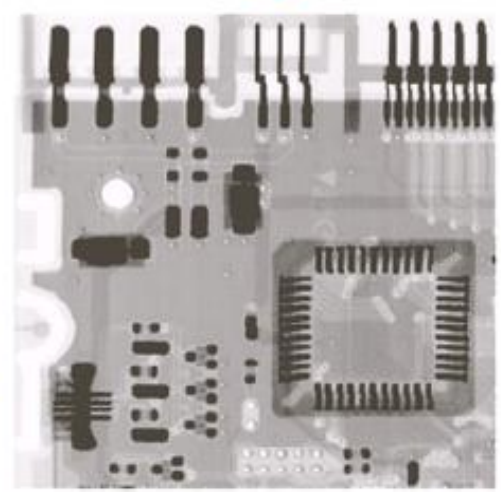


Image filtered with 3x3 min filter



Restoration in the Presence of Noise Only

- **Order-Statistic Filters**
  - **Midpoint Filter**

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

    - It works best for randomly distributed noise such as Gaussian and uniform noise
  - **Alpha-Trimmed Mean Filter**
    - Compute the mean intensity of the trimmed values for the pixels in the neighborhood. Trimming is performed by deleting d/2 lowest and d/2 highest intensity values

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

    - It is useful in situations involving multiple types of noise such as a combination of impulse and Gaussian noise

Restoration in the Presence of Noise Only

Image corrupted by additive uniform noise

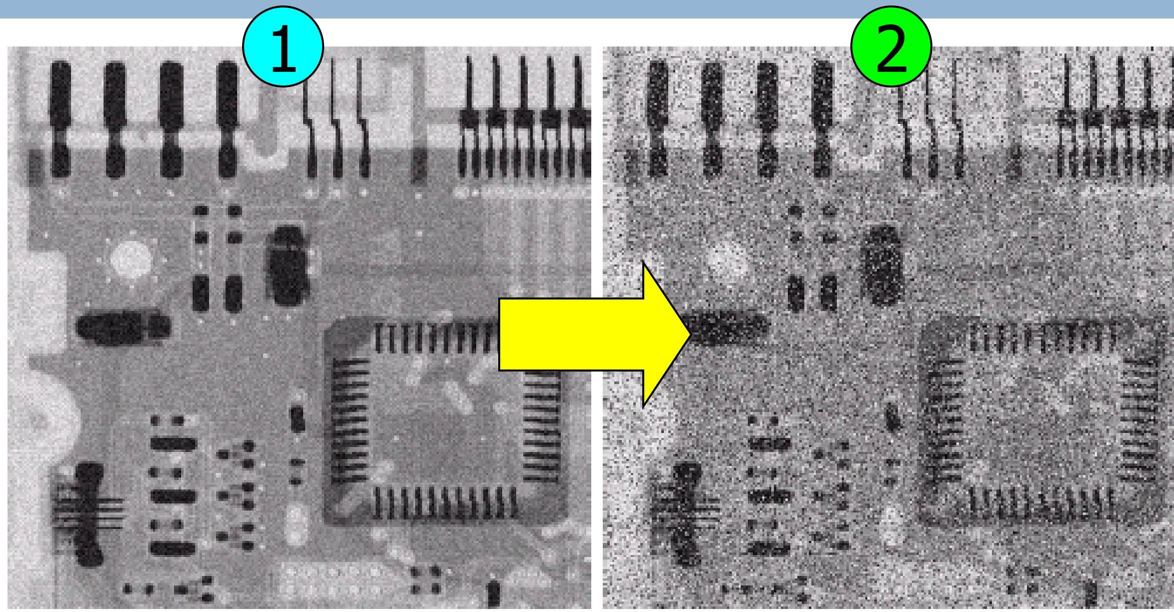
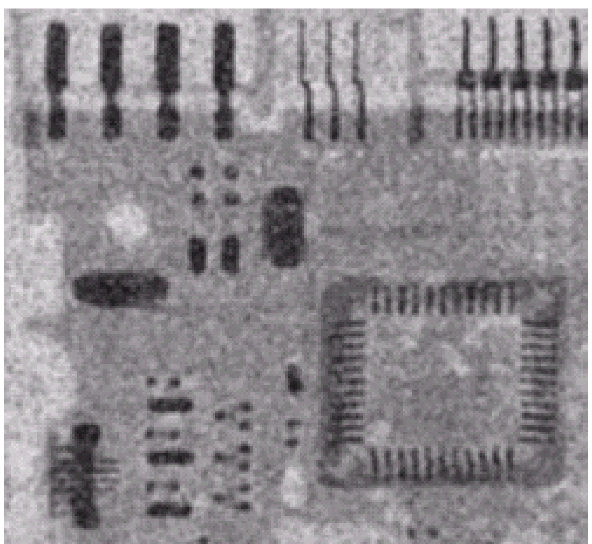


Image additionally corrupted by additive salt-and-pepper noise

Image 2 obtained using a 5x5 arithmetic mean filter



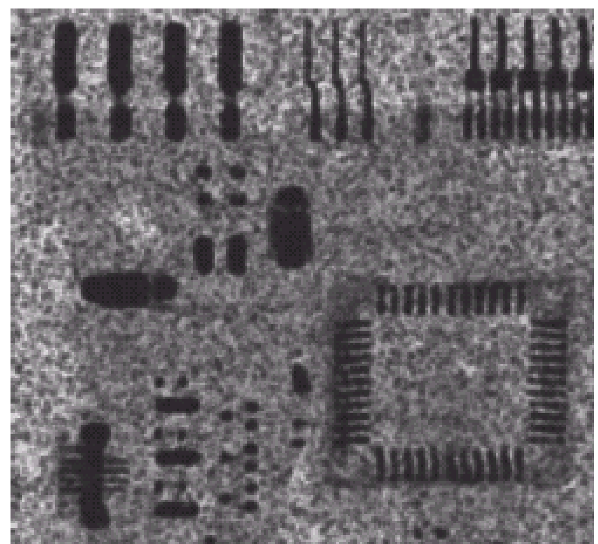
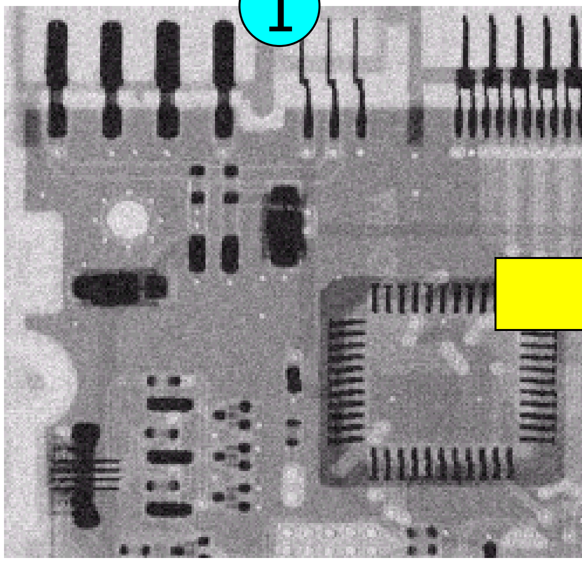


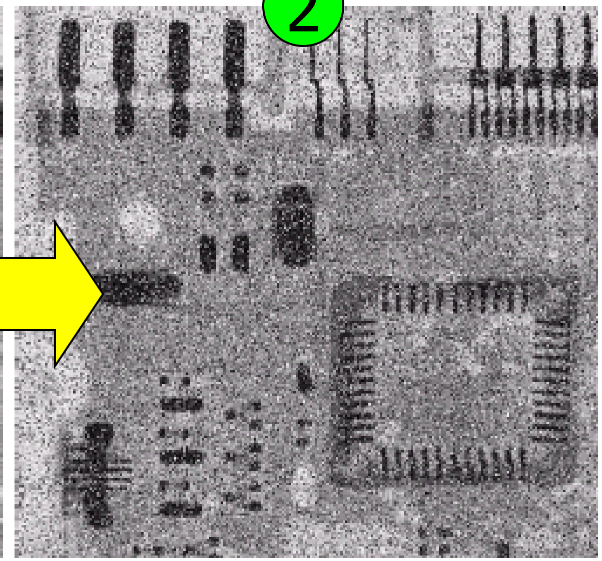
Image 2 obtained using a 5x5 geometric mean filter

Restoration in the Presence of Noise Only

Image corrupted by additive uniform noise



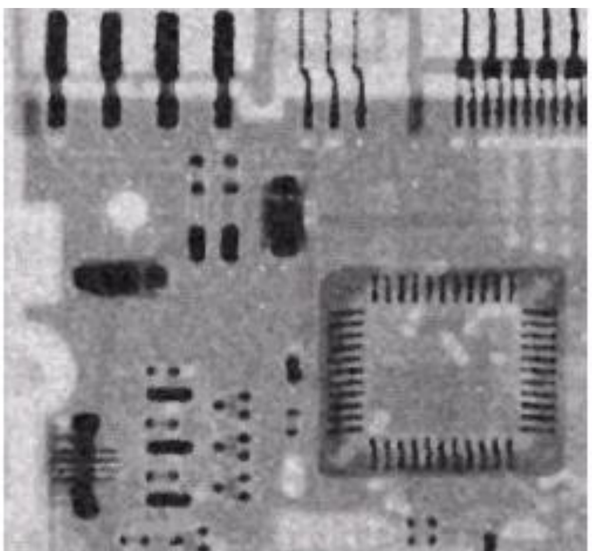
1



2

Image additionally corrupted by additive salt-and-pepper noise

Image 2 obtained using a 5x5 median filter



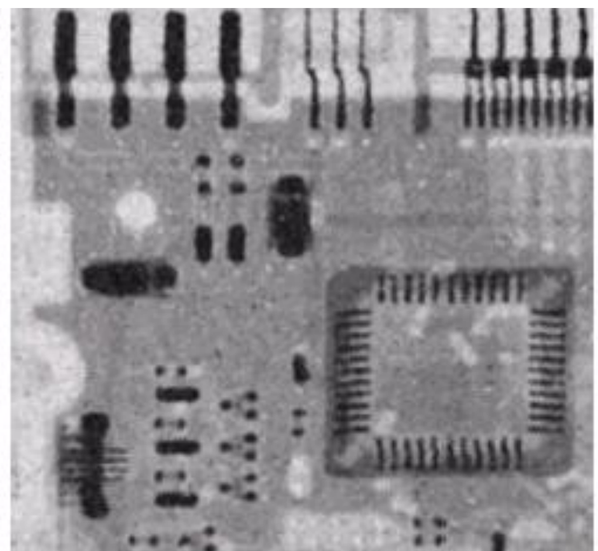


Image 2 obtained using a 5x5 alpha-trimmed mean filter with $d = 5$

- **Adaptive Filters**
  - The filters discussed so far are applied to an image without considering the fact that image characteristics may vary from one location to another
  - In what follow, we discuss a new class of filters called adaptive filters
  - In such filters, the filtering operation is modified from one location to another based on some local measures, such as the mean and variance of the pixel neighborhood
  - Adaptive filters are usually much better but at the expense of increased filter complexity

- **Adaptive , Local Noise Reduction Filter**
  - The mean intensity and variance are two statistical measures that are widely used because of their closely related relation with the appearance of the image
  - The mean gives the average intensity in the region while variance is a measure of contrast
  - The filter under discussion uses four different values to perform filtering in a certain neighborhood

    - g(x,y) : the value of the pixel (x,y) in the noisy image
    - σ_{η}^{2} : the variance of the noise corrupting f(x,y)
    - σ_{L}^{2} : the local variance for the pixels in S_{xy}
    - m_L : the local mean for the pixels in S_{xy}

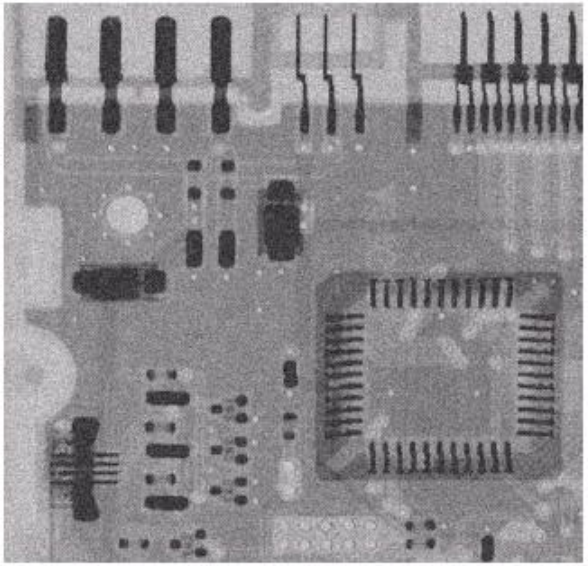
Restoration in the Presence of Noise Only

- **Adaptive , Local Noise Reduction Filter**
  - Assumptions of Operation
    - If σ_η^2 is zero, the filter should return the value of g(x,y). This is the trivial case in which there is no noise in the image
    - If the local variance is high relative to σ_η^2 , the filter should return a value close to g(x,y) since high local variance is usually associated with edges and these have to be preserved
    - If the two variances are equal, the filter should return the mean value of the pixels in S_{xy} . This situation correspond to the case when the local area has similar properties as the overall image, and local noise is to be removed by averaging.
  - Based on these assumptions, we can define this adaptive filter as

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

Restoration in the Presence of Noise Only

Image corrupted by additive Gaussian noise with zero mean and s^2=1000



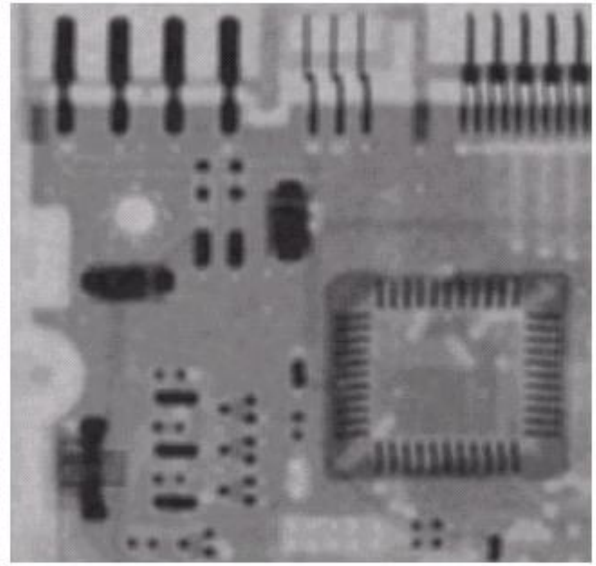
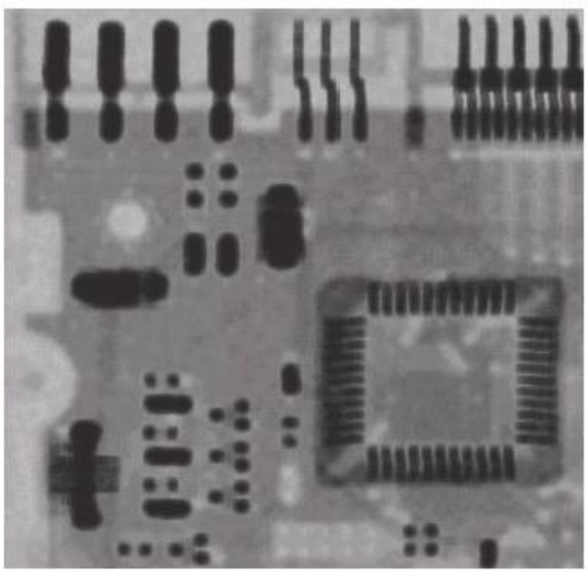


Image obtained using a 7x7 arithmetic mean filter

Image obtained using a 7x7 geometric mean filter



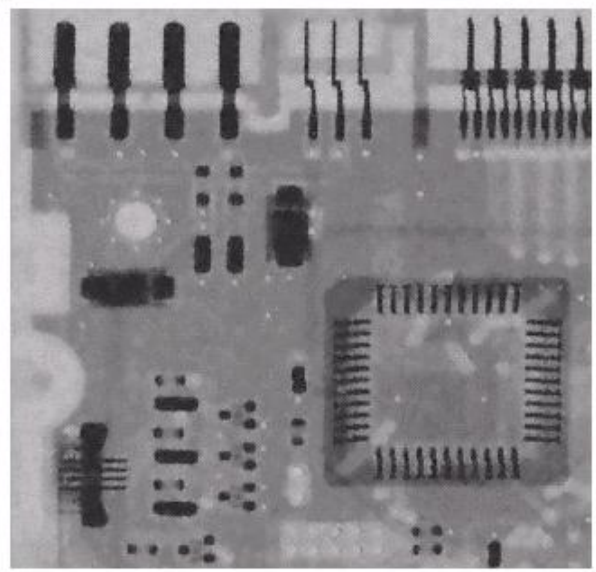


Image obtained using a 7x7 adaptive noise reduction filter

Adaptive Median Filter

Purpose: want to remove impulse noise while preserving edges

Algorithm: Level *A*:

$A1 = z_{median} - z_{min}$
$A2 = z_{median} - z_{max}$
If A1 > 0 and A2 < 0, goto level B
Else increase window size
If window size <= S_{max} repeat level A
Else return z_{xy}

Level *B*:

$B1 = z_{xy} - z_{min}$
$B2 = z_{xy} - z_{max}$
If B1 > 0 and B2 < 0, return z_{xy}
Else return z_{median}

where
z_{min} = minimum gray level value in S_{xy}
z_{max} = maximum gray level value in S_{xy}
z_{median} = median of gray levels in S_{xy}
z_{xy} = gray level value at pixel (x,y)
S_{max} = maximum allowed size of S_{xy}

Adaptive Median Filter

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

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

If A1 > 0 and A2 < 0, goto level B

Else → Window is not big enough

increase window size

If window size <= S_{max} repeat level A

Else return z_{xy}

Determine whether z_{median} is an impulse or not

Level B: → z_{median} is not an impulse

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

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

If B1 > 0 and B2 < 0,

return z_{xy} → to preserve original details

Else

return z_{median} → to remove impulse

Determine whether z_{xy} is an impulse or not

→ z_{xy} is not an impulse

Restoration in the Presence of Noise Only
Adaptive Median Filter

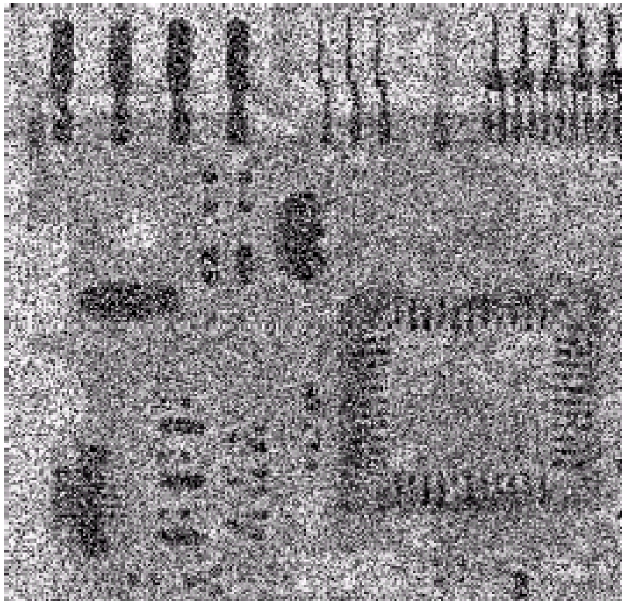


Image corrupted by salt-and-pepper noise with $p_a = p_b = 0.25$

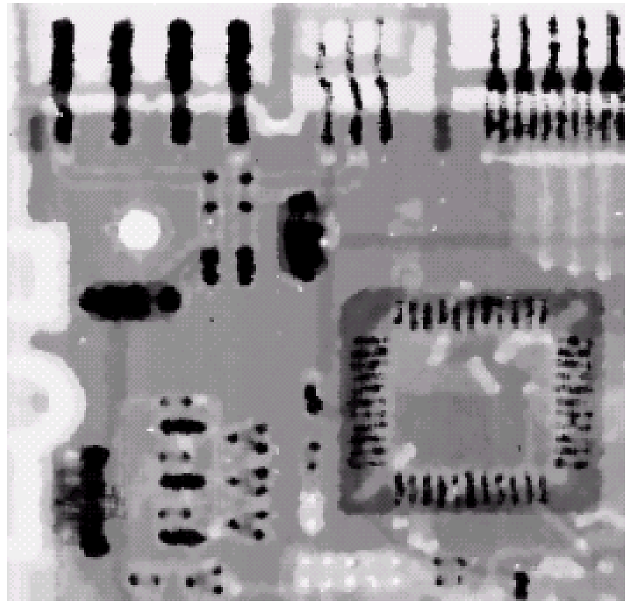


Image obtained using a 7x7 median filter

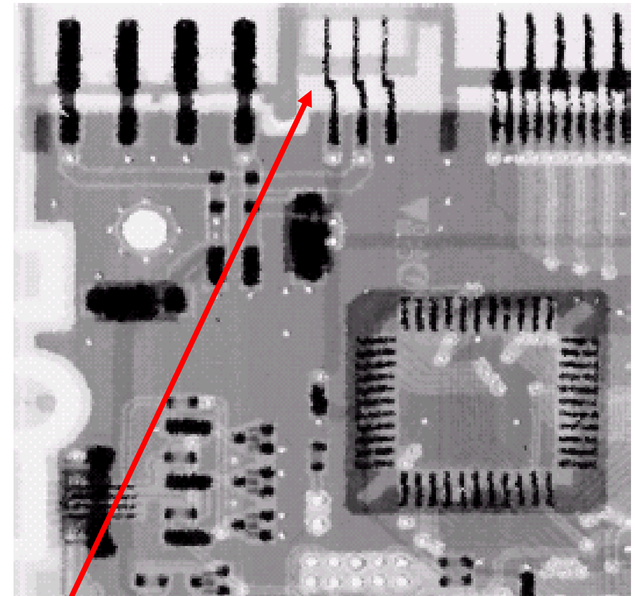


Image obtained using an adaptive median filter with $S_{max} = 7$

More small details are preserved

Periodic Noise Reduction By Frequency Domain Filtering

- Periodic noise appears as concentrated bursts of energy in the frequency domain at locations corresponding to the periodic frequencies

- Thus, periodic noise can be analyzed and filtered effectively using frequency domain filtering

- Use selective filtering
  - Bandpass filters
  - Notch filters

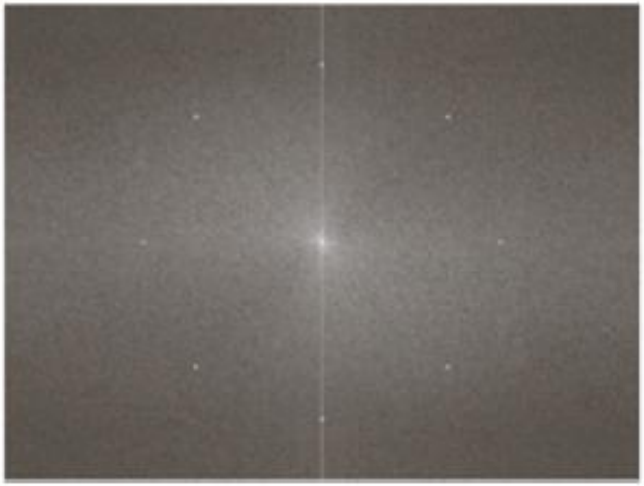
Periodic Noise Reduction By Frequency Domain Filtering

- **Bandreject Filters**
- One principle application of bandreject filters is in removing noise components whose locations are approximately known
- A good example is an image that is corrupted by periodic noise that can be approximated as two-dimensional sinusoidal functions
- We know that the Fourier transform of a sine consists of two impulses that are mirror images of each other about the origin of the transform
- We may use ideal, Butterworth, or Gaussian bandreject filters

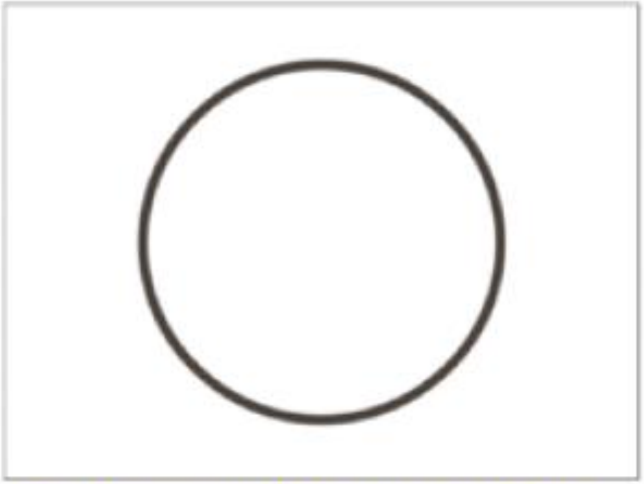
Periodic Noise Reduction By Frequency Domain Filtering

- **Bandreject Filters – Example**

Image corrupted Sinusoidal Noise



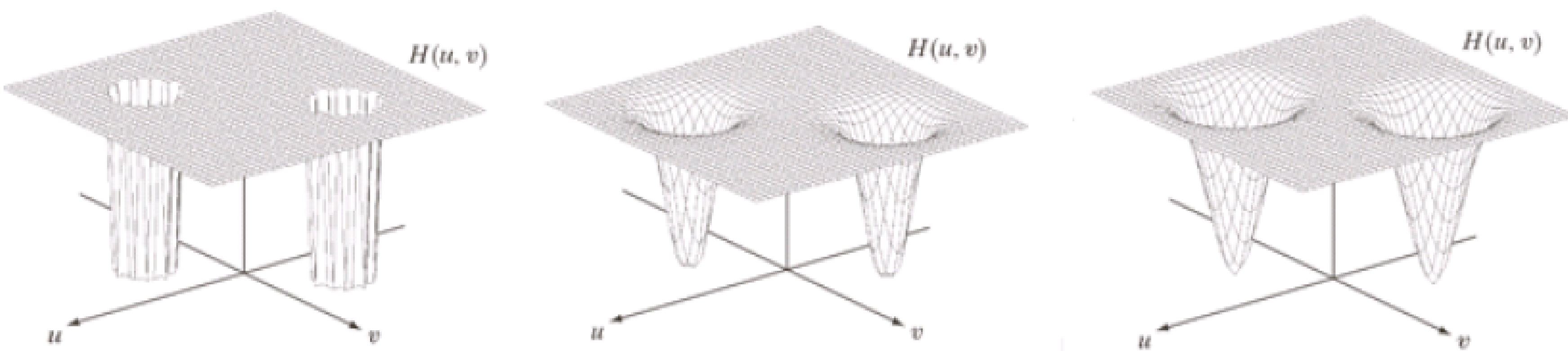
Magnitude of Spectrum



Butterworth Bandreject Filter

Filtered image

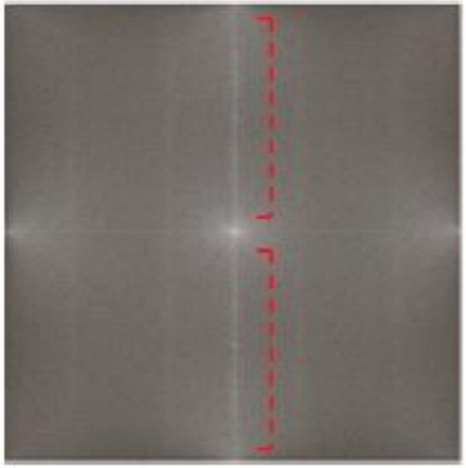
- **Notch Filters**
- Notch filters perform filtering by rejecting frequencies in a predefined neighborhoods about the center of the frequency rectangle
- Due to the symmetry of the Fourier transform, notch filters should appear symmetric about the origin
- We may use ideal, Butterworth, or Gaussian notch filters
- Notch filters can be arbitrary shape



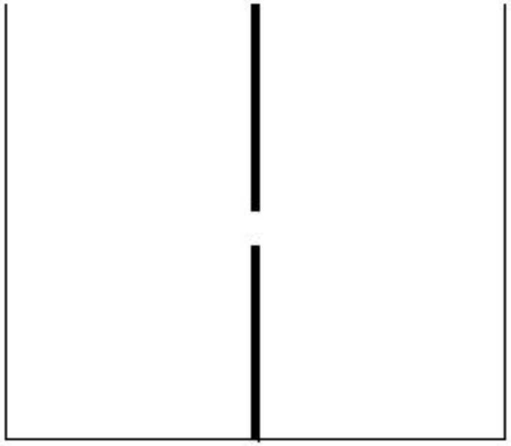
Periodic Noise Reduction By Frequency Domain Filtering

- **Notch-reject Filters – Example**

Image Corrupted by Sinusoidal Noise



Magnitude of Spectrum



Notch-reject Filter along the vertical axis

Filtered image

Estimation of Degradation Model

Degradation model:

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

or

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

Purpose: to estimate $h(x,y)$ or $H(u,v)$

Why? If we know exactly $h(x,y)$, regardless of noise, we can do deconvolution to get $f(x,y)$ back from $g(x,y)$.

Methods:

1. Estimation by Image Observation
2. Estimation by Experiment
3. Estimation by Modeling

Estimation of Degradation Function

- **Estimation by Image Observation**
  - Assume that a degraded image is available without any knowledge about the degradation function, which is assumed to be linear, position-invariant
  - One way to estimate the degradation function is to gather information from the image itself (like a blurred image)
  - For example, we can look at small patch of the image where the noise is minimum and then process the patch to arrive at pleasant result
  - From the degraded image and the processed image (the estimate of original) we compute

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

  - Then we can generalize to find H(u,v) with the same characteristics of Hs(u,v)

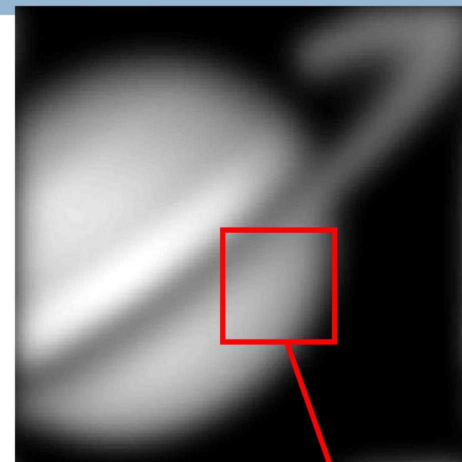
Estimation by Image Observation

Original image (unknown)

f(x,y)

f(x,y)*h(x,y)

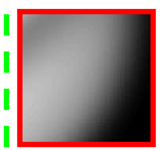
Degraded image



g(x,y)

Observation

Subimage



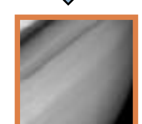
$g_s(x,y)$

DFT

$G_s(u,v)$

Restoration process by estimation

Reconstructed Subimage



$\hat{f}_s(x,y)$

DFT

$\hat{F}_s(u,v)$

Estimated Transfer function

$$H(u,v) \approx H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

This case is used when we know only g(x,y) and cannot repeat the experiment!

Estimation of Degradation Function

- **Estimation by Experimentation**
  - It is possible if the imaging system used to acquire the degraded image is available
  - Imaging is repeated with different system settings until the almost the same degraded image is obtained
  - Using such settings, the system is used to image an impulse (a small dot of light that is as bright as possible) to obtain the impulse response of the degradation
  - In the frequency domain, we can compute the degradation function H(u,v) from the degraded image by

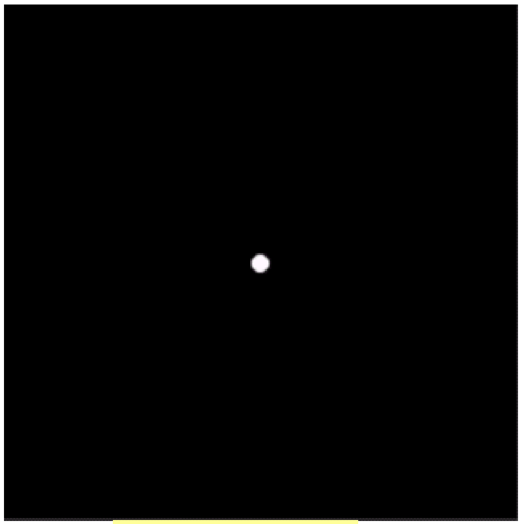
$$H(u,v) = \frac{G(u,v)}{A}$$

where A is the impulse strength

Estimation by Experiment

Used when we have the same equipment set up and can repeat the experiment.

Input impulse image



$A\delta(x,y)$

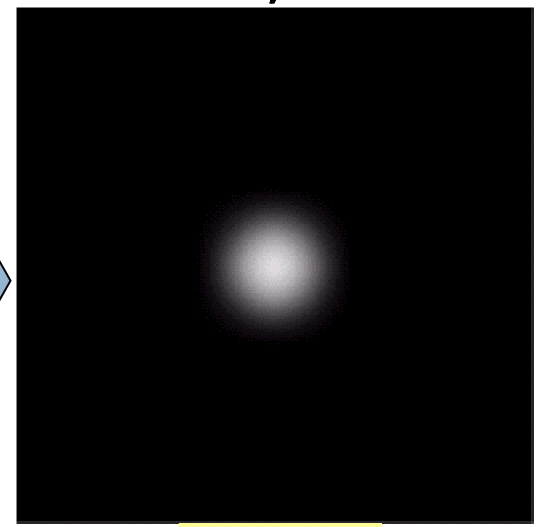
DFT

$DFT\{A\delta(x,y)\} = A$

System

$H(\,)$

Response image from the system



$g(x,y)$

DFT

$G(u,v)$

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

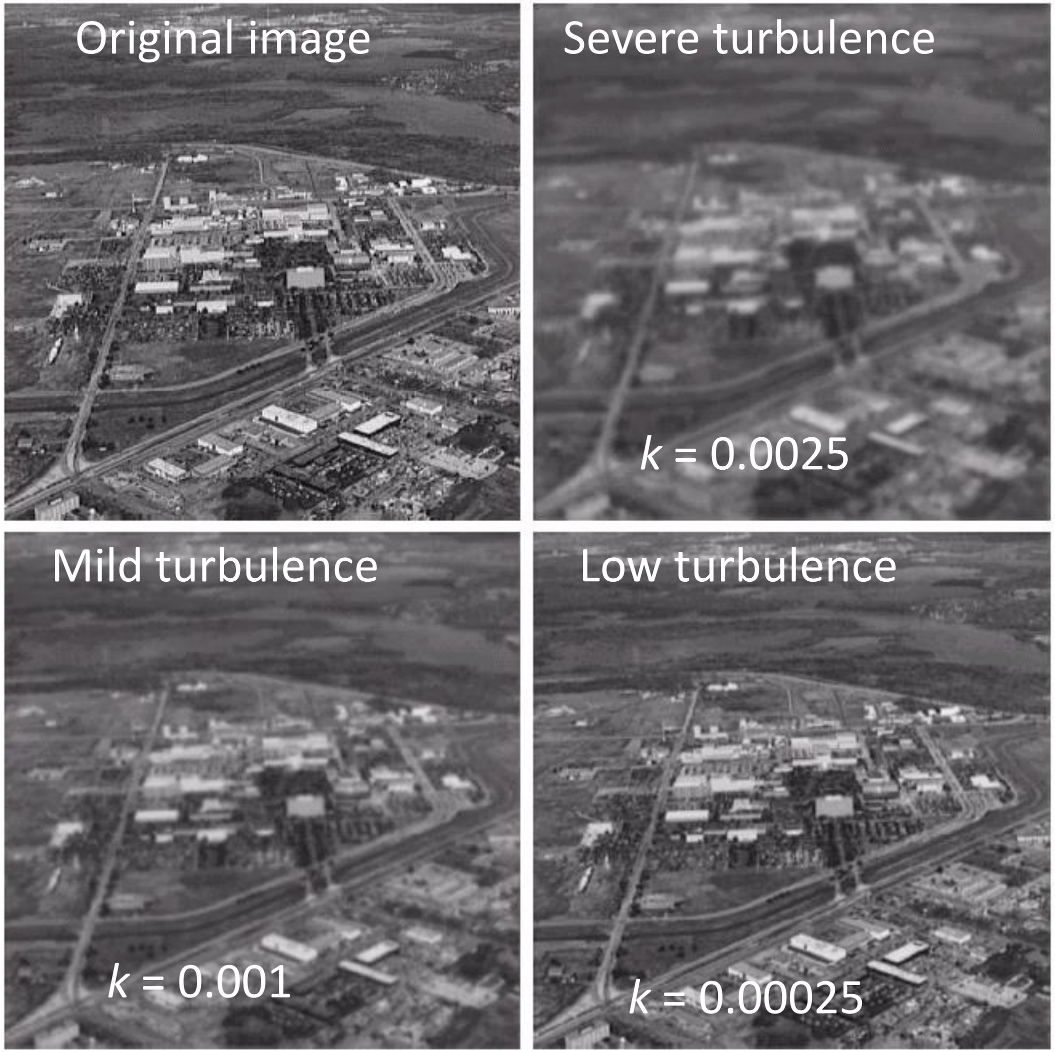
- **Estimation by Modeling**
  - It is based on finding a mathematical expression for the degradation function
  - Finding such mathematical expression can be through
    - some assumptions such as physical characteristics of the environment during the imaging process
    - deriving the mathematical model from basic principles
  - For example, a model proposed to model atmospheric turbulence is

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

where k is a constant that depends on the nature of the turbulence with higher values indicating higher turbulence effects

Estimation of Degradation Function

Used when we know physical mechanism underlying the image formation process that can be expressed mathematically.



Example:

Atmospheric Turbulence model

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

- **Estimation by Modeling - Example**
  - **Estimation of blurring degradation based on linear uniform motion**
  - Assume that the image function undergoes a planner motion and that $x_0(t)$ and $y_0(t)$ are the time-dependant components of motion in the x and y directions
  - The total exposure at any point of the recording medium is obtained by integrating the instantaneous exposure over the time interval of imaging event
  - If T is the duration of exposure , it follows that

$$g(x,y) = \int_0^T f(x - x_0(t), y - y_0(t))dt$$

- **Estimation by Modeling - Example**
  - **Estimation of blurring degradation based on linear uniform motion**
  - The Fourier transform of g(x,y) is given by

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

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{T} f\left[x-x_0(t),y-y_0(t)\right]dt\; e^{-j2\pi(ux+vy)}dxdy$$

  - Reversing the order of integration

$$G(u,v) = \int_{0}^{T}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left[x-x_0(t),y-y_0(t)\right]e^{-j2\pi(ux+vy)}dxdy\right]dt$$

- **Estimation by Modeling - Example**
  - **Estimation of blurring degradation based on linear uniform motion**
  - The term inside the outer brackets from the previous slide is the Fourier transform of the displaced function $f[x-x_0(t),y-y_0(t)]$

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

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

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

  - If $x_0(t)$ and $y_0(t)$ are known, the degradation function is can be directly found

- **Estimation by Modeling - Example**
  - **Estimation of blurring degradation based on linear uniform motion**
  - Let the image undergoes a uniform motion in the x direction only with $x_0(t) = at/T$, then the degradation function is given by

$$H(u,v) = \frac{1}{\pi ua} sin(\pi ua) e^{-j\pi ua}$$

  - If we also have a uniform motion in the y direction also by $y_0(t) = bt/T$, then the degradation function is

$$H(u,v) = \frac{1}{\pi(ua + vb)} sin(\pi(ua + bv)) e^{-j\pi(ua+vb)}$$

Estimation of Degradation Function

- **Estimation by Modeling - Example**
  - **Estimation of blurring degradation based on linear uniform motion**

Example of blurring by the function on the previous slide with a = b = 0.1 and T = 1

Inverse Filter

From degradation model: $G(u,v) = F(u,v)H(u,v) + N(u,v)$

after we obtain $H(u,v)$, we can estimate $F(u,v)$ by the inverse filter:

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

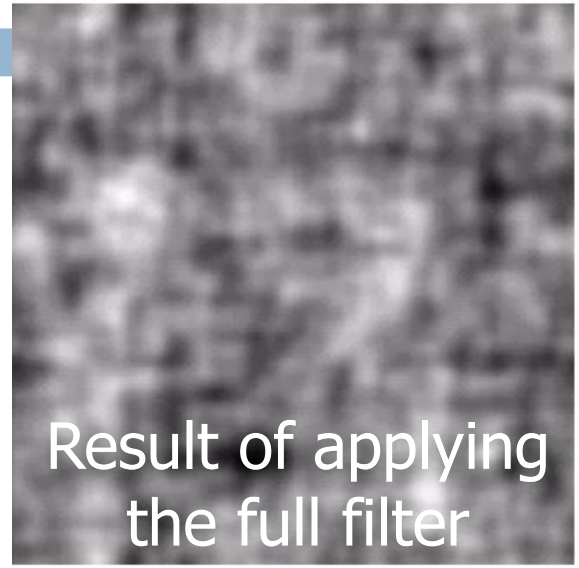
Noise is enhanced when $H(u,v)$ is small.

To avoid the side effect of enhancing noise, we can apply this formulation to freq. component (u,v) with in a radius D_0 from the center of $H(u,v)$.

In practical, the inverse filter is not Popularly used.

Inverse Filter: Example

Original image

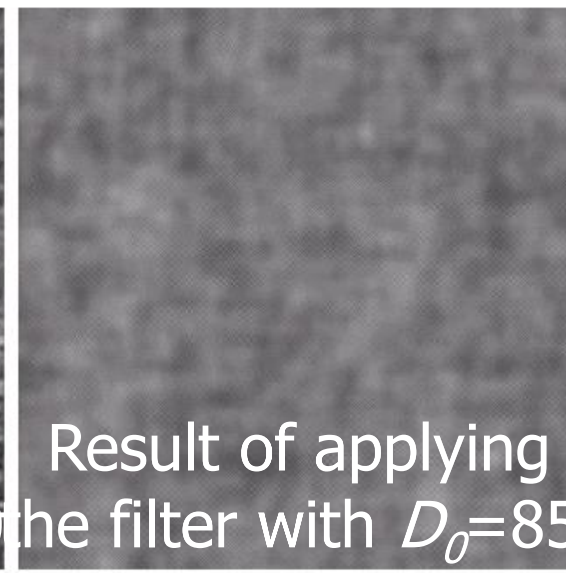


Result of applying the full filter

Result of applying the filter with D_0=40

Blurred image Due to Turbulence

Result of applying the filter with D_0=70



Result of applying the filter with D_0=85

$$H(u,v) = e^{-0.0025(u^2+v^2)^{5/6}}$$

Wiener Filter: Minimum Mean Square Error Filter

Objective: optimize mean square error: $e^2 = E\left\{(f - \hat{f})^2\right\}$

Wiener Filter Formula:

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

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

$$= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

where

$H(u,v)$ = Degradation function
$S_\eta(u,v)$ = Power spectrum of noise
$S_f(u,v)$ = Power spectrum of the undegraded image

Approximation of Wiener Filter

Wiener Filter Formula:

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

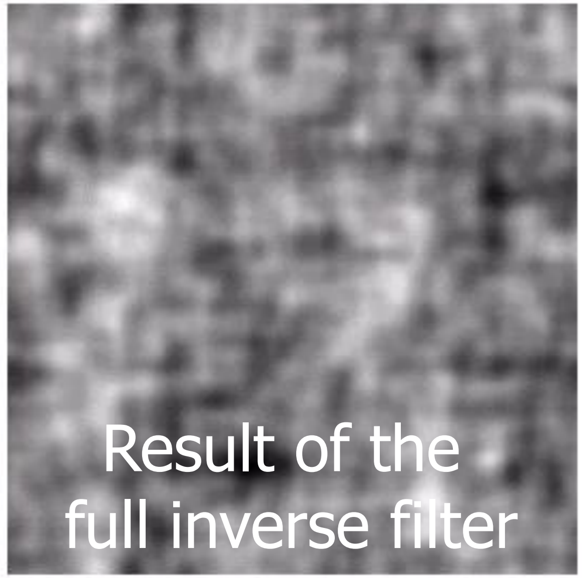
Difficult to estimate

Approximated Formula:

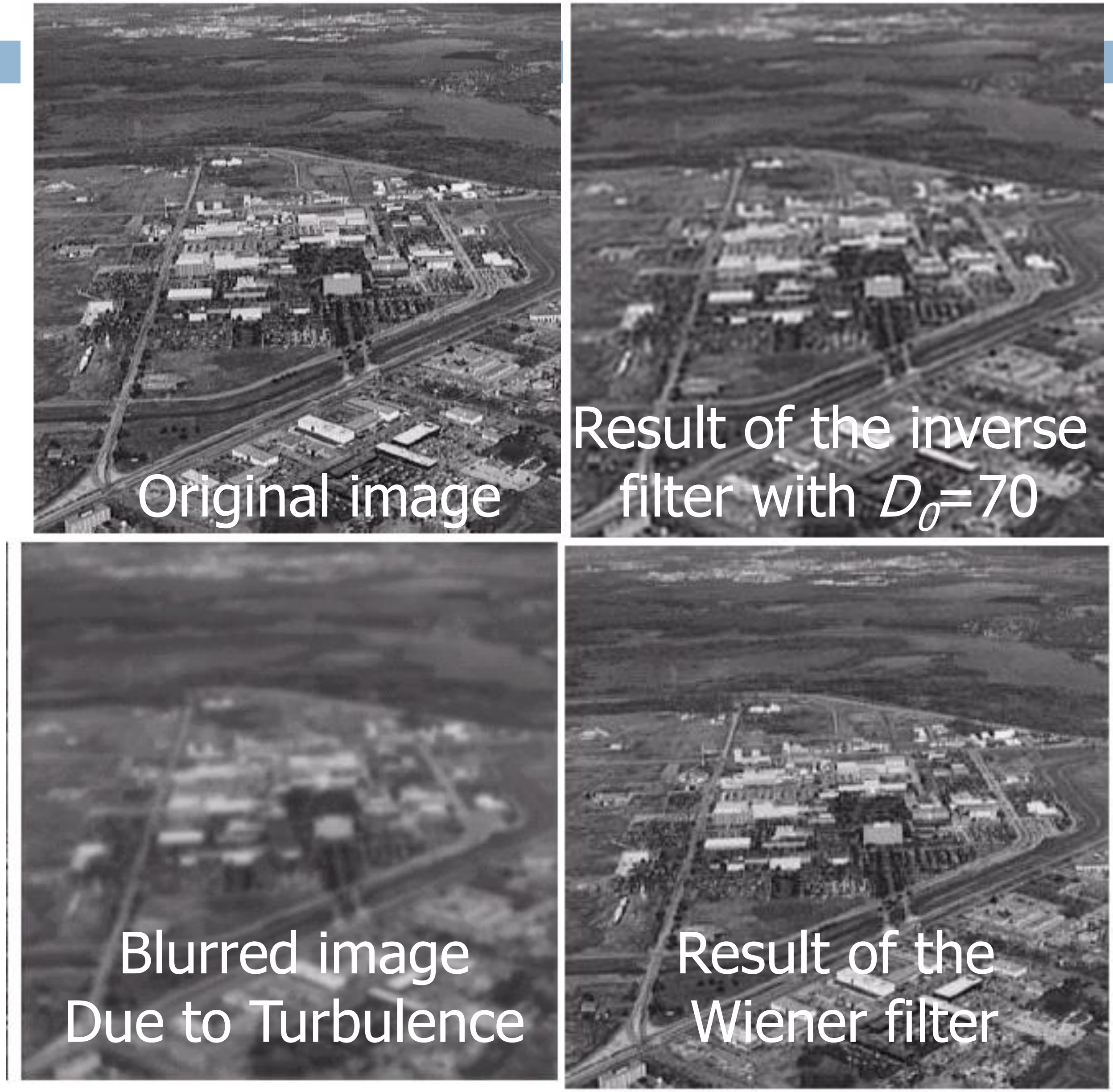
$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \right] G(u,v)$$

Practically, *K* is chosen manually to obtained the best visual result!

Wiener Filter: Example

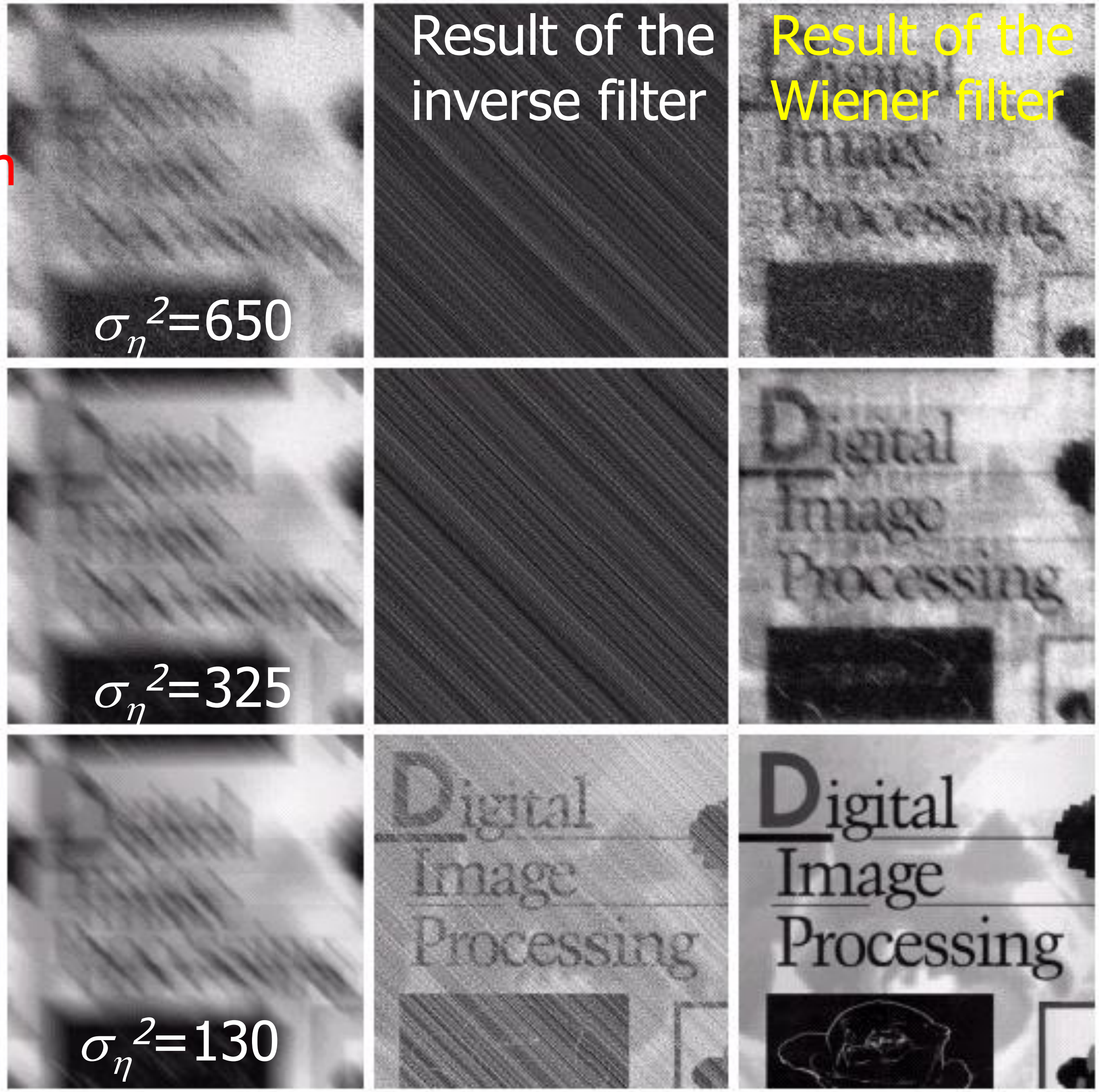


Wiener Filter: Example (cont.)



Example: Wiener Filter and Motion Blurring

Image degraded by motion blur + AWGN



Note: *K* is chosen manually