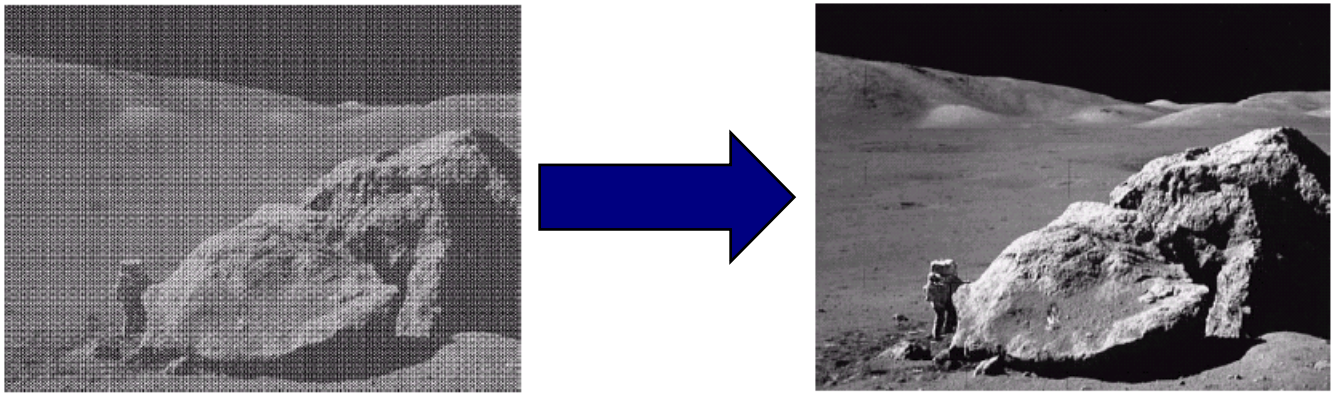
Image & Video Processing

Image Restoration:
Noise Removal

What is Image Restoration?

Image restoration attempts to restore images that have been degraded

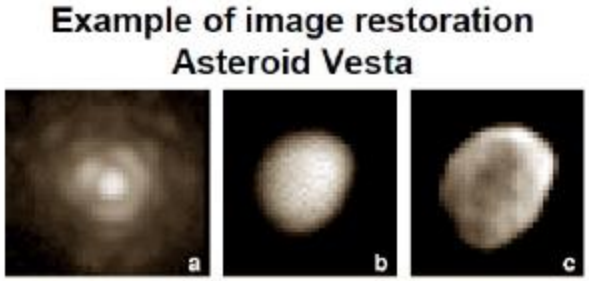
- Identify the degradation process and attempt to remove it in order to go back to the "original"
- Similar to image enhancement, but more objective



Applications

Started from the 1950s

- Scientific explorations
- Legal investigations
- Film making and archival
- Image and video (de-)coding
- …
- Consumer photography



Degradation causes

Degradation examples:

- original

- optical blur

- motion blur

- spatial quantization (discrete pixels)

- additive intensity noise

Causes:

- Camera: translation, shake, out-of-focus …
- Environment: scattered and reflected light
- Device noise: CCD/CMOS sensor and circuitry
- Quantization noise
- Transmission error

A Model for Image Distortion/Restoration

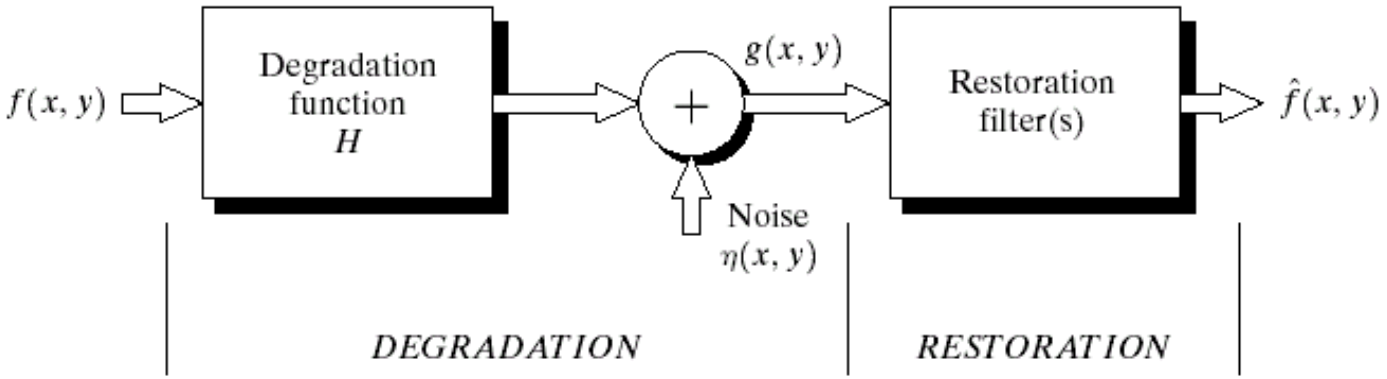


FIGURE 5.1 A model of the image degradation/restoration process.

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

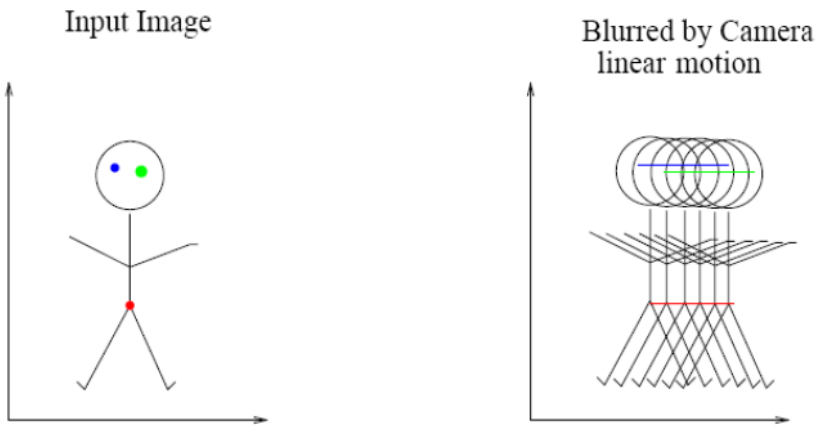
where $f(x, y)$ is the original image pixel, H is the degradation function, $\eta(x, y)$ is the noise term and $g(x, y)$ is the resulting noisy pixel

Assumptions for the Distortion Model

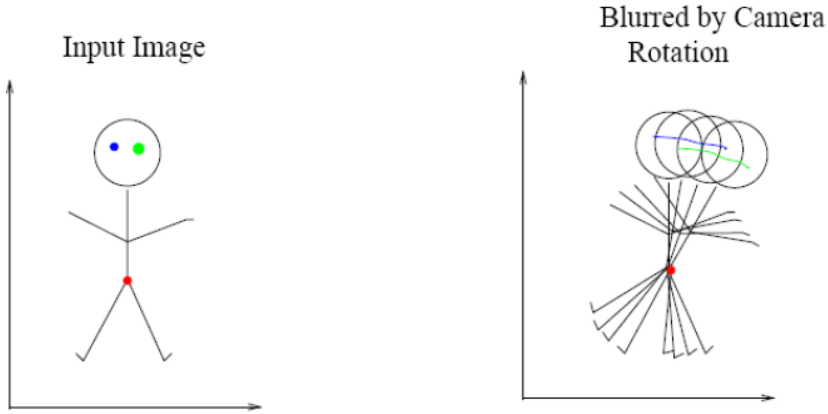
- Noise
  - Independent of spatial location
    - Exception: periodic noise ..
  - Uncorrelated with image
- Degradation function H
  - Linear
  - Position-invariant

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

Step #1: image degraded only by noise.



SPACE-INVARIENT RESPONSE - each point on image gives same response just shifted in position.

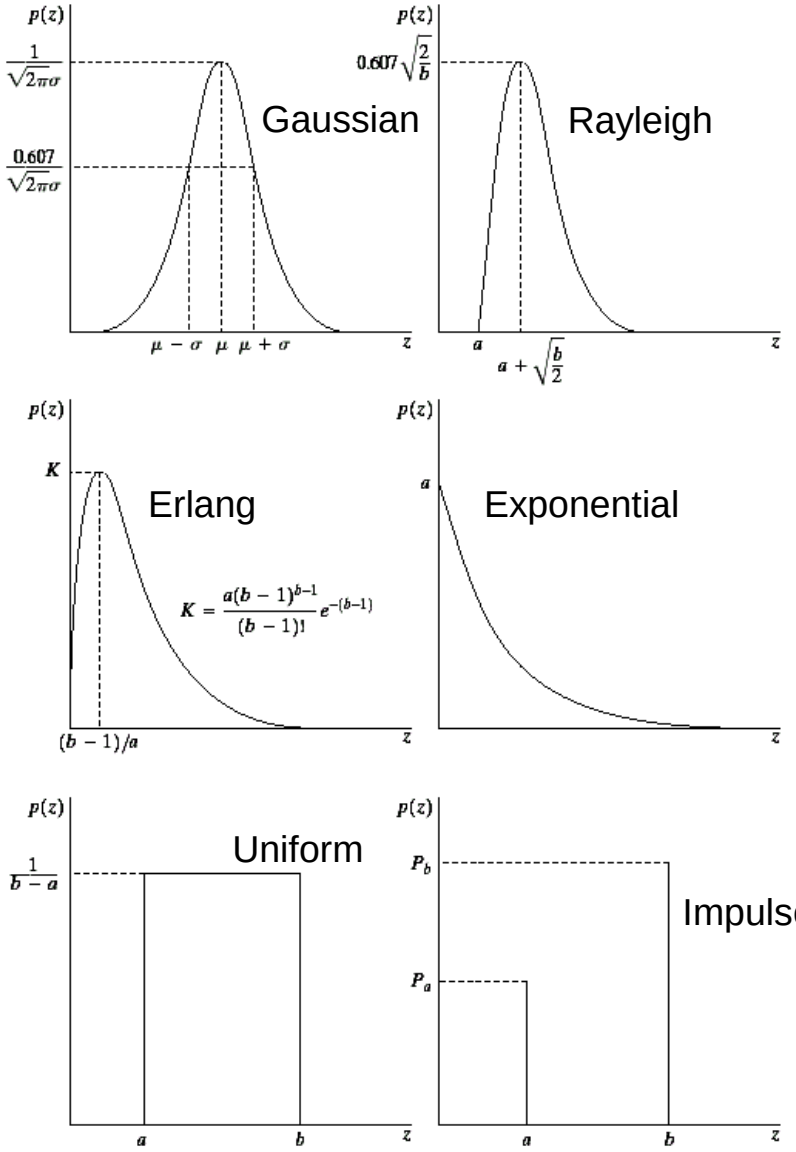


SPACE-VARIENT RESPONSE - each point on image gives a different response

Noise Models

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

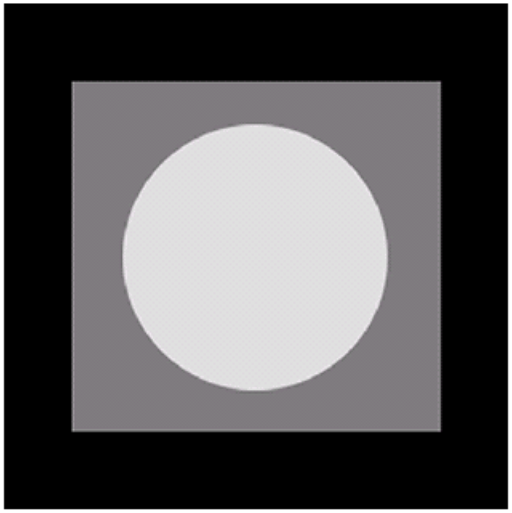
- Gaussian
  - Most common model
- Rayleigh
- Erlang
- Exponential
- Uniform
- Impulse
  - *Salt and pepper* noise



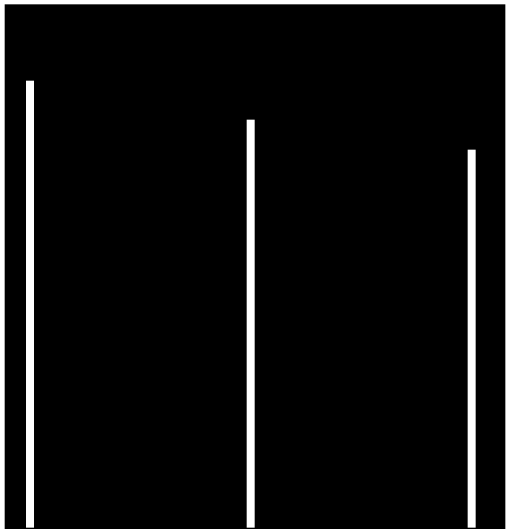
Noise Example

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

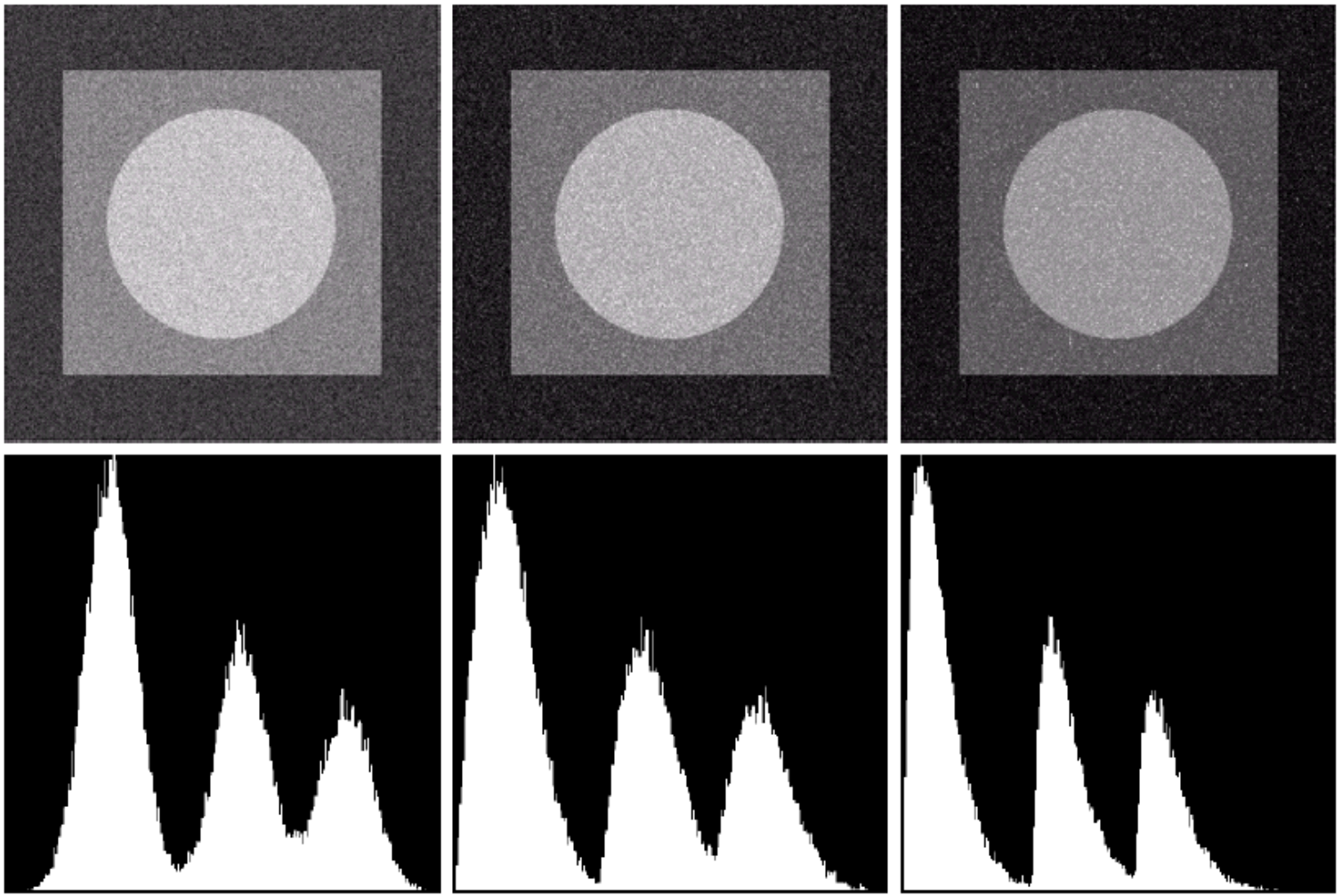


Image



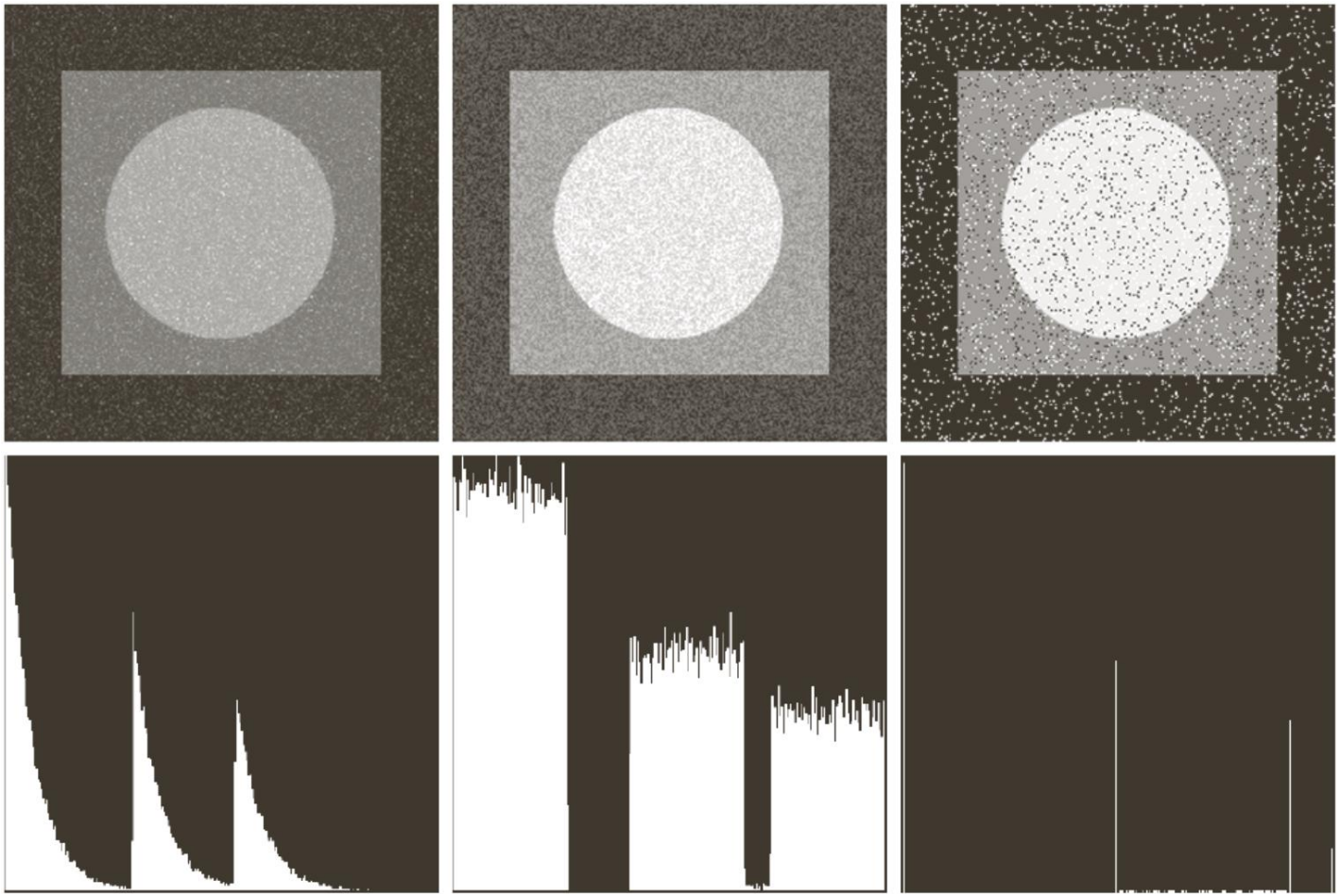
Histogram

Noise Example (cont…)



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

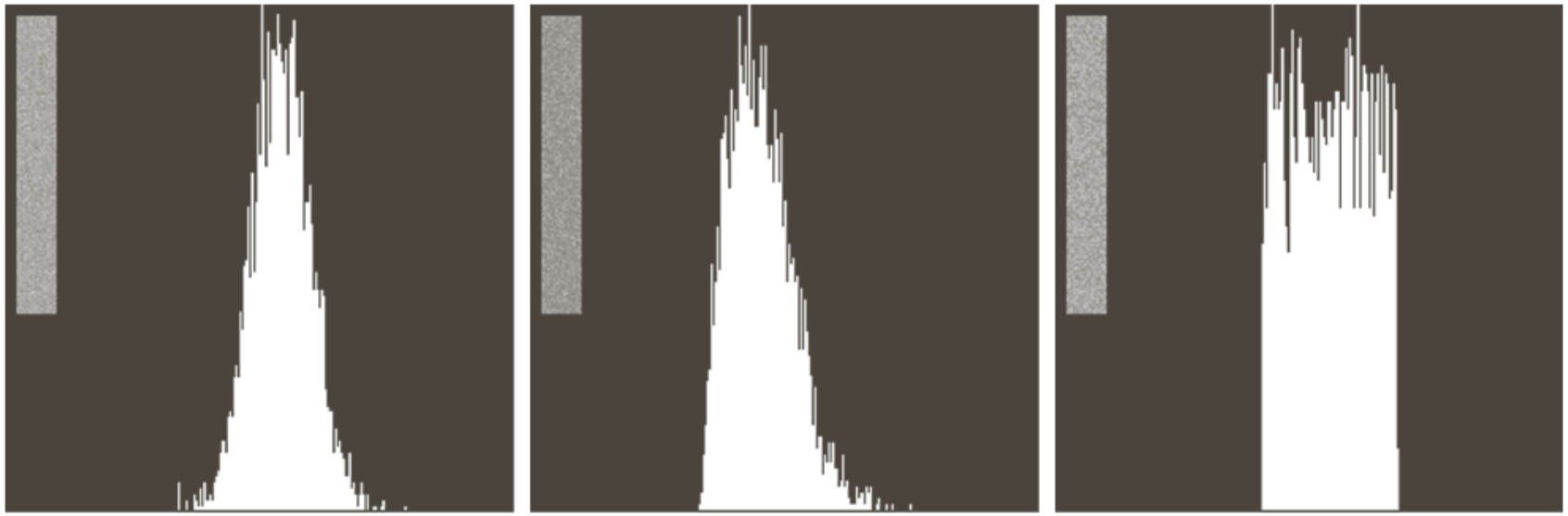
Noise Example (cont…)



Exponential Uniform Impulse

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

Estimation of noise parameters



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

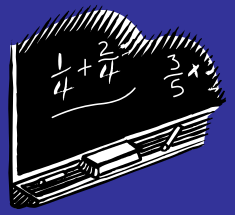
This is implemented as the simple smoothing filter

Blurs the image to remove noise

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

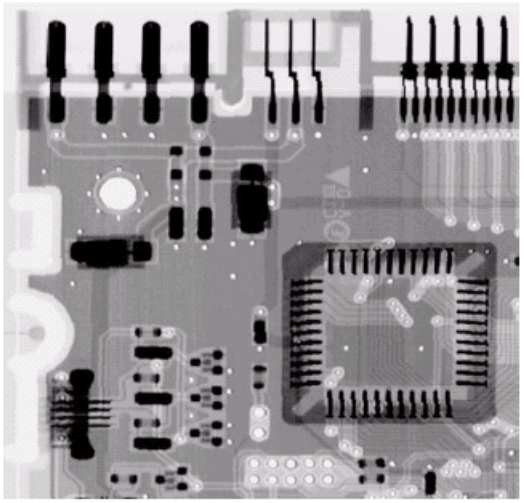
Geometric Mean:

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

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

Noise Removal Examples

Original Image



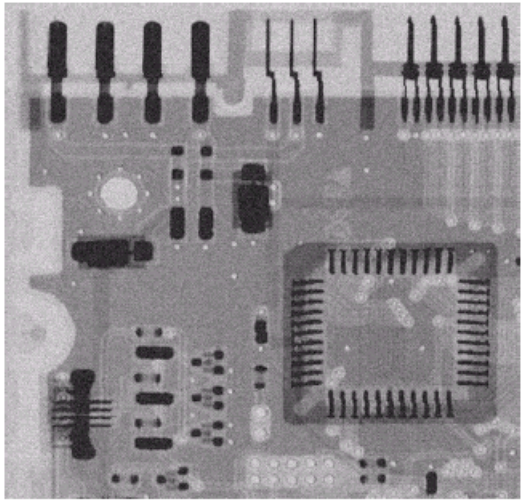
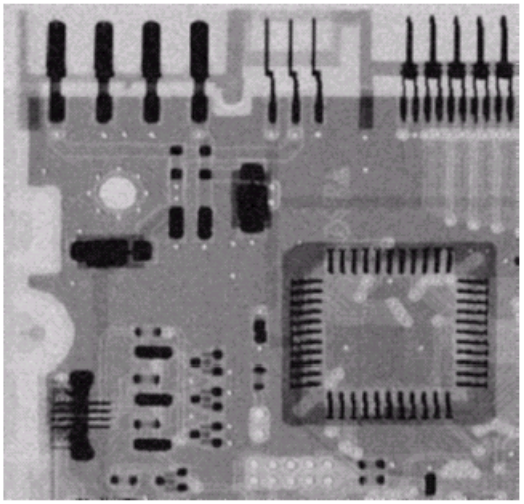
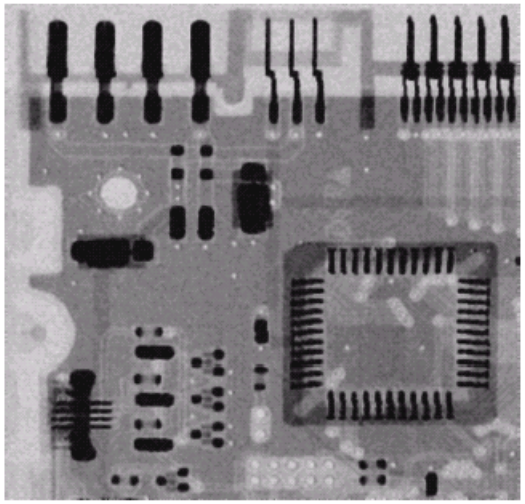


Image Corrupted By Gaussian Noise

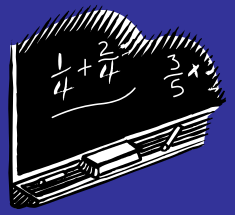
After A 3*3 Arithmetic Mean Filter





After A 3*3 Geometric Mean Filter

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



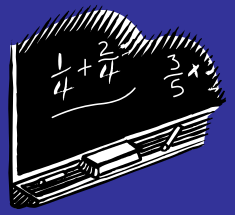
Other Means (cont…)

Harmonic Mean:

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

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:

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

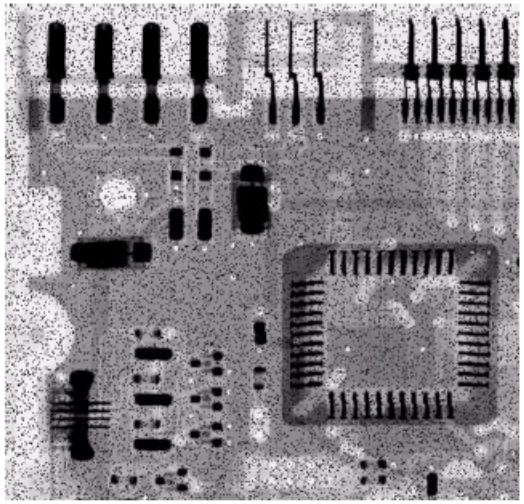
Q is the *order* of the filter and adjusting its value changes the filter's behaviour

Positive values of Q eliminate pepper noise

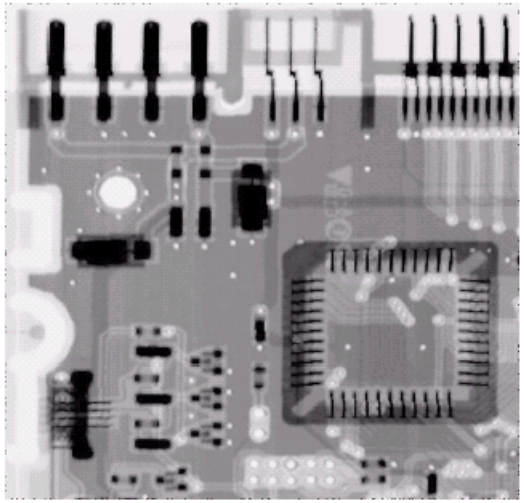
Negative values of Q eliminate salt noise

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



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



Noise Removal Examples (cont…)

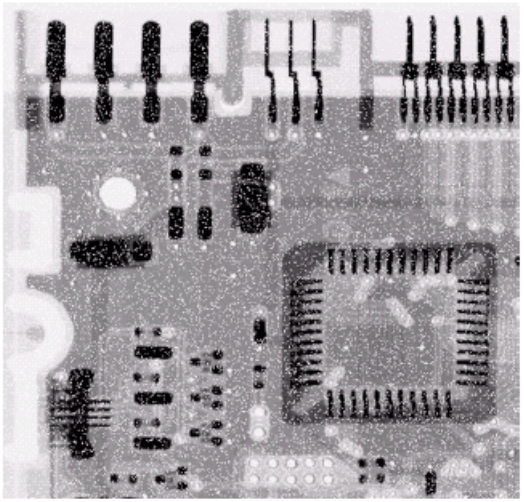
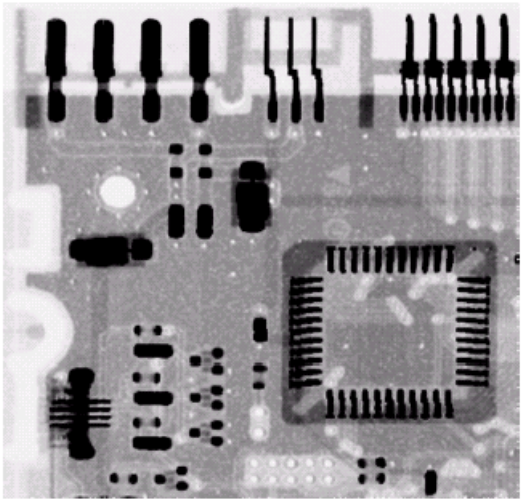


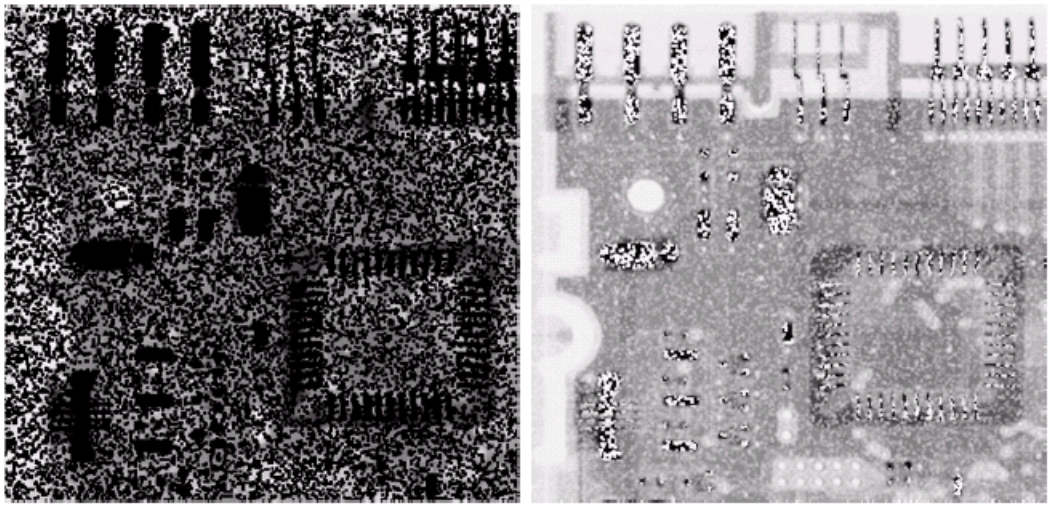
Image Corrupted By Salt Noise



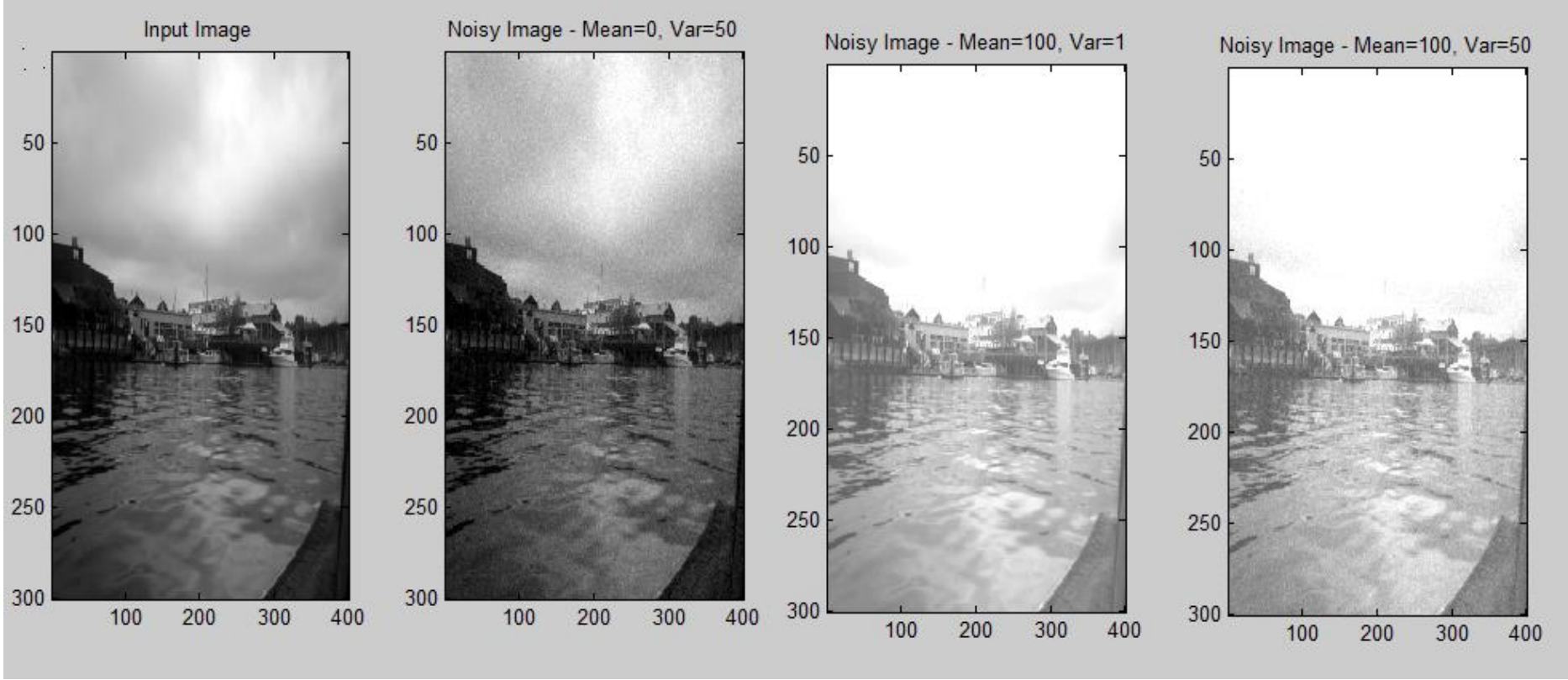
Result of Filtering Above With 3*3 Contraharmonic Q=-1.5

Contraharmonic Filter: Caution!

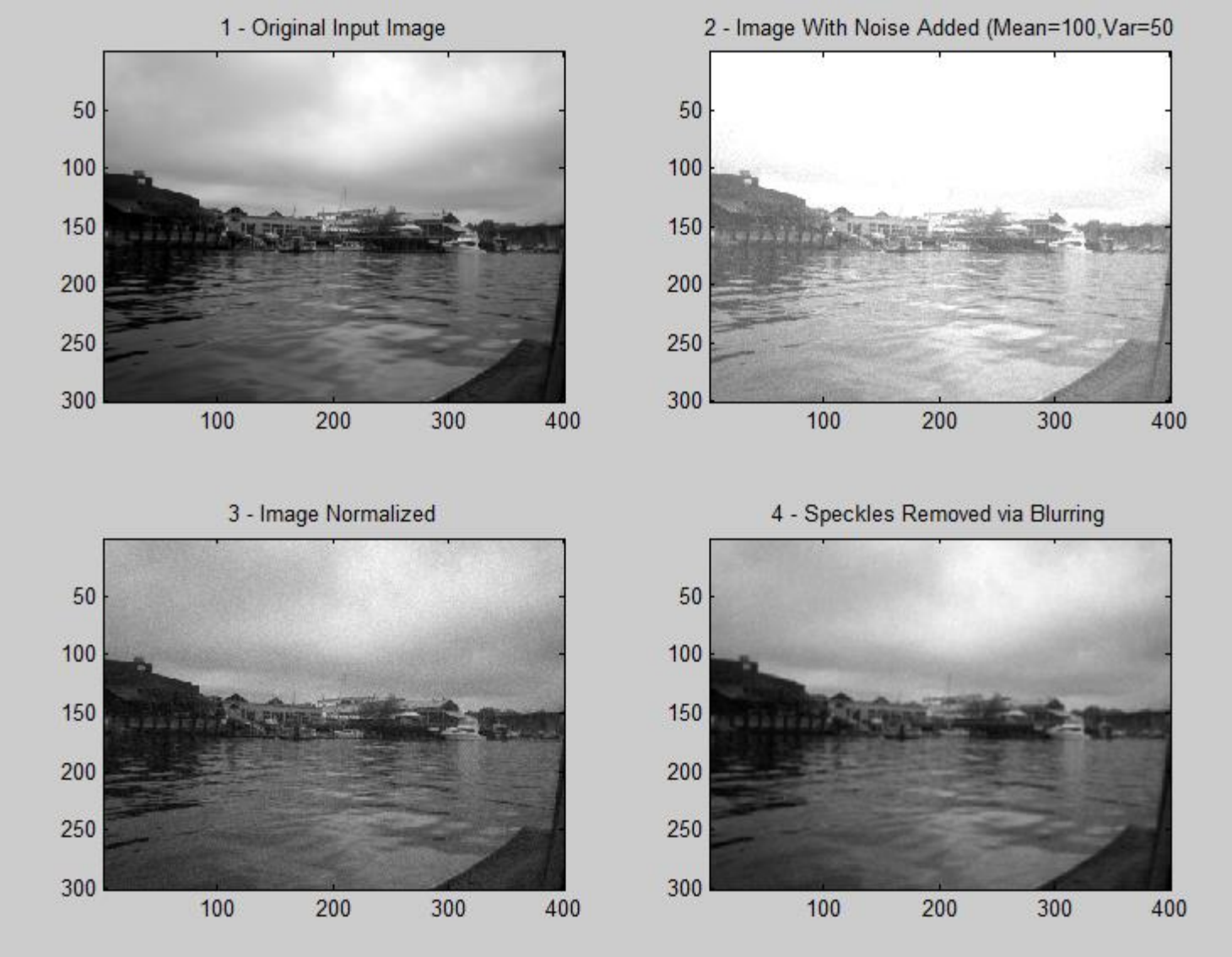
Choosing the wrong value for Q when using the contraharmonic filter can have drastic results



Filtering of Gaussian noise



Filtering of Gaussian noise (contd.)



Rank Filters (non linear filtering)

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

Useful spatial filters include

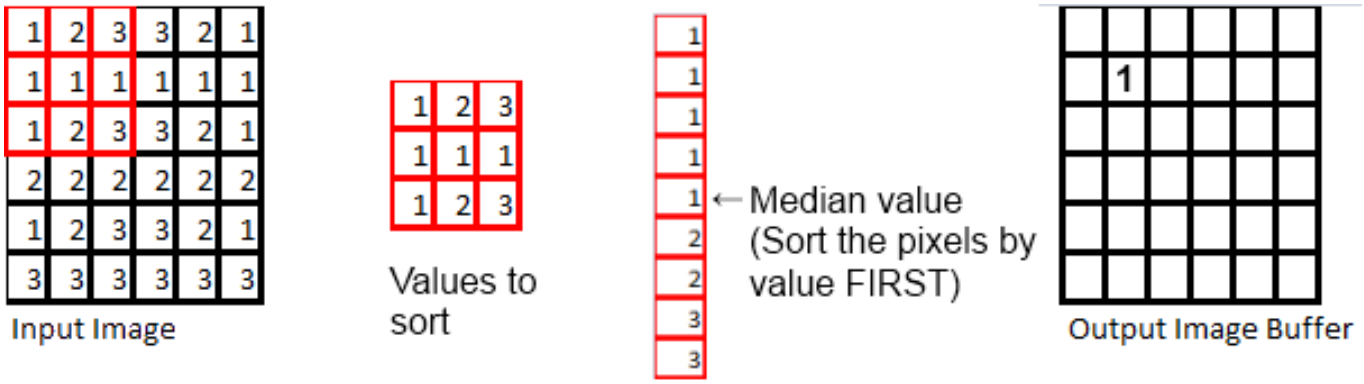
- Median filter
- Max and min filter
- Midpoint filter
- Alpha trimmed mean filter

Median Filter

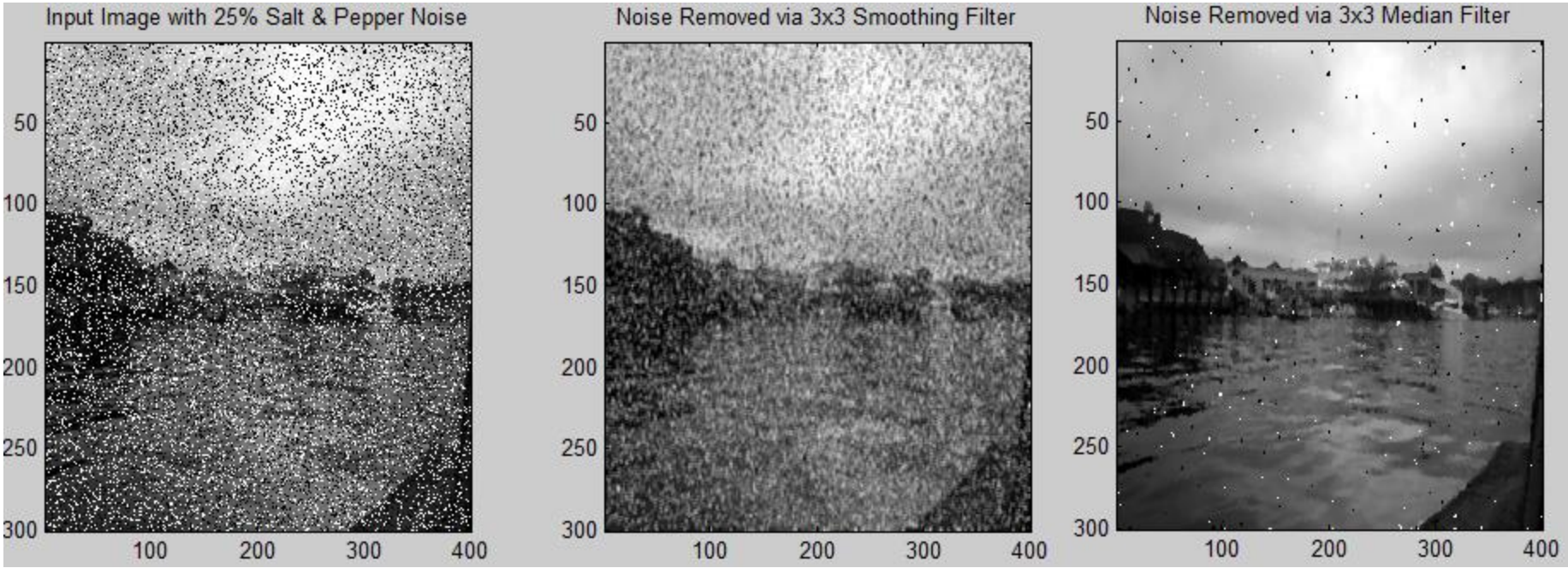
Median Filter:

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

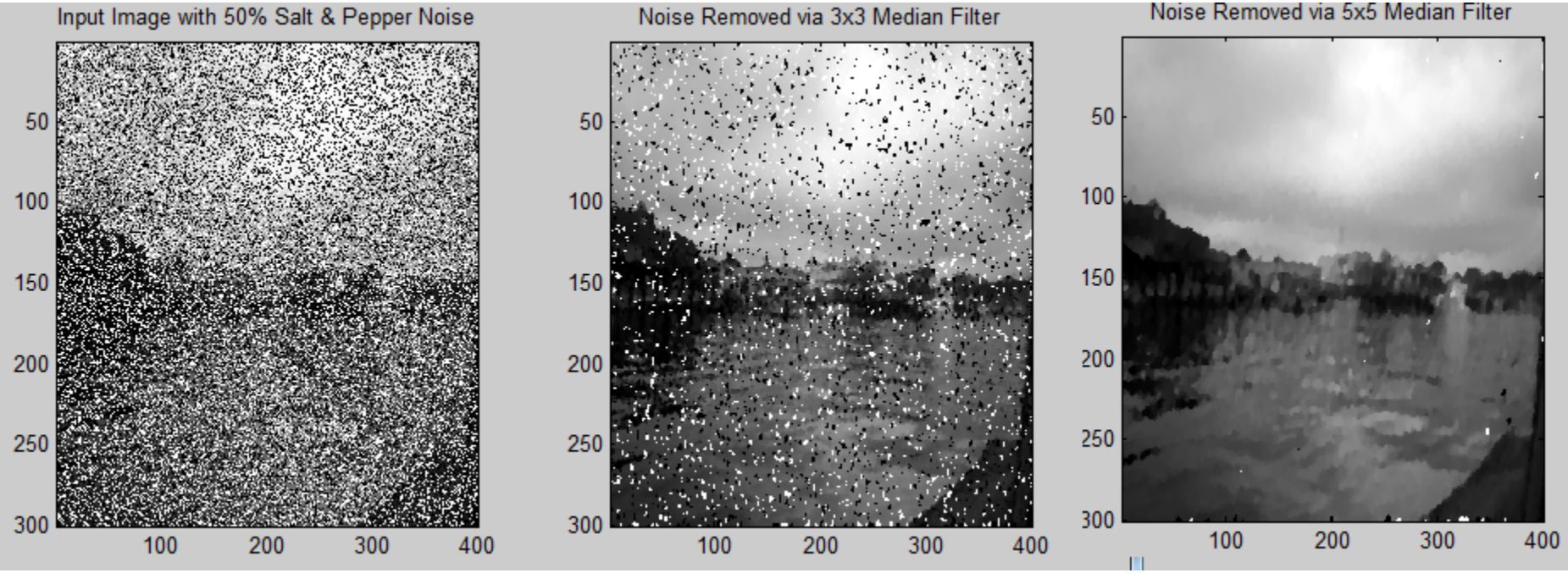
- Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters
- Particularly good when salt and pepper noise is present



Mean Vs. Median filter

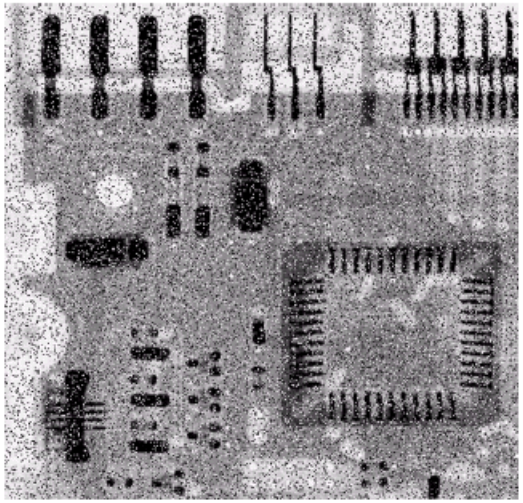


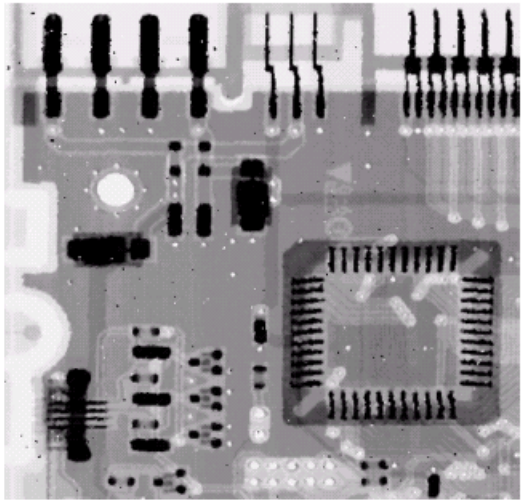
Noise Removal Examples



Noise Removal Examples (contd)

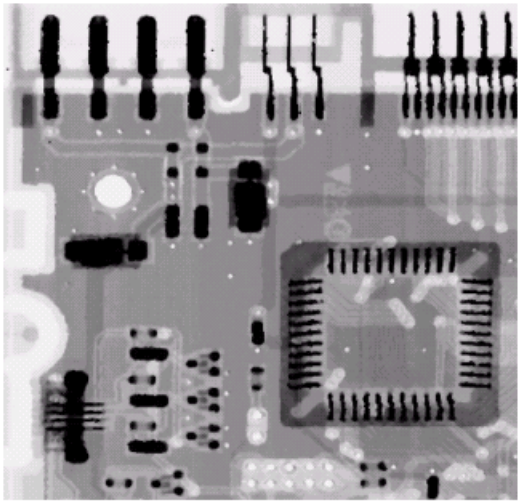
Image Corrupted By Salt And Pepper Noise

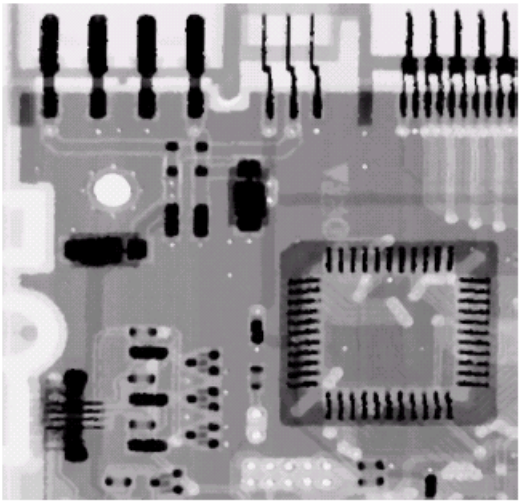




Result of 1 Pass With A 3*3 Median Filter

Result of 2 Passes With A 3*3 Median Filter





Result of 3 Passes With A 3*3 Median Filter

Max and Min Filter

Max Filter:

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

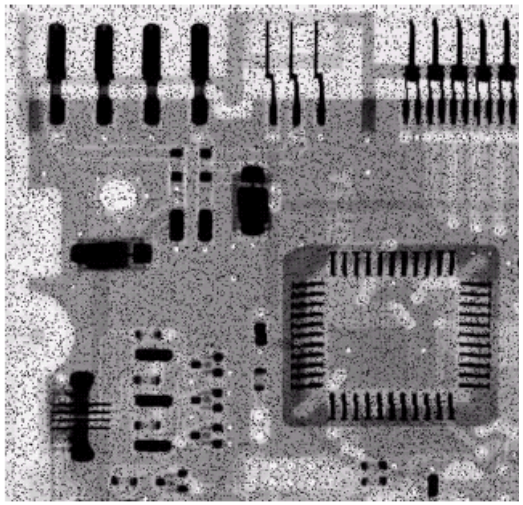
Min Filter:

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

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

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



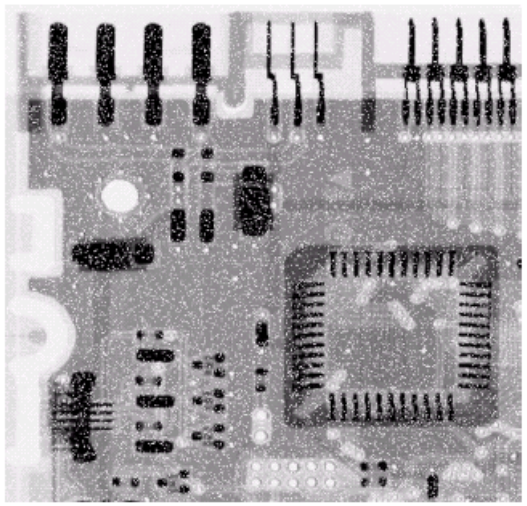
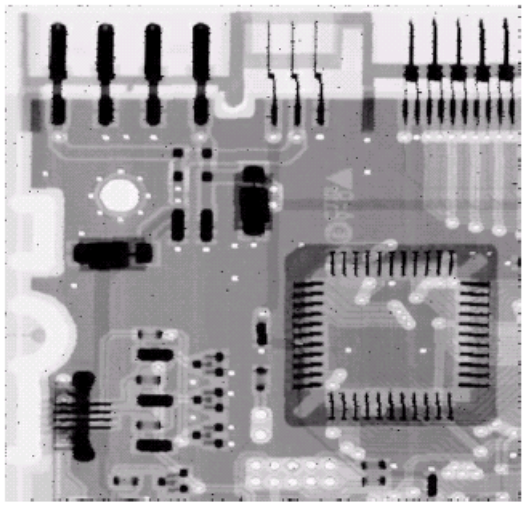
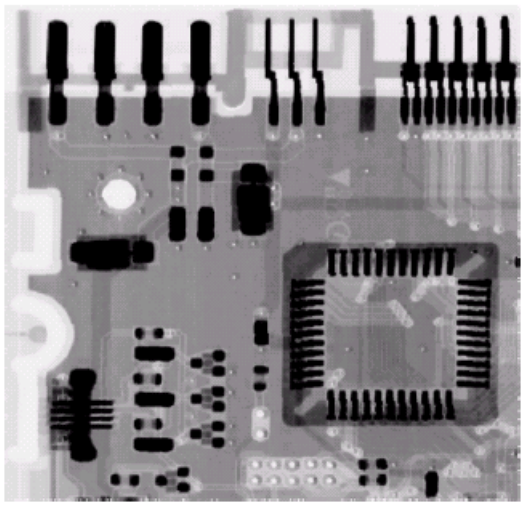


Image Corrupted By Salt Noise

Result Of Filtering Above With A 3*3 Max Filter





Result Of Filtering Above With A 3*3 Min Filter

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

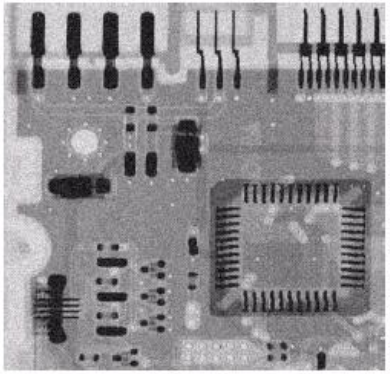
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

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

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

Noise Removal Examples (cont…)

Image Corrupted By Uniform Noise



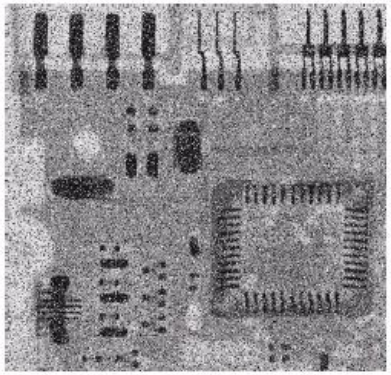
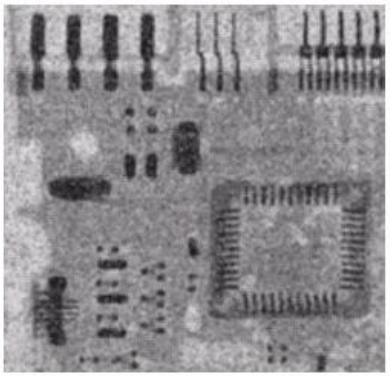
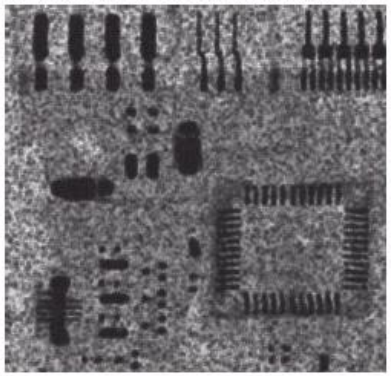


Image Further Corrupted By Salt and Pepper Noise

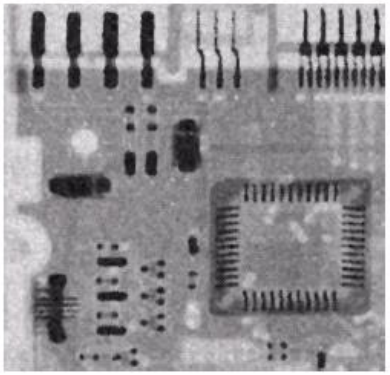
Filtered By 5*5 Arithmetic Mean Filter

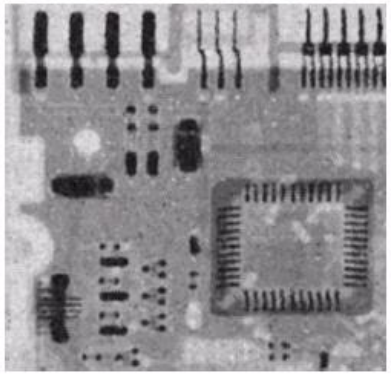




Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Median Filter





Filtered By 5*5 Alpha-Trimmed Mean Filter

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

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

Adaptive filter: Neighborhood-based

Adaptive local noise reduction filter

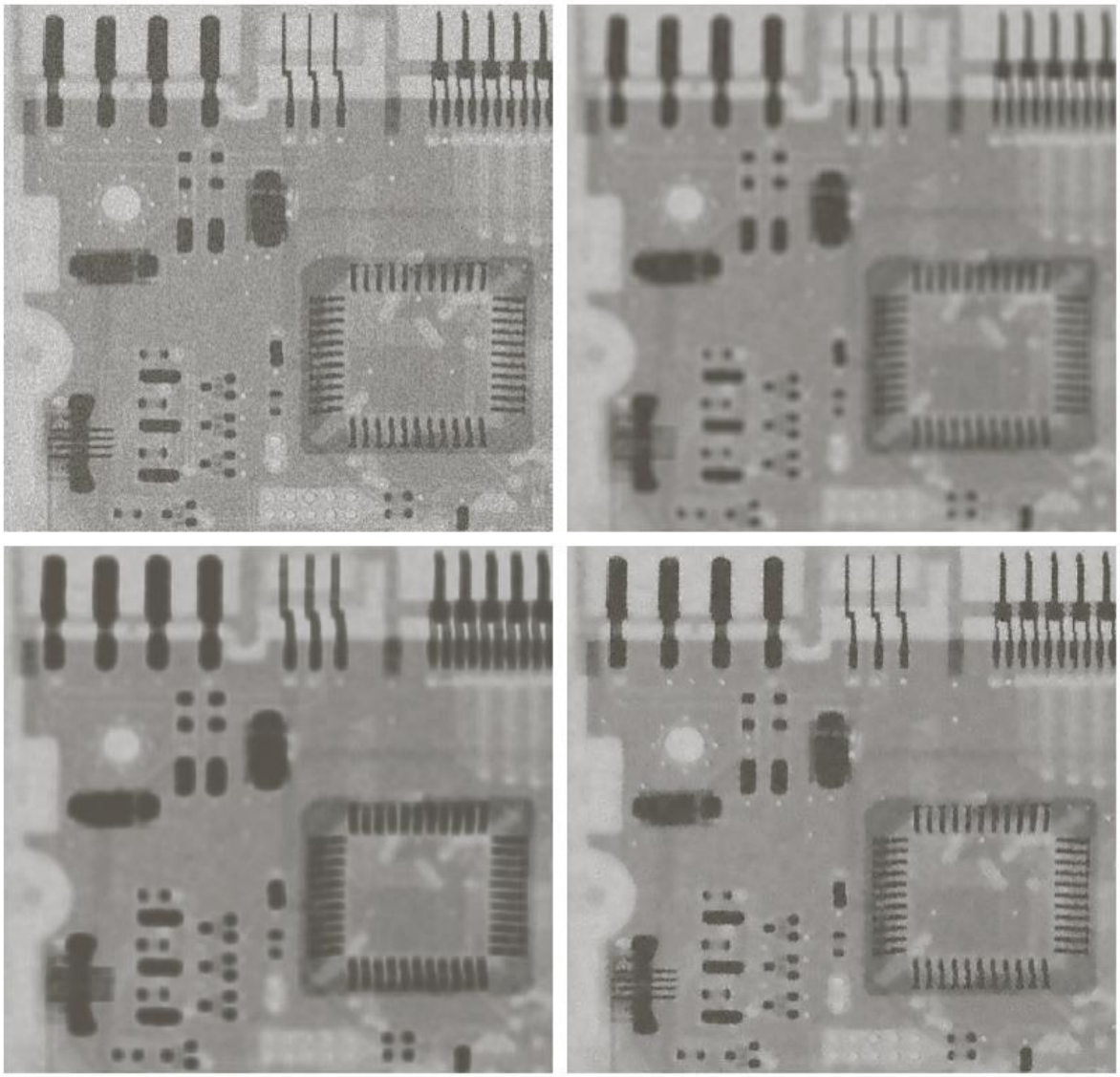
- Response based on 4 quanties
  - Local variance, variance of noise, g(x,y), and local mean
- Behavior of filter
  - If variance of noise is zero, return g(x,y)
  - If the local variance is high compared to the variance of noise, return a value close to g(x,y)
  - If the two variances are equal, return the mean value

$$\hat{f}(x,y) = g(x,y) - \frac{s_h^2}{s_L^2}[g(x,y) - m_L]$$

Adaptive filter: Neighborhood-based

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

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 does less distortion

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…)

The adaptive median filter has following purposes:

- Remove spatially dense impulse noise
- Reduce distortion

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}

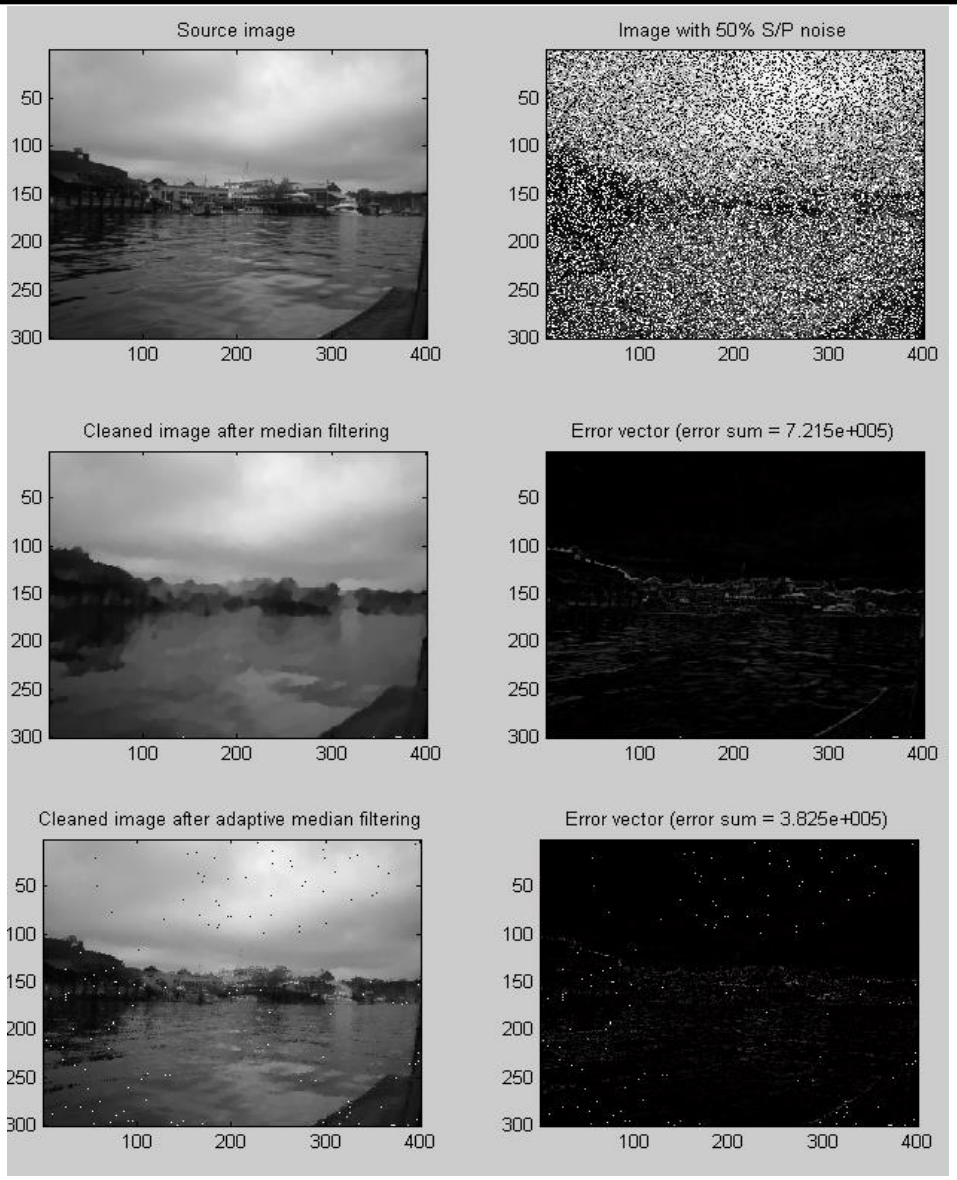
Level B: $B1 = z_{xy} - z_{min}$

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

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

Else output z_{med}

Simple Adaptive Median Filtering Example



Using only level B and filter 7 x 7

Adaptive Median Filtering Examples

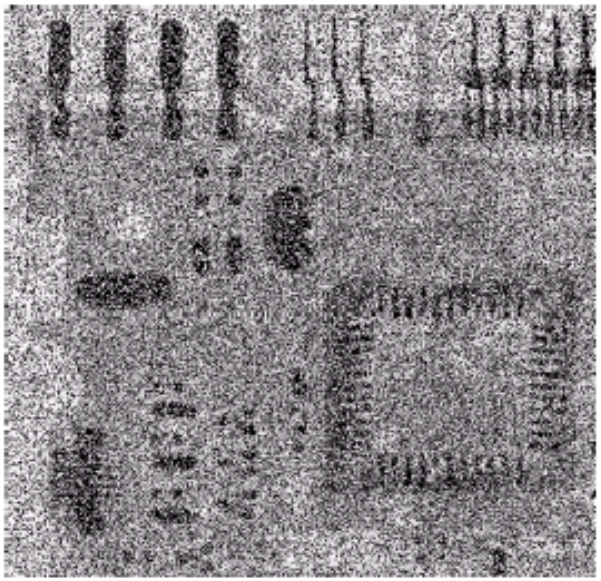
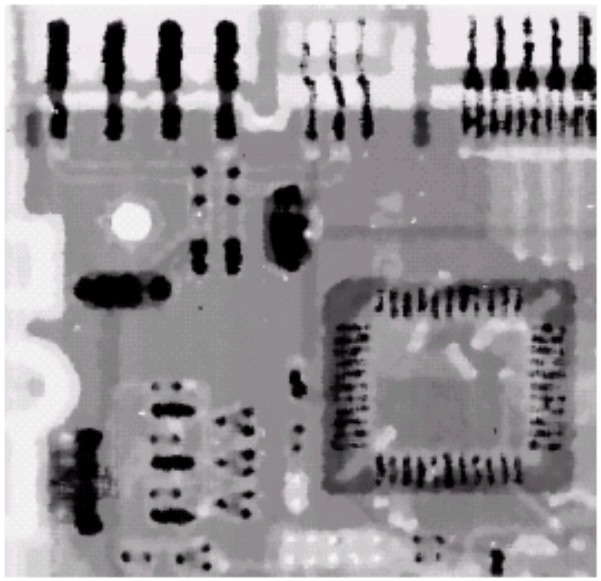
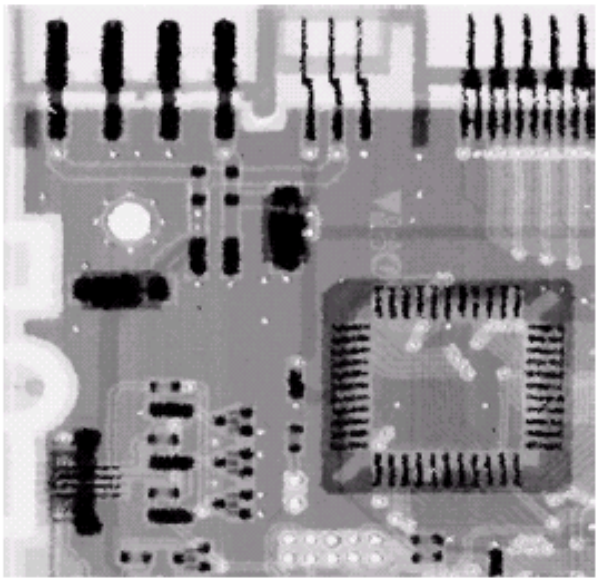


Image corrupted by salt and pepper noise with probabilities $P_a = P_b$=0.25



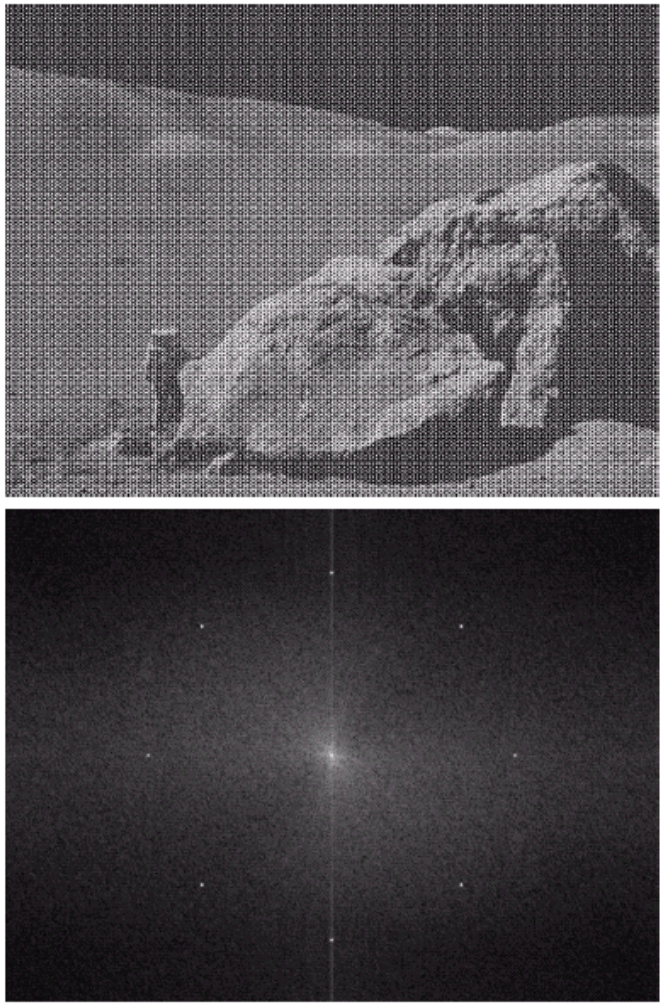
Result of filtering with a 7 * 7 median filter



Result of adaptive median filtering with S_{max} = 7

Periodic Noise removal

- 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 from an image involves removing a particular range of frequencies from that image

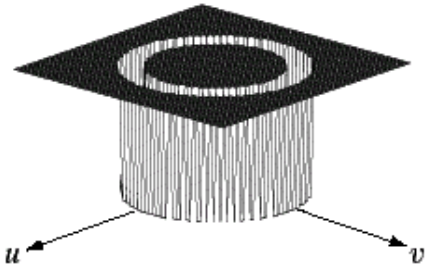
Band reject filters can be used for this purpose

An ideal band reject filter is given as follows:

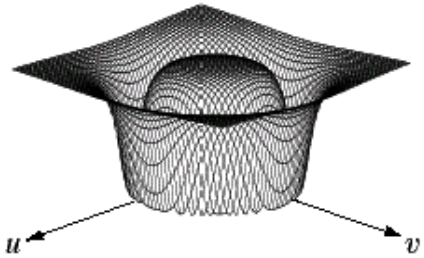
$$H(u,v) = \begin{cases} 1 & if\ D(u,v) < D_0 - \frac{W}{2} \\ 0 & if\ D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & if\ D(u,v) > D_0 + \frac{W}{2} \end{cases}$$

Band Reject Filters (cont…)

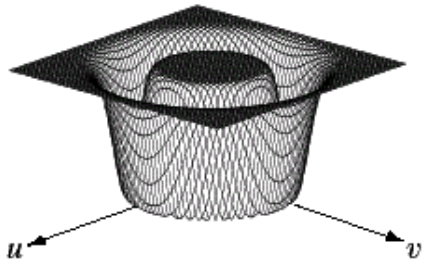
The ideal band reject filter is shown below, along with Butterworth and Gaussian versions



Ideal Band Reject Filter



Butterworth Band Reject Filter (of order 1)

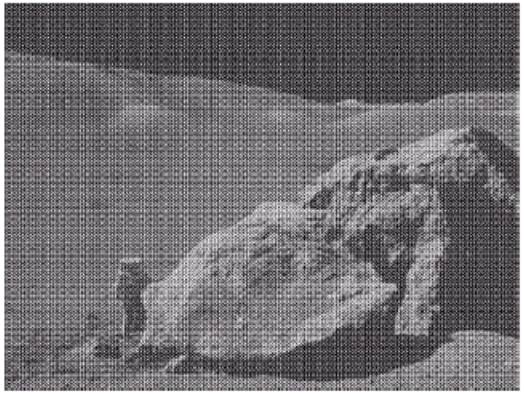


Gaussian Band Reject Filter

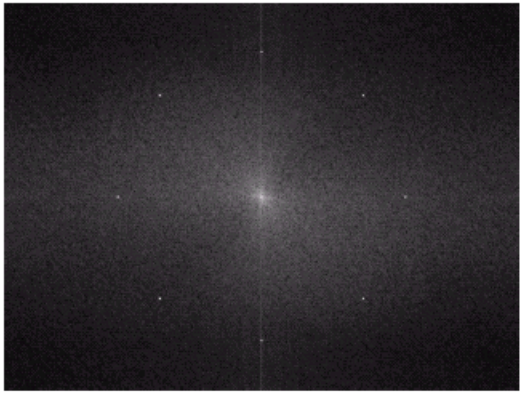
Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \frac{1}{1 + \left[\frac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

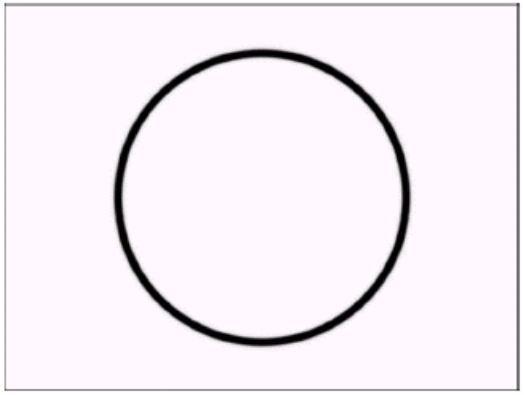
Band Reject Filter Example

Image corrupted by sinusoidal noise



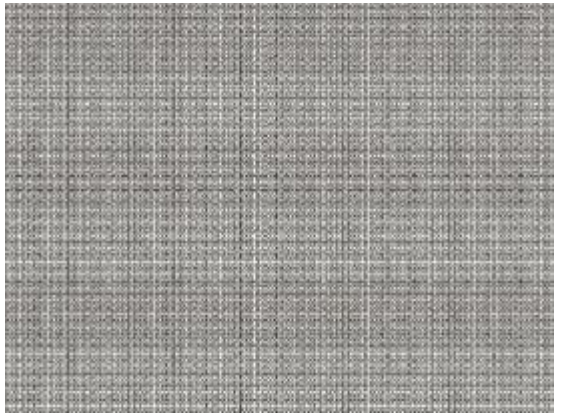
Fourier spectrum of corrupted image



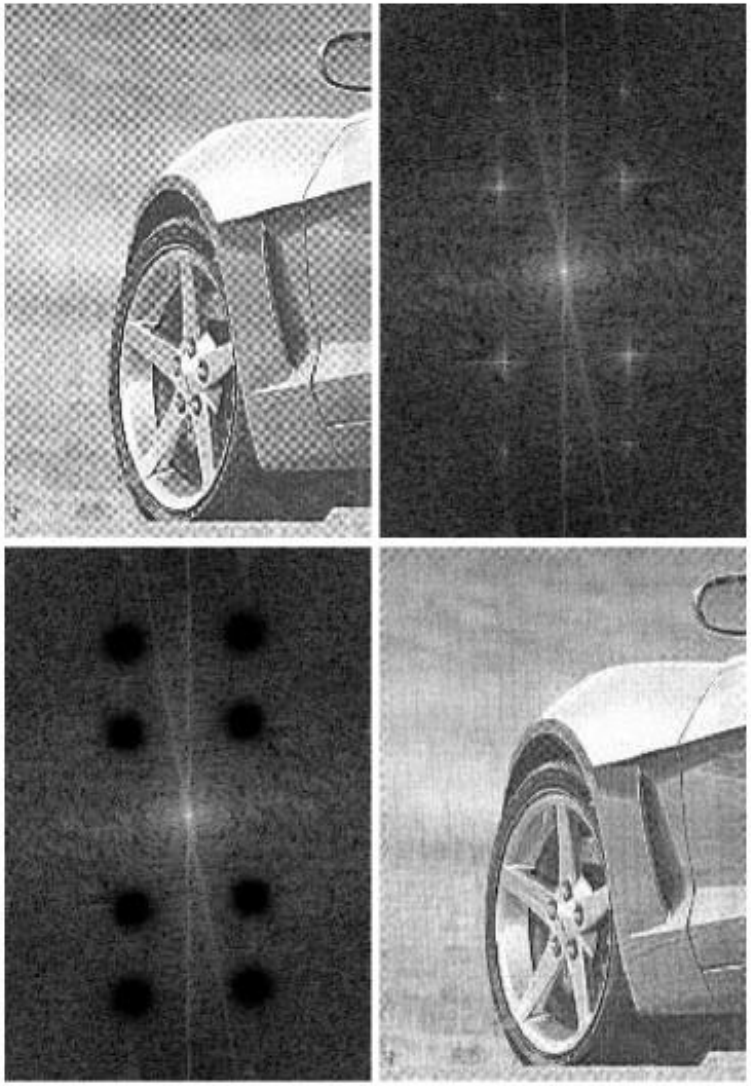


Butterworth band reject filter

Filtered image



Notch Filter



a b
c d

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

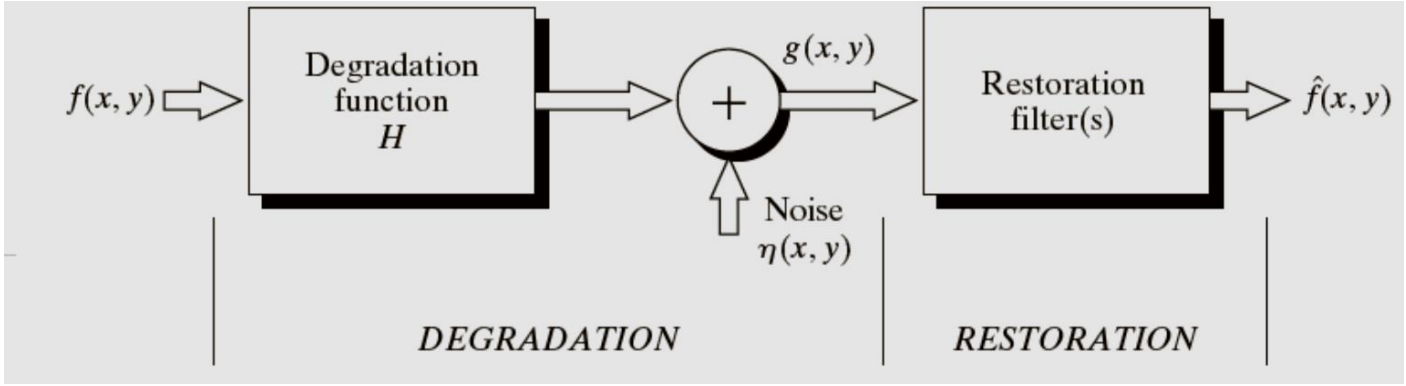
Restoration From Degradation

Linear, position-invariant degradation model

- Many types of degradation can be *approximated* by linear, position-invariant processes.

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

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



- Image deconvolution: find H(u,v) and apply inverse process

Estimating the Degradation Function

- By image observation: Take a region in the image with
  - Simple structure
  - Strong signal content (negligible noise)
- Estimate the original image in the window

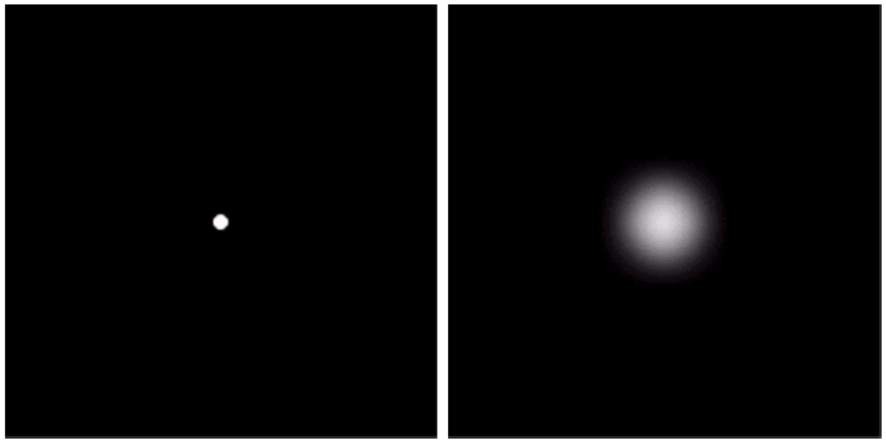
$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

known

estimate

Estimating the Degradation Function

- By experimentation: If the image acquisition system is available, then obtain impulse response
  - G(u,v) = H(u,v)F(u,v) + N(u,v)
  - H(u,v) = G(u,v) / A



a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

impulse

Impulse response

Estimating the Degradation Function

- By modeling: Ex. Atmospheric turbulence

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

original

k=0.0025

k=0.001

k=0.00025

Estimating the Degradation Function

- Derive a mathematical model: ex. Motion blur

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

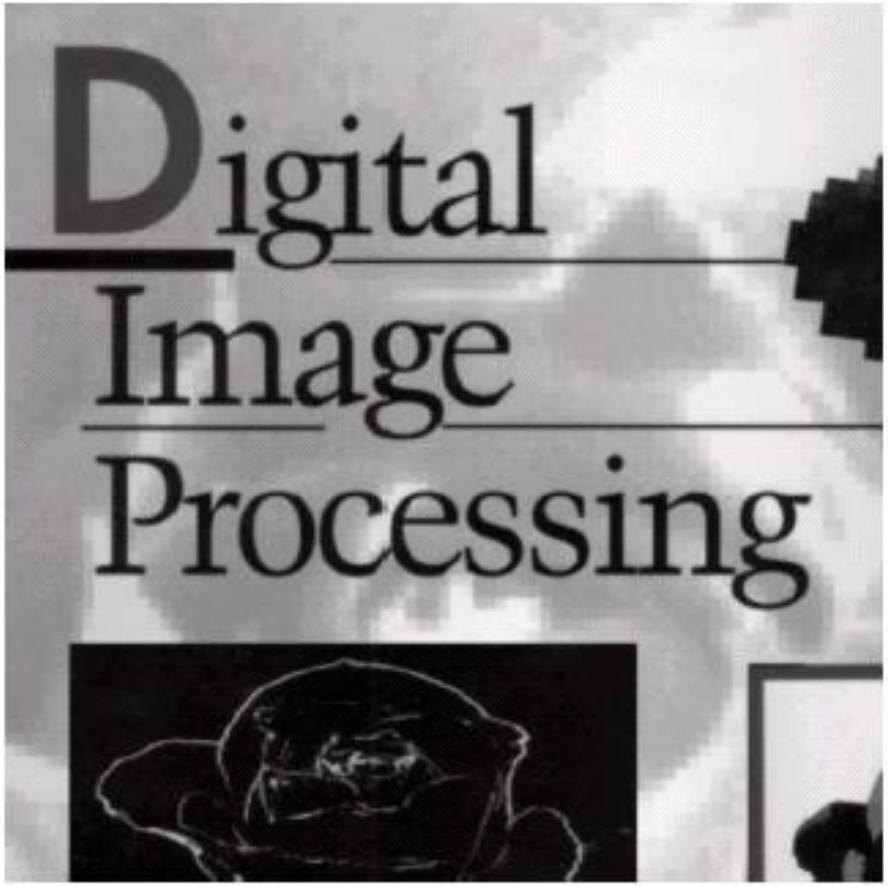
Planar motion

Fourier transform

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

Estimating the Degradation Function

original



Apply motion model

Inverse Filtering

- Compute an estimate by simply dividing the transform of degraded image by the degradation function

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

Unknown noise

Estimate of original image

Problem: 0 or small values

- Suffers from noise amplification

Psuedo inverse filtering

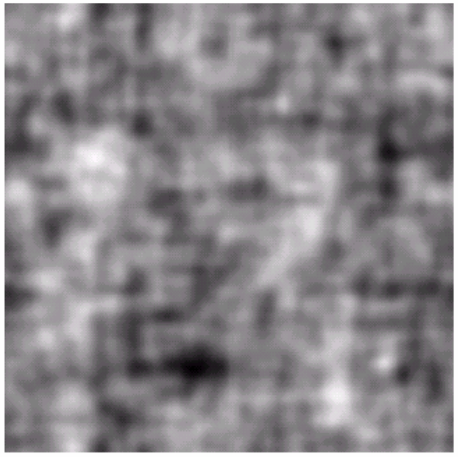
- Limiting the analysis to frequencies near origin

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

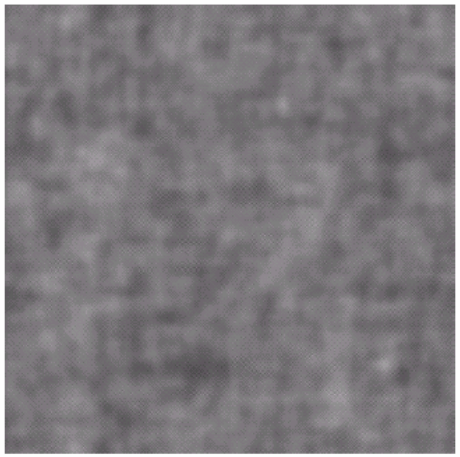
$$\hat{H}(u,v) = \begin{cases} 1/H(u,v), & |u^2+v^2| \le \eta \\ 0, & |u^2+v^2| > \eta \end{cases}$$

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

Full inverse filter for k=0.0025



Cut Outside 70%



Cut Outside 85%

Degraded Image

Wiener Filtering

- Incorporates both the degradation and statistical characteristics of noise
- Objective function: find an estimate of *f* such that the mean square error between them is minimized

$$e^2 = E\{(f - \hat{f})^2\}$$

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

$$S_f(u,v) = \left|F(u,v)\right|^2 = \text{power spectrum of the undegraded image}$$

$$S_\eta(u,v) = \left|N(u,v)\right|^2 = \text{power spectrum of the noise}$$

Wiener Filtering

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

Constant

Unknown

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

Inverse vs. Wiener Filtering

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

Full inverse filtering

Radially limited inverse filtering

Wiener filtering

(K was chosen interactively)

Further comparison

Reduced Variance

a b c
d e f
g h i

FIGURE 5.29 (a) 8-bit image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

Quantative evaluation of image restoration

- Some Performance Measures for evaluating Image restoration method where the original image and reconstructed image from its degraded version is available are as follows:
  - Signal-to-Noise Ratio (SNR)
  - Mean Square Error (MSE)
  - Root Mean Square Error (RMSE)
  - Peak Signal-to-Noise Ratio (PSNR)

Contd..

- Signal to noise ratio

$$SNR = \frac{\sum_{m}\sum_{n} |F(x,y)|^2}{\sum_{m}\sum_{n} |N(x,y)|^2}$$

- ***Mean square error*:**

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left[I'(i,j) - I(i,j) \right]^2$$

Where *I* is the original image and I' is the reconstructed image.

- ***Root mean square error*:**

$$RMSE = \sqrt{MSE}$$

- ***Peak signal-to-noise ratio*:**

$$PSNR = 20 \log_{10} \left[\frac{255}{RMSE} \right]$$