



Digital Image Processing

Lecture #10
Ming-Sui (Amy) Lee

Course Information

■ Following Schedule

04/01	Lecture 5	05/20	Lecture 10
04/08	Lecture 6	05/27	Lecture 11
04/15	Lecture 7	06/03	Lecture 12
04/22	Midterm	06/10	Demo
04/29	Lecture 8	06/17	Demo
05/06	Proposal	06/24	Final Package Due
05/13	Lecture 9		

■ Homework 4

- Due: **11:59 pm, May 27, 2021**

Image Restoration

Image Restoration

- Attempt to reconstruct or recover an image that has been degraded
- Model the degradation and apply the inverse process



- A priori modeling
 - measurements on the physical imaging system, digitizer and display
- A posteriori modeling
 - measurements of a particular image to be restored

Image Restoration



- **Forward Problem**
 - Given X & $H \rightarrow$ find Y
 - **System Identification Problem**
 - Given X & $Y \rightarrow$ find H
 - **Inverse Problem**
 - Given H & $Y \rightarrow$ find X
- **Image enhancement v.s. image restoration**

Image Restoration

- Image Enhancement v.s. Image Restoration
 - Image enhancement
 - Largely subjective
 - No benchmark
 - Basically a heuristic procedure
 - Image restoration
 - More objective
 - With a ground truth
 - Usually formulate a criterion of goodness

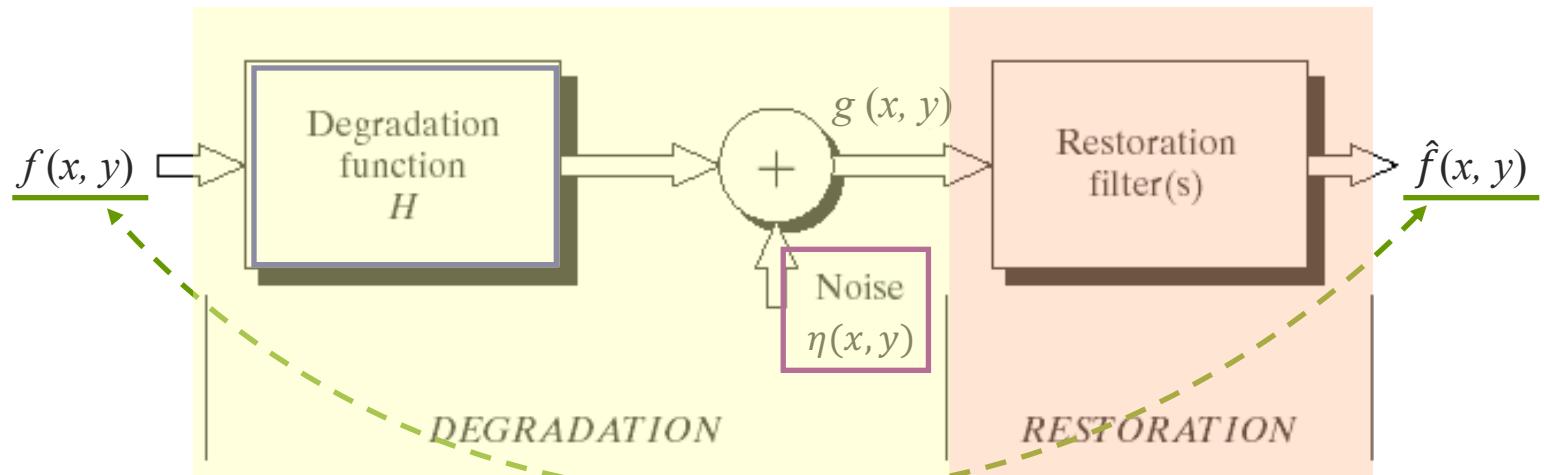
[Image Restoration]

■ Image observation models

- An acquired image could be degraded by the sensing environment
 - Sensor noise
 - Optical system aberrations
 - Image motion blur
 - Atmospheric turbulence effects
- Model the degradation and apply the inverse process
 - A priori modeling
 - A posteriori modeling

[Image Restoration]

■ Image observation models



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

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

[Image Restoration]

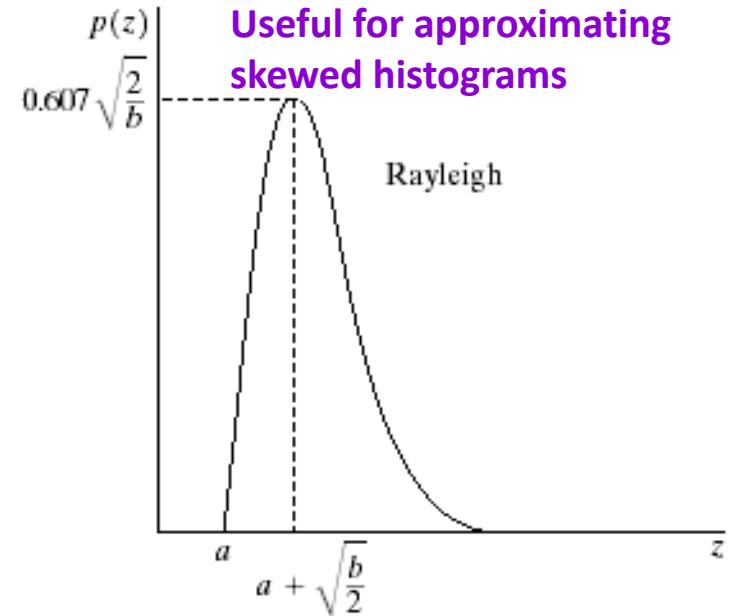
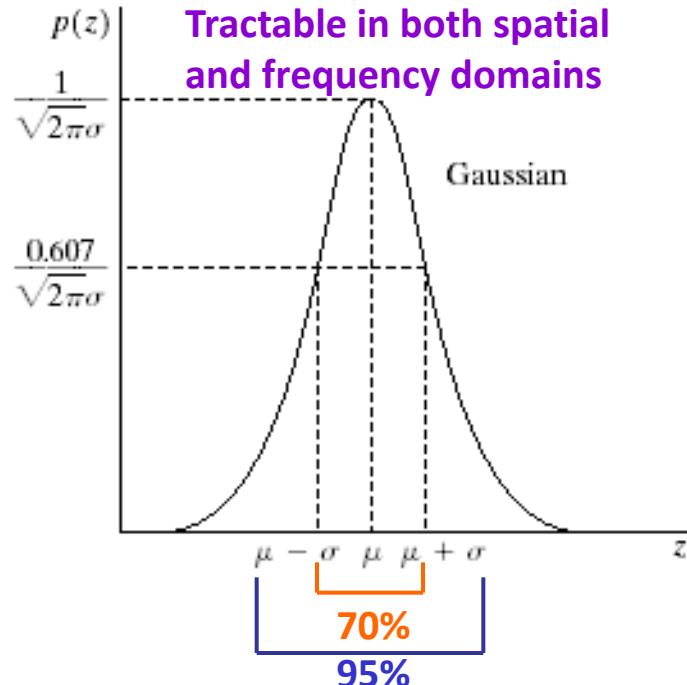
■ Noise models

- Gaussian Noise
- Rayleigh Noise
- Gamma (Erlang) Noise
- Exponential Noise
- Uniform Noise
- Impulse (Salt-and-Pepper) Noise

//Note// assume that noise is independent of spatial coordinates
and uncorrelated w.r.t. the image itself (except periodic noise)

[Image Restoration]

■ Gaussian noise v.s. Rayleigh noise

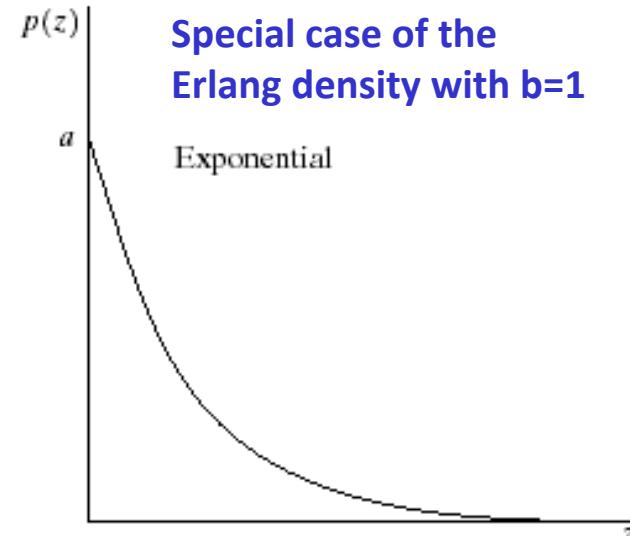
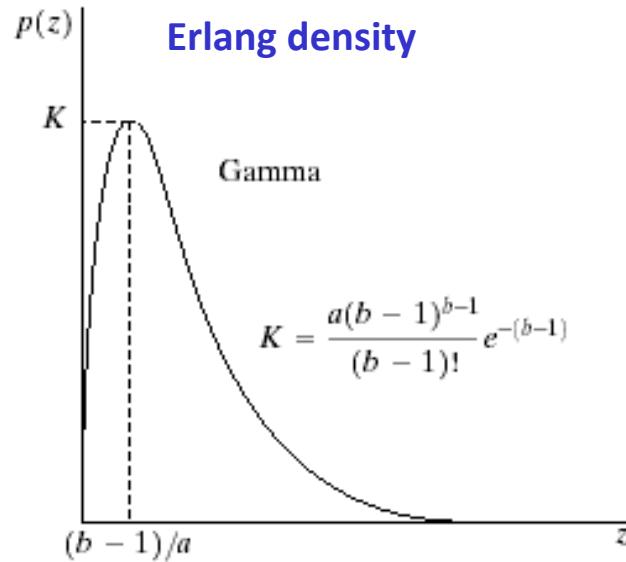


$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}^{10}$$

Image Restoration

■ Gamma noise v.s. Exponential noise

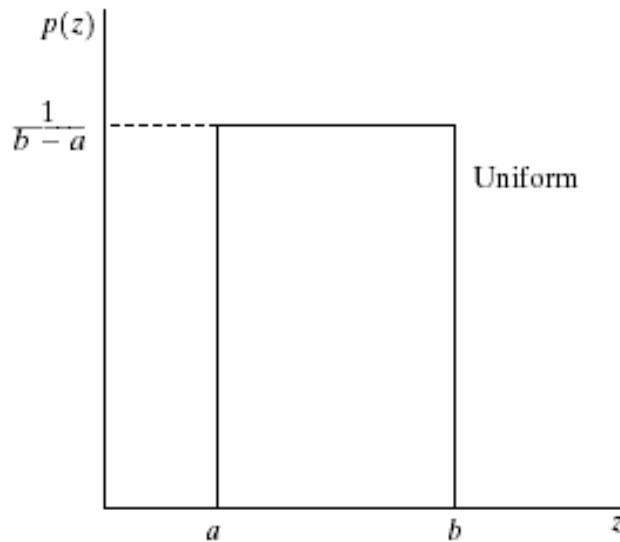


$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

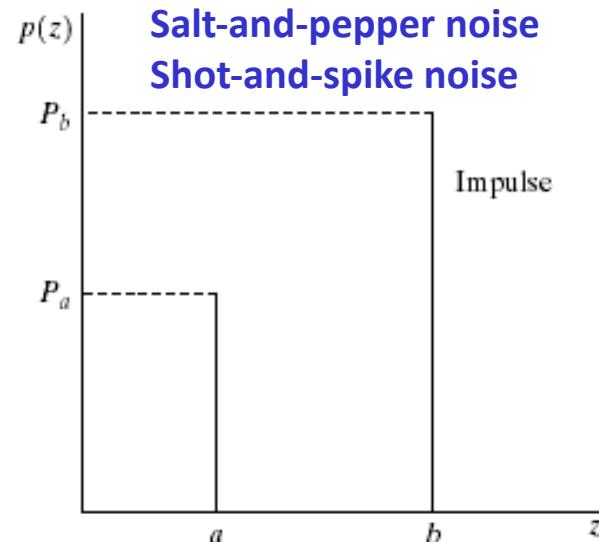
$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

Image Restoration

■ Uniform noise v.s. Impulse noise



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



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

Image Restoration

■ Noise models

- Gaussian Noise
 - Electronic noise and sensor noise due to poor illumination and/or high temperature
- Rayleigh Noise
 - Helpful in characterizing noise phenomena in range imaging
- Gamma (Erlang) Noise & Exponential Noise
 - Applications in laser imaging
- Uniform Noise
 - As basis for random number generators
- Impulse (Salt-and-Pepper) Noise
 - Quick transient (faulty switching) during imaging

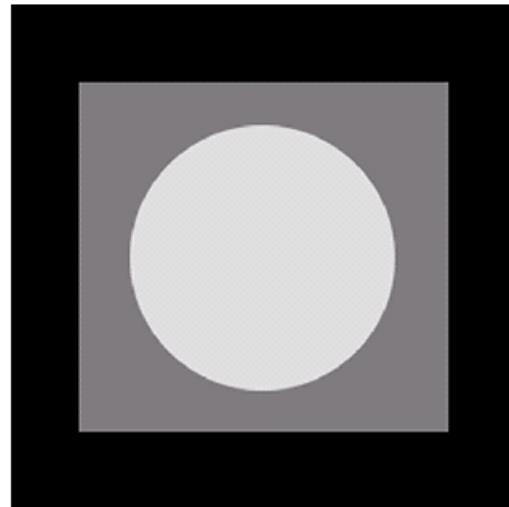
[

Image Restoration

]

- Test pattern

- Three-leveled gray-scale image



[Image Restoration]

■ Experimental Results

image

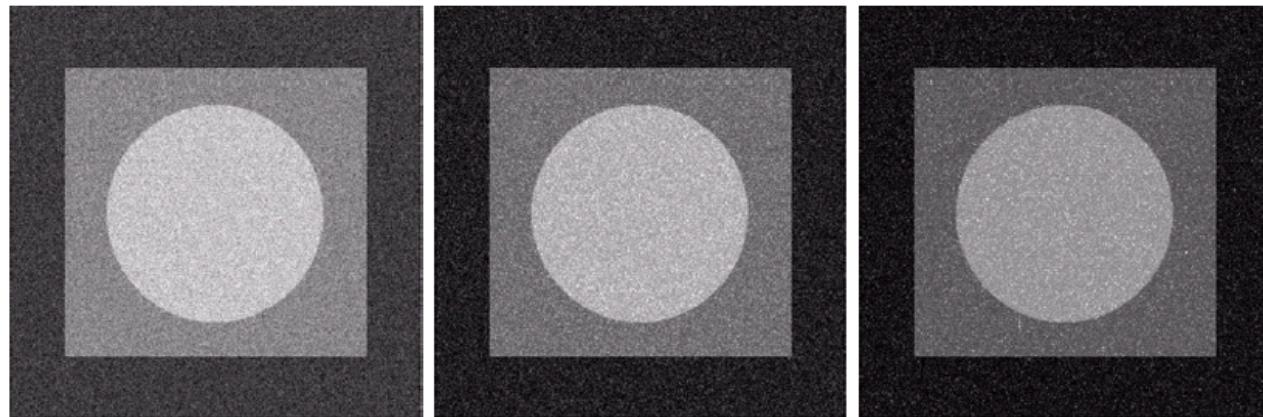
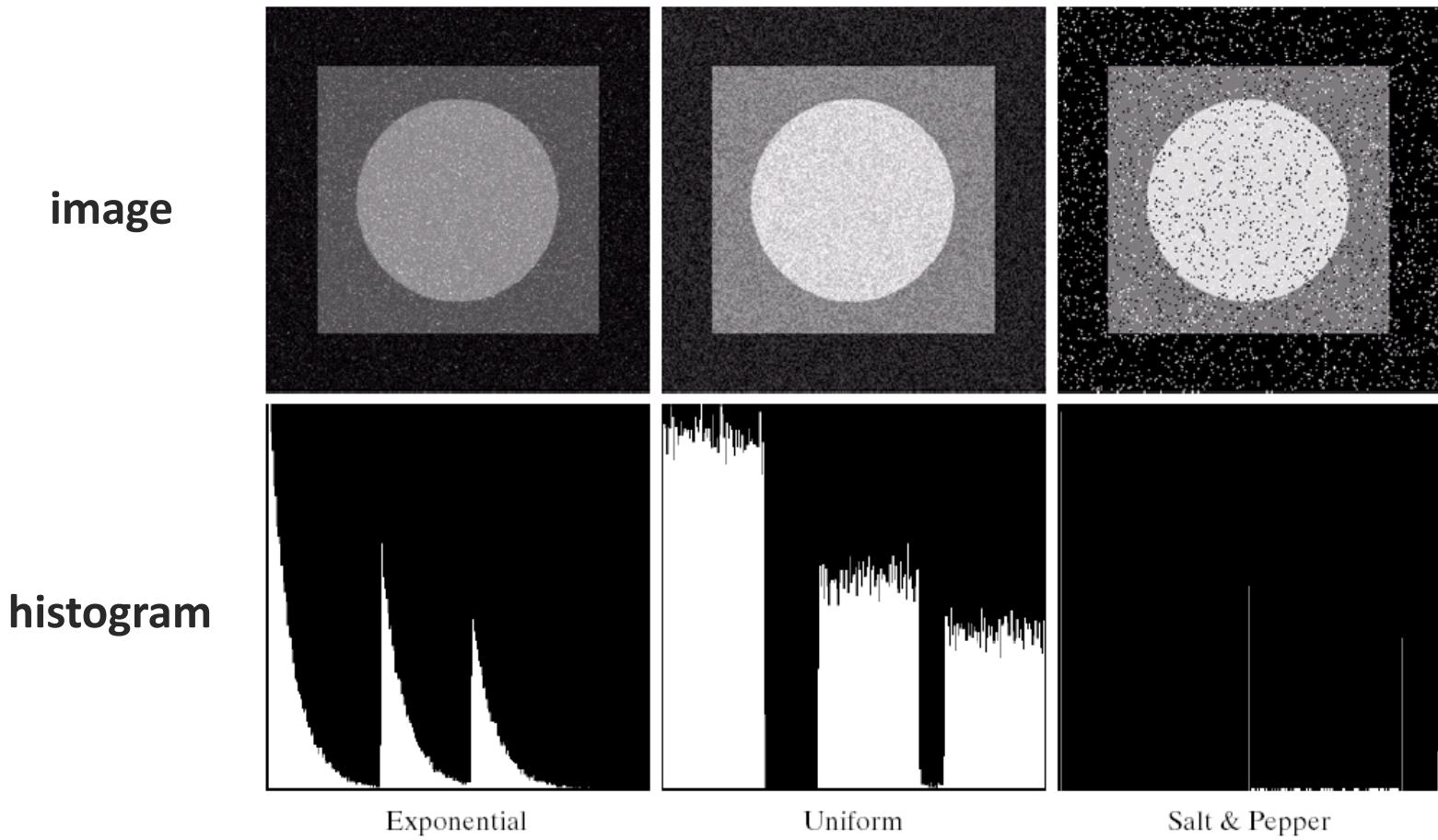


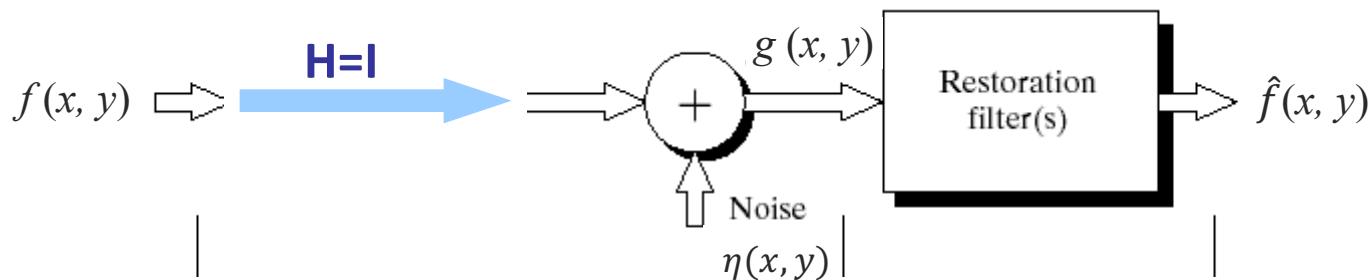
Image Restoration

■ Experimental Results



[Image Restoration]

■ Restoration with only additive noise



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

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

■ Spatial filtering: three solutions

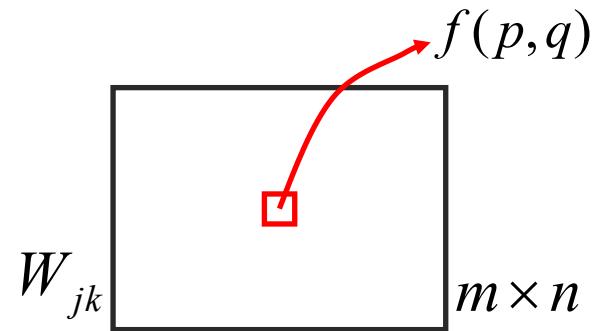
- Mean filters
- Order-statistics filters
- Adaptive filters

[Image Restoration]

■ Mean filters

○ Arithmetic mean filter

$$g(j,k) = \frac{1}{mn} \sum_{(p,q) \in W_{jk}} f(p,q)$$



Smooth local variations as a result of blurring

○ Geometric mean filter

$$g(j,k) = \left[\prod_{(p,q) \in W_{jk}} f(p,q) \right]^{\frac{1}{mn}}$$

Tend to lose less image detail in the process

Image Restoration

Example

- Arithmetic mean filter v.s. Geometric mean filter

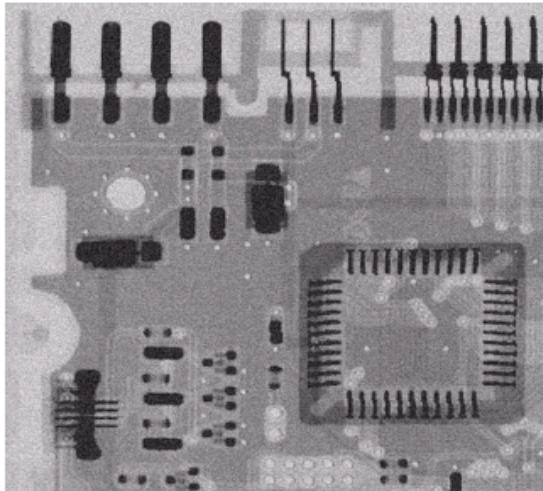
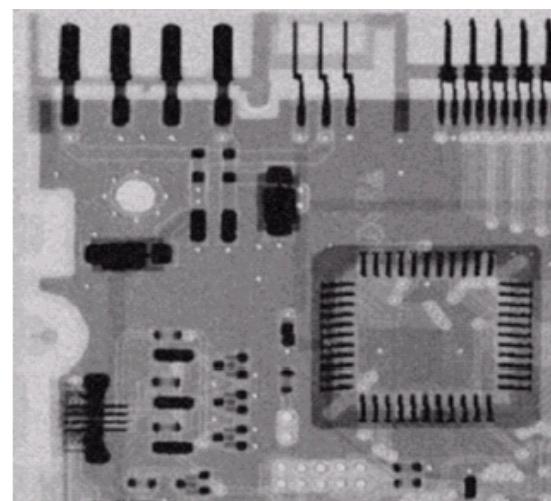
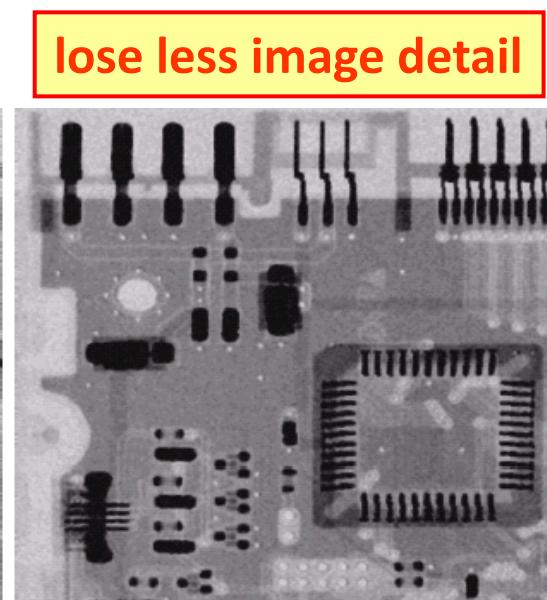


Image corrupted by
additive Gaussian noise



Arithmetic mean filter
of size 3x3



Geometric mean filter
of size 3x3

lose less image detail

Image Restoration

■ Mean filters

○ Harmonic mean filter

$$g(j,k) = \frac{mn}{\sum_{(p,q) \in W_{jk}} \frac{1}{f(p,q)}}$$

→ Good for salt noise
→ Gaussian noise
→ Fails for pepper noise

○ Contraharmonic mean filter

$$g(j,k) = \frac{\sum_{(p,q) \in W_{jk}} f(p,q)^{Q+1}}{\sum_{(p,q) \in W_{jk}} f(p,q)^Q}$$

→ Q>0: pepper noise
Q<0: salt noise
→ Q=0? → Arithmetic
→ Q=-1? → Harmonic

[Image Restoration]

■ Example

- Contraharmonic mean filter with order=1.5
(eliminate pepper noise)

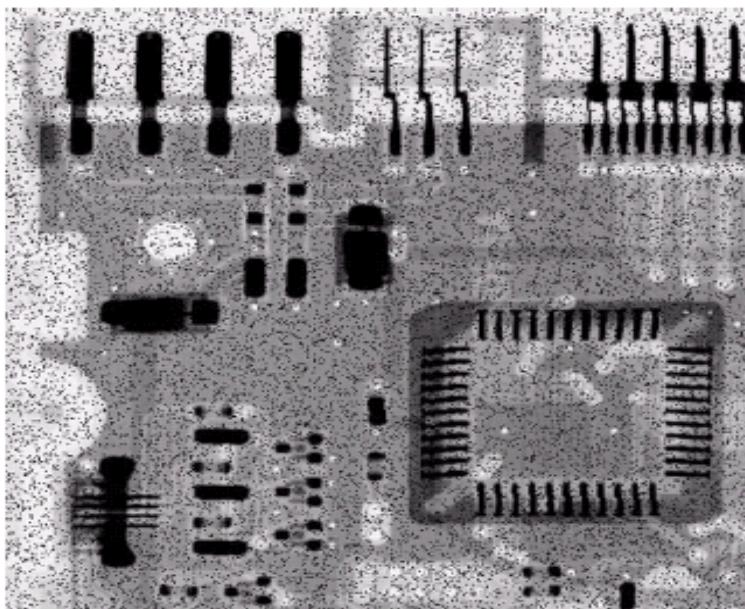
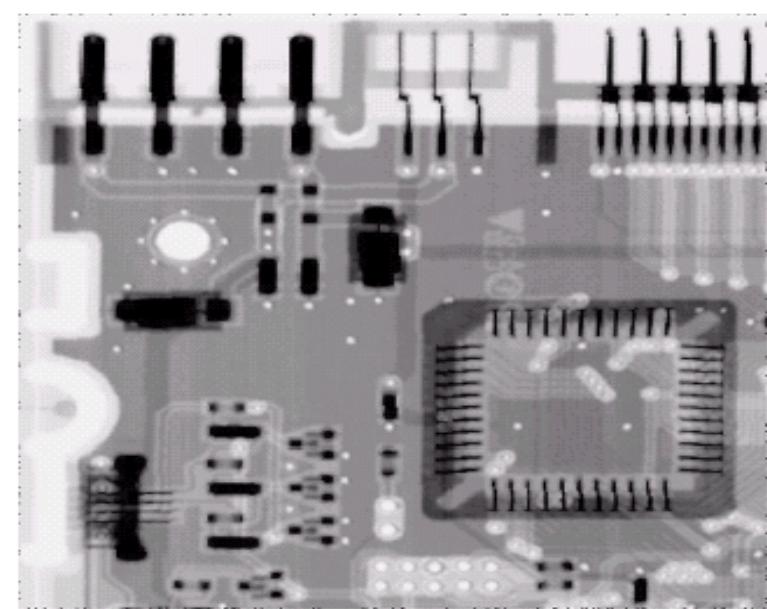


Image corrupted by
pepper noise



3x3 contraharmonic filter
of order 1.5

Image Restoration

Example

- Contraharmonic mean filter with order=-1.5
(eliminate salt noise)

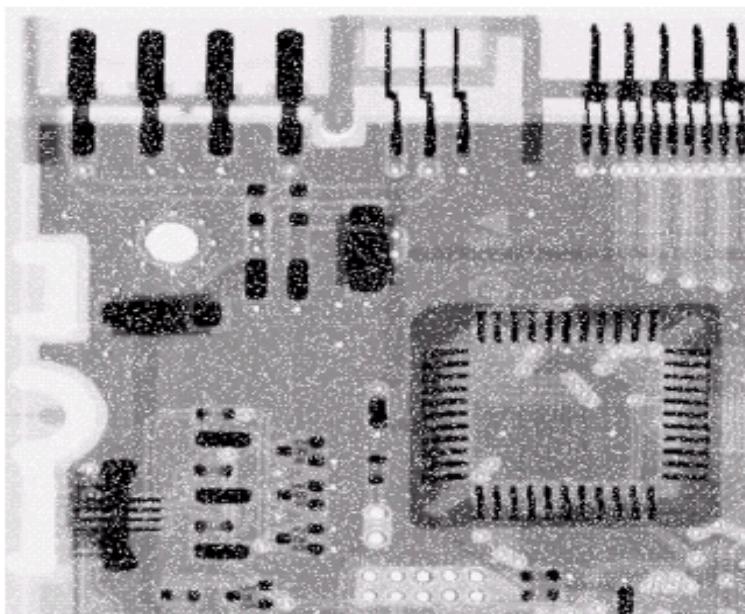
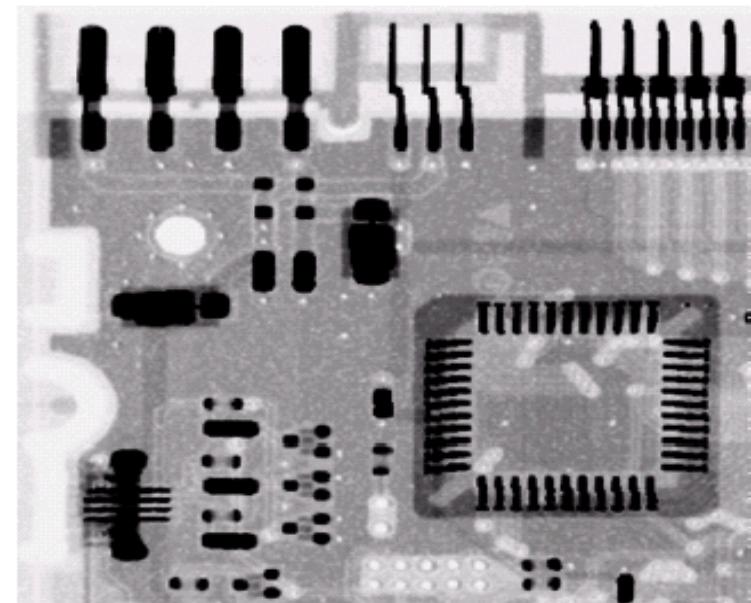


Image corrupted by
salt noise



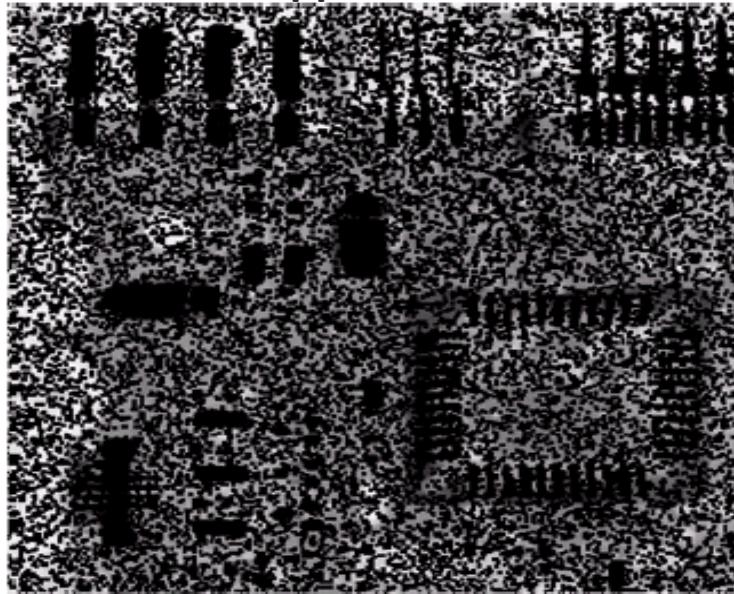
3x3 contraharmonic filter
of order -1.5

[Image Restoration]

■ Contraharmonic mean filters

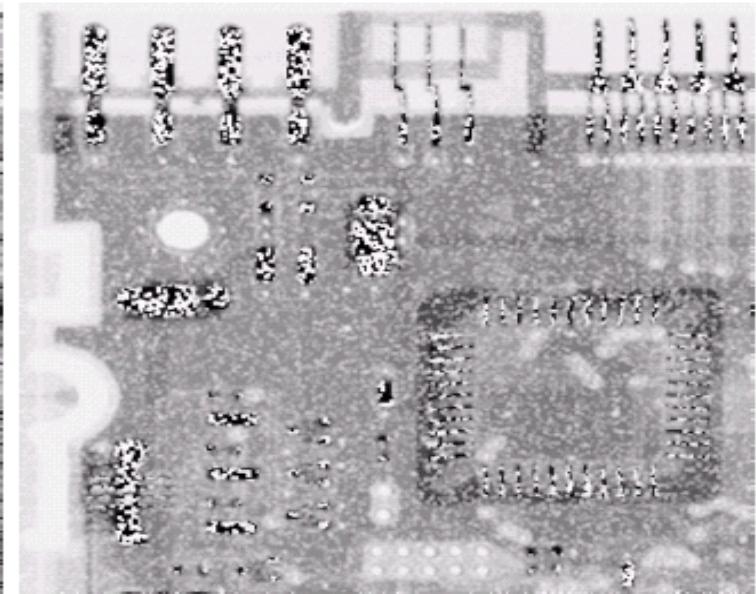
- Choose wrong sign of the filter

Pepper noise



3x3 contraharmonic filter
of order -1.5

Salt noise



3x3 contraharmonic filter
of order 1.5

[

Image Restoration

]

■ Order-statistics filters

- with response that is based on ordering the pixels contained in the image area encompassed by the filter
- Median filter

$$g(j,k) = \underset{(p,q) \in W_{jk}}{\text{median}} \{f(p,q)\}$$

- Effective in the presence of impulse noise
- Less blurring than smoothing filters

[Image Restoration]

Image Restoration

- Example
 - Median filter

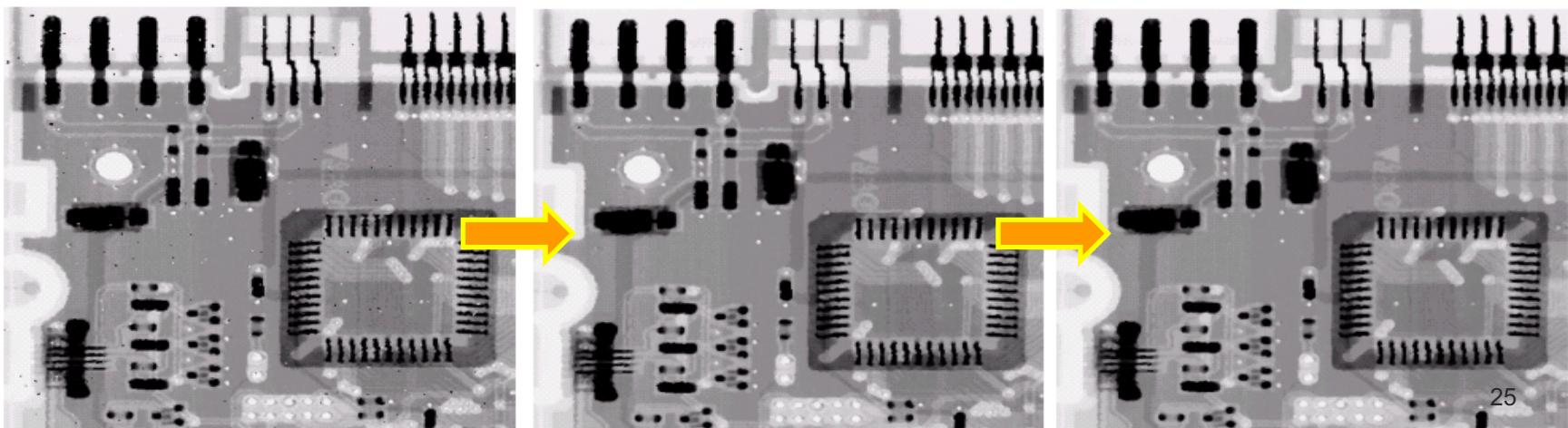
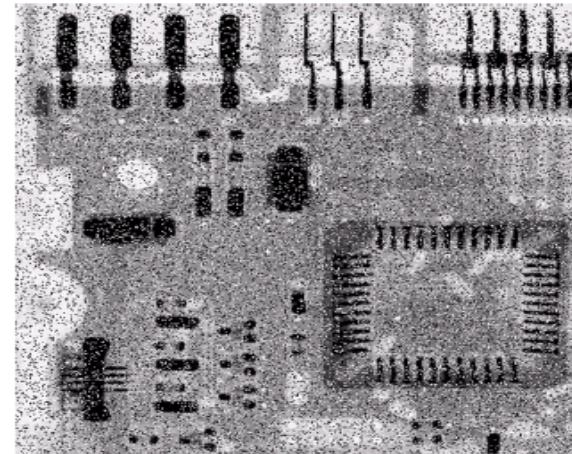


Image Restoration

■ Order-statistics filters

○ Max and min filters

$$g(j,k) = \max_{(p,q) \in W_{jk}} \{f(p,q)\} \rightarrow \text{Remove pepper noise}$$

$$g(j,k) = \min_{(p,q) \in W_{jk}} \{f(p,q)\} \rightarrow \text{Remove salt noise}$$

○ Midpoint filter

$$g(j,k) = \frac{1}{2} \left[\max_{(p,q) \in W_{jk}} \{f(p,q)\} + \min_{(p,q) \in W_{jk}} \{f(p,q)\} \right]$$

- Combine order statistics and averaging
- Best for randomly distributed noise,
eg. Gaussian or uniform noise

Image Restoration

■ Order-statistics filters

○ Alpha-trimmed mean filter

- Suppose we delete $d/2$ lowest and $d/2$ highest gray-level values of $f(p,q)$ in the neighborhood W_{jk}

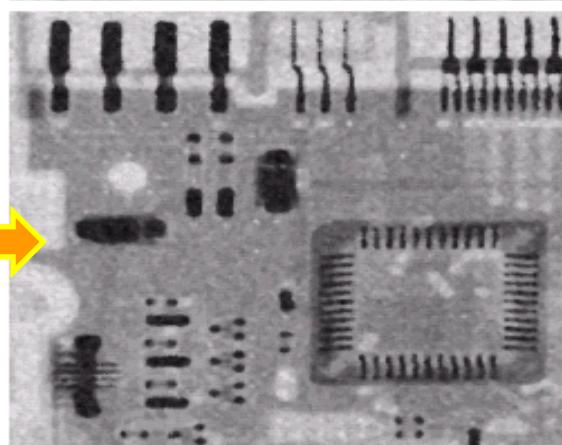
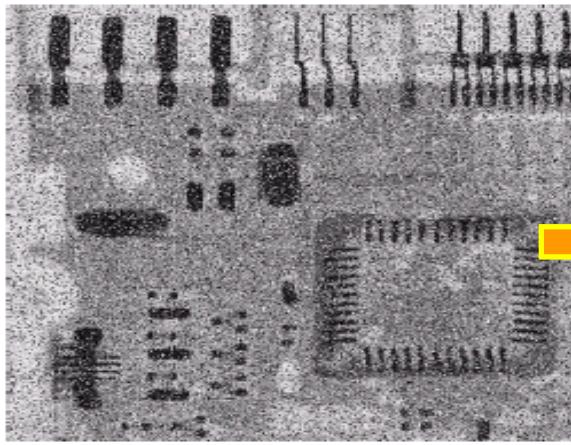
$$g(j,k) = \frac{1}{mn-d} \sum_{(p,q) \in W_{jk}} f_r(p,q)$$

- where $f_r(p,q)$ represents the remaining $mn-d$ pixels
- d ranges from 0 to $mn-1$
- Useful for removing multiple types of noise

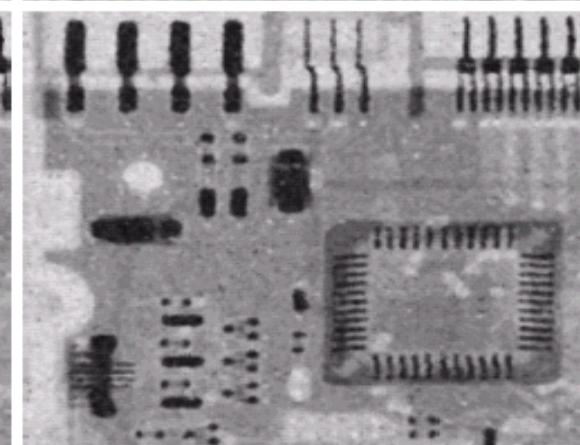
[Image Restoration]

■ Example

- Median and alpha-trimmed filters
 - Image corrupted by both uniform and salt-and-pepper noise



median



Alpha-trimmed 28

Image Restoration

■ Adaptive filters

- **Adaptive, local noise reduction filter
(Adaptive mean filter)**

- σ_η^2 : noise variance
- σ_L^2 : local variance of the pixels in S_{xy}
- m_L : local mean of the pixels in S_{xy}

- If $\sigma_L^2 \gg \sigma_\eta^2$, return a value close to $f(j,k)$

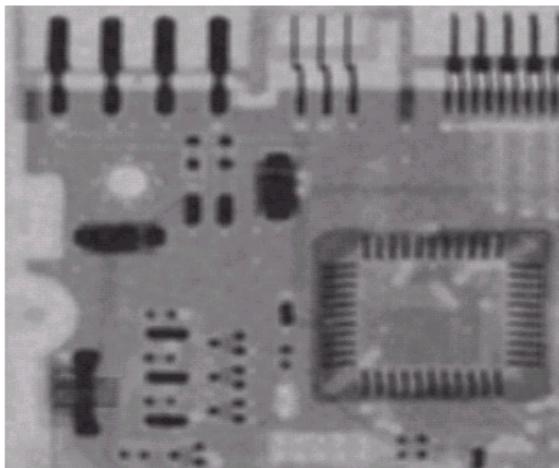
$$g(j,k) = f(j,k) - \frac{\sigma_\eta^2}{\sigma_L^2} (f(j,k) - m_L)$$

- If $\sigma_L^2 \approx \sigma_\eta^2$, return arithmetic mean of the pixels in S_{xy}

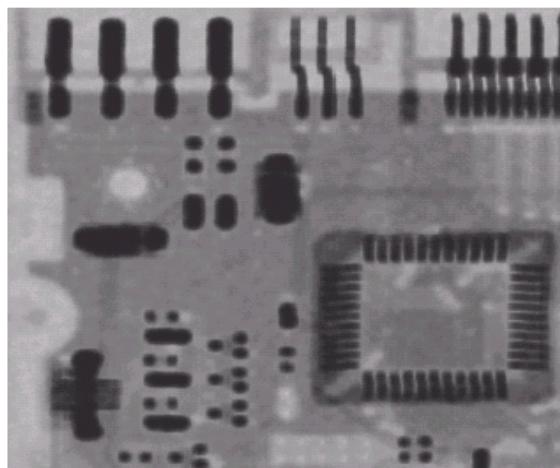
Image Restoration

■ Example

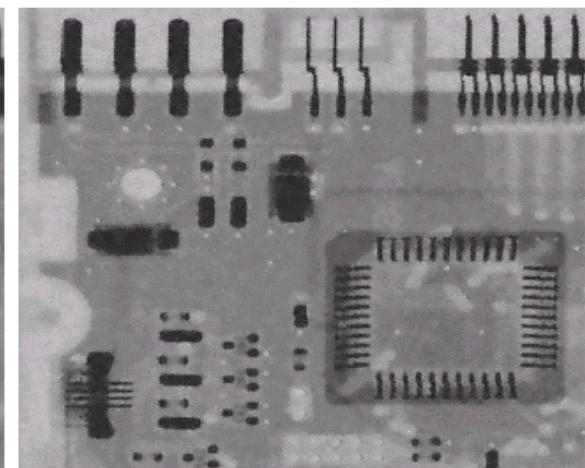
- Adaptive noise reduction filter
 - Compared with arithmetic and geometric mean filters



arithmetic



geometric



adaptive noise reduction

Image Restoration

■ Adaptive filters

○ Adaptive median filter

z_{\min} = minimum gray level value in $W_{j,k}$

z_{\max} = maximum gray level value in $W_{j,k}$

z_{med} = median of gray levels in $W_{j,k}$

z_{jk} = gray level at coordinates (j, k)

W_{\max} = maximum allowed size of W_{jk}

Level A:

$$A_1 = z_{med} - z_{\min}$$

$$A_2 = z_{med} - z_{\max}$$

if $A_1 > 0$ and $A_2 < 0$, go to *Level B*

else increase the window size

if window size $\leq W_{\max}$, repeat Level A

else output z_{jk}

Level B:

$$B_1 = z_{jk} - z_{\min}$$

$$B_2 = z_{jk} - z_{\max}$$

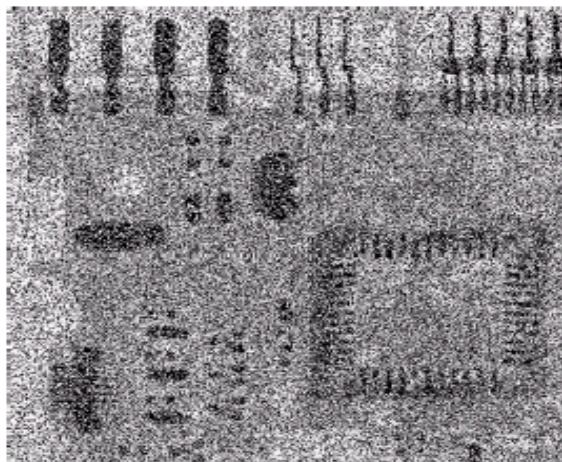
if $B_1 > 0$ and $B_2 < 0$, output z_{jk}

else output z_{med}

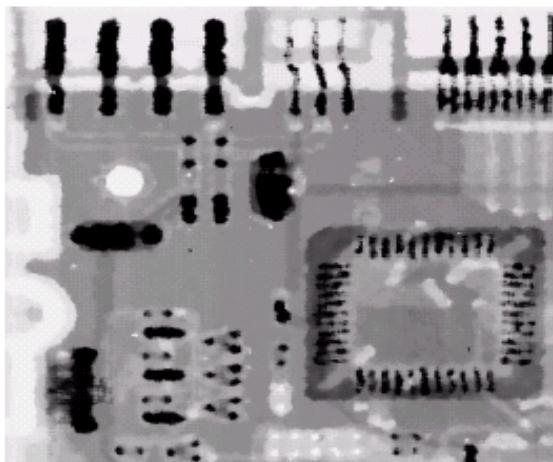
[Image Restoration]

■ Example

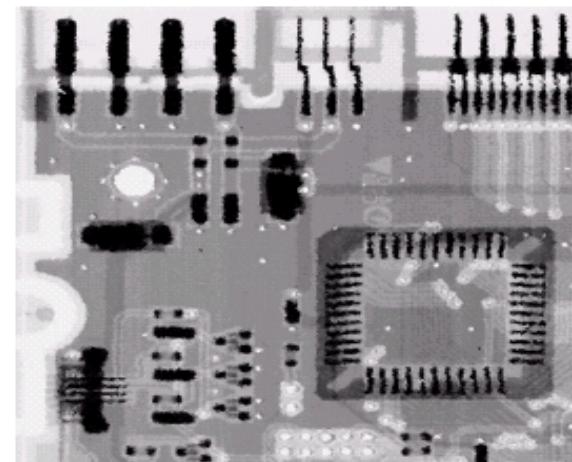
- Adaptive median filter



salt-and-pepper



median



adaptive median

- Handle impulse noise with larger probabilities
- Preserve detail while smoothing nonimpulse noise
- Change the size of filter

[Image Restoration]

- Periodic Noise Reduction by Frequency Domain Filtering
 - Example

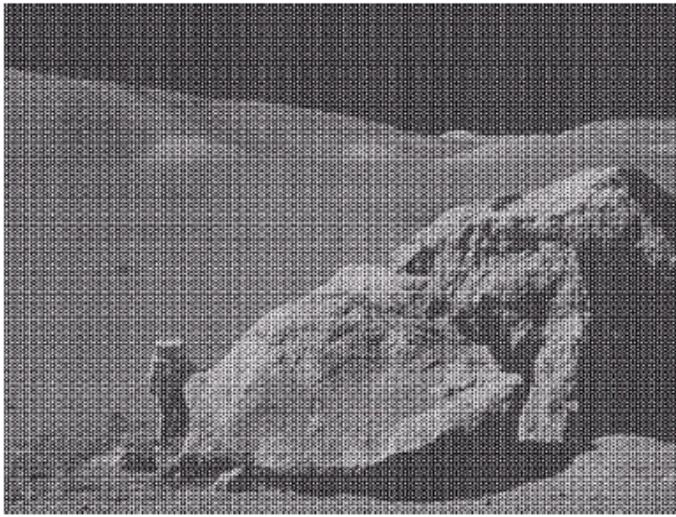
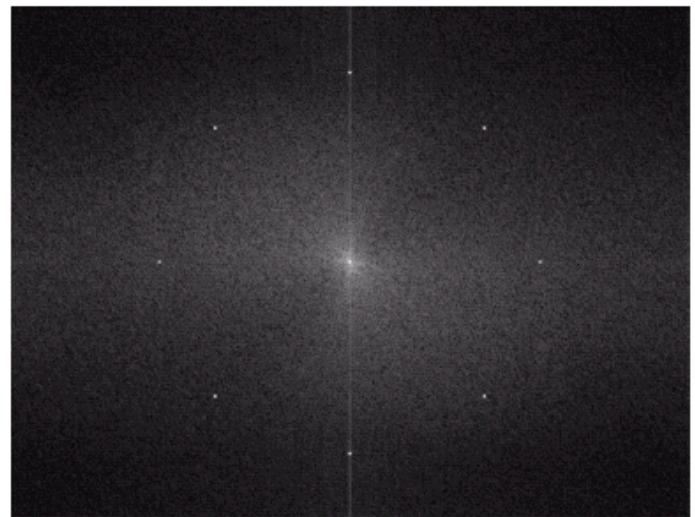


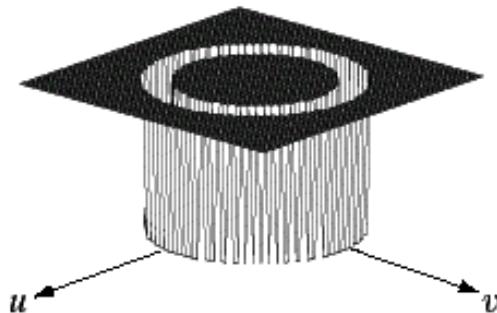
Image corrupted by sinusoidal noise



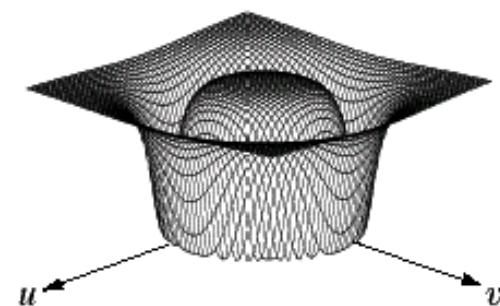
Spectrum of the corrupted image

Image Restoration

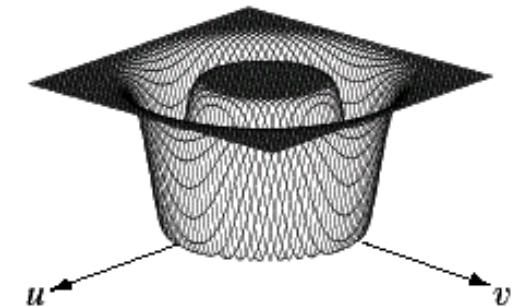
- Periodic Noise Reduction by Frequency Domain Filtering
 - Band-reject Filters



ideal



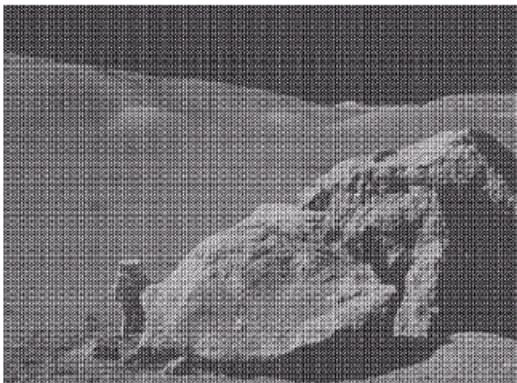
Butterworth (of order 1)



Gaussian

[Image Restoration]

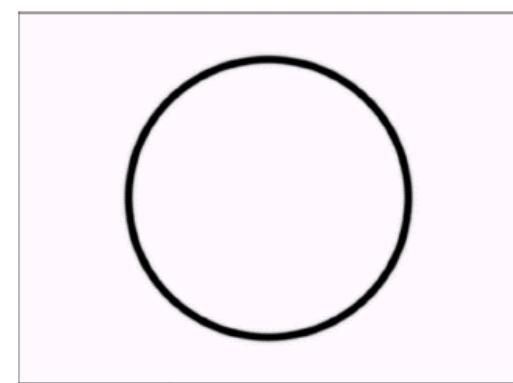
■ Example



Degraded image



Spectrum



Butterworth (of order 1)



Restored image

[

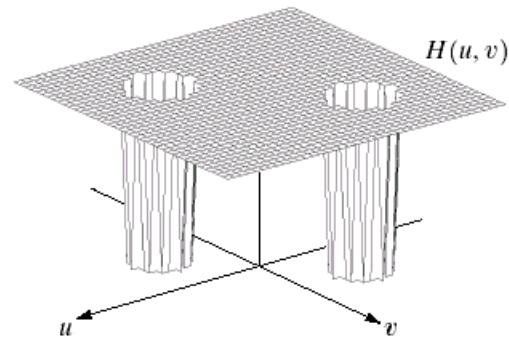
Image Restoration

]

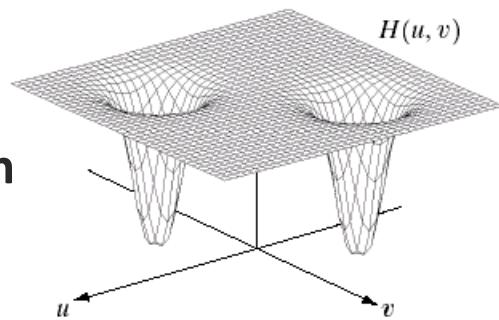
- Periodic Noise Reduction by Frequency Domain Filtering

- Notch Filters

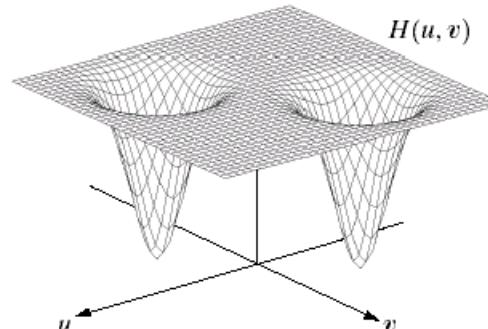
ideal



Butterworth

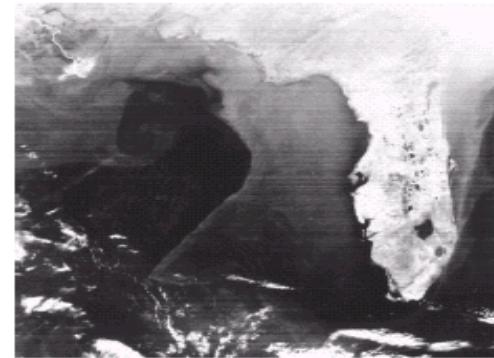


Gaussian

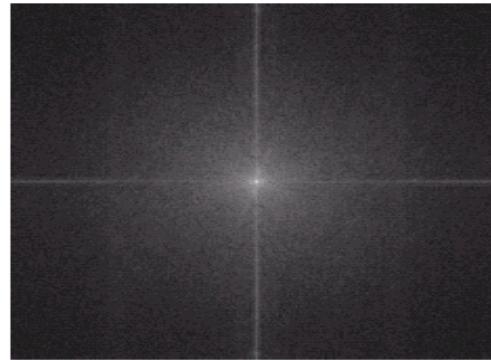


[Image Restoration]

■ Example



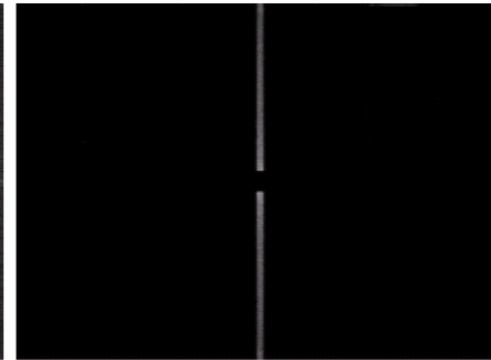
Fourier spectrum



Noise pattern



Notch pass filter



Result

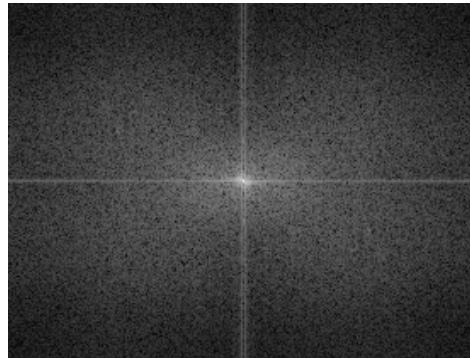


[Image Restoration]

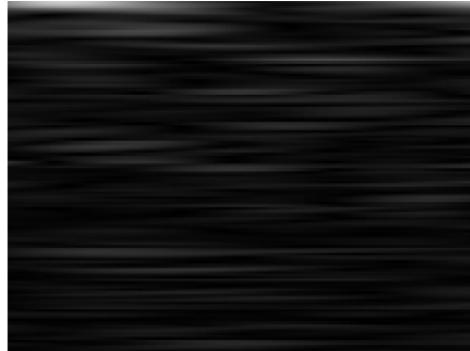
■ Example



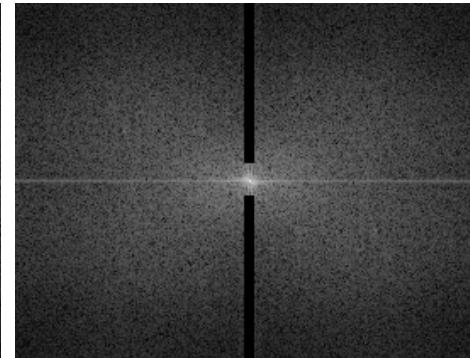
Fourier spectrum



Noise pattern



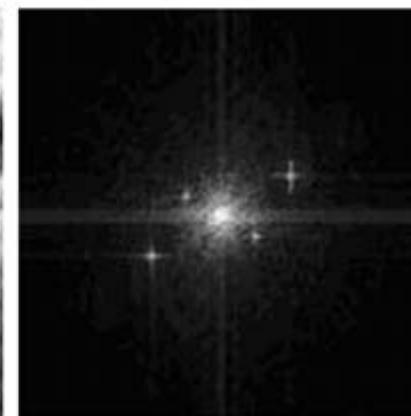
Notch reject filter



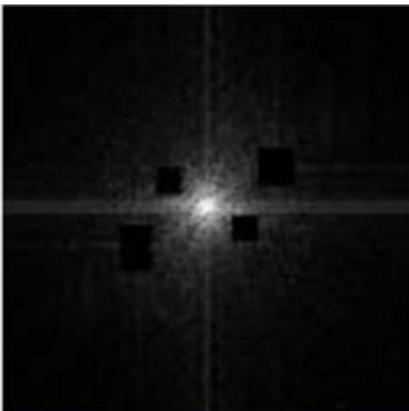
Result

Image Restoration

Example



Fourier spectrum



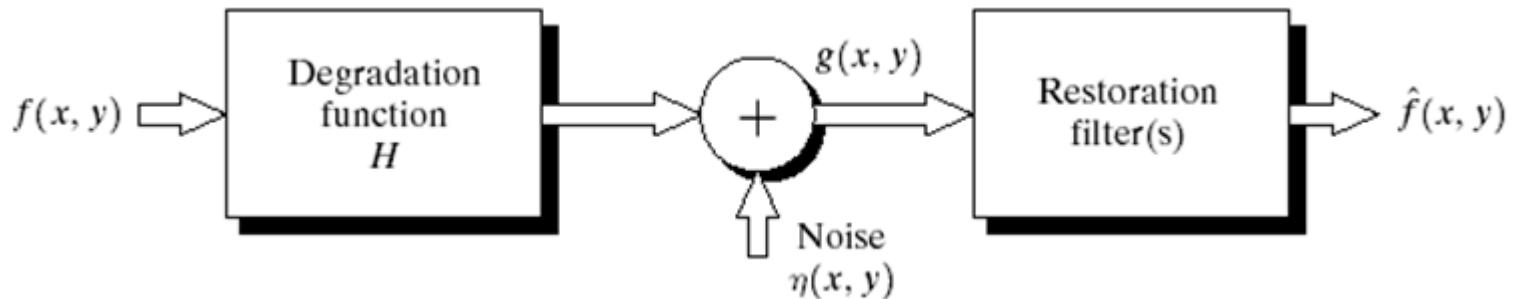
Notch filter



Result

[Image Restoration]

■ Inverse Filter



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

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

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

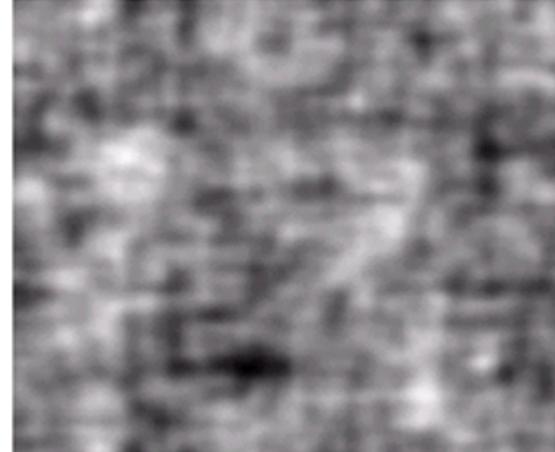
[Image Restoration]

■ Examples

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



degraded image



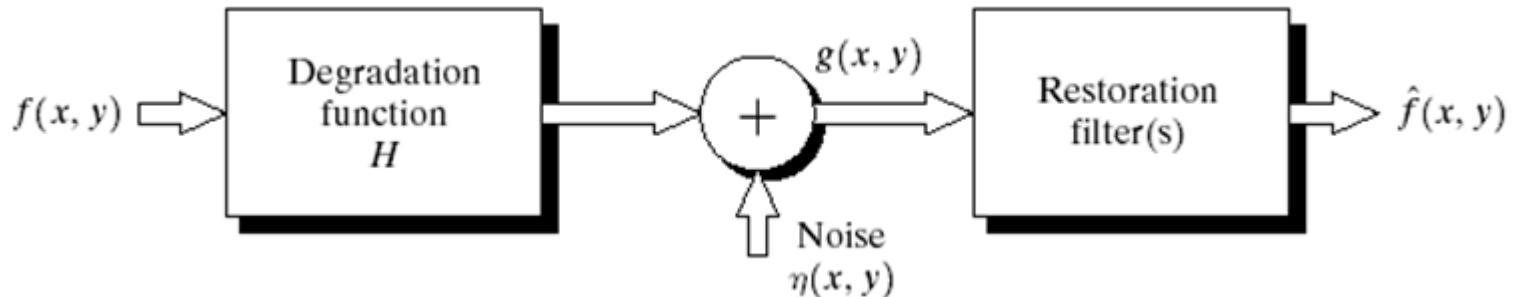
full inverse filter



limited inverse filter

Image Restoration

Wiener Filter



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

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

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

- The objective is to find $H_{Wiener}(u, v)$ with minima MSE between original image and received one.

$$\varepsilon = E[(F(u, v) - \hat{F}(u, v))^2]$$

[Image Restoration]

■ Wiener Filter

$$\begin{aligned}\varepsilon &= E[(F - \hat{F})^2] \\ &= E[(F - H_{Wiener}G)^2] \\ &= E[(F - H_{Wiener}(HF + N))^2] \\ &= E[((1 - HH_{Wiener})F - H_{Wiener}N))^2] \\ &= (1 - HH_{Wiener})\overline{(1 - HH_{Wiener})}E[F^2] - (1 - HH_{Wiener})\overline{H_{Wiener}}E[F\overline{N}] \\ &\quad - \overline{(1 - HH_{Wiener})}H_{Wiener}E[\overline{F}N] + H_{Wiener}\overline{\overline{H_{Wiener}}}E[N^2]\end{aligned}$$

Image Restoration

■ Wiener Filter

- Noise is independent to signal, and its mean is zero

$$E[\bar{F}N] = 0 = E[F\bar{N}]$$

- Define power spectral densities (PSD) as follows

$$D_F(u, v) = E[F(u, v)^2]$$

$$D_N(u, v) = E[N(u, v)^2]$$

- Hence

$$\varepsilon = (1 - H\overline{H}_{Wiener})(\overline{1 - H\overline{H}_{Wiener}})D_F + H_{Wiener}\overline{H_{Wiener}}D_N$$

Image Restoration

■ Wiener Filter

- To minimize the error, let the derivative to be zero

$$\frac{d\varepsilon}{dH_{Wiener}} = 0 = -H\overline{(1 - HH_{Wiener})}D_F + \overline{H_{Wiener}}D_N$$

- Then

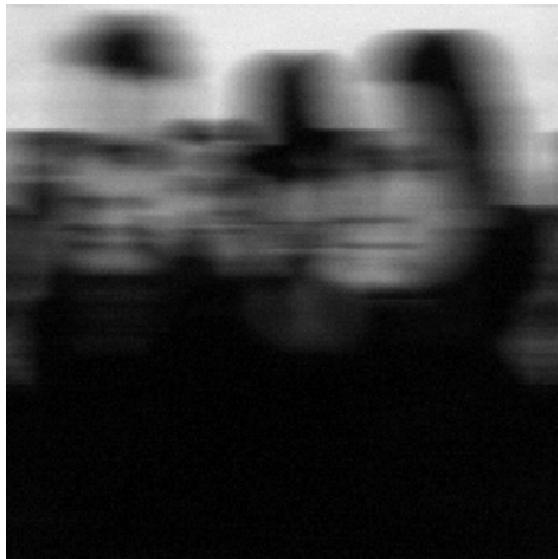
$$H_{Wiener}(u, v) = \frac{\overline{H}(u, v)D_F(u, v)}{|H(u, v)|^2 D_F(u, v) + D_N(u, v)}$$

[Image Restoration]

■ Examples



Original image



Add motion blur
and Gaussian noise



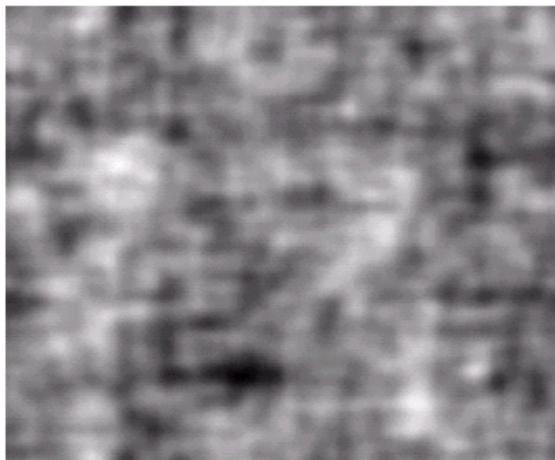
Restored by
Wiener filter

[Image Restoration]

■ Other examples



degraded image



full inverse filter



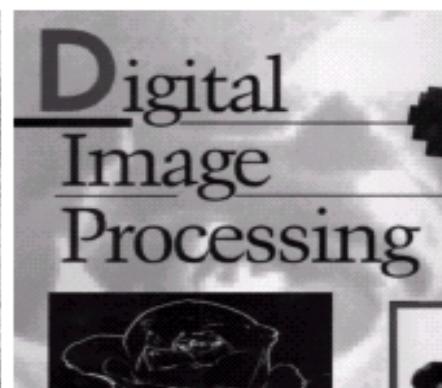
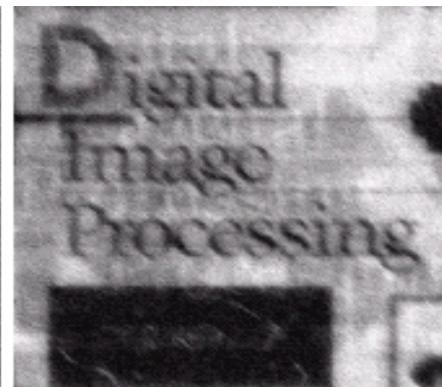
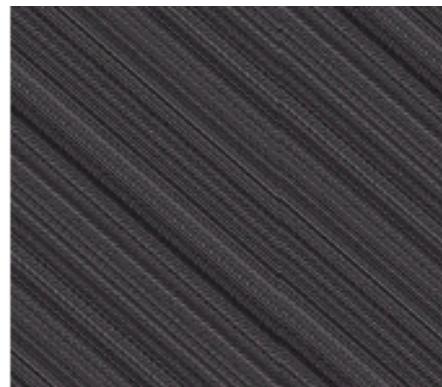
limited inverse filter



Wiener filter

[Image Restoration]

■ Other examples



with motion blur
& additive noise

inverse filtering

Wiener filter