بسمه‌تعالی

Digital Image Processing

Image Restoration
(Chapter 7)

Image Restoration

- Image restoration vs. image enhancement
  - Enhancement:
    - largely a subjective process
    - Priori knowledge about the degradation is not a must (sometimes no degradation is involved)
    - Procedures are heuristic and take advantage of the psychophysical aspects of human visual system
  - Restoration:
    - more an objective process
    - Images are degraded
    - Tries to recover the images by using the knowledge about the degradation

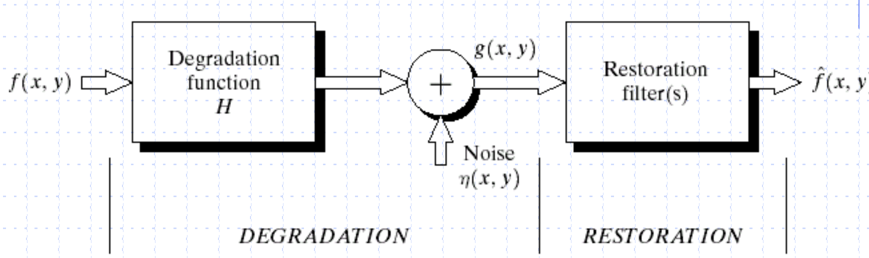
An Image Degradation Model

- Two types of degradation
  - Additive noise
    - Spatial domain restoration (denoising) techniques are preferred
  - Image blur
    - Frequency domain methods are preferred
- We model the degradation process by a degradation function h(x,y), an additive noise term, $\eta(x,y)$, as:

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

  - f(x,y) is the (input) image free from any degradation
  - g(x,y) is the degraded image
  - $*$ is the convolution operator
  - The goal is to obtain an estimate of f(x,y) according to the knowledge about the degradation function h and the additive noise η
  - In frequency domain: G(u,v)=H(u,v)F(u,v)+N(u,v)

A Model of the Image Degradation/Restoration Process



Noise Model

- We first consider the degradation due to noise only
  - h is an impulse for now (H is a constant)
- White noise
  - Autocorrelation function is an impulse function multiplied by a constant
    - $$a(x, y) = \sum_{t=0}^{N-1} \sum_{s=0}^{M-1} \eta(s,t) \cdot \eta(s-x, t-y) = N_0 \delta(x, y)$$
    - It means there is no correlation between any two pixels in the noise image
    - There is no way to predict the next noise value
  - The spectrum of the autocorrelation function is a constant

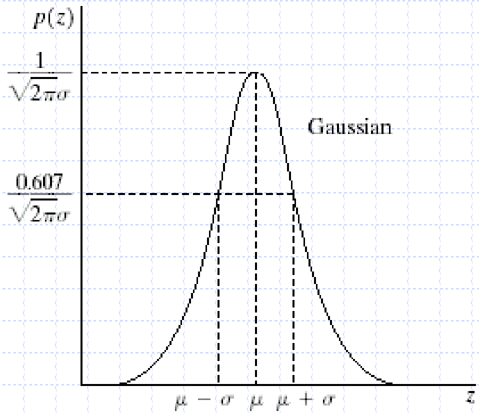
Gaussian Noise

- Noise (image) can be classified according the distribution of the values of pixels (of the noise image) or its (normalized) histogram
- Gaussian noise is characterized by two parameters, μ (mean) and σ^2 (variance), by

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

- 70% values of z fall in the range $[(\mu-\sigma),(\mu+\sigma)]$
- 95% values of z fall in the range $[(\mu-2\sigma),(\mu+2\sigma)]$

Gaussian Noise



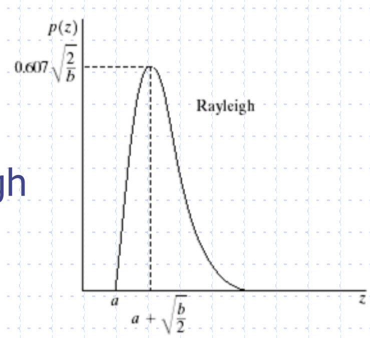
Rayleigh noise *Model*

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

- The mean and variance of this density are given by

$$\mu = a + \sqrt{\pi b / 4} \text{ and } \sigma^2 = \frac{b(4-\pi)}{4}$$

- a and b can be obtained through mean and variance



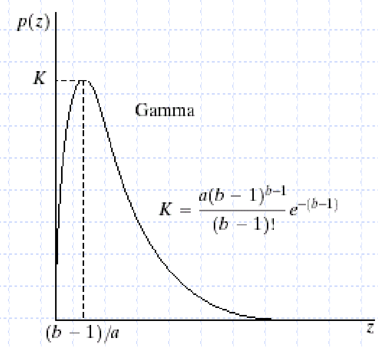
Erlang (Gamma) noise ***Models***

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

- The mean and variance of this density are given by

$$\mu = b/a \ \text{ and } \sigma^2 = \frac{b}{a^2}$$

- a and b can be obtained through mean and variance



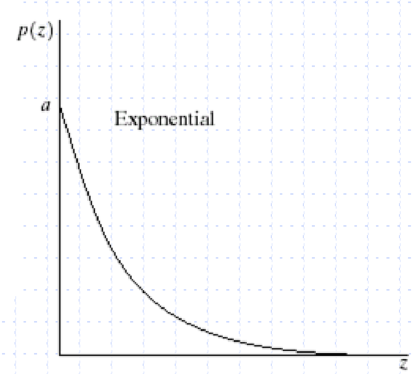
Exponential noise ***Models***

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

- The mean and variance of this density are given by

$$\mu = 1/a \ \text{ and } \sigma^2 = \frac{1}{a^2}$$

- Special case pf Erlang PDF with b=1

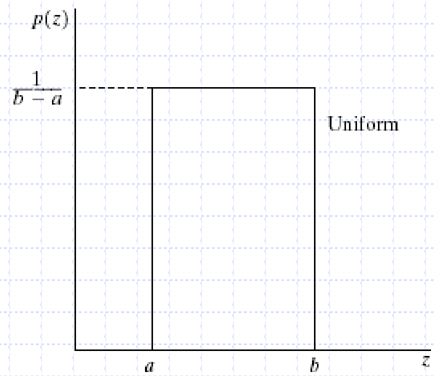


Uniform noise ***Models***

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

- The mean and variance of this density are given by

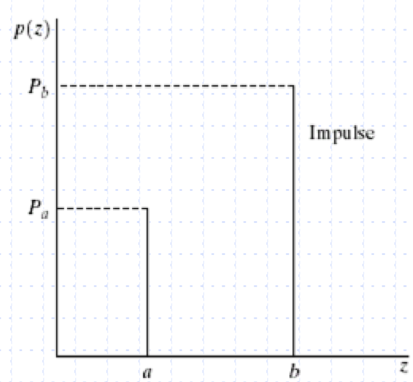
$$\mu=(a+b)/2 \text{ and } \sigma^2=\frac{(b-a)^2}{12}$$



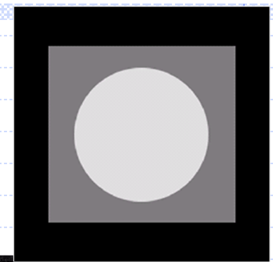
Impulse (salt-and-pepper) noise ***Models***

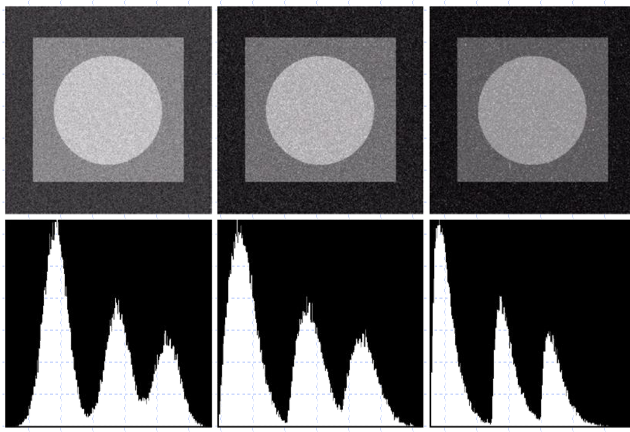
$$p(z)=\begin{cases}P_a & \text{for } z=a\\ P_b & \text{for } z=b\\ 0 & \text{otherwise}\end{cases}$$

- a and b usually are extreme values because impulse corruption is usually large compared with the strength of the image signal
- If either P_a or P_b is zero, the impulse noise is called unipolar



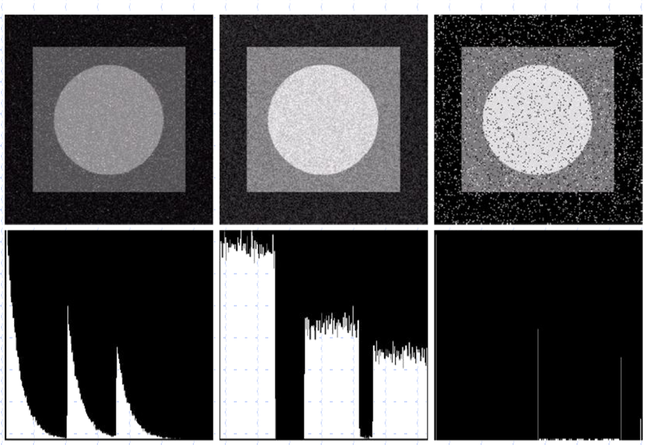
Effect of Adding Noise to Sample Image





Gaussian Rayleigh Gamma

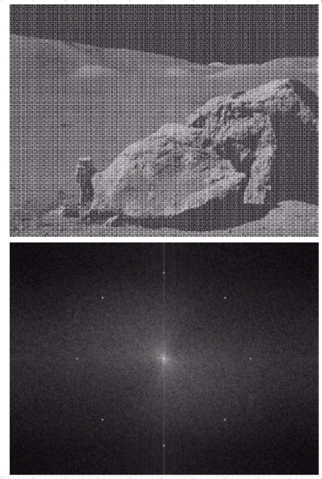
Effect of Adding Noise to Sample Image



Exponential Uniform Salt & Pepper

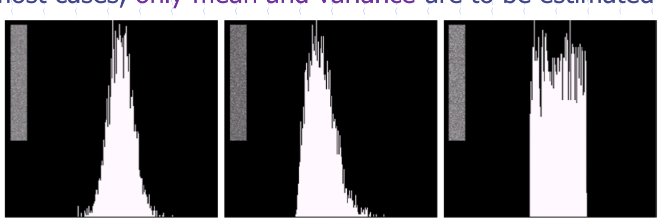
Periodic Noise

- Arises typically from electrical or electromechanical interference during image acquisition
- It can be observed by visual inspection both in the spatial domain and frequency domain
- The only spatially dependent noise will be considered



Estimation of Noise Parameters

- Periodic noise
  - Parameters can be estimated by inspection of the spectrum
- Noise PDFs
  - From sensor specifications
  - Capture a set of images of plain environments
  - Parameters of the PDF can be estimated from small patches of constant regions of the noisy images
  - In most cases, only mean and variance are to be estimated



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.

Restoration of Noise (De-Noising)

- Mean filters
  - Arithmetic mean filter
    - $S_{x,y}$ is the mask

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

  - Geometric mean filters
    - Tends to preserve more details

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

  - Harmonic mean filter
    - Works well for salt noise
    - but fails for pepper noise

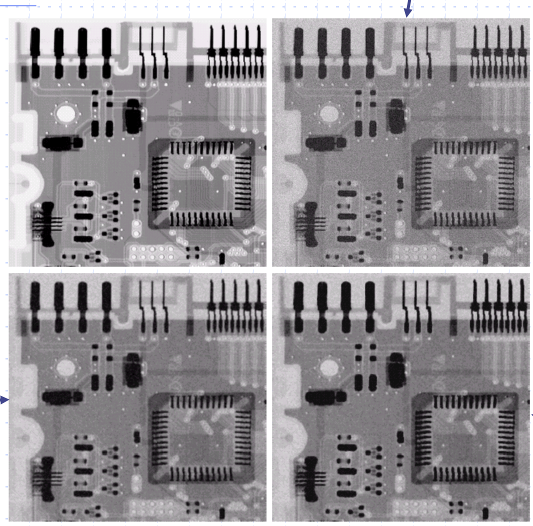
$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{x,y}} \frac{1}{g(s,t)}}$$

  - Contra-harmonic mean filter
    - Q: order of the filter
    - Positive Q works for pepper noise
    - Negative Q works for salt noise
    - Q=0➔arithmetic mean filter
    - Q=-1➔harmonic mean filter

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

De-Noising

Corrupted by Gaussian Noise



Mean Filtering

Geometric Mean Filtering

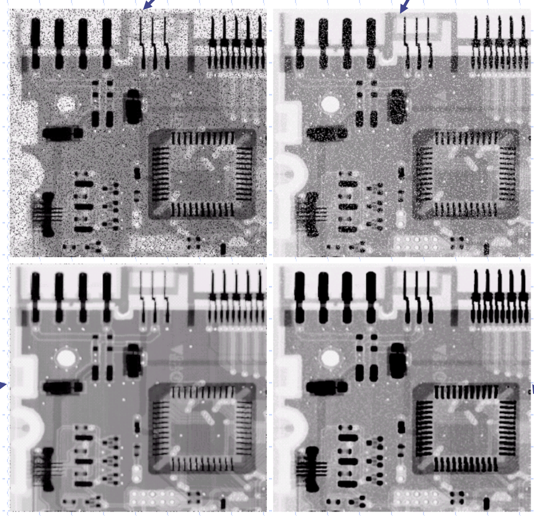
a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

De-Noising

Corrupted by pepper noise

Corrupted by salt noise



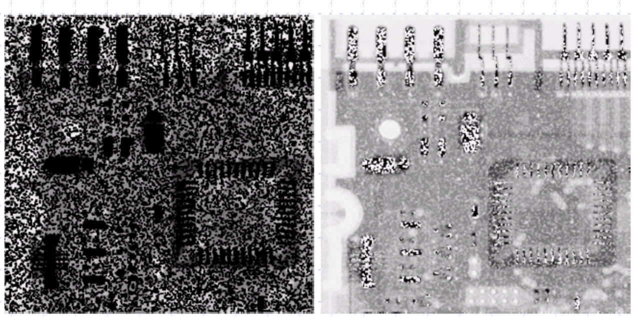
a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

3x3 Contraharmonic Q=1.5

3x3 Contraharmonic Q=-1.5

De-Noising with wrong filter sign



a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

Filters Based on Order Statistics (De-Noising)

- Median filter
  - Median represents the 50th percentile of a ranked set of numbers

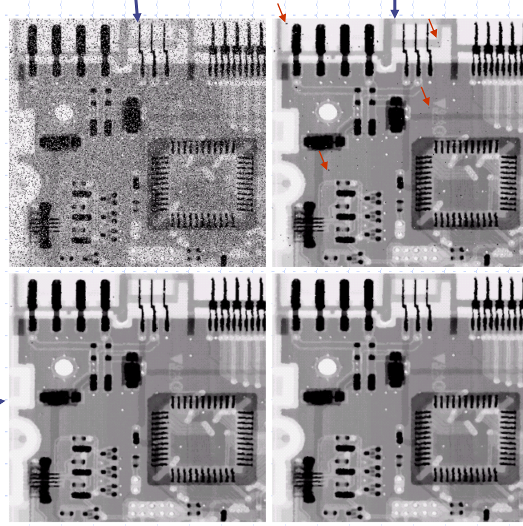
$$\hat{f}(x,y) = \underset{(s,t)\in S_{x,y}}{\text{median}}\{g(s,t)\}$$

- Max and min filter
  - Max filter uses the 100th percentile of a ranked set of numbers
    - Good for removing pepper noise
  - Min filter uses the 1 percentile of a ranked set of numbers
    - Good for removing salt noise
- Midpoint filter

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

    - Works best for noise with symmetric PDF like Gaussian or uniform noise

De-Noising



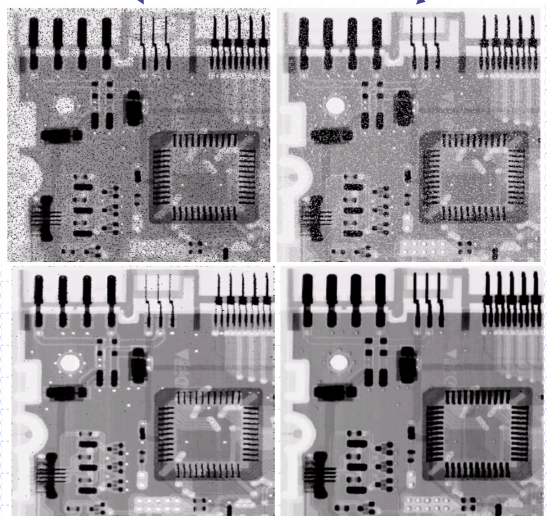
a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3 × 3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

De-Noising

Corrupted by pepper noise

Corrupted by salt noise



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

Max Filtering

Min Filtering

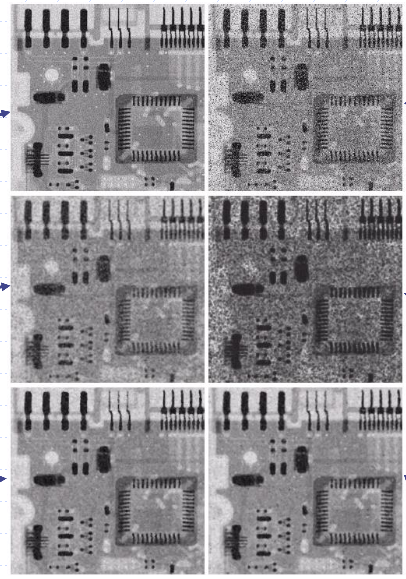
Alpha-Trimmed Mean Filter (De-Noising)

- Take the mean value of the pixels (enclosed by an m×n mask) after deleting the pixels with the d/2 lowest and the d/2 highest gray-level values

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

  - $g_r(s,t)$ represent the remaining mn-d pixels
  - It is useful in situations involving multiple types of noise like a combination of salt-and-pepper and Gaussian

De-Noising



a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5 × 5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

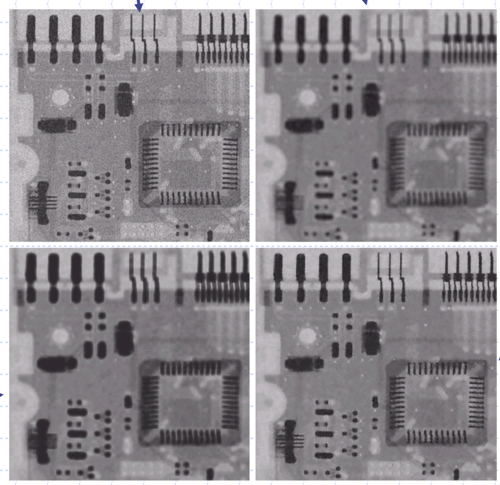
Adaptive Filters (De-Noising)

- Adaptive Local Noise Reduction Filter
  - Assume the variance of the noise σ_η^2 is either known or can be estimated satisfactorily
  - Filtering operation changes at different regions of an image according to local variance σ_L^2 and mean m_L calculated within an M×N region
  - If $\sigma_L^2 > \sigma_\eta^2$, the filtering operation is defined as

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

  - If $\sigma_L^2 < \sigma_\eta^2$, the output takes the mean value
    - That is: $\frac{\sigma_\eta^2}{\sigma_L^2}$ is set to be 1
  - At edges, it is assumes that $\sigma_L^2 > \sigma_\eta^2$

De-Noising



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 Filter (De-Noising)

- Median filter is effective for removing salt-and-pepper noise
  - The density of the impulse noise can not be too large
- Adaptive median filter
  - Notation
    - Z_{min}: minimum gray value in S_{xy}
    - Z_{max}: maximum gray value in S_{xy}
    - Z_{med}: median of gray levels in S_{xy}
    - Z_{xy}: gray value of the image at (x,y)
    - S_{max}: maximum allowed size of S_{xy}

Adaptive Median Filter (De-Noising)

- Two levels of operations
  - 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 by 2
    - If window size <= S_{max} repeat level A
      else output Z_{xy}
  - 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}

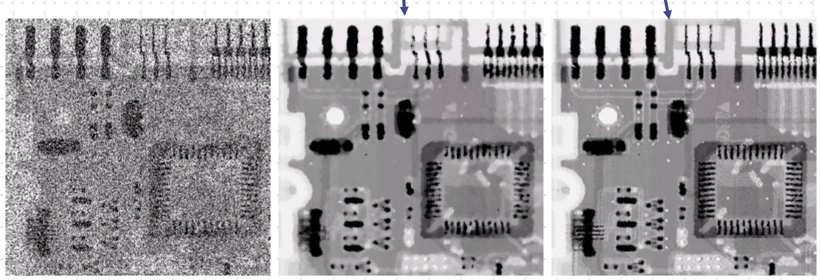
Test whether Z_{med} is part of s-and-p noise.
• If yes, window size is increased

Test whether Z_{xy} is part of s-and-p noise.
• If yes, apply regular median filtering

De-Noising

Median Filtering

Adaptive Median Filtering



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7 × 7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

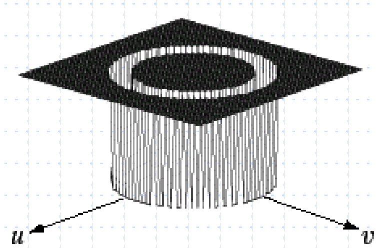
Periodic Noise Reduction by Frequency Domain Filtering

- **Lowpass and highpass filters for image enhancement can be used.**
- **Bandreject, bandpass, and notch filters as tools for periodic noise reduction or removal can also be used.**

Bandreject Filters

- Bandreject filters remove or attenuate a band of frequencies about the origin of the Fourier transform.
- Similar to those LPFs and HPFs, we can construct ideal, Butterworth, and Gaussian bandreject filters.
- **Ideal** bandreject filter

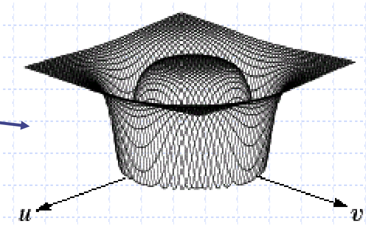
$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)<D_0-\frac{W}{2}\\ 0 & \text{if } D_0-\frac{W}{2}\le D(u,v)\le D_0+\frac{W}{2}\\ 1 & \text{if } D(u,v)>D_0+\frac{W}{2}\end{cases}$$



Butterworth *Bandreject Filters*

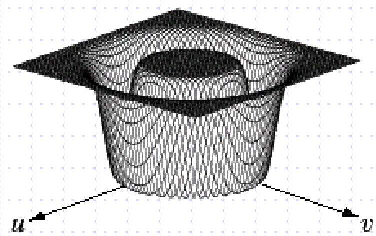
$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}}$$

n =1

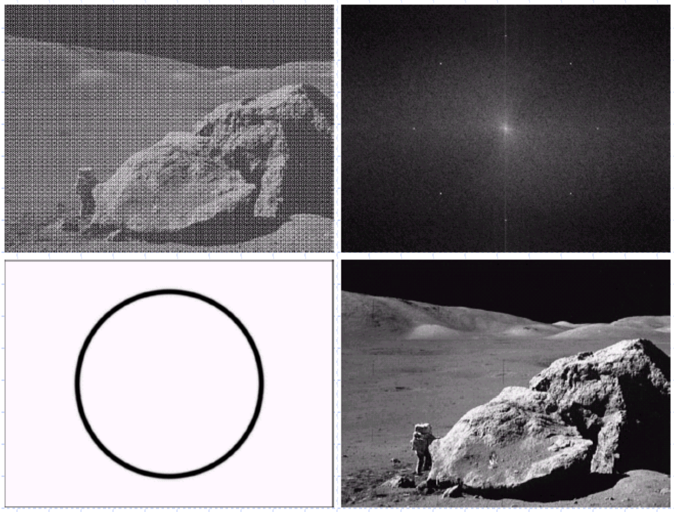


Gaussian *Bandreject Filters*

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



Butterworth *Bandreject Filters*
For reduction of Periodic Noise



a b
c d

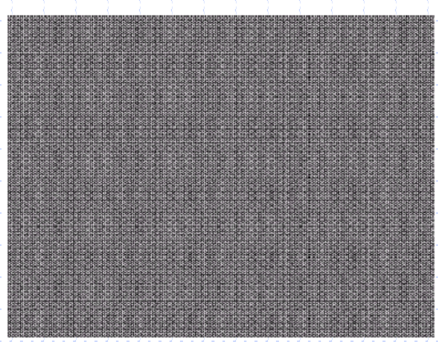
FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Bandbass Filters

Bandpass filter performs the opposite of a bandreject filter

$$H_{bp}(u,v) = 1 - H_{br}(u,v)$$

FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



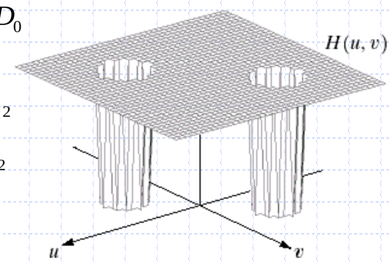
Notch Filters

- Notch filter rejects frequencies in predefined neighborhoods about a center frequency.
- It appears in symmetric pairs about the origin because the Fourier transform of a real valued image is symmetric.
- **Ideal** notch filter

$$H(u,v)=\begin{cases}0 & \text{if } D_1(u,v)\le D_0 \text{ or } D_2(u,v)\le D_0\\ 1 & \text{otherwise}\end{cases}$$

$$D_1(u,v)=\left[(u-M/2-u_0)^2+(v-N/2-v_0)^2\right]^{1/2}$$

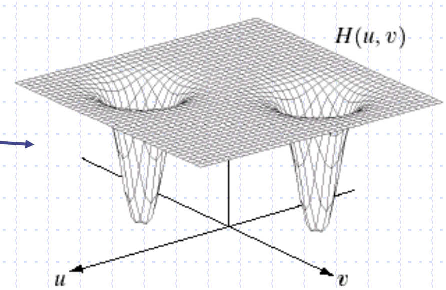
$$D_2(u,v)=\left[(u-M/2+u_0)^2+(v-N/2+v_0)^2\right]^{1/2}$$



Butterworth *Notch Filters*

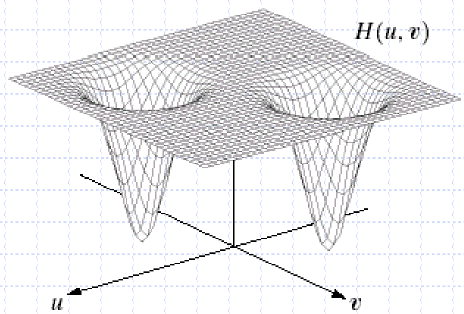
$$H(u,v)=\frac{1}{1+\left[\dfrac{D_0^2}{D_1(u,v)D_2(u,v)}\right]^{2n}}$$

n=2



Gaussian ***Notch Filters***

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]}$$



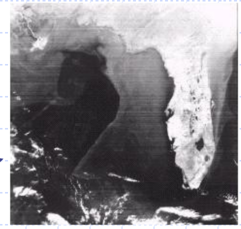
Notch Filters that pass, rather than suppress

$$H_{np}(u,v) = 1 - H_{nr}(u,v)$$

- *NR* filters become highpass filters if *NP* filters become lowpass, and vise versa.

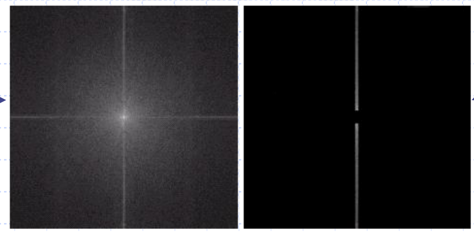
Notch Filters

You can see the effect of scan lines



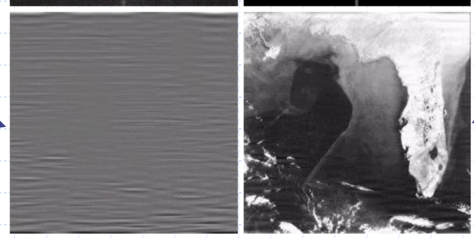
Spectrum of image

Notch pass filter

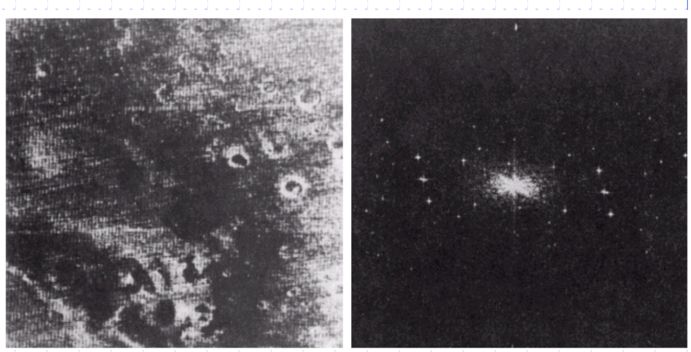


IFT of NP filtered image

Result of NR filter



Optimum Notch Filtering



a b

FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6.* (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

Optimum Notch Filtering

- In the ideal case, the original image can be restored if the noise can be estimated completely.
  - That is: $$f(x,y) = g(x,y) - \eta(x,y)$$
- However, the noise can be only partially estimated. This means the restored image is not exact.
  - Which means $$\hat{f}(x,y) = g(x,y) - \hat{\eta}(x,y)$$

$$\hat{\eta}(x,y) = IFT\{H(u,v)G(u,v)\}$$

Optimum Notch Filtering

- The restored image can be improve by introducing a modulation function

$$\hat{f}(x,y) = g(x,y) - w(x,y)\hat{\eta}(x,y)$$

  - Here the modulation function is a constant within a neighborhood of size (2a+1) by (2b+1) about a point (x,y)
  - We optimize its performance by minimizing the local variance of the restored image at the position (x,y)

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left[\hat{f}(x+s, y+t) - \bar{\hat{f}}(x,y) \right]^2$$

$$\bar{\hat{f}}(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \hat{f}(x+s, y+t)$$

Optimum Notch Filtering

Points on or near Edge of the image can be treated by considering partial neighborhoods

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a}\sum_{t=-b}^{b}\{[g(x+s,y+t) - w(x+s,y+t)\hat{\eta}(x+s,y+t)] - [\bar{g}(x,y) - \overline{w(x,y)\hat{\eta}(x,y})]\}^2$$

Assumption: $w(x+s,y+t) = w(x,y) \;\; for \; -a \le s \le a \; and \; -b \le t \le b$

$$\Rightarrow \overline{w(x,y)\hat{\eta}(x,y)} = w(x,y)\bar{\hat{\eta}}(x,y)$$

Optimum Notch Filtering

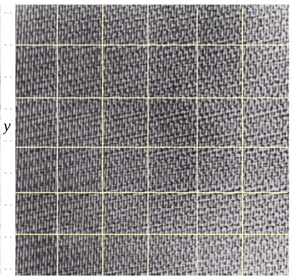
$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a}\sum_{t=-b}^{b}\{[g(x+s,y+t) - w(x,y)\hat{\eta}(x+s,y+t)] - [\bar{g}(x,y) - w(x,y)\bar{\hat{\eta}}(x,y)\}^2$$

To minimize $\sigma^2(x,y)$ $\quad \frac{\partial \sigma^2(x,y)}{\partial w(x,y)} = 0$

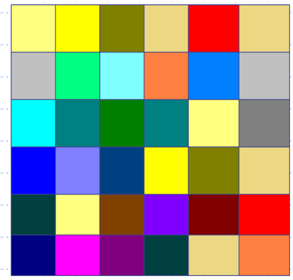
$$\Rightarrow w(x,y) = \frac{\overline{g(x,y)\hat{\eta}(x,y)} - \bar{g}(x,y)\bar{\hat{\eta}}(x,y)}{\overline{\hat{\eta}^2}(x,y) - \bar{\hat{\eta}}^2(x,y)}$$

Optimum Notch Filtering

$g(x, y)$



$\hat{\eta}(x, y)$



Optimum Notch Filtering

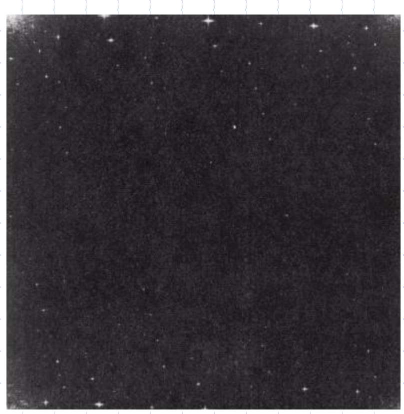
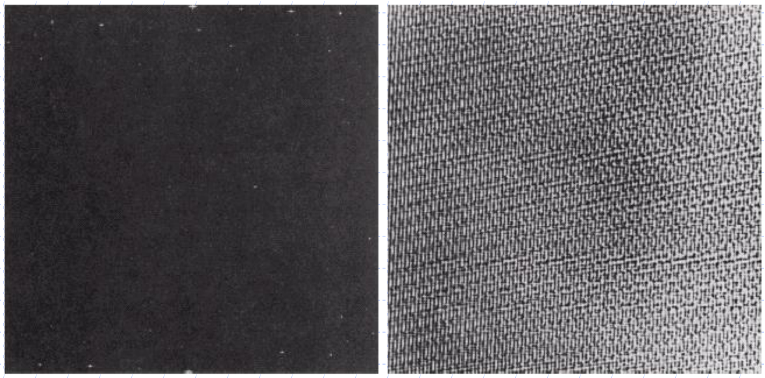


FIGURE 5.21 Fourier spectrum (without shifting) of the image shown in Fig. 5.20(a). (Courtesy of NASA.)

Optimum Notch Filtering



a b

FIGURE 5.22 (a) Fourier spectrum of $N(u, v)$, and (b) corresponding noise interference pattern $\eta(x, y)$. (Courtesy of NASA.)

Optimum Notch Filtering

$g(x, y)$

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

Image size: 512x512

a=b=15

System Degradation effects:
Linear, Position-Invariant Degradation

- Degradation Model

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

In the absence of additive noise:

For scalar values of a and b, H is linear if:

$$H[a\,f_1(x,y) + b\,f_2(x,y)] = a\,H[f_1(x,y)] + b\,H[f_2(x,y)]$$

H is Position-Invariant if:

$$g(x,y) = H[f(x,y)] \Rightarrow H[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta)$$

Linear, Position-Invariant Degradation

In the presence of additive noise:

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)h(x-\alpha, y-\beta)\,d\alpha\,d\beta + \eta(x,y)$$

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

- **Many types of degradation can be approximated by linear, position-invariant processes**
- **Extensive tools of linear system theory are available**
- **In this situation, restoration is *image deconvolution***

Estimating the Degradation Function

- Ways to estimate the degradation function for use in image restoration:
  - **Observation**
  - **Experimentation**
  - **Mathematical modeling**

- ***Estimating by Image Observation***
- We look for a small section of the image that has strong signal content ($g_s(x,y)$) and then construct an un-degradation of this section by using sample gray levels ($\hat{f}_s(x,y)$).

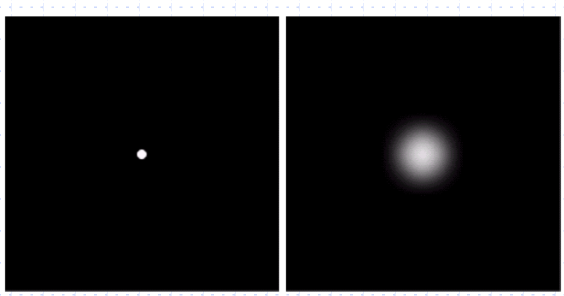
$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

- Now, we construct a function $H(u,v)$ on a large scale, but having the same shape.

Estimating by Experimentation

- We try to obtain impulse response of the degradation by imaging an impulse (small dot of light) using the system. Therefore

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

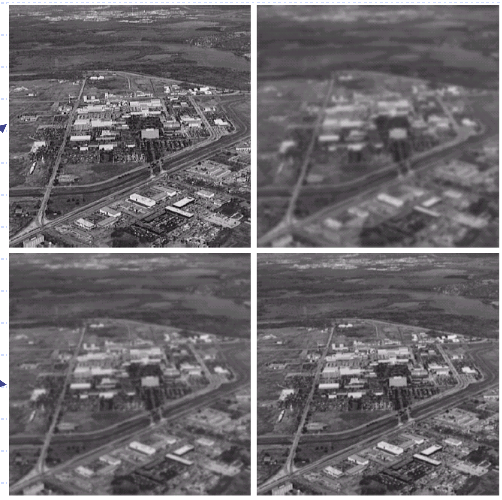


a b

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

Estimating by Modeling

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



Estimating by Modeling

- Blurring by linear motion:

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

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

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

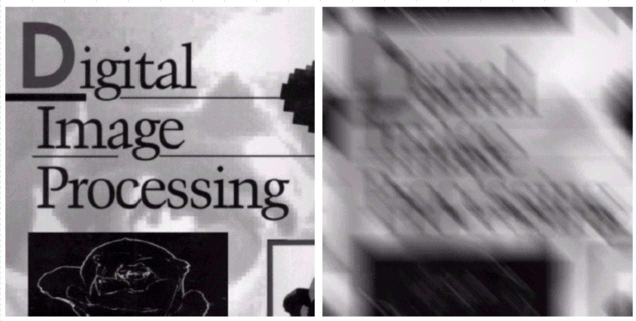
$$if \;\; x_0(t) = at/T \;\; and \;\; y_0(t) = 0 \;\Rightarrow\; H(u,v) = \int_0^T e^{-2\pi uat/T}dt$$

$$= \frac{T}{\pi ua}\sin(\pi ua)e^{-j\pi ua}$$

Estimating by Modeling

$$\text{if} \;\; x_0(t) = at/T \;\; \text{and} \;\; y_0(t) = bt/T \;\; \Rightarrow$$

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



a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Inverse Filtering

The simplest approach to restoration is direct inverse filtering:

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

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

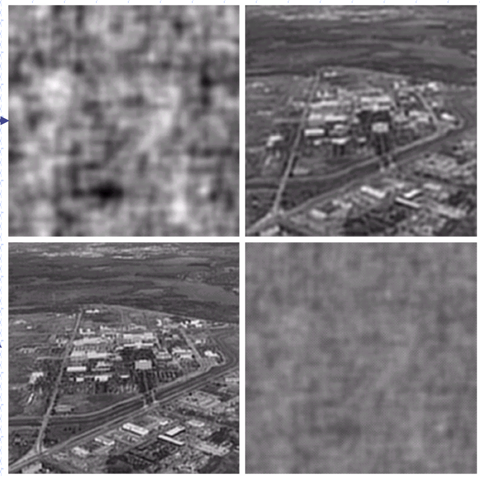
Even if we know the degradation function, we cannot recover the un-degraded image

If the degradation has zero or very small values, then the ratio *N/H could easily dominate our estimation of F*.

One approach to get around the zero or small-value problem is to limit the filter frequencies to value near the origin.

Inverse Filtering

Degraded Image



Full inverse Filtering

Filtering with H cut off outside a radius of 40

Filtering with H cut off outside a radius of 70

Filtering with H cut off outside a radius of 85

Minimum Mean Square Error Filtering (Wiener Filtering)

This approach incorporate both the degradation function and statistical characteristic of noise into the restoration process.

Image and noise are random process

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

The objective is to find an estimation for f such that minimized e^2

Minimum Mean Square Error Filtering (Wiener Filtering)

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

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

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

If the noise is zero, then the Wiener Filter reduces to the inverse filter.

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

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

Minimum Mean Square Error Filtering (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

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

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.

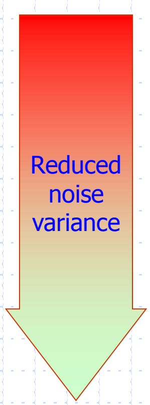
Full inverse filtering

Radially limited inverse filtering

Wiener filtering

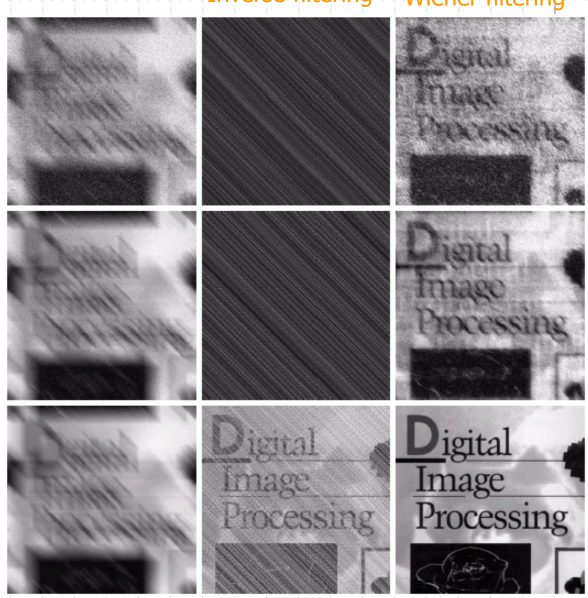
(K was chosen interactively)

Wiener Filtering



Inverse filtering

Wiener filtering



Geometric Transformations

- **Unlike the techniques discussed so far, geometric transformations modify the spatial relationships between pixels in an image.**

Geometric transformation: ***RUBBER-SHEET TRANSFORMATION***

Basic Operations:

1. Spatial Transformation

2. Gray-level Interpolation

Spatial Transformations

$$x' = r(x, y)$$

$$y' = s(x, y)$$

$$x' = r(x, y) = c_1 x + c_2 y + c_3 xy + c_4$$

$$y' = s(x, y) = c_5 x + c_6 y + c_7 xy + c_8$$

Spatial Transformations

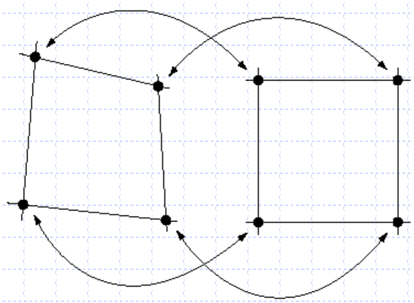


FIGURE 5.32 Corresponding tiepoints in two image segments.

Gray-level Interpolation

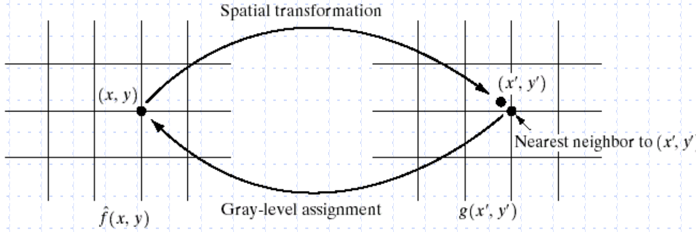


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.

$$v(x,y) = ax' + by' + cx'y' + d$$

Geometric Transformations

Image showing tiepoints

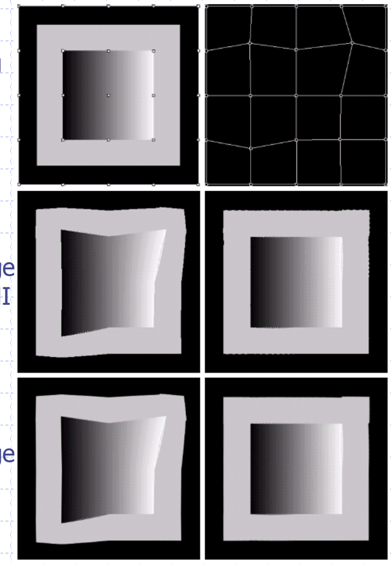
Tiepoints after geo. distortion

Geo. Dist. Image using nearest NI

Restored image

Geo. Dist. Image using Bilinear Interp.

Restored image



Geometric Transformations

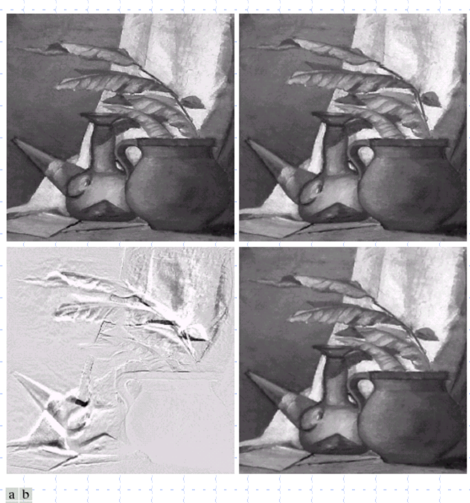


FIGURE 5.35 (a) An image before geometric distortion. (b) Image geometrically distorted using the same parameters as in Fig. 5.34(e). (c) Difference between (a) and (b). (d) Geometrically restored image.