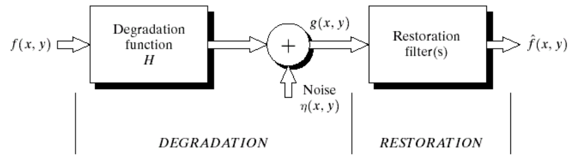
Digital Image Processing
Chapter 5:
Image Restoration

Concept of Image Restoration

Image restoration is to restore a degraded image back to the original image while image enhancement is to manipulate the image so that it is suitable for a specific application.



Degradation model:

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

where $h(x,y)$ is a system that causes image distortion and $\eta(x,y)$ is noise, * is the convolution.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

White Noise

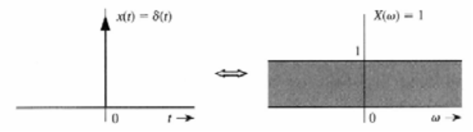
White noise : *Fourier* spectrum of noise is constant

Fourier transform of unit impulse function :

$$F(\delta(t)) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1$$

Unit impulse contains Component at Every Frequency

$$\delta(t) \Longleftrightarrow 1$$



White Noise(white random process)

A continuous time random process $w(t)$ is a white noise process if and only if its mean function and autocorrelation function satisfy the following:

$$\mu_w(t) = \mathbb{E}\{w(t)\} = 0$$

$$R_{ww}(t_1, t_2) = \mathbb{E}\{w(t_1)w(t_2)\} = (N_0/2)\delta(t_1 - t_2)$$

→ Stochastically independent in t_1 and t_2

White Gaussian noise(identically distributed) : $p_{ij}=p=N(0,\sigma)$

Salt and pepper noise :

Noise Models

Noise cannot be predicted but can be approximately described in statistical way using the probability density function (PDF)

Gaussian noise: z denotes *gray level*

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

Rayleigh noise

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

Erlang (Gamma) noise

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

Noise Models (cont.)

Exponential noise

$$p(z) = ae^{-az}$$

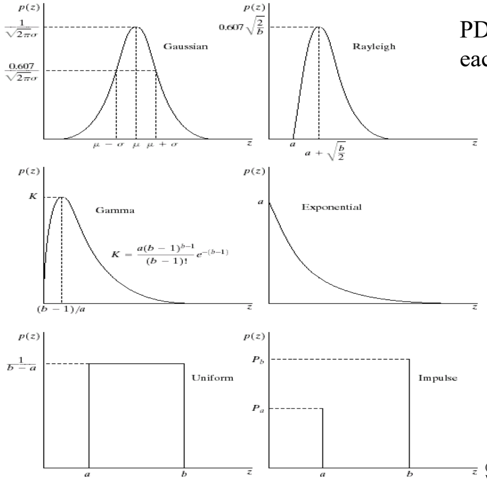
Uniform noise

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

Impulse (salt & pepper) noise

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

PDF: Statistical Way to Describe Noise



PDF tells how much each z value occurs.

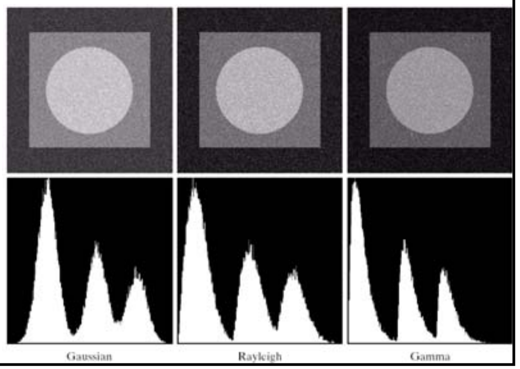
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Image Degradation with Additive Noise

Original image

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

Degraded images



Histogram

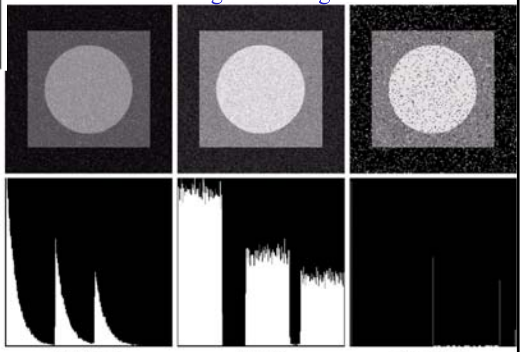
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Image Degradation with Additive Noise (cont.)

Original image

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

Degraded images



Histogram

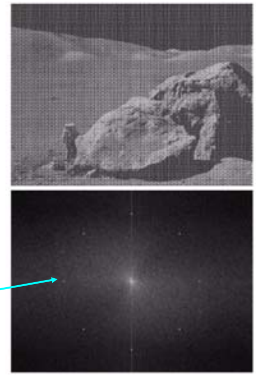
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Periodic Noise

- Periodic noise : by electrical or electromechanical interference during image acquisition
- Can be reduced by frequency domain filtering

a
b

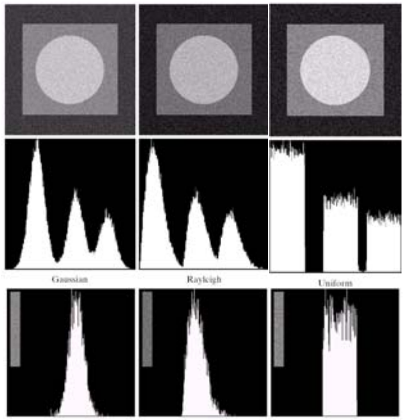
FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)



Periodic noise looks like dots In the frequency domain

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Estimation of Noise



We cannot use the image histogram to estimate noise PDF.

$$\mu = \sum_{z_i \in S} z_i p(z_i)$$

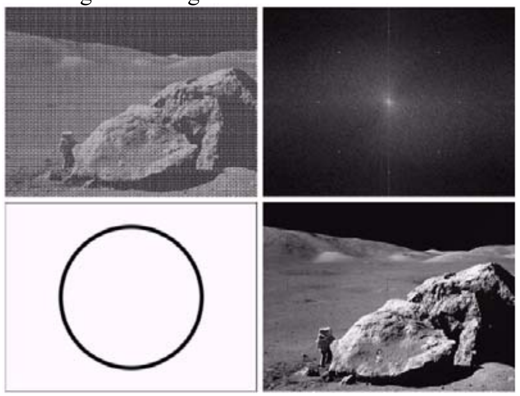
$$\sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 p(z_i)$$

It is better to use the histogram of one area of an image that has constant intensity to estimate noise PDF.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Periodic Noise Reduction by Freq. Domain Filtering

Degraded image | DFT



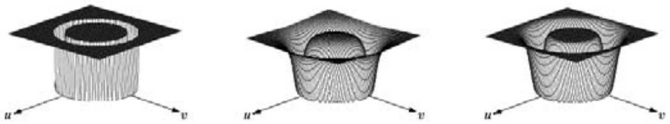
Band reject filter | Restored image

Periodic noise can be reduced by setting frequency components corresponding to noise to zero.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Band Reject Filters

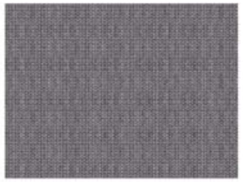
Use to eliminate frequency components in some bands



a b c

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

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

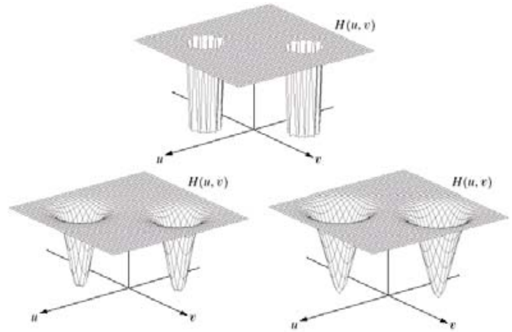


Periodic noise from the previous slide that is Filtered out.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Notch Reject Filters

A notch reject filter is used to eliminate some frequency components.



a
b c

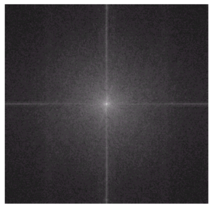
FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

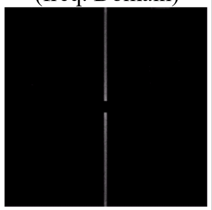
Notch Reject Filter:

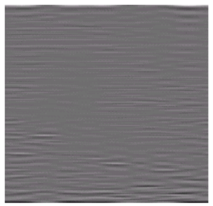
Degraded image

DFT

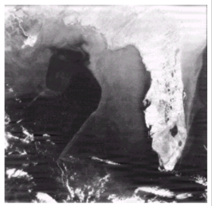


Notch filter (freq. Domain)





Noise



Restored image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Image Degraded by Periodic Noise

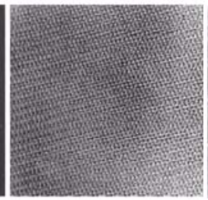
Degraded image

a b

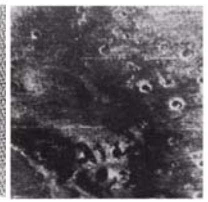
FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6*. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

DFT (no shift)

DFT of noise



Noise



Restored image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Mean Filters

Degradation model:

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

To remove this part

Arithmetic mean filter or moving average filter (from Chapter 3)

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

Geometric mean filter

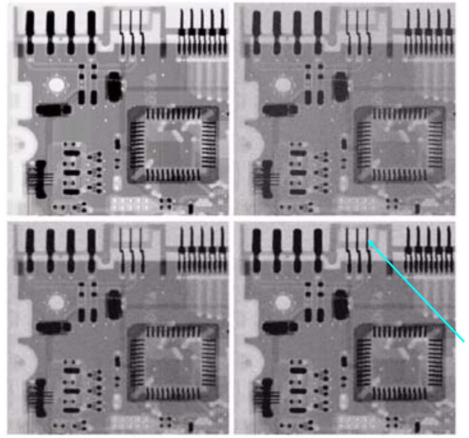
$$\hat{f}(x,y) = \left(\prod_{(s,t)\in S_{xy}} g(s,t) \right)^{\frac{1}{mn}}$$

- mn = size of moving window
- Achieves smoothing, lose less image detail

Geometric Mean Filter: Example

Original image

Image corrupted by AWGN



By 3x3 ***arithmetic*** mean filter

By 3x3 ***geometric*** mean filter :less blurring

AWGN: Additive White Gaussian Noise

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Harmonic and Contraharmonic Filters

Harmonic mean filter

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

Works well for salt noise but fails for pepper noise

Contraharmonic mean filter

mn = size of moving window
Q = order of the filter

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

Positive Q is suitable for eliminating pepper noise.
Negative Q is suitable for eliminating salt noise.

For Q = 0, the filter reduces to an arithmetic mean filter.
For Q = -1, the filter reduces to a harmonic mean filter.

Contraharmonic Filters: Example

Image corrupted by *pepper noise* with prob. = 0.1

Image corrupted by *salt No`ise* with prob. = 0.1

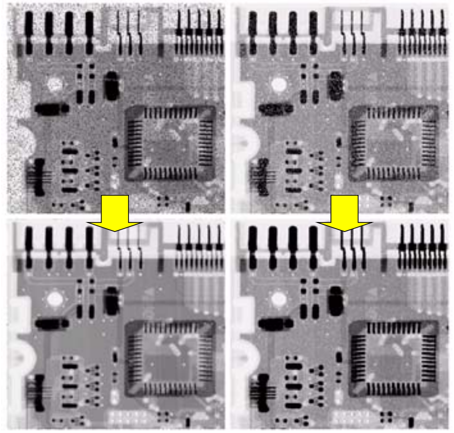


Image obtained using a 3x3 *contra-harmonic* mean filter With **Q = 1.5**

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

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Contraharmonic Filters: Incorrect Use Example

Image corrupted by pepper noise with prob. = 0.1

Image corrupted by salt noise with prob. = 0.1

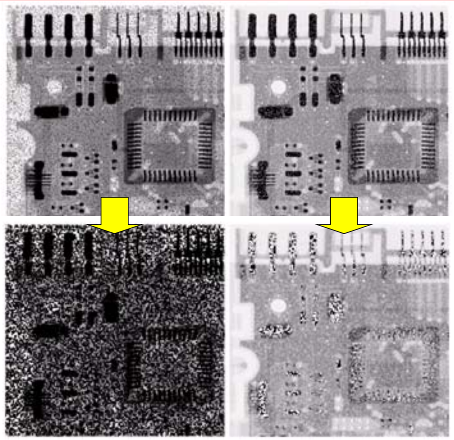
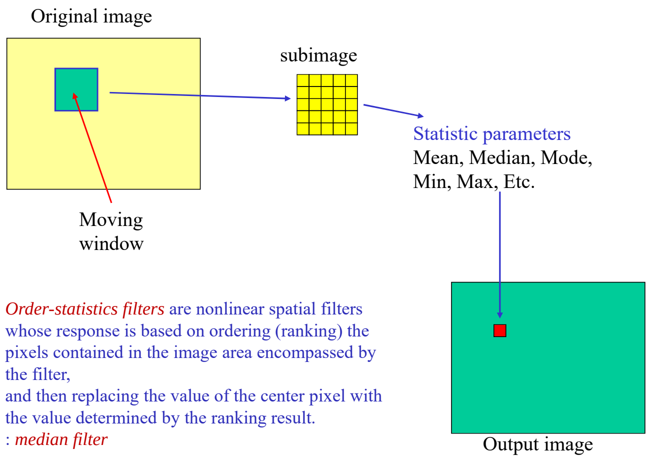


Image obtained using a 3x3 contra-harmonic mean filter With Q= -1.5

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

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Order-Statistic Filters: Revisit



Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter,
and then replacing the value of the center pixel with the value determined by the ranking result.
: *median filter*

Order-Statistics Filters

Median filter

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

Max filter

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

Reduce "dark" noise (pepper noise)

Min filter

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

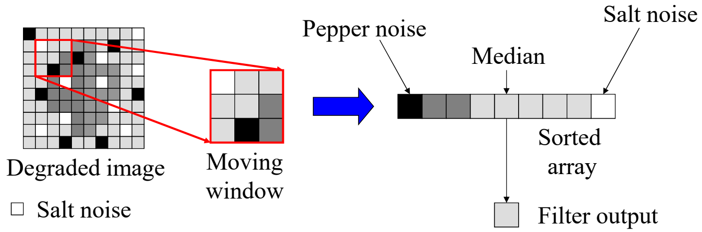
Reduce "bright" noise (salt noise)

Midpoint filter : order statistics + averaging, best for randomly distributed noise

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

Median Filter : How it works

A median filter is good for removing impulse, isolated noise



☐ Salt noise

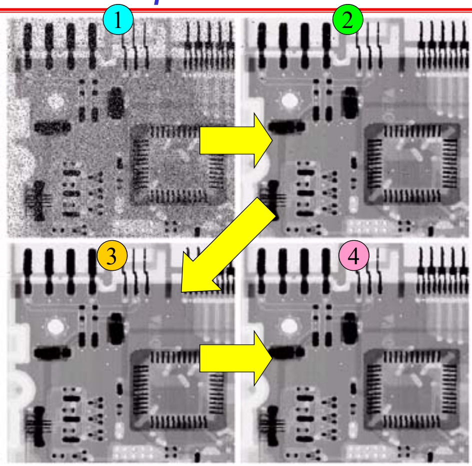
■ Pepper noise

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.

Median Filter : Example

Image corrupted by salt-and-pepper noise with $p_a = p_b = 0.1$



Images obtained using a 3x3 median filter

Max and Min Filters: Example

Image corrupted by pepper noise with prob. = 0.1

Image corrupted by salt noise with prob. = 0.1

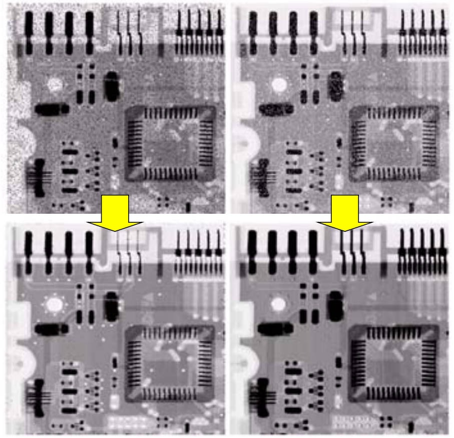


Image obtained using a 3x3 ***max filter***

Image obtained using a 3x3 ***min filter***

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Alpha-trimmed Mean Filter

Formula:

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

where $g_r(s,t)$ represent the remaining *mn-d* pixels after removing the $d/2$ highest and $d/2$ lowest values of $g(s,t)$.
If $d=0$, reduces to *arithmetic mean filter*

This filter is useful in situations involving multiple types of noise such as *a combination of salt-and-pepper and Gaussian noise*.

Alpha-trimmed Mean Filter: Example

Image corrupted by additive uniform noise

Image additionally corrupted by additive salt-and-pepper noise

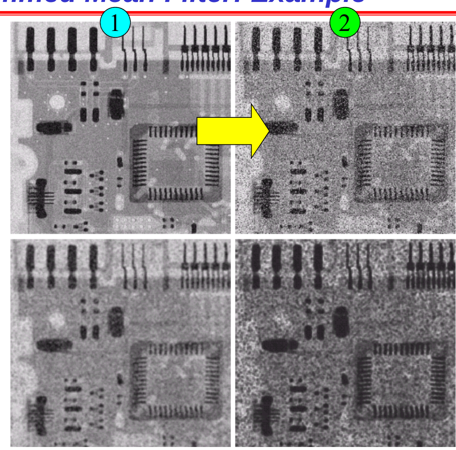


Image 2 obtained using a 5x5 ***arithmetic mean filter***

Image 2 obtained using a 5x5 ***geometric mean filter***

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Alpha-trimmed Mean Filter: Example (cont.)

Image corrupted by additive uniform noise

Image additionally corrupted by additive salt-and-pepper noise

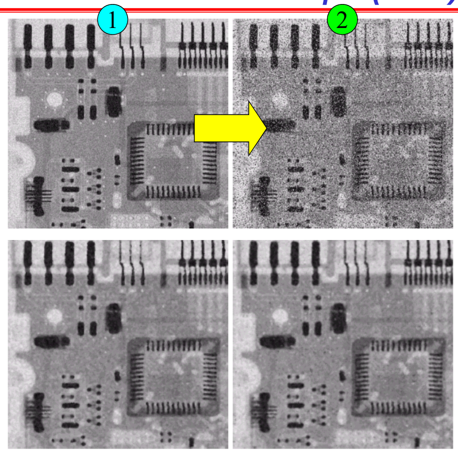


Image obtained using a 5x5 ***median filter*** (2)

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

Alpha-trimmed Mean Filter: Example (cont.)

Image obtained using a 5x5 arithmetic mean filter

Image obtained using a 5x5 geometric mean filter

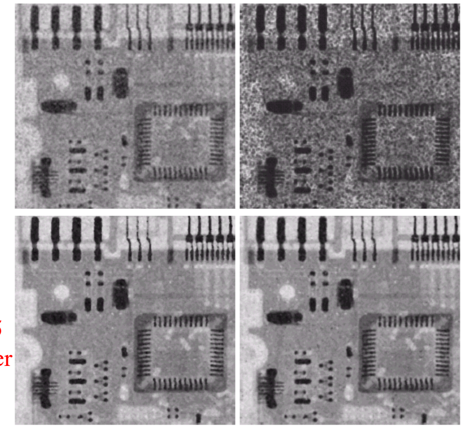


Image obtained using a 5x5 median filter

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

Adaptive Filter

General concept:

- Filter behavior changes based on statistical characteristics of local areas inside *mxn* moving window
- More complex but superior performance compared with "fixed" filters

Statistical characteristics:

Local mean:

$$m_L = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

Noise variance:

$$\sigma_\eta^2$$

Local variance:

$$\sigma_L^2 = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} (g(s,t) - m_L)^2$$

Adaptive, Local Noise Reduction Filter

Purpose: want to preserve edges

Concept:

1. If σ_η^2 is zero, → No noise
   the filter should return $g(x,y)$ because $g(x,y) = f(x,y)$
2. If σ_L^2 is high relative to σ_η^2, → Edges (should be preserved),
   the filter should return the value close to $g(x,y)$
3. If $\sigma_L^2 = \sigma_\eta^2$, → Areas inside objects
   the filter should return the arithmetic mean value m_L

Formula:

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

Adaptive Noise Reduction Filter: Example

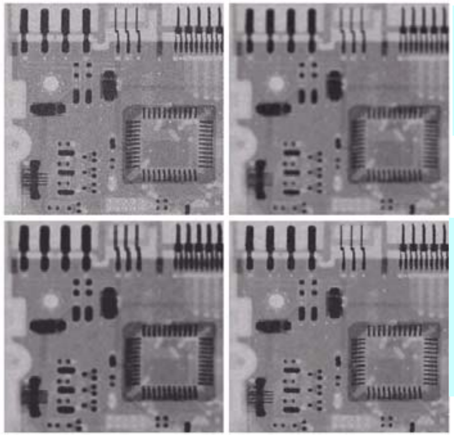


Image corrupted by additive ***Gaussian noise*** with zero mean and σ^2=1000

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***

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Adaptive Median Filter

Purpose: want to remove impulse noise while preserving edges

1) Impulse noise reduction
2) Other noises reduction : smoothing
3) Edge preserving

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: How it works

Level *A*: $A1 = z_{median} - z_{min}$
$A2 = z_{median} - z_{max}$

Determine whether z_{median} is an impulse or not

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}

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

$B1 = z_{xy} - z_{min}$
$B2 = z_{xy} - z_{max}$

Determine whether z_{xy} is an impulse or not

If B1 > 0 and B2 < 0, → z_{xy} is not an impulse

return z_{xy} → to preserve original details

Else

return z_{median} → to remove impulse

Adaptive Median Filter: Example

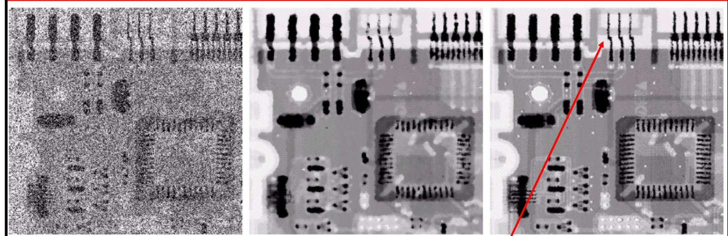


Image corrupted by salt-and-pepper noise with $p_a = p_b = 0.25$

Image obtained using a 7x7 median filter

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

More small details are preserved

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

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 exact $h(x,y)$, regardless of noise, we can do deconvolution to get $f(x,y)$ back from $g(x,y)$.

Methods:

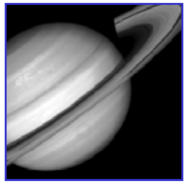
1. Estimation by Image Observation
2. Estimation by Experiment
3. Estimation by Modeling

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Estimation by Image Observation

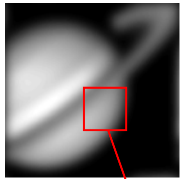
Original image (unknown)

$f(x,y)$



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

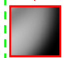
Degraded image



$g(x,y)$

Observation

Subimage



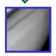
$g_s(x,y)$

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

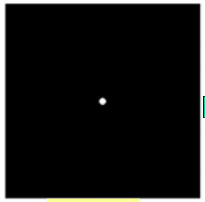
$$H(u,v) \approx H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

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

Estimation by Experiment

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

Input impulse image



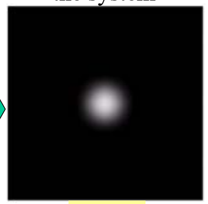
$A\delta(x,y)$

DFT

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

System
H( )

Response image from the system



$g(x,y)$

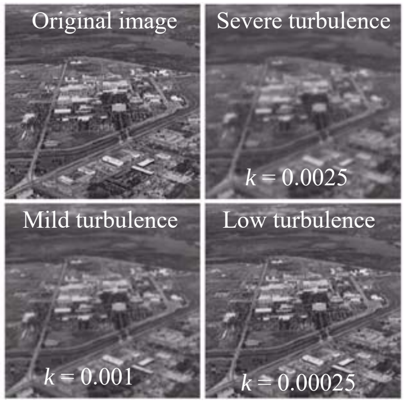
DFT

$G(u,v)$

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

Estimation by Modeling

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



Example:

Atmospheric Turbulence model

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

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Estimation by Modeling: Motion Blurring

Assume that camera velocity is $(x_0(t), y_0(t))$

The blurred image is obtained by

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

where T = exposure time.

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f(x - x_0(t), y - y_0(t))dt\right]e^{-j2\pi(ux+vy)}dxdy$$
$$= \int_0^T\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x - x_0(t), y - y_0(t))e^{-j2\pi(ux+vy)}dxdy\right]dt$$

Estimation by Modeling: Motion Blurring (cont.)

$$G(u,v) = \int_0^T\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x - x_0(t), y - y_0(t))e^{-j2\pi(ux+vy)}dxdy\right]dt$$
$$= \int_0^T\left[F(u,v)e^{-j2\pi(ux_0(t)+vy_0(t))}\right]dt$$
$$= F(u,v)\int_0^T e^{-j2\pi(ux_0(t)+vy_0(t))}dt$$

Then we get, the motion blurring transfer function:

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

For constant motion $(x_0(t), y_0(t)) = (at, bt)$

$$H(u,v) = \int_0^T e^{-j2\pi(ua+vb)}dt = \frac{T}{\pi(ua+vb)}\sin(\pi(ua+vb))e^{-j\pi(ua+vb)}$$

Motion Blurring Example

For constant motion

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

Original image

Motion blurred image
$a = b = 0.1$, $T = 1$

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

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:

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

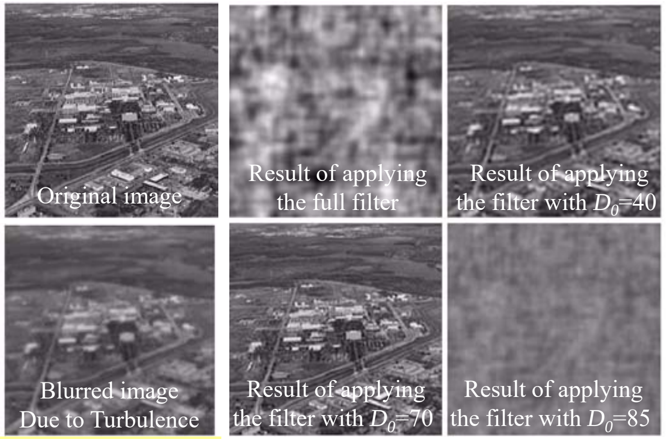
Noise is enhanced
when $H(u,v)$ is small.

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

In practical, the inverse filter is not popularly used.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

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: minimize mean square error: $e^2 = E\{(f - \hat{f})^2\}$

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
$S_\eta(u,v) = |N(u,v)|^2$ = Power spectrum of noise
$S_f(u,v) = |F(u,v)|^2$ = Power spectrum of the undegraded image
$|H(u,v)|^2 = H^*(u,v)\, H(u,v)$

el C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Approximation of Wiener Filter

Wiener Filter Formula:

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

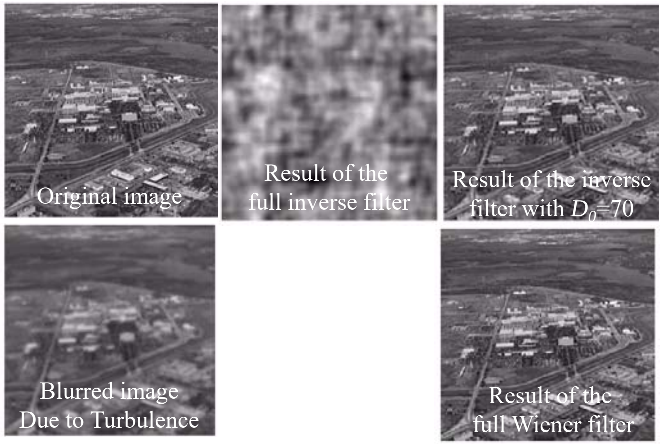
Difficult to estimate

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!
If K=0, what happens?

Wiener Filter: Example



Original image

Result of the full inverse filter

Result of the inverse filter with D_0=70

Blurred image Due to Turbulence

Result of the full Wiener filter

Example: Wiener Filter and Motion Blurring

Image degraded by motion blur + AWGN

Note: K is chosen manually

From Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Constrained Least Squares Filter

Assumptions on *Wiener filtering*:

- degradation function H is known
- *power spectra* of undegraded image and noise is known
- a constant estimate of the ratio of power spectra
- minimize a statistical criterion, is optimal in an average sense

Constrained Least Squares Filtering

- *needs only the mean and variance of the noise?*
- optimal result for each image

Constrained Least Squares Filter

Degradation model:

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

Written in a matrix form

$$\mathbf{g} = \mathbf{Hf} + \boldsymbol{\eta}$$

Restoration problem is reduced to simple matrix manipulations?

1) $\mathbf{g}$ = (MxN), $\mathbf{H}$ = (MNxMN), matrices are too big to handle
2) $\mathbf{H}$ is highly sensitive to noise

Noise sensitivity problem → measure of smoothness

Objective: to find the minimum of a criterion function

$$C = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\nabla^2 f(x,y)\right]^2$$

Subject to the constraint

$$\left\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\right\|^2 = \left\|\boldsymbol{\eta}\right\|^2 \qquad \text{where} \qquad \left\|\mathbf{w}\right\|^2 = \mathbf{w}^T\mathbf{w}$$

Constrained Least Squares Filter

The frequency domain solution to the problem:

a constrained least square filter

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

where

$$P(u,v) = \text{Fourier transform of } p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

If constant γ = 0, it reduces to ? IF.

Constrained Least Squares Filter: Example

Constrained least square filter

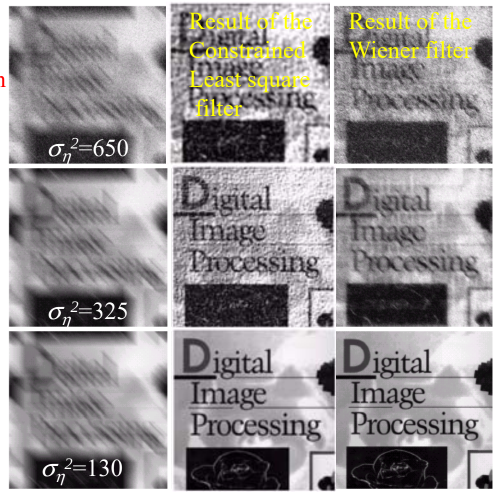
$$\hat{F}(u,v)=\left[\frac{H^*(u,v)}{|H(u,v)|^2+\gamma|P(u,v)|^2}\right]G(u,v)$$

γ is adaptively adjusted to achieve the best result.

Results from the previous slide obtained from the constrained least square filter

Constrained Least Squares Filter: Example (cont.)

Image degraded by motion blur + AWGN



Constrained Least Squares Filter:Adjusting γ

$\mathbf{r}=\mathbf{g}-\mathbf{H}\hat{\mathbf{f}}$ Monotonically increasing: $\phi(\gamma)=\mathbf{r}^T\mathbf{r}=\|\mathbf{r}\|^2$

We want to adjust gamma so that $\|\mathbf{r}\|^2=\|\boldsymbol{\eta}\|^2\pm a \longrightarrow$ 1

where a = accuracy factor

1. Specify an initial value of γ
2. Compute $\|\mathbf{r}\|^2$
3. Stop if 1 is satisfied
   Otherwise return step 2 after increasing γ if $\|\mathbf{r}\|^2<\|\boldsymbol{\eta}\|^2-a$
   or decreasing γ if $\|\mathbf{r}\|^2>\|\boldsymbol{\eta}\|^2+a$
   Use the new value of γ to recompute

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

Constrained Least Squares Filter:Adjusting γ(cont.)

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

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

$$\|\mathbf{r}\|^2=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}r^2(x,y)$$

For computing $\|\mathbf{r}\|^2$

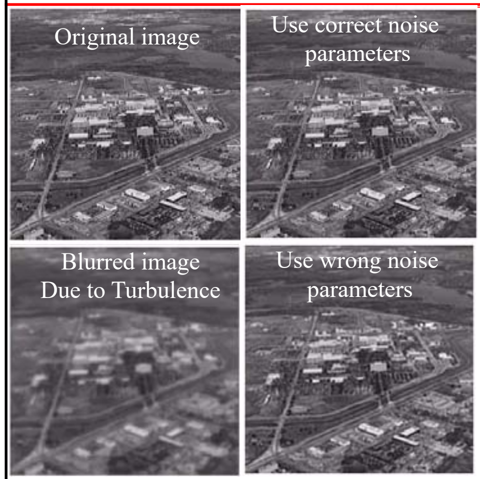
$$m_\eta=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\eta(x,y)$$

$$\sigma_\eta^2=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\eta(x,y)-m_\eta\right]^2$$

$$\|\boldsymbol{\eta}\|^2=MN\left[\sigma_\eta^2-m_\eta\right]$$

For computing $\|\boldsymbol{\eta}\|^2$

Constrained Least Squares Filter: Example



Correct parameters:
Initial $\gamma = 10^{-5}$
Correction factor = 10^{-6}
$a = 0.25$
$\sigma_\eta^2 = 10^{-5}$

Wrong noise parameter
$\sigma_\eta^2 = 10^{-2}$

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Results obtained from constrained least square filters

Geometric Mean filter

This filter represents a family of filters combined into a single expression

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

$\alpha = 1$ → the inverse filter
$\alpha = 0$ → the Parametric Wiener filter
$\alpha = 0$, $\beta = 1$ → the standard Wiener filter
$\beta = 1$, $\alpha < 0.5$ → More like the inverse filter
$\beta = 1$, $\alpha > 0.5$ → More like the Wiener filter

Another name: the spectrum equalization filter

Geometric Transformation

Geometric transformations modify the spatial relationships between pixels in an image.

These transformations are often called rubber-sheet transformations: Printing an image on a rubber sheet and then stretch this sheet according to some predefined set of rules.

A geometric transformation consists of 2 basic operations:

1. **spatial transformation :**
   rearrangement of pixels on the image plane
2. **gray level interpolation :**
   Assign gray level values to pixels in the spatially transformed image.

Geometric Transformation : Algorithm

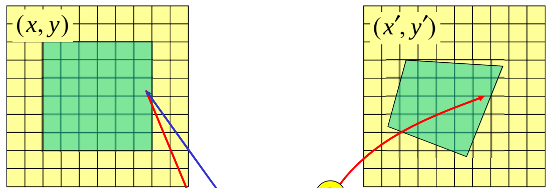


Image ***f*** to be restored

Distorted image ***g***

1. Select coordinate (x,y) in *f* to be restored
2. Compute
   $x' = r(x, y)$
   $y' = s(x, y)$

3. Go to pixel (x', y') in a distorted image *g*

4. get pixel value at $g(x', y')$ By gray level interpolation

5. store that value in pixel *f*(*x*,*y*)

Spatial Transformation

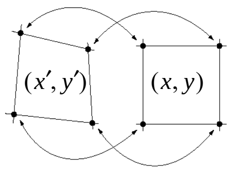
To map between pixel coordinate (x,y) of f and pixel coordinate (x',y') of g

$$x' = r(x,y) \qquad y' = s(x,y)$$

For a bilinear transformation mapping between a pair of Quadrilateral regions

$$x' = r(x,y) = c_1x + c_2y + c_3xy + c_4$$

$$y' = s(x,y) = c_5x + c_6y + c_7xy + c_8$$



To obtain $r(x,y)$ and $s(x,y)$, we need to know 4 pairs of coordinates (x,y) and its corresponding (x',y') which are called tiepoints.

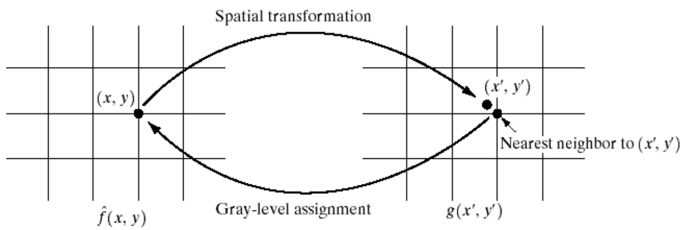
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Gray Level Interpolation: Nearest Neighbor

Since (x',y') may not be at an integer coordinate, we need to Interpolate the value of $g(x',y')$

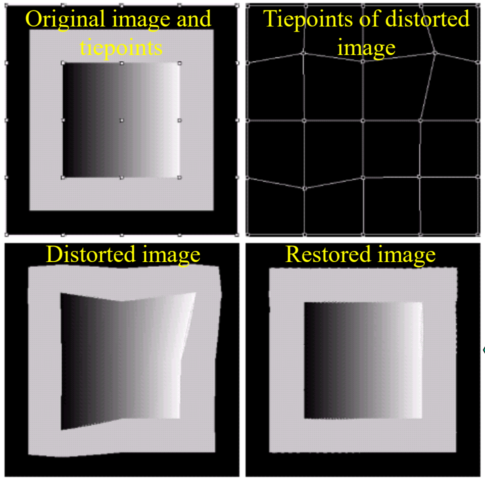
Example interpolation methods that can be used:

1. Nearest neighbor selection
2. Bilinear interpolation
3. Bicubic interpolation



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

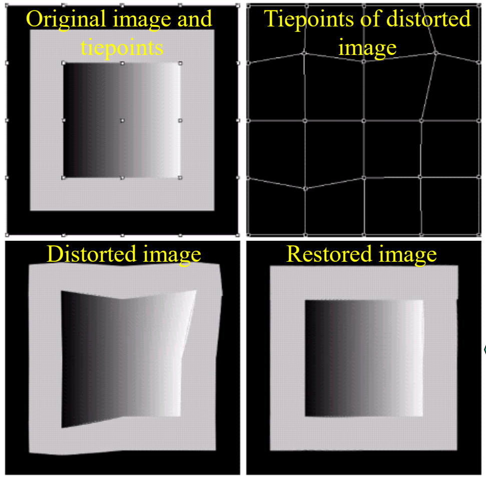
Geometric Distortion and Restoration Example



Use nearest neighbor intepolation

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

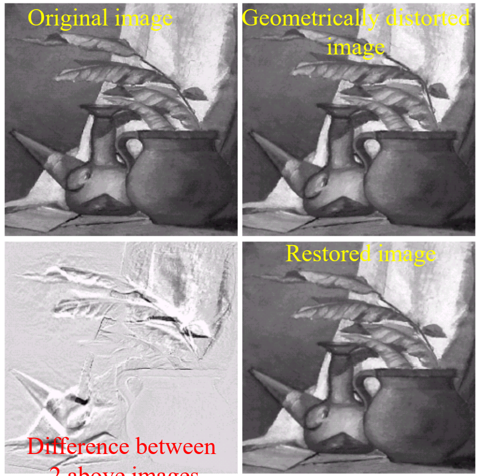
Geometric Distortion and Restoration Example (cont.)



Use bilinear intepolation

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Geometric Restoration



Use the same Spatial Trans. as in the previous example

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Homework – due next class

1. Code the noise generator of AWGN, uniform distribution noise, salt and pepper noise.
   1. For the image in the text book and Lena image, apply these noise generator to create noisy images, probably with different parameters.
   2. Restore the noisy images in your best. Display the results and explain what happened.
2. For the images in the text book(airplace images and the book cover with camera movement) and another sample, try the turbulance model to re-generate the images in the textbook. Explain the meaning of D_0 parameter and display other results with different parameters. What are the differences? Conclude your experiment.
3. Try Wiener filter to do the same thing. Present your point or message in the experiment with detailed explanation.