Digital Image Processing

Lecture 8
(Image Restoration)

Bu-Ali Sina University
Computer Engineering Dep.
Fall 2010
Image Restoration
Image restoration

Outputs of these processes generally are images

CHAPTER 6
Color image processing

CHAPTER 7
Wavelets and multi-resolution processing

CHAPTER 8
Compression

CHAPTER 9
Morphological processing

CHAPTER 5
Image restoration

CHAPTERS 3 & 4
Image enhancement

Knowledge base

CHAPTER 10
Segmentation

CHAPTER 11
Representation & description

CHAPTER 12
Object recognition

Problem domain

Outputs of these processes generally are image attributes
What is Image Restoration?

Image restoration attempts to restore images that have been degraded

- Identify the degradation process and attempt to reverse it
- Similar to image enhancement, but more objective
Image restoration

• **Restoration**
  
  ▪ A process that attempts to reconstruct or recover an image that has been degraded by using some prior knowledge of the degradation phenomenon.
  ▪ Involves modeling the degradation process and applying the inverse process to recover the original image.
  ▪ A criterion for “goodness” is required that will recover the image in an optimal fashion with respect to that criterion.
  ▪ Ex. Removal of blur by applying a deblurring function.

• **Enhancement**
  
  ▪ Manipulating an image in order to take advantage of the psychophysics of the human visual system.
  ▪ Techniques are usually “heuristic.”
  ▪ Ex. Contrast stretching, histogram equalization.
Problem:

- You want to obtain an image X.
- But you only have a degradation version Y.
- How do you determine X from Y?

Degradation may result from:

- Additive noise
- Nonadditive noise
- Linear distortion
- Nonlinear distortion
Image Restoration

- Restoration:
  A process that attempts to reconstruct or recover a degraded image by using some a priori knowledge of the degradation phenomenon.

- Technique:
  - Model the degradation
  - Apply the inverse process to recover the original image.
Degradation Model

\[ f(x, y) + g(x, y) = \hat{f}(x, y) \]

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

- Noise Origin:
  - Image sensor might produce noise because of environmental conditions or quality of sensing elements.
  - Interference in the image transmission channel.

- Assumptions:
  noise is independent of spatial coordinates (except for periodic noise) and independent of the image.

- Spatial description of noise: Gaussian noise, Rayleigh noise, Erlang (Gamma) noise, Exponential noise, Uniform noise, Impulse noise, etc.
Noise Model

We can consider a noisy image to be modelled as follows:

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

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

If we can estimate the model, the noise in an image is based on this will help us to figure out how to restore the image.
Gaussian noise

- Mathematically speaking, it is the most tractable noise model.
- Due to factors such as electronic circuit noise, sensor noise (due to poor illumination or high temperature)

- The pdf of a Gaussian random variable $z$ is given by:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(z - \mu)^2}{2\sigma^2}\right)$$

where $z$ represents (noise) gray value, $\mu$ is the mean, and $\sigma$ is its standard deviation. The squared standard deviation $\sigma^2$ is usually referred to as variance.
• For a Gaussian pdf, approximately 70% of its values will be in the range $[\mu - \sigma, \mu + \sigma]$, and 95% of its values will be in the range $[\mu - 2\sigma, \mu + 2\sigma]$.
Rayleigh noise

- The pdf of a Rayleigh noise is given by:

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

- The mean and variance are given by:

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

- model noise in range imaging
Erlang(Gamma) noise

- The pdf of Erlang noise is given by:

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

where, \( a > 0 \), \( b \) is an integer and “!” represents factorial.

- The mean and variance are given by:

\[
\mu = \frac{b}{a}
\]

\[
\sigma^2 = \frac{b}{a^2}
\]

- Model Noise in Laser Imaging.
Exponential noise

• The pdf of exponential noise is given by:

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

where, \( a > 0 \).

• The mean and variance are given by:

\[ \mu = 1/a \]

\[ \sigma^2 = 1/a^2 \]

• This is a special case of Erlang density with \( b=1 \).
The pdf of uniform noise is given by:

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

The mean and variance are given by:

\[ \mu = \frac{(a+b)}{2} \]
\[ \sigma^2 = \frac{(b-a)^2}{12} \]
The pdf of (bipolar) impulse noise is given by:

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

Where \( a, b > 0 \)

- found in quick transients
Noise Models

**FIGURE 5.2** Some important probability density functions.
**Noise Models**

**FIGURE 5.4** Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.
Noise Models

**Figure 5.4 (Continued)** Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.
Estimation of Noise Parameters

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

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

\[ \mu = \frac{1}{MN} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} g(x, y) \]

\[ \sigma^2 = \frac{1}{MN} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} (g(x, y) - \mu)^2 \]

**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.
When the only degradation is noise:

\[ g(x,y) = f(x,y) + n(x,y) \]

\[ G(u,v) = F(u,v) + N(u,v) \]

Spatial filtering is the method of choice in this case: Mean filters, Order-statistics filters, Adaptive filters
Mean Filter

- $S_{xy}$: subimage of size $m \times n$
- Arithmetic mean filter:
  \[
  \hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)
  \]
- Geometric mean filter: it tends to lose less image details in the process
  \[
  \hat{f}(x, y) = \left| \prod_{(s, t) \in S_{xy}} g(s, t) \right|^{1/mn}
  \]
Mean Filter

- Harmonic mean filter: works well for salt noise and fails for pepper

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

- Contraharmonic mean filter: Positive Q (order of filter) for pepper and negative Q for Salt Noise

\[ \hat{f}(x, y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^Q} \]
**Mean Filter**

**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 $3 \times 3$. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)
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 $3 \times 3$ contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$. 
Mean Filter

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 \( Q = 1.5 \).
Order-Statistics Filter

- **Median Filter**
  Effective for salt and pepper noise

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

- **Max and Min filters**

\[
\hat{f}(x, y) = \min\{g(s, t)\} \quad \hat{f}(x, y) = \max\{g(s, t)\}
\]
\[(s,t) \in S_{xy}\]

- Max filter: useful for finding brightest points in an image (remove pepper noise)
- Min filter: useful for finding darkest points in an image (remove salt noise)
Order-Statistics Filter

- **Midpoint Filter**
  Works best for Gaussian and Uniform noise
  
  \[ \hat{f}(x, y) = \frac{1}{2} \left[ \min_{(s,t)\in S_{xy}} \{g(s, t)\} + \max_{(s,t)\in S_{xy}} \{g(s, t)\} \right] \]

- **Alpha-trimmed mean filter**
  
  \[ \hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t)\in S_{xy}} g_r(s, t) \]
  
  - Useful for combination of Salt-pepper and Gaussian noise
Order-Statistics Filter

**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 \( \times \) 3.
(c) Result of processing (b) with this filter.
(d) Result of processing (c) with the same filter.
**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 $5 \times 5$: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$. 
Response of the filter is based on four quantities:

1. \( g(x, y) \)
2. \( \sigma^2_\eta \) : variance of noise
3. \( m_L \) : mean of pixels in \( S_{xy} \)
4. \( \sigma^2_L \) : variance of pixels in \( S_{xy} \)

\[
\hat{f}(x, y) = g(x, y) - \frac{\sigma^2_\eta}{\sigma^2_L} (g(x, y) - m_L)
\]
This filter does the following:

1. If $\sigma_n^2 = 0$ (or is small), the filter simply returns the value of $g(x,y)$.
2. If the local variance $\sigma_L^2(x,y)$ is high relative to the noise variance the filter returns a value close to $g(x,y)$. This usually corresponds to a location associated with edges in the image.
3. If the two variances are roughly equal, the filter does a simple averaging over window $S_{xy}$.

Only the variance of overall noise has to be estimated.
Adaptive, local noise reduction filter

**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 $7 \times 7$. 
Adaptive median filtering

- Handle dense impulse noise
- Smoothes non-impulse noise
- Preserves details
  - $z_{\text{min}}$: minimum gray level in $S_{xy}$
  - $z_{\text{max}}$: maximum gray level in $S_{xy}$
  - $z_{\text{med}}$: median gray level of $S_{xy}$
  - $z_{xy}$: gray level at coordinate $(x, y)$
  - $S_{\text{max}}$: maximum allowed size of $S_{xy}$
Adaptive median filtering

- A
  - $A_1 = z_{med} - z_{min}$
  - $A_2 = z_{med} - z_{max}$
  - If $A_1 > 0$ and $A_2 < 0$ go to B else increase the window size
  - If window size $< S_{max}$ repeat A
  - Else output $z_{xy}$

- B
  - $B_1 = z_{xy} - z_{min}$
  - $B_2 = z_{xy} - z_{max}$
  - If $B_1 > 0$ and $B_2 < 0$ output $z(x,y)$
  - Else output $z_{med}$
With these observations in mind, we see that the purpose of level A is to determine if the median filter output, $z_{\text{med}}$, is an impulse (black or white) or not. If the condition $z_{\text{min}} < z_{\text{med}} < z_{\text{max}}$ holds, then $z_{\text{med}}$ cannot be an impulse for the reason mentioned in the previous paragraph. In this case, we go to level B and test to see if the point in the center of the window, $z_{xy}$, is itself an impulse (recall that $z_{xy}$ is the point being processed). If the condition $B1 > 0$ AND $B2 < 0$ is true, then $z_{\text{min}} < z_{xy} < z_{\text{max}}$, and $z_{xy}$ cannot be an impulse for the same reason that $z_{\text{med}}$ was not. In this case, the algorithm outputs the unchanged pixel value, $z_{xy}$. By not changing these “intermediate-level” points, distortion is reduced in the image. If the condition $B1 > 0$ AND $B2 < 0$ is false, then either $z_{xy} = z_{\text{min}}$ or $z_{xy} = z_{\text{max}}$. In either case, the value of the pixel is an extreme value and the algorithm outputs the median value $z_{\text{med}}$, which we know from level A is not a noise impulse. The last step is what the standard median filter does. The problem is that the standard median filter replaces every point in the image by the median of the corresponding neighborhood. This causes unnecessary loss of detail.
Adaptive median filtering

**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 $7 \times 7$ median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$. 
Periodic noise reduction

- This type of noise can be very effectively removed using frequency domain filtering.
- Bandreject filters remove or attenuate a band of frequencies around the central frequency, say $D_0$.
- An ideal bandreject filter is given by:

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

Where $W$: width of the (stop) band
$D_0$: central frequency.

$$D(u, v) = \sqrt{u^2 + v^2}$$
Butterworth bandreject filter of order $n$ is given by

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

Gaussian bandreject filter is given by

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

**FIGURE 5.15** From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.
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.)
Bandpass filters are the exact opposite of bandreject filters. They pass a band of frequencies, around some frequency, say $D_0$ (rejecting the rest).

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

Bandpass filter is usually used to isolate components of an image that correspond to a band of frequencies.

It can also be used to isolate noise interference, so that more detailed analysis of the interference can be performed, independent of the image.
It is a kind of bandreject/bandpass filter that rejects/passes a very narrow set of frequencies, around a center frequency.

Due to symmetry considerations, the notches must occur in symmetric pairs about the origin of the frequency plane.

The transfer function of an ideal notch-reject filter of radius $D_0$ with center frequency $(u_0, v_0)$ and $(-u_0, -v_0)$ is given by

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

where

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

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

- Butterworth notch reject filter of order $n$

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

- Gaussian notch reject filter is given by

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

- A notch pass filter can be obtained from a notch reject filter using:

\[ H_{np}(u, v) = 1 - H_{nr}(u, v) \]
Notch filter

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

- Scan line noise removing
(a) Satellite image of Florida and the Gulf of Mexico
(b) Spectrum of (a)
(c) Notch pass filter shown superimposed on (b)
(d) Inverse Fourier transform of filtered image
(e) Result of notch reject filtering

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)
Linear position-invariant degradation

- **Degradation Model**

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

- **Linear Degradation Model**

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

- **Linear position-invariant Degradation Model**

  \[ g(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) \, d\alpha \, d\beta + \eta(x, y) \]
Linear position-invariant degradation

\[ g(x,y) = h(x,y) \ast f(x,y) + \eta(x,y) \]

\[ G(u,v) = H(u,v)F(u,v) + N(u,v) \]

- The function \( h(x,y) \) is Impulse response
- In Optic, It is also referred to as a point-spread function (PSF) and represents the observed image corresponding to point source of light.

- To restore the original image we need to have
  1. The knowledge of the PSF \( h(x,y) \)
  2. The noise function \( \eta(x,y) \) statistics.
Estimation of degradation function

- Estimation by Image observation
  - Identify a relatively noise-free subimage of the observed image, containing simple structures (e.g., part of an object and the background)

  - Construct an unblurred image of the subimage by using sample gray levels of the object and background.

  - Calculate

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

    $G_s(u,v)$: the spectrum of the observed subimage,
    $F_s(u,v)$: the estimate of the spectrum of the original image

  - Based on the characteristic of the function $H_s(u,v)$, we can rescale to obtain the overall PSF $H(u,v)$. 
Estimation of degradation function

Estimation by Experimentation

If equipment similar to the equipment used to acquire the degraded image is available it is possible to obtain an accurate estimate of the degradation.

The idea is to obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the system

\[ H(u, v) = \frac{G(u, v)}{A} \]
Estimation of degradation function

- Estimation by Experimentation

**FIGURE 5.24** Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.
Geometrical transformation consists of two basic operations:

1. Spatial transformation: defines the rearrangement of pixels on the image plane
2. Gray level interpolation: deals with the assignment of gray levels to pixels in the spatially transformed image
Image f with pixels coordinates \((x,y)\) has undergone geometric distortion to produce an image g with coordinates \((x',y')\)

\[ x' = r(x,y), \quad y' = s(x,y) \]

Example: \(x' = r(x,y) = x/2, \quad y' = s(x,y) = y/2\)

- Distortion is a shrinking of the size of \(f(x,y)\) by one-half in both directions.
If \( r(x,y) \) and \( s(x,y) \) are known analytically: the inverse of \( r \) and \( s \) is applied to \( g(x',y') \) to recover \( f(x,y) \).

In practice finding a single set of \( r(x,y) \) and \( s(x,y) \) is not possible.

Solution: spatial relocation is formulated by the use of tiepoints.

Tiepoints: a set of pixels whose locations in distorted and corrected images are known.
Spatial Transformation

- Suppose the geometrical distortion process within the region is modeled by a pair of bilinear equations:
  - \( x' = r(x, y) = c_1x + c_2y + c_3yx + c_4 \)
  - \( y' = s(x, y) = c_5x + c_6y + c_7yx + c_8 \)
- 8 known tiepoints, 8 unknown \( c_i \)
- The model is used for all the points inside the region

**FIGURE 5.32**
Corresponding tiepoints in two image segments.
Spatial Transformation

for \( x = 1 \) to horizontal size {
    for \( y = 1 \) to vertical size {
        \( x' = r(x,y) = c_1 x + c_2 y + c_3 yx + c_4 \)
        \( y' = s(x,y) = c_5 x + c_6 y + c_7 yx + c_8 \)
        \( f^*(x,y) = g(x',y') \)
    }
}

Gray-level Interpolation

- Depending on the values of $c_i$, $x'$ and/or $y'$ can be noninteger for integer values of $(x,y)$
  - $x' = r(x,y) = c_1 x + c_2 y + c_3 x y + c_4$
  - $y' = s(x,y) = c_5 x + c_6 y + c_7 x y + c_8$
- $g$ is a digital image and its pixel values are defined only at integer values of $(x,y)$.
- We need inferring gray-level values at noninteger locations (gray-level interpolation)
Gray-level Interpolation

- Simplest scheme: nearest neighbor approach (zero-order interpolation)
  1. Mapping (x,y) to (x’,y’)
  2. Selection of closest integer coordinate neighbor to (x’,y’)
  3. Assign the gray-level of this nearest neighbor to the pixel at (x,y)

**FIGURE 5.33** Gray-level interpolation based on the nearest neighbor concept.
FIGURE 5.34 (a) Image showing tiepoints. (b) Tiepoints after geometric distortion. (c) Geometrically distorted image, using nearest neighbor interpolation. (d) Restored result. (e) Image distorted using bilinear interpolation. (f) Restored image.