Digital Image Processing

Lecture # 6
Image Restoration
Image Restoration
WHAT IS IMAGE RESTORATION?

• The purpose of image restoration is to restore a degraded/distorted image to its original content and quality
• Restoration attempts to reconstruct or recover an image that has been degraded by using a priori knowledge of the degradation phenomenon
• Restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image
• Image enhancement is largely a subjective process, while image restoration is for the most part an objective process
WHAT IS IMAGE RESTORATION?

- Image enhancement: “improve” an image subjectively.
- Image restoration: remove distortion from image in order to go back to the “original” → objective process.
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
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
Noise and Images

The sources of noise in digital images arise during image acquisition (digitization) and transmission

- Imaging sensors can be affected by ambient conditions
- Interference can be added to an image during transmission
Noise Modeling for General Cases

Variant Acquisition Devices

Noisy Images → Noise Model → Image Processing → Output Images

It is hard to construct a general noise model!
An Example of Application: Sharpness Enhancement

Original Image  Enhanced Image  Enhanced Image with Noise Suppression
Application with Noise Model

\[ \tilde{\sigma}_n(y_1) \]

\[ \tilde{\sigma}_n(y_2) \]
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.
Noise Models

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

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

The test pattern to the right is ideal for demonstrating the addition of noise.
The following slides will show the result of adding noise based on various models to this image.
Noise Example (cont...)
Noise Example (cont...)

Exponential

Uniform

Impulse
Filtering to Remove Noise

We can use spatial filters of different kinds to remove different kinds of noise.

The *arithmetic mean* filter is a very simple one and is calculated as follows:

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

This is implemented as the simple smoothing filter.

Blurs the image to remove noise.

<table>
<thead>
<tr>
<th>1/9</th>
<th>1/9</th>
<th>1/9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
</tbody>
</table>
Noise Removal Examples

Original Image

Image Corrupted By Gaussian Noise

After A 3*3 Arithmetic Mean Filter

After A 3*3 Geometric Mean Filter
Order Statistics Filters

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

Useful spatial filters include

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

Median Filter:
\[
\hat{f}(x, y) = \text{median}\{ g(s, t) \}
\]

\[
(s, t) \in S_{xy}
\]

Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters

Particularly good when salt and pepper noise is present
Examples

- A 4x4 grayscale image is given by

\[
\begin{array}{cccc}
5 & 6 & 7 & 8 \\
0 & 6 & 7 & 8 \\
5 & 6 & 15 & 8 \\
5 & 6 & 7 & 8 \\
\end{array}
\]

1) Filter the image with a 3x3 median filter, after zero-padding

\[
\begin{array}{cccc}
5 & 6 & 7 & 8 \\
0 & 6 & 7 & 8 \\
5 & 6 & 15 & 8 \\
5 & 6 & 7 & 8 \\
\end{array}
\]

Zero-padding

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 5 & 6 & 7 & 8 & 0 \\
0 & 0 & 6 & 7 & 8 & 0 \\
0 & 5 & 6 & 15 & 8 & 0 \\
0 & 5 & 6 & 7 & 8 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Median filtering

\[
\begin{array}{cccccc}
0 & 5 & 6 & 0 \\
5 & 6 & 7 & 7 \\
5 & 6 & 7 & 7 \\
0 & 5 & 6 & 0 \\
\end{array}
\]
2) Filter the image with a 3x3 median filter, after replicate-padding at the image borders
Max and Min Filter

**Max Filter:**

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

**Min Filter:**

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

Max filter is good for pepper noise and min is good for salt noise
Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

\[ \hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} g_r(s, t) \]
Alpha-Trimmed Mean Filter

- Alpha-Trimmed Mean Filter:
  \[
  \hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s, t) \in S_{xy}} g_r(s, t)
  \]

- We can delete the \( d/2 \) lowest and \( d/2 \) highest grey levels
- So \( g_r(s, t) \) represents the remaining \( mn - d \) pixels
- If \( d = 0 \), the filter is reduced to arithmetic mean
- If \( d = mn - 1 \), the filter become median filter
- For other values, the filter is useful in situation involving multiple types of noise
  - Combination of salt-and-pepper and Gaussian noise
Noise Removal Examples

Image Corrupted By Salt And Pepper Noise

Result of 1 Pass With A 3*3 Median Filter

Result of 2 Passes With A 3*3 Median Filter

Result of 3 Passes With A 3*3 Median Filter
Noise Removal Examples (cont...)

Image Corrupted By Pepper Noise

Result Of Filtering Above With A 3*3 Max Filter

Image Corrupted By Salt Noise

Result Of Filtering Above With A 3*3 Min Filter
Noise Removal Examples (cont...)

Image Corrupted By Uniform Noise

Filtered By 5*5 Arithmetic Mean Filter

Filtered By 5*5 Median Filter

Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Alpha-Trimmed Mean Filter

Image Further Corrupted By Salt and Pepper Noise

Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Alpha-Trimmed Mean Filter
Order Filters

Minimum Filter

Image with salt noise
Probability = .04

Result of minimum filtering
Mask 3 x 3
Order Filters

Minimum filtering
Mask 5 x 5

Minimum filtering
Mask 9 x 9
Order Filters

Maximum Filter

Image with pepper noise
Probability = .04

Maximum filtering
Mask 3 x 3
Order Filters

Maximum filtering
Mask 5 x 5

Maximum filtering
Mask 9 x 9
Order Filters

• Order filters can also be defined to select a specific pixel rank within the ordered set.
  – For example, we may find the second highest value is the better choice than the maximum value for certain pepper noise.
  – This type of ordered selection is application specific.

• Minimum filter tend to darken the image and maximum filter tend to brighten the image.
Order Filters

• Midpoint filter:
  – Average of the maximum and minimum within the window.
  – Useful for removing gaussian and uniform noise.

\[
\text{Midpoint} = \frac{I_1 + I_{N^2}}{2}
\]
Order Filters

Image with gaussian noise. Variance = 300, mean = 0

Result of midpoint filter
Mask size = 3
Order Filters

Image with uniform noise. Variance = 300, mean = 0

Result of midpoint filter
Mask size = 3
Order Filters

• Alpha-trimmed mean filter:
  – The average of the pixel values within the window, but with some endpoint-ranked values excluded.

  \[
  \text{Alpha-trimmed mean} = \frac{1}{N^2 - 2T} \sum_{i=T+1}^{N^2-T} I_i
  \]

  – T is the number of pixels excluded at each end of the ordered set.
Order Filters

• The alpha-trimmed mean filter ranges from a mean to median filter, depending on the value selected for the T parameter.
  – If $T = 0$, $\rightarrow$ mean filter.
  – If $T = (N^2 - 1) / 2$, $\rightarrow$ median filter.

• The alpha-trimmed mean filter is useful for images containing multiple types of noise.
  – Example: Gaussian + salt-and-pepper.
Order Filters

Result of alpha-trimmed mean filter
Mask size = 3
Trim size = 1

Result of alpha-trimmed mean filter
Mask size = 3
Trim size = 4
Mean Filters

• The mean filters function by finding some form of an average within the $N\times N$ window.
• The most basic of these filters is the arithmetic mean filter.
  – This filter mitigates the noise effect, but at the same time tend to blur the image.
  – The blurring effect is not desirable, and therefore other mean filters are designed to minimize this loss of detail information.
Mean Filters

• Arithmetic mean filter:
  – Find the arithmetic average of the pixel values in the window.
  – Smooth out local variations in an image.
  – Tend to blur the image.
  – Works best with gaussian and uniform noise.

\[
\text{Arithmetic Mean} = \frac{1}{N^2} \sum_{(r,c) \in w} d(r, c)
\]
Mean Filters

Image with gaussian noise
Variance=300, mean = 0

Result of arithmetic mean filter
Mask size = 3
Mean Filters

Result of arithmetic mean filter
Mask size = 5

Result of arithmetic mean filter
Mask size = 9
Mean Filters

Image with gamma noise
Variance=300, mean = 0

Result of arithmetic mean filter
Mask size = 3
Mean Filters

Result of arithmetic mean filter
Mask size = 5

Result of arithmetic mean filter
Mask size = 9
Mean Filters

• Contra-harmonic mean filter:

\[
\text{Contra-Harmonic Mean} = \frac{\sum_{(r,c) \in w} d(r,c)^{R+1}}{\sum_{(r,c) \in w} d(r,c)^R}
\]

– Works for salt OR pepper noise, depending on the filter order R.
– Negative R \(\rightarrow\) Eliminate salt-type noise.
– Positive R \(\rightarrow\) Eliminate pepper-type noise.
Mean Filters

Image with salt noise
Probability = .04

Result of contra-harmonic filter
Mask size = 3; order = 0
Mean Filters

Result of contra-harmonic filter
Mask size = 3; order = -1

Result of contra-harmonic filter
Mask size = 3; order = -5
Mean Filters

Image with pepper noise
Probability = .04

Result of contra-harmonic filter
Mask size = 3; order = 0
Mean Filters

Result of contra harmonic filter
Mask size = 3; order = +1

Result of contra harmonic filter
Mask size = 3; order = +5
Mean Filters

• Geometric mean filter:

\[
\text{Geometric Mean} = \prod_{(r,c) \in w} \left[ d(r, c) \right]^{1/N^2}
\]

– Works best with gaussian noise.
– Retains detail better than arithmetic mean filter.
– Ineffective in the presence of pepper noise (if very low values present in the window, the equation will return a very small number).
Mean Filters

Image with gaussian noise
Variance = 300, mean = 0

Result of geometric filter
Mask size = 3
Mean Filters

Image with pepper noise
Probability = .04

Result of geometric filter
Mask size = 3
Mean Filters

Image with salt noise
Probability=.04

Result of geometric filter
Mask size = 3
Mean Filters

• Harmonic mean filter:

\[
\text{Harmonic Mean} = \frac{N^2}{\sum_{(r,c) \in w} \frac{1}{d(r,c)}}
\]

– Works with gaussian noise.
– Retains detail better than arithmetic mean filter.
– Works well with pepper noise.
Mean Filters

Image with pepper noise
Probability = .04

Result of harmonic filter
Mask size = 3
Mean Filters

Image with salt noise
Probability = .04

Result of harmonic filter
Mask size = 3
Order-Statistic Filtering

- Output is based on order of gray levels in the masked area
- Some simple neighbourhood operations include:
  - **Min**: Set the pixel value to the minimum in the neighbourhood
  - **Max**: Set the pixel value to the maximum in the neighbourhood
  - **Median**: The median value of a set of numbers is the midpoint value in that set
Median Filtering

Median Filtering is particularly effective when:
- The noise pattern consists of strong impulse noise (salt-and-pepper)

### Example

Sort the values and determine the median:

<table>
<thead>
<tr>
<th>10</th>
<th>20</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>25</td>
<td>100</td>
</tr>
</tbody>
</table>

Median = ? 20
Salt and Pepper Noise
Median Filtering

**FIGURE 3.37** (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a $3 \times 3$ averaging mask. (c) Noise reduction with a $3 \times 3$ median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)
Median Filtering

Noisy

Original
Median Filtering
Median Filtering

3x3-blur x 1  
3x3-median x 1
Median Filtering

3x3-blur x 2

3x3-median x 2
Median Filtering

3x3-blur x 5  3x3-median x 5
Median Filtering

3x3-blur x 10

3x3-median x 10
Adaptive Filters

The filters discussed so far are applied to an entire image without any regard for how image characteristics vary from one point to another. The behaviour of adaptive filters changes depending on the characteristics of the image inside the filter region. We will take a look at the adaptive median filter.
Adaptive Median Filtering

The median filter performs relatively well on impulse noise as long as the spatial density of the impulse noise is not large.
The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise.
The key insight in the adaptive median filter is that the filter size changes depending on the characteristics of the image.
Adaptive Median Filtering (cont...)

Remember that filtering looks at each original pixel image in turn and generates a new filtered pixel.

First examine the following notation:

- $z_{min}$ = minimum grey level in $S_{xy}$
- $z_{max}$ = maximum grey level in $S_{xy}$
- $z_{med}$ = median of grey levels in $S_{xy}$
- $z_{xy}$ = grey level at coordinates $(x, y)$
- $S_{max}$ = maximum allowed size of $S_{xy}$
Adaptive Median Filtering (cont...)

Level A: \[ A1 = z_{med} - z_{min} \]
\[ A2 = z_{med} - z_{max} \]
If \( A1 > 0 \) and \( A2 < 0 \), Go to level B
Else increase the window size
If window size \( \leq \) repeat \( S_{max} \) level A
Else output \( z_{med} \)

Level B: \[ B1 = z_{xy} - z_{min} \]
\[ B2 = z_{xy} - z_{max} \]
If \( B1 > 0 \) and \( B2 < 0 \), output \( z_{xy} \)
Else output \( z_{med} \)
Adaptive Median Filtering (cont...)

The key to understanding the algorithm is to remember that the adaptive median filter has three purposes:

– Remove impulse noise
– Provide smoothing of other noise
– Reduce distortion
Acknowledgements

- Digital Image Processing”, Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Peters, Richard Alan, II, Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008
- Brian Mac Namee, Digital Image Processing, School of Computing, Dublin Institute of Technology
- Computer Vision for Computer Graphics, Mark Borg