Digital Image Processing

Lecture 5
(Enhancement)

- Bu-Ali Sina University
- Computer Engineering Dep.
- Fall 2009
Image Enhancement in Spatial Domain

- Histogram based methods
  - Histogram Equalization
  - Histogram Specification
  - Local Histogram

- Spatial Filtering
  - Smoothing Filters
  - Median Filter
  - Sharpening
  - High Boost filter
  - Derivative filter
Chapter 3: Image Enhancement (Histogram-based methods)

- Histogram equalization yields an image whose pixels are (in theory) uniformly distributed among all graylevels.
- Sometimes, this may not be desirable. Instead, we may want a transformation that yields an output image with a prespecified histogram. This technique is called **histogram specification**.
- Again, we will assume, for the moment, continuous grayvalues.
- Suppose, the input image has probability density \( p_r(r) \). We want to find a transformation \( z = H(r) \), such that the probability density of the new image obtained by this transformation is \( p_z(z) \), which is not necessarily uniform.

**Histogram Specification**

- First apply the transformation

\[
s = T(r) = \int_0^r p_r(w) \, dw
\]

- This gives an image with a uniform probability density.
- If the desired output image were available, then the following transformation would generate an image with uniform density:

\[
G(z) = \int_0^z p_z(t) \, dt = s
\]
It then follows from these two equations that \( G(z) = T(r) \) and, therefore, that \( z \) must satisfy the condition

\[
z = G^{-1}(s) = G^{-1}[T(r)].
\]

For discrete graylevels, we have:

\[
s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}, \quad \text{for } 0 \leq k \leq L - 1 \quad \text{and}
\]

\[
\nu_k = G(z_k) = \sum_{j=0}^{k} p_Z(z_j), \quad \text{for } 0 \leq k \leq L - 1
\]
Example:
Consider the previous 8-graylevel 64 x 64 image histogram:

<table>
<thead>
<tr>
<th>$k$</th>
<th>$r_k$</th>
<th>$n_k$</th>
<th>$p(r_k) = n_k/n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>790</td>
<td>0.19</td>
</tr>
<tr>
<td>1</td>
<td>1/7</td>
<td>1023</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>2/7</td>
<td>850</td>
<td>0.21</td>
</tr>
<tr>
<td>3</td>
<td>3/7</td>
<td>656</td>
<td>0.16</td>
</tr>
<tr>
<td>4</td>
<td>4/7</td>
<td>329</td>
<td>0.08</td>
</tr>
<tr>
<td>5</td>
<td>5/7</td>
<td>245</td>
<td>0.06</td>
</tr>
<tr>
<td>6</td>
<td>6/7</td>
<td>122</td>
<td>0.03</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>81</td>
<td>0.02</td>
</tr>
</tbody>
</table>
It is desired to transform this image into a new image, using a transformation \( z = H(r) = G^{-1}[T(r)] \), with histogram as specified below:

<table>
<thead>
<tr>
<th>( k )</th>
<th>( z_k )</th>
<th>( p_{\text{out}}(z_k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>1/7</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>2/7</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>3/7</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td>4/7</td>
<td>0.20</td>
</tr>
<tr>
<td>5</td>
<td>5/7</td>
<td>0.30</td>
</tr>
<tr>
<td>6</td>
<td>6/7</td>
<td>0.20</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.15</td>
</tr>
</tbody>
</table>
The transformation $T(r)$ was obtained earlier (reproduced below):

<table>
<thead>
<tr>
<th>$r_j \rightarrow s_k$</th>
<th>$n_k$</th>
<th>$p(s_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_0 \rightarrow s_0 = 1/7$</td>
<td>790</td>
<td>0.19</td>
</tr>
<tr>
<td>$r_1 \rightarrow s_1 = 3/7$</td>
<td>1023</td>
<td>0.25</td>
</tr>
<tr>
<td>$r_2 \rightarrow s_2 = 5/7$</td>
<td>850</td>
<td>0.21</td>
</tr>
<tr>
<td>$r_3, r_4 \rightarrow s_3 = 6/7$</td>
<td>985</td>
<td>0.24</td>
</tr>
<tr>
<td>$r_5, r_6, r_7 \rightarrow s_4 = 1$</td>
<td>448</td>
<td>0.11</td>
</tr>
</tbody>
</table>
Chapter 3: Image Enhancement (Histogram-based methods)

Next we compute the transformation $G$ as before:

$$v_0 = G(z_0) = \sum_{j=0}^{0} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) = 0.00 \rightarrow 0$$

$$v_1 = G(z_1) = \sum_{j=0}^{1} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) + p_{\text{out}}(z_1) = 0.00 \rightarrow 0$$

$$v_2 = G(z_2) = \sum_{j=0}^{2} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) + p_{\text{out}}(z_1) + p_{\text{out}}(z_2) = 0.00 \rightarrow 0$$

$$v_3 = G(z_3) = \sum_{j=0}^{3} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) + p_{\text{out}}(z_1) + \cdots + p_{\text{out}}(z_3) = 0.15 \rightarrow \frac{1}{7}$$

$$v_4 = G(z_4) = \sum_{j=0}^{4} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) + p_{\text{out}}(z_1) + \cdots + p_{\text{out}}(z_4) = 0.35 \rightarrow \frac{2}{7}$$

$$v_5 = G(z_5) = \sum_{j=0}^{5} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) + p_{\text{out}}(z_1) + \cdots + p_{\text{out}}(z_5) = 0.65 \rightarrow \frac{5}{7}$$

$$v_6 = G(z_6) = \sum_{j=0}^{6} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) + p_{\text{out}}(z_1) + \cdots + p_{\text{out}}(z_6) = 0.85 \rightarrow \frac{6}{7}$$

$$v_7 = G(z_7) = \sum_{j=0}^{7} p_{\text{out}}(z_j) = p_{\text{out}}(z_0) + p_{\text{out}}(z_1) + \cdots + p_{\text{out}}(z_7) = 1.00 \rightarrow 1$$
Notice that $G$ is not invertible. But we will do the best possible by setting:

\[
\begin{align*}
G^{-1}(0) &= \text{(this does not matter, since } s \neq 0) \\
G^{-1}(1/7) &= 3/7 \\
G^{-1}(2/7) &= 4/7 \quad \text{(this does not matter, since } s \neq 2/7) \\
G^{-1}(3/7) &= 4/7 \quad \text{(this is not defined, but we use a close match)} \\
G^{-1}(4/7) &= \text{(this does not matter, since } s \neq 4/7) \\
G^{-1}(5/7) &= 5/7 \\
G^{-1}(6/7) &= 6/7 \\
G^{-1}(1) &= 1
\end{align*}
\]
Combining the two transformation $T$ and $G^{-1}$, we get our required transformation $H$:

<table>
<thead>
<tr>
<th>$r \rightarrow T(r) = s$</th>
<th>$s \rightarrow G^{-1}(s) = z$</th>
<th>$r \rightarrow G^{-1}[T(r)] = H(r) = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_0 = 0 \rightarrow 1/7$</td>
<td>$0 \rightarrow ?$</td>
<td>$r_0 = 0 \rightarrow z_3 = 3/7$</td>
</tr>
<tr>
<td>$r_1 = 1/7 \rightarrow 3/7$</td>
<td>$1/7 \rightarrow 3/7$</td>
<td>$r_1 = 1/7 \rightarrow z_4 = 4/7$</td>
</tr>
<tr>
<td>$r_2 = 2/7 \rightarrow 5/7$</td>
<td>$2/7 \rightarrow 4/7$</td>
<td>$r_2 = 2/7 \rightarrow z_5 = 5/7$</td>
</tr>
<tr>
<td>$r_3 = 3/7 \rightarrow 6/7$</td>
<td>$3/7 \rightarrow 4/7$</td>
<td>$r_3 = 3/7 \rightarrow z_6 = 6/7$</td>
</tr>
<tr>
<td>$r_4 = 4/7 \rightarrow 6/7$</td>
<td>$4/7 \rightarrow ?$</td>
<td>$r_4 = 4/7 \rightarrow z_6 = 6/7$</td>
</tr>
<tr>
<td>$r_5 = 5/7 \rightarrow 1$</td>
<td>$5/7 \rightarrow 5/7$</td>
<td>$r_5 = 5/7 \rightarrow z_7 = 1$</td>
</tr>
<tr>
<td>$r_6 = 6/7 \rightarrow 1$</td>
<td>$6/7 \rightarrow 6/7$</td>
<td>$r_6 = 6/7 \rightarrow z_7 = 1$</td>
</tr>
<tr>
<td>$r_7 = 1 \rightarrow 1$</td>
<td>$1 \rightarrow 1$</td>
<td>$r_7 = 1 \rightarrow z_7 = 1$</td>
</tr>
</tbody>
</table>
Applying the transformation $H$ to the original image yields an image with histogram as below:

<table>
<thead>
<tr>
<th>$k$</th>
<th>$z_k$</th>
<th>$n_k$</th>
<th>$n_k/n$ (actual hist.)</th>
<th>$p_{out}(z_k)$ (specified hist.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>1/7</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>2/7</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>3/7</td>
<td>790</td>
<td>0.19</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td>4/7</td>
<td>1023</td>
<td>0.25</td>
<td>0.20</td>
</tr>
<tr>
<td>5</td>
<td>5/7</td>
<td>850</td>
<td>0.21</td>
<td>0.30</td>
</tr>
<tr>
<td>6</td>
<td>6/7</td>
<td>985</td>
<td>0.24</td>
<td>0.20</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>448</td>
<td>0.11</td>
<td>0.15</td>
</tr>
</tbody>
</table>
Again, the actual histogram of the output image does not exactly but only approximately matches with the specified histogram. This is because we are dealing with discrete histograms.
Example:

Original image and its histogram

Histogram equalized image, Actual histogram of output

Histogram specified image, Actual Histogram, and Specified Histogram
FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA’s Mars Global Surveyor. (b) Histogram. (Original image courtesy of NASA.)
**FIGURE 3.21**
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).
**FIGURE 3.22**
(a) Specified histogram.
(b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17).
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).
FIGURE 3.24 SEM image of a tungsten filament and support, magnified approximately 130×. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene).
Image enhancement in the spatial domain can be represented as:

\[ g(m,n) = T(f)(m,n) \]

The transformation \( T \) maybe linear or nonlinear. We will mainly study linear operators \( T \) but will see one important nonlinear operation.

**How to specify \( T \)

- If the operator \( T \) is linear and shift invariant (LSI), characterized by the point-spread sequence (PSS) \( h(m, n) \), then (recall convolution):

\[
g(m,n) = h(m,n) * f(m,n)
= \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(m-k, n-l) f(k,l)
= \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(m-k, n-l) h(k,l)
\]
In practice, to reduce computations, \( h(m, n) \) is of “finite extent:”

\[
h(k, l) = 0, \quad \text{for } (k, l) \notin \Delta
\]

where \( \Delta \) is a small set (called neighborhood). \( \Delta \) is also called as the support of \( h \).

In the frequency domain, this can be represented as:

\[
G(u, v) = H_e(u, v)F_e(u, v)
\]

where \( H(u, v) \) and \( F(u, v) \) are obtained after appropriate zeropadding.

Many LSI operations can be interpreted in the frequency domain as a “filtering operation.” It has the effect of filtering frequency components (passing certain frequency components and stopping others).

The term \textit{filtering} is generally associated with such operations.
Examples of some common filters (1-D case):

- **Lowpass filter**
  
  ![Lowpass filter](image)

- **Highpass filter**
  
  ![Highpass filter](image)
Spatial Filtering

**FIGURE 3.32** The mechanics of spatial filtering. The magnified drawing shows a $3 \times 3$ mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.
If $h(m, n)$ is a 3 by 3 mask given by:

\[
h = \begin{bmatrix}
w_1 & w_2 & w_3 \\
w_4 & w_5 & w_6 \\
w_7 & w_8 & w_9
\end{bmatrix}
\]

then

\[
g(m,n) = w_1 f(m-1,n-1) + w_2 f(m-1,n) + w_3 f(m-1,n+1) \\
+ w_4 f(m,n-1) + w_5 f(m,n) + w_6 f(m,n+1) \\
+ w_7 f(m+1,n-1) + w_8 f(m+1,n) + w_9 f(m+1,n+1)
\]
The output \( g(m, n) \) is computed by sliding the mask over each pixel of the image \( f(m, n) \). This filtering procedure is sometimes referred to as **moving average** filter.

Special care is required for the pixels at the border of image \( f(m, n) \). This depends on the so-called boundary condition. Common choices are:
- The mask is truncated at the border (**free boundary**)
- The image is extended by appending extra rows/columns at the boundaries. The extension is done by repeating the first/last row/column or by setting them to some constant (**fixed boundary**).
- The boundaries “wrap around” (**periodic boundary**).

In any case, the final output \( g(m, n) \) is restricted to the support of the original image \( f(m, n) \).

The mask operation can be implemented in matlab using the \texttt{filter2} command, which is based on the \texttt{conv2} command.
Image smoothing refers to any image-to-image transformation designed to "smooth" or flatten the image by reducing the rapid pixel-to-pixel variation in grayvalues.

Smoothing filters are used for:

- **Blurring**: This is usually a preprocessing step for removing small (unwanted) details before extracting the relevant (large) object, bridging gaps in lines/curves,
- **Noise reduction**: Mitigate the effect of noise by linear or nonlinear operations.

**Image smoothing by averaging (lowpass spatial filtering)**

Smoothing is accomplished by applying an **averaging mask**.

An averaging mask is a mask with positive weights, which sum to 1. It computes a weighted average of the pixel values in a neighborhood. This operation is sometimes called **neighborhood averaging**.
Spatial Filtering - Smoothing Filters

- Some 3 x 3 averaging masks:

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 0
\end{bmatrix} \quad \frac{1}{5}
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix} \quad \frac{1}{9}
\begin{bmatrix}
1 & 3 & 1 \\
3 & 16 & 3 \\
1 & 3 & 1
\end{bmatrix} \quad \frac{1}{32}
\begin{bmatrix}
0 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 0
\end{bmatrix} \quad \frac{1}{8}
\]

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix} \quad \frac{1}{9}
\begin{bmatrix}
1 & 1 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1
\end{bmatrix} \quad \frac{1}{16}
\]

**FIGURE 3.34** Two 3 × 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.
Spatial Filtering - Smoothing Filters

- This operation is equivalent to lowpass filtering.

**Example of Image Blurring**

![Original Image](image1.png) ![Avg. Mask](image2.png)

\[
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
1 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 1 \\
\end{bmatrix}_{N\times N}
\]

\[\frac{1}{N^2}\]

\[N = 3\quad N = 5\quad N = 7\]

\[N = 11\quad N = 15\quad N = 21\]
Spatial Filtering - Smoothing Filters

**Figure 3.35** (a) Original image, of size 500 \( \times \) 500 pixels. (b)-(f) Results of smoothing with square averaging filter masks of sizes \( n = 3, 5, 9, 15, \) and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 \( \times \) 120 pixels.
Example of noise reduction:

Spatial Filtering - Smoothing Filters

Noise-free Image

Zero-mean Gaussian noise, Variance = 0.01

Zero-mean Gaussian noise, Variance = 0.05