DIGITAL IMAGE PROCESSING (COM-3371)
Week 2 - January 14, 2002

Topics:

• Human eye
• Visual phenomena
• Simple image model
• Image enhancement
• Point processes
• Histogram
• Lookup tables
• Contrast compression and stretching
• Image negatives
• Intensity-level slicing
• Thresholding
• Bit-plane slicing
• Histogram Equalization
• Arithmetic operations

• Introduction to MATLAB
Readings

• Chapters 2 (§2.1-2.4, 2.6) and 3 (§3.1-3.2, 3.3.1, 3.3.3, 3.3.4, 3.4) of text
• Examples of point processing (examples_part1 and examples_part2 - PDF files on our Web site)
• Introduction to MATLAB (posted in MATLAB on our Web page)
• Short article about artificial retina (retina.html on our Web site) - optional

Homework

• Homework 1 - due January 28, 2002 (posted in Homeworks on our Web page)
Elements of Visual Perception

- Digital image processing field is built on a foundation of mathematical and probabilistic formulations, but human intuition and analysis play an important role in the choice of one technique versus another --> it is important to develop a basic understanding of human visual perception.

- For example, we would like to know:

  How the image is formed in the eye?

  What are the physical limitations of human vision in terms of resolution and ability to adapt to changes in illumination?
**Human Eye**

chromaticity - color quality of light defined by its wavelength
  --> perceived as color
luminance - amount of light  --> perceived as brightness

____________________________

**Photoreceptors:**

**CONES** ~ 6-7 million; located in the center of retina, called *fovea*; each cone is connected to its own nerve; highly sensitive to color and fine details;
*photopic vision* - bright light vision

**RODS** ~ 75-150 million; several rods are connected to a single nerve; highly sensitive to low levels of illumination;
*scotopic vision* - dim-light vision
Human Eye
Human Eye

• rods and cones are **not evenly** distributed throughout the retina
• **fovea** - most of the cones are distributed here; it is the region of most accurate vision in bright light
• **blind spot** - small region where the optic nerves bundle up
• when light hits cones and rods, a complex electrochemical reaction takes place - light is converted to **neural impulses**, which are transmitted to the brain
Mach Band Pattern Phenomenon
(Ernst Mach, 1865)

• The perceived brightness is not a simple function of intensity
• Visual system tends to undershoot or overshoot around the boundary of regions of different intensities
Simultaneous Contrast Phenomenon

• Region’s perceived brightness does not depend simply on its intensity

• The small squares have value of 100; the background is changing: 0, 30, 60, 150 (clockwise from upper left); display range: 0-255
Optical illusions

- Other examples of human perception phenomena are optical illusions, in which the eye fills in non-existing information or wrongly perceives geometrical properties of objects --> see Fig. 2.9 in the textbook
Summary of important points:

• The range of light intensity levels to which the human visual system can adapt is enormous - on the order of $10^{10}$
• Subjective brightness = brightness as perceived by human visual system
• Subjective brightness is a logarithmic function of the light intensity incident on the eye
• Human eye has a considerable acuity in discriminating fine detail (high spatial information)
• Human eye is not particularly sensitive to low-frequency (slowly varying) information in the image
• Some images may be more easily understood if they are displayed indirectly by using contour lines, shading, color, or some other graphical representation
A SIMPLE IMAGE MODEL

• Image - 2D light intensity function $f(x,y)$, where the value of $f$ at spatial coordinates $(x,y)$ gives the intensity of the image at that point:

$$0 < f(x,y) < \infty$$

• The intensity of image $f$ at coordinates $(x,y)$ is the gray level $(l)$ of the image

$$L_{\text{min}} \leq l \leq L_{\text{max}}$$

• The interval $[L_{\text{min}}, L_{\text{max}}]$ is called the gray scale;
typical scale for 8-bit images: $[0,255]$, where 0 is considered black, 255 - white, and all intermediate values are shades of gray varying from black to white

• Continuous image $f(x)$ is approximated by equally spaced samples arranged in the form of an $N\times M$ array:

$$f(x) \approx \begin{bmatrix}
                        f(0,0) & f(0,1) & \ldots & f(0,M-1) \\
                        f(1,0) & f(1,1) & \ldots & f(1,M-1) \\
                          \vdots & \vdots & \ddots & \vdots \\
                        f(N-1,0) & f(N-1,1) & \ldots & f(N-1,M-1)
                    \end{bmatrix}$$

• Usually, $N$ and $M$ are integer powers of two: $N=2^n$, $M=2^k$; number of gray levels: $G=2^m$; therefore number of bits required to store an image: $b=N \times M \times m$; for example: 128 by 128 image with 256 gray levels requires 131,072 bits =16,384 bytes =16KB of storage --- see Table 2.1 in the textbook
Image sampling and quantization

- To create a digital image, we need to convert the continuous sensed data into digital form --> this involves sampling and quantization
- Digitizing the coordinate values is called **sampling**
- Digitizing the amplitude values is called **quantization**
  ---> see Fig. 2.16 in the textbook
- In practice, the method of sampling is determined by the sensor arrangement used to generate the image
- The quality of a digital image is determined by the number of samples and the number of gray levels
Spatial resolution and intensity resolution

- Sampling is the principal factor defining the spatial resolution of an image, and quantization is the principal factor defining the intensity resolution
- **Spatial resolution - number of rows and columns**
  for example: 128 x128, 256 by 256, etc.; -->see Figs. 2.19 and 2.20
- **Intensity resolution - number of gray levels**
  for example: 8 bits, 16 bits, etc.; --->see Fig. 2.21
Image Enhancement

Image processing in spatial domain:
\[ g(x,y) = T[f(x,y)] \]
f(x,y) - input image, g(x,y) - processed image, T - operator on f, defined over some neighborhood of (x,y)

Image processing in frequency domain:
\[ g(x,y) = h(x,y) * f(x,y) \]
f(x,y) - input image, g(x,y) - image formed by the convolution of an image f(x,y) and a linear, position invariant operator h(x,y)
From the convolution theorem: \( G(u,v) = H(u,v) F(u,v) \), where \( G \), \( H \) and \( F \) are the Fourier transforms of \( g \), \( h \), and \( f \). \( H \) is called transfer function.
In a typical image enhancement application, \( f(x,y) \) is given and the goal is to select \( H(u,v) \) so the desired image

\[
g(x,y) = \mathcal{F}^{-1}[H(u,v)F(u,v)]
\]

exhibits some highlighted feature of \( f(x,y) \).

For example, edges in \( f(x,y) \) can be enhanced by using a function \( H(u,v) \) that emphasizes the high-frequency components of \( F(u,v) \).
Point Processes

- **Point processes** are the simplest of basic image processing operations. A point operation takes a **single input image into a single output image** in such a way that each output pixel's gray level depends **only upon the gray level of the corresponding input pixel**. Thus, a point operation cannot modify the spatial relationships within an image.

- Point operations **transform the gray scale of an image**

- **Linear and nonlinear** point operations

- Applications: photometric calibration, display calibration, enhancement and histogram modification

- **Examples:**
  - Contrast Stretching
  - Image Negatives
  - Intensity-level Slicing
  - Bit-plane Slicing
  - Other Intensity Transformations
  - Histogram Equalization
Histogram

- Gray level **histogram** of an image - a function showing for each gray level the number of pixels in the image that have that gray level; it is simply a bar graph of the pixel intensities.

**CT image**

**Histogram of the CT image**

- Histogram gives us a convenient, easy-to-read representation of concentration of pixels versus intensity in an image.
- **Dynamic range** - an range of intensity values that occur in an image.
- **Contrast stretching** - if image has low-dynamic range; low-dynamic range can result from poor illumination, lack of dynamic range in imaging sensor, wrong setting of the sensor parameters, etc.
- **Compression of dynamic range** - if the dynamic range of the image far exceeds the capability of the display device.
Histogram calculation

<table>
<thead>
<tr>
<th>4</th>
<th>4</th>
<th>3</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Examples of several types of image histograms:
### Lookup Tables (LUTs)

- LUTs are arrays that use the current pixel value as the array index; the new value is the array element pointed by this index:

![LUT diagram]

- Computational savings - by using LUTs you avoid needless repeated computations for each pixel; for example, if you were to add some value to every pixel in a 512 x 512 image, it would require 262,144 operations without LUT, and 256 operations with LUT
Examples

**Linear LUTs**

Output = a*Input + b; where a and b are constants

**Nonlinear LUTs**

Output

Input
Compression of Dynamic Range

Image 1

Image 2
Contrast Stretching

Image 2

Image 3
There is a difference between image1 and image3:

\[
\text{image1} - \text{image3} = \text{image4}
\]
(scaled to show the difference between image 1 and image3)
Image Negatives

Output

255

0

Input

0

255

Original image

Result of transformation
Intensity-level (or gray-level) slicing

Original image

Result of transformation
Thresholding

Output

255

0

Input

0

T

255

Original image

Result of transformation
Other transformations

• Log transformations ---> Fig. 3.5
• Power-law transformations -- > Fig. 3.6
• Piecewise-linear transformations -- Fig. 3.10
Bit-plane slicing

• We can “slice” an image that has $n$ bits per pixel in bit planes. Assuming that zero is the least significant bit (LSB) and 7 is the most significant bit (MSB) we get the following slices:

  slice 0 which displays all pixels with bit 0 set: 0000 0001
  slice 1 which displays all pixels with bit 1 set: 0000 0010
  slice 2 which displays all pixels with bit 2 set: 0000 0100
  slice 3 which displays all pixels with bit 3 set: 0000 1000
  slice 4 which displays all pixels with bit 4 set: 0001 0000
  slice 5 which displays all pixels with bit 5 set: 0010 0000
  slice 6 which displays all pixels with bit 6 set: 0100 0000
  slice 7 which displays all pixels with bit 7 set: 1000 0000

• Bit-plane slicing is useful for analyzing the relative performance played by each bit of the image

• Application of bit-level slicing: testing the adequacy of the number of bits used to quantize each pixel; data compression
Bit-level slicing - example

Original

Bit-level 0

Bit-level 1

Bit-level 2

Bit-level 3

Bit-level 4

Bit-level 5

Bit-level 6

Bit-level 7
Histogram Equalization

• Images with poor intensity distributions can often be enhanced with histogram equalization

• The goal is to obtain a uniform histogram

• Histogram equalization will not flatten a histogram; if a histogram has peaks and valleys it will still have them after equalization - they will be shifted and spread over the entire range of image intensities

• Works best on images with fine details in darker regions

• Use it carefully - good images can be often degraded by histogram equalization
Consider a 128 x 128 pixels image that contains L=8 gray levels with the following distribution of pixels:

<table>
<thead>
<tr>
<th>gray levels $r_k$</th>
<th>$n_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1028</td>
</tr>
<tr>
<td>1</td>
<td>3544</td>
</tr>
<tr>
<td>2</td>
<td>5023</td>
</tr>
<tr>
<td>3</td>
<td>3201</td>
</tr>
<tr>
<td>4</td>
<td>1867</td>
</tr>
<tr>
<td>5</td>
<td>734</td>
</tr>
<tr>
<td>6</td>
<td>604</td>
</tr>
<tr>
<td>7</td>
<td>383</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>16384</strong></td>
</tr>
</tbody>
</table>

- The goal is to find a gray scale transformation $s_k = T(r_k)$ that yields a uniform histogram
- The transformation that converts an image to a histogram equalized image is simply a transformation that sums the elements of the histogram from the original image and is given in the following equation:

$$ s_k = T(r_k) = (L - 1) \cdot \sum_{j=0}^{k} \frac{n_j}{N} = (L - 1) \cdot \sum_{j=0}^{k} p_r(r_j) $$
The new equalized gray levels:

<table>
<thead>
<tr>
<th>$r_k$</th>
<th>$n_k$</th>
<th>$n_k / N$</th>
<th>$\sum_{j=0}^{k} n_j / N$</th>
<th>$(L - 1) \sum_{j=0}^{k} n_j / N$</th>
<th>$s_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1028</td>
<td>0.0627</td>
<td>0.0627</td>
<td>0.4392</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3544</td>
<td>0.2163</td>
<td>0.2791</td>
<td>1.9534</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>5023</td>
<td>0.3066</td>
<td>0.5856</td>
<td>4.0994</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3201</td>
<td>0.1954</td>
<td>0.7810</td>
<td>5.4670</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>1867</td>
<td>0.1140</td>
<td>0.8950</td>
<td>6.2647</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>734</td>
<td>0.0448</td>
<td>0.9398</td>
<td>6.5783</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>604</td>
<td>0.0369</td>
<td>0.9766</td>
<td>6.8364</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>383</td>
<td>0.0234</td>
<td>1.0000</td>
<td>7.0000</td>
<td>7</td>
</tr>
</tbody>
</table>

$N = 16384$
NOTE: Please check the result of histogram equalization on Fig. 3.17 (Gonzalez & Woods)
Histogram Equalization - Example
Local enhancement

- The histogram equalization previously described is easily adaptable to local enhancement:
  1. Define a square or rectangular area (region of interest, ROI) and move the center of this area pixel by pixel
  2. At each location, compute the histogram of the ROI
  3. Perform histogram equalization in the ROI
  4. Move the ROI to the adjacent pixel location; repeat steps 2 and 3

   Example ---> See Fig. 3.23 in the textbook

- Local statistics (mean, standard deviation, etc.) can be used for image enhancement ---> Figs. 3.24-3.26
Arithmetic Operations (one image)

• Examples: adding, subtracting, dividing and multiplying pixels by a constant value

• The arithmetic operations can create images with negative values and/or values greater than the maximum possible dynamic range --> so you have to do image scaling (or clamping)

A

Dynamic range: 0-200
(Display range: 0-255)

A +100

Dynamic range: 100-300
(Display range: 0-255)

A / 2 + 100

Dynamic range: 100-200
(Display range: 0-255)
Arithmetic Operations (two or more images)

Image A  Image B  A* B  A + B

Image C  Image D  Pixelwise maximum (C,D)