ECE 472/572 - Digital Image Processing

Lecture 3 - Image Enhancement - Point Processing
08/30/11

Roadmap

- Introduction
  - Image format (vector vs. bitmap)
  - IP vs. CV vs. CG
  - HLIP vs. LLIP
  - Image acquisition
- Perception
  - Structure of human eye
  - Rods vs. cones (Scotopic vision vs. photopic vision)
  - Fovea and blind spot
  - Flexible lens (near-sighted vs. far-sighted)
- Brightness adaptation and Discrimination
  - Weber ratio
  - Dynamic range
- Image resolution
  - Sampling vs. quantization
- Image enhancement
  - Enhancement vs. restoration
  - Spatial domain methods
    - Point-based methods
      - Negative
      - Log transformation
      - Power law
      - Contrast stretching
      - Gray-level slicing
      - Histogram equalization
    - Averaging
  - Mask-based (neighborhood-based) methods - spatial filter
  - Frequency domain methods

Questions

- Point-based vs. Mask-based (or neighbor-based)
- Spatial domain vs. Frequency domain
- Log transformation vs. Power-law
  - Gamma correction
  - Dynamic range compression
- Contrast stretching vs. Histogram equalization
  - Histogram
  - Uniform histogram
  - HE derivation (572)
- Gray-level vs. Bit-plane slicing
  - MSB
- What’s the philosophy behind Image averaging?
  - Derivation (572)
Intuitively

From system point of view

Different approaches

* Spatial domain
  - Point-based processing
  - Mask-based processing (neighbor-based processing) (spatial filters)
* Frequency domain
  - Frequency domain filters
Point processing

- Simple gray level transformations
  - Image negatives
  - Log transformations
  - Power-law transformations
  - Contrast stretching
  - Gray-level slicing
- Bit-plane slicing
- Histogram processing
  - Histogram equalization
  - Histogram matching (specification)

Arithmetic/logic operations
- Image averaging

Some transformations

\[ s = T(r) = L - 1 - r \]
\[ s = T(r) = c \log(1 + |r|) \]
\[ s = cr^\gamma \]
\[ s = T(r) = mr + b \]
\[ s = T(r) = \begin{cases} 255 & \text{if } A \leq r \leq B \\ 0 & \text{otherwise} \end{cases} \]

Image negatives

\[ s = T(r) = L - 1 - r \]
Log transformation (Dynamic range compression)

\[ s = T(r) = c \log(1 + r) \]

Power-Law transformation

\[ s = cr^\gamma \]

Gamma correction
Contrast stretching

\[ s = T(r) = mr + b \]

Gray-level slicing

\[ s = T(r) = \begin{cases} 255 & \text{if } A \leq r \leq B \\ 0 & \text{otherwise} \end{cases} \]

Highlighting an intensity range

Bit-plane slicing

- Highlighting the contribution made by a specific bit.
- For pgm images, each pixel is represented by 8 bits.
- Each bit-plane is a binary image
Point processing

- Simple gray level transformations
  - Image negatives
  - Log transformations
  - Power-law transformations
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing
- Histogram processing
  - Histogram equalization
  - Histogram matching (specification)
- Arithmetic/logic operations
  - Image averaging

Histogram

- Gray-level histogram is a function showing, for each gray level, the number of pixels in the image that have that gray level.
  \[ n_k = \text{hist}[k] = \sum_{l=0}^{L-1} 1 \]
- Normalized histogram (probability):
  \[ p_k = n_k / N \]
**Histogram - Examples**

**Usage of histogram**

- Find threshold
- Reveal the intensity distribution
Question

What does the histogram of a low-contrast image look like?
How about high-contrast?

Histogram equalization

Why and when do we want to use HE?

HE – Example 2
**Histogram equalization**

* Transformation function

\[ s = T(r) = \int p_r(w) \, dw \quad 0 \leq r \leq 1 \]

* \( p_r(w) \) is the probability density function (pdf)

* The transformation function is the cumulative distribution function (CDF)

* To make the pdf of the transformed image uniform, i.e. to make the histogram of the transformed image uniform

**HE – Derivation (572)**

\[ s = T(r) \Rightarrow r = T^{-1}(s) \]

\[ p_r(s) = \frac{p_r(r)}{dr} \]

\[ s_i = T(r_i) = \sum_{j=1}^{n} n \sum_{i=1}^{s_i} p_r(r_i) \]

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

\[ p_r(s) = \frac{p_r(r)}{dr} \]

**HE – Discrete case**

\[ s_i = T(r_i) = \sum_{j=1}^{n} n \sum_{i=1}^{s_i} p_r(r_i) \]

| \( r \) | hist(r) | | \( r \) | s | | \( s \) | hist(s) |
|---|---|---|---|---|---|---|
| 0 | 10 | | 0 | 10 | | 10 | 10 |
| 1 | 70 | | 1 | 80 | | 80 | 70 |
| 2 | 15 | | 2 | 90 | | 95 | 15 |
| 3 | 5 | | 3 | 100 | | 100 | 5 |

---

---

---

---

---

---

---
**HE - Discussion**

- Can contrast stretching achieve similar result as histogram equalization?
- If it can, why histogram equalization then?
- Why isn’t the transformed histogram uniform?
Problems with HE

- ???
- Solutions
  - ???

*Histogram specification*

- Step 1: Equalize the levels of the original image
- Step 2: Specify the desired pdf and obtain the transformation function
- Step 3: Apply the inverse transformation function to the levels obtained in step 1

**HS - Example**

<table>
<thead>
<tr>
<th>r</th>
<th>hist(r)</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>70</td>
<td>1</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>2</td>
<td>95</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>3</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>z</th>
<th>hist(z)</th>
<th>z</th>
<th>G(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>50</td>
<td>30</td>
<td>80</td>
</tr>
<tr>
<td>60</td>
<td>15</td>
<td>60</td>
<td>95</td>
</tr>
</tbody>
</table>

Specified histogram
Comparison (HE vs. HM/HS)

Local enhancement

Point processing

- Simple gray level transformations
  - Image negatives
  - Log transformations
  - Power-law transformations
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing
- Histogram processing
  - Histogram equalization
  - Histogram matching (specification)
- Arithmetic/logic operations
  - Image averaging
Image averaging

original image \( f(x, y) \) ——> noisy image \( g(x, y) \)

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

\[
\sum_{m=0}^{M-1} g(x, y) = \sum_{m=0}^{M-1} f(x, y) + \sum_{m=0}^{M-1} \eta(x, y)
\]

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

Image average (cont’)

* If the noise is uncorrelated and has zero expectation, then

\[
E[g(x, y)] = f(x, y)
\]

\[
\sigma^2\tau(x, y) = \frac{1}{M}\sigma^2\eta(x, y)
\]

Image averaging – How to generate Gaussian noise? (572)

```java
// This function creates a gaussian random number between -3 and 3
double gaussian() {
    static double V1, V2, S;
    static int phase = 0;
    double X;
    if (phase == 0) {
        do {
            double U1 = (double)rand() / RAND_MAX;
            double U2 = (double)rand() / RAND_MAX;
            V1 = 2 * U1 - 1;
            V2 = 2 * U2 - 1;
            S = V1 * V1 + V2 * V2;
        } while(S >= 1) S = 0;
        X = V1 * sqrt(-2 * log(S) / S);
    }
    else
        X = V2 * sqrt(-2 * log(S) / S);
    phase = 1 - phase;
    return X;
}
```
Summary - Point processing

• Simple gray level transformations
  - Image negatives
  - Log transformations
  - Power-law transformations
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing
• Histogram processing
  - Histogram equalization
    • Derivation (572 only)
  - Histogram matching (specification) (572 only)
• Arithmetic/logic operations
  - Image averaging
    • Generation of Gaussian noise (572 only)