Digital Image Processing

Intensity Transformations
(Histogram Processing)

Christophoros Nikou
cnikou@cs.uoi.gr

Contents

Over the next few lectures we will look at image enhancement techniques working in the spatial domain:
- Histogram processing
- Spatial filtering
- Neighbourhood operations
The histogram of an image shows us the distribution of grey levels in the image. Massively useful in image processing, especially in segmentation.
Histogram Examples (cont...)


C. Nikou – Digital Image Processing (E12)
Histogram Examples (cont...)


C. Nikou – Digital Image Processing (E12)
Histogram Examples (cont...)


C. Nikou – Digital Image Processing (E12)
Histogram Examples (cont…)

• A selection of images and their histograms
• Notice the relationships between the images and their histograms
• Note that the high contrast image has the most evenly spaced histogram
Contrast Stretching

- We can fix images that have poor contrast by applying a pretty simple contrast specification
- The interesting part is how do we decide on this transformation function?

Histogram Equalisation

- Spreading out the frequencies in an image (or equalising the image) is a simple way to improve dark or washed out images.
- At first, the continuous case will be studied:
  - $r$ is the intensity of the image in $[0, L-1]$.
  - we focus on transformations $s=T(r)$:
    - $T(r)$ is strictly monotonically increasing.
    - $T(r)$ must satisfy:
      \[
      0 \leq T(r) \leq L-1, \quad \text{for} \quad 0 \leq r \leq L-1
      \]
Histogram Equalisation (cont...)

- The condition for $T(r)$ to be monotonically increasing guarantees that ordering of the output intensity values will follow the ordering of the input intensity values (avoids reversal of intensities).
- If $T(r)$ is strictly monotonically increasing then the mapping from $s$ back to $r$ will be 1-1.
- The second condition ($T(r)$ in $[0,1]$) guarantees that the range of the output will be the same as the range of the input.

---

Histogram Equalisation (cont...)

a) We cannot perform inverse mapping (from $s$ to $r$).

b) Inverse mapping is possible.
• We can view intensities $r$ and $s$ as random variables and their histograms as probability density functions (pdf) $p_r(r)$ and $p_s(s)$.

• Fundamental result from probability theory:
  – If $p_r(r)$ and $T(r)$ are known and $T(r)$ is continuous and differentiable, then

$$p_s(s) = p_r(r) \frac{1}{ds} \left| \frac{dr}{ds} \right| = p_r(r) \int dr$$

The pdf of the output is determined by the pdf of the input and the transformation.

This means that we can determine the histogram of the output image.

A transformation of particular importance in image processing is the cumulative distribution function (CDF) of a random variable:

$$s = T(r) = (L-1) \int_0^r p_r(w) \, dw$$
Histogram Equalisation (cont...)

- It satisfies the first condition as the area under the curve increases as \( r \) increases.
- It satisfies the second condition as for \( r=L-1 \) we have \( s=L-1 \).
- To find \( p_s(s) \) we have to compute

\[
\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1) \left[ \int_0^r p_r(w) \, dw = (L-1)p_r(r) \right]
\]

Substituting this result:

\[
\frac{ds}{dr} = (L-1)p_r(r)
\]

to

\[
p_s(s) = p_r(r) \left| \frac{dr}{ds} \right|
\]

yields

\[
p_s(s) = p_r(r) \left| \frac{1}{(L-1)p_r(r)} \right| = \frac{1}{L-1}, \quad 0 \leq s \leq L-1
\]
The formula for histogram equalisation in the discrete case is given

\[ s_k = T(r_k) = (L - 1) \sum_{j=0}^{k} p_r(r_j) = \frac{(L - 1)}{MN} \sum_{j=0}^{k} n_j \]

where
- \( r_k \): input intensity
- \( s_k \): processed intensity
- \( n_j \): the frequency of intensity \( j \)
- \( MN \): the number of image pixels.

### Example

A 3-bit 64x64 image has the following intensities:

<table>
<thead>
<tr>
<th>( r_k )</th>
<th>( n_k )</th>
<th>( p_r(r_k) )</th>
<th>( \frac{n_k}{MN} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>700</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1023</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>850</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>656</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>329</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>245</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>122</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>81</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

Applying histogram equalization:

\[ s_0 = T(r_0) = 7 \sum_{j=0}^{3} p_r(r_j) = 7 p_r(r_0) = 1.33 \]

\[ s_1 = T(r_1) = 7 \sum_{j=0}^{1} p_r(r_j) = 7 p_r(r_0) + 7 p_r(r_1) = 3.08 \]
Histogram Equalisation (cont...)

Example

Rounding to the nearest integer:

\[ s_0 = 1.33 \rightarrow 1 \quad s_1 = 3.08 \rightarrow 3 \quad s_2 = 4.55 \rightarrow 5 \quad s_3 = 5.67 \rightarrow 6 \]
\[ s_4 = 6.23 \rightarrow 6 \quad s_5 = 6.65 \rightarrow 7 \quad s_6 = 6.86 \rightarrow 7 \quad s_7 = 7.00 \rightarrow 7 \]

Notice that due to discretization, the resulting histogram will rarely be perfectly flat. However, it will be extended.
Equalisation Transformation Function


Equalisation Examples

Equalisation Transformation Functions

The functions used to equalise the images in the previous example

Equalisation Examples


C. Nikou – Digital Image Processing (E12)
Equalisation Transformation Functions

The functions used to equalise the images in the previous example

Equalisation Examples (cont…)

Equalisation Transformation Functions

The functions used to equalise the images in the previous examples

Histogram Specification

- Histogram equalization does not always provide the desirable results.

- Image of Phobos (Mars moon) and its histogram.
- Many values near zero in the initial histogram
In these cases, it is more useful to specify the final histogram.

Problem statement:
- Given $p_x(r)$ from the image and the target histogram $p_z(z)$, estimate the transformation $z=T(r)$.
- The solution exploits histogram equalization.
Histogram specification (cont...) 

- Equalize the initial histogram of the image:
  \[ s = T(r) = (L - 1) \sum_{w} p_r(w) \, dw \]
  \[ G(z) = T(r) \]

- Equalize the target histogram:
  \[ s = G(z) = (L - 1) \sum_{w} p_z(w) \, dw \]

- Obtain the inverse transform:
  \[ z = G^{-1}(s) = G^{-1}(T(r)) \]

In practice, for every value of \( r \) in the image:
- get its equalized transformation \( s = T(r) \).
- perform the inverse mapping \( z = G^{-1}(s) \), where \( s = G(z) \) is the equalsed target histogram.

The discrete case:

- Equalize the initial histogram of the image:
  \[ s_k = T(r_k) = (L - 1) \sum_{j=0}^{k} p_r(r_j) = \frac{(L - 1)}{MN} \sum_{j=0}^{k} n_j \]
  \[ G(z) = T(r) \]

- Equalize the target histogram:
  \[ s_k = G(z_q) = (L - 1) \sum_{i=0}^{q} p_z(r_i) \]

- Obtain the inverse transform:
  \[ z_q = G^{-1}(s_k) = G^{-1}(T(r_k)) \]
Consider again the 3-bit 64x64 image:

<table>
<thead>
<tr>
<th>$r_i$</th>
<th>$n_i$</th>
<th>$p_i = n_i / MN$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>790</td>
<td>0.19</td>
</tr>
<tr>
<td>1</td>
<td>1023</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>850</td>
<td>0.21</td>
</tr>
<tr>
<td>3</td>
<td>636</td>
<td>0.16</td>
</tr>
<tr>
<td>4</td>
<td>820</td>
<td>0.08</td>
</tr>
<tr>
<td>5</td>
<td>245</td>
<td>0.06</td>
</tr>
<tr>
<td>6</td>
<td>122</td>
<td>0.03</td>
</tr>
<tr>
<td>7</td>
<td>81</td>
<td>0.02</td>
</tr>
</tbody>
</table>

It is desired to transform this histogram to:

$p_z(z_0) = 0.00$  
$p_z(z_1) = 0.00$  
$p_z(z_2) = 0.00$  
$p_z(z_3) = 0.15$  
$p_z(z_4) = 0.20$  
$p_z(z_5) = 0.30$  
$p_z(z_6) = 0.20$  
$p_z(z_7) = 0.15$

with $z_0 = 0, z_1 = 1, z_2 = 2, z_3 = 3, z_4 = 4, z_5 = 5, z_6 = 6, z_7 = 7$.

The first step is to equalize the input (as before):

$s_0 = 1, s_1 = 3, s_2 = 5, s_3 = 6, s_4 = 6, s_5 = 7, s_6 = 7, s_7 = 7$

The next step is to equalize the output:

$G(z_0) = 0$  
$G(z_1) = 0$  
$G(z_2) = 0$  
$G(z_3) = 1$  
$G(z_4) = 2$  
$G(z_5) = 5$  
$G(z_6) = 6$  
$G(z_7) = 7$

Notice that $G(z)$ is not strictly monotonic. We must resolve this ambiguity by choosing, e.g. the smallest value for the inverse mapping.
Perform inverse mapping: find the smallest value of $z_q$ that is closest to $s_k$:

\[
\begin{align*}
    s_k &= T(r_i) \\
    G(z_q) &= s_k ightarrow z_q \\
    s_0 &= 1 \quad G(z_0) = 0 \\
    s_1 &= 3 \quad G(z_1) = 0 \\
    s_2 &= 5 \quad G(z_2) = 0 \\
    s_3 &= 6 \quad G(z_3) = 1 \\
    s_4 &= 6 \quad G(z_4) = 2 \\
    s_5 &= 7 \quad G(z_5) = 5 \\
    s_6 &= 7 \quad G(z_6) = 6 \\
    s_7 &= 7 \quad G(z_7) = 7
\end{align*}
\]

E.g. every pixel with value $s_0 = 1$ in the histogram-equalized image would have a value of $3 (z_3)$ in the histogram-specified image.

---

Notice that due to discretization, the resulting histogram will rarely be exactly the same as the desired histogram.

Histogram equalization

C. Nikou – Digital Image Processing (E12)
Specified histogram

Transformation function and its inverse

Resulting histogram

Local Histogram Processing

- Image in (a) is slightly noisy but the noise is imperceptible.
- HE enhances the noise in smooth regions (b).
- Local HE reveals structures having values close to the values of the squares and small sizes to influence HE (c).
We have looked at:

– Different kinds of image enhancement
– Histograms
– Histogram equalisation
– Histogram specification

Next time we will start to look at spatial filtering and neighbourhood operations