Practical Image and Video Processing Using MATLAB®

Chapter 9 – Histogram processing
What will we learn?

- What is the histogram of an image?
- How can the histogram of an image be computed?
- How much information does the histogram provide about the image?
- What is histogram equalization and what happens to an image whose histogram is equalized?
- How can the histogram be modified through direct histogram specification and what happens to an image when we do it?
- What other histogram modification techniques can be applied to digital images and what is the result of applying such techniques?
What is a histogram?

- The histogram of a monochrome image is a graphical representation of the frequency of occurrence of each gray level in the image.

- The data structure that stores the frequency values is a 1D array of numerical values, $h$, whose individual elements store the number (or percentage) of image pixels that correspond to each possible gray level.
What is a histogram?

\[ h(k) = n_k = \text{card}\{(x, y) | f(x, y) = k\} \]

where:

\( k = 0, 1, \ldots, L - 1 \), where \( L \) is the number of gray levels of the digitized image; and

\( \text{card}\{\cdots\} \) denotes the cardinality of a set, i.e. the number of elements in that set \((n_k)\).

A normalized histogram can be mathematically defined as:

\[ p(r_k) = \frac{n_k}{n} \]

where:

\( n = \text{total number of pixels in the image} \); and

\( p(r_k) = \text{probability (percentage) of the } k\text{-th gray level } (r_k). \)
Example 9.1: histogram for a hypothetical image containing 128×128 pixels and 8 gray levels.

<table>
<thead>
<tr>
<th>Gray level ( r_k )</th>
<th>( n_k )</th>
<th>( p(r_k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1120</td>
<td>0.068</td>
</tr>
<tr>
<td>1</td>
<td>3214</td>
<td>0.196</td>
</tr>
<tr>
<td>2</td>
<td>4850</td>
<td>0.296</td>
</tr>
<tr>
<td>3</td>
<td>3425</td>
<td>0.209</td>
</tr>
<tr>
<td>4</td>
<td>1995</td>
<td>0.122</td>
</tr>
<tr>
<td>5</td>
<td>784</td>
<td>0.048</td>
</tr>
<tr>
<td>6</td>
<td>541</td>
<td>0.033</td>
</tr>
<tr>
<td>7</td>
<td>455</td>
<td>0.028</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>16384</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Examples of images and their histograms (Example 9.2)

In MATLAB: `imhist`
Examples of images and their histograms (Example 9.2)
Examples of images and their histograms (Example 9.2)
Examples of images and their histograms (Example 9.2)
Interpreting image histograms

- Histograms have become a popular tool for conveying image statistics and helping determine certain problems in an image.
- A histogram carries significant qualitative and quantitative information about the corresponding image (e.g., minimum, average, and maximum gray level values, dominance of bright or dark pixels, etc.).
- A histogram is not enough to draw qualitative conclusions about the overall quality of the image, presence or absence of noise, etc.
Interpreting image histograms

- Although a histogram provides the frequency distribution of gray levels in an image, it tells us nothing about the spatial distribution of the pixels whose gray levels are represented in the histogram.
- Histograms can be used whenever a statistical representation of the gray level distribution in an image is desired.
- Histograms can also be used to enhance or modify the characteristics of an image, particularly its contrast.
Histogram equalization

- Example 9.3:
Histogram equalization

• In MATLAB: `histeq`

• Example 9.4:

```matlab
I = imread('sydney_low_contrast.png');
I = im2double(I);
J = histeq(I);
figure, subplot(2,2,1), imshow(I), ...
   subplot(2,2,2), imshow(J), ...
   subplot(2,2,3), imhist(I), ylim('auto'),...
   subplot(2,2,4), imhist(J), ylim('auto')
```
Histogram equalization

- Example 9.4:

(a) [Image of a grayscale image of Sydney Opera House with a corresponding histogram showing a peak at the middle values.]

(b) [Histogram showing a peak at the middle values.]

(c) [Image of Sydney Opera House with a histogram below it showing a more uniform distribution compared to (b).]

(d) [Histogram showing a more uniform distribution compared to (b).]
Global vs. local histogram equalization

- In MATLAB: `histeq` and `adapthisteq`

- Example 9.5:

```matlab
I = imread('coins.png');
figure, subplot(1,2,1), imshow(I), ...
    subplot(1,2,2), imhist(I), ylim('auto')

J = histeq(I);
figure, subplot(1,2,1), imshow(J), ...
    subplot(1,2,2), imhist(J), ylim('auto')

K = adapthisteq(I);
figure, subplot(1,2,1), imshow(K), ...
    subplot(1,2,2), imhist(K), ylim('auto')
```
Global vs. local histogram equalization

- **Example 9.5:**

  ![Image](image_url)
  - (a) Original image
  - (b) Histogram of (a)
  - (c) Local histogram equalization
  - (d) Histogram of (c)
  - (e) Global histogram equalization
  - (f) Histogram of (e)
Direct histogram specification

- Example 9.6:

Original

Desired

Result
Direct histogram specification

- In MATLAB: \texttt{histeq}

Example 9.7:

\begin{verbatim}
I = imread('sydney_low_contrast.png');
Id = im2double(I);
figure, imhist(Id), ylim('auto'), ...
    title ('Original histogram');

des_hist = uint8(zeros(1,256));
des_hist(1:128) = linspace(256,0,128);
des_hist(129:end) = linspace(0,256,128);
x_axis = 0:255;
figure, bar(x_axis, des_hist), axis tight,
    ...
    title('Desired histogram');

hgram = im2double(des_hist);
Jd = histeq(Id,hgram);
figure, imhist(Jd), ylim('auto'), ...
    title ('Resulting histogram');
\end{verbatim}
Direct histogram specification

- **Example 9.7:**

![Sydney Opera House images with histograms](image)
Direct histogram specification

- Interactive histogram matching tool (ihmdemo)
Other histogram modification techniques

- Histogram sliding (Example 9.8):

(a) (b) (c)

(d) (e) (f)
Histogram sliding

- In MATLAB: `imadd` and `imsubtract`

- Example 9.8:

```matlab
I = imread('schonbrunn_gray_low_contrast.png');
figure, imhist(I), ylim('auto'), title ('Original histogram');

I2 = imadd(I, 50);
figure, imhist(I2), ylim('auto'), ...
    title ('Sliding to the right by 50');

I3 = imsubtract(I,50);
figure, imhist(I3), ylim('auto'), ...
    title ('Sliding to the left by 50');
```
Other histogram modification techniques

- Histogram stretching (Example 9.9):

\[ s = \frac{r - r_{\text{min}}}{r_{\text{max}} - r_{\text{min}}} \cdot (L - 1) \]

(a)  
(b)  
(c)  
(d)
Other histogram modification techniques

- Histogram shrinking (Example 9.10):

\[
s = \left[ \frac{s_{\text{max}} - s_{\text{min}}}{r'_{\text{max}} - r'_{\text{min}}} \right] (r - r'_{\text{min}}) + s_{\text{min}}
\]
Histogram stretching and shrinking

- **In MATLAB:** `imadjust`

```matlab
%% Histogram stretching
I = imread('schonbrunn_gray_low_contrast.png');
figure, imhist(I), ylim('auto'), title ('Original histogram');
I2 = imadjust(I);
figure, imhist(I2), ylim('auto'), title ('After histogram stretching');
figure, subplot(1,2,1), imshow(I), subplot(1,2,2), imshow(I2)

%% Histogram shrinking
I = imread('schonbrunn_gray.png');
figure, imhist(I), ylim('auto'), title ('Original histogram');
Id = im2double(I);
Jd = imadjust(Id, [], [49/255 140/255]);
J = uint8(255.*Jd);
figure, imhist(J), ylim('auto'), title ('After histogram shrinking');
figure, subplot(1,2,1), imshow(I), subplot(1,2,2), imshow(J)
```
Hands-on

- Tutorial 9.1: Image histograms (page 188)
- Tutorial 9.2: Histogram equalization and specification (page 191)
- Tutorial 9.3: Other histogram modification techniques (page 195)