What is an image histogram

Count: 1920000
Mean: 118.848
StdDev: 59.179
Min: 0
Max: 251
Mode: 184 (30513)
How to define a histogram

$$h(i) = \text{the number of pixels in } I \text{ with the intensity value } i$$

Or, formally,

$$h(i) = \text{card}\{(u, v) \mid I(u, v) = i\}$$
Are histograms unique?

Three images with same histogram
Histogram interpretation

$\alpha_{\text{low}}$  

Contrast Range  

$\alpha_{\text{high}}$

linear

logarithmic

0  

256
Exposure

(a) Under exposed
(b) Normal exposure
(c) Over exposed
Contrast

(a) Low  
(b) Normal  
(c) High
Dynamic range

(a) High

(b) Low

(c) Extremely low
How to compute a histogram in ImageJ

```java
public class Compute_Histogram implements PlugInFilter {

    public int setup(String arg, ImagePlus img) {
        return DOES_8G + NO_CHANGES;
    }

    public void run(ImageProcessor ip) {
        int[] H = new int[256]; // histogram array
        int w = ip.getWidth();
        int h = ip.getHeight();

        for (int v = 0; v < h; v++) {
            for (int u = 0; u < w; u++) {
                int i = ip.getPixel(u,v);
                H[i] = H[i] + 1;
            }
        }

        // histogram H[] can now be used
    }
}
```
Cumulative histogram

\[ H(i) = \sum_{j=0}^{i} h(j) \quad \text{for} \quad 0 \leq i < K \]

\[ H(i) = \begin{cases} 
  h(0) & \text{for } i = 0 \\
  H(i-1) + h(i) & \text{for } 0 < i < K 
\end{cases} \]

\[ H(K-1) = \sum_{j=0}^{K-1} h(j) = M \cdot N \]
Example of cumulative histogram
Normalized histogram

- Suppose $h(i)$ is a histogram
- It’s normalized version is defined as
  \[ p(i) = \frac{h(i)}{\sum_{i=0}^{K-1} h(i)} \]
- One can think of the normalized histogram as a probability mass function: $p(i)$ means the probability of a pixel value to be $i$
Binning for histogram

Sometimes, we need to create bins for histograms. Suppose, we have a 14 bit image. So, the range of values is 0 to 16384. If we want to create a histogram of 256 bins, then it would look like:

\[
\begin{align*}
  h(0) & \gets 0 \leq I(u, v) < 64 \\
  h(1) & \gets 64 \leq I(u, v) < 128 \\
  h(2) & \gets 128 \leq I(u, v) < 192 \\
  \vdots & \vdots \vdots \vdots \\
  h(j) & \gets a_j \leq I(u, v) < a_{j+1} \\
  \vdots & \vdots \vdots \vdots \\
  h(255) & \gets 16320 \leq I(u, v) < 16384
\end{align*}
\]

Notice that the bin widths are same.
Color image histogram

(a) 

(b) $h_{Luma}$

(c) $R$

(d) $G$

(e) $B$

(f) $h_R$

(g) $h_G$

(h) $h_B$
Matching histograms and some applications

• Histogram matching has many applications in image processing
  – Image segmentation
  – Tracking
  – Content-based image retrieval
  – ...

• A quick way for content-based image retrieval can be based on histogram matching
  – In a database of images, find out those that have histograms similar to the histogram of a query image
Histogram match metrics

• Before matching two histograms, they are converted into normalized histograms
• Several metrics exist for matching normalized histograms
  – Bhattacharya coefficient
  – Kullback-Liebler divergence
  – **Diffusion distance**
    (http://www.ist.temple.edu/~hbling/publication/Ling&Okada06cvpr.pdf)
  – ....
Matching two normalized histograms with Bhattacharya coefficient

• Suppose two \( p(i) \) and \( q(i) \) are two normalized histograms

• Bhattacharya coefficient is defined as

\[
BC(p,q) = \sum_{i=0}^{K-1} \sqrt{p(i)q(i)}
\]

• For a perfect match \( BC \) is 1, for a complete mismatch \( BC \) is 0

• A higher \( BC \) value implies a better match
Bhattacharyya coefficient: Examples

Notice that BC value is higher for the similar pair of images