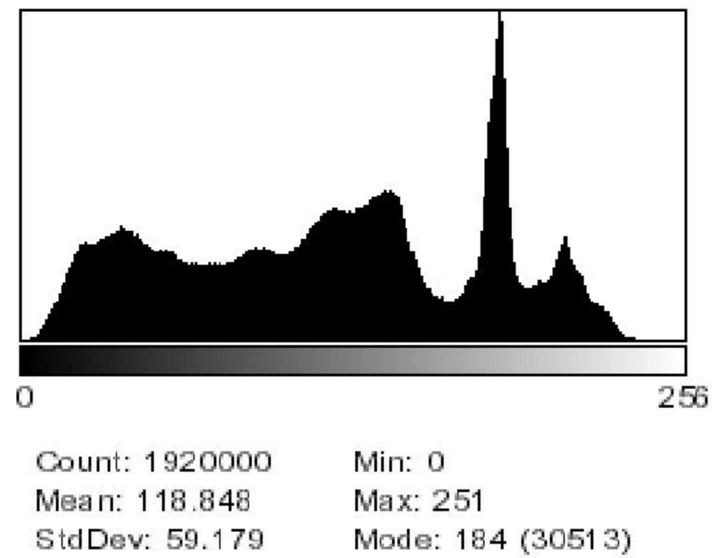
Histograms

CMPUT 206

Instructor: Nilanjan Ray

Source: http://www.imagingbook.com/

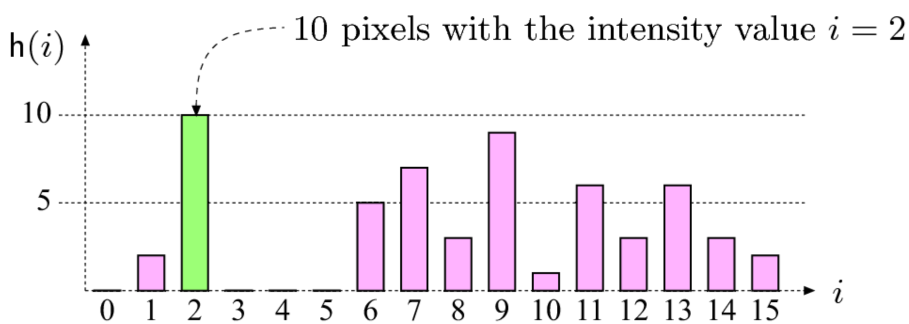
What is an image histogram



How to define a histogram

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

Or, formally, $$\mathsf{h}(i) = \mathrm{card}\{(u,v) \mid I(u,v) = i\}$$



$\mathsf{h}(i)$	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Example: Histogram

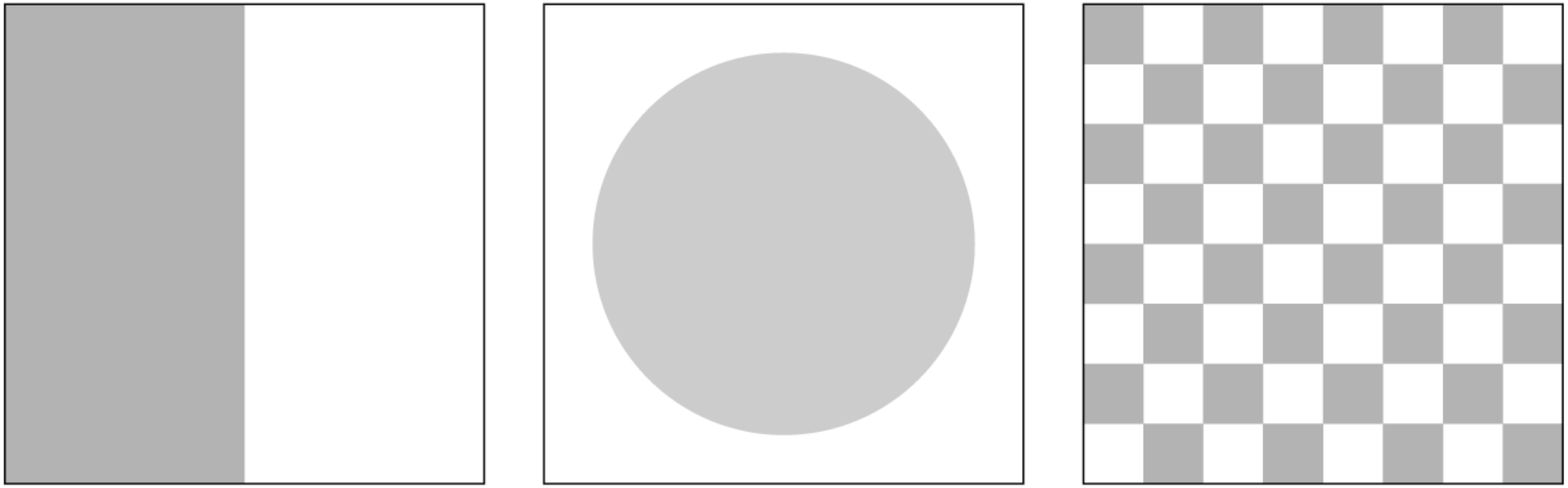
- Consider a 3-by-4 image with 8 gray levels

3	0	1	2
4	3	6	7
3	2	1	4

- Its histogram h is as follows:

Gray level i	0	1	2	3	4	5	6	7
Histogram h[i]	1	2	2	3	2	0	1	1

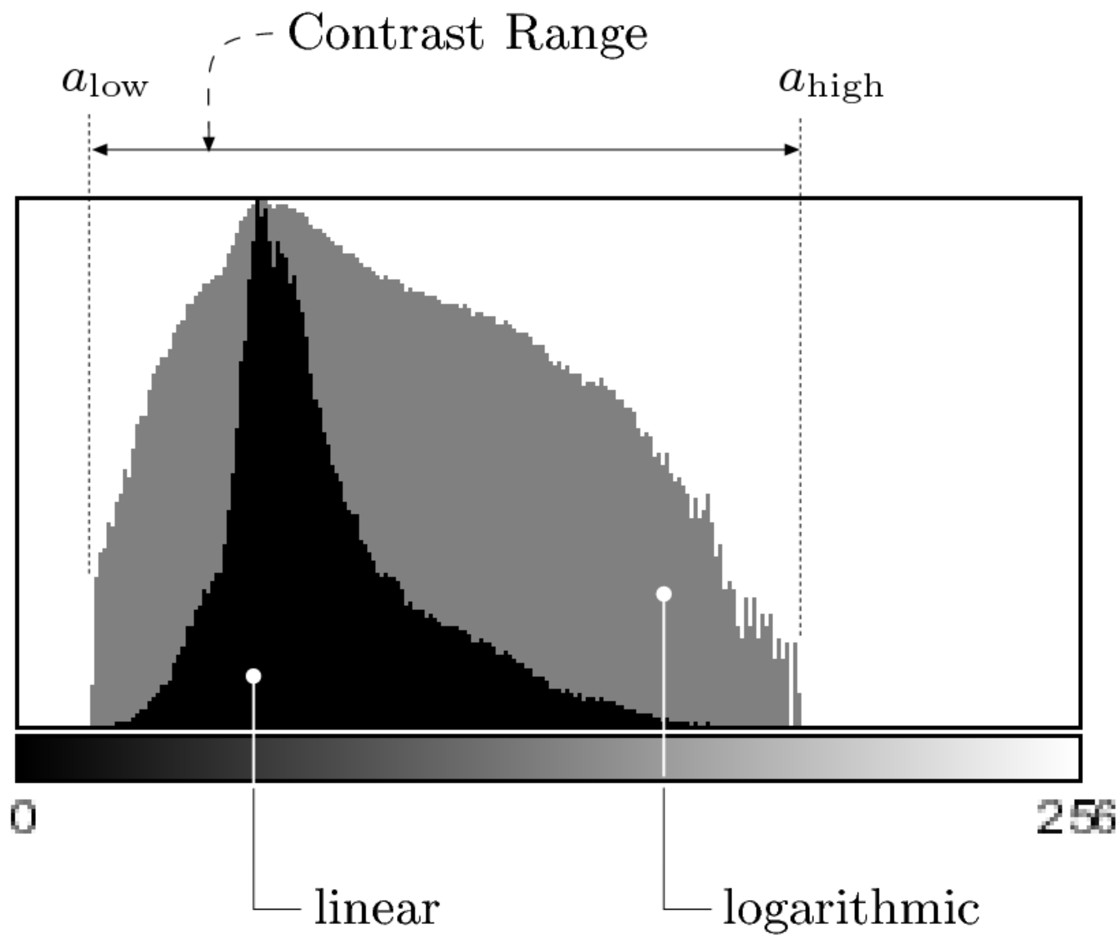
Are histograms unique?



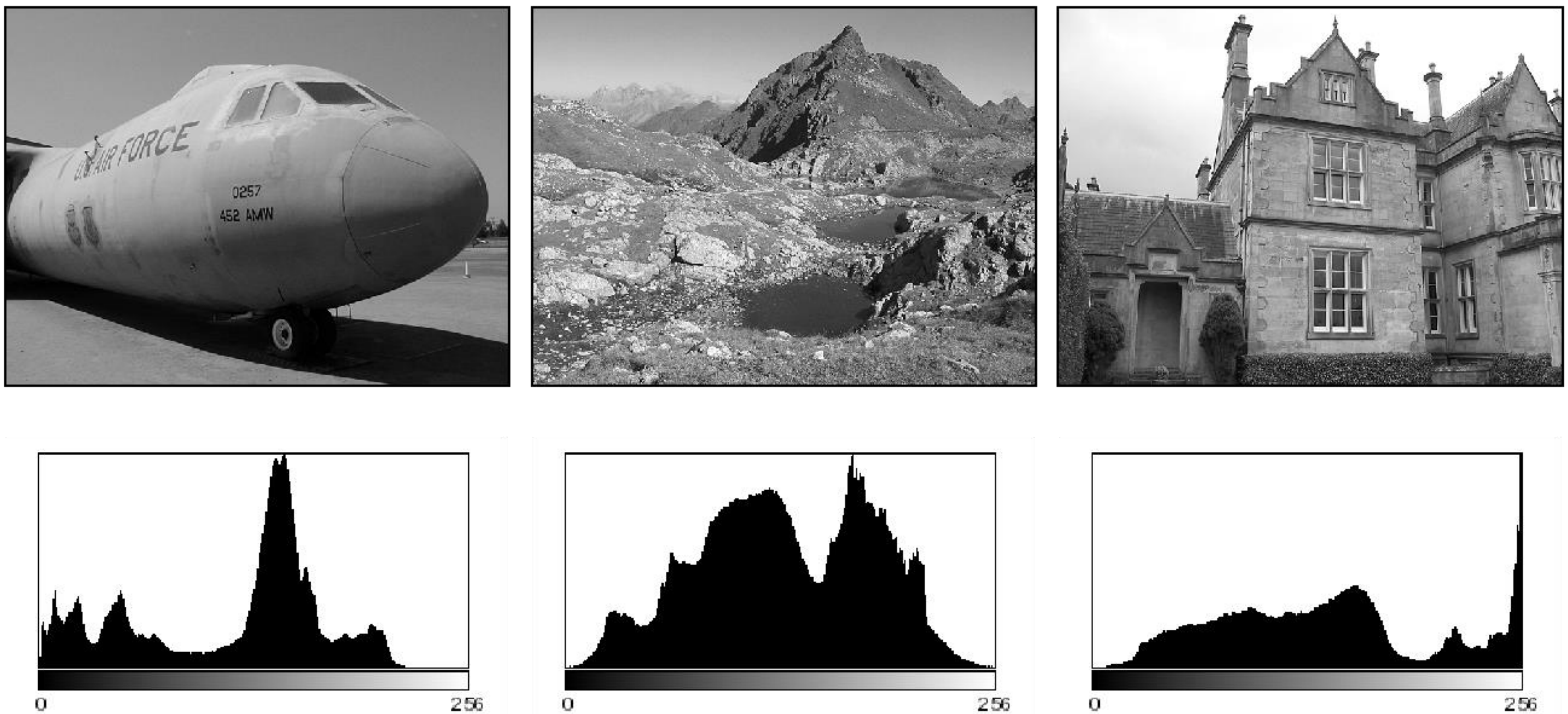
Three images with same histogram

Going from an image to its histogram, what information is lost?

Histogram interpretation



Exposure



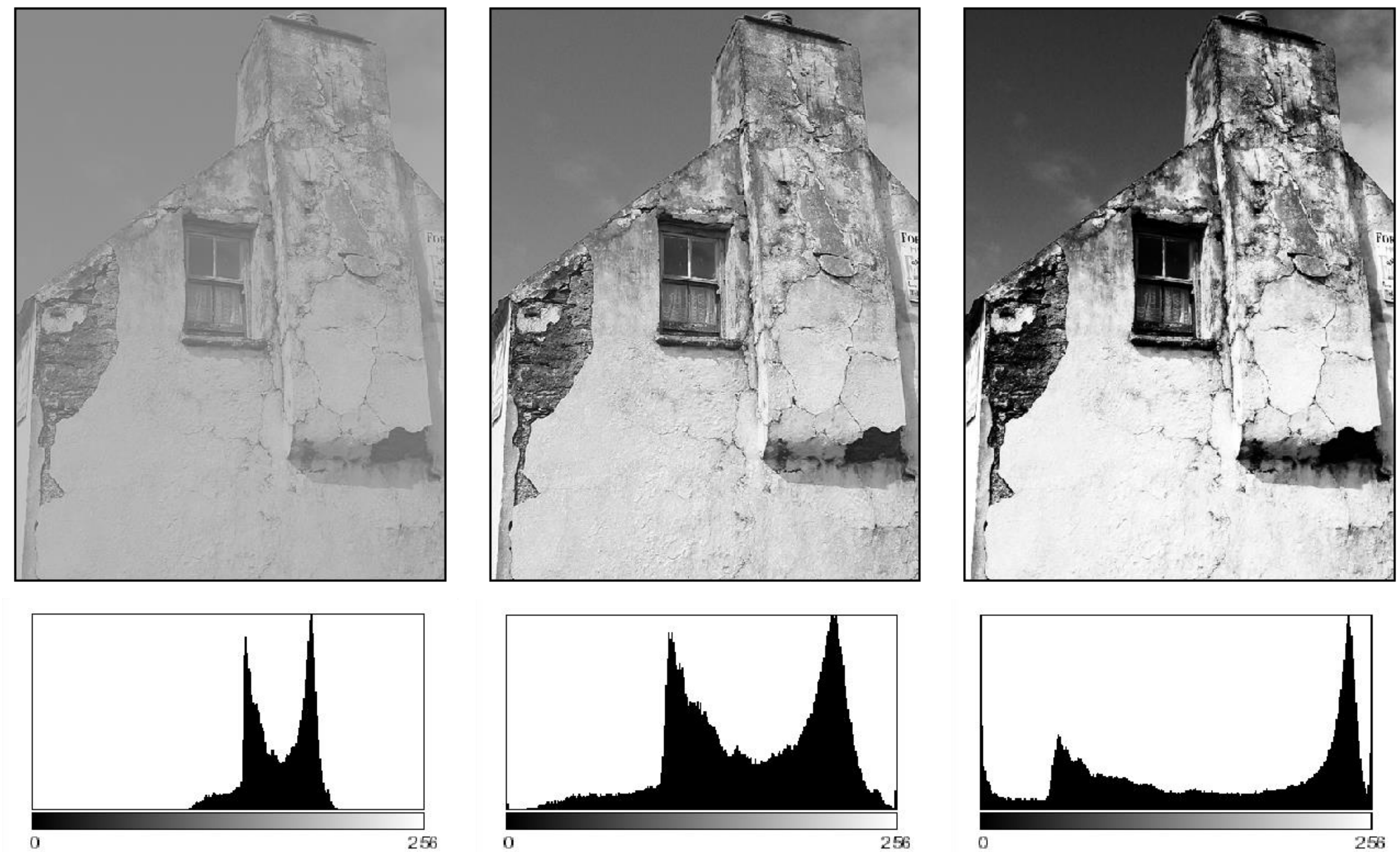
(a) (b) (c)

Under exposed

Normal exposure

Over exposed

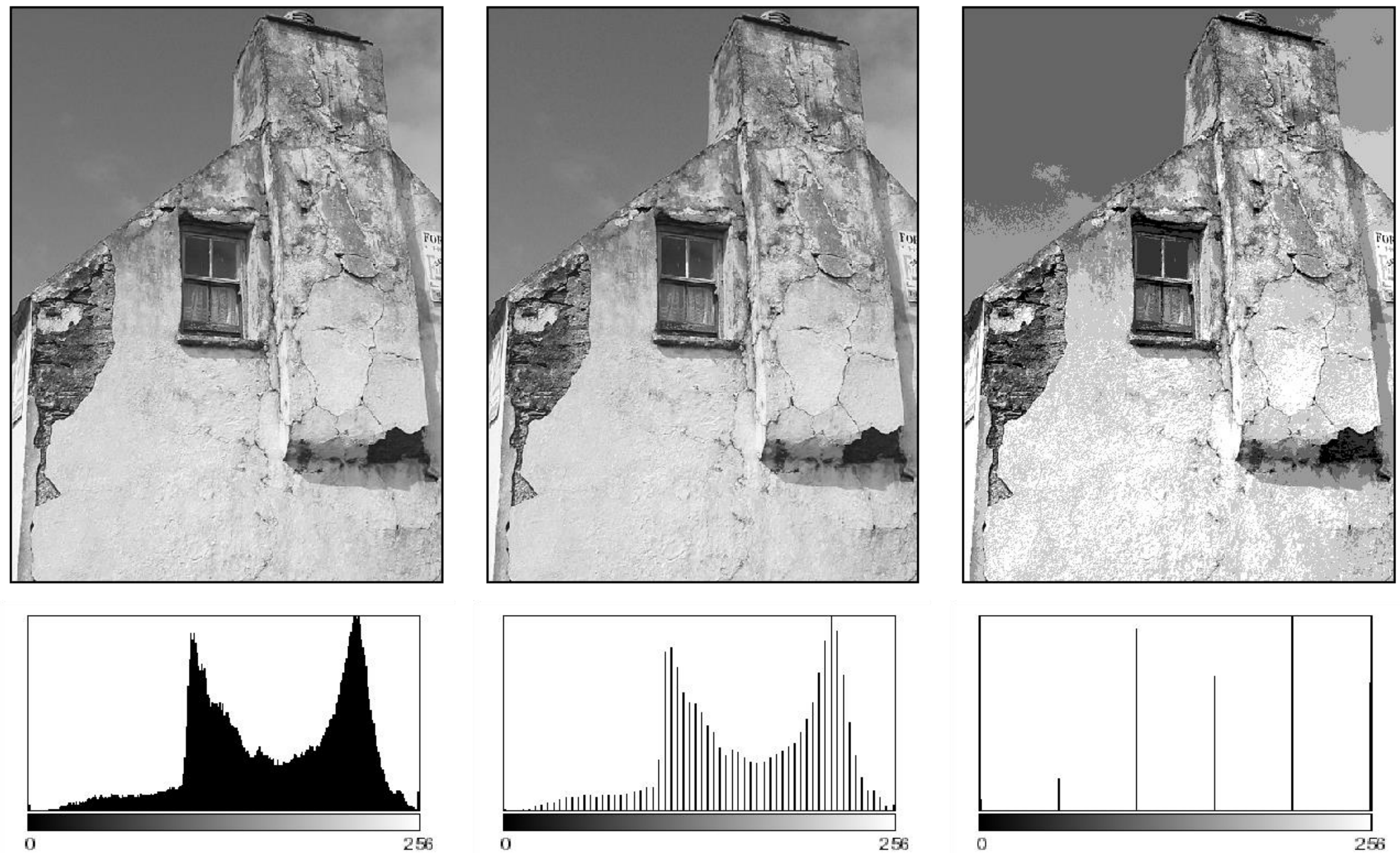
Contrast



(a) (b) (c)

Low Normal High

Dynamic range



(a) (b) (c)

High Low Extremely low

How to compute a histogram

- Let’s compute histogram for a L gray level image
- Initialize histogram array: h[n] = 0, n=0,…,L-1
- Let height and width be the number of row and columns of image I, respectively
- For i from 0 to height-1
  - For j from 0 to width-1
    - h[I[i, j]] += 1

Cumulative histogram

$$\mathsf{H}(i) = \sum_{j=0}^{i} \mathsf{h}(j) \quad \text{for} \;\; 0 \leq i < K$$

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

$$\mathsf{H}(K-1) = \sum_{j=0}^{K-1} \mathsf{h}(j) = M \cdot N$$

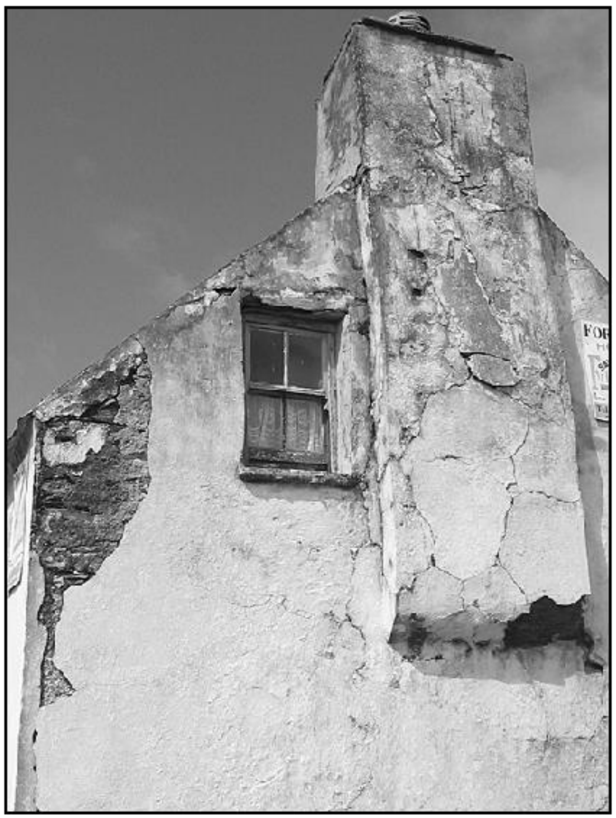
Example: Cumulative histogram

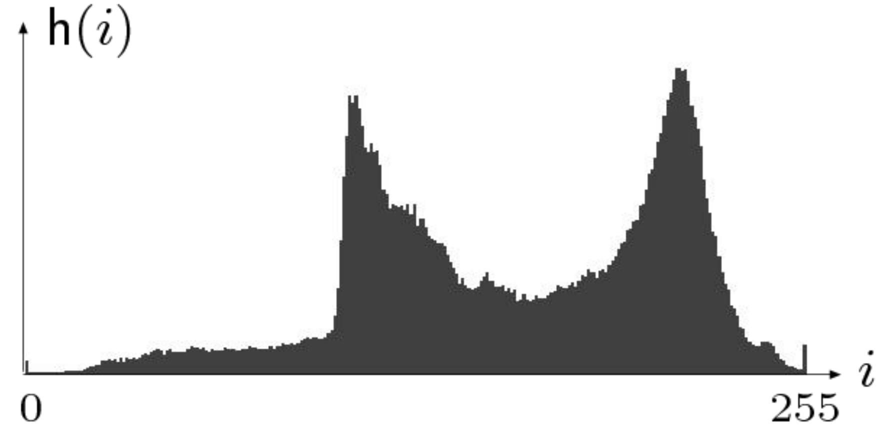
8 gray level Image

3	0	1	2
4	3	6	7
3	2	1	4

Gray level i	0	1	2	3	4	5	6	7
Histogram h[i]	1	2	2	3	2	0	1	1
Cumulative histogram H[i]	1	3	5	8	10	10	11	12

Example of cumulative histogram





Normalized histogram

- Suppose *h*(*i*) is a histogram
- It's normalized version is defined as

$$p(i) = h(i) / \sum_{i=0}^{K-1} h(i) = h(i) / (MN)$$

- One can think of the normalized histogram as a probability mass function: *p*(*i*) means the probability of a pixel value to be *i*

Normalized cumulative histogram

- Suppose $H(i)$ is a cumulative histogram
- It's normalized version is defined as: $H(i)/(MN)$

Example: Normalized histogram and normalized cumulative histogram

8 gray level Image

3	0	1	2
4	3	6	7
3	2	1	4

Gray level i	0	1	2	3	4	5	6	7
Histogram h[i]	1	2	2	3	2	0	1	1
Cumulative histogram H[i]	1	3	5	8	10	10	11	12
Normalized histogram	1/12	2/12	2/12	3/12	2/12	0	1/12	1/12
Normalized cumulative histogram	1/12	3/12	5/12	8/12	10/12	10/12	11/12	1

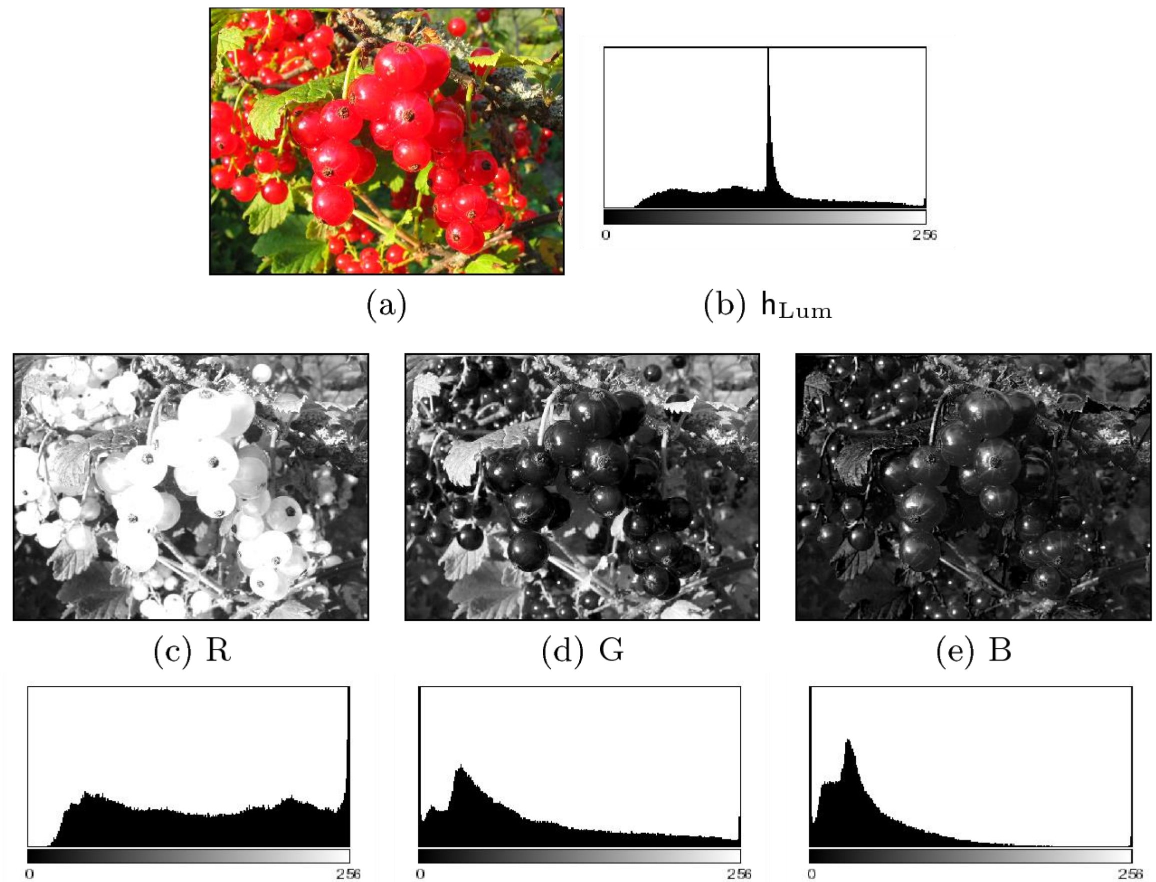
Binning for histogram

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:

$$
\begin{array}{lcrcl}
\mathsf{h}(0) & \leftarrow & 0 & \leq I(u,v) < & 64 \\
\mathsf{h}(1) & \leftarrow & 64 & \leq I(u,v) < & 128 \\
\mathsf{h}(2) & \leftarrow & 128 & \leq I(u,v) < & 192 \\
\vdots & & \vdots & \vdots & \vdots \\
\mathsf{h}(j) & \leftarrow & a_j & \leq I(u,v) < & a_{j+1} \\
\vdots & & \vdots & \vdots & \vdots \\
\mathsf{h}(255) & \leftarrow & 16320 & \leq I(u,v) < & 16384
\end{array}
$$

Notice that the bin widths are same.

Color image histogram



(a) (b) h_{Lum}

(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 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)
  - ….

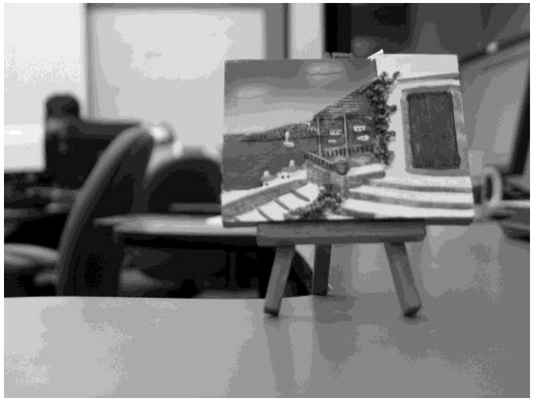
Matching two normalized histograms with Bhattacharya coefficient

- 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

Bhattacharya coefficient: Examples



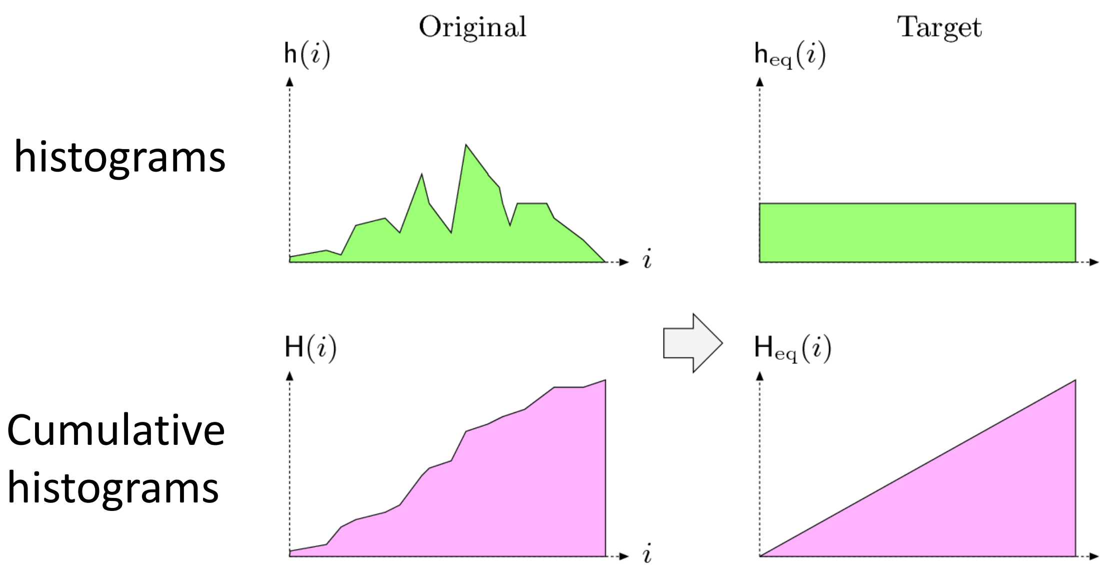
0.9161

0.9802

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

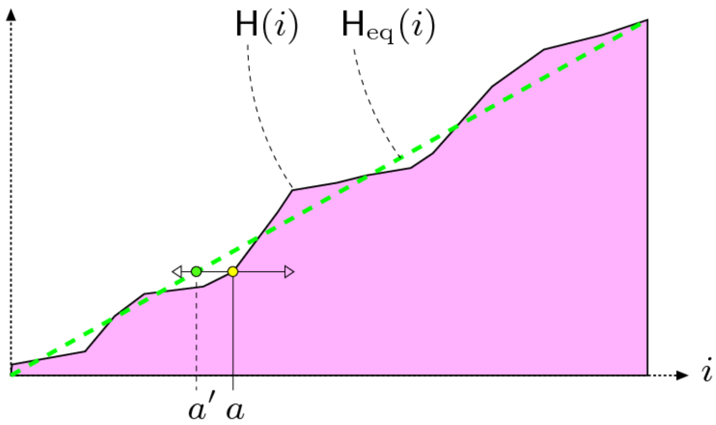
Histogram equalization

- Also known as “histogram flattening”



Histogram equalization…

Cumulative histograms



$$a' = floor[\frac{K-1}{MN} H(a) + 0.5]$$

Can we derive this formula?

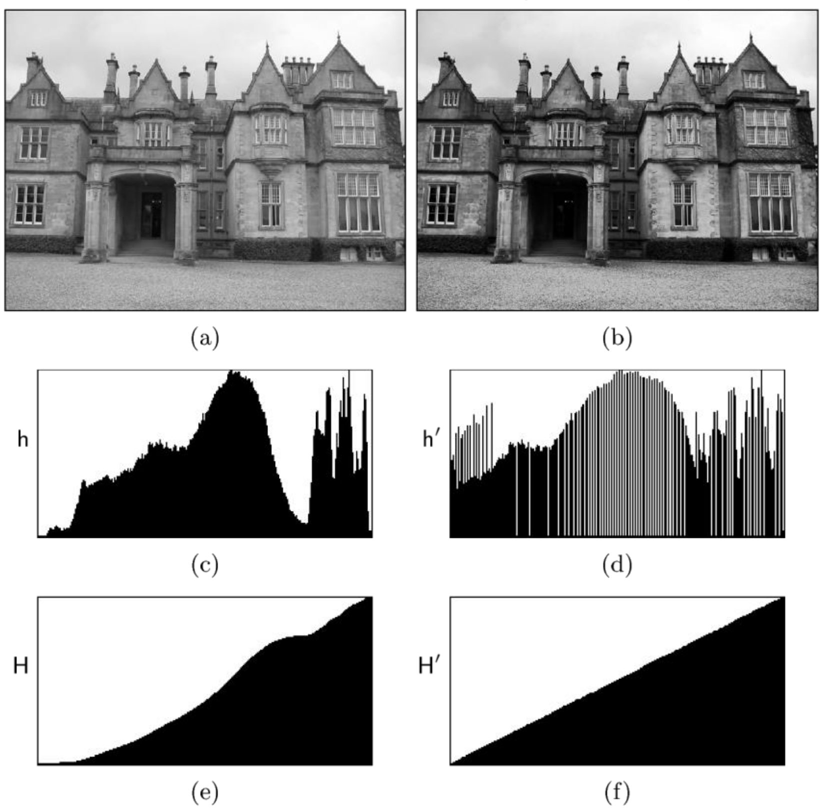
Histogram equalization: example

Original image

The image after histograms equalization

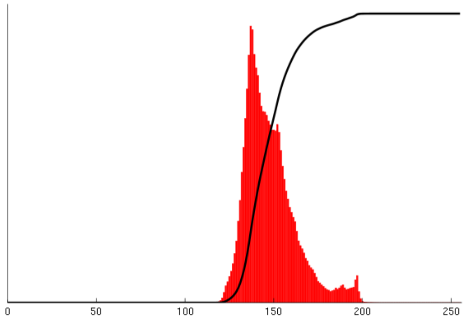
histograms

Cumulative histograms

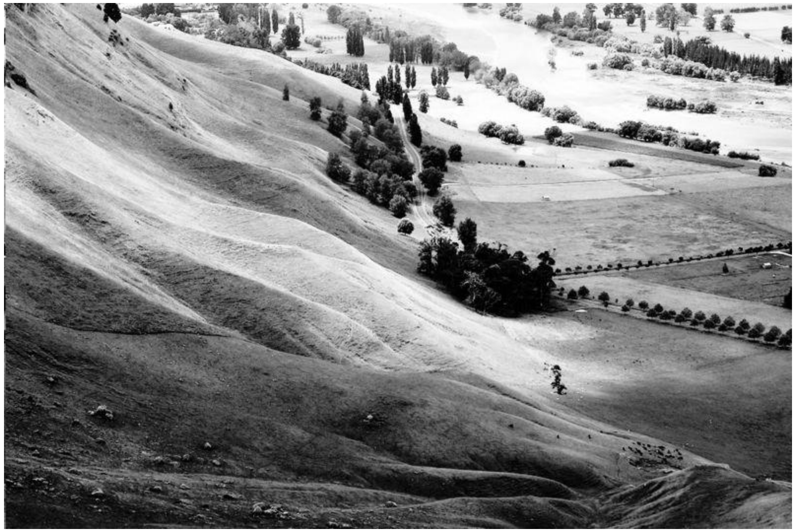


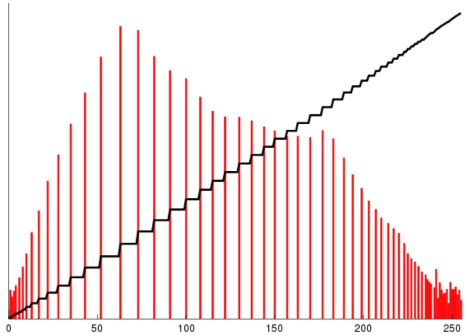
A histogram equalization example from Wikipedia

Original image



After histogram equalization





Histograms (red) and cumulative histograms (black)

Histogram specification

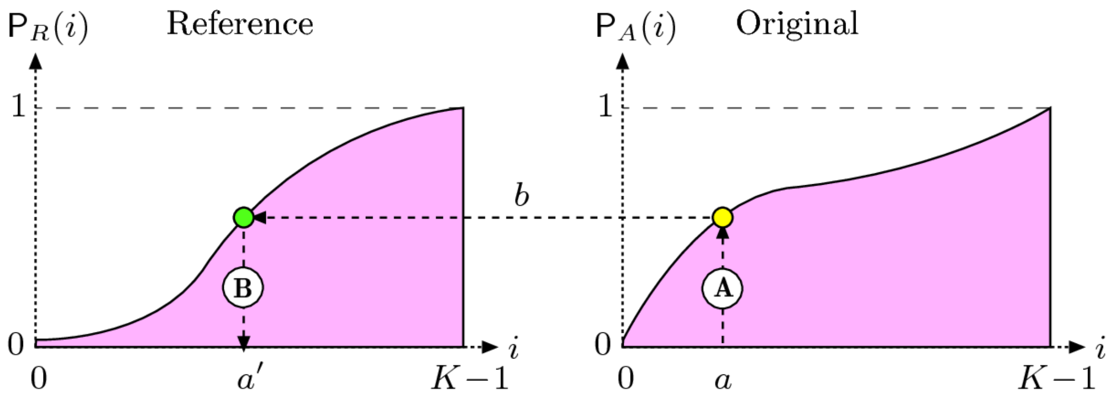
$$\text{Prob}[\text{output image pixel value} \le a'] = \text{Prob}[\text{input image pixel value} \le a]$$

$$\Rightarrow \text{P}_\text{R}(a') = \text{P}_\text{A}(a)$$

$$\Rightarrow a' = \text{P}_\text{R}^{-1}(\text{P}_\text{A}(a))$$

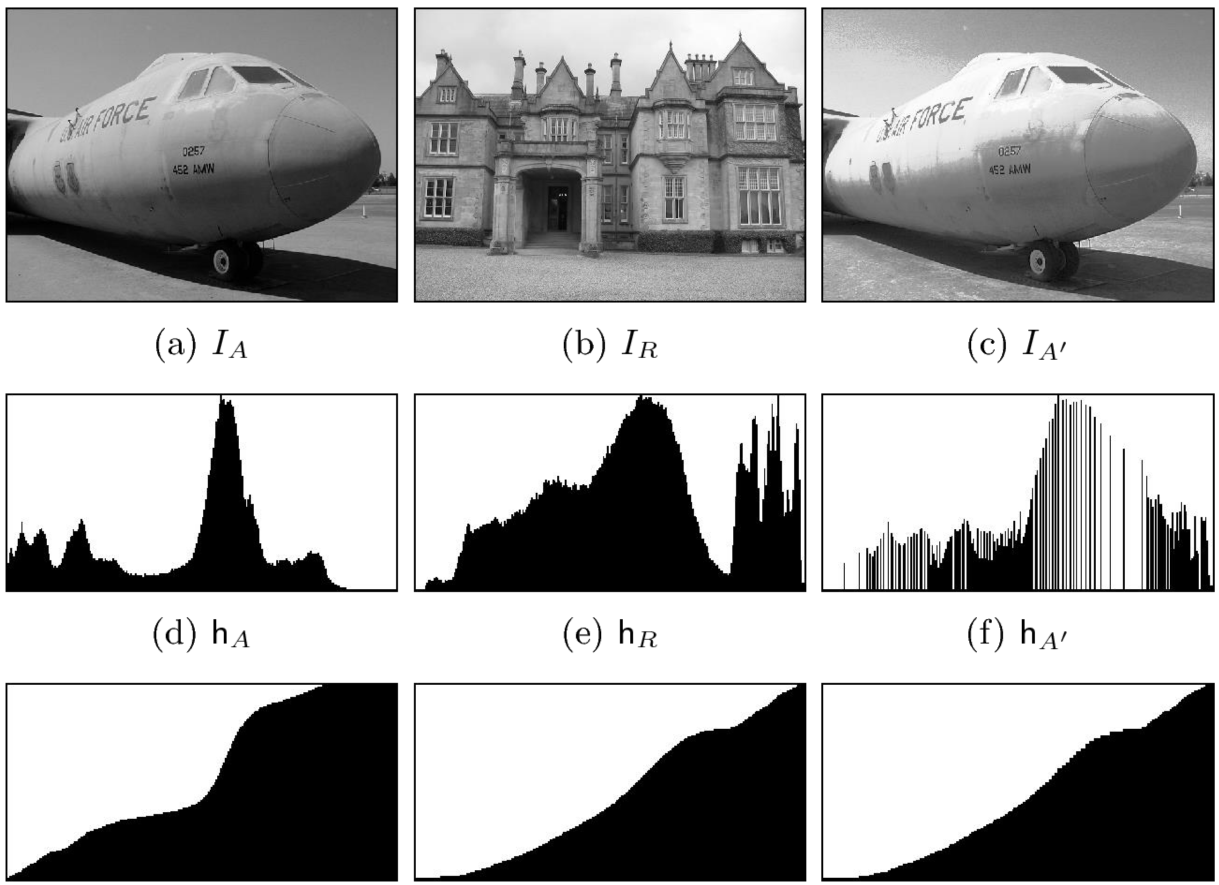
P_A and P_R are normalized cumulative histograms

Pictorially:



Histogram specification: example

Target Image | Reference Image | Modified Image



(a) I_A (b) I_R (c) $I_{A'}$

(d) h_A (e) h_R (f) $\mathsf{h}_{A'}$

(g) H_A (h) H_R (i) $\mathsf{H}_{A'}$

Local (Adaptive) histogram equalization

- Histogram equalization does not work well when the distribution of pixel values varies a lot over local windows in an image.
- So, doing histogram equalization within a sliding window is often more useful than the global histogram equalization.
- By the way, what is a sliding window?

Local histogram equalization: example

http://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_local_equalize.html