Image Processing INT3404 20

Lecturer: Nguyen Thi Ngoc Diep, Ph.D.

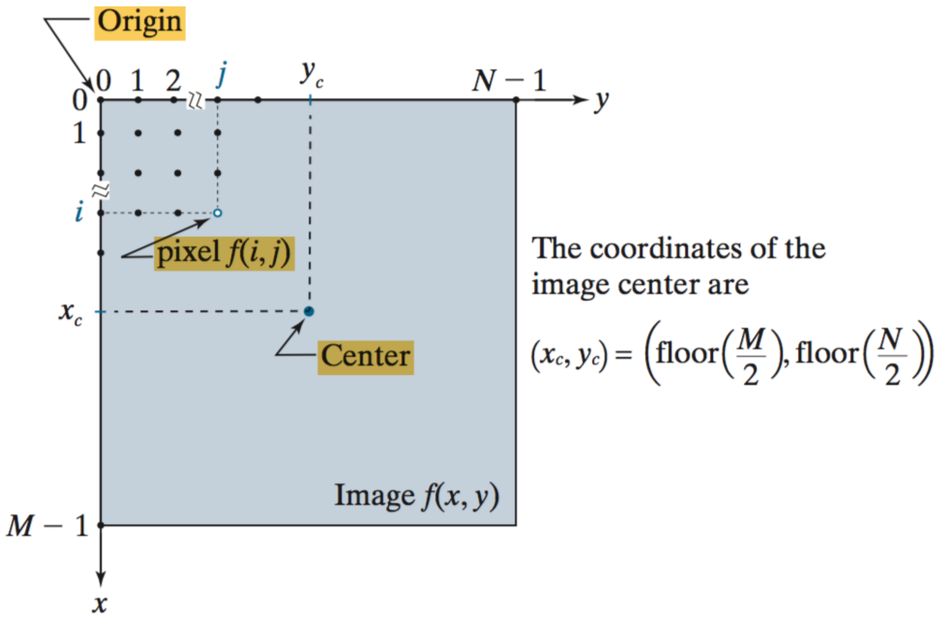
Email: ngocdiep@vnu.edu.vn

Schedule

Tuần	Nội dung	Yêu cầu đối với sinh viên (ngoài việc đọc tài liệu tham khảo)
1	Giới thiệu môn học	Cài đặt môi trường: Python 3, OpenCV 3, Numpy, Jupyter Notebook
2	Ảnh số (Digital image) – Phép toán điểm (Point operations) Làm quen với OpenCV + Python Điều chỉnh độ tương phản (Contrast adjust) – Ghép ảnh (Combining images)	
3	Histogram - Histogram equalization	Làm bài tập 1: điều chỉnh gamma tìm contrast hợp lý
4	Phép lọc trong không gian điểm ảnh (linear processing filtering)	Thực hành ở nhà
5	Phép lọc trong không gian điểm ảnh (linear processing filtering) (cont.) Thực hành: Cách tìm filters	Thực hành ở nhà
6	Thực hành: Ứng dụng của histogram; Tìm ảnh mẫu (Template matching)	Bài tập mid-term
7	Trích rút đặc trưng của ảnh Cạnh (Edge) và đường (Line) và texture	Thực hành ở nhà
8	Các phép biến đổi hình thái (Morphological operations)	Làm bài tập 2: tìm barcode
9	Chuyển đổi không gian – Miền tần số – Phép lọc trên miền tần số Thông báo liên quan đồ án môn học	Đăng ký thực hiện đồ án môn học
10	Xử lý ảnh màu (Color digital image)	Làm bài tập 3: Chuyển đổi mô hình màu và thực hiện phân vùng
11	Các phép biến đổi hình học (Geometric transformations)	Thực hành ở nhà
12	Nhiễu – Mô hình nhiễu – Khôi phục ảnh (Noise and restoration)	Thực hành ở nhà
13	Nén ảnh (Compression)	Thực hành ở nhà
14	Hướng dẫn thực hiện đồ án môn học	Trình bày đồ án môn học
15	Hướng dẫn thực hiện đồ án môn học Tổng kết cuối kỳ	Trình bày đồ án môn học

Recall week 2: Digital image

An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are *spatial* (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the *intensity* or *gray level* of the image at that point. When x, y, and the intensity values of f are all finite, discrete quantities, we call the image a *digital image*.

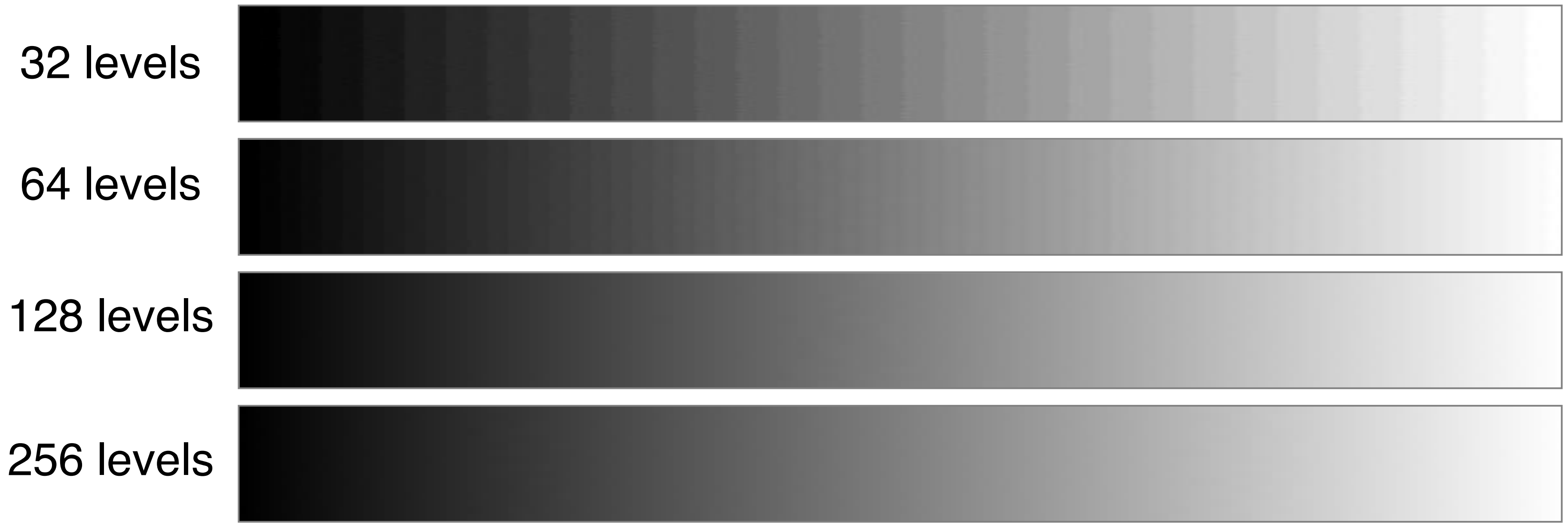

Origin
0 1 2 j y_c $N-1$
y
0
1
i
x_c
$M-1$
x
pixel $f(i, j)$
Center
Image $f(x, y)$
The coordinates of the image center are
$(x_c, y_c) = \left(\text{floor}\left(\frac{M}{2}\right), \text{floor}\left(\frac{N}{2}\right)\right)$

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

$$a_{ij} = f(i, j)$$

Recall week 2: Quantization levels

Contouring is most visible for a ramp



Conventional grayscale image has 256 levels

Image credit: Bernd Girod

Recall week 2: Point operations

- Aka: Point-processing transformations
  - Image pixel intensities are transformed into a different set of intensities based on input-output mappings
- Gamma transformation
  - Concept of “Lookup table (LUT) based mapping”
- Thresholding
- Arithmetic operations
- Set and logical operations

Week 3

Histograms

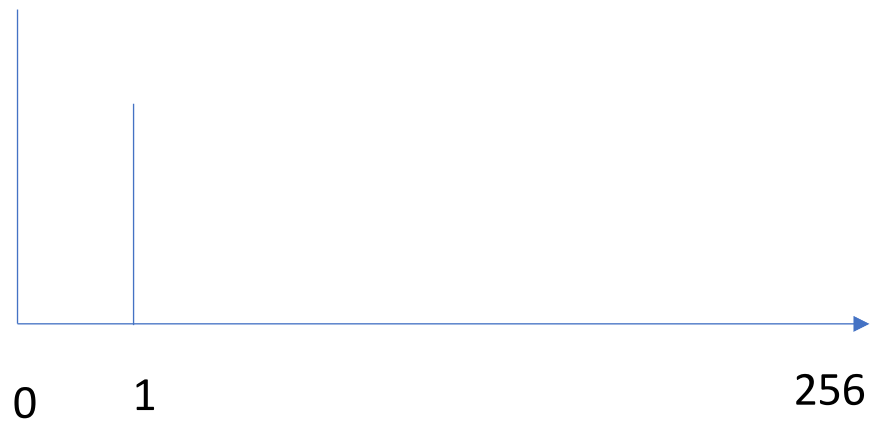
Why study histogram processing?

- Histogram manipulation is a fundamental tool in image processing
- Histograms:
  - Simple to compute
  - Suitable for fast hardware implementations
  - Popular tool for real-time image processing
- Histogram shape is related to image appearance

Histogram

- An image with L-level intensities
- r_k : intensity level k   (k = 0, 1, 2, ... , L-1)
- n_k : number of pixels with intensity r_k

Normalized histogram:



The subdivisions of the intensity scale are called histogram bins

- Unnormalized histogram:

$$h(r_k) = n_k$$

- Normalized histogram:

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}$$

M: height of image = number of image rows
N: width of image = number of image columns

The sum of $p(r_k)$ for all values of k is always 1

Histogram shape and image appearance

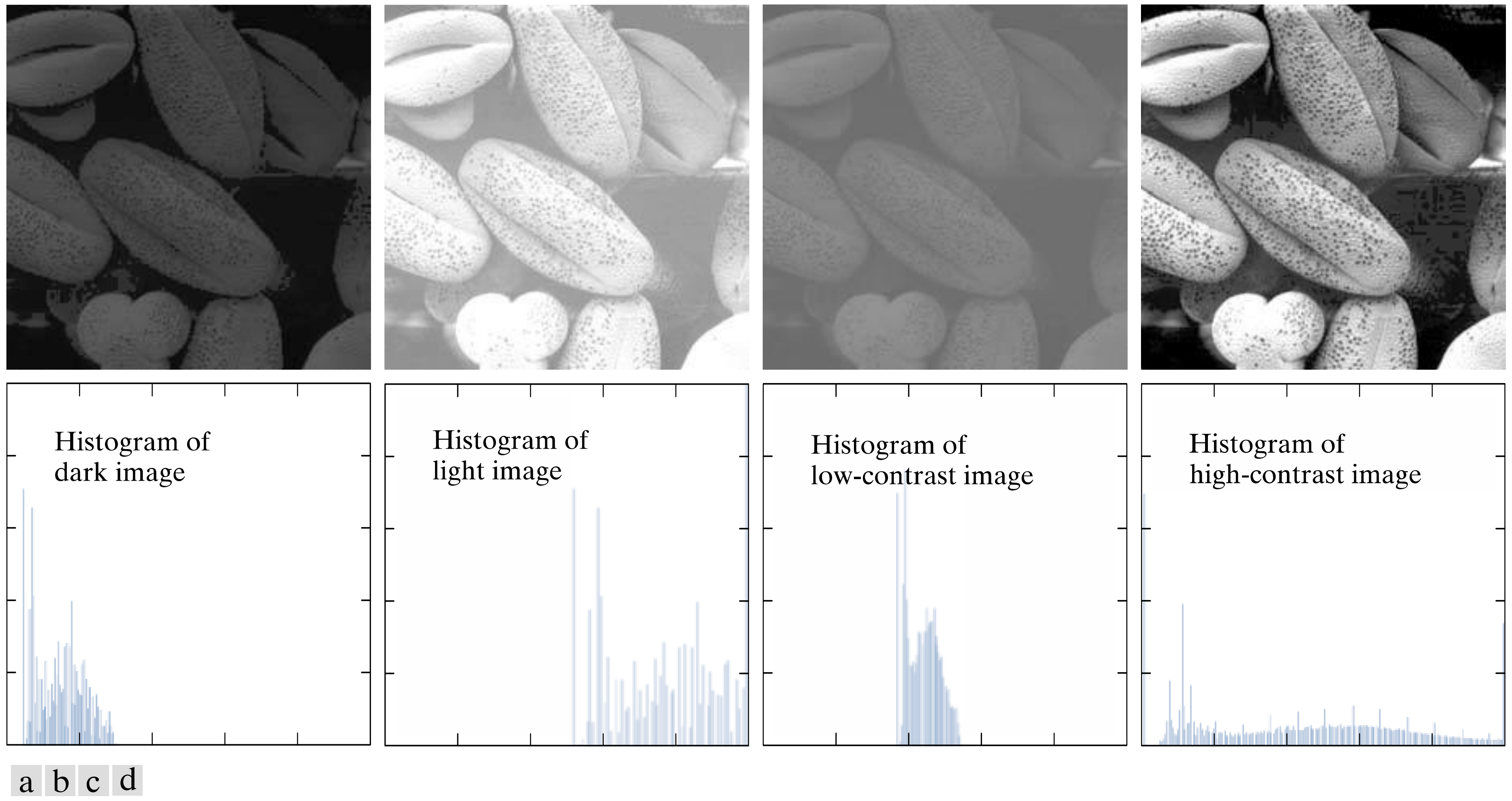
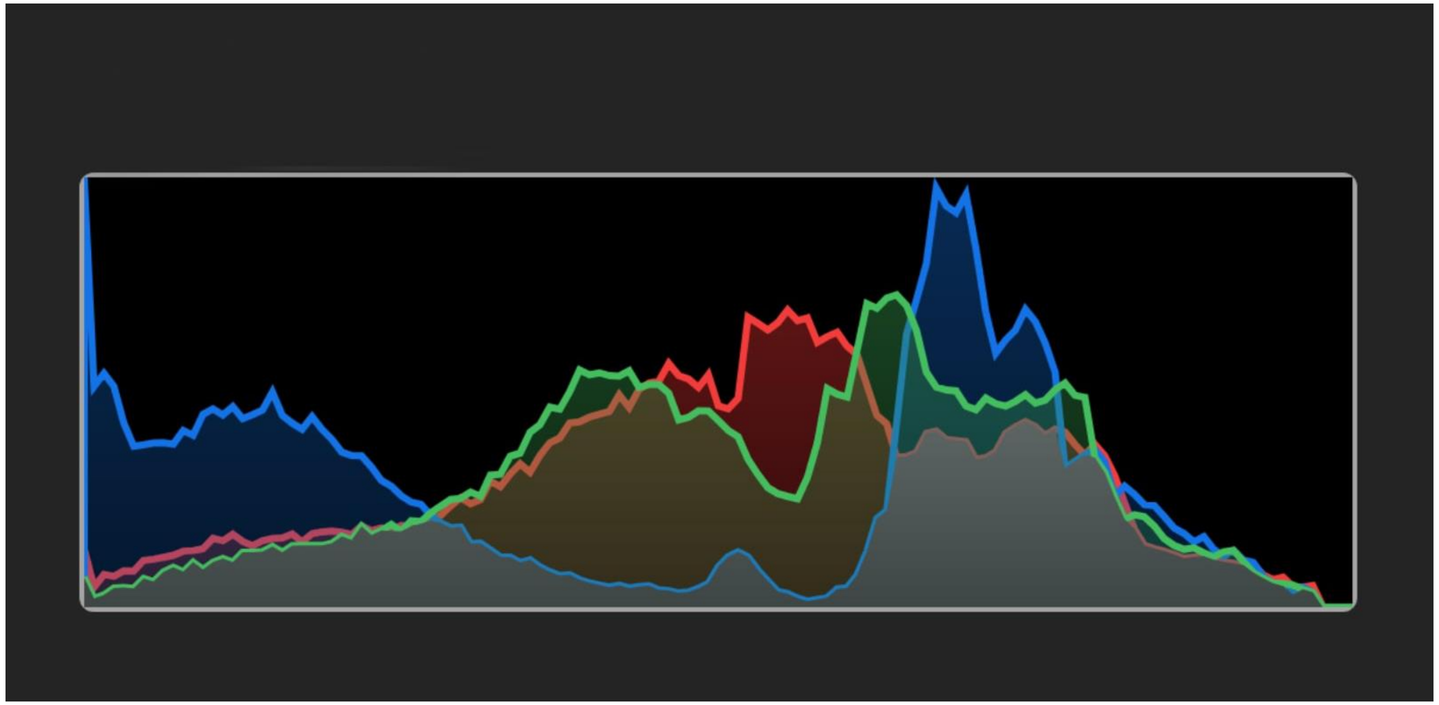


FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

Color image



Histogram equalization

(Cân bằng biểu đồ mức xám)

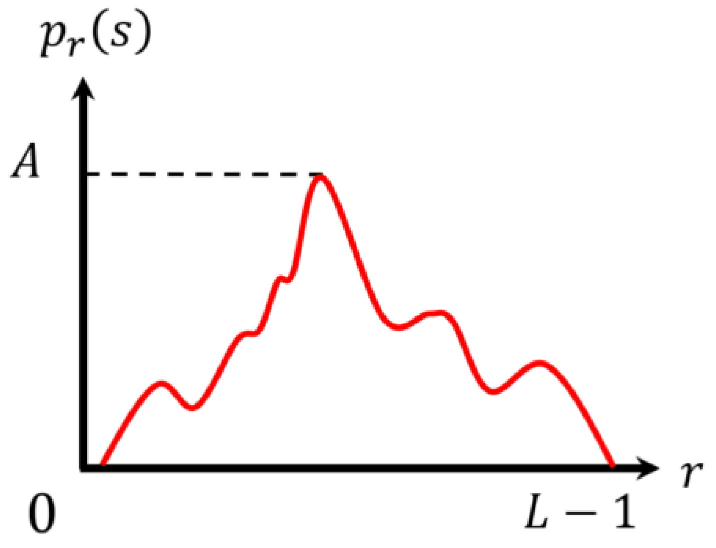
A special contrast enhancement technique

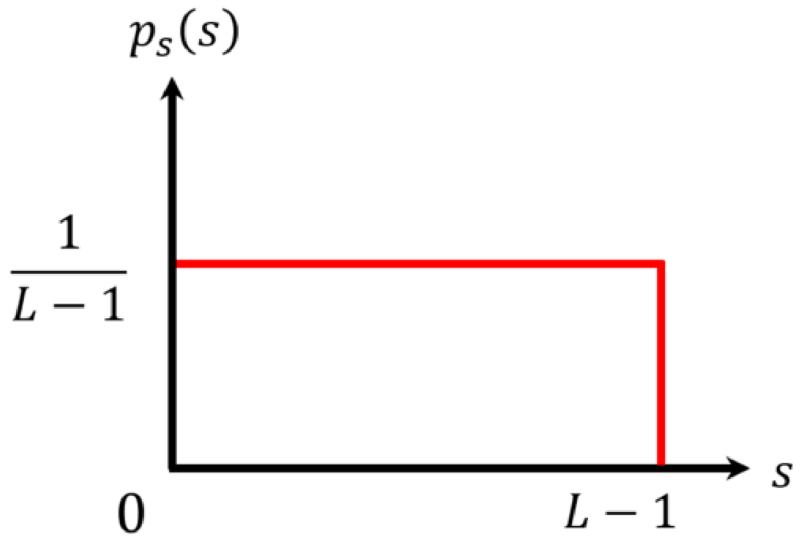
Histogram equalization

Objective: To improve the dynamic range and contrast of an image

→ Reassigning pixel intensity values such that the resulting image has a (close to) uniform distribution across its entire range of values

→ Meaning, having a flat intensity histogram





How?

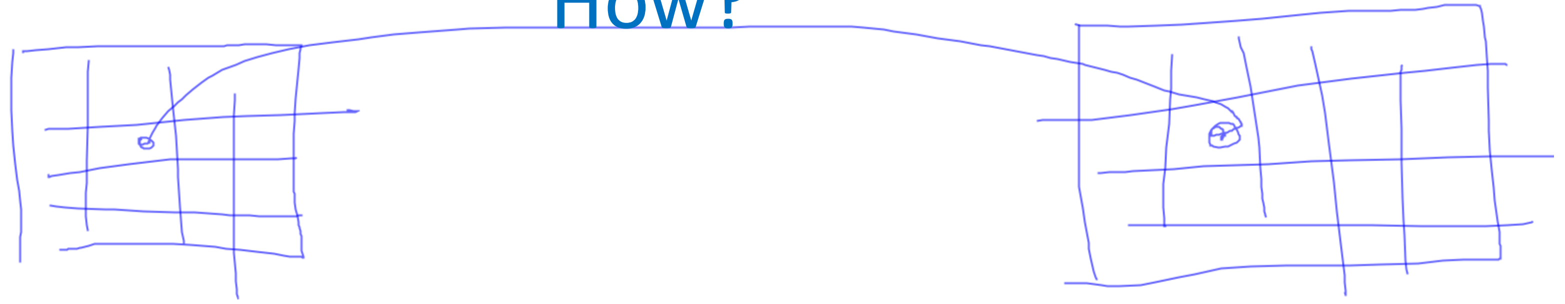
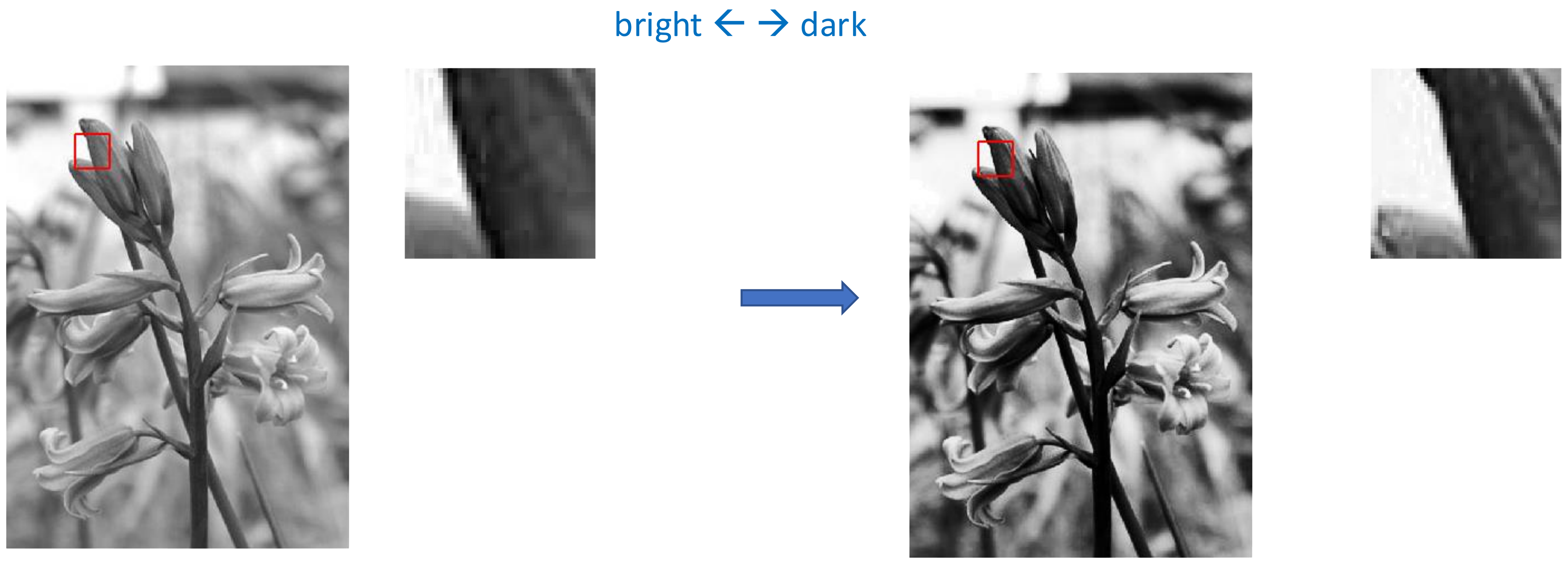


Image integrity

- Intensity reassignments must preserve image integrity
  - Not affect the intensity information structure (intensity ranking) with respect to the pixel geometry



”Pixel geometry”

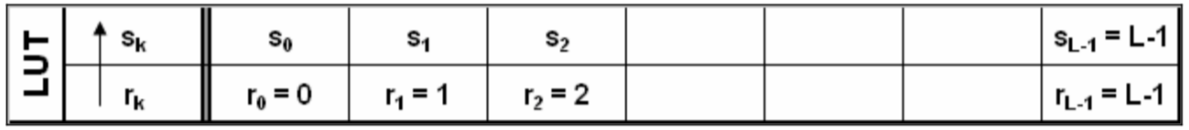
Map/
Transform

intensity r_k ((r_k=k; k=0,1,2, …, L-1) in the original image

to an intensity level s_k

$$s = T(r) \qquad 0 \le r \le L-1$$

Pixel lookup table



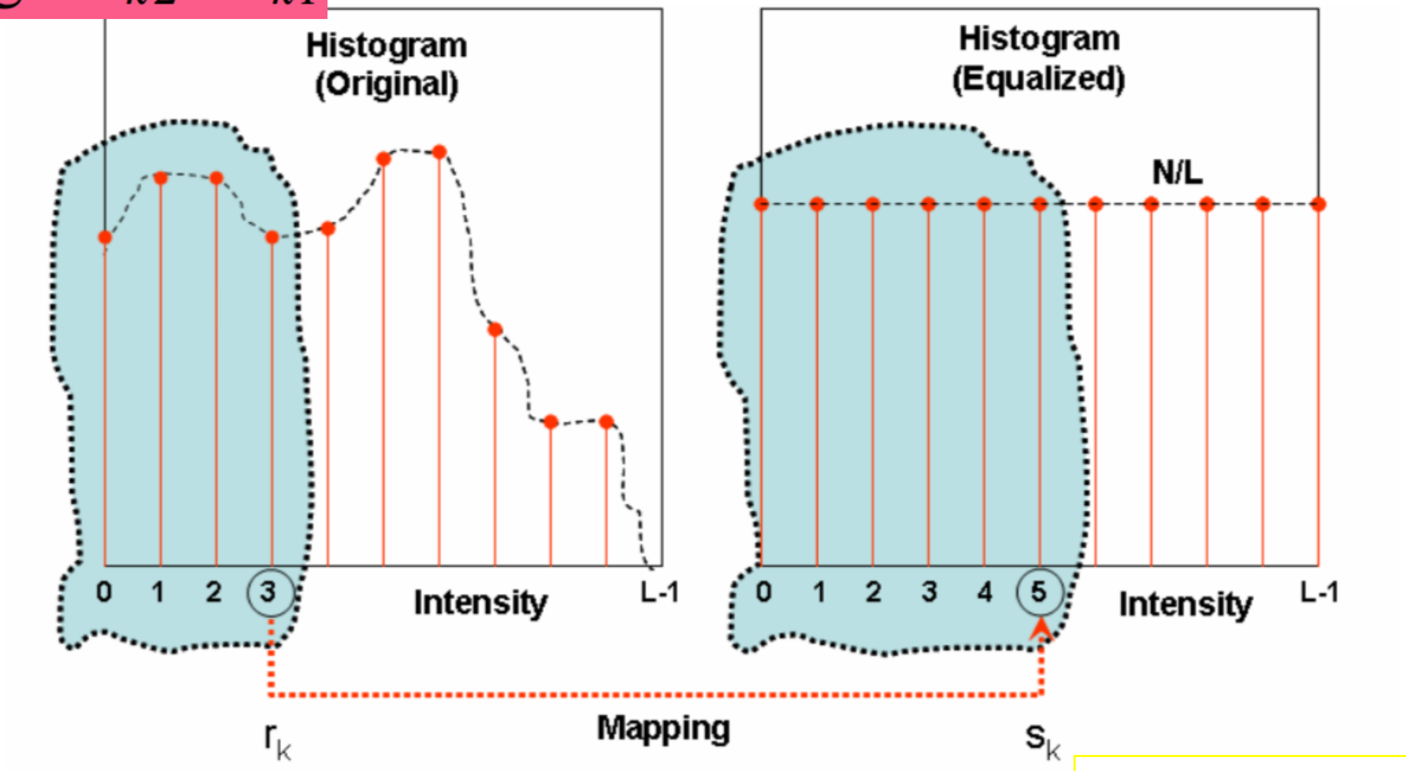
LUT	↑ s_k	s_0	s_1	s_2				s_{L-1} = L-1
	r_k	r_0 = 0	r_1 = 1	r_2 = 2				r_{L-1} = L-1

Preserving the image structure by:

if $r_{k2} > r_{k1}$, then we get $s_{k2} \ge s_{k1}$

Histogram equalization algorithm

if $r_{k2} > r_{k1}$, then we get $s_{k2} \geq s_{k1}$



N pixels
L muc xam
N/L

Total number of image pixels with intensities up to r_k

= total number of image pixels up to s_k in the mapped intensities

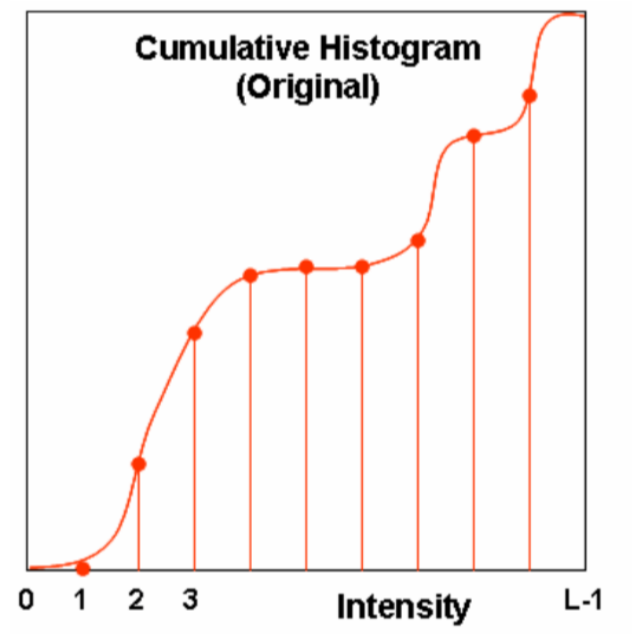
$$\sum_{i=0}^{r_k} n_R(i) = \left(\frac{N}{L}\right)(s_k+1) \quad : \quad k=0,1,2,3,\ldots,L\text{-}1;\ r_k = k$$

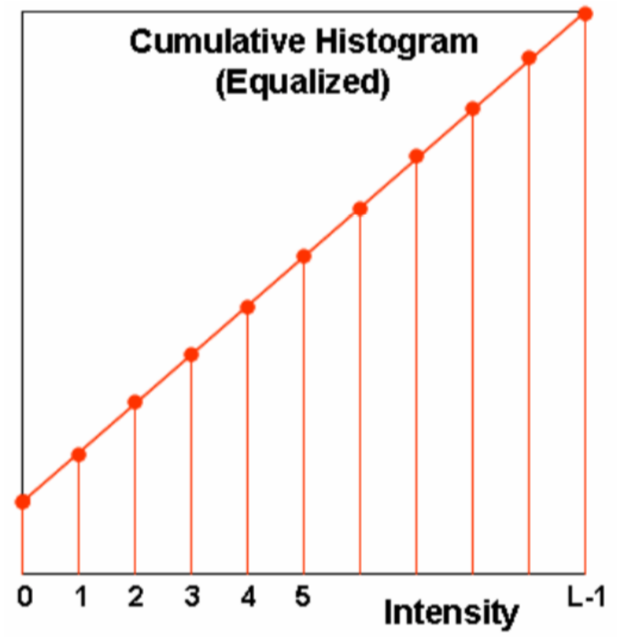
Or

$$\sum_{i=0}^{r_k} h_R(i) = \left(\frac{N}{L}\right)(s_k+1)$$

Cumulative Intensity Histogram

C_r(k) = C_s(k)





Total number of image pixels with intensities up to r_k

= total number of image pixels up to s_k in the mapped intensities

cumulative histogram at r_k = at s_k

$$\sum_{i=0}^{r_k} n_R(i) = \left(\frac{N}{L}\right)(s_k + 1) \quad : \quad k = 0,1,2,3,\ldots,L-1; \; r_k = k$$

Note

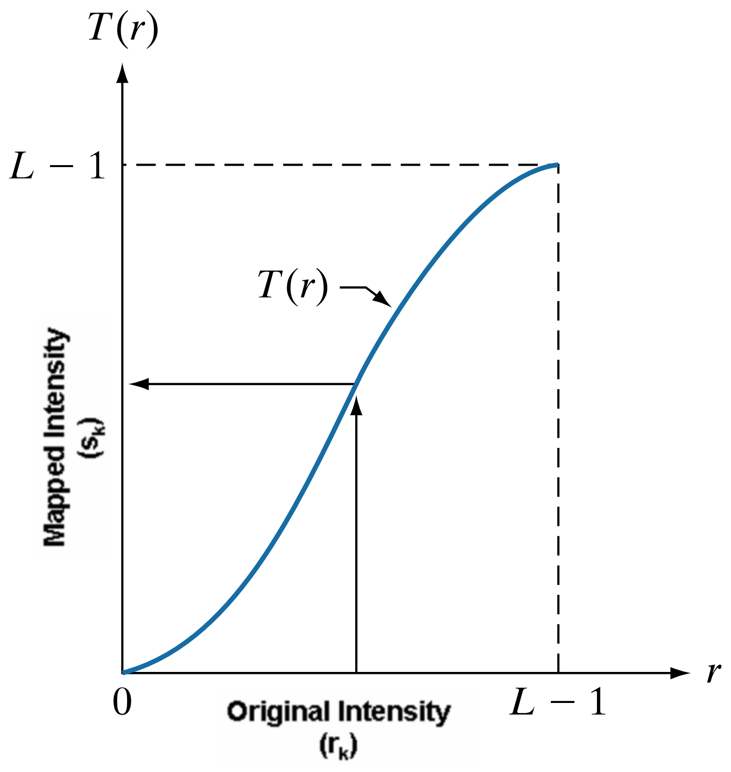
- All image pixels with the same intensity in the original image cannot be assigned to different intensities but be mapped to the same new value all together.
- → Perfect histogram equalization will not be possible for digital images due to discrete values

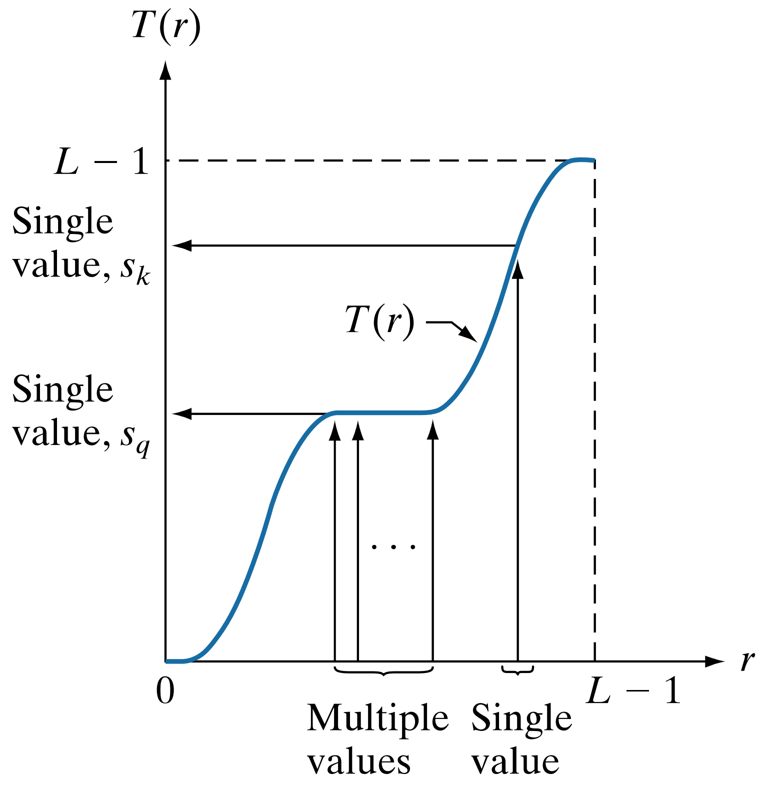
$$\sum_{i=0}^{r_k} h_R(i) = \left(\frac{N}{L}\right)(s_k+1) \Longrightarrow$$

$$s_k = \left(\frac{L}{N}\right)\sum_{i=0}^{r_k} h_R(i) - 1 : \; k=0,1,2,3,\ldots,L-1; \; r_k = k \quad \text{(w.r.t. Histogram)}$$

$$s_k = \left(\frac{L}{N}\right) C_R(k) - 1 \quad : \; k=0,1,2,3,\ldots,L-1; \; r_k = k \quad \text{(w.r.t. Cumulative Histogram)}$$

Characteristics of histogram equalization





Monotonically increasing function

Steps of Histogram equalization

1. Compute the intensity histogram and/or cumulative intensity histogram of the original image
2. Compute the $r_k \rightarrow s_k$ lookup table

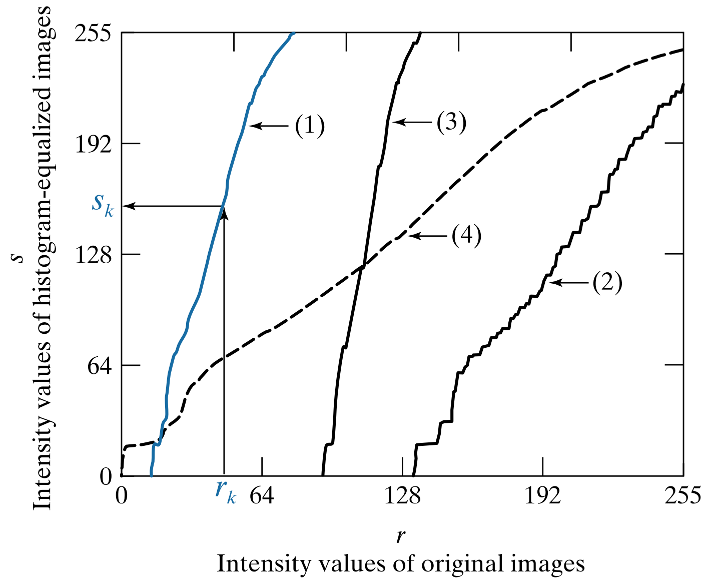
$$s_k = \left(\frac{L}{N}\right) \sum_{i=0}^{r_k} h_R(i) - 1 : \; k=0,1,2,3,\ldots,L-1; \; r_k = k \quad \text{(w.r.t. Histogram)}$$

$$s_k = \left(\frac{L}{N}\right) C_R(k) - 1 \quad : \; k=0,1,2,3,\ldots,L-1; \; r_k = k \quad \text{(w.r.t. Cumulative Histogram)}$$

3. Transform the original image using the lookup table

HE example

FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained using Eq. (3-15) and the histograms of the images on the left column of Fig. 3.20. Mapping of one intensity value r_k in image 1 to its corresponding value s_k is shown.



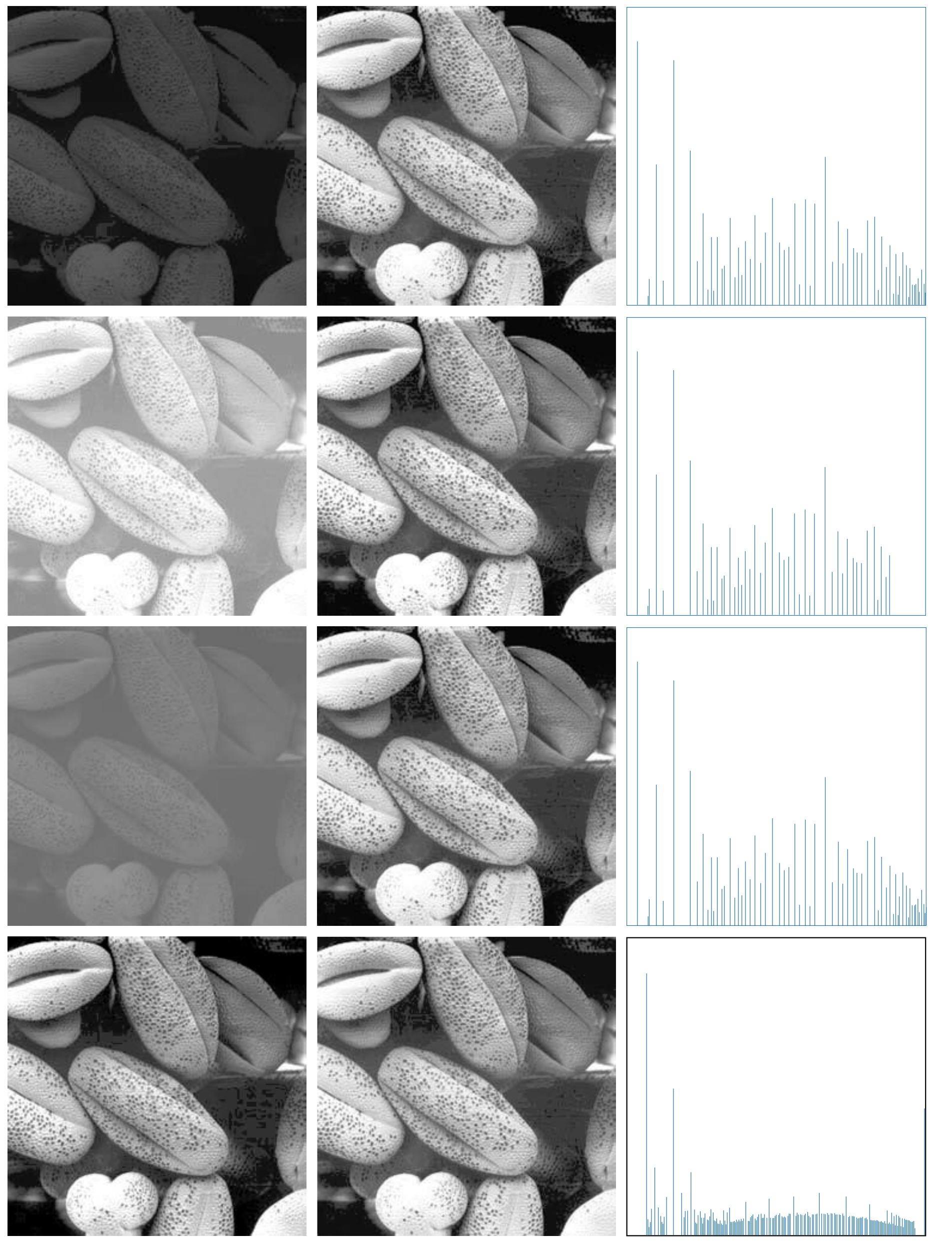
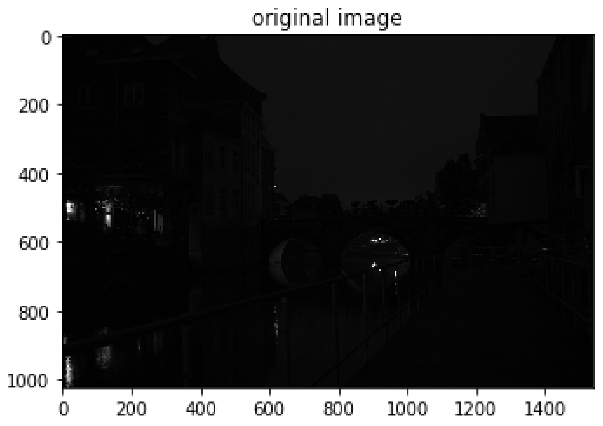
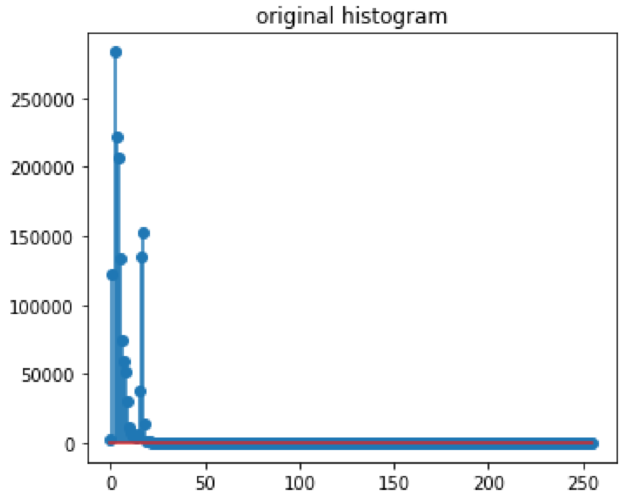


FIGURE 3.20 Left column: Images from Fig. 3.16. Center column: Corresponding histogram-equalized images. Right column: histograms of the images in the center column (compare with the histograms in Fig. 3.16).

Source: Gonzalez et al.


original image


original histogram

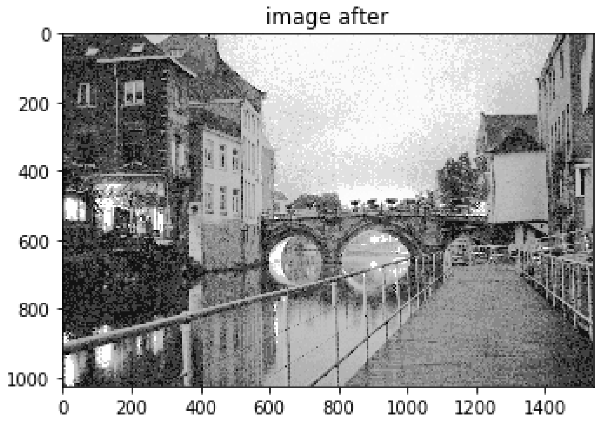
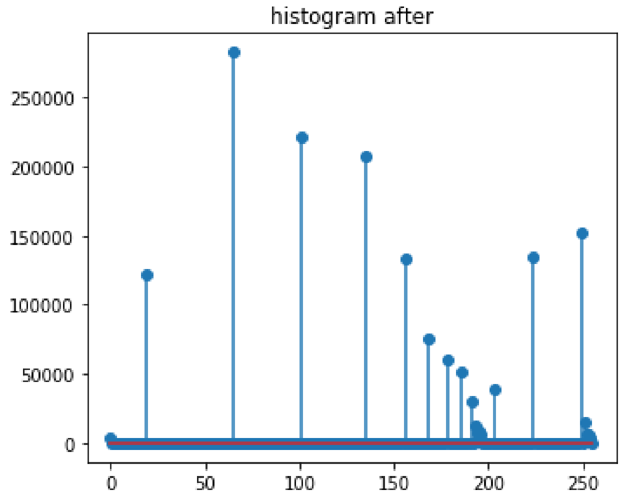


image after


histogram after

Implementation

- Refer to source code

Local histogram equalization

Local histogram processing

- Global histogram processing: a transformation function modifies the intensity distribution of an entire image
  - Suitable for overall enhancement
- Local histogram processing: modifies intensity distribution of pixel neighborhoods
  - To enhance details over small areas in an image

  - Two approaches:
    - Non-overlapping regions → can produce an undesirable “blocky” effect
    - Overlapping → more computation

Example of local histogram equalization

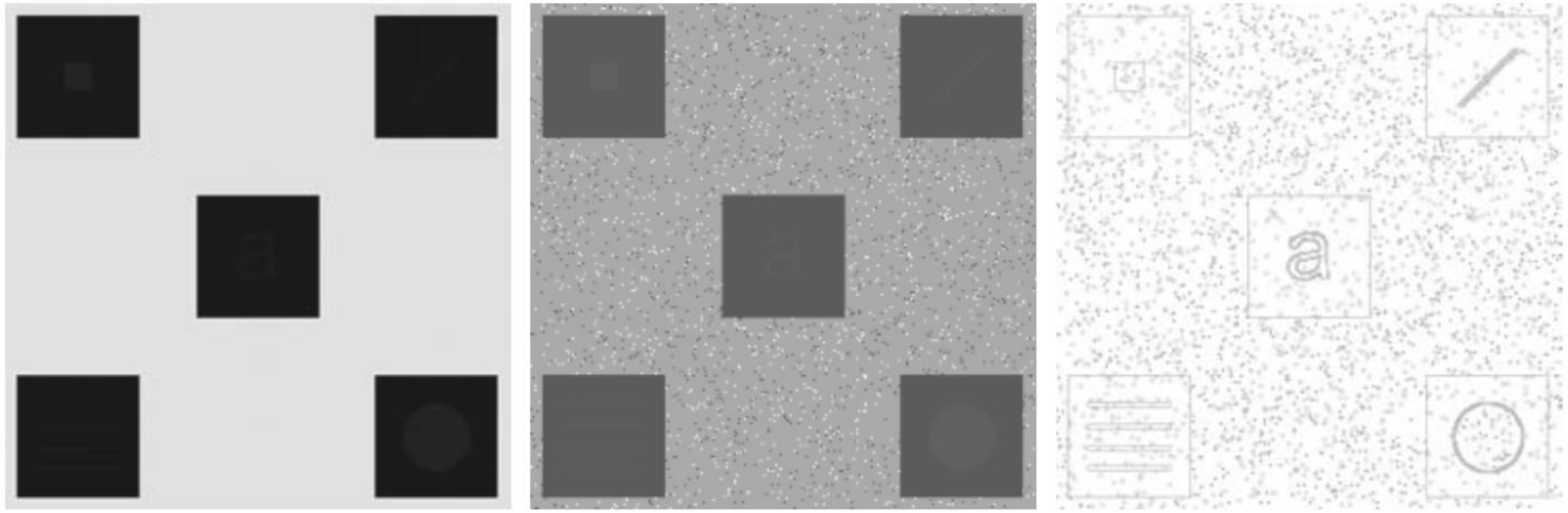
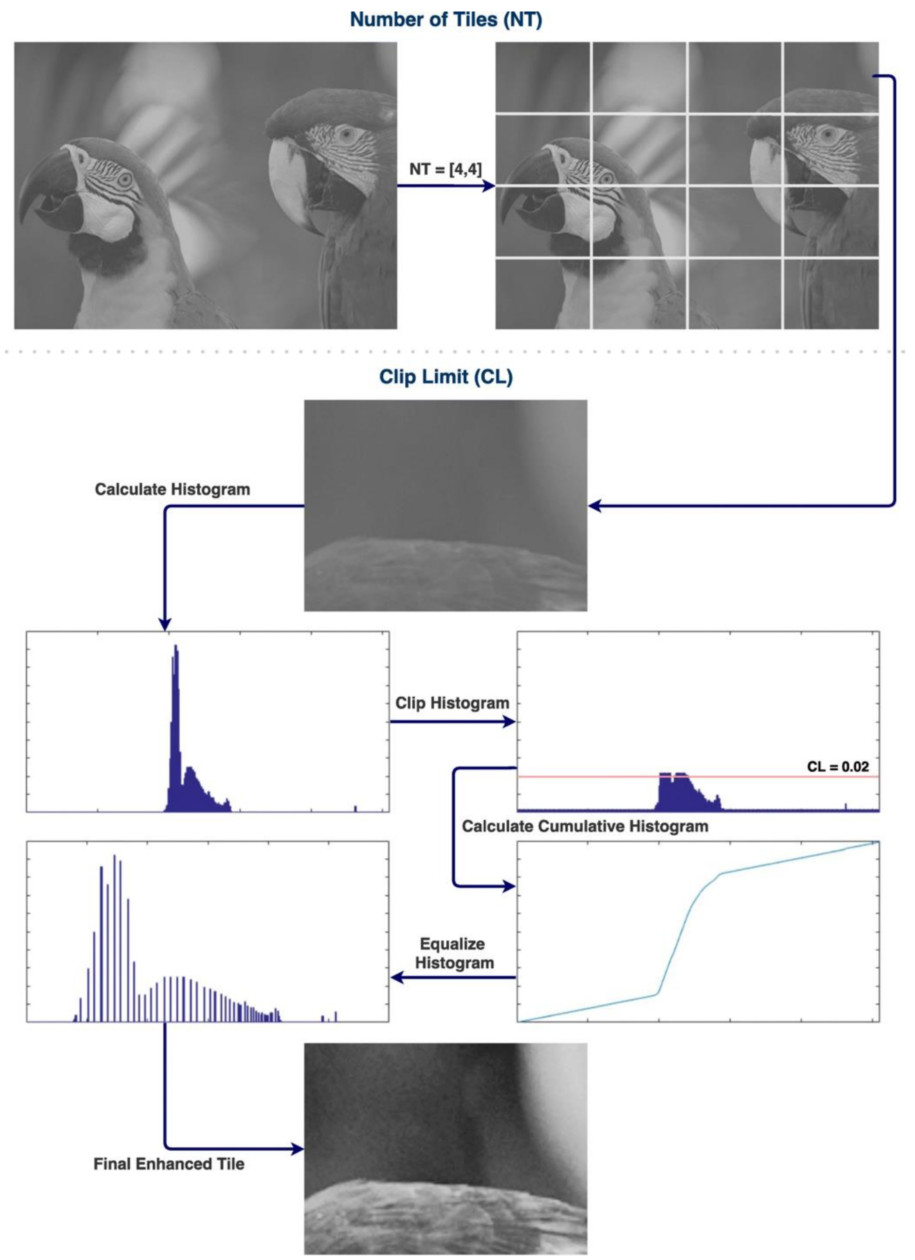


FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization.

Using a neighborhood of size 3x3

Contrast Limited Adaptive Histogram Equalization (CLAHE)

- Image is divided into small blocks called "tiles" (e.g., 8x8 pixels).
- Each block is histogram equalized as usual.
- In a small area, histogram would confine to a small region (unless there is noise).
- If noise is there, it will be amplified --> to avoid this, **contrast limiting* is applied.
- If any histogram bin is above the specified contrast limit (e.g., 40), those pixels are clipped and distributed uniformly to other bins before applying histogram equalization. After equalization, to remove artifacts in tile borders, bilinear interpolation is applied.



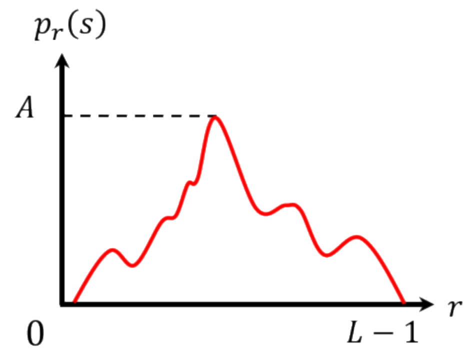
Ref: - Campos, Gabriel Fillipe Centini, et al. "Machine learning hyperparameter selection for Contrast Limited Adaptive Histogram Equalization." *EURASIP Journal on Image and Video Processing* 2019.1 (2019): 59.

- https://docs.opencv.org/master/d2/d74/tutorial_js_histogram_equalization.html

Histogram matching

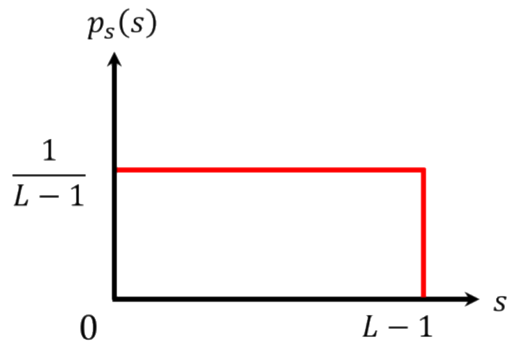
As known as “Histogram Specification”

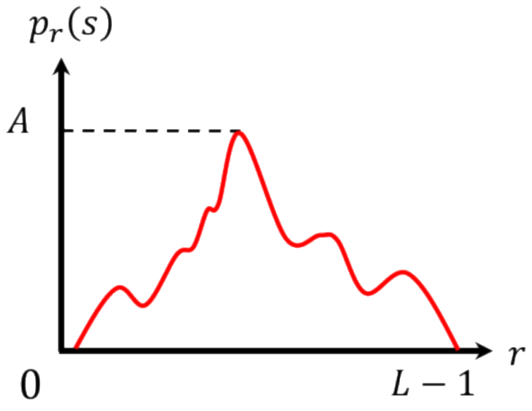
Histogram matching



Histogram Equalization $\rightarrow$

A transformation function that generates an output image with a uniform histogram

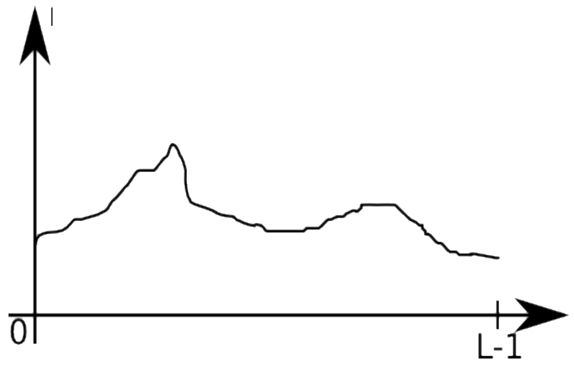




source

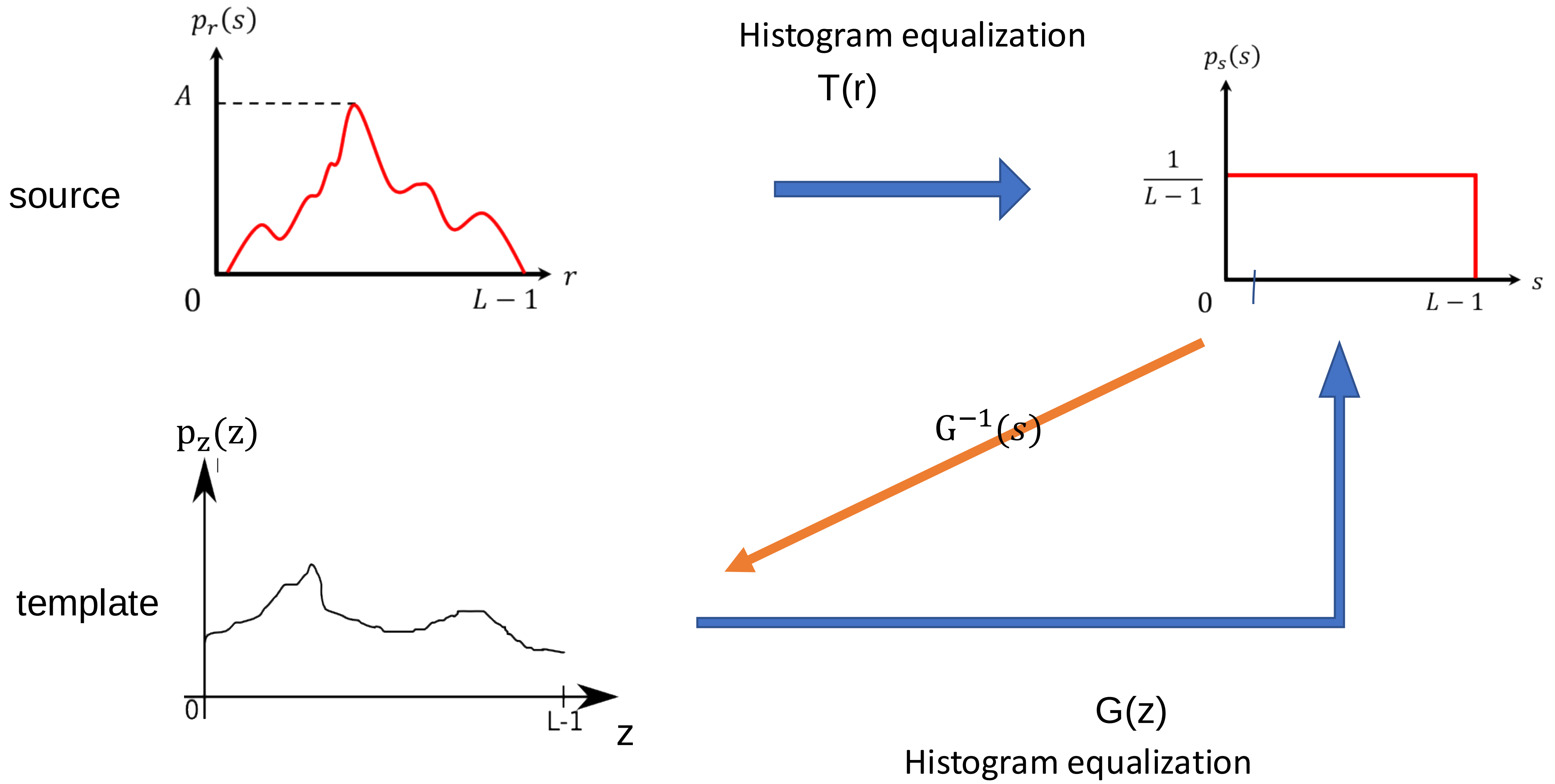
Histogram Matching $\rightarrow$

A transformation function that generates an output image with a specified histogram



template

Histogram matching



Steps of histogram matching

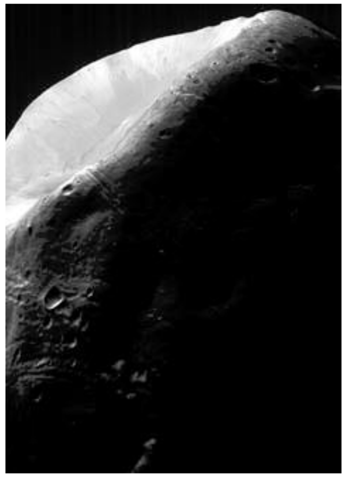
1. Compute the histogram of the input image and do the histogram equalization $T(r_k)$ from r_k to s_k

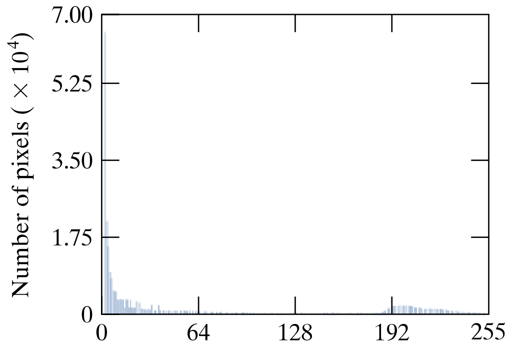
2. Compute histogram of the template image and do the histogram equalization $G(z_q)$ from z_q to s'_q

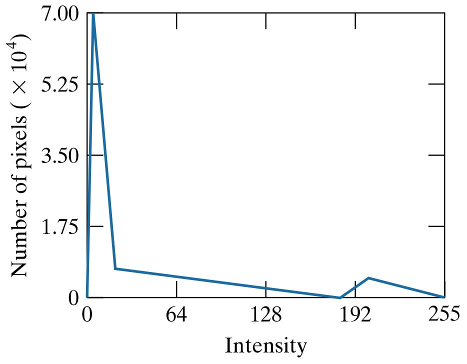
3. For each s_k, find z_q such that $G(z_q)$ is close to s_k
4. Do the transformation

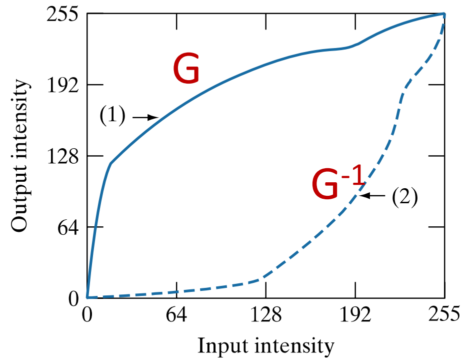
$$r \rightarrow s_k \rightarrow z_q$$

Histogram matching example

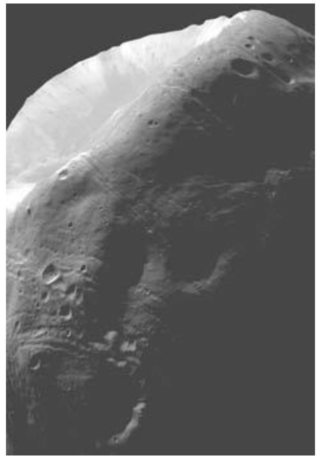








z



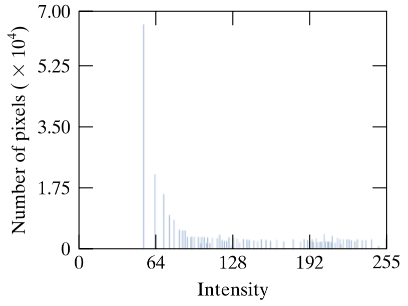
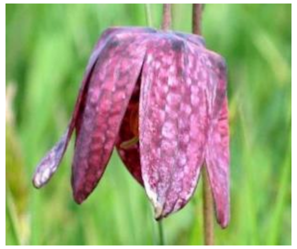


Image retrieval using histogram

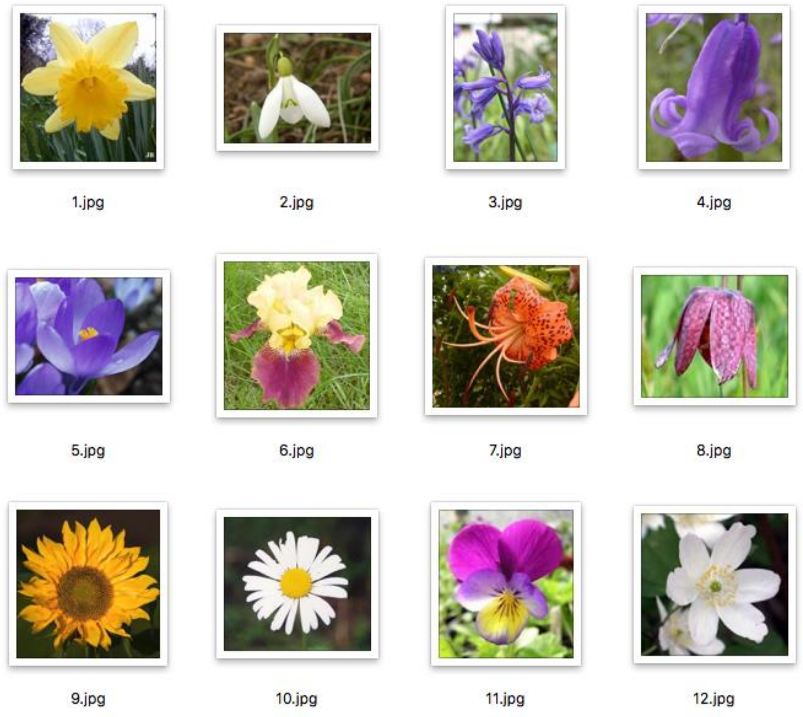
Giới thiệu bài toán

- Tìm ảnh trong tập dữ liệu ảnh gần giống với ảnh đầu vào nhất

Input



Dataset



→ Bài toán sắp xếp

Image retrieval algorithm

1. For each image in the dataset *D*
    1. Calculate the channel-wise histogram
    2. Stack histograms to make 1D vector (as feature vector)
2. Given an input image, do step 1 to obtain the feature vector (*q*)
3. Do vector similarity calculation between vector *q* and vectors in *D*
4. Return the corresponding images which have the most similar feature vectors to *q*

Implementation

- Refer to source code