

Image Processing

INT3404 20

Lecturer: Nguyen Thi Ngoc Diep, Ph.D.

Email: ngocdiep@vnu.edu.vn

1

Schedule

Week	Content	Homework
1	Introduction	Set up environments: Python 3, OpenCV 3, Numpy, Jupyter Notebook
2	Digital image – Point operations Contrast adjust – Combining images	HW1: adjust gamma to find the best contrast
3	Histogram - Histogram equalization – Histogram-based image classification	Self-study
4	Spatial filtering - Template matching	Self-study
5	Feature extraction Edge, Line, and Texture	Self-study
6	Morphological operations	HW2: Barcode detection → Require submission as mid-term test
7	Filtering in the Frequency domain Announcement of Final project topics	Final project registration
8	Color image processing	HW3: Conversion between color spaces, color image segmentation
9	Geometric transformations	Self-study
10	Noise and restoration	Self-study
11	Compression	Self-study
12	Final project presentation	Self-study
13	Final project presentation Class summarization	Self-study

2

2

1

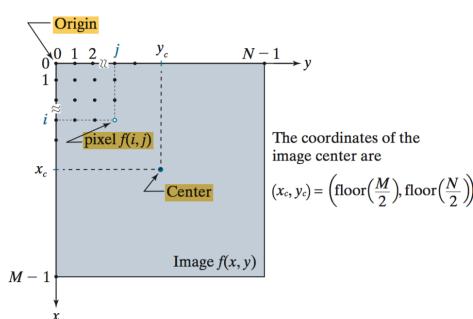
Schedule

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

3

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*.



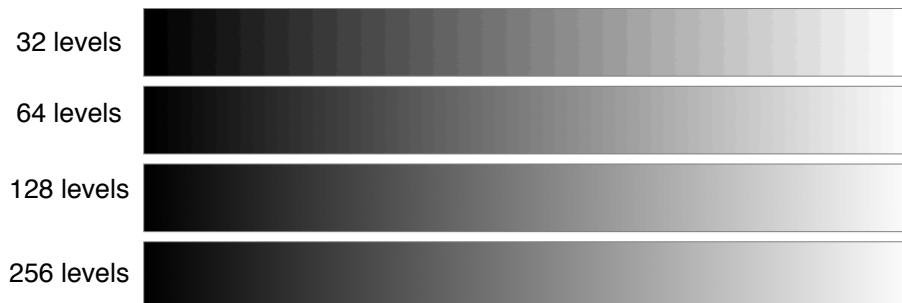
$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

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

4

Recall week 2: Quantization levels

Contouring is most visible for a ramp



Conventional grayscale image has 256 levels

Image credit: Bernd Girod

5

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

who did it? 😊

6

Week 3

Histograms

7

Why study histogram processing?

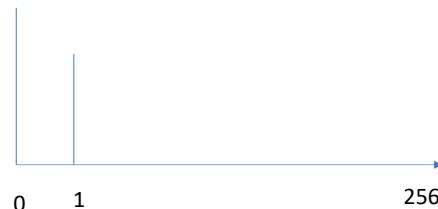
- Histogram manipulation is a fundamental tool in image processing
- Histograms:
 - Simple to compute
 - Suitable for fast hardware implementations
 - Popular tool for real-time image processing
- Histogram shape is related to image appearance

8

Histogram

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

Normalized histogram:



- Unnormalized histogram:

The subdivisions of the intensity scale are called **histogram bins**

$$h(r_k) = n_k$$

- 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

9

Histogram shape and image appearance

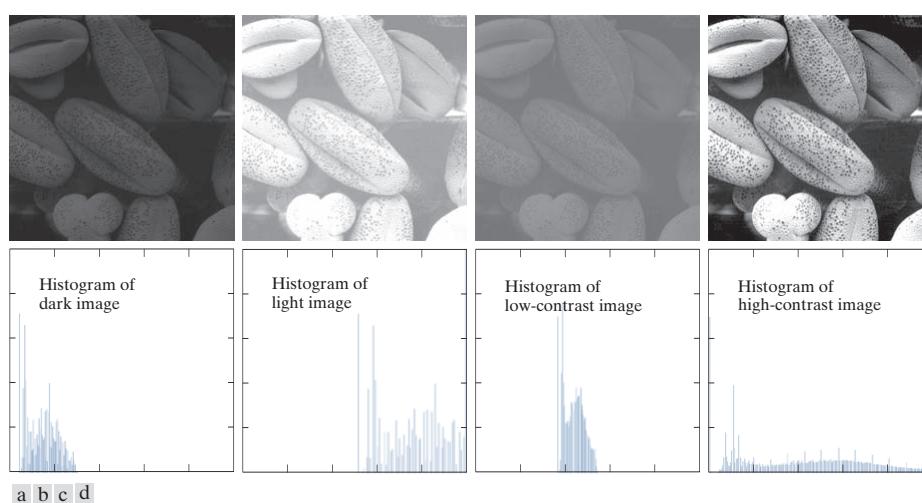
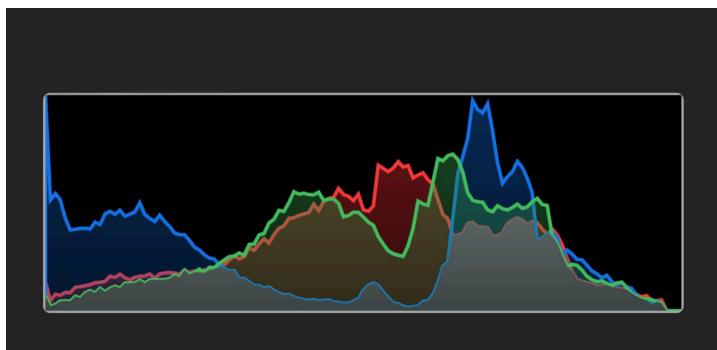


FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

10

Color image



11

Histogram equalization

(Cân bằng biểu đồ mức xám)

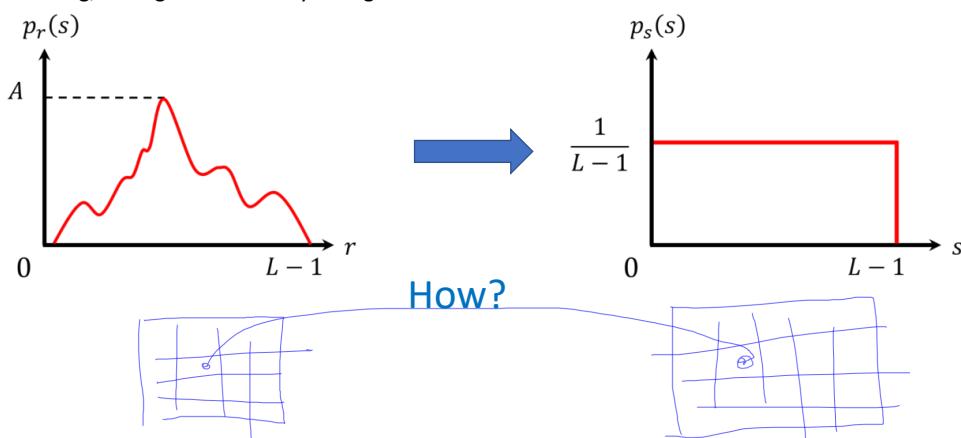
A special contrast enhancement technique

12

Histogram equalization

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

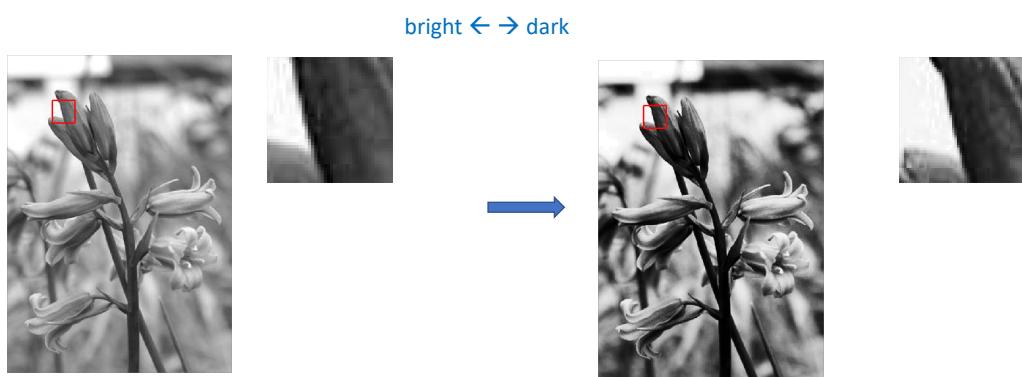
- Reassigning pixel intensity values such that the resulting image has a (close to) uniform distribution across its entire range of values
- Meaning, having a flat intensity histogram



13

Image integrity

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



14

"Pixel geometry"

Map/
Transform intensity r_k ($(r_k=k; k=0,1,2, \dots, L-1)$ in the original image
 to an intensity level s_k

$$s = T(r) \quad 0 \leq r \leq L-1$$

Pixel lookup table

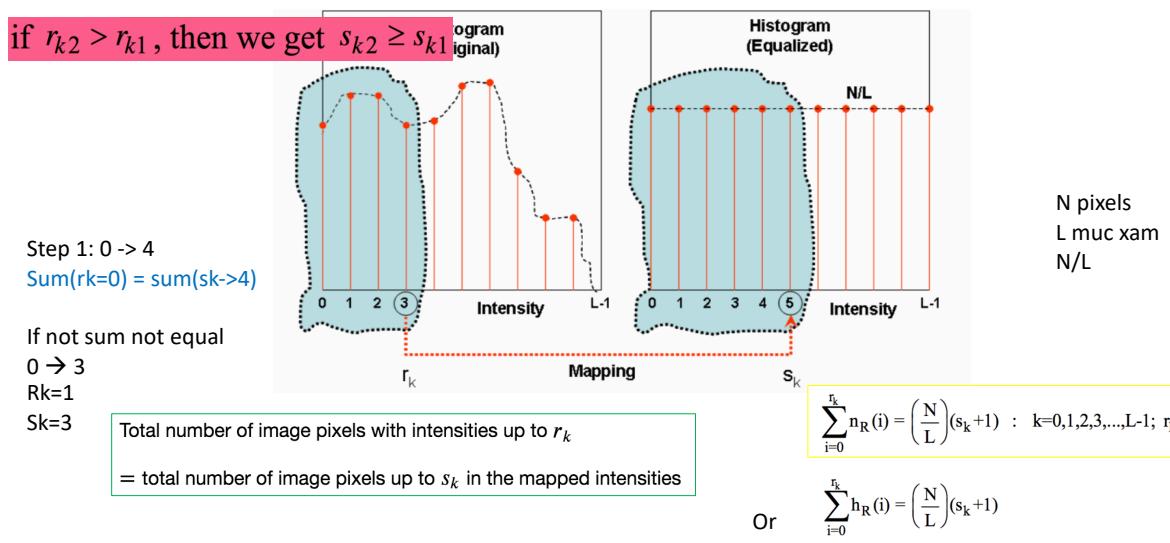
r_k	s_k	s_0	s_1	s_2			$s_{L-1} = L-1$
5	$\uparrow s_k$ r_k	$r_0 = 0$	$r_1 = 1$	$r_2 = 2$			$r_{L-1} = L-1$

Preserving the image structure by:

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

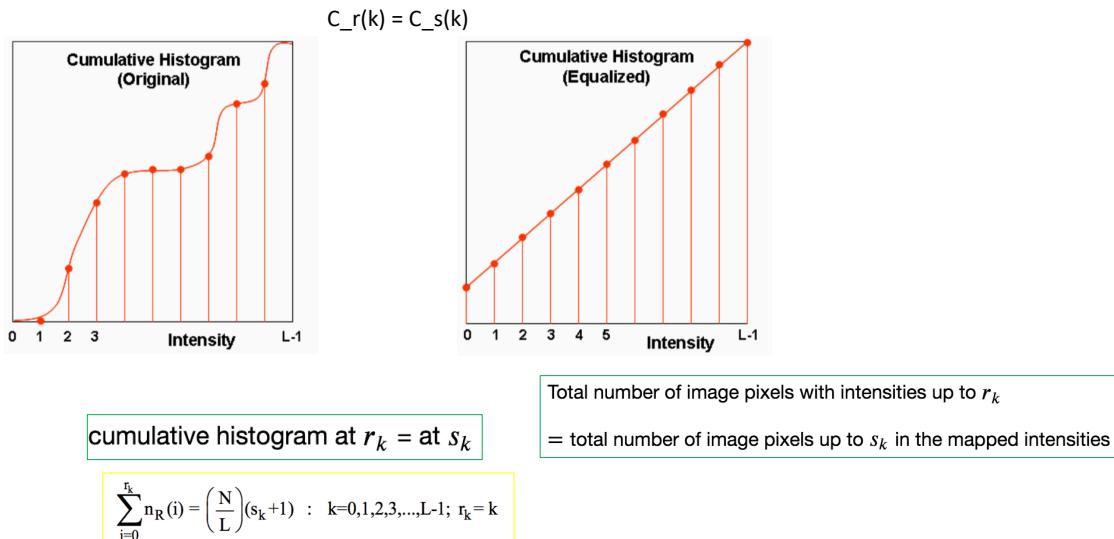
15

Histogram equalization algorithm



16

Cumulative Intensity Histogram



17

Note

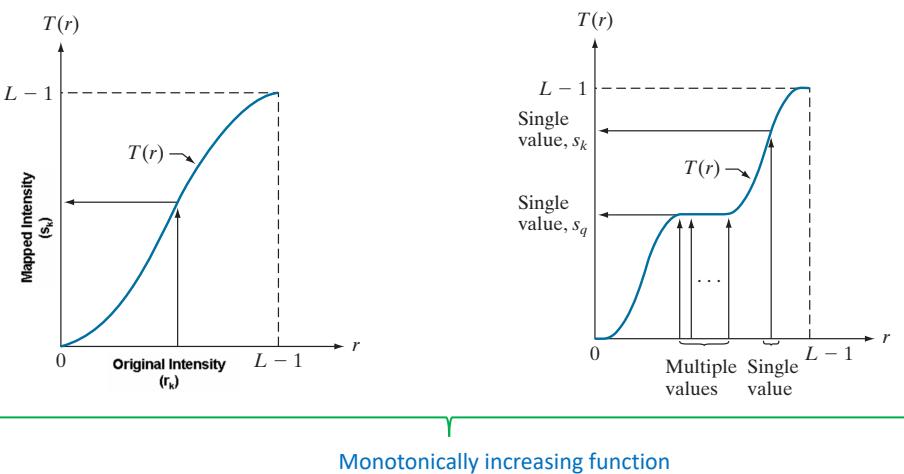
- All image pixels with the same intensity in the original image cannot be assigned to different intensities but be mapped to the same new value all together.
- → Perfect histogram equalization will not be possible for digital images due to discrete values

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

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

18

Characteristics of histogram equalization



19

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

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

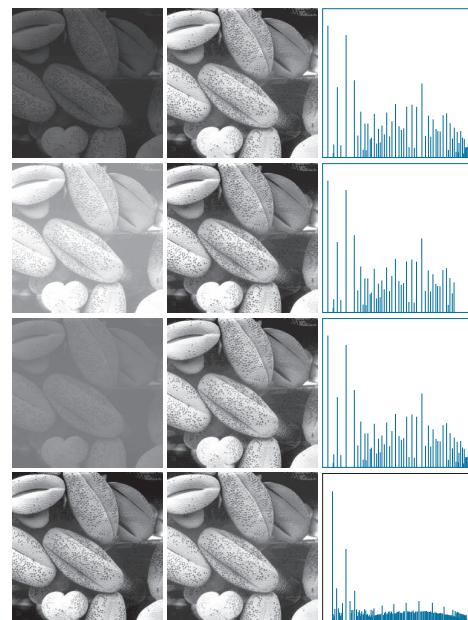
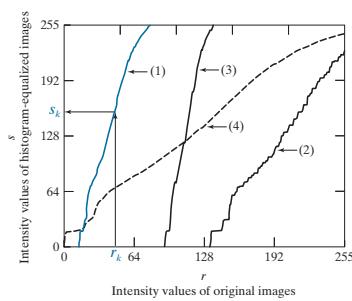
$$s_k = \left(\frac{L}{N} \right) C_R(k) - 1 : k=0,1,2,3,\dots,L-1; r_k = k \text{ (w.r.t. Cumulative Histogram)}$$

3. Transform the original image using the lookup table

20

HE example

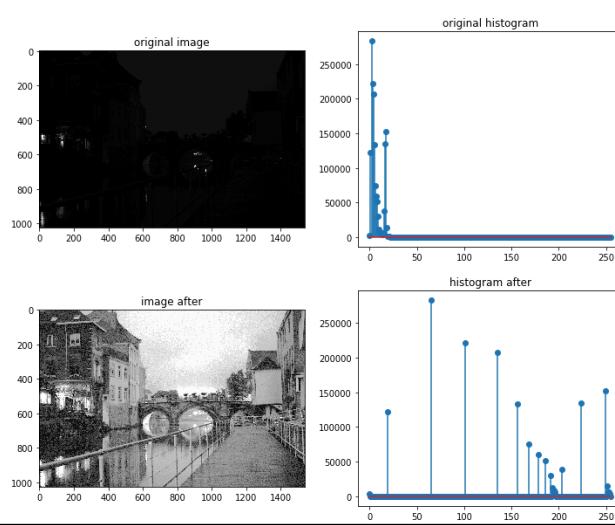
FIGURE 3.21
Transformation functions for histogram equalization. Transformations (1) through (4) were obtained using Eq. (3-15) and the histograms of the images on the left column of Fig. 3.20. Mapping of one intensity value r_k in image 1 to its corresponding value s_k is shown.



Source: Gonzalez et al.

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).

21



22

Implementation

- Refer to source code

23

Local histogram equalization

24

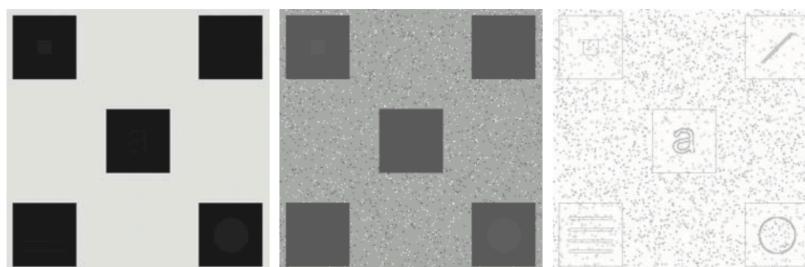
Local histogram processing

- 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

25

Example of local histogram equalization

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

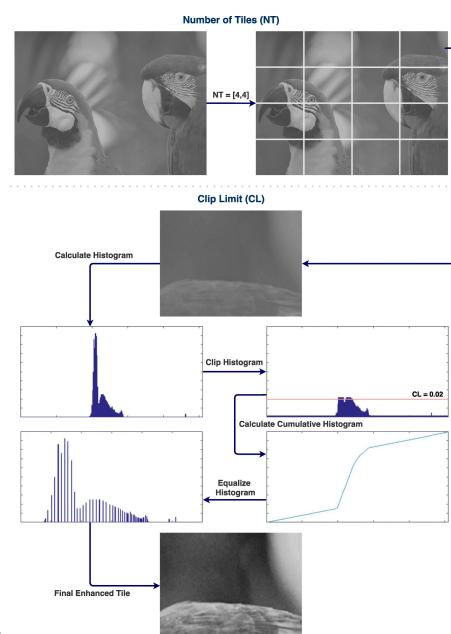


Using a neighborhood of size 3x3

26

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 Filipe 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/d7d/tutorial_js_histogram_equalization.html

27

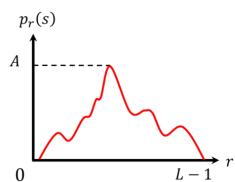
Histogram matching

As known as "Histogram Specification"

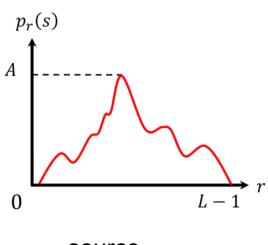
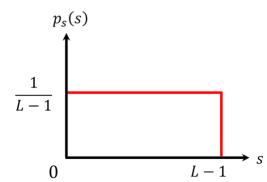
28

14

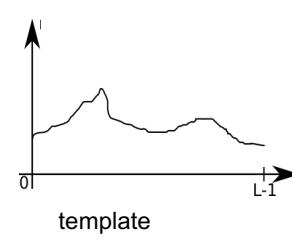
Histogram matching



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

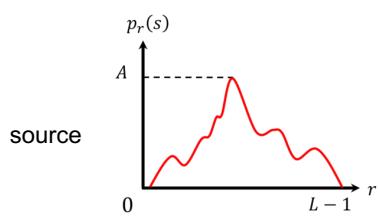


Histogram Matching
A transformation function that generates an output image with a specified histogram

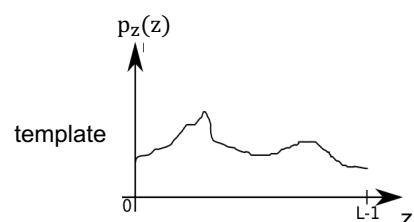
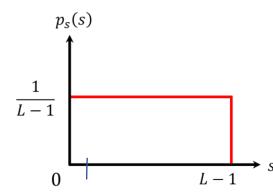


29

Histogram matching



Histogram equalization
 $T(r)$



$G^{-1}(s)$
Histogram equalization

30

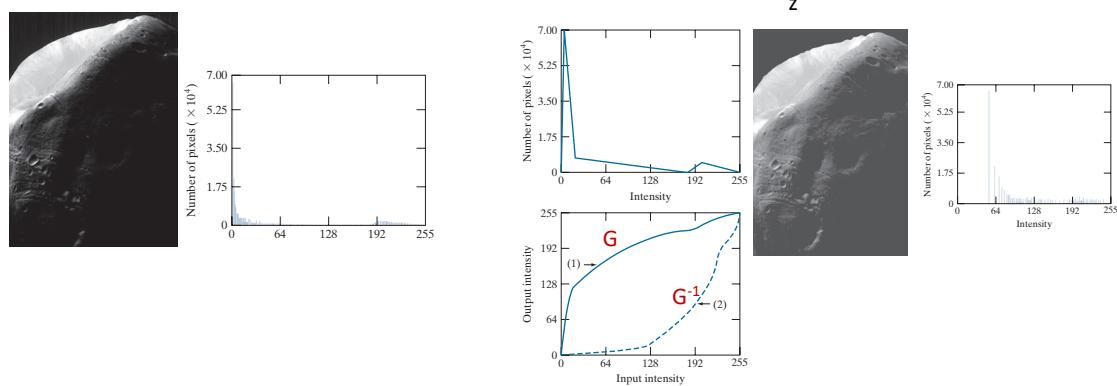
Steps of histogram matching

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

$$r \rightarrow s_k \rightarrow z_q$$

31

Histogram matching example



32

Image retrieval using histogram

33

Giới thiệu bài toán

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



34

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

36

Implementation

- Refer to source code

37