Thị giác máy tính

Ch3.1: Tăng cường chất lượng ảnh – lọc ảnh

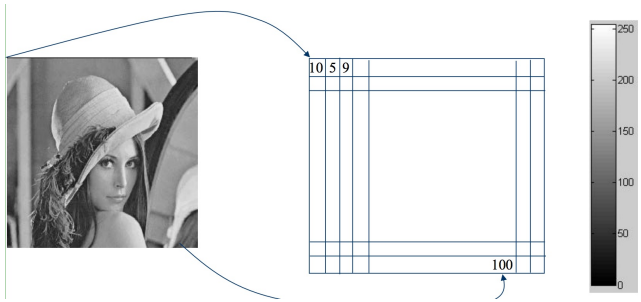
2

Nội dung

- Nhắc lại: biểu diễn ảnh số
- Tăng cường độ tương phản trên ảnh
  - Kéo giãn histogram
  - Cân bằng histogram
  - Biến đổi phi tuyến
- Nhân chập và lọc ảnh
  - Nhân chập
  - Một số bộ lọc tuyến tính
- Một số phép toán khác

3

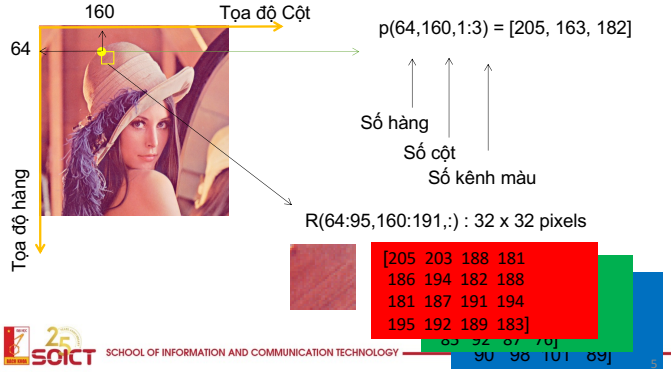
Biểu diễn ảnh số



4

Biểu diễn ảnh số

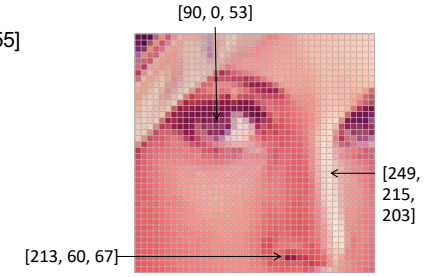
- Giá trị màu tại mỗi điểm ảnh, vùng ảnh



5

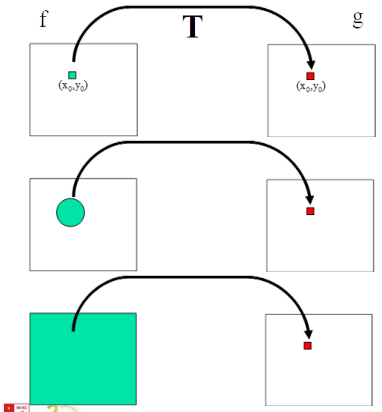
Biểu diễn ảnh số

- Ảnh số chứa các điểm ảnh rời rạc
- Giá trị điểm ảnh:
  - "grayscale" (or "intensity"): [0,255]
  - "color"
    - RGB: [R, G, B]
    - Lab: [L, a, b]
    - HSV: [H, S, V]



6

Biến đổi giá trị điểm ảnh



Isolated: $g(x_0,y_0)=T[f(x_0,y_0)]$

Local: $g(x_0,y_0)=T[f(V)]$
V:neighbors of (x_0,y_0)

Global: $g(x_0,y_0)=T[f(x,y)]$
example: FFT

Source : Caroline Rougier. Traitement d'images (IFT2730). Univ. de Montréal.

8

Nội dung

- Nhắc lại: biểu diễn ảnh số
- Tăng cường độ tương phản trên ảnh
  - Kéo giãn histogram
  - Cân bằng histogram
  - Biến đổi phi tuyến
- Nhân chập và lọc ảnh
  - Nhân chập
  - Một số bộ lọc tuyến tính
- Một số phép toán khác

9

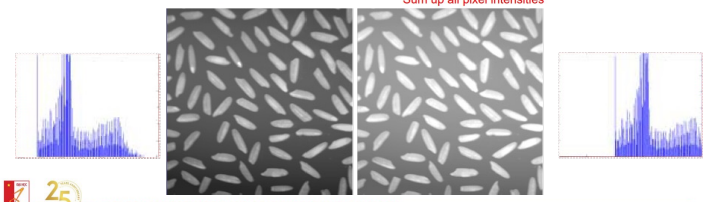
Độ sáng của ảnh (Brightness)

- Trung bình cường độ sáng của tất cả các điểm ảnh
  - Thể hiện mức sáng, tối trên ảnh

$$B(I) = \frac{1}{wh}\sum_{v=1}^{h}\sum_{u=1}^{w} I(u,v)$$

Divide by total number of pixels

Sum up all pixel intensities



10

Độ tương phản (contrast)

- Thể hiện sự phân biệt dễ dàng các đối tượng trong ảnh
- Một số cách tính
  - Standard deviation of intensity values of pixels in the image

$$C = \sqrt{\frac{1}{wh}\sum_{u=1}^{h}\sum_{v=1}^{w}(I(u,v) - mean)^2}$$

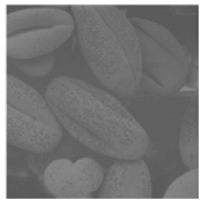
  - Difference between intensity value maximum et minimum

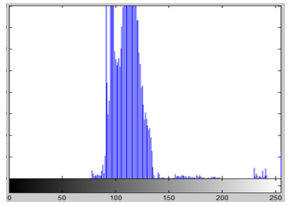
$$C = \frac{\max(I(u,v)) - \min(I(u,v))}{\max(I(u,v)) + \min(I(u,v))}$$

11

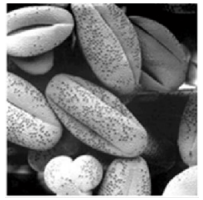
Histogram & độ tương phản

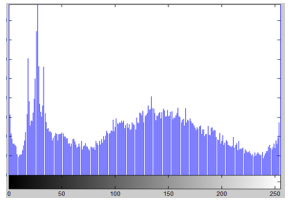
Low- contrast image





High- contrast image

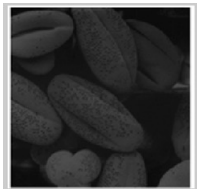


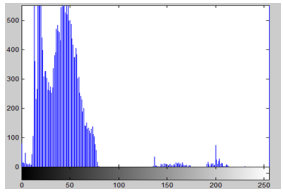


12

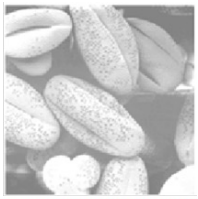
Histogram & độ sáng

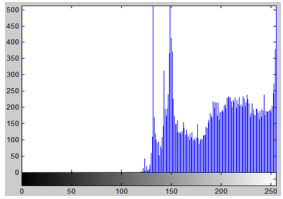
Dark image





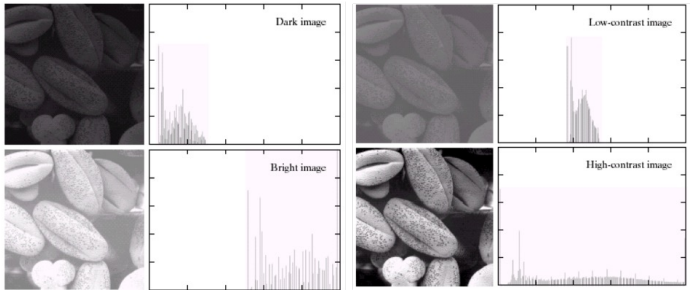
Light image





13

Ví dụ



14

Tăng cường độ tương phản

- Thay đổi giá trị điểm ảnh để có độ tương phản cao hơn
- Một số phương pháp:
  - Kéo giãn dải động ảnh (Linear stretching of intensity range):
    - Linear stretching
    - Linear stretching with saturation
    - Piecewise linear transformation
  - Biến đổi phi tuyến. VD: log, gama correction
  - Cân bằng histogram

15

Phép toán ở mức điểm ảnh

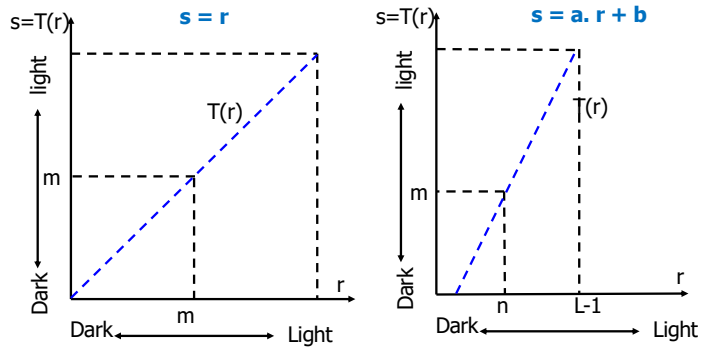
- Giá trị mới của điểm ảnh chỉ phụ thuộc vào giá trị hiện tại của điểm ảnh đó, không phụ thuộc vào các điểm lân cận

s = T(r)

  - r = f(x, y)
  - s = g(x, y)
  - T: transformation function

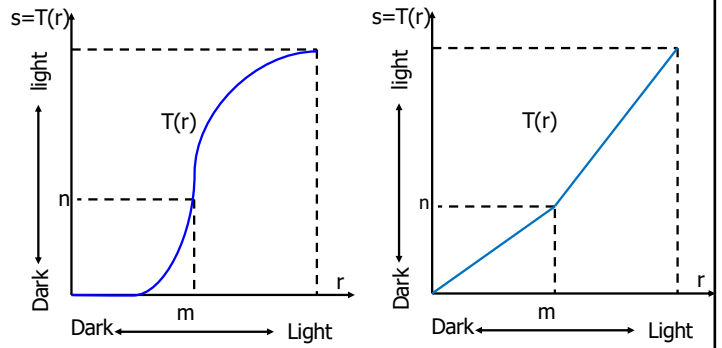
16

Biến đổi đồng nhất / tuyến tính



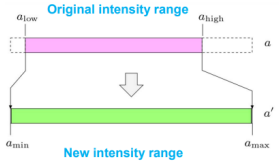
17

Biến đổi tuyến tính / phi tuyến theo đoạn



18

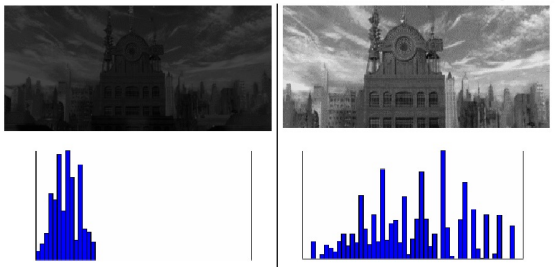
Linear stretching



$$f_{ac}(a) = a_{min} + (a - a_{low}) \cdot \frac{a_{max} - a_{min}}{a_{high} - a_{low}}$$

If $a_{min} = 0$ and $a_{max} = 255$

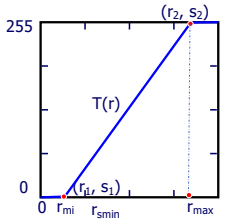
$$f_{ac}(a) = (a - a_{low}) \cdot \frac{255}{a_{high} - a_{low}}$$

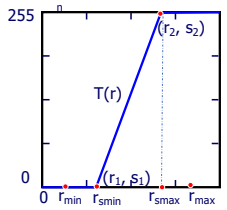


19

Linear stretching

- If $(r_1, s_1) = (r_{min}, 0)$;
  $(r_2, s_2) = (r_{max}, 255)$
  - Linear stretching
- If $(r_1, s_1) = (r_{smin}, 0)$, $r_{smin} > r_{min}$
  $(r_2, s_2) = (r_{smax}, 255)$ $r_{smax} < r_{max}$
  - Linear stretching with saturation



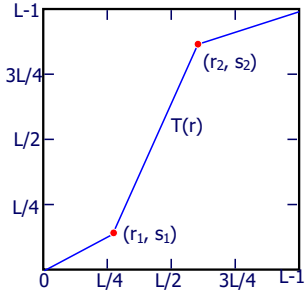


SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

20

Piecewise-Linear transformation

- Biến đổi tuyến tính theo đoạn:
  - Tăng hoặc giảm độ tương phản ở các khoảng mức xám khác nhau
  - Mức độ tăng, giảm khác nhau ở mỗi khoảng giá trị
- VD hàm bên:
  - Độ tương phản giảm trong khoảng
    - từ 0 đến r1 và
    - từ r_2 đến L-1
  - Độ tương phản tăng trong khoảng từ r_1 đến r_2

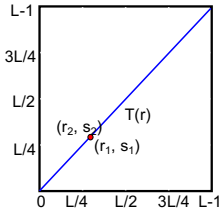


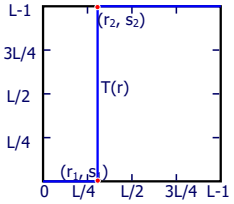
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

21

Piecewise-Linear transformation

- Trường hợp đặc biệt
  - Nếu r1 = s1, r2 = s2,
    - Biến đổi đồng nhất: I' = I
  - Nếu s1 = 0, s2 = L-1,
    - Trở thành hàm lấy ngưỡng
    - Đầu ra: ảnh nhị phân

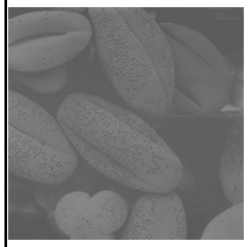




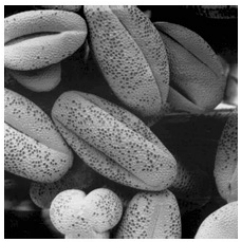
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

22

Piecewise-Linear transformation



Original image, low contrast

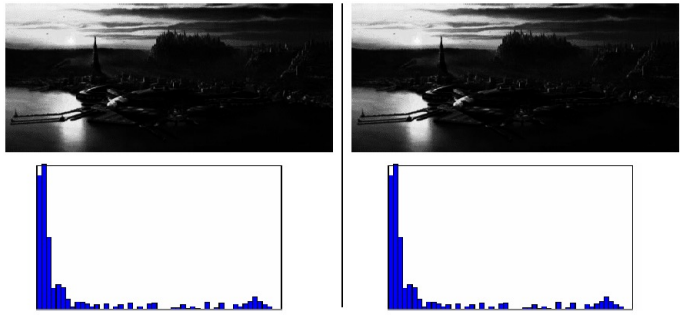


Contrast sketching
$(r_1, s_1) = (r_{min}, 0)$
$(r_2, s_2) = (r_{max}, L-1)$

Thresholding
$(r_1, s_1) = (m, 0)$
$(r_2, s_2) = (m, L-1)$
m: mean of intensities

23

Linear stretching

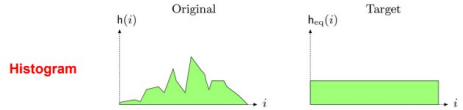


No efficace?

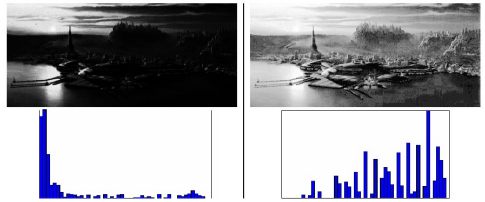
24

Cân bằng histogram

- Histogram của ảnh sau thay đổi hướng tới phân phối đều



- Không tham số. OpenCV:cv2.equalizeHist(img)



25

Cân bằng histogram (histogram equalization)

- s1. đếm số điểm ảnh có mức xám k trong ảnh: n_k *k = 0, 1, .., L-1.*
- s2. tính histogram chuẩn hóa p(k)

$$p(k) = \frac{n_k}{n}$$

- s3. Tính histogram tích lũy T(k):

$$T(k) = \sum_{j=0}^{k} p(j) = \frac{1}{n}\sum_{j=0}^{k} n_j$$

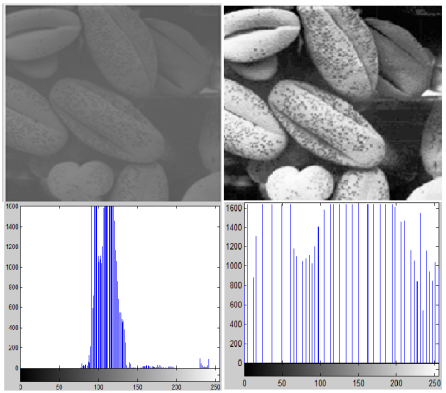
- s4. Mức xám s_k của ảnh đầu ra J, tương ứng mức xám k của ảnh đầu vào:

$$s_k = round[(L-1)T(k)]$$

- s5. J được xác định bằng cách thay các giá trị mức xám k trong ảnh đầu vào bằng mức xám s_k

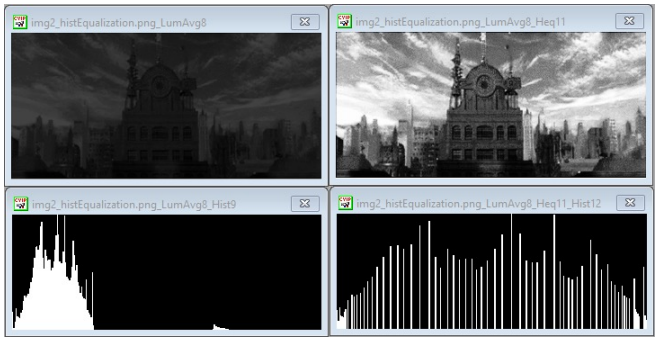
26

Histogram equalization



28

Cân bằng histogram



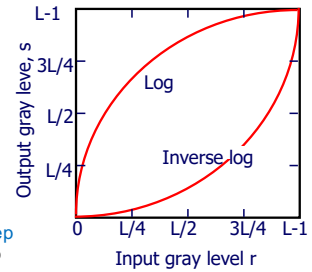
29

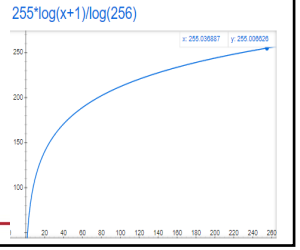
Log transformations

- Biến đổi log:

$$s = c \times \log(1 + r)$$

  - c: constant
  - $r \geq 0$
- Log
  - Các điểm ảnh có khoảng giá trị hẹp ở mức xám thấp trên ảnh đầu vào thành khoảng rộng hơn ở ảnh đầu ra
  - Ngược lại với các điểm ảnh có mức xám cao
- Inverse log: ngược lại

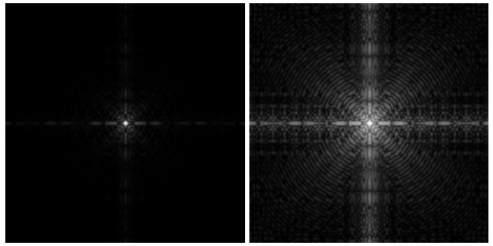




31

Log transformations

- Làm sáng hơn các điểm có mức xám thấp trong khi nén các giá trị có mức xám cao



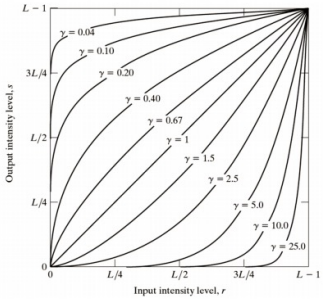
32

Power-Law (Gamma) transformations

- The general form of power-law transformation is:

$$s = c.r^{\gamma}$$

  - $\gamma > 1$: compress values in dark area, while expanding values in light area
  - $\gamma < 1$: expand values in dark area, while compressing values in light area
  - r : normalized values to [0, 1]
  - c : scaling constant c corresponding to the bit size used

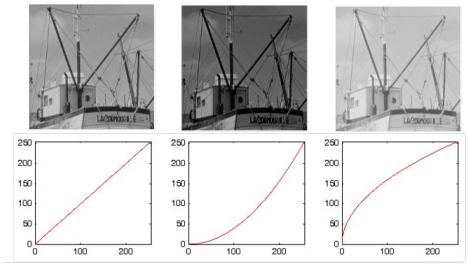


Power-Law (Gamma) transformations

- $\gamma 1 = 1;\ \gamma 2 = 3; \gamma 3 = 4; \gamma 4 = 5;$

Increasing the dynamic ranges of high intensities

Gama correction



Ex: Gamma correction (g)

$$s = c.r^{\gamma}$$

$$\gamma_{screen} \in [1.3,\ 3.0]$$

$$\gamma_{eye} \approx \frac{1}{2} \text{ to } \frac{1}{3}$$

In Practice with grey level image :
c = 255
r = gray_level/255

Gama correction

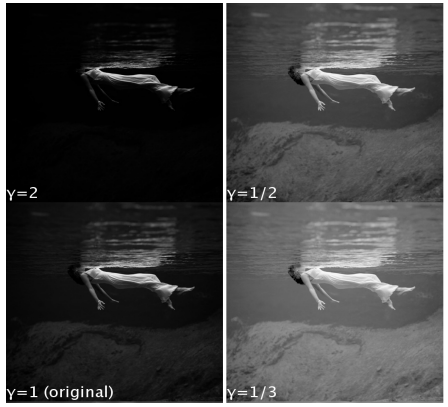
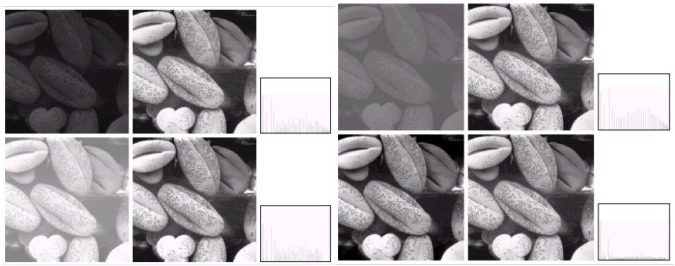


Image: Wikipedia

Cân bằng histogram

*If we take the **same image** with **different contrasts**, histogram equalization will give the **same results** for all images*



a b c
FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

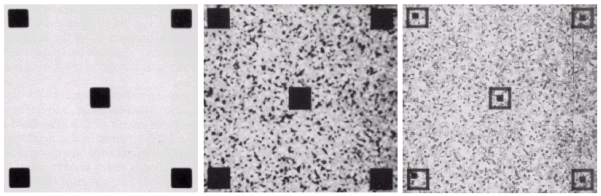
37

Cân bằng histogram vs. pp khác

- Discussion
  - Cân bằng histogram có luôn tốt?
  - Cân bằng histo vs Piecewise-Linear transformation ?
  - Cân bằng histo vs hiệu chỉnh gamma / log ?

38

Cân bằng histogram cục bộ

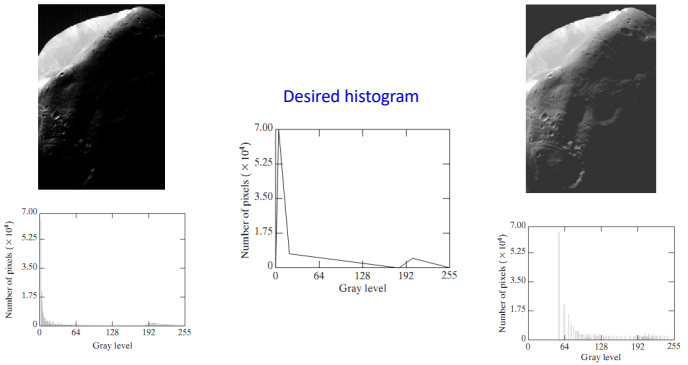


a b c
FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7 × 7 neighborhood about each pixel.

39

Một số phép biến đổi ảnh khác

Histogram matching



➔ Additional lecture

40

Một số phép biến đổi ảnh khác

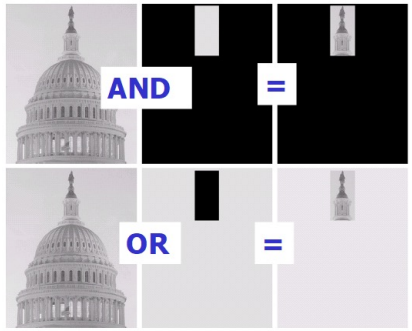
G(x,y) = F(x,y) + c

Ảnh gốc → Ảnh sau biến đổi

Lấy âm bản: s = (L – 1) – r

41

Phép toán logic trên ảnh



a b c
d e f
FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

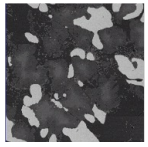
42

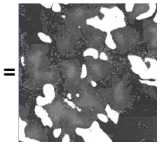
Phép toán +, -, x trên ảnh

f(x,y)

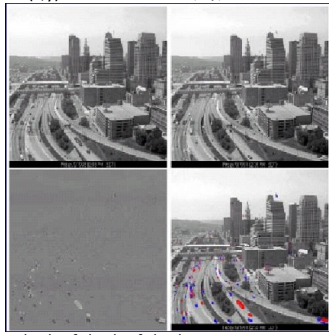
g(x,y) = f(x,y) + f(x,y)

x1,5 =





f1(x,y) f2(x,y)



g(x,y) = f2(x,y) – f1(x,y)

43

Tăng cường độ tương phản cho ảnh màu?

- Cân bằng histogram trên ảnh màu ?
  - Thực hiện trên mỗi kênh màu
    - Có thể tạo ra những thay đổi màu bất bình thường
    - KHÔNG KHUYẾN NGHỊ trừ khi có lý do cụ thể
  - Phương pháp gợi ý
    - Chuyển ảnh sang không gian màu khác như Lab, HSL/HSV
    - Thực hiện cân bằng histogram trên kênh thể hiện độ sáng tối trên ảnh như L hay V
    - Chuyển ngược lại sang RGB

44

Nội dung

- Nhắc lại: biểu diễn ảnh số
- Tăng cường độ tương phản trên ảnh
  - Kéo giãn histogram
  - Cân bằng histogram
  - Biến đổi phi tuyến
- Nhân chập và lọc ảnh
  - Nhân chập
  - Một số bộ lọc tuyến tính
- Một số phép toán khác khác

Lọc ảnh (image filtering)

- Lọc ảnh: Với mỗi điểm ảnh, tính giá trị mới của điểm ảnh dựa trên 1 hàm theo các điểm trong lân cận của nó
  - Lọc trong **miền không gian**:
    - Với mỗi điểm ảnh, tính giá trị mới của điểm ảnh dựa trên **1 hàm theo các điểm trong lân cận** của nó
  - Lọc ảnh trong **miền tần số** (bài sau)
  - Ảnh đầu vào và ra thường có **cùng kích thước**
- Nhân chập: phép lọc tuyến tính, hàm số là tổng có trọng số của các điểm ảnh trong lân cận của điểm ảnh xét.

$$I' = I * K$$

- Có vai trò quan trọng!
  - Tăng cường ảnh: giảm nhiễu, làm trơn, tăng độ tương phản, …
  - Trích chọn thông tin từ ảnh: Texture, edges, distinctive points, etc.
  - Phát hiện mẫu (template matching)

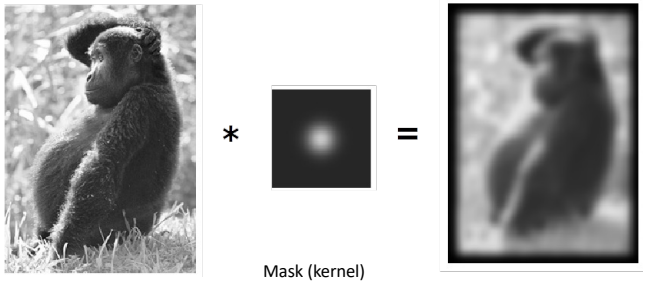
Lọc ảnh (image filtering)

- Lọc trong miền không gian
  - Biến đổi cục bộ được thực hiện trên ma trận điểm ảnh

    $g(x,y) = T(f(x,y))$
  - T: hàm tính toán dựa trên giá trị điểm (x,y) và các điểm hàng xóm (K)

    (K: Filter/Mask/Kernel/Window/Template Processing)
  - **T** giống nhau tại mọi vị trí trên ảnh

Lọc ảnh



Original image

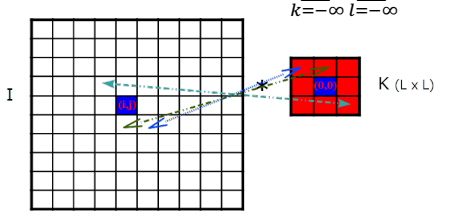
Mask (kernel)

Filtered image

Nhân chập (Convolution)

- Convolution: tổng có trọng của các điểm ảnh trong lân cận của điểm ảnh xét

$$f(n,m) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \times h[n-k, m-l]$$

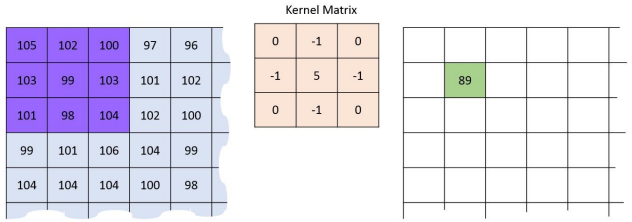


$$I'(i,j) = \sum_{u=-(L-1)/2}^{(L-1)/2} \sum_{v=-(L-1)/2}^{(L-1)/2} I(i-u, j-v)\, K(u,v)$$

49

Nhân chập

- New value of a pixel(i,j) is a weighted sum of its neigbors

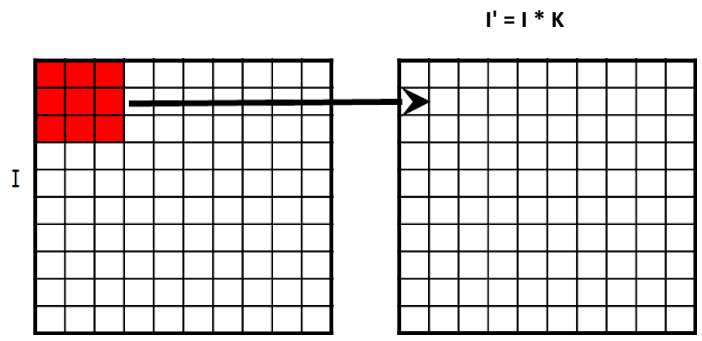


$$105*0 + 102*-1 + 100*0 + 103*-1 + 99*5 + 103*-1 + 101*0 + 98*-1 + 104*0 = 89$$

Source: http://machinelearninguru.com

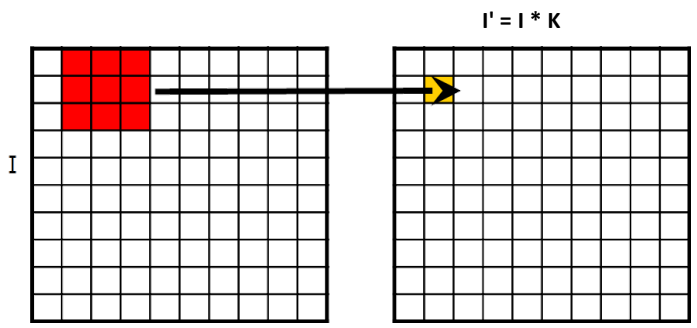
50

Nhân chập



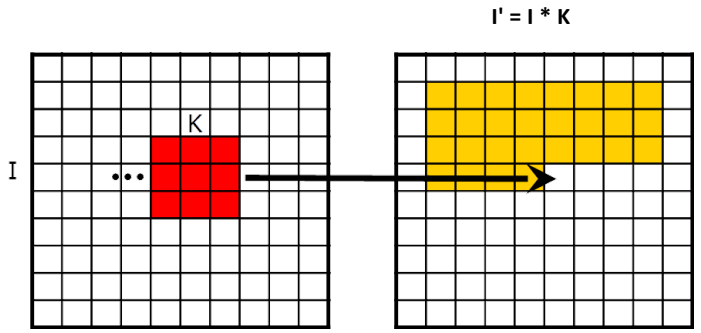
51

Nhân chập



52

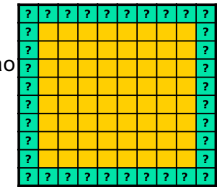
Nhân chập



53

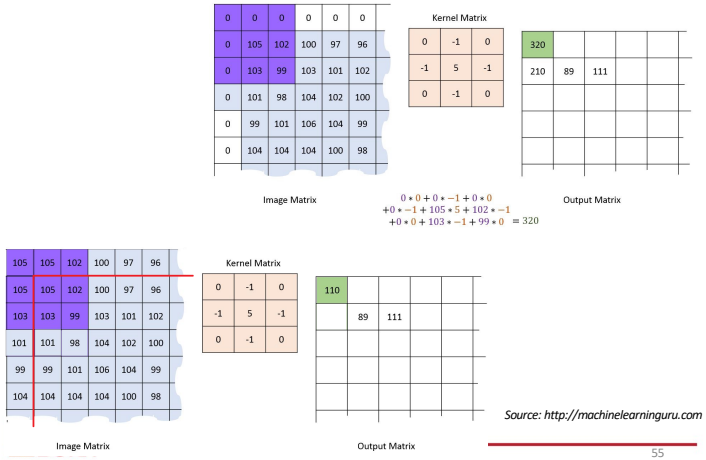
Nhân chập

- Vấn đề ở cạnh ảnh?
  - Thêm dòng/cột 0 vào ma trận đầu vào
  - Đối xứng gương:
    - f(-x,y) = f(x,y)
    - f(-x,-y) = f(x,y)



54

Nhân chập



55

Spatial Correlation vs Convolution

Spatial Correlation (☆) and Convolution (★)

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

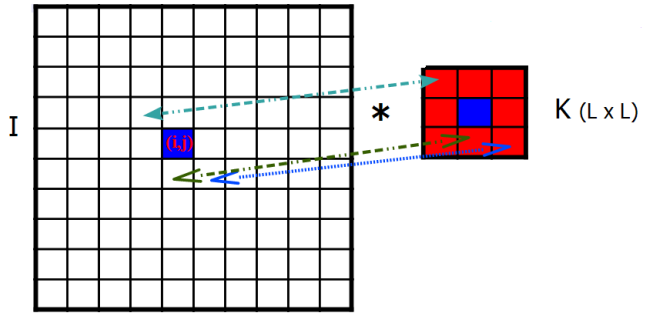
$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

56

Spatial Correlation vs Convolution

- Correlation



57

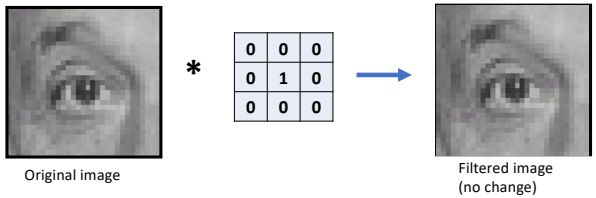
Spatial Correlation vs Convolution

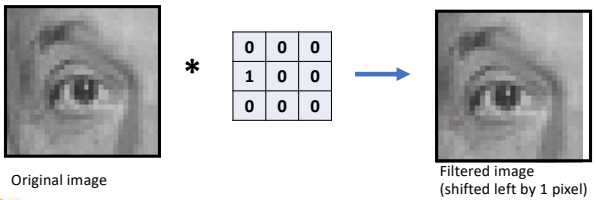
- Nếu mặt nạ đối xứng thì 2 phép toán này là một
- Correlation:
  - Sử dụng để tìm một mẫu ("template") nào đó có xuất hiện trong ảnh không
  - Không có tính kết hợp ➔ nếu thực hiện tìm mẫu (template matching), thì correlation là đủ
- Convolution:
  - Thường sử dụng lọc ảnh (noise removing, enhancement, ..)
  - Có tính chất kết hợp ➔ hữu ích khi cần thực hiện nhiều bộ lọc liên tiếp:

    I*h*g = I * (h*g)
- In Matlab: correlation: filter2, convolution: conv2

58

Một số bộ lọc (Some kernels)





Source: David Lowe

59

Một số bộ lọc (Some kernels)

- Nhân chập 2D
  - Chủ yếu được sử dụng để trích chọn đặc trưng trên ảnh
  - Được sử dụng như phép toán trong khối cơ sở của mạng Neuron tích chập: Convolutional Neural Networks (CNNs)
- Mỗi bộ lọc có hiệu ứng riêng và hữu ích cho các nhiệm vụ cụ thể như:
  - Làm mờ (lọc nhiễu),
  - Làm nét biên,
  - Phát hiện cạnh,
  - …..

60

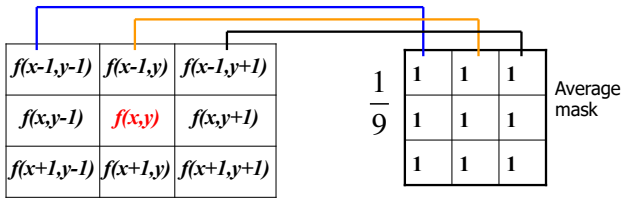
Bộ lọc làm trơn ảnh

- Mục đích
  - Lọc nhiễu
  - Làm trơn ảnh
  - Còn gọi là bộ lọc thông thấp
- Một số bộ lọc thông thấp
  - Bộ lọc trung bình
  - Bộ lọc Gauss
- Để tránh thay đổi độ sáng của ảnh, tổng các hệ số trong mặt nạ phải = 1

61

Bộ lọc làm trơn ảnh

- Lọc trung bình (**mean filter**):



$$g(x,y) = \frac{1}{9}[f(x-1,y-1) + f(x-1,y) + f(x-1,y+1) + f(x,y-1) + f(x,y) + f(x,y+1) + f(x+1,y-1) + f(x+1,y) + f(x+1,y+1)]$$

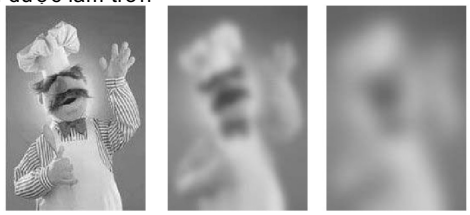
62

Bộ lọc làm trơn ảnh

- Lọc trung bình (**mean filter**):
  - Thay giá trị bởi giá trị trung bình của các hàng xóm
  - Ảnh được làm trơn

$1/9 \times$

1	1	1
1	1	1
1	1	1



Original image | Filtered image with box size 5x5 | Filtered image with box size 11x11

63

Bộ lọc làm trơn ảnh

- Lọc trung bình có trọng số
  - The pixel corresponding to the center of the mask is more important than the other ones.

$\frac{1}{16} \times$

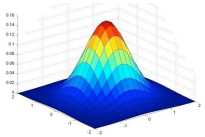
1	2	1
2	4	2
1	2	1

$$g(x,y) = \frac{1}{16}[f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1) + 2f(x,y-1) + 4f(x,y) + 2f(x,y+1) + f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)]$$

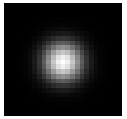
64

Bộ lọc làm trơn ảnh

- Gaussian filter



Gaussian function in 3D



Gaussian image

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

Gaussian filter with size 5 x5 , sigma =1

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Rule for Gaussian filter:
set **filter half-width to about 3σ**

65

Bộ lọc làm trơn ảnh

- Bộ lọc Gauss:
  - **Bộ lọc thông thấp**: loại bỏ các thành phần tương ứng tần số cao trên ảnh
    - Ảnh trơn hơn
    - Tốt hơn bộ lọc trung bình
  - Nhập chập Gauss với chính nó ta được một hàm Gauss
    - Lặp nhận chập với bộ lọc có kích thước nhỏ => thu được kết quả như nhận chập với bộ lọc có kích thước lớn hơn.
    - Nhân chập **2 lần** với bộ lọc Gauss có độ rộng **σ** giống như nhận chập **1 lần với bộ lọc có độ rộng σ√2** : $I*G_\sigma*G_\sigma = I * G_{\sigma\sqrt{2}}$
  - **Bộ lọc có thể phân tách được**: Hàm Gauss 2D có thể được biểu diễn như tích của 2 hàm 1D: 1 hàm theo x và 1 hàm theo y:
    - $G_\sigma(x,y) = G_\sigma(x).G_\sigma(y)$

66

Bộ lọc làm trơn ảnh

- Bộ lọc Gauss

Original image

Filtered image
with box size 5x5

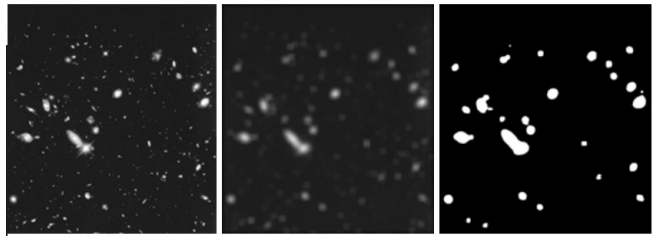
Filtered image
with box size 11x11

67

Bộ lọc làm trơn ảnh

- VD: sử dụng bộ lọc thông thấp để loại bỏ các vùng nhỏ

Original image | Average filtering: 15x15 | Thresholding of blurring image



68

Bộ lọc làm trơn ảnh

- Tổng quát:

$$g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$$

- Để tránh ảnh hưởng đến độ sáng của ảnh, tổng các hệ số của mặt nạ =1

69

Bộ lọc làm sắc nét cạnh

- Dựa trên ý tưởng của đạo hàm bậc 1 và đạo hàm bậc 2 trên ảnh

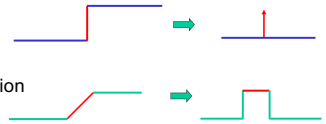
$$\frac{\partial f}{\partial x} \approx \begin{cases} f(x+1,y) - f(x,y) \\ f(x,y) - f(x-1,y) \\ 0.5\left(f(x+1,y) - f(x-1,y)\right) \end{cases}$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x+1,y) - 2f(x,y) + f(x-1,y)$$

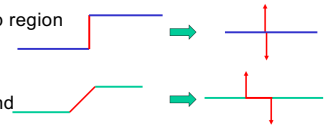
70

Đạo hàm bậc 1 và bậc 2

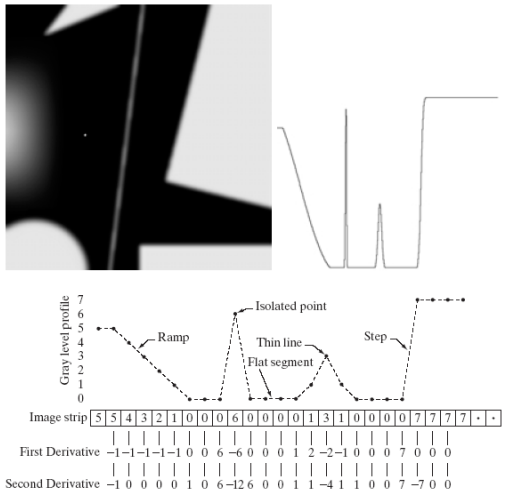
- Đạo hàm bậc 1:
  - Zero in flat region
  - Non-zero at start of step/ramp region
  - Non-zero along ramp
  - Strong response for step changes



- Đạo hàm bậc 2:
  - Zero in flat region
  - Non-zero at start/end of step/ramp region
  - Zero along ramp
  - Double response at step changes
  - Strong response for fine details and isolated points;



71



72

Đạo hàm bậc 1

- Một số bộ lọc để tính đạo hàm bậc 1 trên ảnh
  - Robert

1	0
0	-1

0	1
-1	0

  - Prewitt
    - Ít nhạy cảm với nhiễu
    - Mean + đạo hàm bậc 1

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

  - Sobel:
    - Ít nhạy cảm với nhiễu
    - Gauss + đạo hàm bậc 1

-1	-2	-1
0	0	0
1	2	1

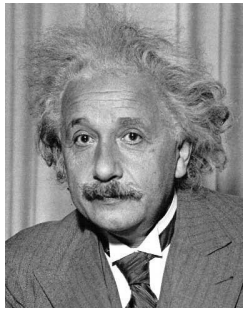
-1	0	1
-2	0	2
-1	0	1

y x

73

Đạo hàm bậc 1

- Sobel



*

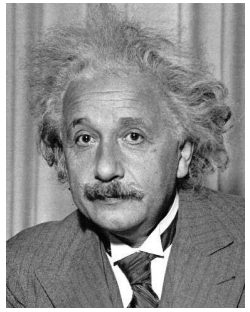
-1	0	1
-2	0	2
-1	0	1

Vertical Edge
(absolute value)

74

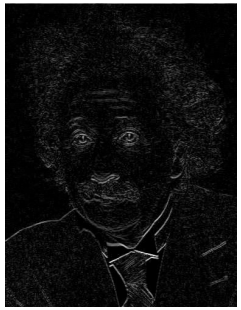
Đạo hàm bậc 1

- Sobel



*

-1	-2	-1
0	0	0
1	2	1



Horizontal Edge
(absolute value)

75

Đạo hàm bậc 2 - Laplacian filtering

- Bộ lọc Laplace:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

76

Đạo hàm bậc 2 - Laplacian filtering

- Có thể được tính như sau:

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

0	1	0
1	-4	1
0	1	0

- Hoặc

$$\nabla^2 f = 4f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

0	-1	0
-1	4	-1
0	-1	0

- 90° isotropic filter

77

Đạo hàm bậc 2 - Laplacian filtering

- Có thể được tính với:

$$\nabla^2 f = [f(x+1,y+1) + f(x+1,y) + f(x+1,y-1) + f(x-1,y+1) + f(x-1,y) + f(x-1,y-1) + f(x,y+1) + f(x,y-1)] - 8f(x,y)$$

1	1	1
1	-8	1
1	1	1

- Hoặc

$$\nabla^2 f = 8f(x,y) - [f(x+1,y+1) + f(x+1,y) + f(x+1,y-1) + f(x-1,y+1) + f(x-1,y) + f(x-1,y-1) + f(x,y+1) + f(x,y-1)]$$

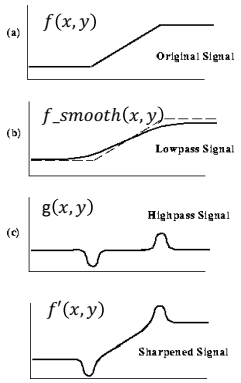
-1	-1	-1
-1	8	-1
-1	-1	-1

- 45° isotropic filter

78

Làm sắc nét cạnh (sharpening filter) sử dụng đạo hàm bậc 1

$$g(x,y) = f(x,y) - f_{smooth}(x,y)$$



$$f'(x,y) = f(x,y) + k * g(x,y)$$

(k = 0.2..0.7)

79

Làm sắc nét cạnh (sharpening filter) sử dụng bộ lọc laplacian

PP phổ biến: ảnh gốc – đạo hàm bậc 2(scaled)

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & -sign \\ f(x,y) + \nabla^2 f(x,y) & +sign \end{cases}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & +5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad 90^\circ \ isotropic \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad 45^\circ \ isotropic$$

80

Note

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

0	-1	0
-1	5	-1
0	-1	0

=

0	0	0
0	1	0
0	0	0

+

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

=

0	0	0
0	1	0
0	0	0

+

-1	-1	-1
-1	8	-1
-1	-1	-1

81

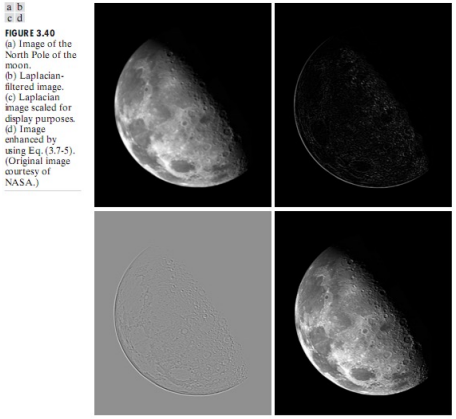
Sharpening filter

a) Original image

b) Laplacian filtering

c) Laplacian with scaling

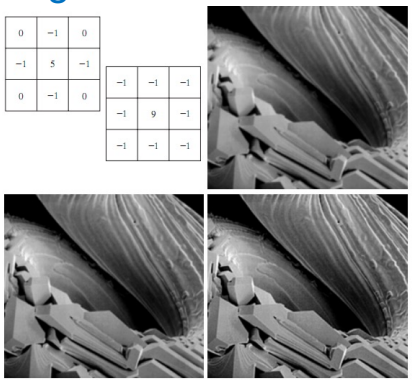
d) Adding original with Laplacian image



a b c d
FIGURE 3.40 (a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

82

Sharpening filter



a b c d e
FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

83

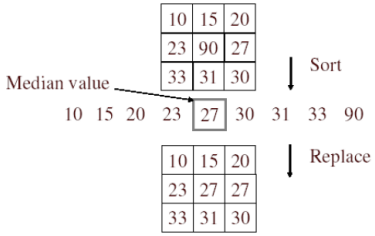
Nội dung

- Nhắc lại: biểu diễn ảnh số
- Tăng cường độ tương phản trên ảnh
  - Kéo giãn histogram
  - Cân bằng histogram
  - Biến đổi phi tuyến
- Nhân chập và lọc ảnh
  - Nhân chập
  - Một số bộ lọc tuyến tính
- Một số phép toán khác

84

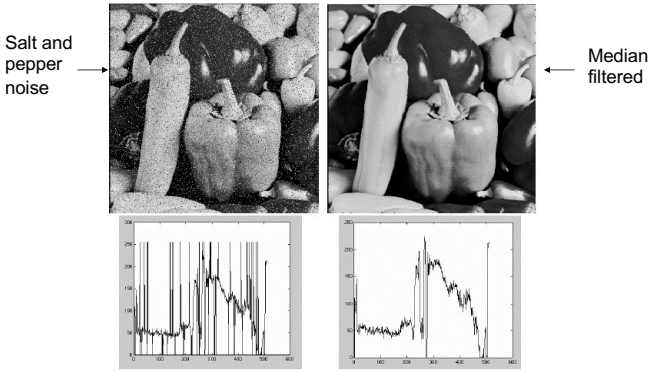
Bộ lọc trung vị (median filter)

- Không có giá trị mới nào được tính
- Loại bỏ những điểm bất thường:
  - Phù hợp nhiễu xung và nhiễu muối tiêu (salt & pepper)
- Bộ lọc phi tuyến



85

Bộ lọc trung vị (median filter)



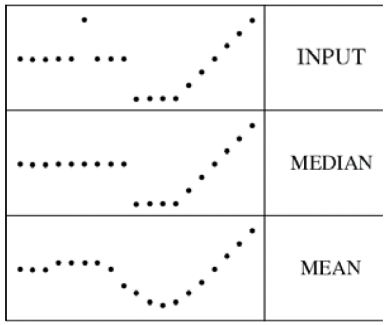
Plots of a row of the image

Source: M. Hebert

86

Bộ lọc trung vị (median filter)

- Median filter is edge preserving



87

Bộ lọc Max/ Min

– Bộ lọc min: thay giá trị ở giữa bằng giá trị nhỏ nhất trong cửa sổ xét

Original Image | with Minimum Filter

– Bộ lọc max: thay giá trị ở giữa bằng giá trị lớn nhất trong cửa sổ xét

Original Image | with Maximum Filter

88

Bộ lọc Max/ Min

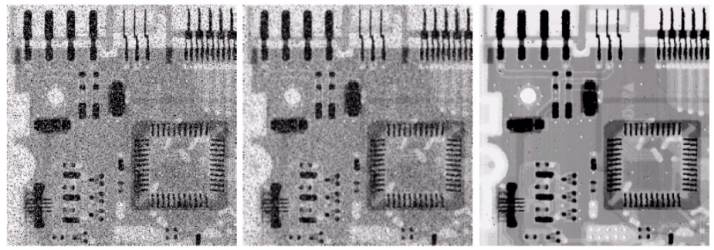
– Bộ lọc min: thay giá trị ở giữa bằng giá trị nhỏ nhất trong cửa sổ xét

Original Image | with Minimum Filter

– Bộ lọc max: thay giá trị ở giữa bằng giá trị lớn nhất trong cửa sổ xét

Original Image | with Maximum Filter

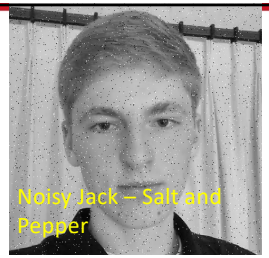
89

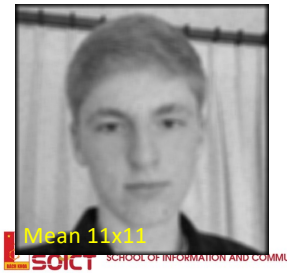


a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

90





91


Initial image


Adding Salt and pepper noise


Mean filter 3x3


Min filter 3x3

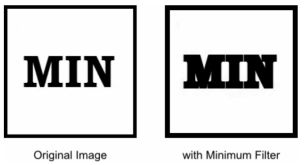

Max filter 3x3


Median filter 3x3

92

Median/max/min: problem?

- Size filter !



Original Image | with Minimum Filter

Original Image | with Maximum Filter

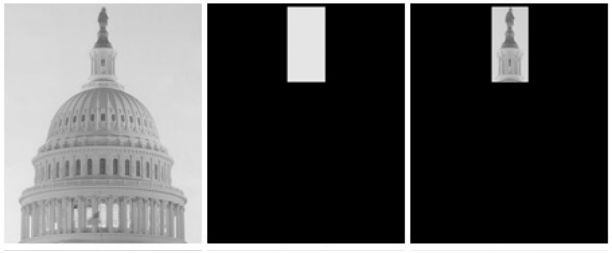
93

Arithmetical/Logical Operations

- AND operation
- OR operation
- Image subtraction
- Image addition

94

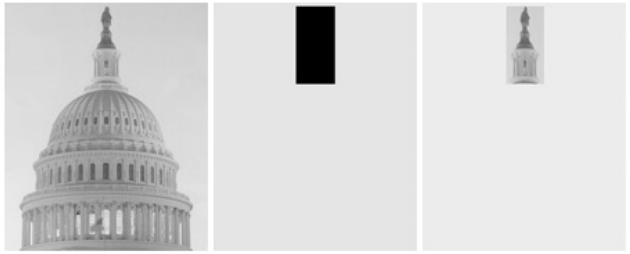
AND operation



Original | And mask | Output image

95

OR operation



Original | OR mask | Output image

96

Image Addition

- If f and g are two images, the pixelwise addition R is defined as:
  - R(x,y) = Min(f(x,y)+g(x,y) ; 255)
- Image addition is used to
  - lower the noise in a serie of images
  - increase the luminance by adding the image to itself

Source : Eric Favier. L'analyse et le traitement des images. ENISE.

97

Average Images

- g(x,y) is the addition of f(x,y) and noise $\eta(x,y)$

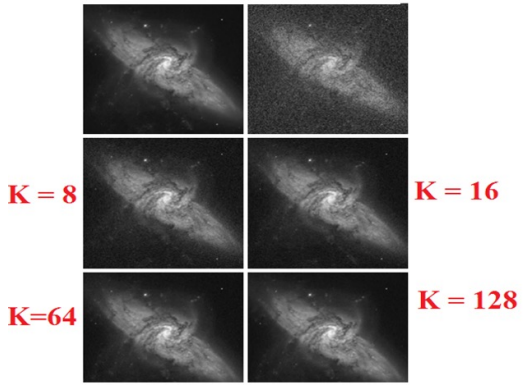
$$g(x,y) = f(x,y) + \eta(x,y)$$

- If we have several images {g(x,y)}, we can compute the average one

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

98

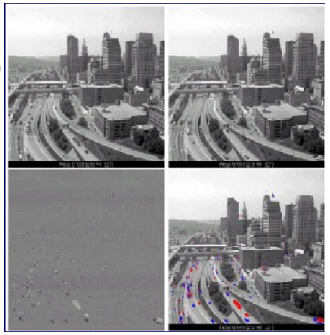
Average Images



99

Image subtraction

- The pixelwise substraction of two images f and g is:
  S(x,y) = Max(f(x,y)-g(x,y) ; 0)
- Image substraction is used to
  - detect defaults, detect difference between images
  - detect motion in images



Source : Eric Favier. L'analyse et le traitement des images. ENISE.

100

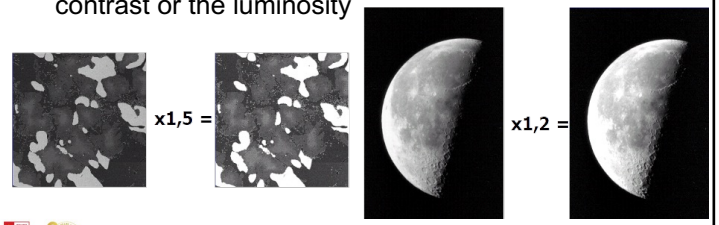
Image subtraction

After detection, we still have some noise, that we can clean to keep only the object of interest

101

Image multiplication

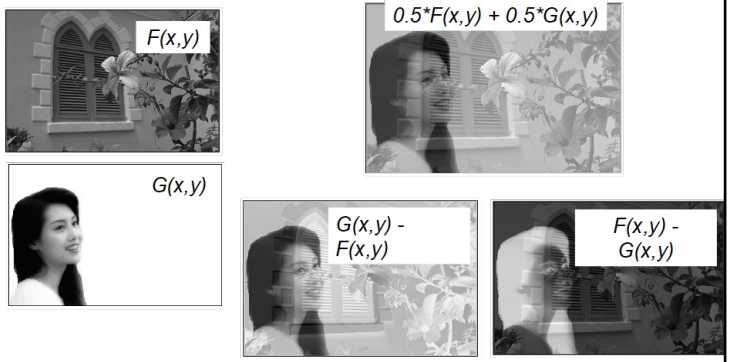
- The multiplication S of an image f by a ratio (factor) is defined as:
  - S(x,y) = Max(f(x,y)*ratio ; 255)
- Image multiplication can be used to increase the contrast or the luminosity



Source : Eric Favier. L'analyse et le traitement des images. ENISE.

102

Operations on images

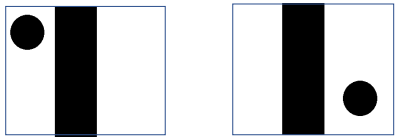


Source : www.nte.montaigne.u-bordeaux.fr/SuppCours/5314/Dai/TraitImage01-02.ppt

103

Exercise

- Given two images as bellow



1. Transform images to negative ones
2. Process to have an image which has only the "ball"

104

Binary images

105

Binary images

- Two pixel values: foreground (object, 1) and background (0)
- Be used
  - To mark region(s) of interest
  - As results of thresholding method

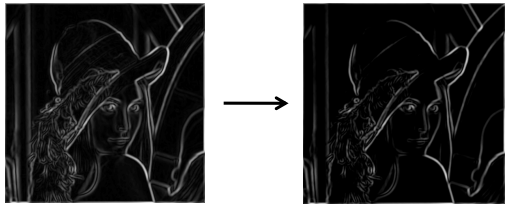
1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

106

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: edge detection



Gradient magnitude

```
fg_pix = find(gradient_mag > t);
```

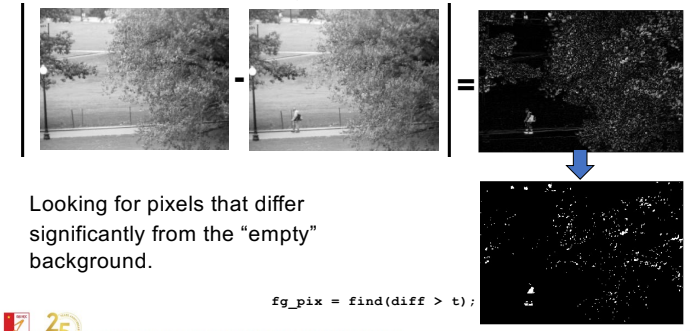
Looking for pixels where gradient is strong.

107

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: background subtraction



Looking for pixels that differ significantly from the "empty" background.

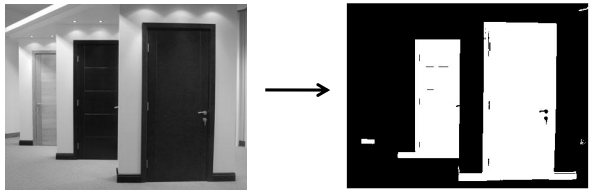
```
fg_pix = find(diff > t);
```


108

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: intensity-based detection



```
fg_pix = find(im < 65);
```

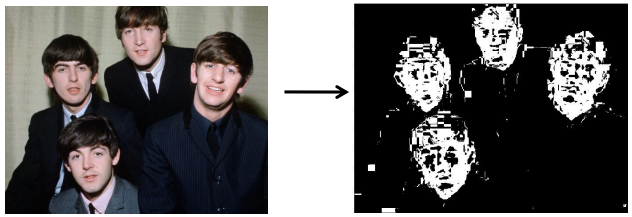
Looking for dark pixels

109

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: color-based detection



```
fg_pix = find(hue > t1 & hue < t2);
```

Looking for pixels within a certain hue range.

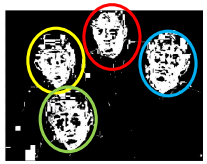
Slide credit: Kristen Grauman

110

Issues

- What to do with "noisy" binary outputs?
  - Holes
  - Extra small fragments

- How to demarcate multiple regions of interest?
  - Count objects
  - Compute further features per object



Slide credit: Kristen Grauman

111

Morphological operators

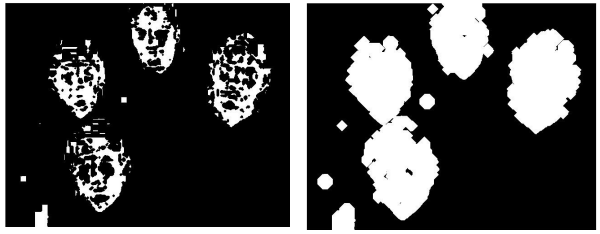
- Change the shape of the foreground regions via intersection/union operations between a scanning structuring element and binary image.
- Useful to clean up result from thresholding

- Main components
  - Structuring element
  - Operators:
    - Basic operators: Dilation, Erosion
    - Others: Opening, Closing, …

112

Dilation

- Expands connected components
- Grow features
- Fill holes



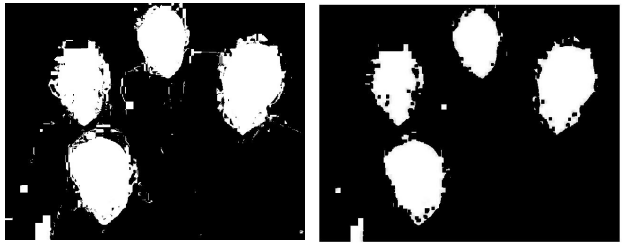
Before dilation | After dilation

Slide credit: Kristen Grauman

113

Erosion

- Erode connected components
- Shrink features
- Remove bridges, branches, noise



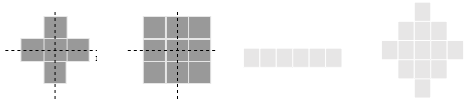
Before erosion | After erosion

Slide credit: Kristen Grauman

114

Structuring elements

- **Masks** of varying shapes and sizes used to perform morphology, for example:

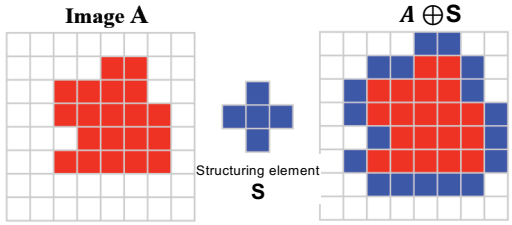


- Scan mask (structuring element) over the object (foreground) borders (inside and outside) and transform the binary image

115

Dilation

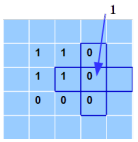
- Moving S on each pixel of A
  - check if the intersection (pixels belonging to object) is not empty
  - If yes, the center of B belongs to the result image
- If a pixel of S is onto object pixels (A), then the central pixel belongs to object
  - Otherwise (i.e. all pixels of are background), set to background (no change)

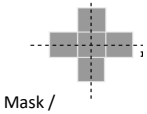


116

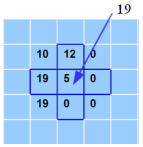
Dilation

- As max filter
- Can be applied both on
  - binary images
  - or grayscale images





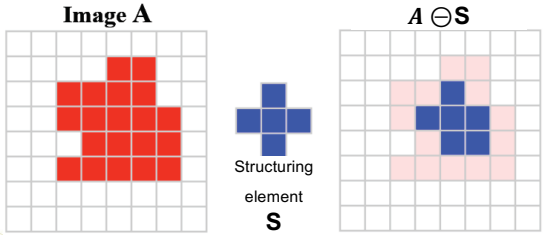
Mask /
Structuring element



117

Erosion

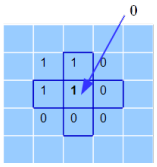
- We put the element S on each pixel x of A
  - like convolution
- If all pixels of S are onto object pixels (A), then the central pixel belongs to object
  - Otherwise (i.e. a mask pixel is background), set to background

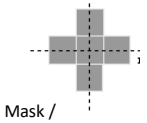


118

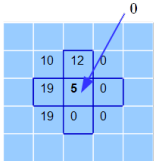
Erosion

- As min filter
- Can be applied both on
  - binary images
  - or grayscale images





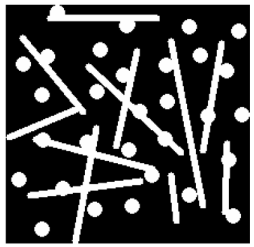
Mask /
Structuring element



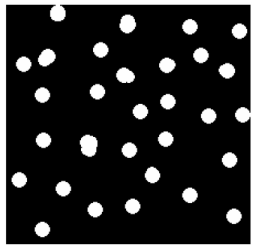
119

Opening

- Erode, then dilate
- Remove small objects, keep original shape



Before opening

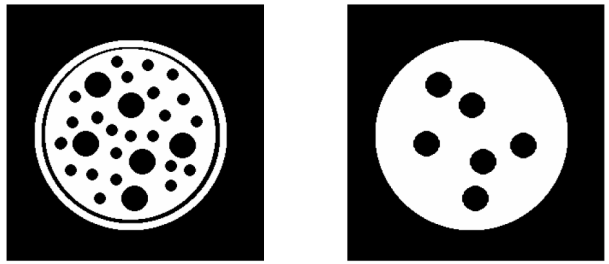


After opening

120

Closing

- Dilate, then erode
- Fill holes, but keep original shape

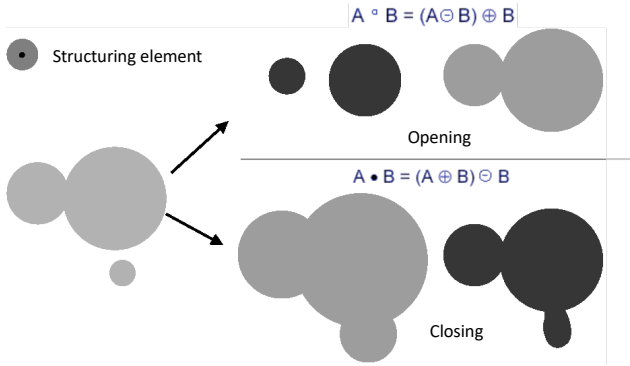


Before closing **After closing**

demo: http://bigwww.epfl.ch/demo/jmorpho/start.php

121

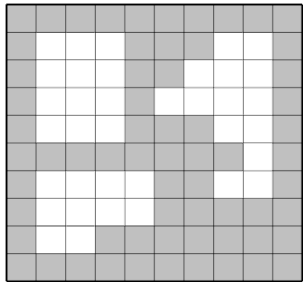
Opening vs Closing



122

Connected component labeling

- We loop over all the image to give a unique number (label) for each region
- All pixels from the same region must have the same number (label)
- Objectifs:
  - Counting objects
  - Separating objets
  - Creating a mask for each object
  - …

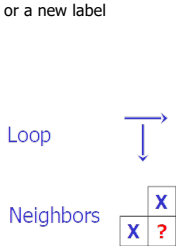


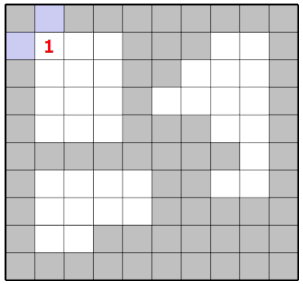
123

Connected component labeling

First loop over the image

- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label





124

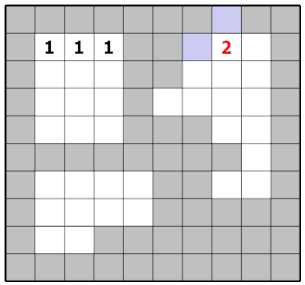
Connected component labeling

First loop over the image

- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label

Loop

Neighbors



125

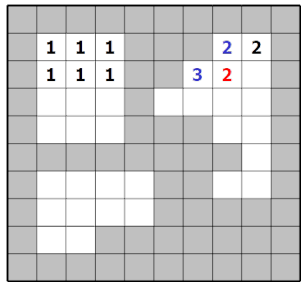
Connected component labeling

First loop over the image

- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label

Loop

Neighbors



126

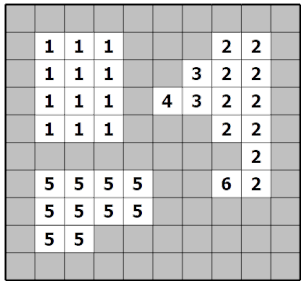
Connected component labeling

First loop over the image

- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label

Loop

Neighbors



127

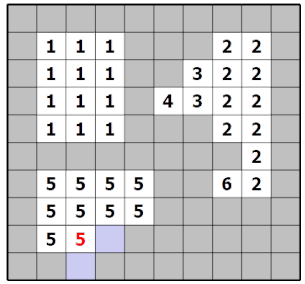
Connected component labeling

Second loop over the image

- For each pixel in a region, we set
  - the smallest from its own label and the labels from its down and right neighbors

Loop

Neighbors



128

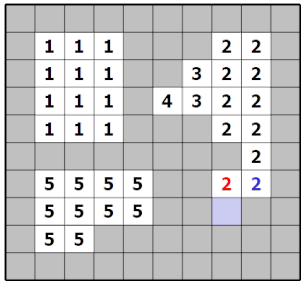
Connected component labeling

Second loop over the image

- For each pixel in a region, we set
  - the smallest from its own label and the labels from its down and right neighbors

Loop

Neighbors



129

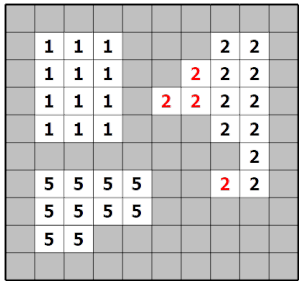
Connected component labeling

Second loop over the image

- For each pixel in a region, we set
  - the smallest from its own label and the labels from its down and right neighbors

Loop

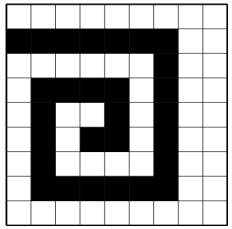
Neighbors

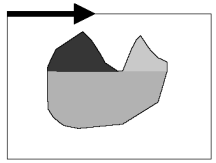


130

Connected component labeling

- Two loops are enough?
  - *example: spiral region !*
- Solutions
  - We continue, go and back two ways, until no new change in labels
  - It is possible to do only one loop: manage a table of equivalences when 2 different labels are neighbors

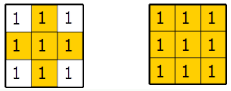


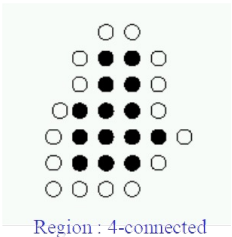


131

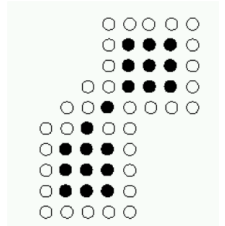
CC labeling: how many neighbors?

- Advice: Use different connexities for edges and regions
  - *4-Connexity for regions*
  - *8-Connexity for edges*





Region : 4-connected
Edge : 8-connected

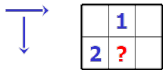


Region : 8-connected
Edge : 4-connected

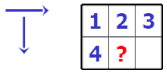
132

CC labeling: how many neighbors?

- Regions labeling
  - We use 4-connexity
  - Each loop, we compare 2 neighbors



- Edge labeling
  - 8-connexity
  - Each loop, we compare 4 neighbors



133

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!

soict.hust.edu.vn/ fb.com/groups/soict

134

134