HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thị giác máy tính

Ch4.1: Phát hiện biên

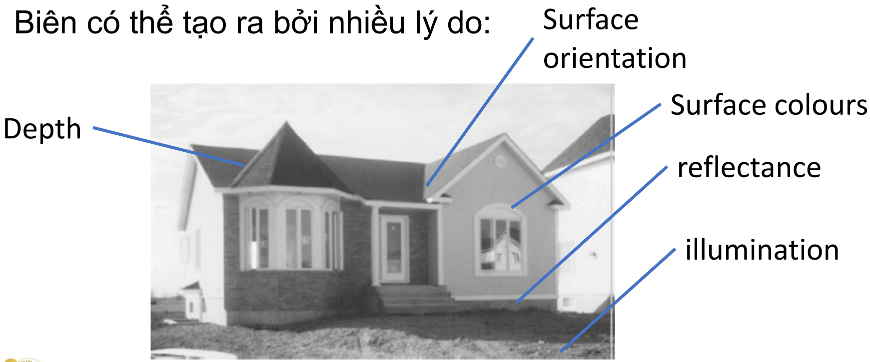
Nguyễn Thị Oanh
oanhnt@soict.hust.edu.vn

Nội dung

- Vai trò của biên và cách tiếp cận chung
- Phát hiện biên
  - Image gradient: sobel, prewitt
  - Canny detector
  - Laplacian
- Phát hiện đường thẳng
  - Hough transform
  - Ransac

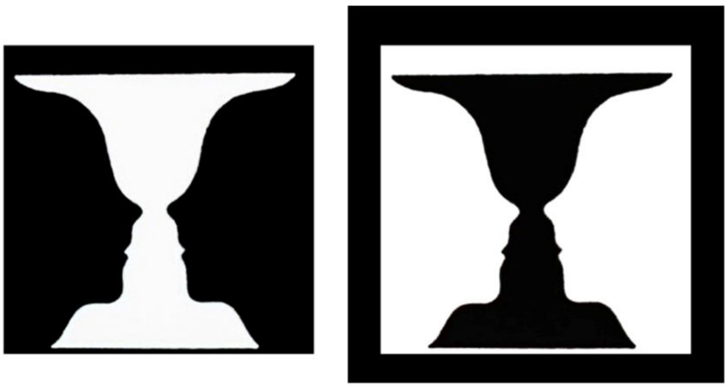
Biên là gì (edge/contour) ?

- Nơi có sự thay đổi cường độ sáng trong ảnh
- Thường sảy ra ở ranh giới giữa các vùng khác nhau trong ảnh
- Biên có thể tạo ra bởi nhiều lý do:



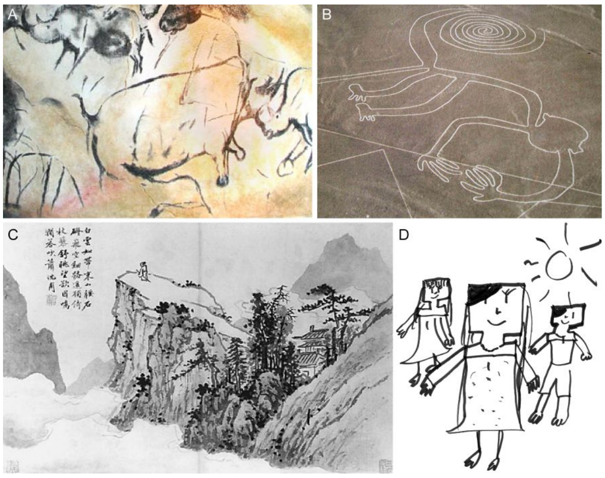
Vai trò của biên

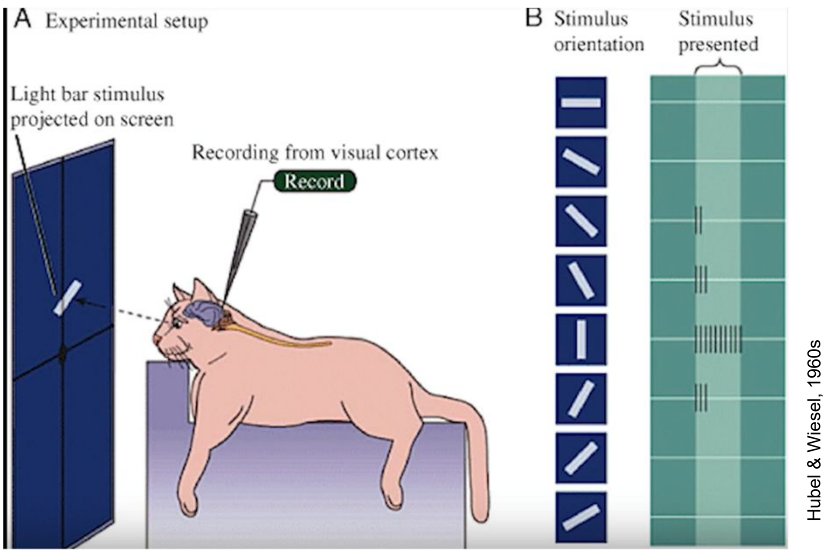
- What do you see ?

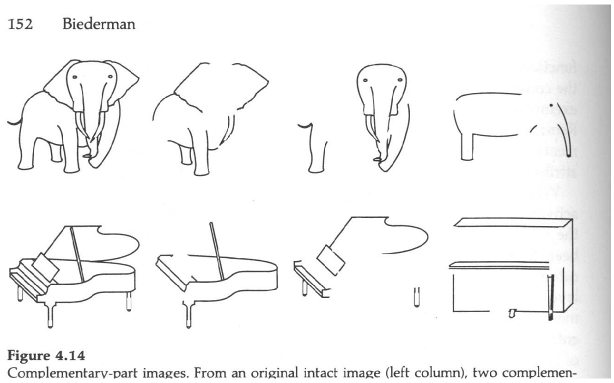


Vai trò của biên

(A)Cave painting at Chauvet, France, about 30,000 B.C.;
(B)Aerial photograph of the picture of a monkey as part of the Nazca Lines geoglyphs, Peru, about 700 – 200 B.C.;
(C)Shen Zhou (1427-1509 A.D.): Poet on a mountain top, ink on paper, China;
(D)Line drawing by 7-year old I. Lleras (2010 A.D.).



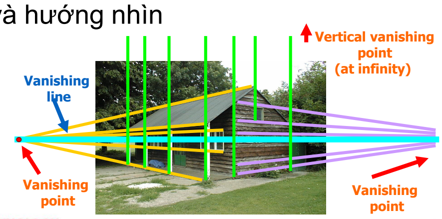




Can we recognize these objects?

Phát hiện biên

- Mục tiêu: xác định nơi có sự thay đổi cường độ sáng trên ảnh
  - Về mặt trực quan, thông tin ngữ nghĩa hoặc hình dáng trong ảnh được thể hiện thông qua biên
  - Biên thể hiện thông tin cấp cao hơn so với điểm
- Lý do?
  - Trích chọn thông tin, nhận dạng đối tượng
  - Xác định thông tin hình học và hướng nhìn



Source: J. Hayes

Phát hiện biên ?

- Intensity profile:
  - Biểu diễn giá trị của các điểm ảnh được lấy đều dọc theo một đường nào đó trên ảnh

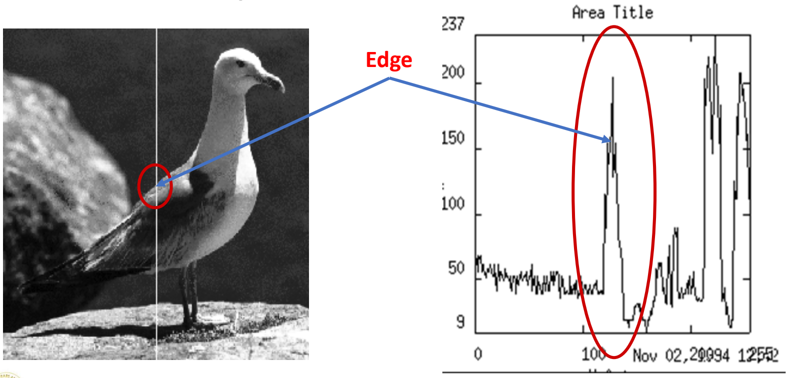
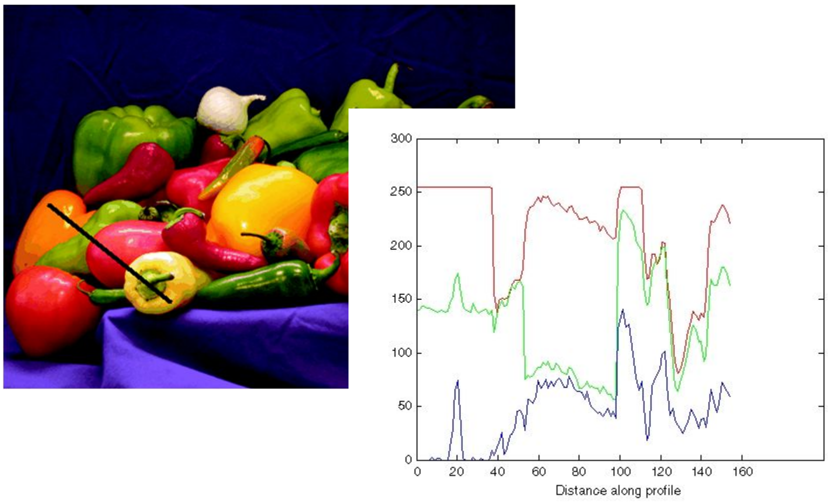


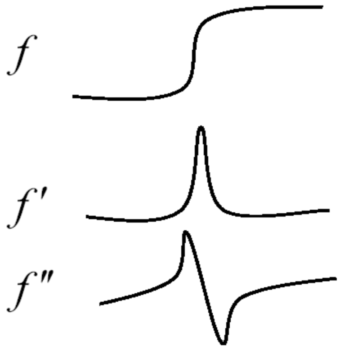
Image profile



https://www.mathworks.com/help/images/intensity-profile-of-images.html

Phát hiện biên ?

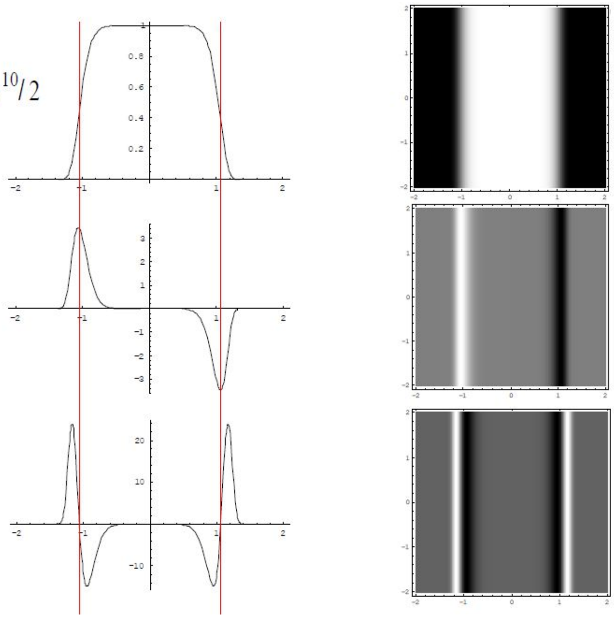
- Biên là vị trí có sự thay đổi nhanh về cường độ
  - Đạt cực trị trên đạo hàm bậc 1
  - Đi qua không (zero-crossing) ở đạo hàm bậc 2



$$f(x,y)=e^{-x^{10}/2}$$

$$\frac{\partial f}{\partial x}$$

$$\frac{\partial^2 f}{\partial x^2}$$



Image

The first derivative

The second derivative

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

Đạo hàm bậc 1

- Derivative in 1D:

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x$$

- Discrete derivative in 1D

$$\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)$$

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Một số cách tính đạo hàm bậc 1

Backward $\frac{df}{dx} = f(x) - f(x-1) = f'(x)$

Forward $\frac{df}{dx} = f(x) - f(x+1) = f'(x)$

Central $\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$

Bộ lọc tính đạo hàm bậc 1

- Backward filter:

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x) \qquad [0 \quad 1 \quad -1]$$

- Forward:

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x) \qquad [-1 \quad 1 \quad 0]$$

- Central:

$$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x) \qquad [1 \quad 0 \quad -1]$$

Ví dụ:

- Backward filter: [0 1 -1]

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$
$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

f(x) :	**0**	**0**	**0**	**0**	**0**	**50**	**50**	**50**	**50**	**50**
f'(x):	**0**	**0**	**0**	**0**	**0**	**50**	**0**	**0**	**0**	**0**

Image gradient

Đạo hàm rời rạc trên ảnh

Given function $f(x,y)$

Gradient vector $\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$

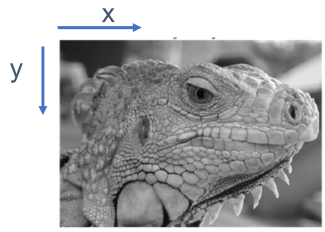
Gradient magnitude $|\nabla f(x,y)| = \sqrt{f_x^2 + f_y^2}$

Gradient direction $\theta = \tan^{-1} \dfrac{f_x}{f_y}$

Đạo hàm bậc 1 theo x và y

0	1	-1

0
1
-1



-1	1	0

-1
1
0

1
0
-1

1	0	-1

Đạo hàm bậc 1 theo x và y

- Robert filter (the first approximation filter for image derivative - 1965)

$$\begin{matrix} \boxed{0} & 1 \\ -1 & 0 \end{matrix} \qquad \begin{matrix} \boxed{1} & 0 \\ 0 & -1 \end{matrix}$$

- Prewitt filter

1/3 x

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

1/3 x

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

- Sobel filter

1/4 x

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

1/4 x

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

Image I (9 x 8)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100

M_x: Derivative filter wrt x

1	**0**	**-1**

1
0
-1

M_y: Derivative filter wrt y

Ix = I * Mx = the 1st derivative wrt x

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	100	100	0	0	0	0
0	100	100	0	0	0	0
0	100	100	0	0	0	0
0	100	100	0	0	0	0
0	100	100	0	0	0	0

Image I (9 x 8)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100
0	0	0	100	100	100	100	100	100

M_x: Derivative filter wrt x

1	**0**	**-1**

1
0
-1

M_y: Derivative filter wrt y

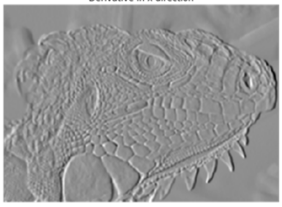
Iy = I * My = the 1st derivative wrt y

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	100	100	100	100	100
0	0	100	100	100	100	100
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

3x3 image gradient filters

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

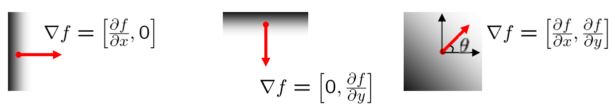
Derivative in x direction



Derivative in y direction

Image gradient

- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

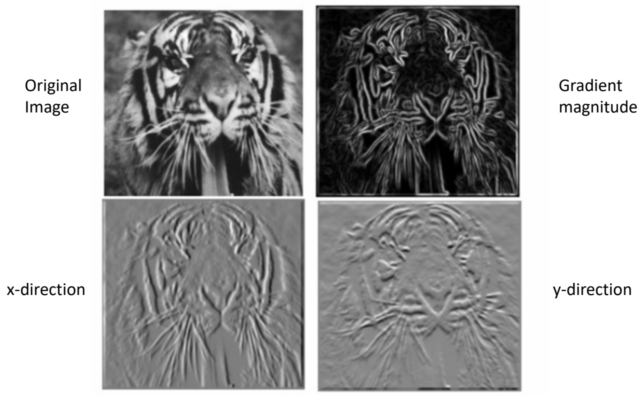


- The **gradient vector** points in the direction of most rapid increase in intensity
- The **gradient direction** is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$
  - how does this relate to the direction of the edge?
- The *edge strength* is given by the **gradient magnitude**

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad \approx \ |\partial f/\partial x| + |\partial f/\partial y|$$

Source: Steve Seitz

Example

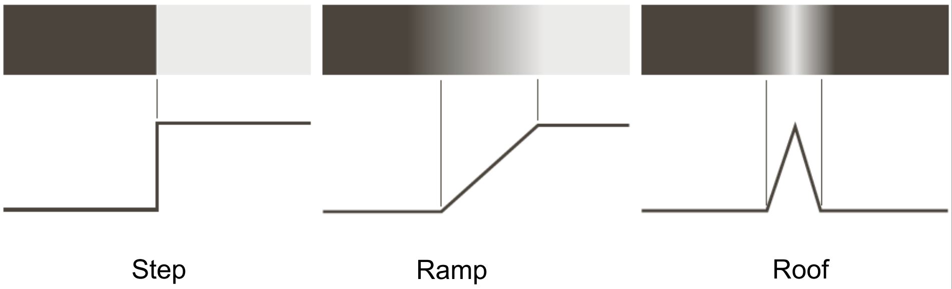


- Which one is the gradient in the x-direction? How about y-direction?

CS231: Lecture 3 Juan Carlos Niebles and Ranjay Krishna

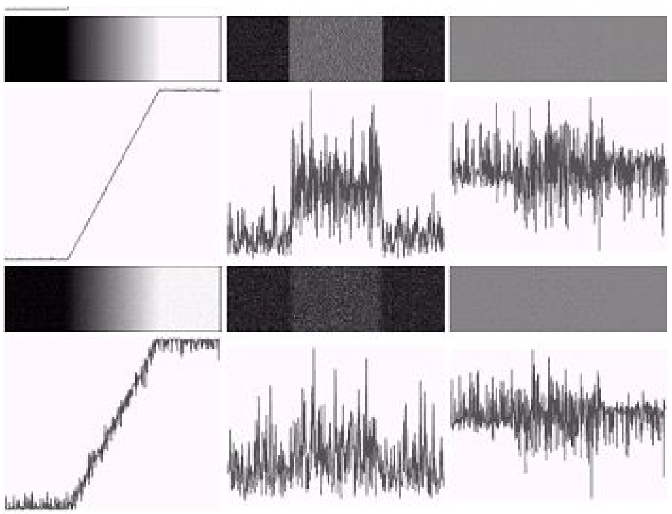
Phát hiện biên

Các kiểu biên



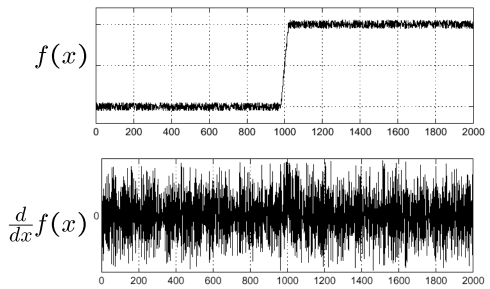
Source : Gonzalez and Woods. Digital Image Processing 3ed. Prentice-Hall, 2008.

Nhiễu trên biên



Source : Gonzalez and Woods. Digital Image Processing 3ed. Prentice-Hall, 2008.

Ảnh hưởng của nhiễu

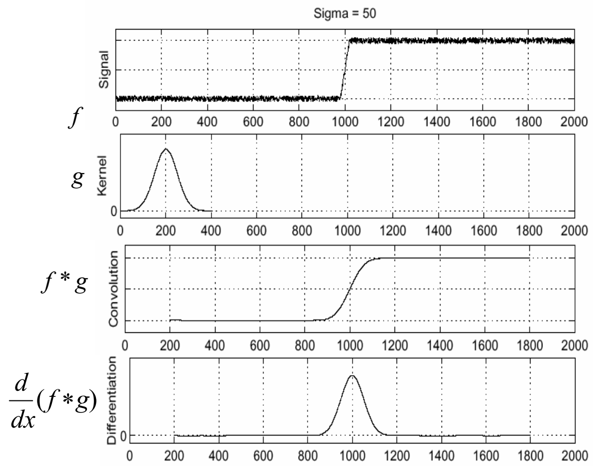


Where is the edge?

➔ **Solution: làm trơn ảnh trước**

Source: S. Seitz

Solution: làm trơn + tính đạo hàm



- To find edges, look for peaks in $\frac{d}{dx}(f * g)$

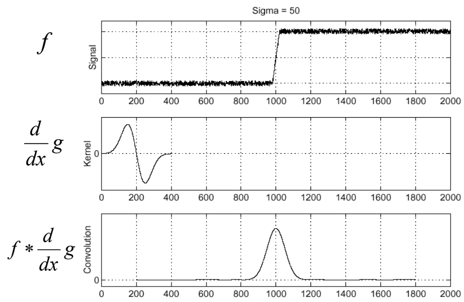
Source: S. Seitz

Derivative theorem of convolution

- Tính chất hữu ích:

$$\frac{d}{dx}(f * g) = \frac{df}{dx} * g = f * \frac{dg}{dx}$$

- Điều này giúp giảm phép toán:



Source: S. Seitz

Sobel Operator

- Gaussian Smoothing + differentiation

Gaussian smoothing

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} [1 \quad 0 \quad -1]$$

differentiation

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} [1 \quad 2 \quad 1]$$

➔ **Less sensible to noise**

Prewitt Operator

- Mean smoothing + differentiation

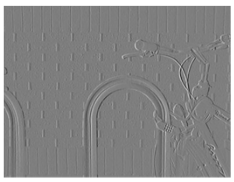
$$Gx = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [1 \quad 0 \quad -1]$$

$$Gy = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} [1 \quad 1 \quad 1]$$

➔ **Less sensible to noise**

Phát hiện cạnh đơn giản sử dụng đạo hàm bậc 1

- Nhân chập ảnh với 2 mặt nạ để xấp xỉ đạo hàm bậc 1 theo x và y



- Tính độ lớn của gradient

- Lấy ngưỡng: cạnh là điểm có độ lớn gradient > T

Phát hiện cạnh đơn giản sử dụng đạo hàm bậc 1

Original image

Gradient magnitude using Sobel operator

Threshold T = 25

Threshold T = 60

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

Không làm trơn trước khi tính đạo hàm

a b
c d

FIGURE 10.16 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

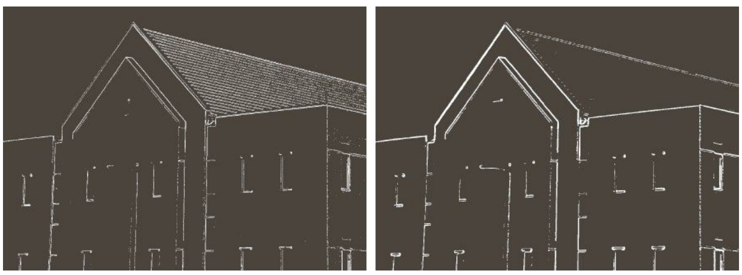
Làm trơn trước khi tính đạo hàm

a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5 × 5 averaging filter prior to edge detection.

Không làm trơn trước tính đạo hàm

Làm trơn trước tính đạo hàm

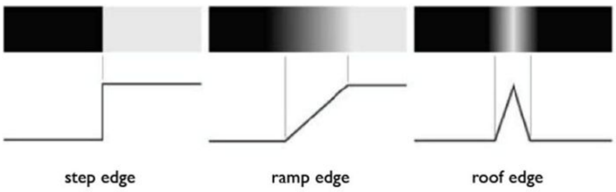


a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

Vấn đề

- Vị trí không chính xác (biên dày)
- Giá trị ngưỡng ưu ái cạnh theo 1 vài hướng hơn là các hướng khác
  - Có thể thiếu các đường biên chéo hơn là biên ngang hoặc dọc ➔ bỏ sót biên



Một số mặt nạ khác

- Để tránh ưu ái cạnh theo 1 vài hướng ➔ sử dụng kỹ thuật la bàn:
  - Nhân chập ảnh với 8 mặt nạ theo 8 hướng (0, 45, 90, ..)
  - Cộng kết quả nhận chập lại

$$H_1 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \qquad H_2 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \qquad H_4 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix}$$

$$H_5 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix} \qquad H_6 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix}$$

$$H_7 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \qquad H_8 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix}$$

Một số mặt nạ khác

- Mặt nạ Prewitt, Sobel cho phát hiện biên chéo

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

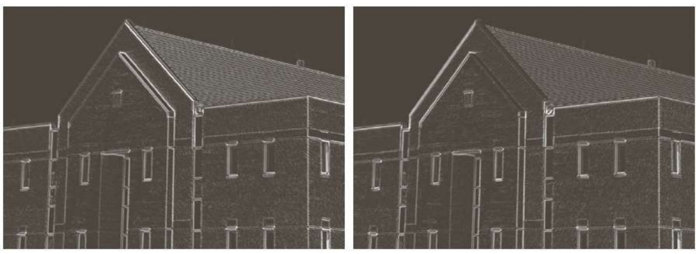
Sobel

a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

Một số mặt nạ khác

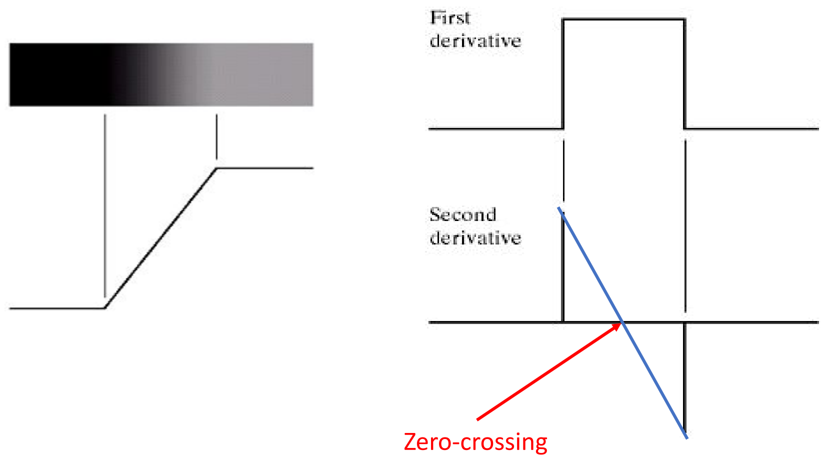
- Mặt nạ Sobel cho phát hiện biên chéo



a b

FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

Phát hiện biên với đạo hàm bậc 2



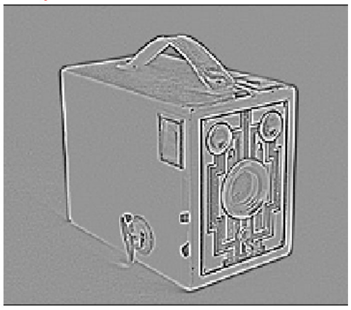
Phát hiện biên với đạo hàm bậc 2

- Sử dụng bộ lọc Laplacian:
  - Nhân chập ảnh với một trong 2 mặt nạ

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Phát hiện biên:
  - Tính đạo hàm bậc 2
  - Tìm điểm qua 0 ➔ cạnh

Phát hiện biên với đạo hàm bậc 2



- Một đáp ứng biên
- Nhạy với nhiễu

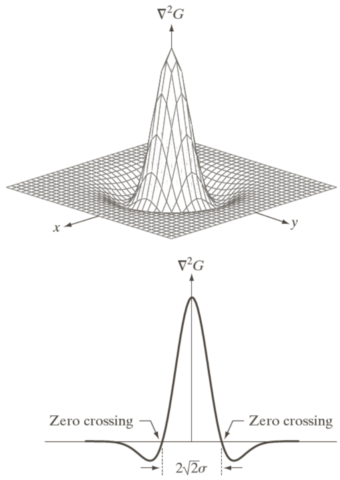
Laplace of Gaussian (LoG)

- Laplace nhạy với nhiễu ➔ làm trơn ảnh với trước khi nhân chập với laplace
- I * G * L = I * (G*L) ➔ G*L: bộ lọc LoG
- Đạo hàm bậc 2 = Ảnh * LoG

$$LoG(x,y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

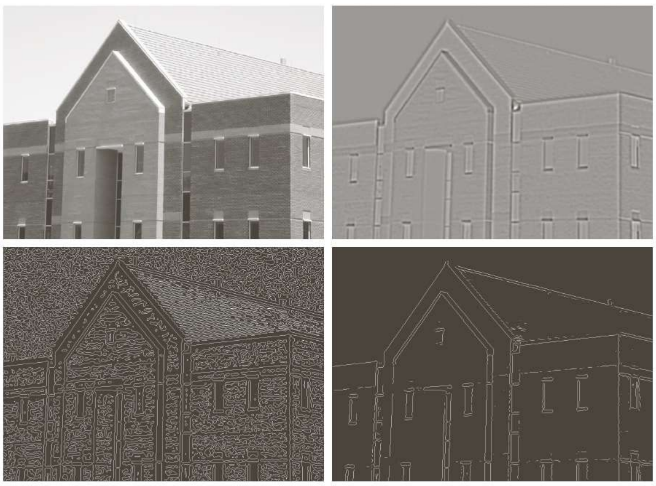
Laplace of Gaussian (LoG)



0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5 × 5 mask approximation to the shape in (a). The negative of this mask would be used in practice.



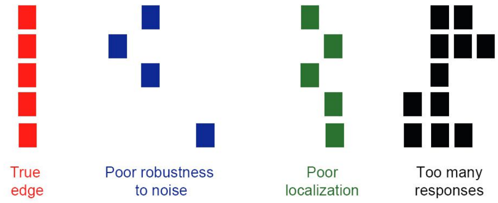
a b
c d

FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Còn gọi phương pháp Marr–Hildreth (tên 2 tác giả)

Bộ phát hiện cạnh "tối ưu"

- Tiêu chí:
  - **Good detection:** bộ phát hiện tối thiểu "nhận nhầm" cũng như "bỏ sót"
  - **Good localization:** cạnh phát hiện phải gần biên đúng nhất
  - **Single response:** bộ phát hiện 1 điểm duy nhất tại vị trí biên; nghĩa là tối thiểu số lượng các cực đại địa phương xung quanh cạnh



Bộ phát hiện Canny

- Được sử dụng phổ biến nhất
- Giả thiết:
  - step-edges corrupted by additive Gaussian noise
- Canny:
  - Sử dụng đạo hàm bậc 1
  - Đạo hàm bậc 1 Gaussian xấp xỉ gần đúng phép toán tối ưu tích của tỷ số tín hiệu trên nhiễu và vị trí cạnh
- Đặc điểm :
  - Detection: phát hiện được biên yếu, nhưng loại được nhiễu
  - Good location: gần với biên thật
  - Unique response: độ dày biên = 1

J. Canny, *A Computational Approach To Edge Detection*, IEEE Trans. PAMI, 8:679-714, 1986.

Bộ phát hiện biên: Các bước chính

1) Áp dụng lọc với bộ lọc Gaussian

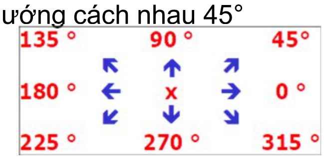
- Bộ lọc thông thấp để lọc nhiễu

2) Tính độ lớn gradient của các điểm ảnh

- Áp dụng bộ lọc Sobel theo hướng X và Y
- Tính độ lớn grandient | G | = |Gx | + |Gy |

3) Tính hướng gradient

- Hướng gradient θ = arctan (Gy / Gx)
- Làm tròn về 8 hướng cách nhau 45°

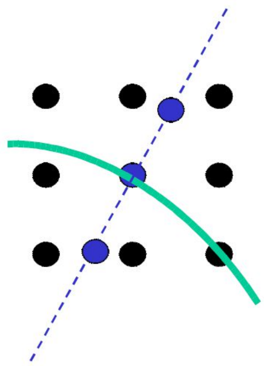


Bộ phát hiện biên: Các bước chính

4) Loại bỏ các điểm không phải cực trị (non-maxima suppression)

- Nếu độ lớn gradient tại điểm (x,y) bé hơn 1 trong 2 hàng xóm theo hướng gradient

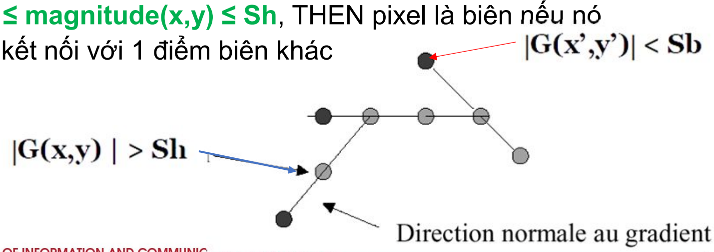
➔ đặt giá trị độ lớn gradient tại (x,y) bằng 0



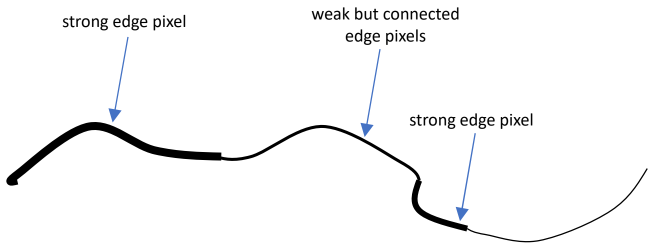
Bộ phát hiện biên: Các bước chính

5) Lấy ngưỡng (hysteresis thresholding)

- Sử dụng 2 ngưỡng: một ngưỡng cao (Sh) và một ngưỡng thấp (Sb)
- Tại mỗi điểm:
  - IF magnitude(x,y) < Sb, THEN đặt pixel = 0 (không phải biên)
  - IF magnitude(x,y) > Sh, THEN đặt pixel = 1 (biên)
  - IF **Sb ≤ magnitude(x,y) ≤ Sh**, THEN pixel là biên nếu nó được kết nối với 1 điểm biên khác

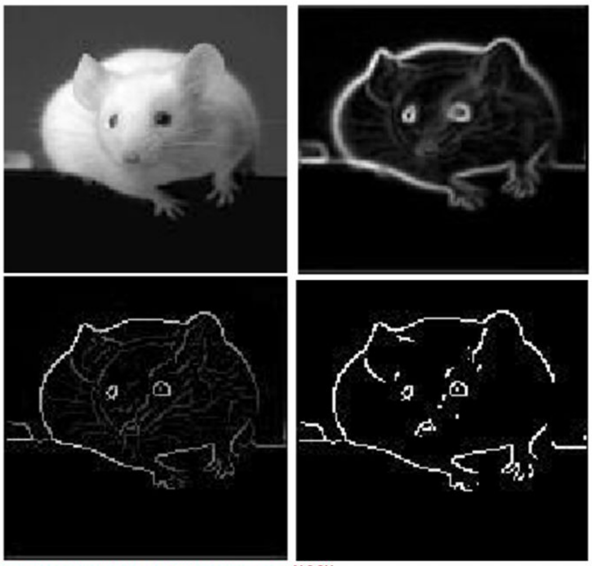


Hysteresis thresholding



Source: S. Seitz

Canny detector



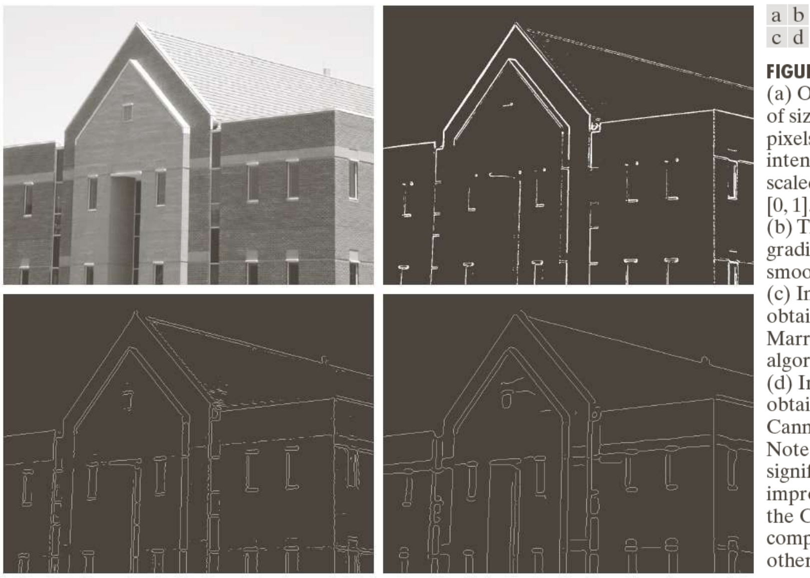
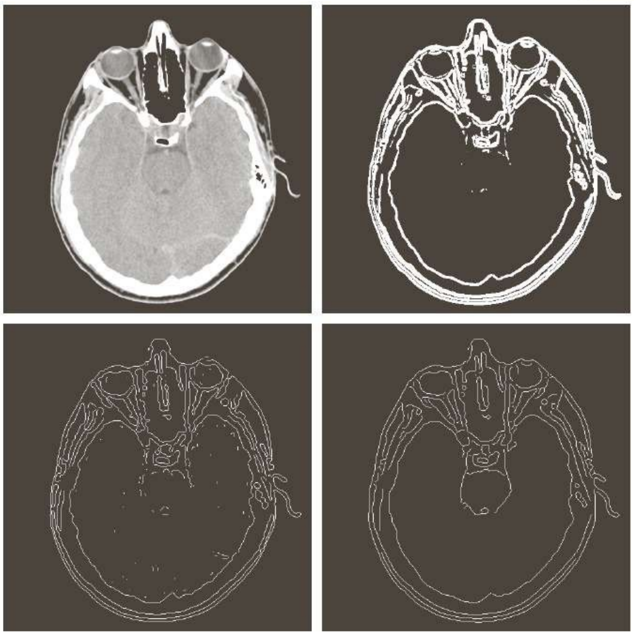


FIGURE 10.25 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

$T_L = 0.04; T_H = 0.10; \sigma = 4$ and a mask of size 25×25

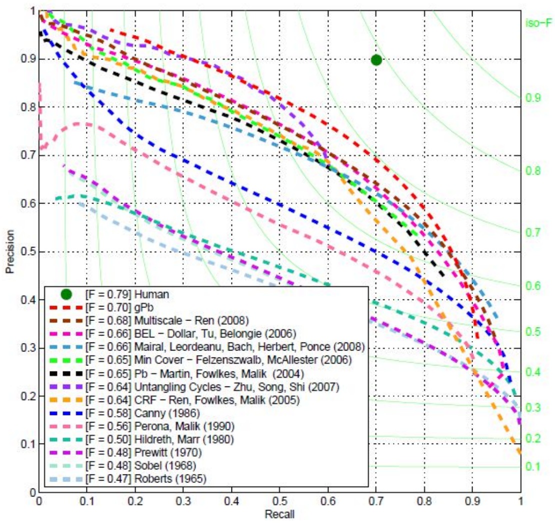


a b
c d

FIGURE 10.26 (a) Original head CT image of size 512 × 512 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

$T_L = 0.05; T_H = 0.15; \sigma = 2$ and a mask of size 13×13

45 years of boundary detection



Source: Arbelaez, Maire, Fowlkes, and Malik. TPAMI 2011 (pdf)

Nội dung

- Vai trò của biên và cách tiếp cận chung
- Phát hiện biên
  - Image gradient: sobel, prewitt
  - Canny detector
  - Laplacian
- Phát hiện đường thẳng
  - Hough transform
  - Ransac

Hough transform (HT)

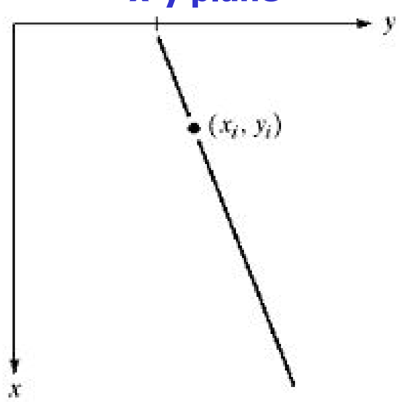
- HT
  - Dùng để phát hiện các đường thẳng
  - Giới thiệu năm 1962 (Hough 1962) và đầu tiên được sử dụng để phát hiện đường năm 1972 (Duda 1972)
  - Goal: tìm vị trí các đường thẳng trong ảnh
- HT có thể phát hiện các đường thẳng (*lines*), đường tròn (*circles*) và các cấu trúc khác (*structures*) CHỈ nếu *phương trình tham số là xác định*
- Phát hiện **hiệu quả** khi có *nhiễu* hay bị *che khuất 1 phần*

Hough transform

- **Global** approach to detect continuous edges
  - *From the x-y plane to the parametric plane a-b*
- ***x-y plane***
  - **$y_i = a\,x_i + b$**
  - an infinity of lines going though one (x_i, y_i) pair
  - one sole line for the (a,b) pair
- ***a-b parametric plane***
  - **$b = -\,x_i\,a + y_i$**
  - one sole line for the (x_i, y_i) pair
  - an infinity of lines going through one (a,b) pair

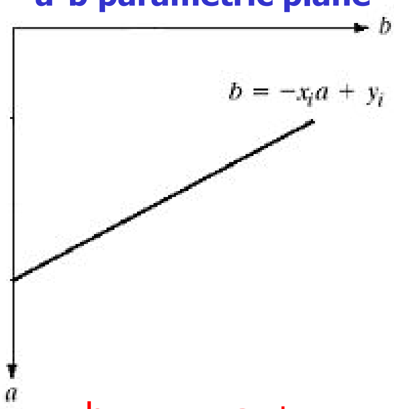
x-y plane vs a-b plane

x-y plane



$y_i = a\,x_i + b$

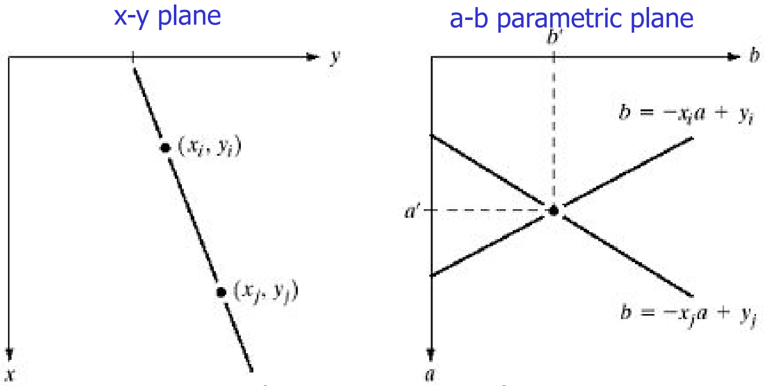
a-b parametric plane



$b = -\,x_i\,a + y_i$

Line vs Points

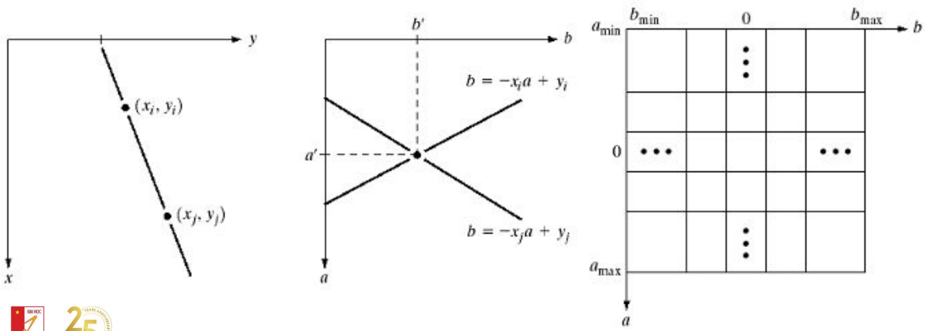
All the points (x,y) on a line in the x-y plane are going through one sole point (a', b') in the a-b parametric plane



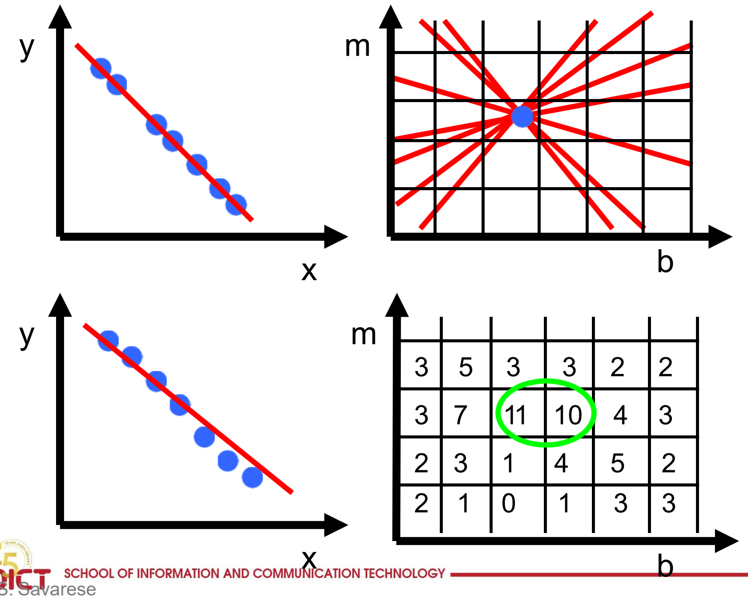
$y_i = a\ x_i + b$ $\qquad$ $b = -\ x_i\ a + y_i$

HT: ý tưởng chính

- Tạo ma trận tính tích lũy - M (a,b)
- **Bỏ phiếu**
  - *Mỗi điểm bỏ phiếu cho đường đi qua nó*
- Đường nào nhận được nhiều phiếu thì được giữ lại



Hough transform



Slide from S. Savarese

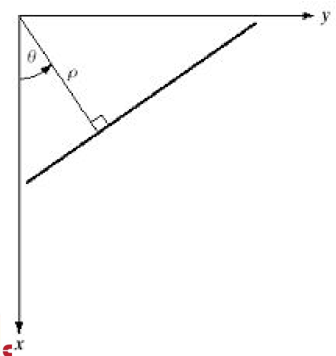
Thực hiện biến đổi Hough

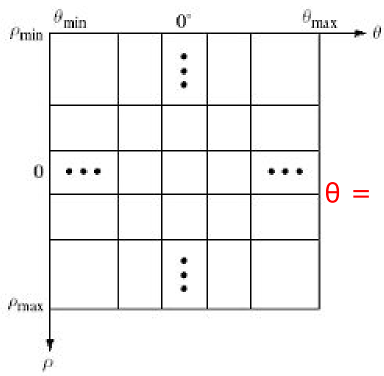
- Xác định các điểm biên của ảnh đầu vào
  - *Sobel, Prewitt, Canny, …*
- Với mỗi điểm biên, tính 1 đường trong không gian (a,b)
  - *1 đường thẳng trong kgian a-b tương ứng 1 điểm biên trong kgian (x,y)*
- Các *điểm đạt cực đại* trong *không gian tham số a-b* tương ứng với các đường có nhiều điểm nhất trong không gian x-y
  - *Điểm giao của các đường trong kgian a-b là tương ứng các đường thẳng trong kgian x-y*

Vấn đề với không gian (a,b)

- Problem: for a vertical line, a=∞ !
- Solution: representing using polar coordinates (ρ,θ)

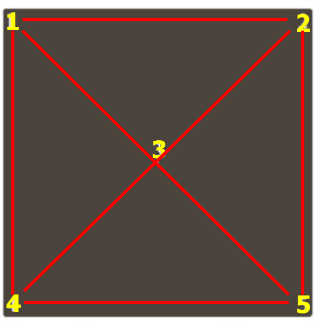
$$\rho = x \cos \theta + y \sin \theta$$



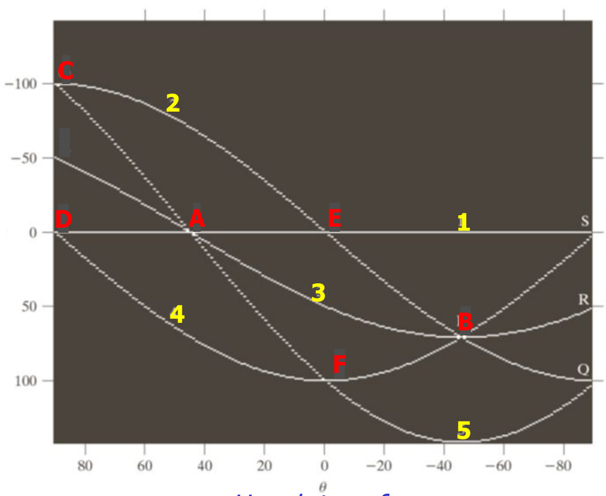


θ = ±90

HT: ví dụ



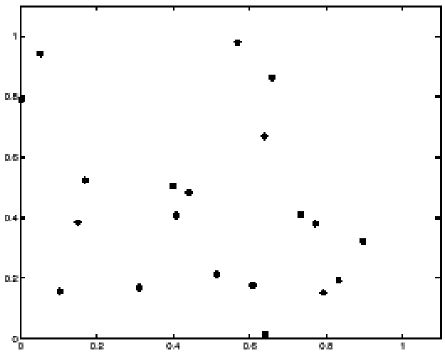
Ảnh với 5 điểm

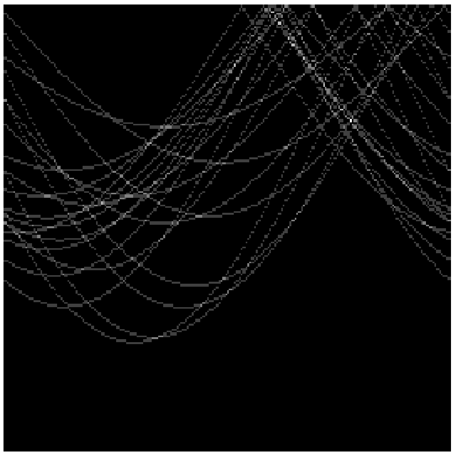


Hough transform

Source : Gonzalez and Woods. Digital Image Processing 3ed. Prentice-Hall, 2008.

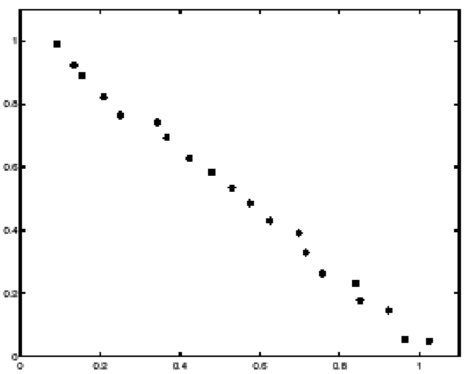
HT: ví dụ với tập điểm ngẫu nhiên

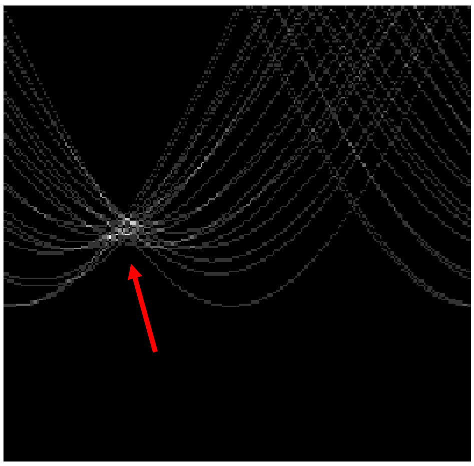




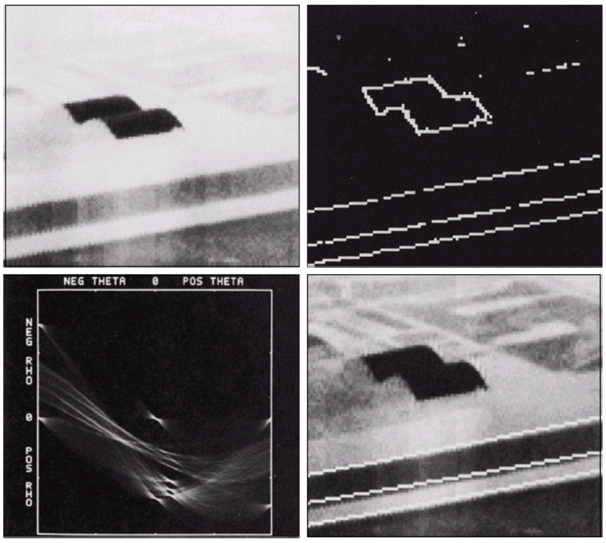
The transform of random points does not give any precise results

HT: ví dụ với tập điểm tạo thành đường thẳng



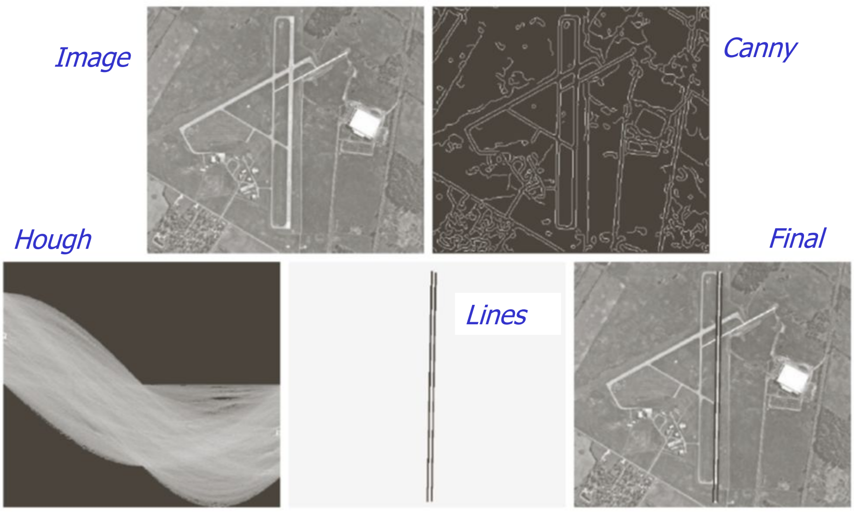


HT: ví dụ



Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Ví dụ



Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall 3ed, 2008.

Hough transform

- Good
  - Ít ảnh hưởng bởi các điểm không nằm trên đường : do các điểm bỏ phiếu riêng
  - Khá hiệu quả (nhanh hơn việc thử tất cả tham số)
  - Đưa ra nhiều đường phù hợp (multiple good fits)
- Bad
  - Đôi khi nhạy cảm với nhiễu (vd: các điểm nhiễu rời rạc tạo thành đường thẳng)
  - Kích thước bin quyết định mức độ cân bằng giữa khả năng chịu nhiễu, độ chính xác, và tốc độ/bộ nhớ
  - Không phù hợp cho các đường có nhiều tham số
    - Kích thước lưới chia tăng theo hàm mũ
- Một số ứng dụng
  - Line fitting (also circles, ellipses, etc.)
  - Object instance recognition (parameters are affine transform)
  - Object category recognition(parameters are position/scale)

Ransac

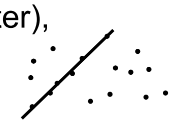
CS231: Ransac, Juan Carlos Niebles and Ranjay Krishna, Stanford Vision and Learning Lab

Ransac

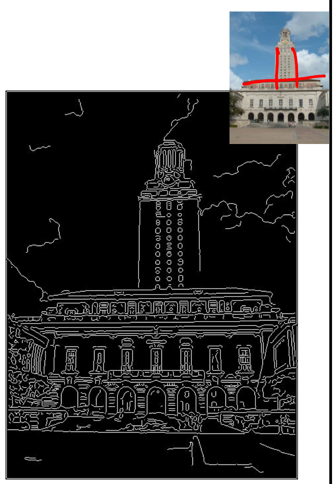
- Phương pháp khớp mô hình:
  - Phương pháp học để ước lượng tham số cho 1 mô hình từ việc lấy mẫu ngẫu nhiên từ dữ liệu quan sát
  - Sử dụng cho:
    - Phát hiện đường thẳng (tròn, elipse, etc)
    - Bài toán về so khớp: tìm phép biến đổi giữa 2 tập dữ liệu
    - …

Difficulty of Line Fitting

- Extra edge points (clutter), multiple models:
  - Which points go with which line, if any?



- Only some parts of each line detected, and some parts are missing:
  - How to find a line that bridges missing evidence?
- Noise in measured edge points, orientations:
  - How to detect true underlying parameters?



Slide credit: Kristen Grauman

Voting

- It's not feasible to check all combinations of features by fitting a model to each possible subset.
- Voting is a general technique where we let the features vote for all models that are compatible with it.
  - Cycle through features, cast votes for model parameters.
  - Look for model parameters that receive a lot of votes.
- Noise & clutter features will cast votes too, *but* typically their votes should be inconsistent with the majority of "good" features.
- Ok if some features not observed, as model can span multiple fragments.

Slide credit: Kristen Grauman

RANSAC [Fischler & Bolles 1981]

- RANdom SAmple Consensus
- Approach:
  - we want to avoid the impact of outliers, so let's look for "inliers", and use only those
  - Intuition: if an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points.

Slide credit: Kristen Grauman

RANSAC [Fischler & Bolles 1981]

RANSAC loop:

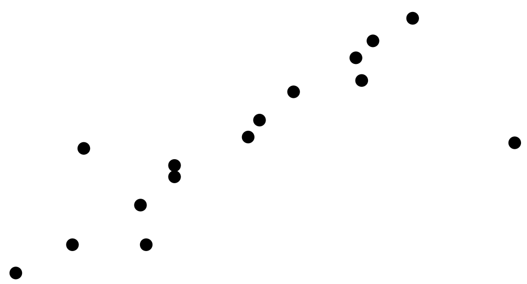
1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

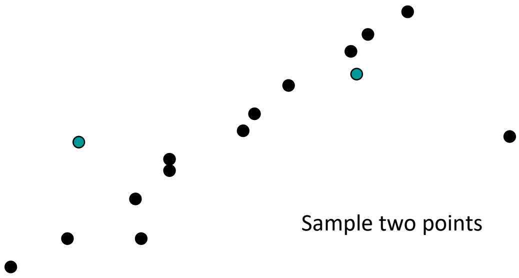
- Task: Estimate the best line
  - *How many points do we need to estimate the line?*



Slide credit: Kristen Grauman

RANSAC Line Fitting Example

- Task: Estimate the best line

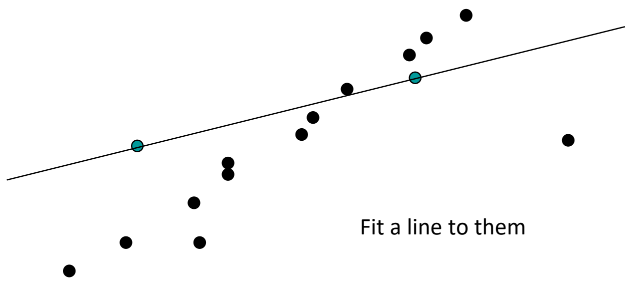


Sample two points

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

- Task: Estimate the best line

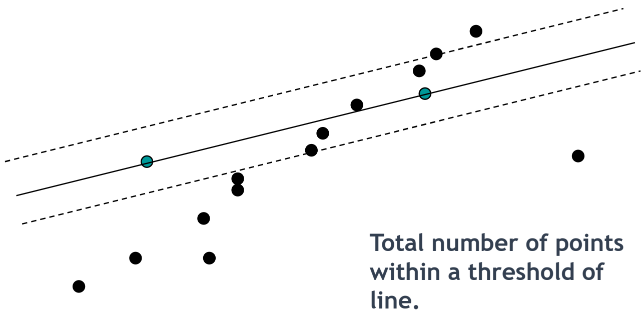


Fit a line to them

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

- Task: Estimate the best line

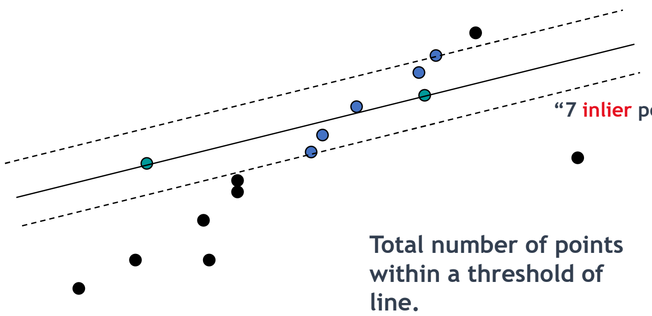


Total number of points within a threshold of line.

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

- Task: Estimate the best line

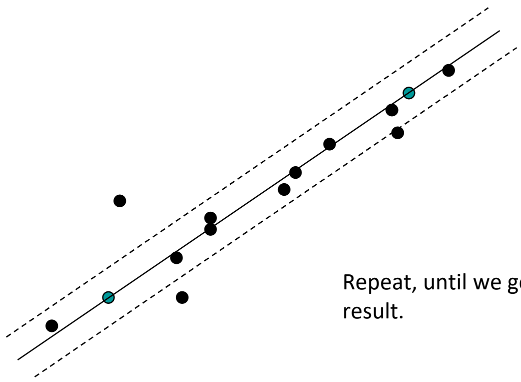


Total number of points within a threshold of line.

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

- Task: Estimate the best line

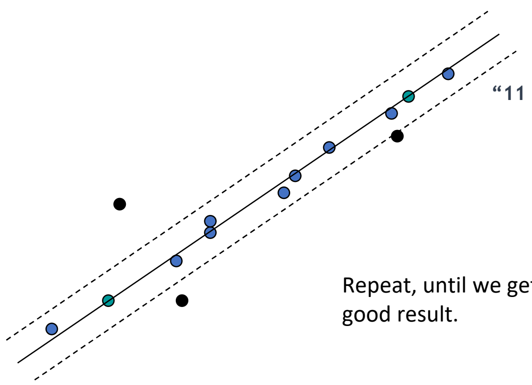


Repeat, until we get a good result.

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

- Task: Estimate the best line



“11 **inlier** points”

Repeat, until we get a good result.

Slide credit: Kristen Grauman

Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:
  n — the smallest number of points required
  k — the number of iterations required
  t — the threshold used to identify a point that fits well
  d — the number of nearby points required
    to assert a model fits well
Until k iterations have occurred
  Draw a sample of n points from the data
    uniformly and at random
  Fit to that set of n points
  For each data point outside the sample
    Test the distance from the point to the line
      against t; if the distance from the point to the line
      is less than t, the point is close
  end
  If there are d or more points close to the line
    then there is a good fit. Refit the line using all
    these points.
end
Use the best fit from this collection, using the
  fitting error as a criterion

RANSAC: How many samples?

- How many samples are needed?
  - Suppose *w* is fraction of inliers (points from line)
  - *n* points needed to define hypothesis (2 for lines)
  - *k* samples chosen
  - p: desired probability that we get a good sample
- Prob. that a single sample of *n* points is correct: w^n
- Prob. that all *k* samples fail is: $(1-w^n)^k$

$\Rightarrow$ Choose ***k*** high enough to keep this below desired failure rate (1-p).

Slide credit: David Lowe

RANSAC: Computed k (p=0.99)

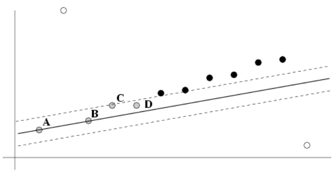
Sample size n	Proportion of outliers 5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

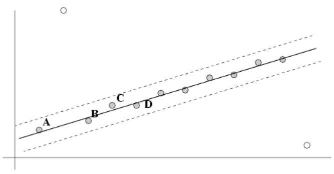
p: desired probability that we get a good sample

Slide credit: David Lowe

After RANSAC

- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers.
- Improve this initial estimate with estimation over all inliers (e.g. with standard least-squares minimization).
- But this may change inliers, so alternate fitting with re-classification as inlier/outlier.





Slide credit: David Lowe

RANSAC: Pros and Cons

- **Pros**:
  - Robust to outliers
  - General method suited for a wide range of model fitting problems (larger number of parameters than Hough transform)
  - Easy to implement and easy to calculate its failure rate
- **Cons**:
  - Only handles a moderate percentage of outliers without cost blowing up
  - Many real problems have high rate of outliers (but sometimes selective choice of random subsets can help)
    - A voting strategy, the Hough transform can handle high percentage of outliers
  - Not good for getting multiple fits

- Common applications
  - Computing a homography (e.g., image stitching)
  - Estimating fundamental matrix (relating two views)

References

- Lecture 3: CS231 - Juan Carlos Niebles and Ranjay Krishna, Stanford Vision and Learning Lab
- Vision par Ordinateur, Alain Boucher, IFI

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