

# Edge Detection

## (Phát hiện biên trong ảnh)

# Content

---

1. Giới thiệu

2. Các phương pháp phát hiện biên:

2.1. Phương pháp đạo hàm sử dụng bộ lọc

- Bộ lọc Sobel
- Bộ lọc Prewitt
- Bộ lọc **La bân**

2.2. Phát hiện biên sử dụng Canny Detector

2.3. Phát hiện biên Derivative of Gaussian

2.4. Một số nhận xét

# 1. Giới thiệu

---

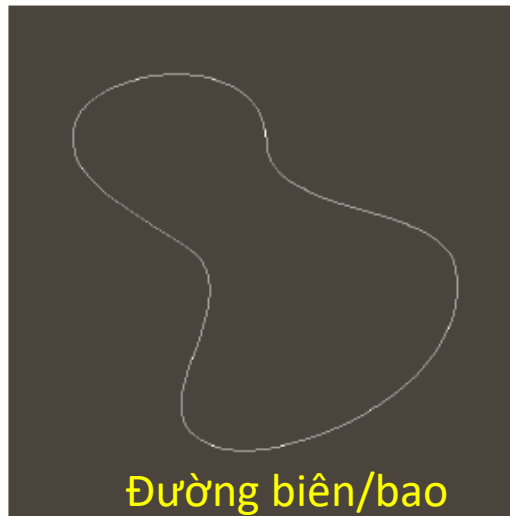
- **Điểm biên**: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu).
  - Ví dụ: Trong ảnh nhị phân, điểm đen được gọi là điểm biên nếu lân cận của nó có ít nhất một điểm trắng.
- **Đường biên/đường bao (boundary)**: Là tập hợp các điểm biên liên tiếp.



# 1. Giới thiệu

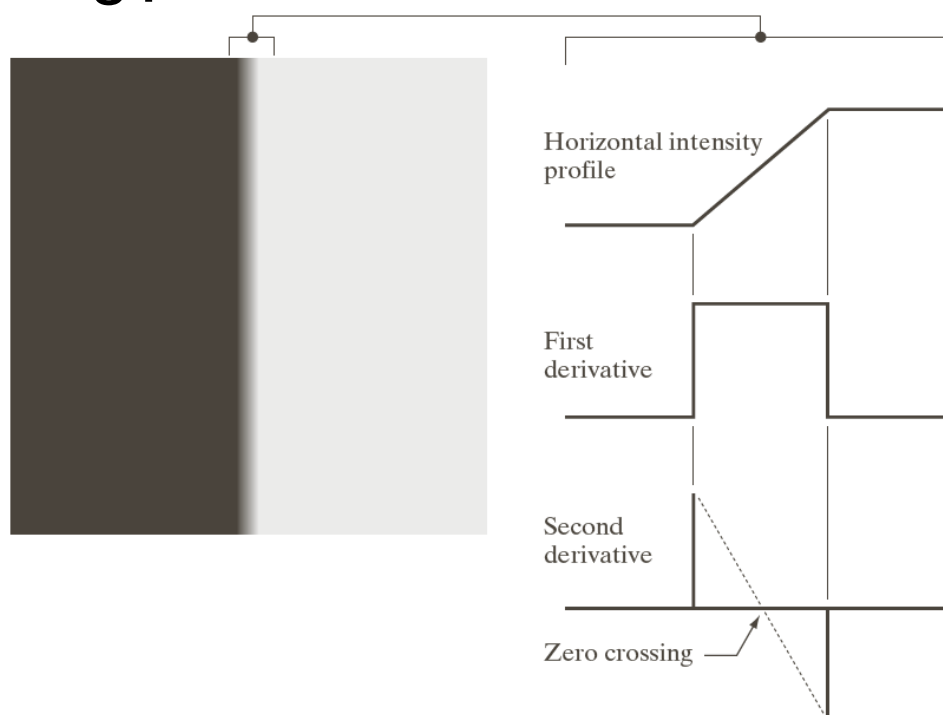
---

- Ý nghĩa của bài toán phát hiện biên:
  - Đường biên là một loại **đặc trưng cục bộ** tiêu biểu trong **phân tích, nhận dạng ảnh**.
  - Đường biên giúp phân đoạn các vùng trong ảnh (xám, màu).
- <> Sử dụng **các vùng ảnh** để tìm đường biên.



## 2. Phương pháp phát hiện biên

- Theo toán học, điểm ảnh có sự biến đổi mức xám  $u(x)$  một cách đột ngột



- Nếu lấy đạo hàm bậc nhất của  $u(x) \rightarrow$  Gradient
- Nếu lấy đạo hàm bậc hai của  $u(x) \rightarrow$  Laplace

## 2.1. Phát hiện biên gradient

---

- Gradient là một vector có các thành phần biểu thị tốc độ thay đổi mức xám của điểm ảnh (theo hai hướng  $x, y$  đối với ảnh 2 chiều) :

$$\Delta f = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} \quad \begin{aligned} \frac{\partial f(x, y)}{\partial x} &= f'_x \approx \frac{f(x+dx, y) - f(x, y)}{dx} \\ \frac{\partial f(x, y)}{\partial y} &= f'_y \approx \frac{f(x, y+dy) - f(x, y)}{dy} \end{aligned}$$

- Trong đó  $dx, dy$  là khoảng cách giữa 2 điểm kế cận theo hướng  $x, y$  tương ứng (thực tế chọn  $dx=dy=1$ )

## 2.1. Phát hiện biên gradient

---

- Để đơn giản tính toán, tính toán bằng  $h_x, h_y$ 
  - Một số cặp mặt nạ tiêu biểu như Prewitt, Sobel, Robert đẳng hướng (Isometric)
  - $g_x, g_y$ : Gradient theo hai hướng x,y

$$\text{Edge normal: } \nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad G(m,n) = \sqrt{G_x^2(m,n) + G_y^2(m,n)}$$

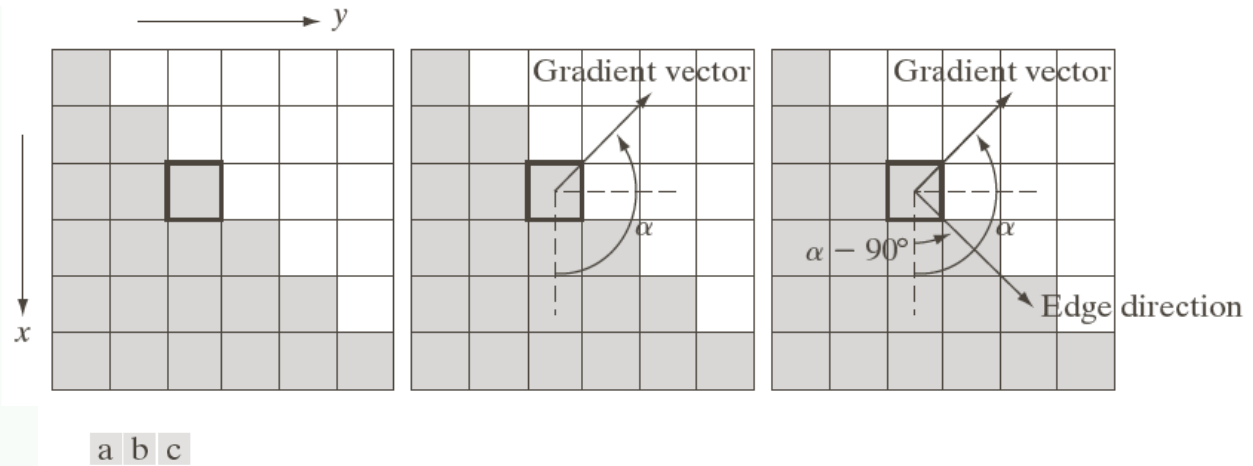
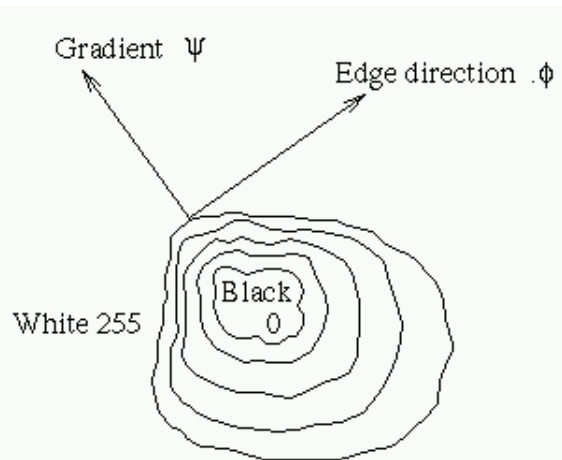
Edge unit normal:  $\nabla f / \text{mag}(\nabla f)$

In practice, sometimes the magnitude is approximated by

$$\text{mag}(\nabla f) = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \quad \text{or} \quad \text{mag}(\nabla f) = \max \left( \left| \frac{\partial f}{\partial x} \right|, \left| \frac{\partial f}{\partial y} \right| \right)$$

## 2.1. Phát hiện biên gradient

- Các bước thực hiện



**FIGURE 10.12** Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.



## 2.1.1. Kỹ thuật Prewitt

---

- Sử dụng 2 mặt nạ và xấp xỉ đạo hàm theo 2 hướng x và y là:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$
$$H_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

- Bước 1: Tính  $I \otimes H_x$  và  $I \otimes H_y$
- Bước 2: Tính  $I \otimes H_x + I \otimes H_y$

## 2.1.1. Kỹ thuật Prewitt

---

- Ví dụ:

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$I \otimes H_x = \begin{pmatrix} 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -5 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

$$I \otimes H_y = \begin{pmatrix} 15 & 15 & 10 & 5 & * & * \\ 0 & 0 & 0 & 0 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

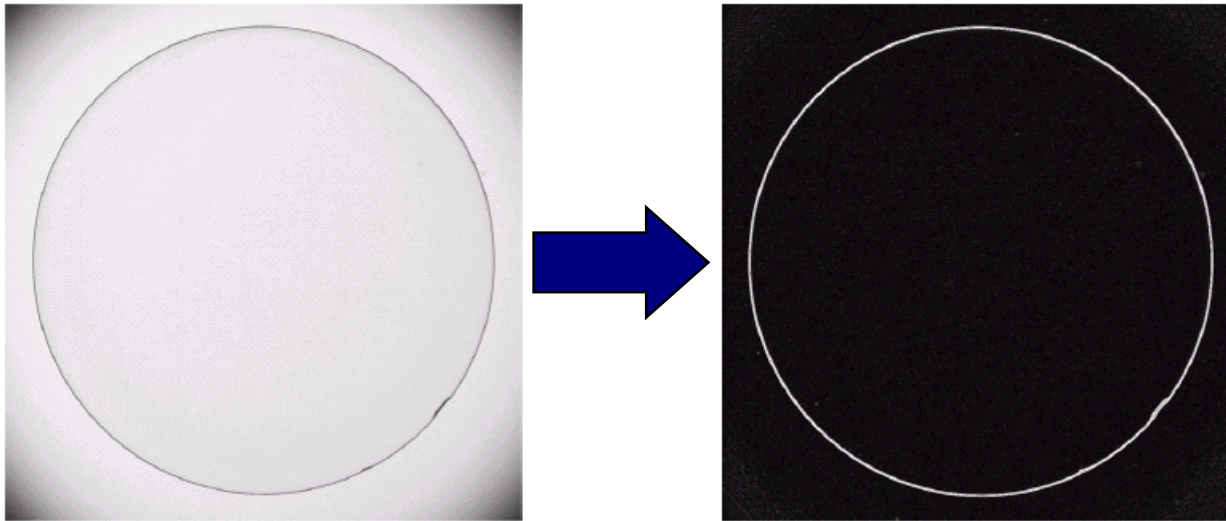
## 2.1.1. Kỹ thuật Prewitt

---

$$I \otimes H_x + I \otimes H_y = \begin{pmatrix} 15 & 15 & 0 & -5 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ -15 & -15 & -20 & -15 & * & * \\ -15 & -15 & -15 & -10 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

## 2.1.2. Bộ lọc Sobel

---



- Được sử dụng để tách cạnh
- Tổng tất cả các hệ số trong mặt nạ bằng 0. Điều này nhằm làm cho đáp ứng tại những vùng cấp xám không thay đổi có giá trị bằng 0.

## 2.1.2. Bộ lọc Sobel

---

- Để phát hiện biên theo đường chéo ta sử dụng 2 mặt nạ:

$$H_1 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}; \quad H_2 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Sử dụng bộ lọc Sobel:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad H_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

## 2.1.3. Bộ lọc La bàn

---

- Kỹ thuật sử dụng 8 mặt nạ nhân chập theo 8 hướng  $0^0$ ,  $45^0$ ,  $90^0$ ,  $135^0$ ,  $180^0$ ,  $225^0$ ,  $270^0$ ,  $315^0$
- Các bước tính toán thuật toán La bàn
  - Bước 1: Tính  $I \otimes H_i$  ;  $i = 1,8$
  - Bước 2: Tính  $\sum_{i=1}^8 I \otimes H_i$

## 2.1.3. Bộ lọc La bàn

---

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

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

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

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

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

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

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

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

## 2.1.3. Bộ lọc La bàn



**FIGURE 10.16**

(a) Original image of size  $834 \times 1114$  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,  $|g_x| + |g_y|$ .



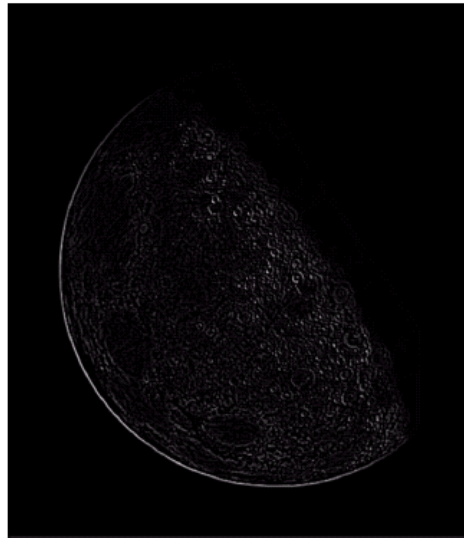
## 2.2. Phương pháp Laplacian

---



Ảnh gốc

-



Ảnh lọc Laplacian

=



Ảnh làm sắc cạnh

Các chi tiết cạnh của ảnh kết quả tốt hơn so với ảnh gốc

## 2.2. Phương pháp Laplacian

---

- Các phương pháp đánh giá gradient ở trên làm việc khá tốt khi mà độ sáng thay đổi rõ nét.
- Khi mức xám thay đổi chậm, miền chuyển tiếp trải rộng, phương pháp cho hiệu quả hơn đó là phương pháp sử dụng đạo hàm bậc hai Laplace
- Toán tử Laplace được xây dựng trên cơ sở đạo hàm bậc 2 của hàm biến đổi mức xám.

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

## 2.2. Phương pháp Laplacian

---

- $\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x} \left( \frac{\partial f}{\partial x} \right) \approx \frac{\partial}{\partial x} (f(x+1, y) - f(x, y))$   
 $\approx [f(x+1, y) - f(x, y)]$   
 $- [f(x, y) - f(x-1, y)]$   
 $\approx f(x+1, y) - 2f(x, y) + f(x-1, y)$
- Tương tự ta có
- $\frac{\partial^2 f}{\partial y^2} \approx f(x, y+1) - 2f(x, y) + f(x, y-1)$
- $\Delta^2 f \approx f(x+1, y) + f(x-1, y) - 4f(x, y) + f(x, y+1) + f(x, y-1)$

## 2.2. Phương pháp Laplacian

---

- Dưới đây là ba kiểu mặt nạ thường dùng:

$$H_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

$$H_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

## 2.2. Phương pháp Laplacian

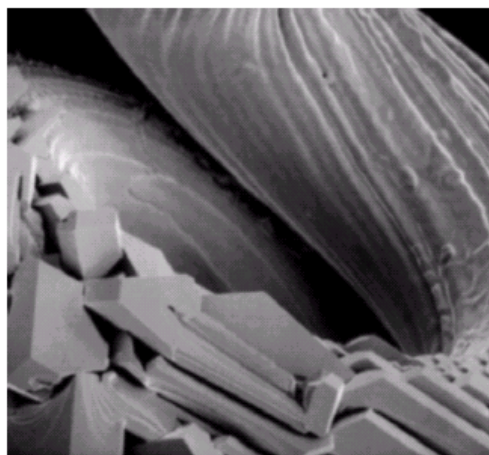
- Có **nhiều phiên bản** Laplacian khác nhau:

0	1	0
1	-4	1
0	1	0

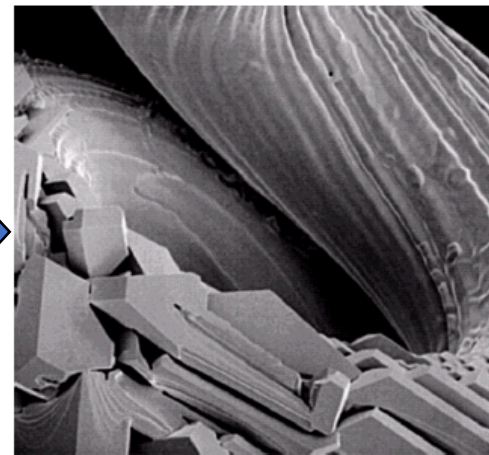
Bản gốc

1	1	1
1	-8	1
1	1	1

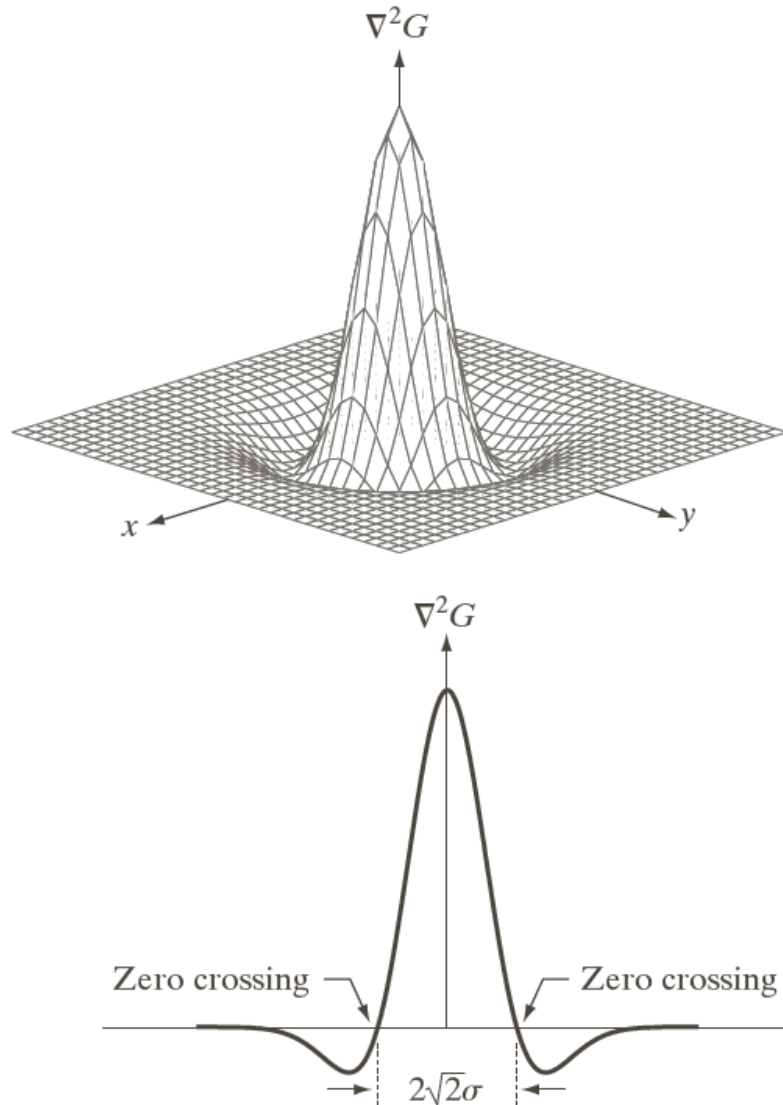
Biến thể



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



## 2.2. Phương pháp Laplacian



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)  $5 \times 5$  mask approximation to the shape in (a). The negative of this mask would be used in practice.

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

## 2.2. Phương pháp Laplacian

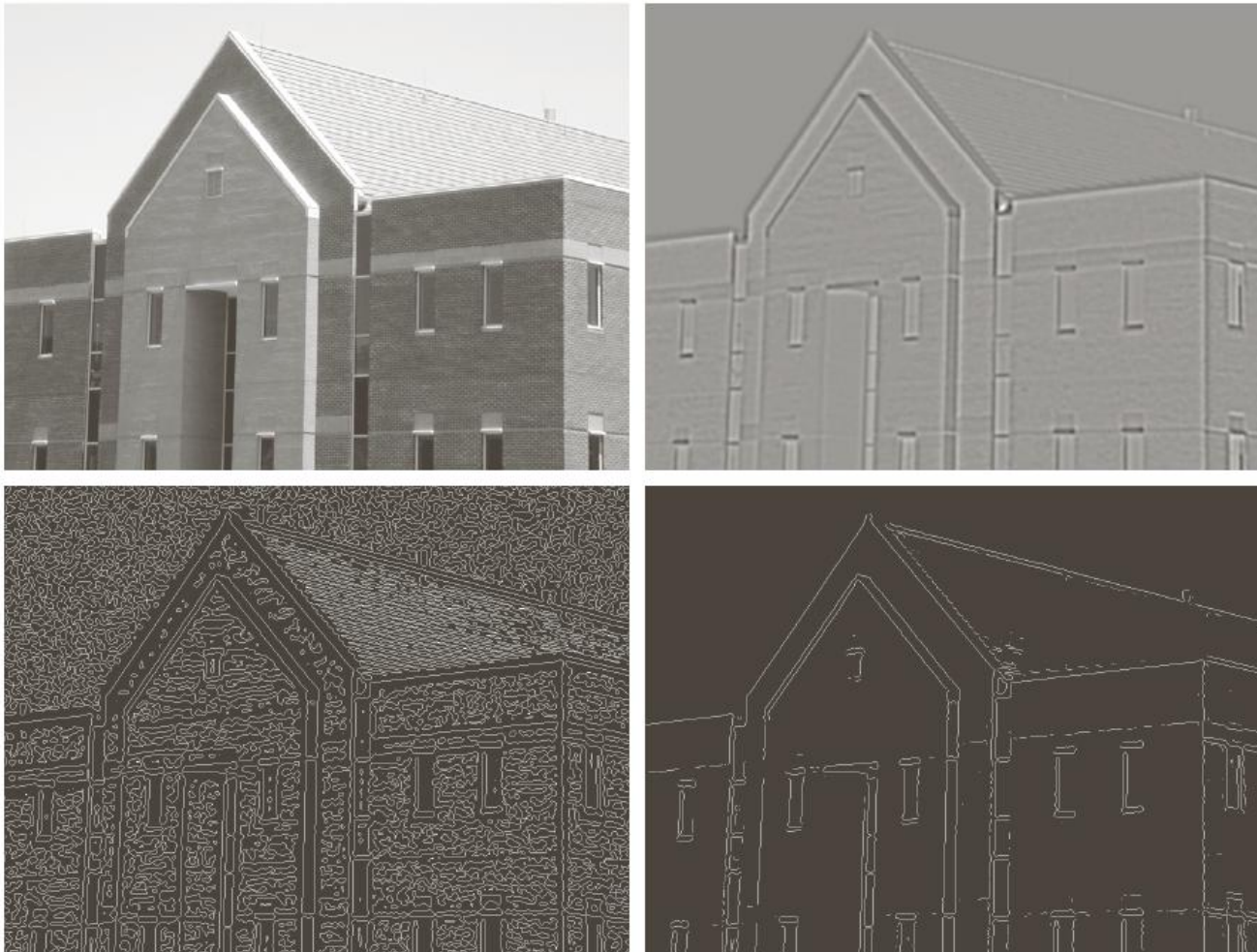
---

1. Filter the input image with an  $n \times n$  Gaussian lowpass filter.  $N$  is the smallest odd integer greater than or equal to 6
2. Compute the Laplacian of the image resulting from step 1

$$g(x, y) = \nabla^2 [G(x, y) \quad f(x, y)]$$

3. Find the zero crossing of the image from step 2

## 2.2. Phương pháp Laplacian



a	b
c	d

**FIGURE 10.22**

(a) Original image of size  $834 \times 1114$  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.



## 2.3. Kỹ thuật Canny

---

- Đây là thuật toán tương đối tốt, đưa ra đường biên mạnh, phát hiện chính xác điểm biên với điểm nhiễu
- Các bước của thuật toán
  - **Bước 1:** Làm trơn ảnh. Tính tích chập  $G = I \otimes H$ ,

với

$$H = \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

- **Bước 2:** Tính Gradient của ảnh bằng mặt nạ **Prewitt** theo hai hướng x, y. Gọi là  $G_x$ ,  $G_y$ .

## 2.3. Kỹ thuật Canny

---

- Bước 3: Tính Gradient theo 8 hướng tương ứng với 8 lân cận của 1 điểm ảnh
- Bước 4: Loại bỏ những điểm không phải cực nhằm xóa bỏ những điểm không thuộc biên
- Bước 5: Phân ngưỡng. Thực hiện lấy Gradient lần cuối.

# The Canny Edge Detector: Algorithm (1)

---

Let  $f(x, y)$  denote the input image and  $G(x, y)$  denote the Gaussian function:

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

We form a smoothed image,  $f_s(x, y)$  by convolving  $G$  and  $f$ :

$$f_s(x, y) = G(x, y) \star f(x, y)$$

## 2.3. Kỹ thuật Canny

---



# Nhân xét kết quả phát hiện biên

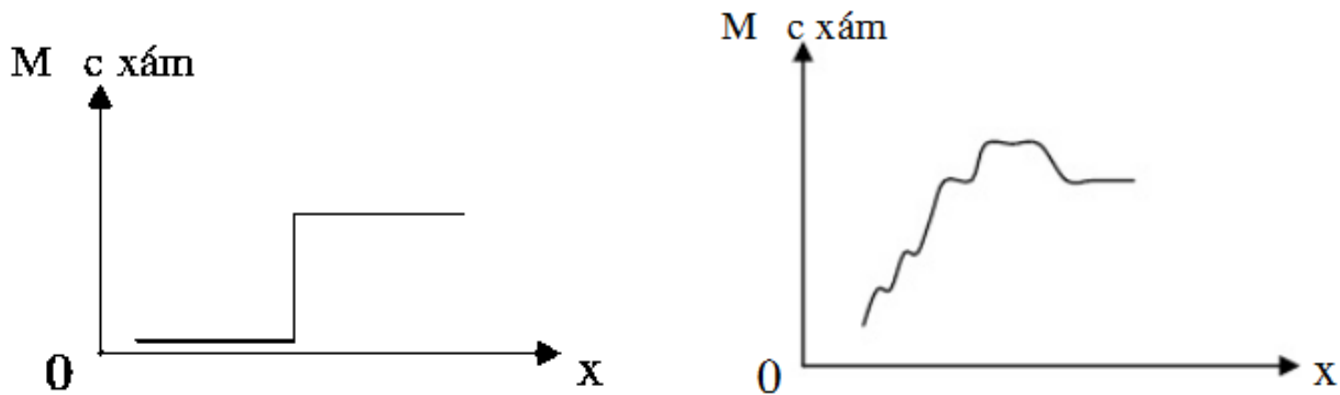
---

- Các kỹ thuật dò biên đã nêu ở trên (Gradient, Laplacian, Canny):
  - Rất nhạy cảm với nhiễu.
  - Thực tế chỉ làm nổi biên
  - Khó điều chỉnh ảnh biên
  - Khó sử dụng cho việc nhận dạng đối tượng
- ➔ Các kỹ thuật này chưa đạt được sự hoàn thiện mong muốn
- Phương pháp phát hiện biên = học biên

# Lý do:

---

- Lý tưởng (ảnh đen trắng) đồ thị sự biến thiên mức xám rõ ràng
- Thực tế với các ảnh đa mức xám  $\rightarrow$  xác định ngưỡng tách biên rất khó



# Các PP phát hiện biên mới



image

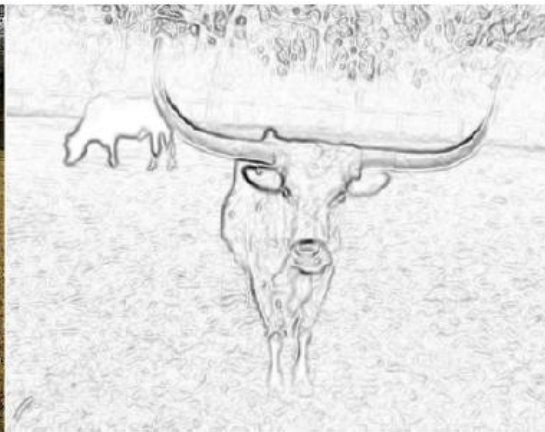
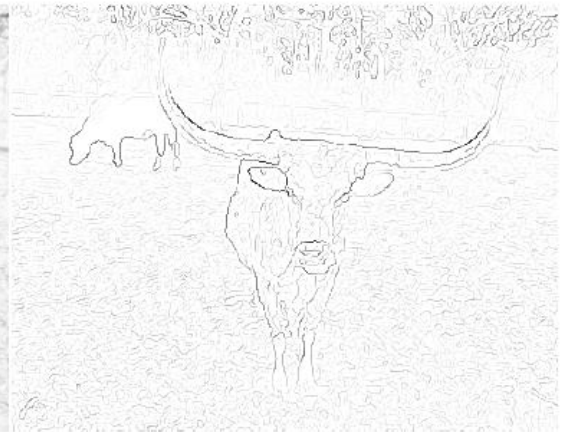
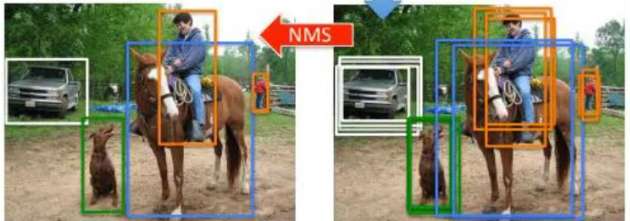
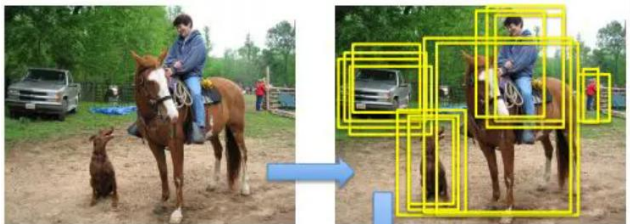


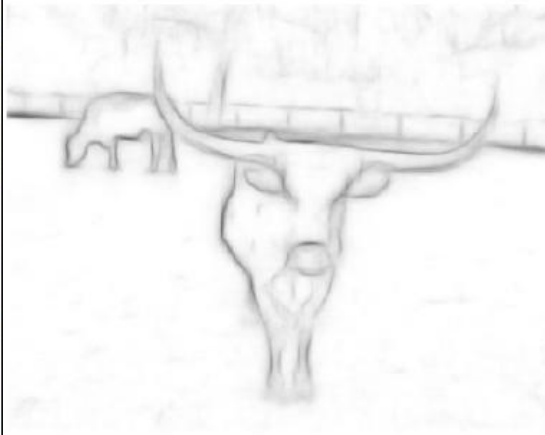
image gradients



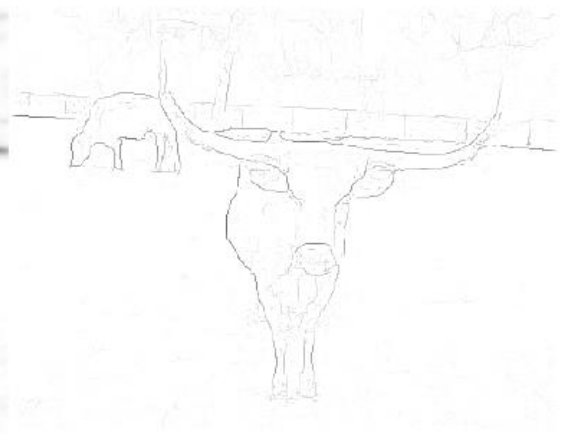
gradients + NMS



NMS



"edginess" score



score + NMS

**Non-Maximum Suppression (NMS)**



# Các PP phát hiện biên mới



image



image gradients



gradients + NMS



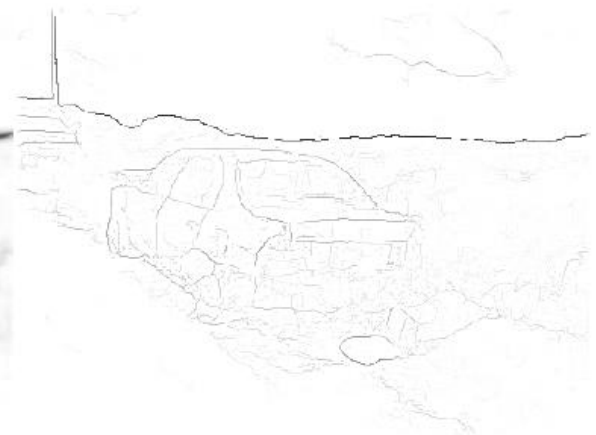
image gradient



"edginess" score



"edginess" score



score + NMS



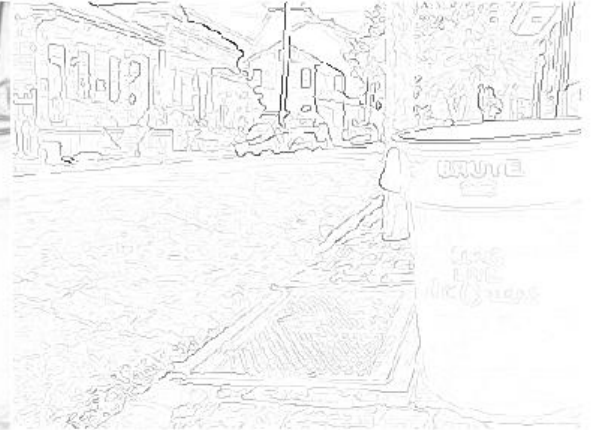
# Các PP phát hiện biên mới



image



image gradients



gradients + NMS



image gradient



"edgeness" score



"edgeness" score

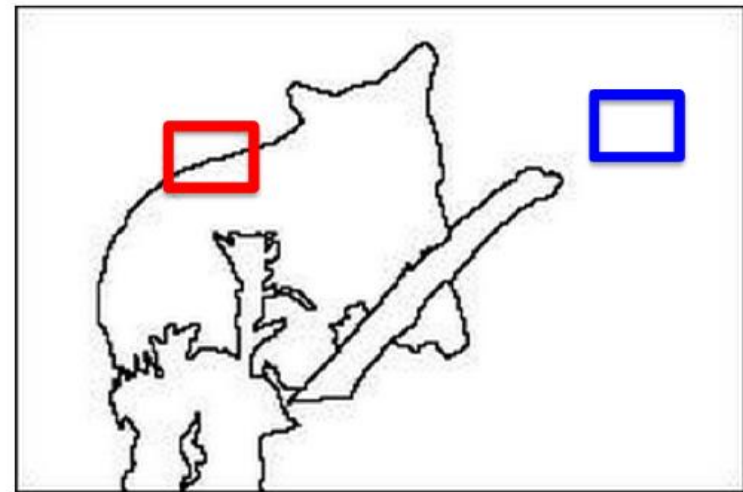


score + NMS

# Edgeness = learning edge

---

- We extract lots of image patches
- These are our training data



Ảnh gốc

Ảnh biên chuẩn bị bằng tay



→ edge



→ no edge



our training data

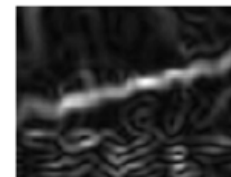
# Edginess = learning edge

---

- We extract lots of image patches
- These are our training data
- We convert each image patch  $\mathbf{P}$  (a matrix) into a vector  $\mathbf{x}$
- Well... This works better: Extract **image features** for each patch
- Image features are mappings from images (or patches) to other (vector) meaningful representations. More on this in the next class!



compute gradients  
 $\rightarrow$



matrix  $\mathbf{G}$

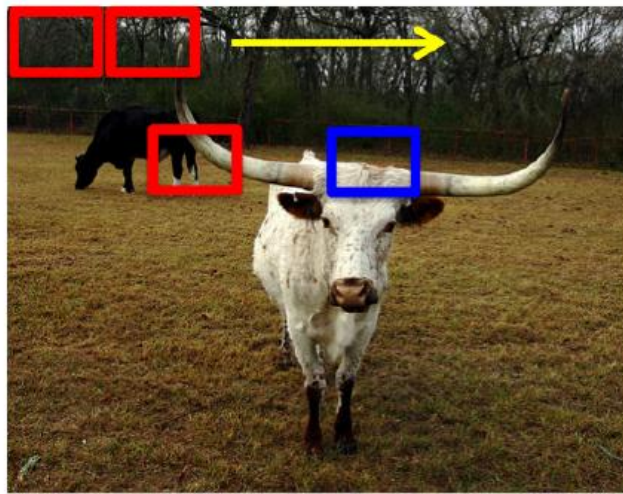
$$\mathbf{x} = \mathbf{G}(:,)$$

compute color  
 $\rightarrow$

# Edgeness = learning edge

---

- We extract all image patches
- Extract features and use our trained classifier
- Place the predicted value (score) in the output matrix

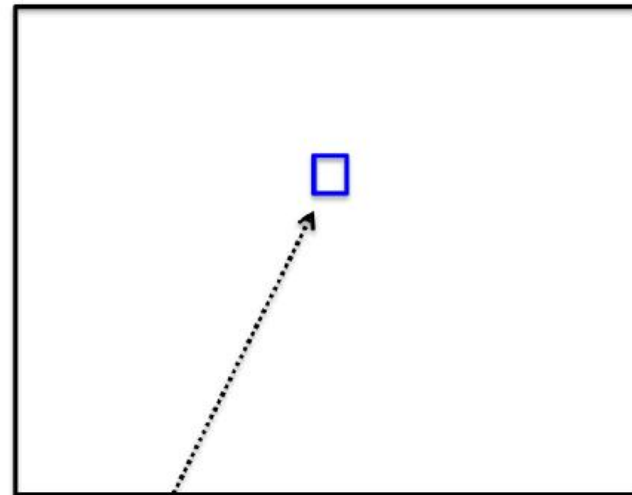


image



classify  
→

e.g.  $\text{score} = \mathbf{w}^T \mathbf{x} + b$

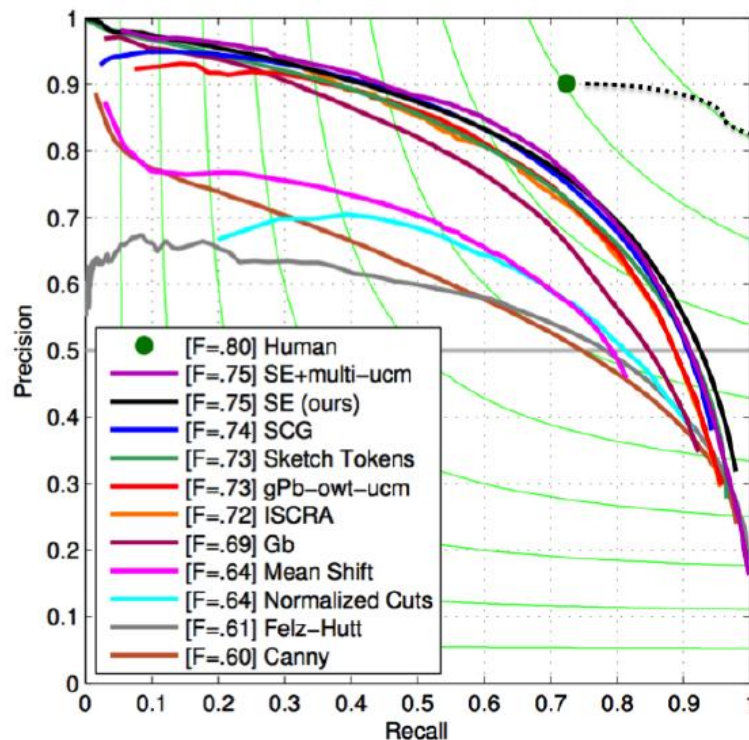


prediction



# Đánh giá các phương pháp

- **Recall:** How many of all **annotated** edges we got correct (best is 1)
- **Precision** How many of all **output** edges we got correct (best is 1)

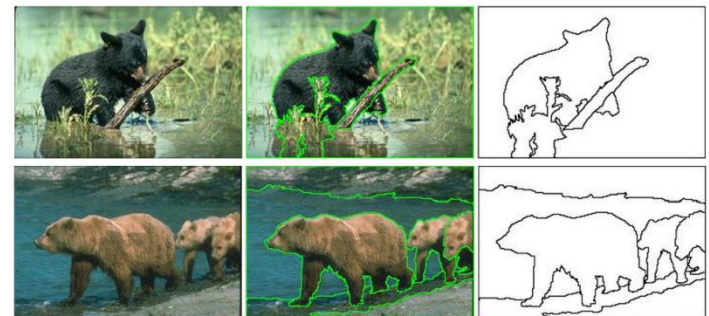


human agreement

## Evaluation Dataset

### The Berkeley Segmentation Dataset and Benchmark

by D. Martin and C. Fowlkes and D. Tal and J. Malik



---

# Lý thuyết tập hợp

- $A$  là một tập hợp thuộc không gian  $Z^2$ . Một phần tử  $a=(a_1, a_2)$  thuộc  $A$  được ký hiệu

$$a \in A$$

- Tương tự, nếu phần tử  $a$  không thuộc  $A$ , ta ký hiệu

$$a \notin A$$

- Nếu  $A$  không chứa một phần tử nào,  $A$  là tập rỗng

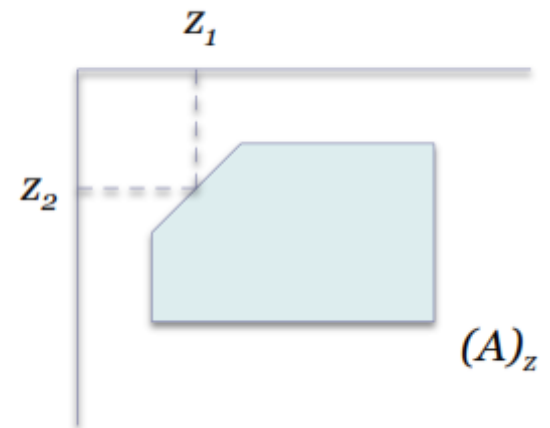
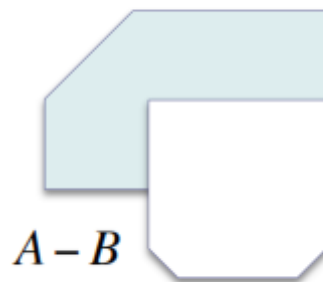
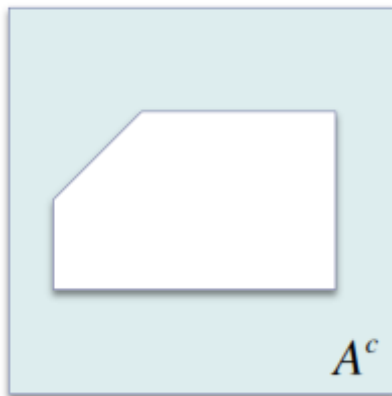
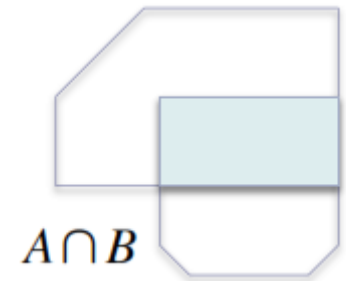
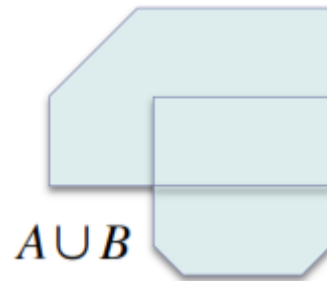
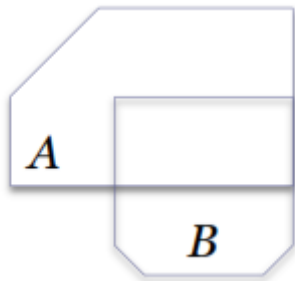
$$A = \emptyset$$

---

# Lý thuyết tập hợp

$A$ là tập con của $B$	$A \subseteq B$
Hợp của hai tập hợp $A$ và $B$	$C = A \cup B$
Giao của hai tập hợp $A$ và $B$	$C = A \cap B$
Phần bù của tập hợp $A$	$A^c = \{w \mid w \notin A\}$
Hiệu của hai tập hợp $A$ và $B$	$A - B = \{w \mid w \in A, w \notin B\}$
Phản chiếu của tập hợp $A$	$\hat{A} = \{w \mid w = -a, \forall a \in A\}$
Dịch tập hợp $A$ với $z=(z_1, z_2)$	$(A)_z = \{c \mid c = a + z, \forall a \in A\}$

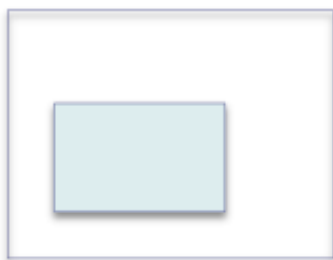
# Lý thuyết tập hợp





---

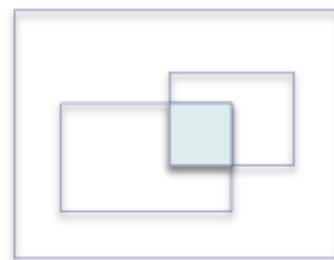
# Toán tử Logic



$A$



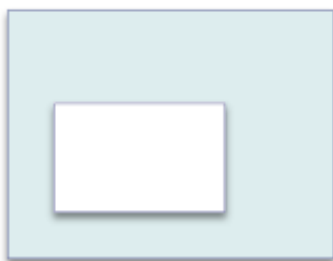
$B$



$A \text{ AND } B$



$A \text{ XOR } B$



$\text{NOT } A$



$A \text{ OR } B$



$[\text{NOT } A] \text{ AND } B$

Original  
image

dilated

eroded



*dilated*

*eroded*



Open



Close

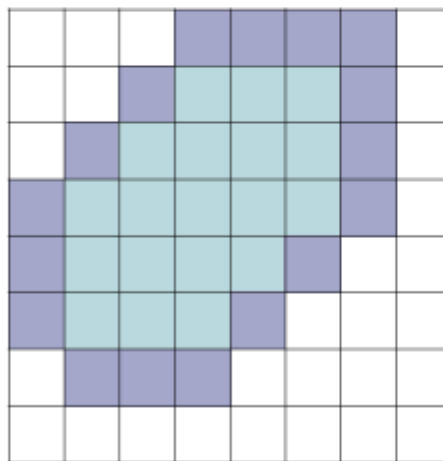
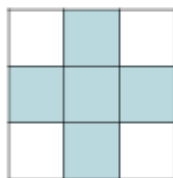
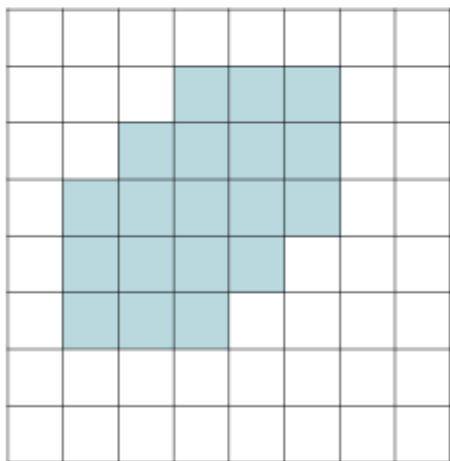
---

## Toán tử hình thái

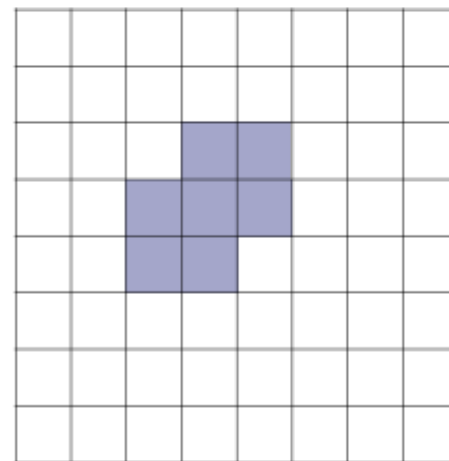
- Toán tử hình thái là công cụ toán học để xử lý hình dạng trong ảnh. Toán tử hình thái sử dụng hướng tiếp cận lý thuyết tập hợp.
- Toán tử hình thái bao gồm phép co (erosion), phép giãn (dilation), phép mở (opening), phép đóng (closing).
- Toán tử hình thái được ứng dụng trong tách biên ảnh, lấp đầy vùng ảnh, tạo kết nối giữa các vùng ảnh hoặc làm xương ảnh.

# Phép dẫn và phép co

Phép dẫn	Phép co
$A \oplus B = \{z \mid (\widehat{B})_z \cap A \neq \emptyset\}$	$A \ominus B = \{z \mid (B)_z \subseteq A\}$



$A \oplus B$



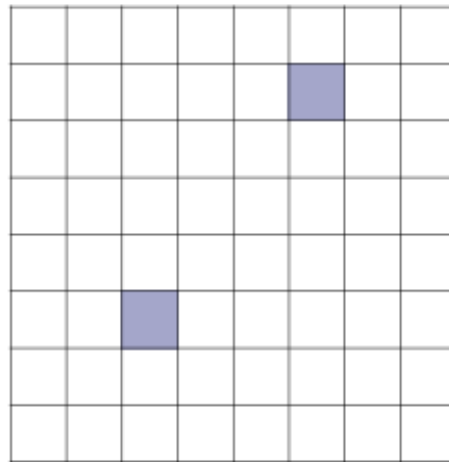
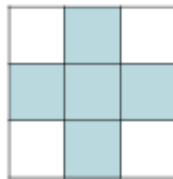
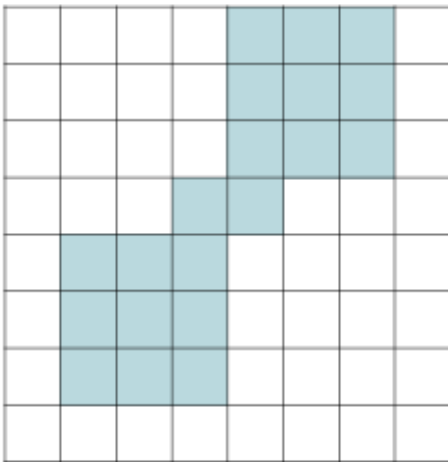
$A \ominus B$

---

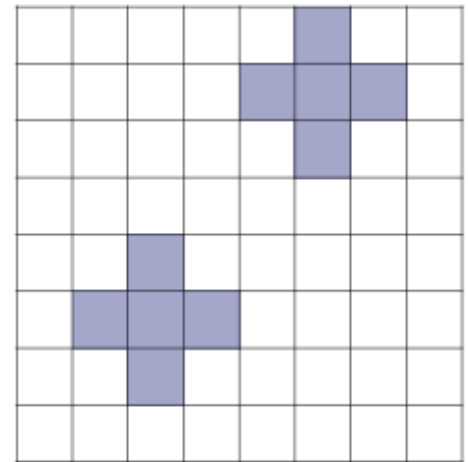
# Phép mở

Phép mở xóa bỏ những đoạn mảnh, loại bỏ nhiễu nhưng làm tăng số đoạn đứt gãy

$$A \circ B = (A \ominus B) \oplus B$$



$A \ominus B$



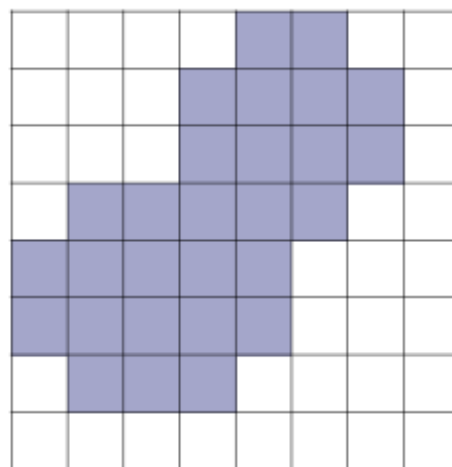
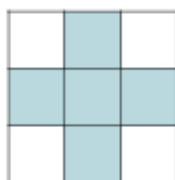
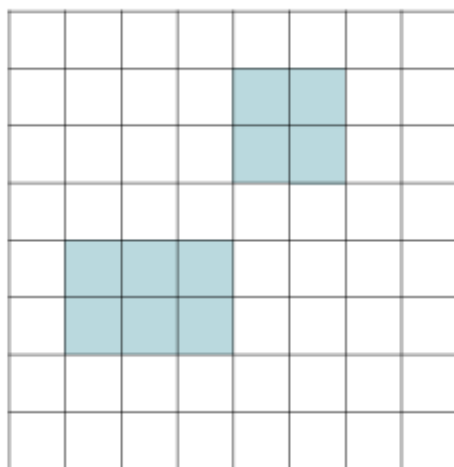
$A \circ B$

---

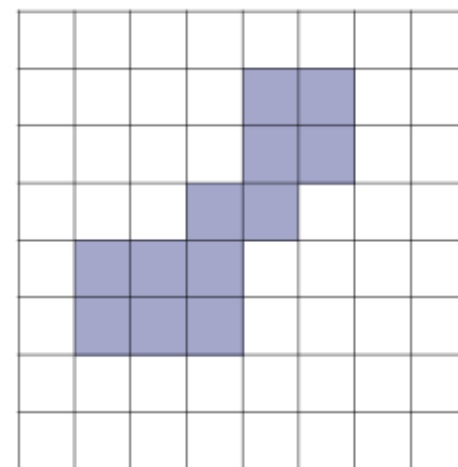
# Phép đóng

Phép đóng có thể làm trơn biên ảnh và kết nối các vùng của cùng một đối tượng.

$$A \bullet B = (A \oplus B) \ominus B$$



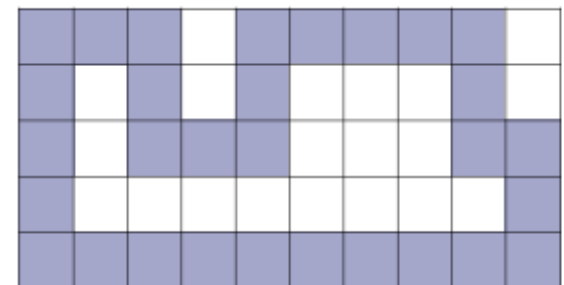
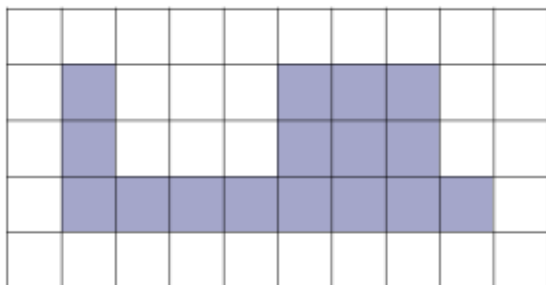
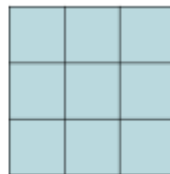
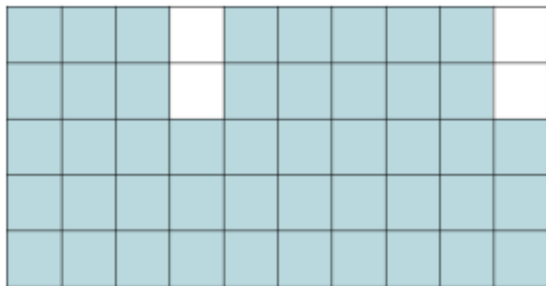
$A \oplus B$



$A \bullet B$

## Tách biên ảnh

$$\beta(A) = A - (A \ominus B)$$



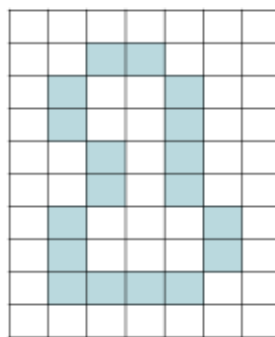
# Lấp đầy vùng ảnh

$$X_k = (X_{k-1} \oplus B) \cap A^c, \quad k = 1, 2, 3 \dots$$

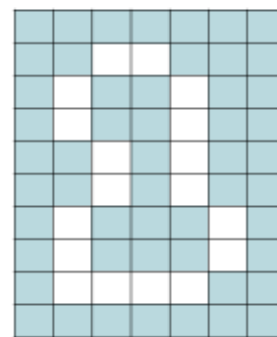
Lặp quá trình trên đến khi  $X_{k-1} = X_k$



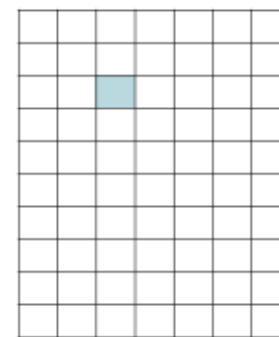
$B$



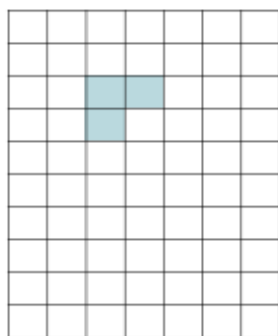
$A$



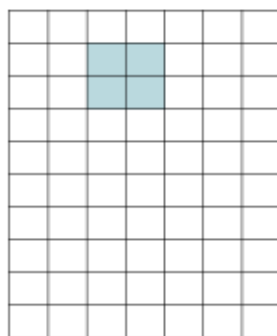
$A^c$



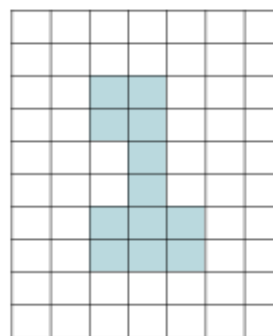
$X_0$



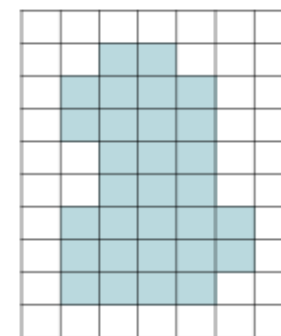
$X_1$



$X_2$



$X_7$

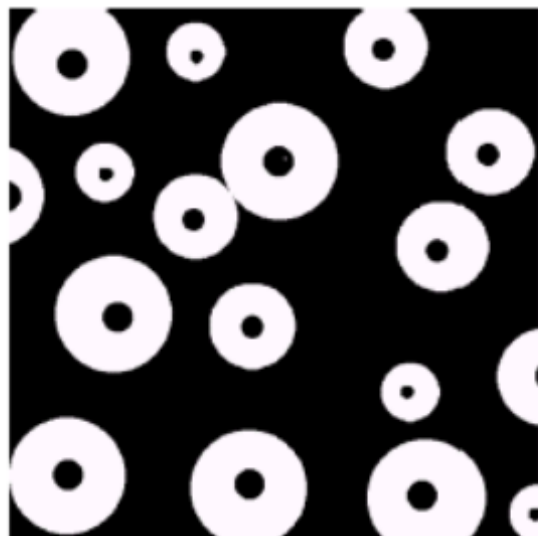


$X_7 \cup A$

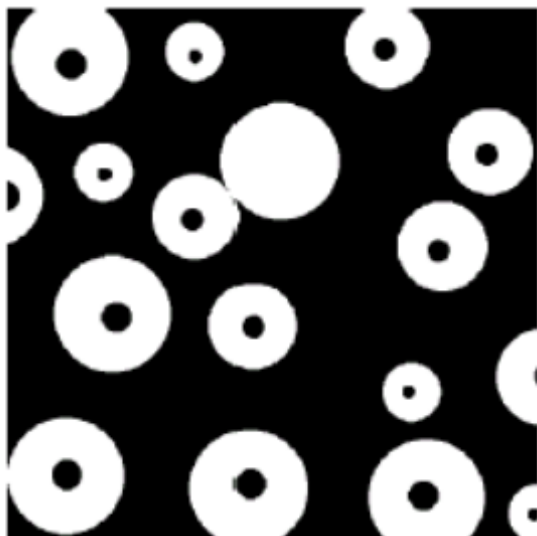


---

## Lấp đầy vùng ảnh



Ảnh gốc



Lấp đầy một vùng ảnh



Lấp đầy vùng ảnh

© 2006 The Authors  
Journal compilation © 2006 Blackwell Publishing Ltd

