

CHƯƠNG 5. PHÁT HIỆN BIÊN (EDGE DETECTION)

NỘI DUNG

- Giới thiệu
- Các loại biên
- Các bước trong quá trình phát hiện biên
- Các phương pháp phát hiện biên
 - Phương pháp sử dụng đạo hàm bậc 1
 - Phương pháp sử dụng đạo hàm bậc 2
 - Phương pháp phát hiện biên tối ưu: Phương pháp Canny

GIỚI THIỆU

- **Biên là gì?**

Biên là các thay đổi đáng kể của cường độ xám trong một bức ảnh.

- **Mục tiêu của phát hiện biên:**

Xác định các vùng có độ tương phản cường độ mạnh hay nói cách khác tạo ra một bản phác thảo (“drawing”) của cảnh từ bức ảnh chứa cảnh đó.

- **Phát hiện biên để làm gì?**

Để trích xuất thông tin từ bức ảnh, ví dụ: vị trí của các đối tượng trong bức ảnh, hình dạng và kích thước của chúng, làm sắc nét ảnh, tăng cường ảnh, phục vụ cho quá trình nhận dạng ảnh.

VÍ DỤ

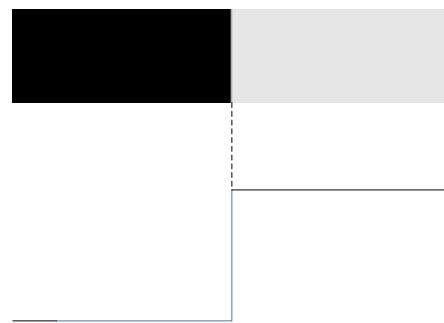


CÁC LOẠI BIÊN

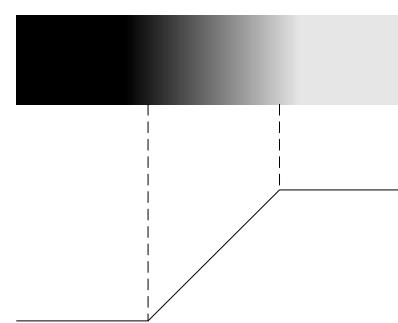
- Tùy thuộc vào sự thay đổi cường độ sáng/ mức xám mà ta có các loại biên khác nhau:
 - Biên kiểu bước nhảy (step edge)
 - Biên thoai thoải (ramp edge)
 - Biên kiểu mái nhà (roof edge)

Step edge và ramp edge, roof edge

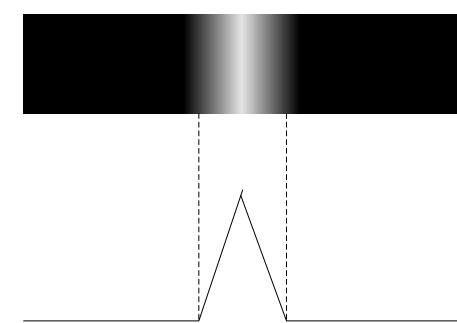
- **Step edge** (ideal edge): Cường độ sáng thay đổi một cách đột ngột.
- **Ramp edge**: Cường độ sáng không thay đổi ngay lập tức mà xảy ra trên một khoảng cách nhất định.
- **Roof edge**: Thường là mô hình đường thẳng chạy qua một vùng với độ rộng của mái nhà tùy thuộc vào độ dày và độ sắc nét của đường thẳng.



A. Step edge



B. Ramp edge



C. Roof edge

BIÊN VÀ ĐẠO HÀM

- Như đã thảo luận ở phần trước, trung bình cục bộ thường làm mịn ảnh.
- Cách tính trung bình giống như tích phân vì vậy để xác định các thay đổi cục bộ, đột ngột của cường độ ta có thể sử dụng đạo hàm.
- Vì lý do này, đạo hàm bậc 1 và đạo hàm bậc 2 là giải pháp hoàn hảo để phát hiện biên.

YÊU CẦU KHI XẤP XỈ ĐẠO HÀM BẬC 1

1. Phải bằng 0 tại các vùng có cường độ sáng không đổi.
2. Phải khác 0 tại điểm bắt đầu của cường độ có thay đổi bước nhảy hoặc thoai thoả (step và ramp).
3. Phải khác 0 tại các điểm dọc theo dốc thoai thoả (ramp).

YÊU CẦU KHI XẤP XỈ ĐẠO HÀM BẬC 2

1. Phải bằng 0 tại các vùng có cường độ sáng không đổi.
2. Phải khác 0 tại điểm bắt đầu và kết thúc của cường độ thay đổi kiểu bước nhảy hoặc thoai thoả.
3. Phải bằng 0 đọc theo các điểm nằm trên đường dốc thoai thoả.

Ảnh hưởng của nhiễu đến đạo hàm

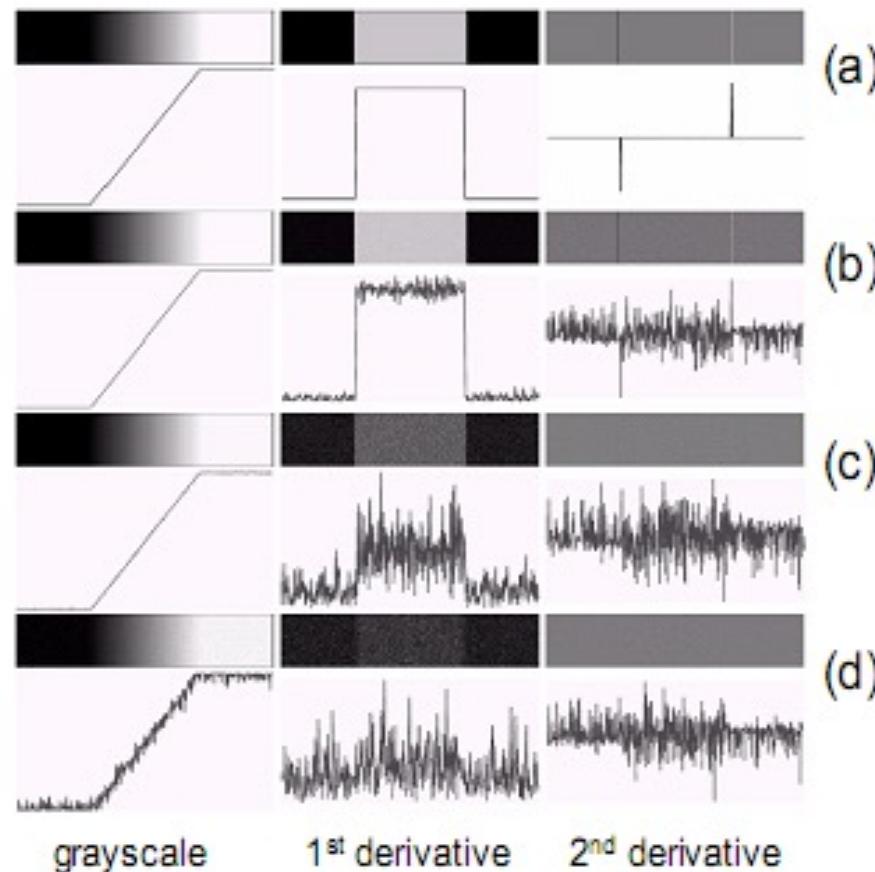


FIGURE 10.7 First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0, 0.1, 1.0$, and 10.0, respectively. Second column: first-derivative images and gray-level profiles. Third column: second-derivative images and gray-level profiles.

a
b
c
d

Ba bước cơ bản để phát hiện biên

1. Làm mịn ảnh để giảm nhiễu.
2. Phát hiện điểm biên: Đây là quá trình trích xuất từ ảnh tất cả các điểm ảnh là ứng cử viên tiềm năng của điểm biên.
3. Định vị biên: Mục đích của quá trình này là lựa chọn từ các điểm biên ứng cử viên ở bước 2 các điểm là điểm biên thực sự để tạo thành biên.

Các phương pháp phát hiện biên

- **Đạo hàm bậc 1 / Phương pháp Gradient**
 - Toán tử Roberts
 - Toán tử Sobel
 - Toán tử Prewitt
- **Đạo hàm bậc 2**
 - Laplacian
 - Laplacian of Gaussian
 - Difference of Gaussian
- **Phát hiện biên tối ưu**
 - Phương pháp Canny

Phát hiện biên sử dụng đạo hàm bậc 1

- **Toán tử Roberts**
- **Toán tử Sobel**
- **Toán tử Prewitt**

TÍNH CHẤT CỦA GRADIENT

- Gradient của ảnh $f(x,y)$ tại vị trí (x,y) được xác định như sau:

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

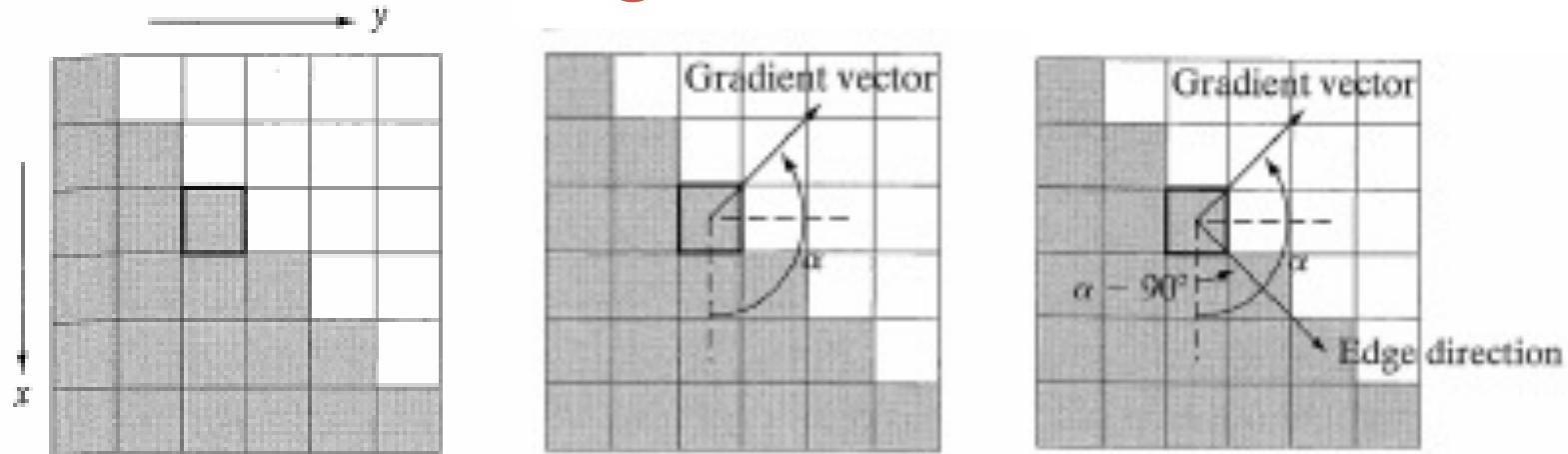
- Một số tính chất của vector gradient:

- Chỉ ra hướng của tốc độ thay đổi cực đại của ảnh tại vị trí (x,y) .
- Độ lớn:

➤ Góc: $|G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Độ lớn và hướng tại một điểm ảnh



- Giả sử cửa sổ kích thước 3×3 , đạo hàm bậc 1 theo x được tính bằng cách trừ pixel hàng trên cùng cho hàng dưới cùng. Đạo hàm bậc 1 theo y được tính bằng cách trừ pixel hàng bên phải cho hàng ngoài cùng bên trái.
- Pixel màu xám có giá trị 0, pixel màu trắng có giá trị 1.

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix} \quad |G| = 2\sqrt{2} \quad \theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) = -45^0$$

XẤP XỈ GRADIENT- MẶT NẠ 1D

- Với cách xấp xỉ:

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

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

- Ta có thể thực hiện $\frac{\partial f}{\partial x}$ và $\frac{\partial f}{\partial y}$ sử dụng mặt nạ 1D:

-1
1

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

-1	1
----	---

$$\frac{\partial f}{\partial y}$$

XẤP XỈ GRADIENT- TOÁN TỬ ROBERTS

- Một cách xấp xỉ gradient khác theo công thức:

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

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

- Khi đó $\frac{\partial f}{\partial x}$ và $\frac{\partial f}{\partial y}$ có thể được thực hiện bằng mặt nạ sau (toán tử Roberts):

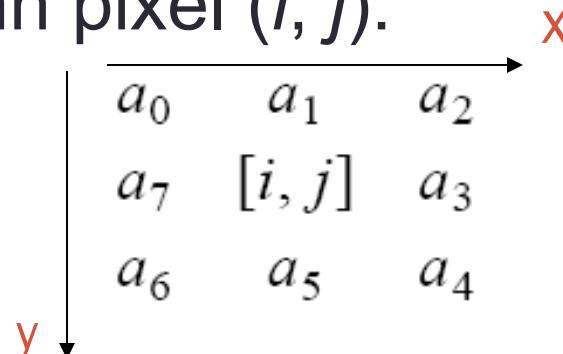
$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

MỘT CÁCH XÂP XỈ GRADIENT KHÁC

- Xét cửa sổ quanh pixel (i, j) :

3 x 3 neighborhood:



- Các đạo hàm thành phần có thể tính như sau:

$$M_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$M_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)$$

- Hàng số c nhấn mạnh rằng các pixel đó gần tâm cửa sổ.

Toán tử Prewitt và toán tử Sobel

- Toán tử Prewitt (c=1)

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Toán tử Sobel (c=2)

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Các bước phát hiện biên sử dụng gradient

(1) Smooth the input image ($\hat{f}(x, y) = f(x, y) * G(x, y)$)

$$(2) \hat{f}_x = \hat{f}(x, y) * M_x(x, y) \longrightarrow \frac{\partial f}{\partial x}$$

$$(3) \hat{f}_y = \hat{f}(x, y) * M_y(x, y) \longrightarrow \frac{\partial f}{\partial y}$$

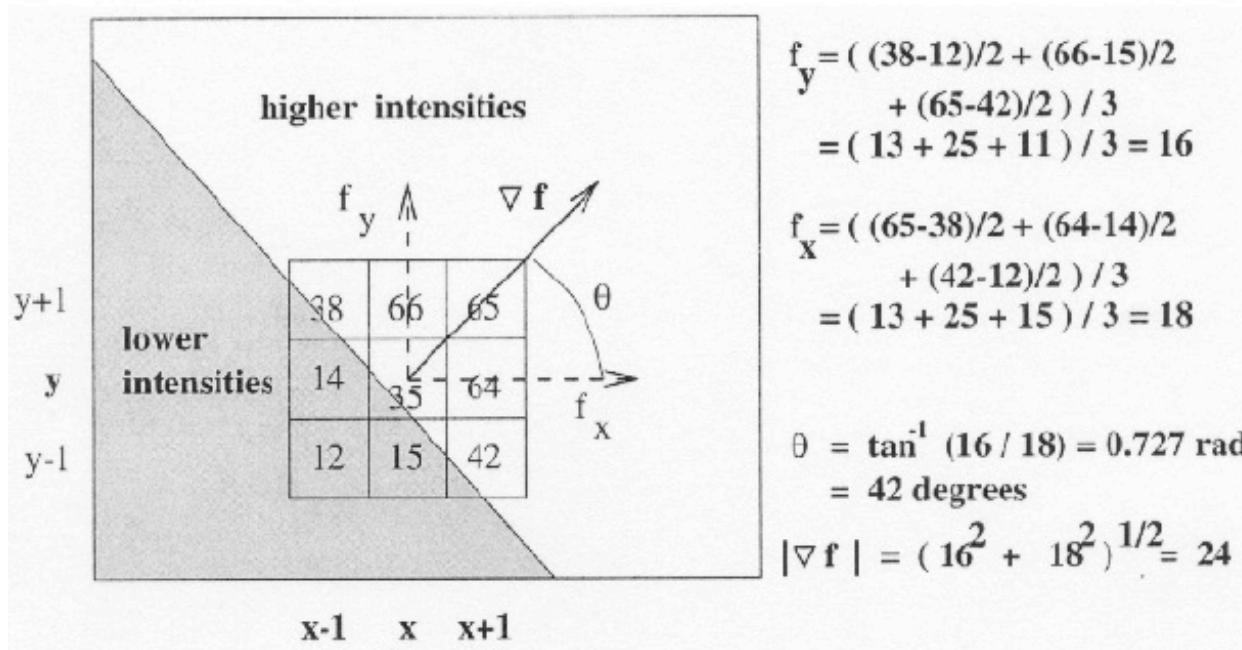
$$(4) magn(x, y) = |\hat{f}_x| + |\hat{f}_y|$$

$$(5) dir(x, y) = \tan^{-1}(\hat{f}_y / \hat{f}_x)$$

(6) If $magn(x, y) > T$, then possible edge point

Ví dụ (sử dụng toán tử Prewitt)

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

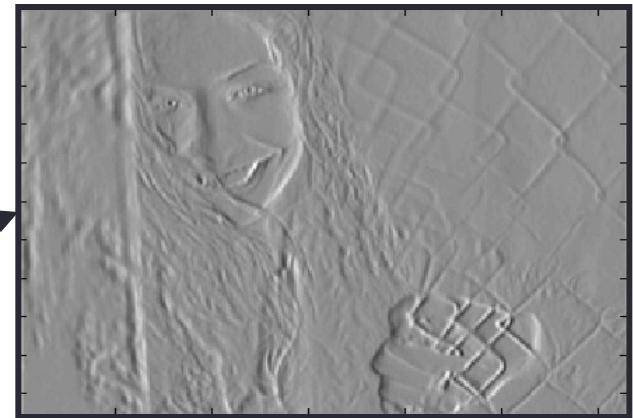


Chú ý: Trong ví dụ này, f_x và f_y được chia cho 2 và 3 với mục đích chuẩn hóa.

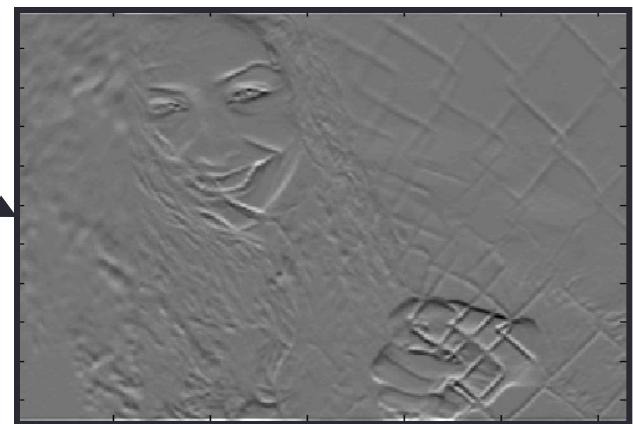
Ví dụ (1)



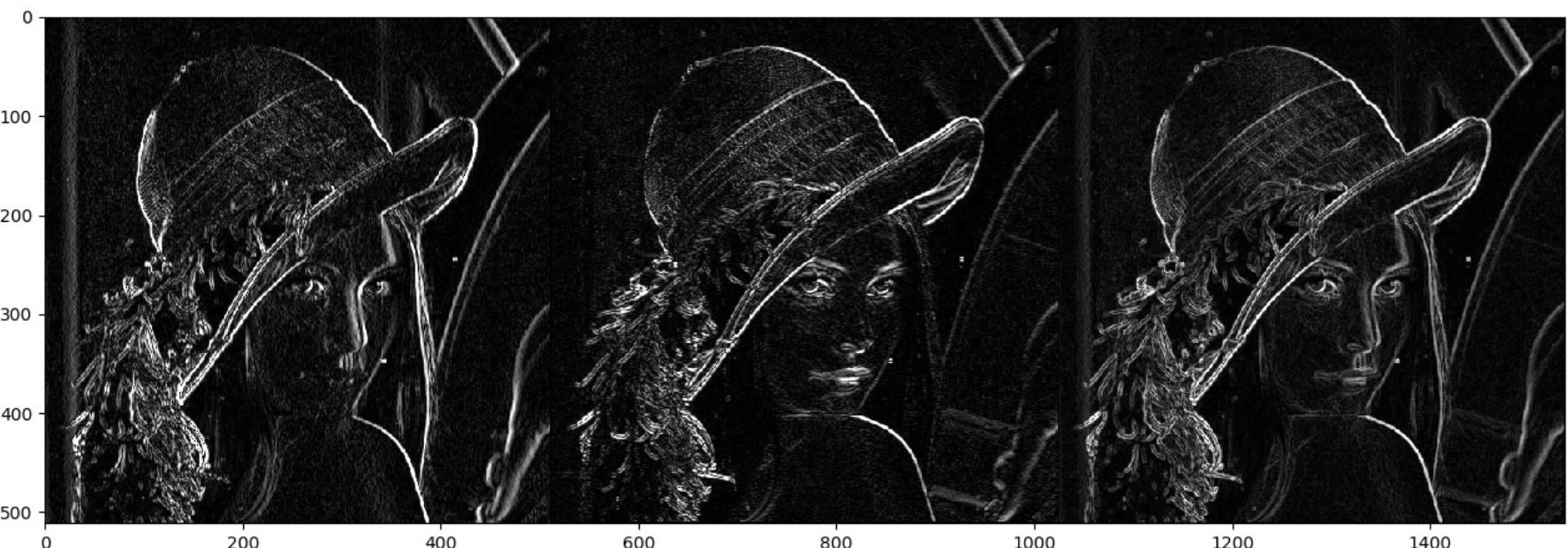
$$\frac{d}{dx} I$$



$$\frac{d}{dy} I$$

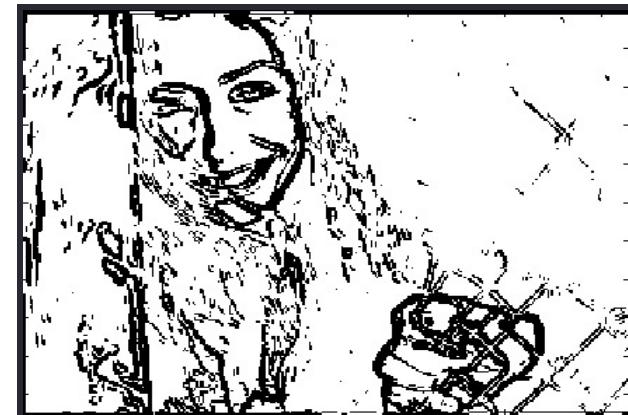
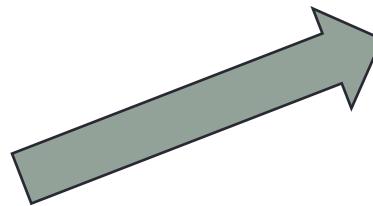


Bộ lọc Sobel



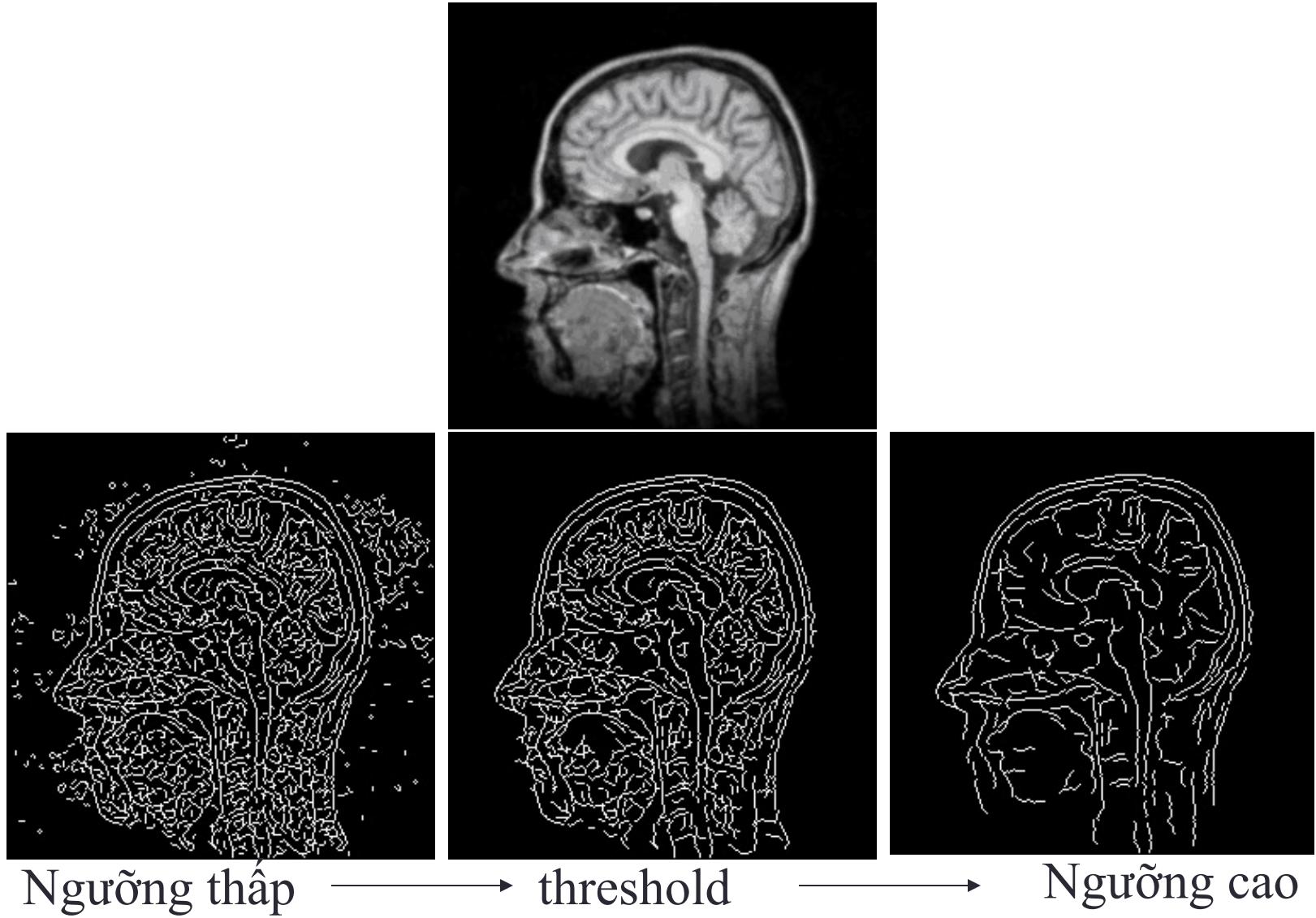
Ví dụ (2)

$$\nabla = \sqrt{\left(\frac{d}{dx} I\right)^2 + \left(\frac{d}{dy} I\right)^2}$$



$\nabla \geq Threshold = 100$

Ảnh hưởng của giá trị ngưỡng



4.1(D) Cho ảnh I như sau:

$$I = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 15 & 15 & 15 & 2 & 2 \\ 2 & 2 & 13 & 12 & 16 & 2 & 2 \\ 2 & 2 & 15 & 15 & 15 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

a. Thực hiện nhân chập ảnh I với các ma trận H_x và H_y rồi cộng với nhau để được ảnh I_1

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ và } H_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Phát hiện biên sử dụng đạo hàm bậc 2

- **Toán tử Laplacian**
- **Biến thể của toán tử Laplacian**

Toán tử Laplacian

❖ Toán tử Laplacian

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

❖ Trong đó đạo hàm bậc 2 theo hướng x:

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

Theo hướng y:

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

Toán tử Laplacian (cont...)

- ❖ Toán tử Laplacian được tính bởi:

$$\begin{aligned}\nabla^2 f = & [f(x+1, y) + f(x-1, y) \\ & + f(x, y+1) + f(x, y-1)] \\ & - 4f(x, y)\end{aligned}$$

- ❖ Bộ lọc Laplacian:

0	1	0
1	-4	1
0	1	0

Toán tử Laplacian (cont...)

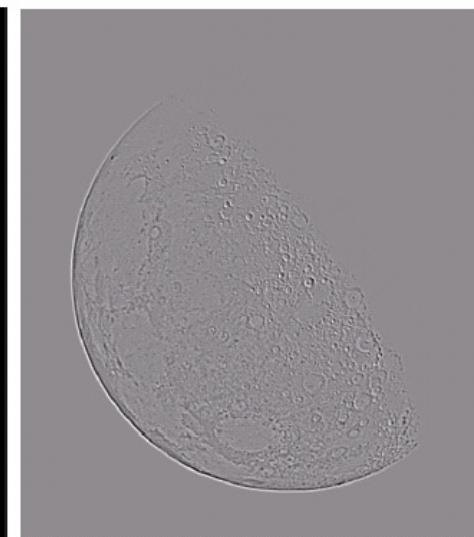
- ❖ Sử dụng bộ lọc Laplacian ta được ảnh mới trong đó biên của các đối tượng hoặc các vùng không liên tục sẽ được làm rõ



Original
Image



Laplacian
Filtered Image

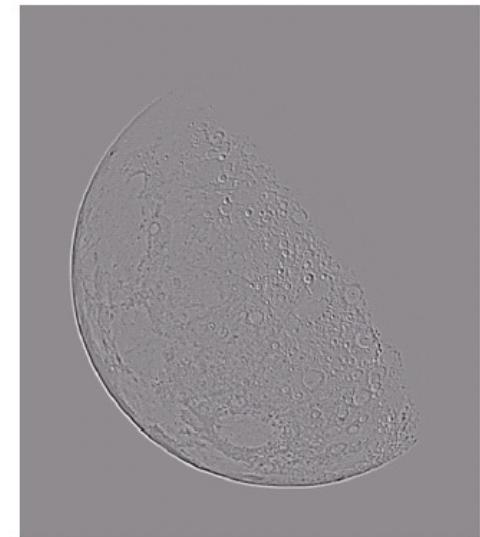


Laplacian
Filtered Image
Scaled for Display

Toán tử Laplacian (cont...)

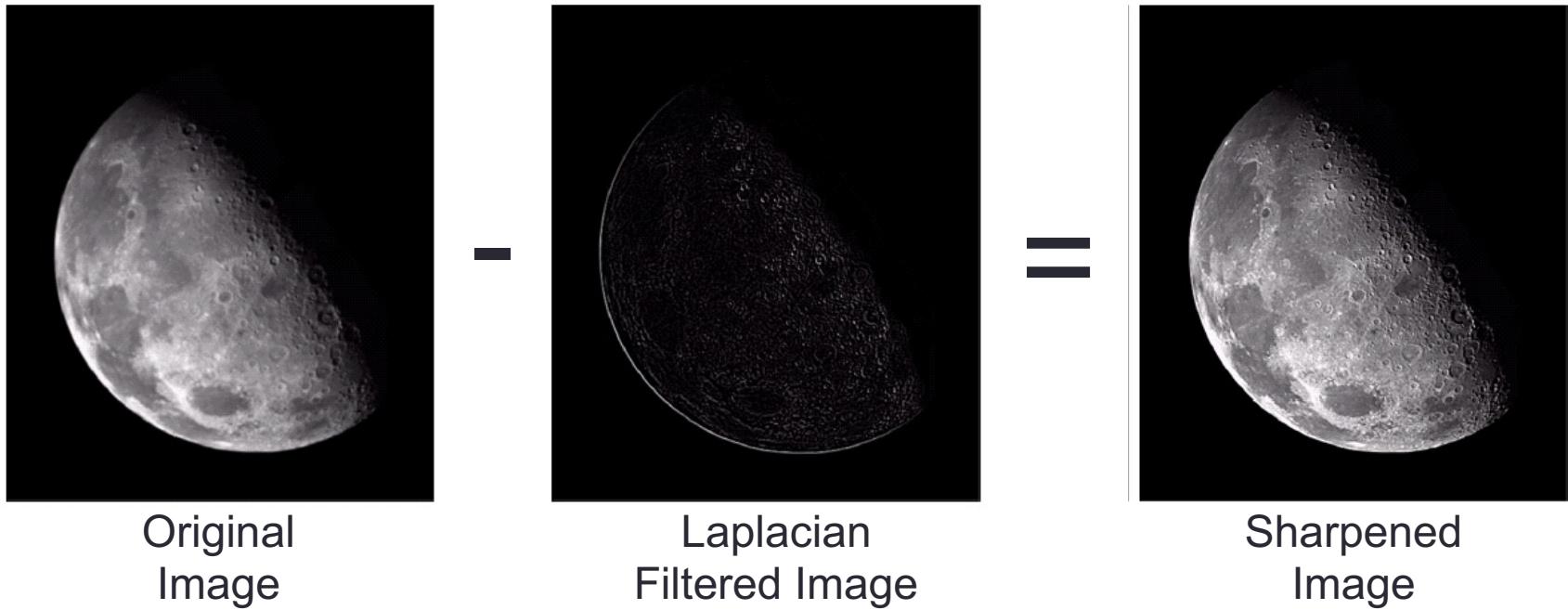
- Bản thân bộ lọc Laplacian không làm tăng cường chất lượng ảnh mà chỉ làm rõ nét các vị trí thay đổi trong ảnh.
- Để tăng cường chất lượng ảnh, ta phải thực hiện công việc sau:

$$g(x, y) = f(x, y) - \nabla^2 f$$



Laplacian
Filtered Image
Scaled for Display

Tăng cường ảnh bằng Laplacian

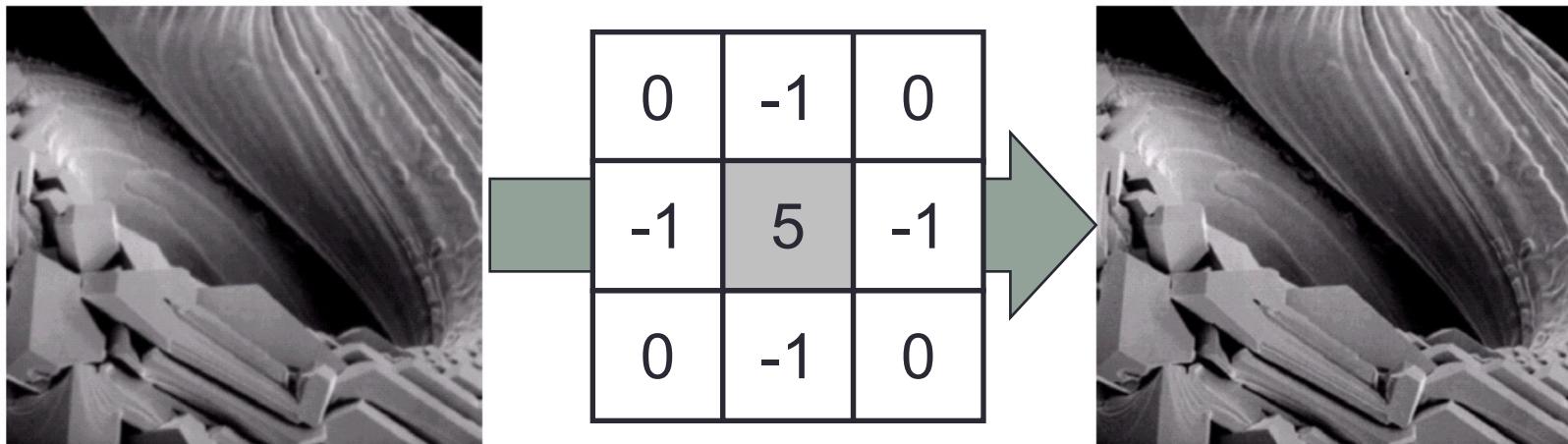


Tăng cường ảnh bằng Laplacian

- ❖ Toàn bộ quá trình trên có thể thực hiện bằng bộ lọc như sau:

$$\begin{aligned} g(x, y) &= f(x, y) - \nabla^2 f \\ &= f(x, y) - [f(x+1, y) + f(x-1, y) \\ &\quad + f(x, y+1) + f(x, y-1) \\ &\quad - 4f(x, y)] \\ &= 5f(x, y) - f(x+1, y) - f(x-1, y) \\ &\quad - f(x, y+1) - f(x, y-1) \end{aligned}$$

Tăng cường ảnh bằng Laplacian



Biến thể của bộ lọc Laplacian

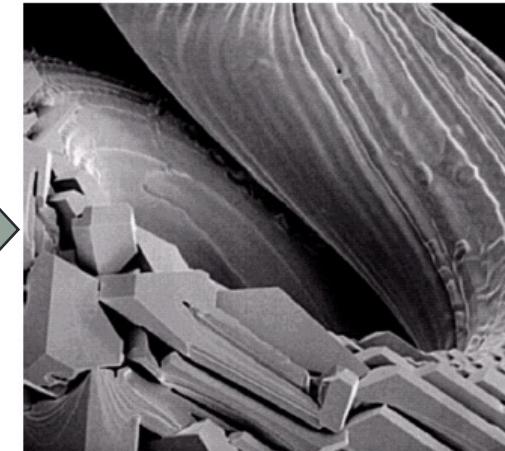
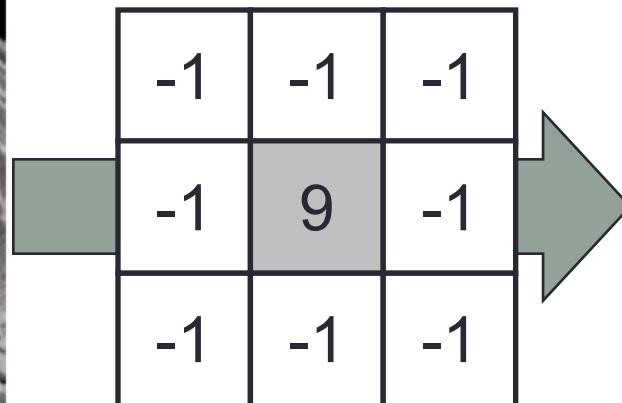
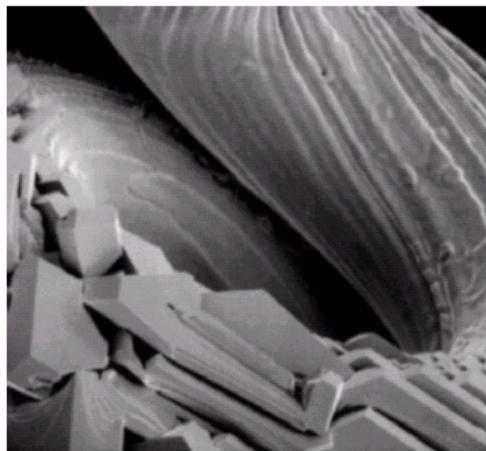
- ❖ Biến thể của bộ lọc Laplacian

0	1	0
1	-4	1
0	1	0

Simple
Laplacian

1	1	1
1	-8	1
1	1	1

Variant of
Laplacian



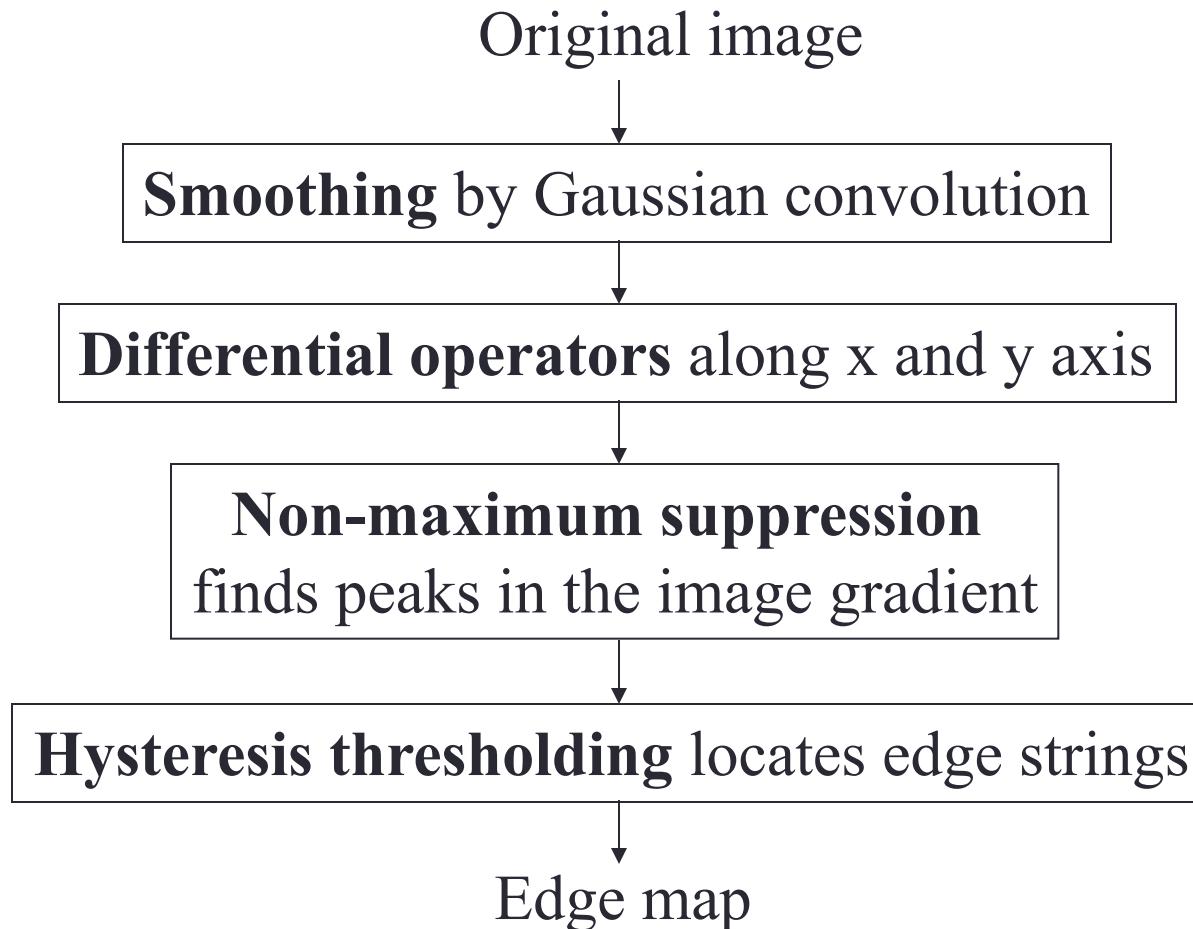
PHÁT HIỆN BIÊN TỐI ƯU PHƯƠNG PHÁP CANNY

ĐẶC ĐIỂM CỦA PHƯƠNG PHÁP CANNY

- **Tỉ lệ lỗi thấp:** Tìm tất cả các biên và không gì ngoài biên. Điều đó nghĩa là biên tìm thấy phải gần nhất có thể so với biên gốc.
- **Xác định tốt vị trí biên:** Khoảng cách giữa biên tìm thấy và biên thực cần phải tối thiểu.
- **Đáp ứng điểm biên duy nhất:** Bộ phát hiện biên trả về duy nhất một điểm đối với mỗi điểm biên thực. Điều này có nghĩa là bộ phát hiện biên không được xác định nhiều pixel biên khi chỉ có duy nhất một pixel biên tồn tại.

Flow-chart of Canny Edge Detector*

(J. Canny'1986)



CÁC BƯỚC CỦA PHƯƠNG PHÁP CANNY

1. **Smoothing** (làm mịn): Sử dụng bộ lọc Gaussian để giảm nhiễu cho ảnh.
2. **Gradient operator**: Tính độ lớn và hướng của gradient theo trục x và y sử dụng toán tử roberts hoặc Sobel.
3. **Non-maximum suppression**: Loại bỏ những điểm không đạt giá trị cực đại.
4. **Hysteresis thresholding**: Phân ngưỡng kép để tìm ra điểm đầu và cuối của biên

Bước 1: Giảm nhiễu

- Do bộ lọc Gaussian sử dụng cửa sổ lọc đơn giản nên được sử dụng để giảm nhiễu trong thuật toán Canny.
- Kích thước cửa sổ lọc càng lớn thì độ nhạy đối với nhiễu của thuật toán càng giảm.
- Ví dụ một xấp xỉ của bộ lọc Gaussian:

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

Bước 2: Tính gradient thành phần

- Sự rõ ràng của biên được phát hiện bằng cách tính gradient của ảnh.
- Có thể tính gradient sử dụng toán tử Roberts hoặc toán tử Sobel.
- Ví dụ:

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

Gx

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

Gy

- $|G| = |G_x| + |G_y|$
- Hướng của biên: $\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$

Bước 2 (tiếp)

- Khi đã biết hướng của biên, bước tiếp theo là liên hệ hướng này với một hướng có thể lân theo trong ảnh.
- Ví dụ các pixel của ảnh kích thước 5×5 như sau:

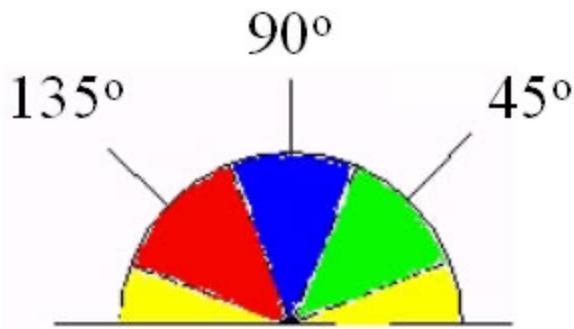
x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

Khi xem xét "a", chỉ có thể có 4 hướng khi mô tả các pixel lân cận - **0 độ** (theo hướng nằm ngang), **45 độ** (theo hướng dương của đường chéo), **90 độ** (theo hướng dọc), hoặc **135 độ** (theo hướng âm của đường chéo).

- Vì thế biên được định hướng vào một trong 4 hướng này tùy thuộc vào hướng của biên gần cãi nhất.

Bước 2 (tiếp)

- Nếu hướng biên rơi vào trong **dải màu vàng** (0 đến 22.5 & 157.5 đến 180 độ) thì được coi là 0 độ.
- Nếu hướng biên rơi vào dải **màu xanh lá cây** (22.5 đến 67.5 độ) được coi là 45 độ.
- Nếu hướng biên rơi vào dải **màu xanh da trời** (67.5 đến 112.5 độ) thì được coi là 90 độ
- Nếu hướng biên rơi vào dải **màu đỏ** (112.5 đến 157.5 độ) thì được coi là 135 độ.



Bước 3: Loại bỏ các giá trị không phải cực đại

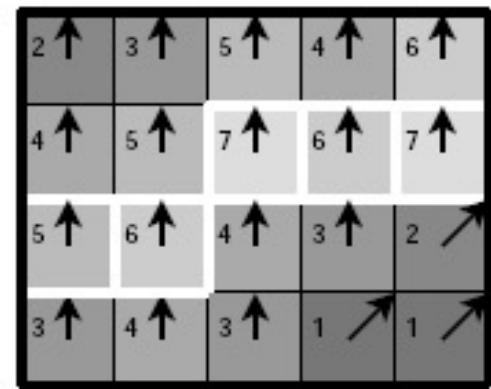
- Đây là kỹ thuật nhằm làm mảnh đường biên.
- Thực hiện tìm kiếm để xác định độ lớn gradient có phải là giá trị cực đại địa phương theo hướng gradient hay không.
- Tại mỗi pixel, thiết lập giá trị pixel bằng 0 nếu độ lớn của nó không lớn hơn độ lớn của 2 lân cận theo **hướng gradient** và giữ lại pixel có độ lớn cực đại.

Ví dụ bước 3

- Nếu góc gradient được làm tròn thành 0 độ (nghĩa là biên theo hướng **bắc- nam**), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng **đông và tây**.
- Nếu góc gradient được làm tròn thành 90 độ (nghĩa là biên theo hướng **đông- tây**), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng **nam và bắc**.
- Nếu góc gradient được làm tròn thành 135 độ (nghĩa là biên theo hướng **đông bắc- tây nam**), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng **tây bắc và đông nam**.
- Nếu góc gradient được làm tròn thành 45 độ (nghĩa là biên theo hướng **tây bắc- đông nam**), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng **đông bắc và tây nam**.

Ví dụ (bước 3)

- Minh họa quá trình loại bỏ các giá trị không phải cực đại.
 - Độ lớn biên được biểu diễn bằng màu và số, trong khi hướng biên được biểu diễn bằng các mũi tên.
 - Các pixel biên kết quả được đánh dấu bằng đường viền màu trắng.



Bước 4: Phát hiện biên bằng phân ngưỡng kép

- Sử dụng ngưỡng T để giảm số biên giả.
 - Tất cả các pixel có giá trị nhỏ hơn T được thiết lập giá trị 0
 - Chọn giá trị T phù hợp rất khó.
 - Nếu chọn T quá thấp sẽ xuất hiện nhiều biên giả.
 - Nếu chọn T quá cao sẽ làm mất một số biên thực.
- **Giải pháp:** Sử dụng 2 ngưỡng **T_1 cao và T_2 thấp** (thường chọn $T_1 = 2 \cdot T_2$)

Bước 4: Phát hiện biên bằng phân ngưỡng kép

- Nếu pixel có giá trị **lớn hơn T_1** sẽ được coi là điểm biên ngay lập tức.
- Các pixel có giá trị **nhỏ hơn T_2** sẽ không được coi là điểm biên.
- Pixel có giá trị nằm giữa T_2 và T_1 sẽ được xem xét tiếp: nếu pixel này có kết nối với một trong số các điểm biên khác thì sẽ được coi là điểm biên, ngược lại sẽ không coi là điểm biên.

Ví dụ về các bước của phương pháp Canny



(a) Original



(b) Smoothed



(c) Gradient magnitudes



(d) Edges after non-maximum suppression



(e) Double thresholding



(f) Edge tracking by hysteresis



(g) Final output

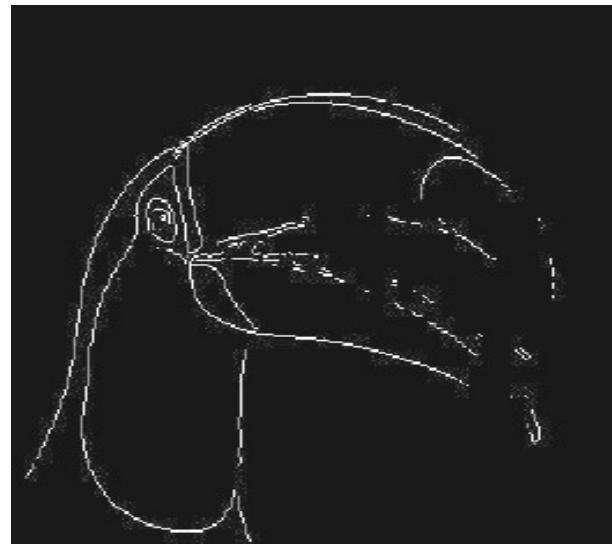
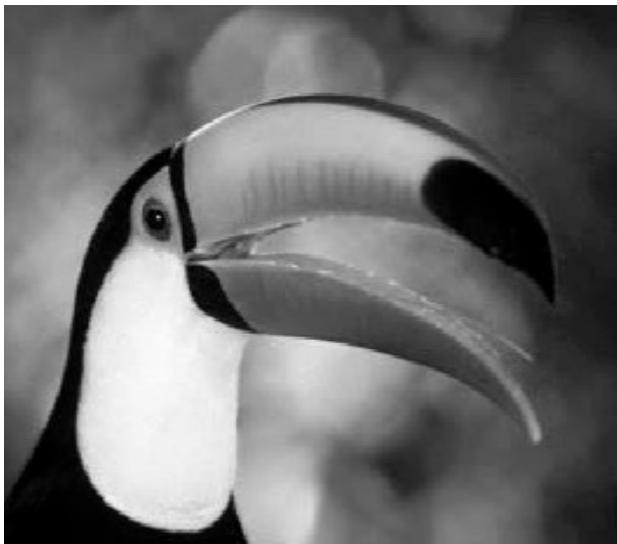


Sobel

LOG

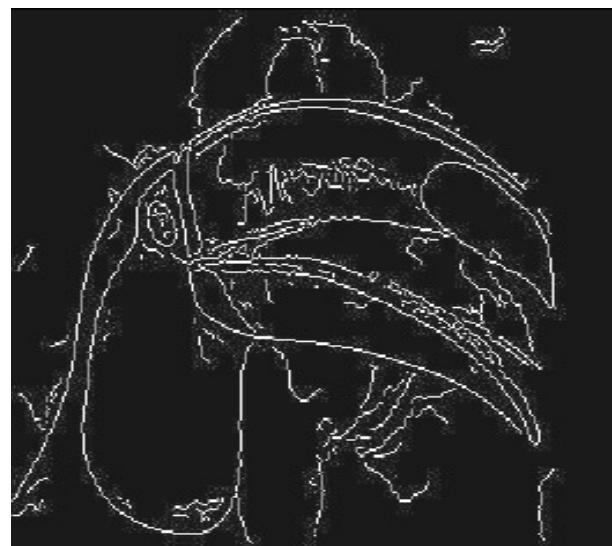
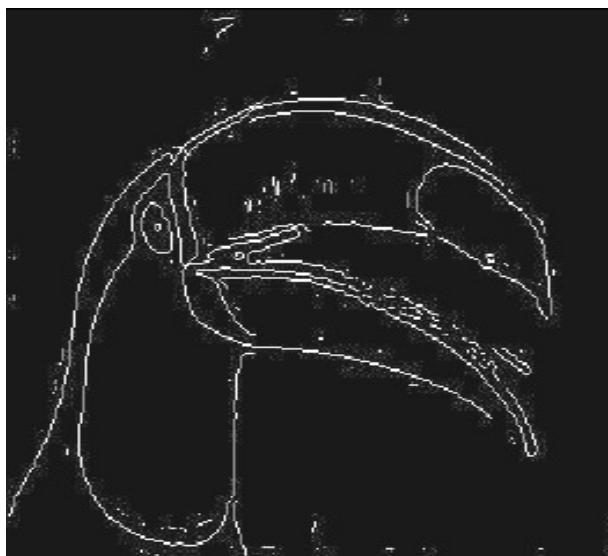


Canny



Sobel

LOG



Canny



Sobel

LOG



Canny

$$I = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 15 & 15 & 15 & 2 & 2 \\ 2 & 2 & 13 & 12 & 16 & 2 & 2 \\ 2 & 2 & 15 & 15 & 15 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

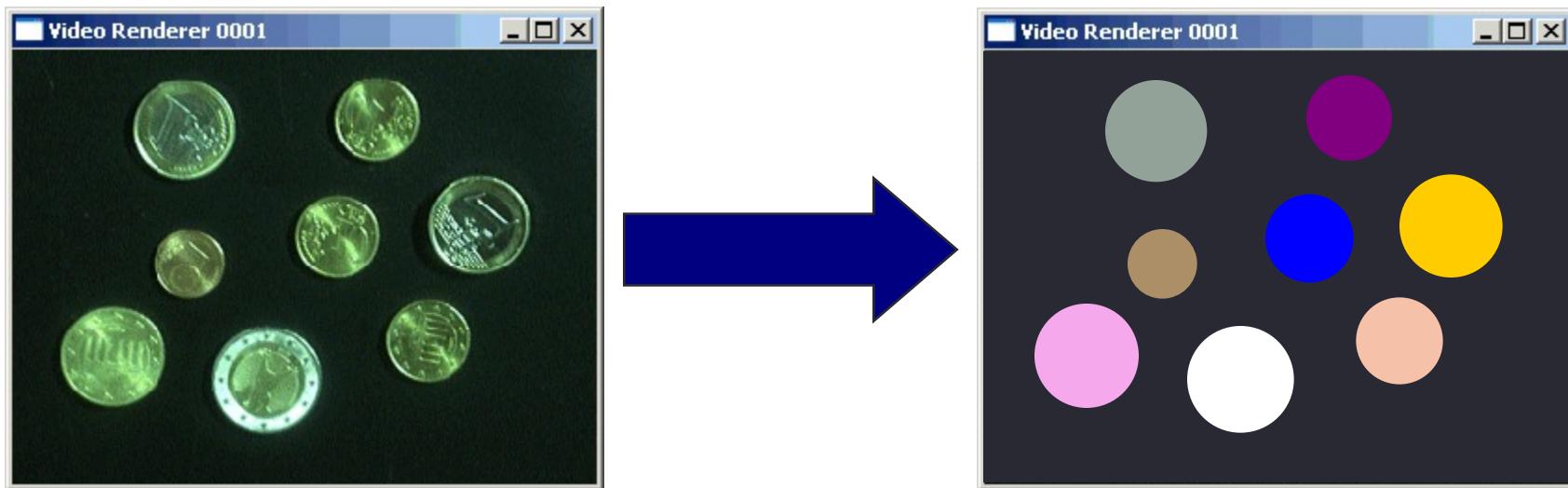
Thực hiện nhân chập ảnh I ở trên với H_z dưới đây để
được I_2

$$H_z = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

PHÂN VÙNG ẢNH

Khái niệm phân vùng ảnh

- Phân vùng ảnh là quá trình chia bức ảnh thành các vùng đồng nhất hoặc các đối tượng có cùng đặc điểm theo các tiêu chí định trước.
- Các tiêu chí có thể là: cường độ, lược đồ xám, giá trị trung bình (mean), phương sai (variance),...



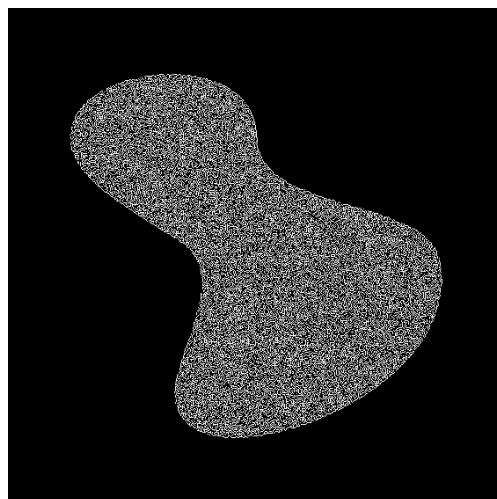
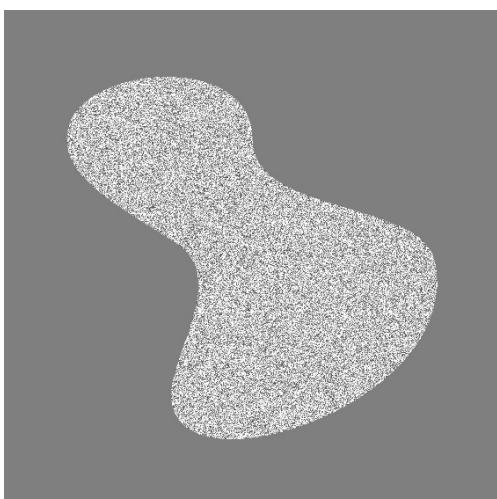
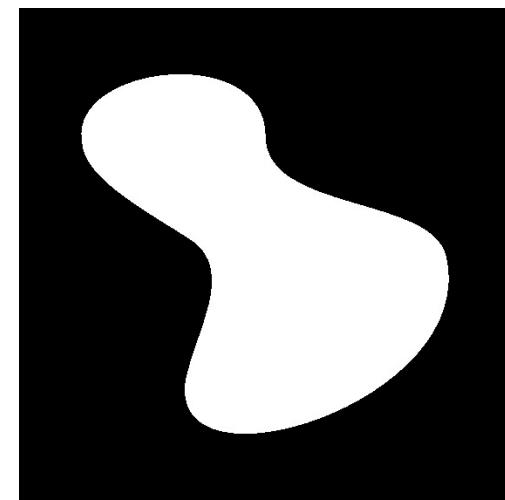
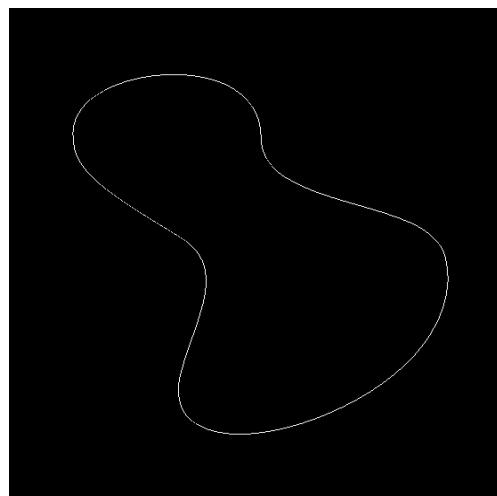
Khái niệm phân vùng ảnh

- Phần lớn các thuật toán phân vùng đều dựa vào một trong hai đặc điểm cơ bản của giá trị cường độ: **sự không liên tục và tính tương đồng** (discontinuity and similarity).

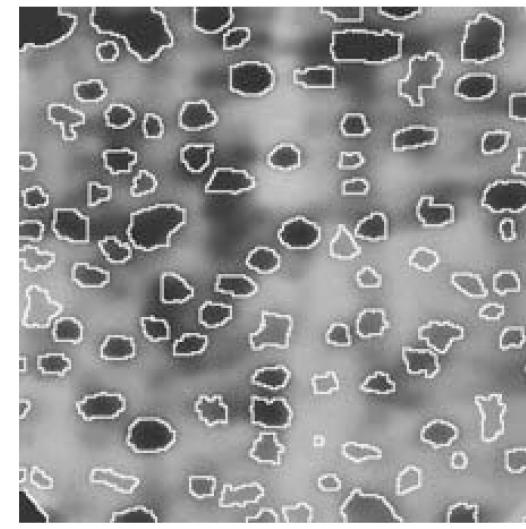
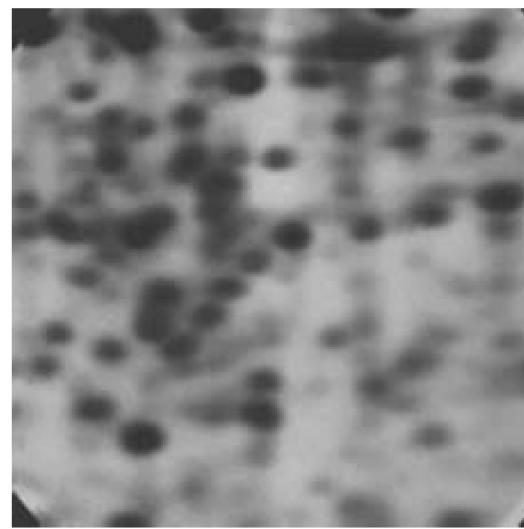
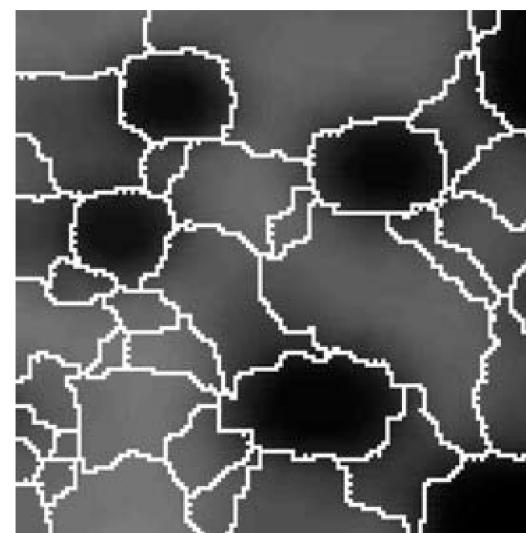
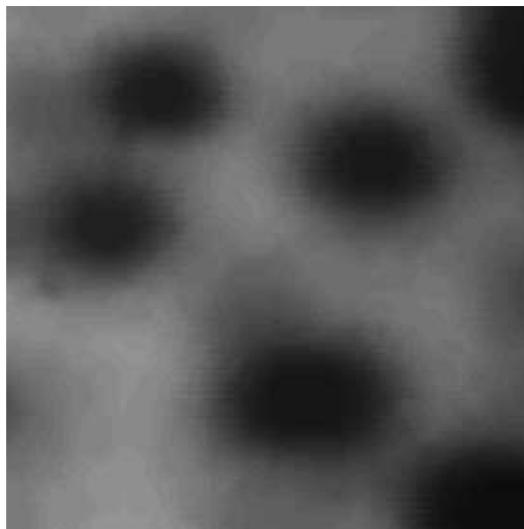
Phân vùng ảnh

- ❑ Cách tiếp cận 1: phân chia bức ảnh dựa vào sự thay đổi đột ngột của mức xám (ví dụ biên) hay còn gọi là *phân vùng dựa vào biên*.
 - Phát hiện biên
- ❑ Cách tiếp cận 2: phân chia bức ảnh thành các vùng tương tự nhau theo một tập các tiêu chí cho trước- *phân vùng dựa vào miền*.
 - Phân ngưỡng biên độ
 - Phân vùng theo miền đồng nhất
 - Phân vùng theo kết cấu bề mặt

Phân vùng dựa vào biên và phân vùng ảnh dựa vào miền



Ví dụ về phân vùng ảnh



PHÂN VÙNG ẢNH

R biểu diễn toàn bộ ảnh. Phân vùng ảnh là quá trình phân chia R thành n vùng R_1, R_2, \dots, R_n sao cho:

$$(1) \quad \bigcup_{i=1}^n R_i = R$$

(2) R_i là tập kết nối

(3) $R_i \cap R_j = \Phi$

(4) $P(R_i) = \text{TRUE}, \forall i = 1, 2, \dots, n$

(5) $P(R_i \cup R_j) = \text{FALSE}$ với R_i và R_j kề cận nhau

PHÂN VÙNG THEO NGƯỜNG BIÊN ĐỘ

- ❑ Thuật toán OTSU
- ❑ Thuật toán đẳng liệu (isodata)
- ❑ Thuật toán đối xứng nền (background symmetry)
- ❑ Thuật toán tam giác (triangle)

Reference

LÀM TRƠN LƯỢC ĐỒ XÁM

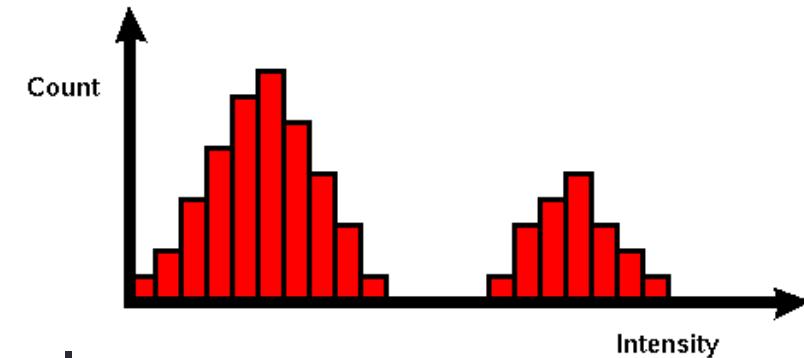
- Trong hầu hết các trường hợp, ngưỡng được chọn một cách tự động từ lược đồ xám của ảnh.
- Để loại bỏ các dao động nhỏ về độ sáng, các thuật toán thường sử dụng kỹ thuật làm trơn lược đồ xám ban đầu nhưng không dịch chuyển vị trí đỉnh bằng công thức sau:

$$h_{smooth}(b) = \frac{1}{W} \sum_{w=-(W-1)/2}^{(W-1)/2} h_{raw}(b-w)$$

- W thường chọn là số lẻ (3 hoặc 5)

TÁCH NGƯỠNG

- ▶ Xét lược đồ xám của ảnh như hình bên:
- ▶ Lược đồ xám cho thấy ảnh này gồm các đối tượng sáng trên một nền tối.
- ▶ Các giá trị cường độ của pixel đối tượng và pixel nền được nhóm lại thành hai miền rõ rệt.
- ▶ Như vậy: Để tách các đối tượng khỏi nền ta chọn ngưỡng T sao cho nó phân tách được 2 miền này.



TÁCH NGƯỠNG TOÀN CỤC CƠ BẢN

- ◆ **Ngưỡng là gì?**
- ➔ Bất cứ điểm (x,y) nào thỏa mãn $f(x,y) > T$ được gọi là *điểm đối tượng*, ngược lại nó được gọi là *điểm nền*.
- ➔ Ngưỡng đơn được cho theo công thức sau:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases}$$

- ➔ Khi T không đổi trên toàn bộ ảnh, ta gọi đây là phân ngưỡng toàn cục (global thresholding).

PHÂN NGƯỠNG TOÀN CỤC TỐI ƯU SỬ DỤNG PHƯƠNG PHÁP OTSU

- ◆ Phân ngưỡng được xem như bài toán lý thuyết quyết định theo thống kê ở đó *mục tiêu là tối thiểu sai số trung bình xảy ra khi ấn định pixel vào 2 miền (nhóm)*.
- ◆ Giải pháp là dựa vào luật quyết định của Bayes (Bayes decision rule) và dựa trên 2 tham số: hàm mật độ xác suất (PDF: Probability density function) của các mức sáng của mỗi nhóm và xác suất xảy ra của mỗi nhóm trong các ứng dụng.
- ◆ Vấn đề là ước lượng PDF không dễ vì thế bài toán thường được đơn giản hóa bằng cách giả định về dạng của PDF, ví dụ chúng có phân bố Gaussian.
- ➔ Tuy nhiên thậm chí khi đã đơn giản hóa, quá trình thực hiện giải pháp sử dụng các giả định này cũng rất phức tạp và không phải lúc nào cũng phù hợp với các ứng dụng thực tế ➔ **Sử dụng giải pháp thay thế: Phương pháp OTSU**

PHÂN NGƯỠNG TOÀN CỤC TỐI ƯU SỬ DỤNG PHƯƠNG PHÁP OTSU

- ◆ Phương pháp này tối ưu theo nghĩa là tối đa hóa phương sai giữa các nhóm (Between class variance).
- ◆ Ý tưởng cơ bản của phương pháp là các nhóm được phân ngưỡng tốt phải khác biệt với các giá trị cường độ của các pixel của chúng. Ngược lại, ngưỡng nào cho sự phân biệt tốt nhất giữa các nhóm (theo khía cạnh giá trị cường độ) sẽ được coi là **ngưỡng tối ưu**.
- ◆ Phương pháp này dựa hoàn toàn vào việc tính toán trên lược đồ xám của ảnh.

THUẬT TOÁN OTSU

1. Tính lược đồ xám chuẩn hóa của ảnh đầu vào. Tính các giá trị $p_i = n_i / MN$ ($i=0,1,\dots, L-1$).
 2. Tính tổng tích lũy $P_1(k)$ với $k=0,\dots,L-1$ theo công thức: $P_1(k) = \sum_{i=0}^k p_i$
 3. Tính trung bình tích lũy đến mức k là $m(k)$: $m(k) = \sum_{i=0}^k i.p_i$
 4. Tính trung bình cường độ toàn cục m_G : $m_G = \sum_{i=0}^{L-1} i.p_i$ (cường độ trung bình của toàn bộ ảnh).
 5. Tính phương sai giữa các nhóm $\sigma_B^2(k)$
- $$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$
6. Ngưỡng Otsu k^* là giá trị k thỏa mãn điều kiện $\sigma_B^2(k)$ đạt cực đại. Nếu cực đại không duy nhất thì k^* được tính là trung bình các giá trị k tương ứng với các cực đại tìm thấy được.

Ví dụ

- Tìm ngưỡng tự động theo phương pháp Otsu của ảnh sau:

$$\text{I} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Bài giải

i	p _i	P _{i(k)}	m(k)	m(G)	$\sigma_B^2(k)$
0	15/30	0.5000	0.0000	1.1667	1.36
1	5/30	0.6667	0.1667	1.1667	1.68
2	4/30	0.8000	0.4333	1.1667	1.56
3	3/30	0.9000	0.7333	1.1667	1.11
4	2/30	0.9667	1.0000	1.1667	0.51
5	1/30	1.0000	1.1667	1.1667	0.00

- Như vậy ngưỡng T = 1 với $\sigma_B^2(k) = 1.68$

Bài tập

- Sử dụng thuật toán tìm ngưỡng tự động để tách ngưỡng ảnh I, được biết ảnh có 10 mức xám:

$$I = \begin{bmatrix} 1 & 4 & 1 & 1 & 2 & 3 & 2 & 3 \\ 3 & 2 & 4 & 2 & 5 & 2 & 6 & 2 \\ 2 & 1 & 8 & 2 & 5 & 2 & 5 & 6 \\ 2 & 5 & 2 & 4 & 7 & 9 & 1 & 4 \\ 2 & 2 & 3 & 0 & 0 & 1 & 2 & 1 \\ 1 & 5 & 7 & 1 & 2 & 4 & 5 & 6 \end{bmatrix}$$

THUẬT TOÁN ĐẲNG LIỆU (ISODATA ALGORITHM)

Ngưỡng toàn cục cơ bản T được tính như sau:

1. Chọn giá trị ước lượng ban đầu cho $T=t_0$ (thông thường là mức xám trung bình trong bức ảnh).
2. Phân vùng bức ảnh sử dụng t_k để tạo ra tạo ra 2 nhóm pixel: G_1 gồm các pixel với mức xám $\leq t_k$ và G_2 gồm các pixel có mức xám $> t_k$
3. Tính mức xám trung bình của các pixel trong G_1 là $\mu_1(t_k)$ và trong G_2 là $\mu_2(t_k)$

THUẬT TOÁN ĐẲNG LIỆU (ISODATA ALGORITHM)

4. Tính giá trị ngưỡng mới:

$$t_{k+1} = \frac{\mu_1(t_k) + \mu_2(t_k)}{2} = \frac{1}{2} \begin{pmatrix} \sum_{z=0}^{t_k} z \cdot p(z) & \sum_{z=t_k+1}^{L-1} z \cdot p(z) \\ \sum_{z=0}^{t_k} p(z) & \sum_{z=t_k+1}^{L-1} p(z) \end{pmatrix}$$

5. Lặp lại các bước 2 – 4 cho đến khi sự khác nhau giữa các T trong các vòng lặp liên tiếp nhỏ hơn một giá trị giới hạn định trước ΔT hay $|t_{k+1} - t_k| < \Delta T$

Thuật toán này làm việc hiệu quả trong việc tìm giá trị ngưỡng T khi ảnh có lược đồ xám thích hợp.

Bài tập

- Sử dụng thuật toán tìm ngưỡng toàn cục đơn giản (hay còn gọi là thuật toán lặp, thuật toán đẳng liệu) cho bức ảnh với lược đồ xám chuẩn hóa với ngưỡng khởi đầu $To=5$:

r	0	1	2	3	4	5	6	7	8	9
p(r)	0,09	0,2	0,1	0,06	0,06	0,08	0,35	0,04	0,01	0,01

Bài tập

z	0	1	2	3	4	5
$p(z)$	0.500	0.167	0.133	0.100	0.067	0.033
$z.p(z)$	0.000	0.167	0.267	0.300	0.267	0.167
$t_0=3$						
$t_1=$	2.574					
$t_2=$	2.438					
$t_3=$	2.438					

Vậy ngưỡng $t = 2$

Bài tập

Câu 4. (4 Điểm)

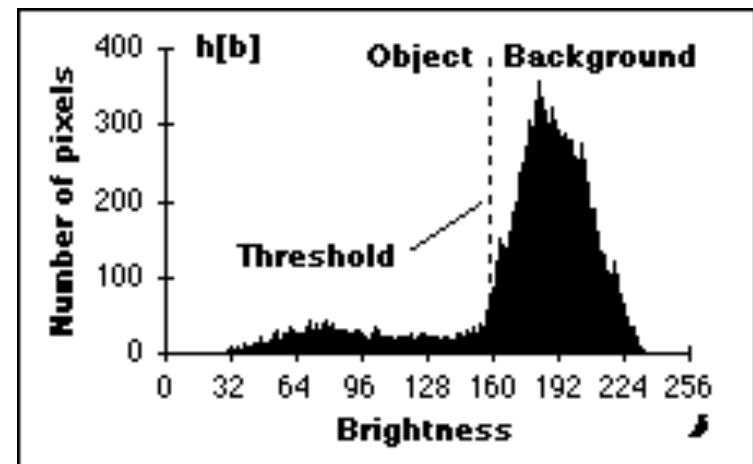
Thực hiện tìm ngưỡng tự động với thuật toán đẳng liêu cho bức ảnh I có biểu đồ tần suất sau:

g	0	1	2	3	4	5	6	7	8	9
$h(g)$	13	45	33	22	22	36	47	43	32	13

Mô tả từng bước cho đến khi tìm được ngưỡng mong muốn. Được biết ảnh có 10 mức xám.

THUẬT TOÁN ĐỐI XỨNG NỀN (BACKGROUND SYMMETRY ALGORITHM)

- Gia định trong lược đồ xám có đỉnh nằm ở phần nền và đối xứng quanh giá trị cực đại đó.
- Giá trị đỉnh ($\max p$) được xác định bằng cách tìm giá trị cực đại trong lược đồ xám.
- Thuật toán tìm kiếm phía các pixel không phải đối tượng quanh giá trị cực đại để tìm điểm $p\%$ (Điểm có giá trị mức sáng a sao cho $P(a) = p\%$).



VÍ DỤ

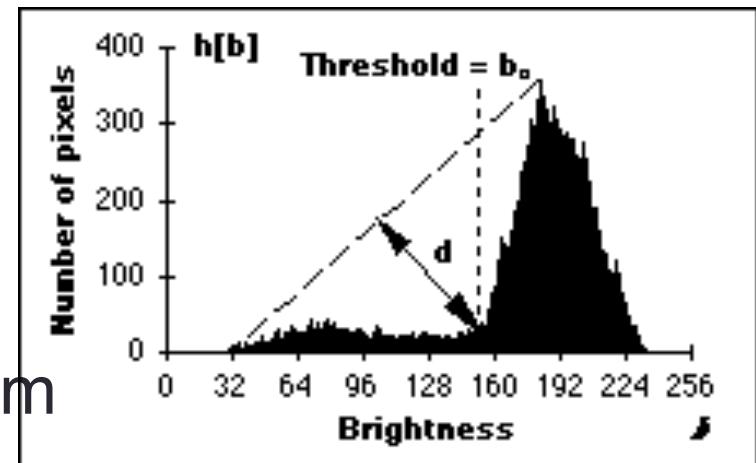
- Trong hình trước, các pixel đối tượng nằm ở bên trái của đỉnh nền (đỉnh nền nằm ở mức sáng **maxp = 183**).
- Do đó việc tìm kiếm thực hiện ở bên phải của đỉnh. Ví dụ chọn $p = 95\%$
- Điểm bên phải tương ứng với 95% mức sáng cực đại có độ sáng 216 (**p%**).
- Do giả định về tính đối xứng, ta tính được ngưỡng nằm ở bên trái của giá trị cực đại theo công thức

$$\theta = \text{max } p - (p\% - \text{max } p)$$

- Ngưỡng **$\theta=183 - (216 - 183) = 150$** .

THUẬT TOÁN TAM GIÁC (TRIANGLE ALGORITHM)

- Thuật toán này do Zack đề xuất được mô tả ở hình bên.
- Vẽ một đường thẳng Δ nối 2 điểm trên lược đồ xám (điểm cao nhất lược đồ xám có giá trị cường độ sáng b_{\max} và điểm có giá trị nhỏ nhất b_{\min} , $p=0\%$).
- Tại mỗi giá trị b từ $b = b_{\min}$ to $b = b_{\max}$ tính khoảng cách d giữa đường thẳng Δ và lược đồ xám $h[b]$.
- Giá trị b_0 nào cho d lớn nhất sẽ được coi là ngưỡng hay $\theta = b_0$



Bài tập

- Thực hiện tìm ngưỡng tự động với thuật toán đối xứng nền cho bức ảnh I có biểu đồ tần suất sau:

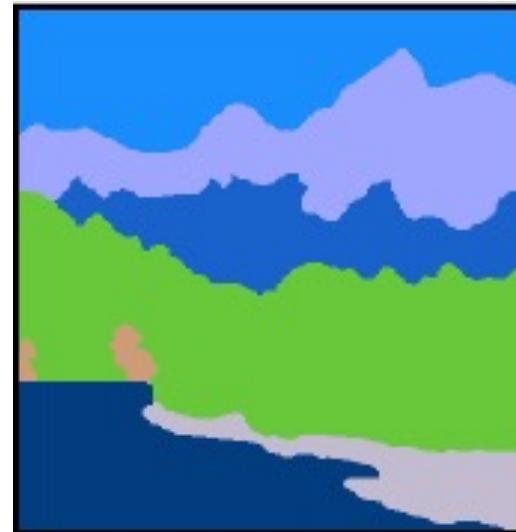
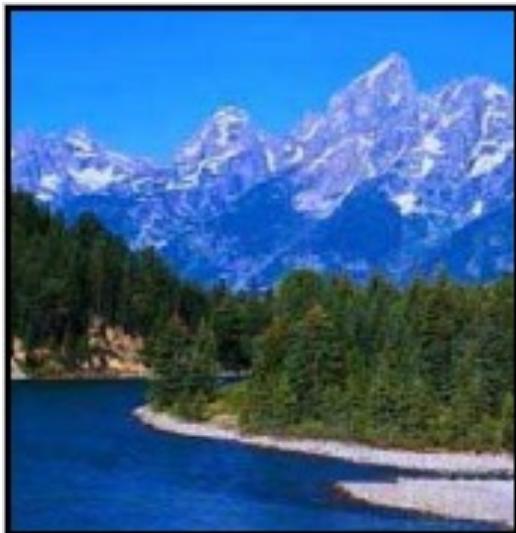
g	0	1	2	3	4	5	6	7	8	9
$h(g)$	2	3	4	5	7	8	12	47	10	2

Được biết độ chính xác cần tính là 88%.

PHÂN VÙNG DỰA VÀO MIỀN ĐỒNG NHẤT

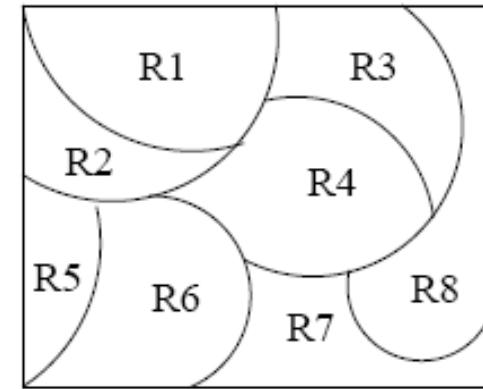
GIỚI THIỆU

- Không giống với phân vùng dựa vào biên (dựa vào sự không liên tục của cường số sáng), phân vùng dựa vào miền tìm ra các vùng một cách trực tiếp dựa trên sự **tương đồng** của các pixel lân cận nhau.
- Một số thuật toán cơ bản:
 - Phát triển vùng (region growing)
 - Hợp vùng (region merge)
 - Tách vùng (region split)
 - Chia tách và hợp vùng (Region splitting and merging)



Phân vùng dựa vào miền

- Thực hiện chia ảnh R thành các vùng nhỏ hơn R_1, R_2, \dots, R_n



- Gia sử $P(R_i)$ là một mệnh đề logic mà giá trị pixel trong vùng R_i thỏa mãn. (ví dụ, giá trị mức xám nằm trong khoảng từ 100 và 120).
- $P(R_i)=\text{True}$ trong vùng R_i nhưng không đúng vùng R_j

Phân vùng dựa vào miền

Các vùng nhỏ phải thỏa mãn các điều kiện sau:

$$(1) R_1 \cup R_2 \cup \dots \cup R_n = R$$

(2) R_i is connected

(3) $R_i \cap R_j = \text{empty}$

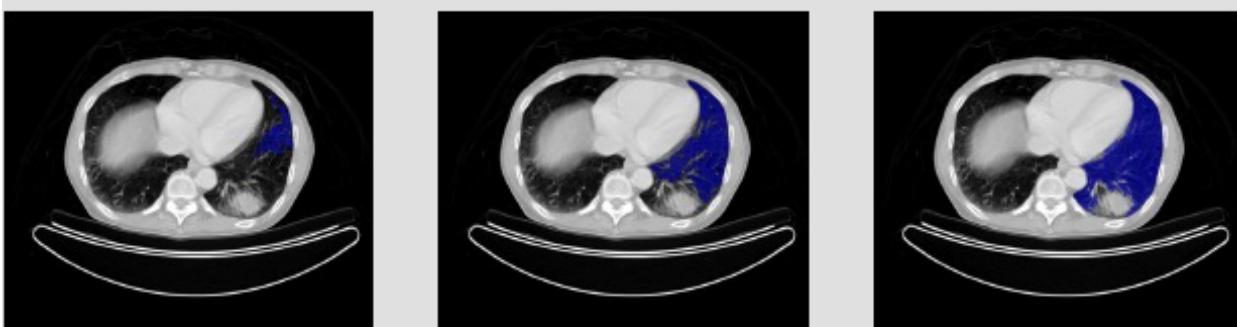
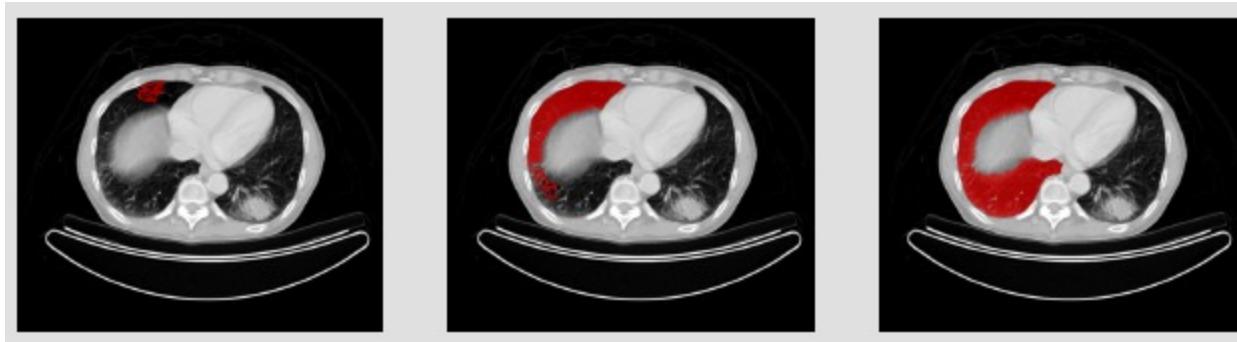
(4) $P(R_i) = \text{True}$ ($P()$ is a predicate)

(5) $P(R_i \cup R_j) = \text{False}$

Thuật toán Phát triển vùng

- Phát triển vùng khai thác trực tiếp vào thực tế là các pixel gần nhau thường có giá trị mức xám tương tự nhau.
- Thuật toán bắt đầu với một pixel đơn lẻ (hạt giống - **seed**) và dần thêm vào các pixel khác theo một tập các tiêu chuẩn đã định trước.
- Thuật toán:
 - (1) Chọn pixel hạt giống
 - (2) Kiểm tra các pixel lân cận và thêm vào vùng nếu nó tương tự pixel hạt giống.
 - (3) Lặp lại bước 2 cho mỗi pixel mới thêm vào; dừng lại khi không có pixel nào được thêm vào.

Ví dụ



*Convergence
obtained after
10085 iterations*

HỢP VÙNG

- Hợp vùng nhằm loại bỏ các biên và vùng giả bằng cách hợp nhất các vùng lân cận cùng thuộc một đối tượng.
- Cơ chế hợp nhất bắt đầu với một vùng thỏa mãn điều kiện (4) (ví dụ, các vùng được tạo ra nhờ phân ngưỡng)
- Sau đó, tiếp tục quá trình để thỏa mãn điều kiện (5) bằng cách hợp dần các vùng lân cận nhau.

$$(4) P(R_i) = \text{True}$$

$$(5) P(R_i \cup R_j) = \text{False}$$

Thuật toán hợp vùng

1. Chọn các vùng khởi tạo trong bức ảnh.
2. Xây dựng lược đồ lân cận vùng (regions adjacency graph – RAG).
3. Đối với mỗi vùng thực hiện:
 - (3.1) Xét vùng lân cận của nó và kiểm tra xem có tương tự vùng đang xét không.
 - (3.2) Nếu các vùng này tương tự nhau (có nghĩa là $P(R_i \cup R_j) = \text{TRUE}$) thì thực hiện hợp nhất chúng và cập nhật lại RAG.
4. Lặp lại bước 3 cho đến khi không có vùng nào được hợp nhất.

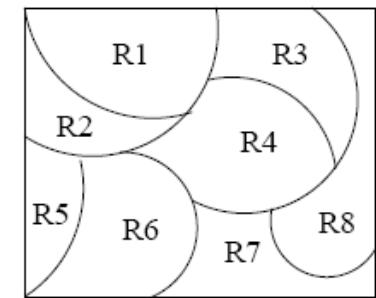
TÁCH VÙNG

- Chia tách vùng là cách tiếp cận đối lập với phát triển vùng.
- Đây là cách tiếp cận top-down và bắt đầu với giả định là toàn bộ bức ảnh là đồng nhất.
- Nếu điều này không đúng, bức ảnh được chia thành 4 ảnh nhỏ.
- Quá trình chia tách này được lặp lại cho đến khi ta chia tách bức ảnh thành các vùng đồng nhất.

TÁCH VÙNG

- Tách vùng thực hiện thêm các biên còn thiếu bằng cách chia tách các vùng chứa các phần đối tượng khác nhau.
- Cơ chế tách bắt đầu với một phần thỏa mãn điều kiện (5), ví dụ toàn bộ bức ảnh.

$$(5) P(R_i \cup R_j) = \text{False}$$



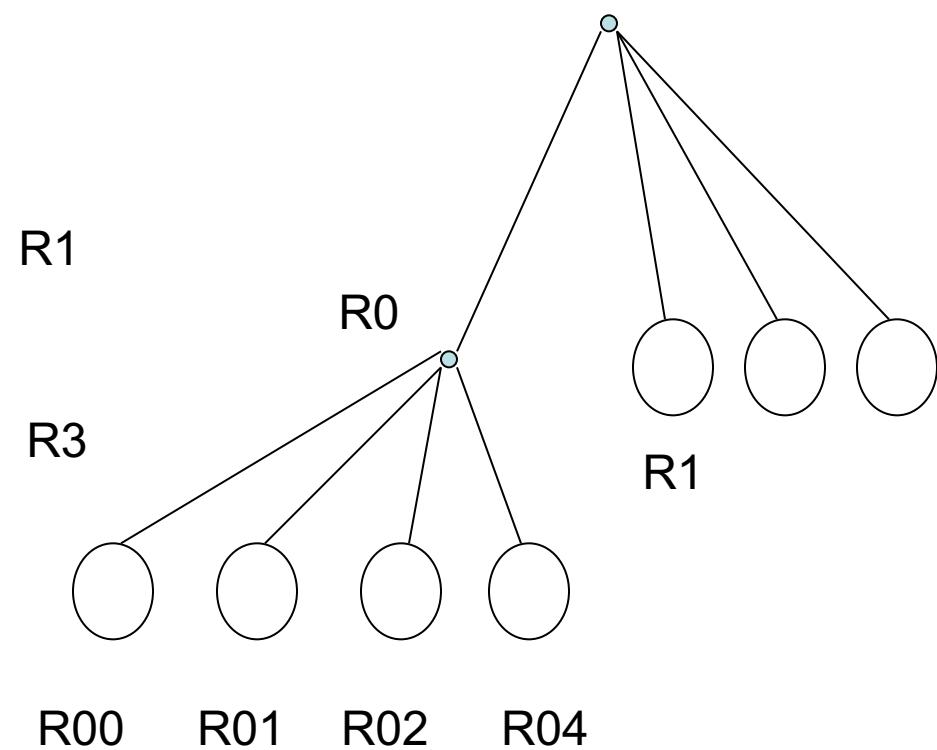
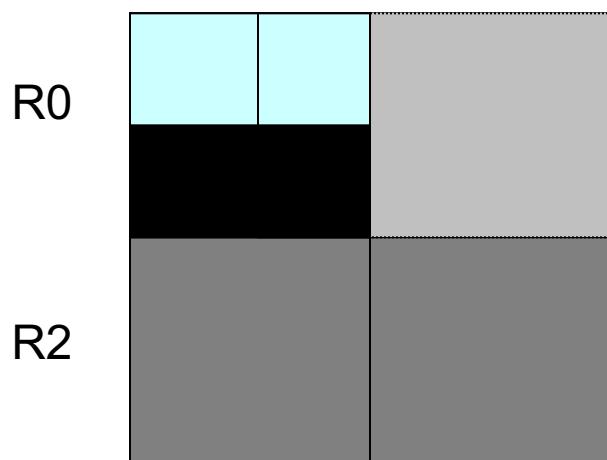
- Sau đó, chúng tiếp tục quá trình để thỏa mãn điều kiện (4) bằng cách chia tách dần các vùng ảnh.

$$(4) P(R_i) = \text{True}$$

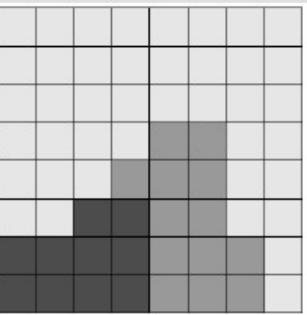
TÁCH VÙNG

- Giả sử ảnh gốc có kích thước $N \times N$, với N là lũy thừa của 2.
- Tất cả các vùng được tạo ra từ thuật toán chia tách có kích thước $M \times M$, ở đó M cũng là lũy thừa của 2.
- Bởi vì quá trình này được lặp lại, nó tạo ra cách biểu diễn có thể mô tả bằng cây ở đó mỗi node có 4 nhánh.
- Cây này được gọi là cây tứ phân (Quadtree)

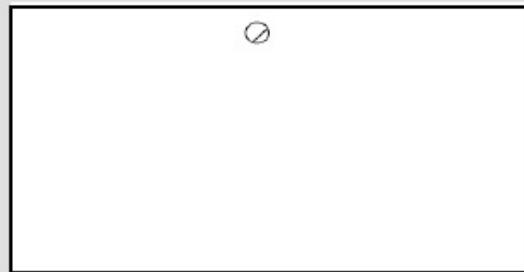
TÁCH VÙNG



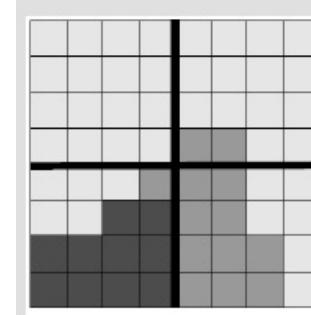
Ví dụ



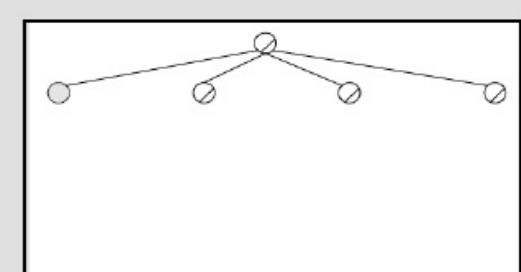
Initial image



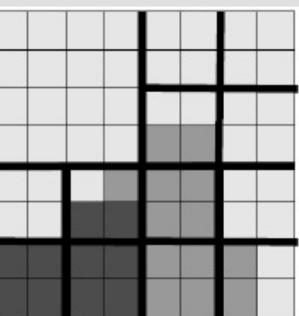
Building of the corresponding Quad-tree



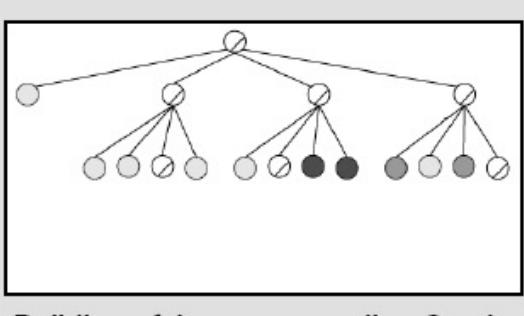
Initial image



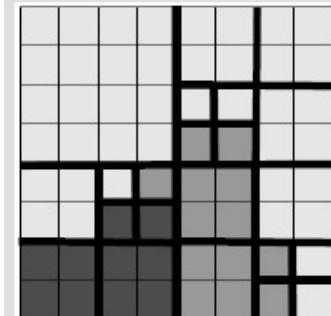
Building of the corresponding Quad-tree



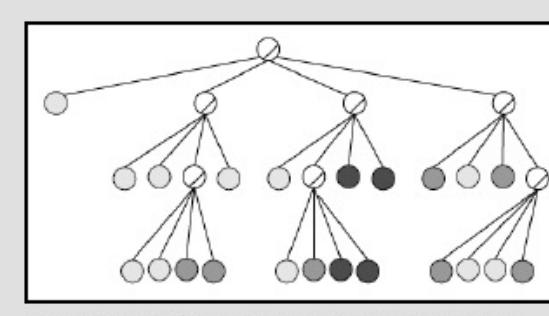
Initial image



Building of the corresponding Quad-tree



Initial image



Building of the corresponding Quad-tree

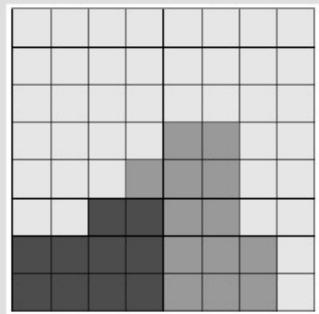
TÁCH VÀ HỢP VÙNG

- Kỹ thuật tách vùng có nhược điểm là chúng tạo ra những vùng có thể cận kề và đồng nhất nhưng không được hợp nhất với nhau.
- Phương pháp tách và hợp vùng là thuật toán lặp trong đó bao gồm cả quá trình tách và hợp trong mỗi vòng lặp.

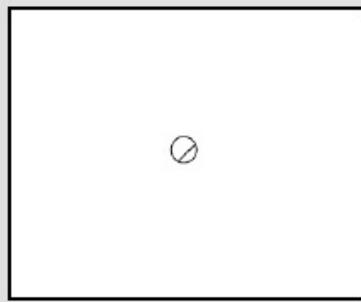
TÁCH VÀ HỢP VÙNG

- Nếu vùng R_i là không đồng nhất ($P(R_i) = \text{False}$) thì tách R_i thành 4 vùng con.
- Nếu 2 vùng kề cận nhau R_i, R_j là đồng nhất ($P(R_i \cup R_j) = \text{TRUE}$) thì hợp nhất chúng với nhau.
- Thuật toán dừng lại khi không thể tách hay hợp nhất.

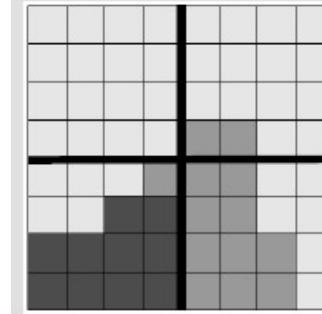
Xây dựng lược đồ lân cận



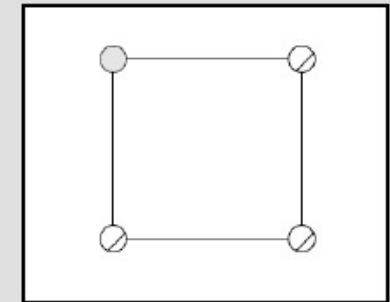
Initial image



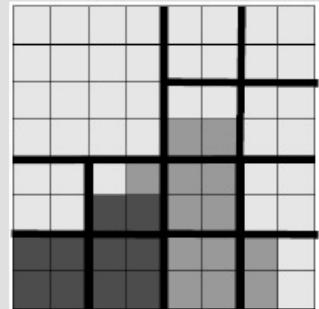
Design of the corresponding adjacency graph



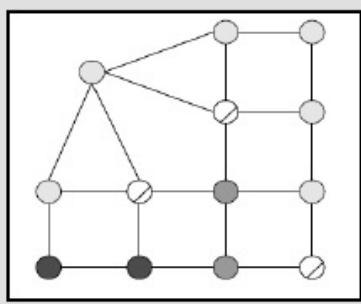
Initial image



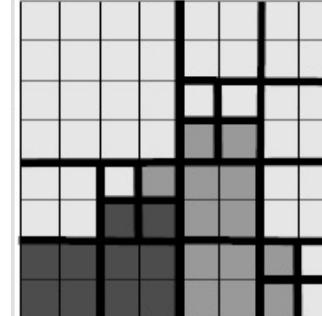
Design of the corresponding adjacency graph



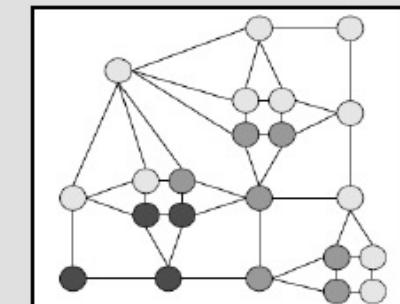
Initial image



Design of the corresponding adjacency graph

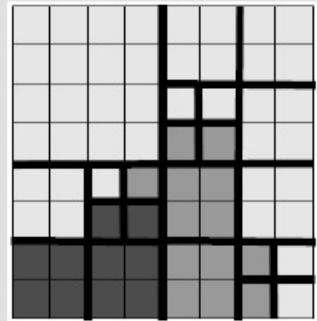


Initial image

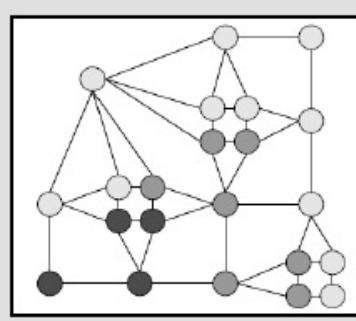


Design of the corresponding adjacency graph

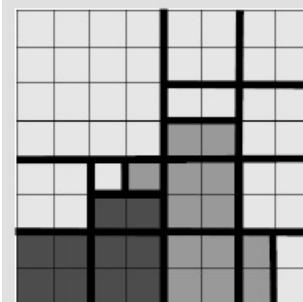
Khai thác lược đồ lân cận



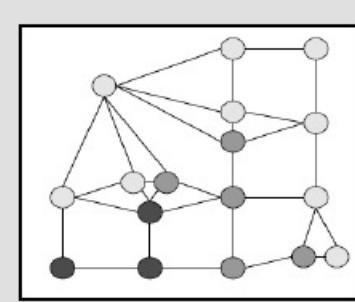
Initial image



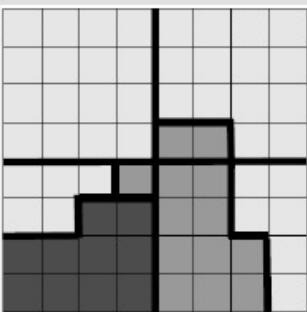
*Building of the corresponding
region-based segmentation*



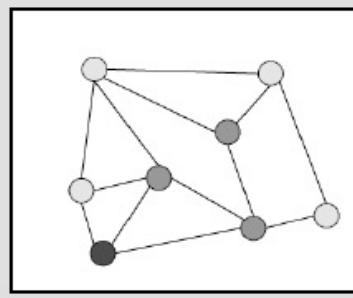
Initial image



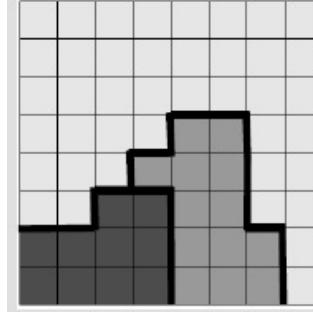
*Building of the corresponding
region-based segmentation*



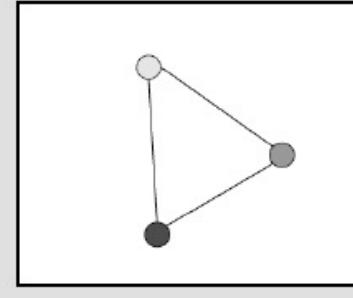
Initial image



*Building of the corresponding
region-based segmentation*



Initial image



*Building of the corresponding
region-based segmentation*

Kết quả - Phát triển vùng



Kết quả - Tách vùng



Kết quả - Tách và hợp vùng

