



XỬ LÝ ẢNH SỐ

Chương 5. Phân đoạn ảnh

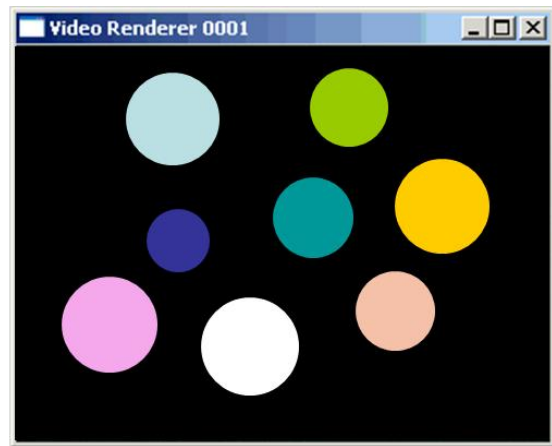
- 5.1 Khái niệm cơ bản
- 5.2 Phát hiện cạnh, đường, điểm trong ảnh
- 5.3 Phân ngưỡng
- 5.4 Phân đoạn bằng phát triển vùng và tách vùng và ghép vùng
- ...

1

Khái niệm cơ bản

Giới thiệu

- Phân đoạn là phân chia các pixel của hình ảnh thành các nhóm có tương quan chặt chẽ với các đối tượng trong hình ảnh.
- Phân đoạn thường là bước đầu tiên trong bất kỳ ứng dụng thị giác máy tính tự động nào.



Cơ bản về phân vùng

Gọi R đại diện cho toàn bộ vùng không gian được chiếm bởi một hình ảnh. Chúng ta có thể xem phân đoạn hình ảnh như một quá trình phân chia R thành n tiểu vùng, R_1, R_2, \dots, R_n sao cho:

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

(b) R_i là tập kết nối với $i = 1, 2, \dots, n$

(c) $R_i \cap R_j = \emptyset$ với mọi i và j , $i \neq j$

(d) $Q(R_i) = \text{True}$ với $i = 1, 2, \dots, n$

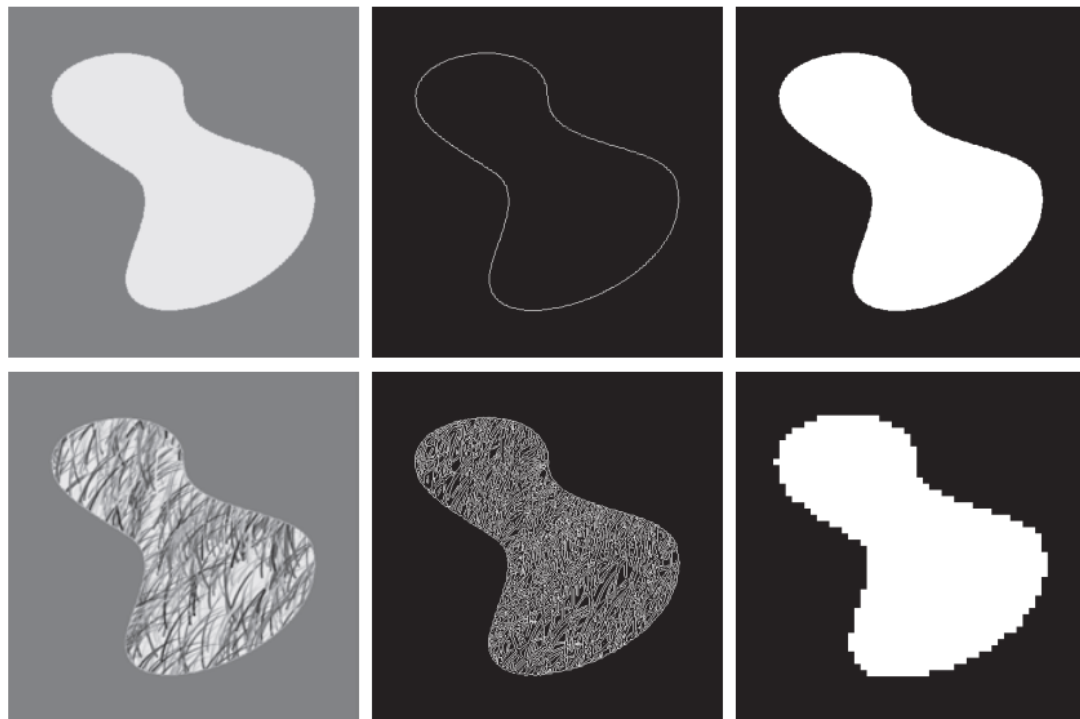
(e) $Q(R_i \cup R_j) = \text{False}$ đối với bất kỳ vùng lân cận R_i và R_j

Giới thiệu

Phân đoạn ảnh dựa vào một trong hai tính chất cơ bản của giá trị cường độ ảnh:

- **Sự gián đoạn:** phân vùng hình ảnh thành các vùng dựa trên sự thay đổi đột ngột về cường độ, chẳng hạn như các cạnh.
- **Tính tương đồng:** phân vùng hình ảnh dựa trên việc phân chia một hình ảnh thành các vùng tương tự nhau theo một tập hợp tiêu chí xác định trước ví dụ phân ngưỡng, phát triển vùng...

Giới thiệu



- a) Ảnh của vùng cường độ không đổi.
- b) Biên dựa trên sự gián đoạn cường độ.
- c) Kết quả của phân đoạn
- d) Ảnh của vùng có kết cấu
- e) Kết quả của tính toán sự gián đoạn cường độ.
- f) Kết quả của phân đoạn dựa trên thuộc tính vùng (tính độ lệch chuẩn của vùng 8 x 8, nó sẽ bằng 0 trên vùng nền)

a	b	c
d	e	f

2

**Phát hiện điểm,
đường, cạnh**

Phát hiện sự gián đoạn

Có ba loại gián đoạn mức xám mà chúng ta có xu hướng tìm kiếm trong ảnh số:

- Điểm (point)
- Đường (lines)
- Cạnh (edge)

Chúng ta thường tìm kiếm sự gián đoạn sử dụng các mặt nạ và tương quan.

Cơ sở lý thuyết

- Trong phần trước, tính trung bình cục bộ làm mịn hình ảnh.
- Trung bình tương tự như tích phân nên một cách trực quan thấy rằng những thay đổi đột ngột và cục bộ về cường độ có thể được phát hiện bằng cách sử dụng đạo hàm.
- Do đó, đạo hàm bậc 1 và bậc 2 đặc biệt thích hợp cho mục đích này.
- Xấp xỉ các đạo hàm số bằng cách nào?

Cơ sở lý thuyết

- Ta có giá trị gần đúng của đạo hàm bậc 1 tại một điểm x tùy ý của một hàm một chiều $f(x)$ bằng cách khai triển hàm $f(x + \Delta x)$ thành chuỗi Taylor về x :

$$\begin{aligned} f(x + \Delta x) &= f(x) + \Delta x \frac{\partial f(x)}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 f(x)}{\partial x^2} + \frac{(\Delta x)^3}{3!} \frac{\partial^3 f(x)}{\partial x^3} + \dots \\ &= \sum_{n=0}^{\infty} \frac{(\Delta x)^n}{n!} \frac{\partial^n f(x)}{\partial x^n} \end{aligned}$$

- Δx là khoảng cách giữa các mẫu của f .
- Trong xử lý ảnh, khoảng cách này được đo bằng đơn vị pixel. Quy ước, $\Delta x = 1$ cho mẫu trước x và $\Delta x = -1$ cho mẫu sau x .
- Khi $\Delta x = 1$, $\Delta x = -1$ ta có:

$$\begin{aligned} f(x + 1) &= f(x) + \frac{\partial f(x)}{\partial x} + \frac{1}{2!} \frac{\partial^2 f(x)}{\partial x^2} + \frac{1}{3!} \frac{\partial^3 f(x)}{\partial x^3} + \dots \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \frac{\partial^n f(x)}{\partial x^n} \end{aligned}$$

$$\begin{aligned} f(x - 1) &= f(x) - \frac{\partial f(x)}{\partial x} + \frac{1}{2!} \frac{\partial^2 f(x)}{\partial x^2} - \frac{1}{3!} \frac{\partial^3 f(x)}{\partial x^3} + \dots \\ &= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{\partial^n f(x)}{\partial x^n} \end{aligned}$$

Cơ sở lý thuyết

- Đối với các **đạo hàm bậc 1**, chúng ta chỉ sử dụng các số hạng tuyến tính và chúng ta có thể tạo ra sự sai khác theo một trong ba cách.

- ☐ Sai khác tiến (*forward difference*)

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

- ☐ Sai khác lùi (*backward difference*)

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

- ☐ Sai khác trung tâm (*central difference*)

$$\frac{\partial f(x)}{\partial x} = f'(x) = \frac{f(x+1) - f(x-1)}{2}$$

Sai khác trung tâm cho sai số nhỏ nhất với cùng một số lượng điểm.

- Đối với **đạo hàm bậc 2** dựa trên sự sai khác trung tâm, $\partial^2 f(x) / \partial^2 x$ được tính như sau:

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

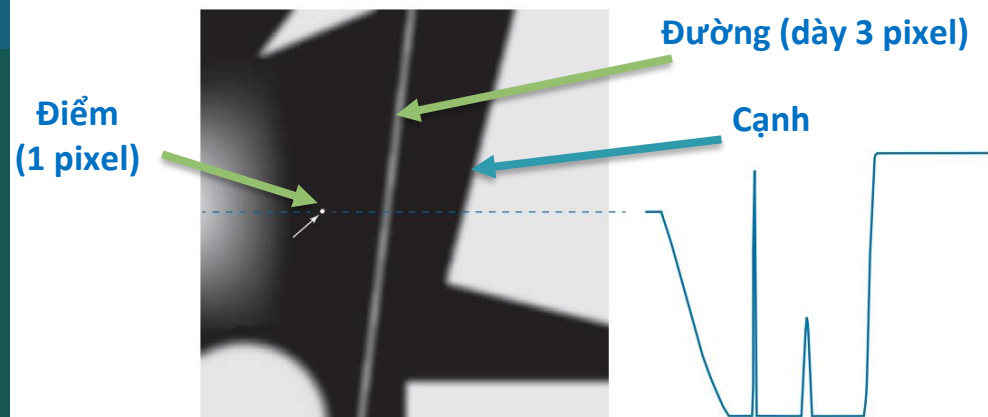
Cơ sở lý thuyết

- Đối với hai biến:

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

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

Cơ sở lý thuyết



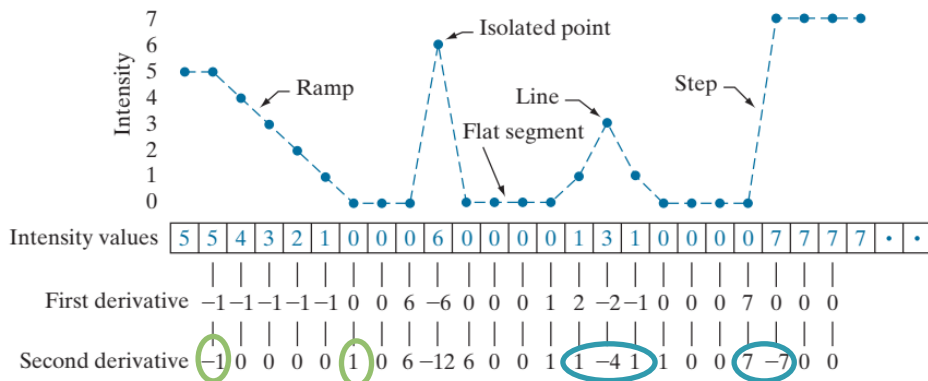
- Đạo hàm bậc 1 là khác 0 khi bắt đầu và dọc theo toàn bộ đoạn dốc cường độ, trong khi đạo hàm bậc 2 chỉ khác 0 khi bắt đầu và kết thúc đoạn dốc cường độ \Rightarrow đạo hàm bậc 1 tạo ra các cạnh "dày", và đạo hàm bậc 2 mỏng hơn nhiều.

- Tại điểm nhiễu, độ lớn của đáp ứng tại điểm này của đạo hàm bậc 2 mạnh hơn nhiều so với đạo hàm bậc 1 \Rightarrow đạo hàm bậc 2 sẽ tăng cường chi tiết tốt hơn (bao gồm cả nhiễu) so với các đạo hàm bậc 1.

- Đường thẳng trong ví dụ này là khá mảnh, vì vậy nó cũng là chi tiết sắc nét, và chúng ta lại thấy rằng đạo hàm cấp 2 có độ lớn lớn hơn.

- Tại cạnh ramp và step, đạo hàm bậc 2 có dấu trái ngược (âm sang dương hoặc dương sang âm) khi nó chuyển tiếp vào và ra khỏi một cạnh. Hiệu ứng "hai cạnh" này là một đặc tính có thể được sử dụng để định vị các cạnh.

- Dấu của đạo hàm bậc 2 cũng được sử dụng để xác định liệu một cạnh chuyển đổi từ sáng sang tối (đạo hàm bậc 2 âm) hay từ tối đến sáng (đạo hàm bậc 2 dương)



Cơ sở lý thuyết

- Cách tiếp cận được lựa chọn để tính toán các đạo hàm bậc nhất và bậc hai tại mọi vị trí pixel trong ảnh là sử dụng tích chập không gian.
- Đáp ứng của bộ lọc tại tâm của bộ lọc là:

$$\begin{aligned} Z &= w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 \\ &= \sum_{k=1}^9 w_k z_k \end{aligned}$$

trong đó z_k là cường độ của pixel có vị trí không gian tương ứng với vị trí của hệ số thứ k của bộ lọc.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Phát hiện điểm

- Dựa trên các kết luận đạt được trong phần trước, chúng ta biết rằng phát hiện điểm nên dựa trên đạo hàm bậc hai, sử dụng công thức Laplacian:

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

- Một điểm được phát hiện tại vị trí (x,y) nếu giá trị tuyệt đối của đáp ứng tại vị trí đó lớn hơn một ngưỡng nào đó:

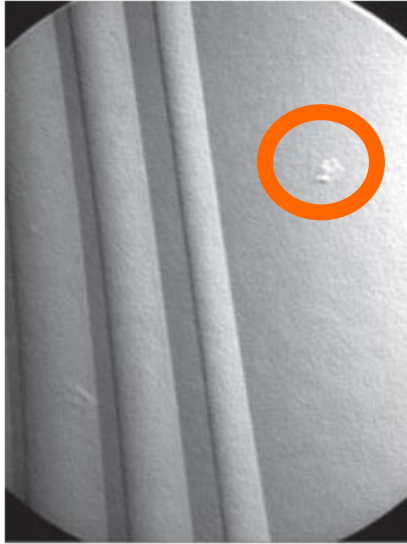
$$g(x, y) = \begin{cases} 1 & \text{if } |Z(x, y)| > T \\ 0 & \text{otherwise} \end{cases}$$

0	1	0	1	1	1	0	-1	0	-1	-1	-1
1	-4	1	1	-8	1	-1	4	-1	-1	8	-1
0	1	0	1	1	1	0	-1	0	-1	-1	-1

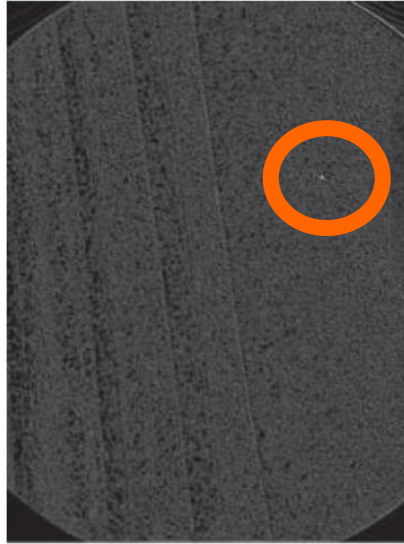
a b c d

FIGURE 3.45 (a) Laplacian kernel used to implement Eq. (3-53). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

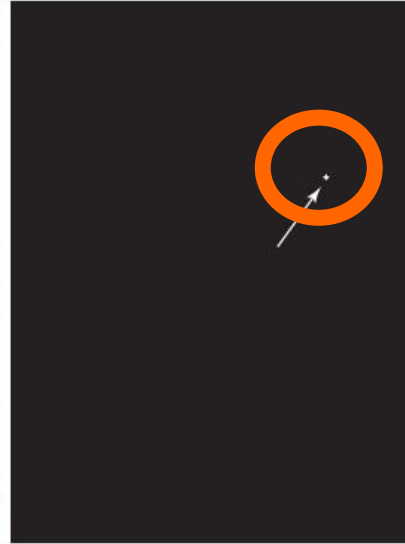
Ví dụ: Phát hiện điểm



Hình ảnh tia X
của cánh
tuabin.



Kết quả sau nhân
chập



Kết quả sau phân
ngưỡng

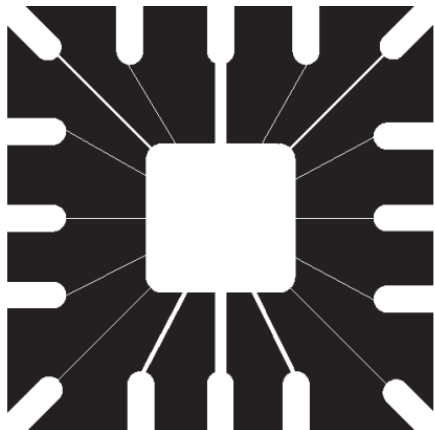
1	1	1
1	-8	1
1	1	1

Phát hiện đường

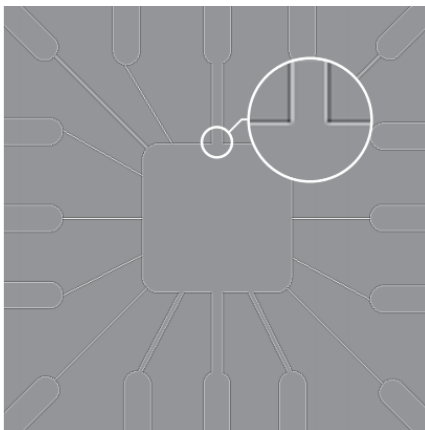
- Ta biết rằng để phát hiện đường, các đạo hàm bậc hai cho kết quả với đáp ứng bộ lọc mạnh hơn và tạo ra các đường mảnh hơn so với đạo hàm bậc 1.
- Vì vậy, có thể sử dụng nhân Laplacian để phát hiện đường, với lưu ý rằng hiệu ứng dòng kép của đạo hàm bậc hai phải được xử lý đúng cách.
- Ví dụ: nhân bộ lọc

1	1	1
1	-8	1
1	1	1

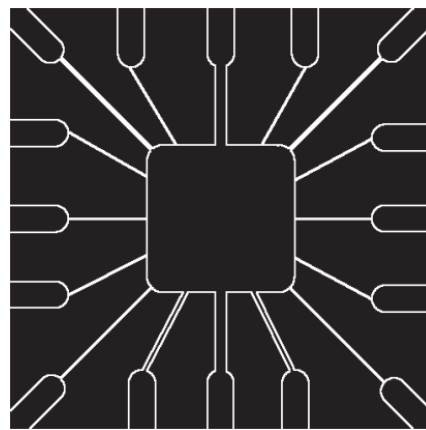
Ví dụ: Phát hiện đường



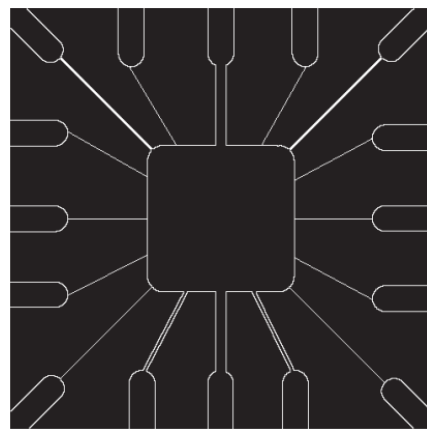
(a) Ảnh gốc



(b) Ảnh Laplacian;
phần phóng đại mô
tả hiệu ứng dòng kép
dương/âm



(c) Giá trị tuyệt
đối của Laplacian



(d) Giá trị dương
của Laplacian

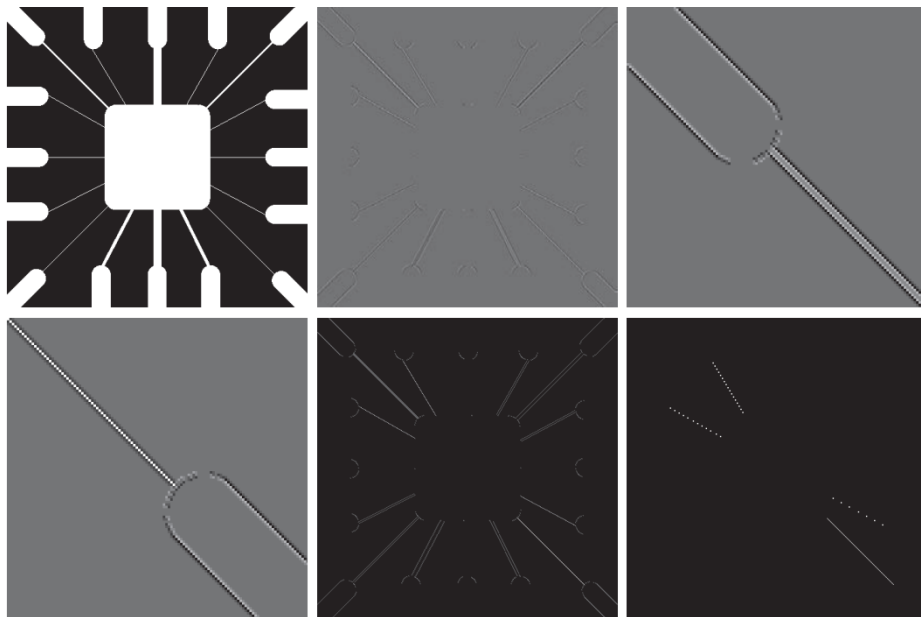
Lưu ý: màu xám trung bình đại diện cho mức 0, màu xám đậm hơn đại diện cho các giá trị âm và màu xám sáng hơn là dương.

Phát hiện đường

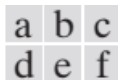
- Các bộ lọc dưới đây sẽ trích xuất các đường có độ dày 1 pixel và chạy theo một hướng cụ thể.

-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
Horizontal			+45°			Vertical			-45°		

Phát hiện đường



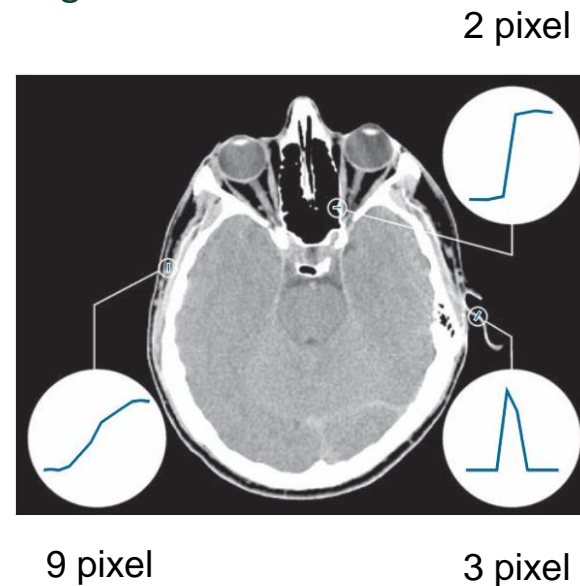
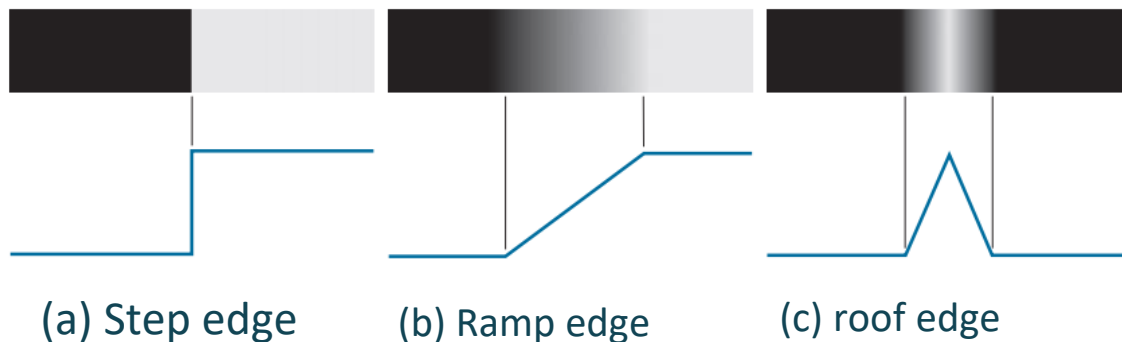
- a) Ảnh gốc
- b) Kết quả của việc lọc hình ảnh với nhân đó.
- c) (c) và (d) cho thấy các phần được phóng to của (b) tương ứng với hai khu vực này.
- e) Các giá trị dương của Hình (b).
- (f) hiển thị màu trắng các điểm có giá trị thỏa mãn điều kiện $g > T$, trong đó g là hình ảnh trong Hình (e).



Tìm tất cả các đường có độ dày một pixel và được định hướng ở $+45^\circ$?
Sử dụng bộ lọc $+45^\circ$

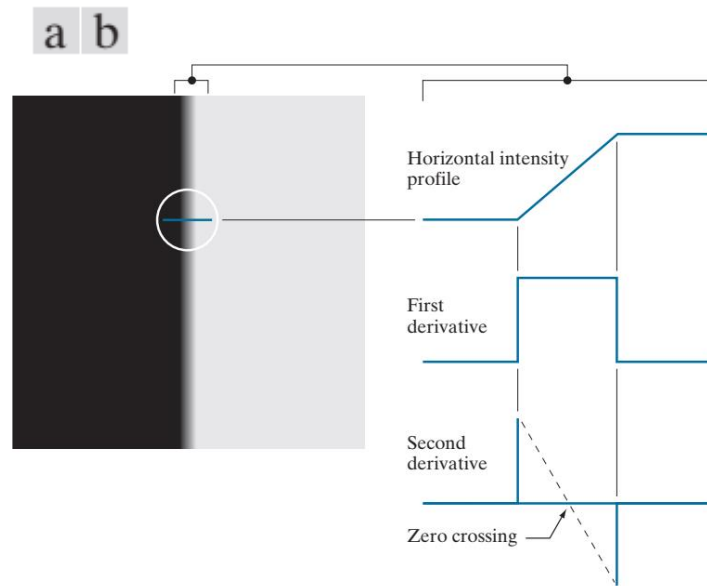
Phát hiện cạnh

- Cạnh là một tập các pixel kết nối nằm trên ranh giới giữa hai vùng.
- Một số mô hình cạnh:



Phát hiện cạnh

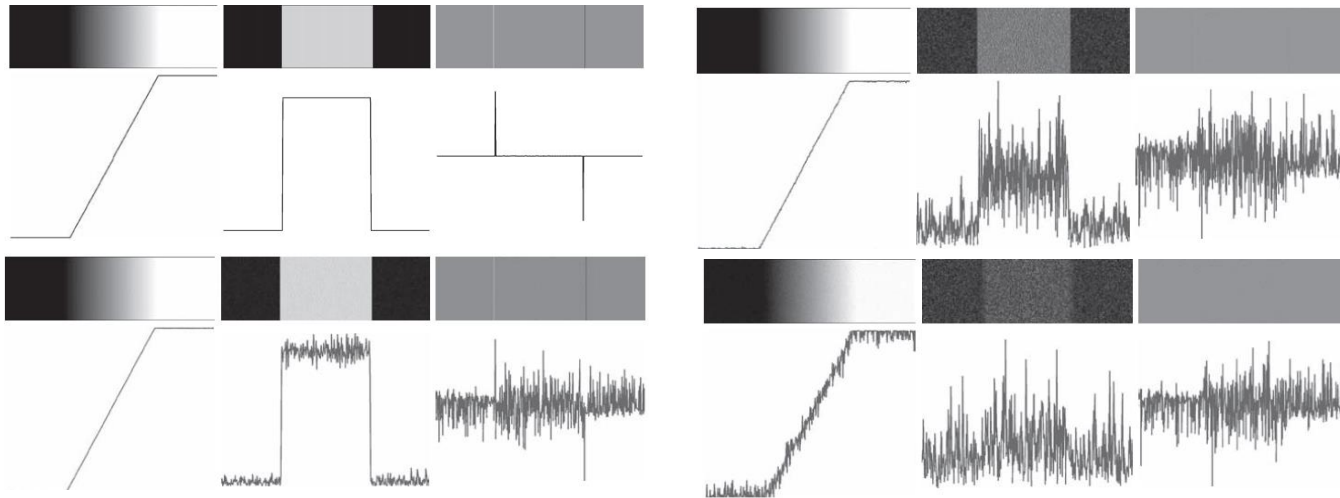
- *Độ lớn của đạo hàm bậc 1* có thể được sử dụng để **phát hiện sự hiện diện** của một cạnh tại một điểm trong hình ảnh.
- *Dấu của đạo hàm bậc 2* có thể được sử dụng để xác định xem **một pixel cạnh nằm trên mặt tối hoặc mặt sáng của cạnh.**
- Với đạo hàm bậc 2:
 - Tạo ra hai giá trị cho mọi cạnh trong một hình ảnh;
 - Các giao cắt 0 có thể được sử dụng để định vị tâm của các cạnh dày.



- Hai vùng có cường độ không đổi được phân tách bởi một cạnh dốc lý tưởng.
- Cận cảnh của cạnh: biểu đồ cường độ theo chiều ngang, đạo hàm bậc 1 và đạo hàm bậc 2.

Đạo hàm và nhiễu

- Các bộ phát hiện cạnh dựa trên đạo hàm luôn cực nhạy với nhiễu.



Ảnh gốc và các ảnh bị nhiễu Gauss trung bình 0 và độ lệch chuẩn của các mức cường độ tương ứng là 0,1; 1 và 10

Các bước phát hiện cạnh

1. *Làm mịn hình ảnh* để giảm nhiễu.
2. *Phát hiện các điểm cạnh*. Như đã đề cập trước đó, đây là một bước cục bộ để trích xuất từ một hình ảnh tất cả các điểm là ứng cử viên điểm cạnh tiềm năng.
3. *Cục bộ hóa cạnh*. Mục tiêu của bước này là chọn từ ứng viên chỉ các điểm là thành viên của tập hợp các điểm bao gồm một cạnh.

Các bộ lọc phát hiện cạnh phổ biến

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0	0	-1
0	1	1	0

Roberts

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

Prewitt

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

Sobel

Ví dụ bộ lọc phát hiện cạnh

Ảnh gốc



Thành phần gradient
ngang $|g_x|$



Thành phần gradient
dọc $|g_y|$

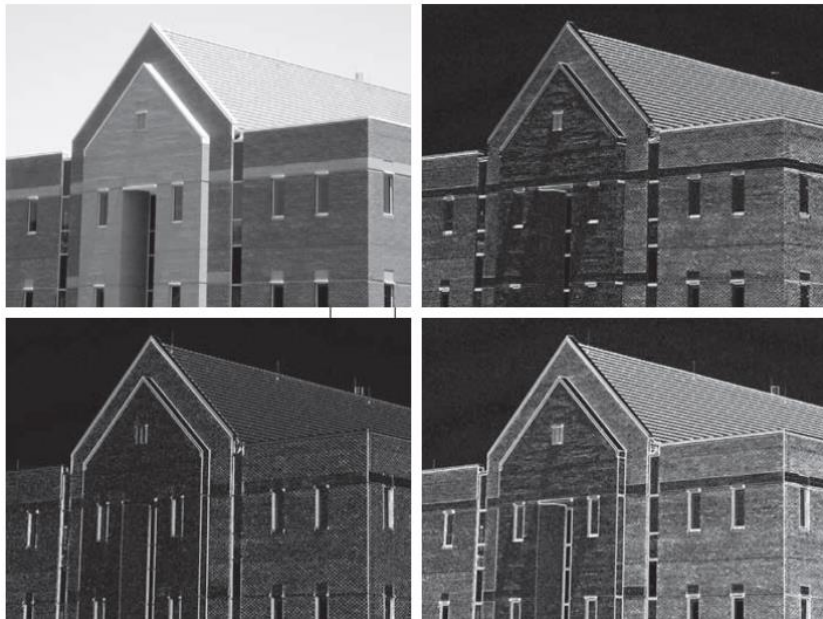


Ảnh gradient $|g_x| + |g_y|$

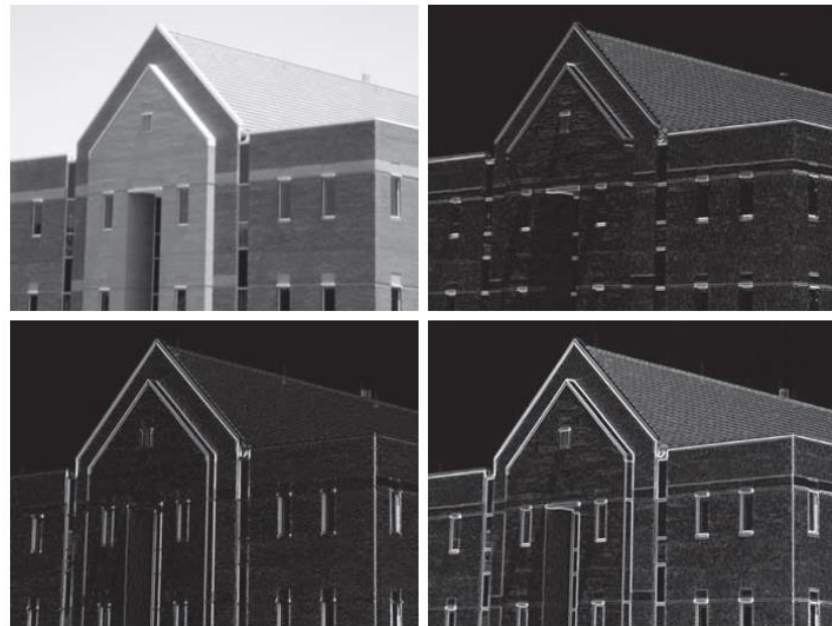
Các vấn đề đối với phát hiện cạnh

- Thông thường, vấn đề nảy sinh với phát hiện cạnh là có quá nhiều chi tiết (ví dụ, phần gạch trong ví dụ trước).
- Một cách để giải quyết vấn đề này là làm mịn ảnh trước khi phát hiện cạnh.

Các vấn đề đối với phát hiện cạnh



Ảnh không làm mịn trước khi
phát hiện cạnh



Ảnh có làm mịn trước khi
phát hiện cạnh

Bộ lọc phát hiện cạnh Laplacian

- Chúng ta đã gặp bộ lọc Laplacian dựa trên đạo hàm bậc 2.
- Laplacian thường không được sử dụng một mình vì nó quá nhạy cảm với nhiễu.
- Vì vậy để sử dụng để phát hiện cạnh, Laplacian thường được kết hợp với bộ lọc Gaussian làm mịn.

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Bộ lọc Laplacian of Gaussian (LoG)

- Bộ lọc Laplacian of Gaussian (LoG) sử dụng bộ lọc Gaussian để loại bỏ nhiễu và Laplacian để phát hiện cạnh

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

$$\nabla^2 G(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

a b
c d

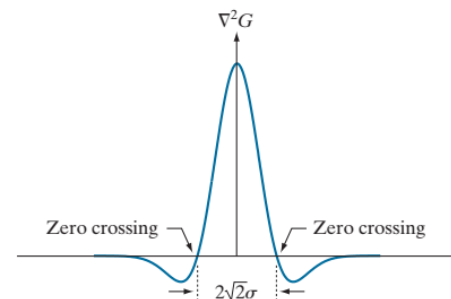
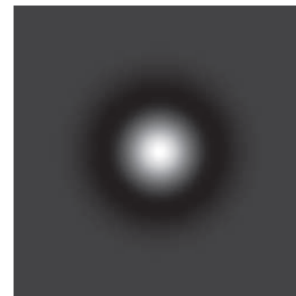
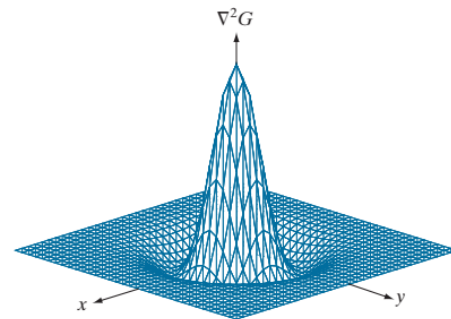
FIGURE 10.21

(a) 3-D plot of the *negative* of the LoG.

(b) Negative of the LoG displayed as an image.

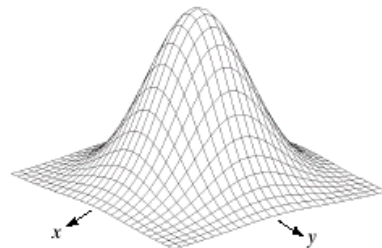
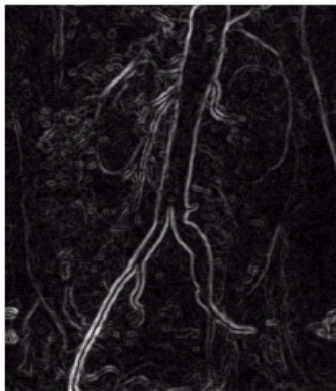
(c) Cross section of (a) showing zero crossings.

(d) 5×5 kernel approximation to the shape in (a). The negative of this kernel would be used in practice.

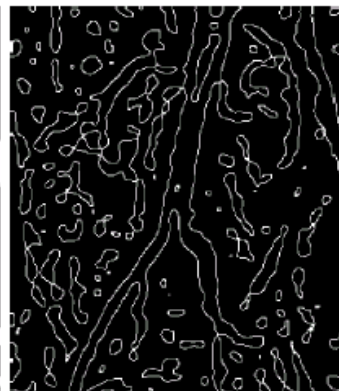
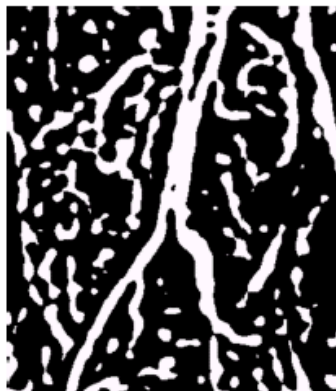


0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Bộ lọc Laplacian of Gaussian (LoG)



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



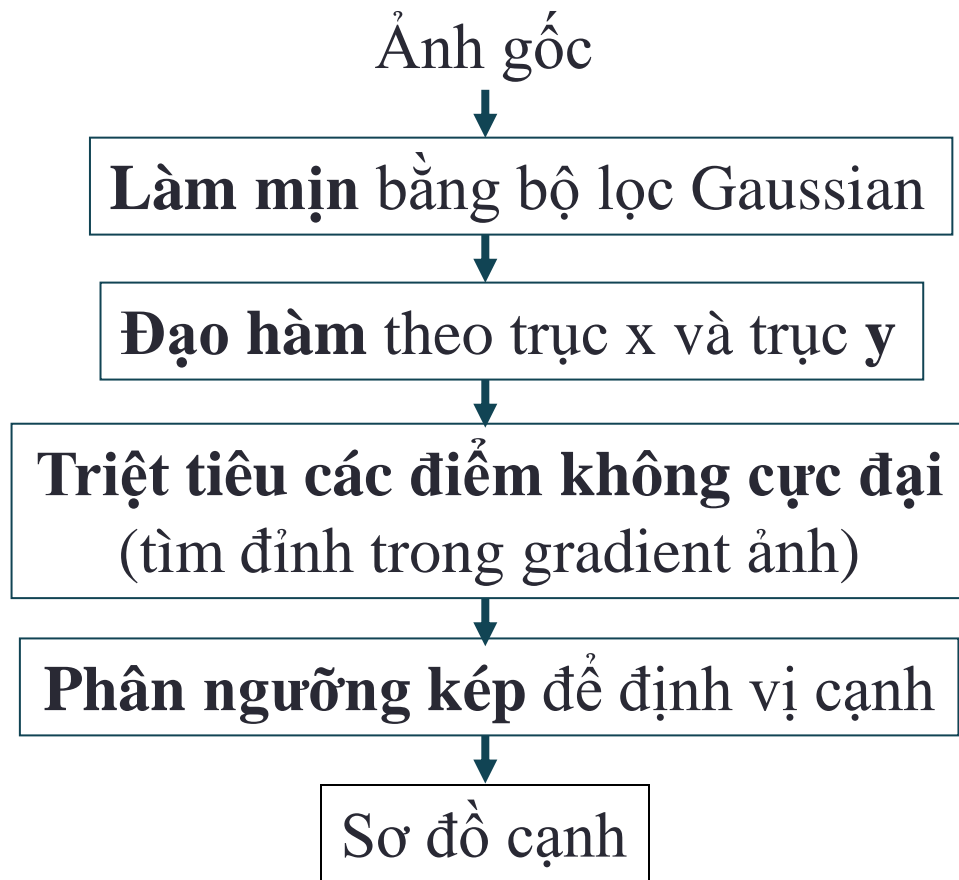
3

Phát hiện cạnh tối ưu Canny

Bộ lọc phát hiện cạnh 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 đó có nghĩa là bộ phát hiện biên không được xác định nhiều pixel biên khi chỉ có duy nhất một pixel biên tồn tại.

Bộ lọc phát hiện cạnh Canny



Các bước phát hiện cạnh 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 connectivity analysis* : Phân ngưỡng kép và phân tích kết nối để tìm và liên kết cạnh.

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

- Sự rõ ràng của cạnh 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

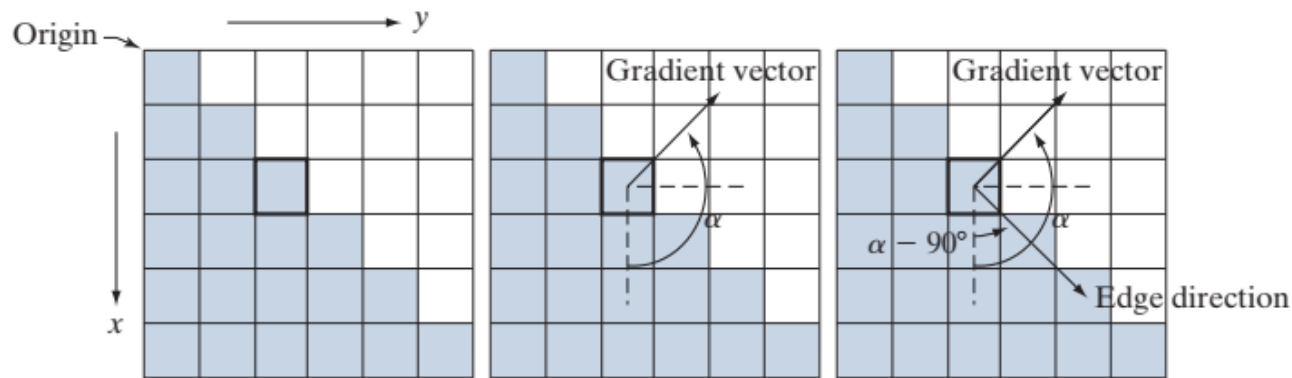
Gx

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

Gy

- $|G| = |Gx| + |Gy|$
- Hướng của cạnh: $\theta = \tan^{-1}\left(\frac{Gy}{Gx}\right)$

Bước 2. Tính gradient



$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix} \quad \|\nabla f\| = 2\sqrt{2} \quad \alpha = \tan^{-1}(g_y/g_x) = -45^\circ$$

Hướng của cạnh tại một điểm vuông góc với vector gradient. Vì vậy **vector gradient** còn gọi là pháp tuyến của cạnh (*edge normal*)

Bước 2 (tiếp)

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

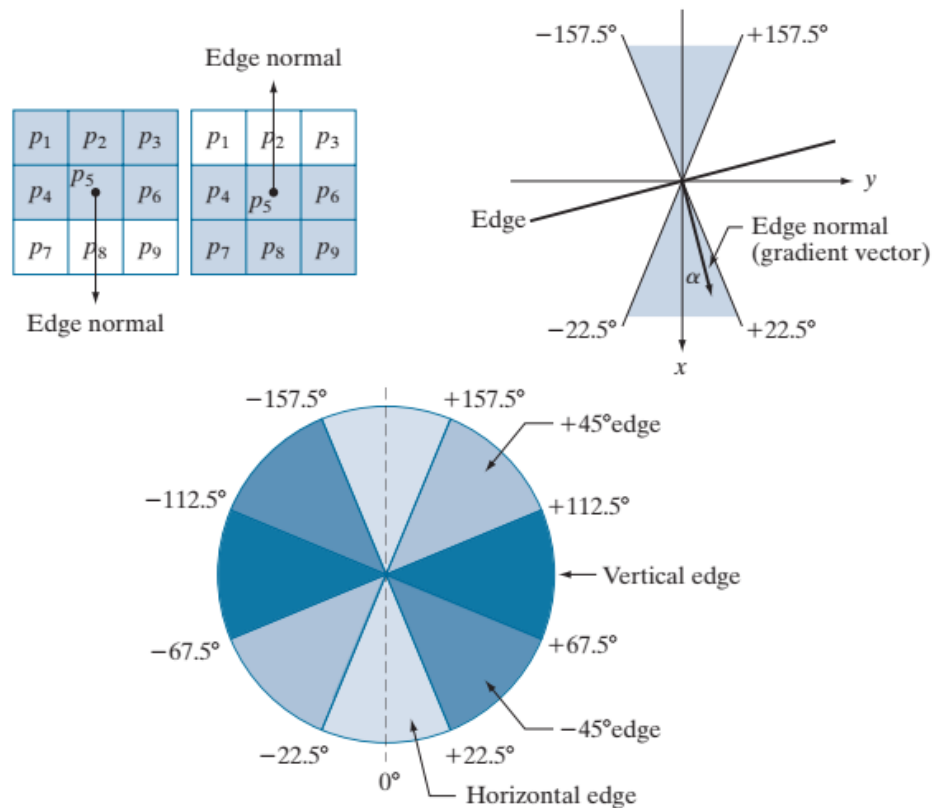
x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

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 cạnh được định hướng vào một trong 4 hướng này tùy thuộc vào hướng của cạnh gần cái nào nhất.

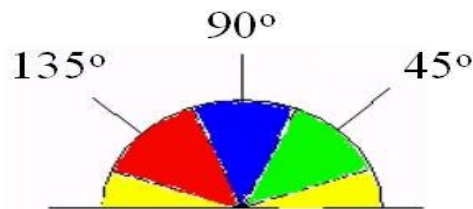
Bước 2 (tiếp)

- Nếu pháp tuyến cạnh (hướng gradient) nằm trong phạm vi từ $-22,5^\circ$ đến $22,5^\circ$ hoặc từ $-157,5^\circ$ đến $157,5^\circ$, ta gọi cạnh là cạnh ngang.



Bước 2 (tiếp)

- Nếu hướng cạnh rơi vào trong **dải màu vàng** (0 đến 22.5 & 157.5 đến 180 độ) thì được coi là 0 độ.
- Nếu hướng cạnh rơi vào dải **màu xanh lá cây** (22.5 đến 67.5 độ) được coi là 45 độ.
- Nếu hướng cạnh rơi vào dải **màu xanh da trời** (67.5 đến 112.5 độ) thì được coi là 90 độ
- Nếu hướng cạnh 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 cạnh.
- 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 cạnh theo hướng **bắc- nam**), điểm đó được coi là nằm trên cạnh 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 cạnh theo hướng **đông- tây**), điểm đó được coi là nằm trên cạnh 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 cạnh theo hướng **đông bắc- tây nam**), điểm đó được coi là nằm trên cạnh 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 cạnh theo hướng **tây bắc- đông nam**), điểm đó được coi là nằm trên cạnh nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng **đông bắc và tây nam**.

Bước 4: 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 **T1 cao và T2 thấp**
(thường chọn T1: T2 trong dải 2:1 đến 3:1)

Bước 4: Phân ngưỡng kép

- Nếu pixel có giá trị **lớn hơn T1** sẽ được coi là điểm cạnh ngay lập tức.
- Các pixel có giá **trị nhỏ hơn T2** sẽ không được coi là điểm cạnh.
- Pixel có giá trị nằm giữa T2 và T1 sẽ được xem xét tiếp: nếu pixel này có kết nối với một trong số các điểm cạnh khác thì sẽ được coi là điểm cạnh, ngược lại sẽ không coi là điểm cạnh.



Ví dụ về phương pháp Canny



(a) Original



(b) Smoothed



(c) Gradient magnitudes



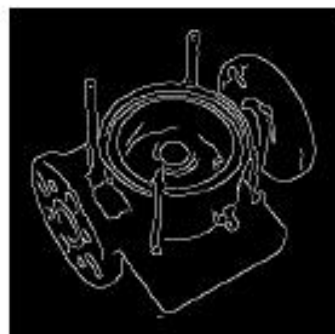
(d) Edges after non-maximum suppression



(e) Double thresholding



(f) Edge tracking by hysteresis



(g) Final output

Ví dụ về phương pháp Canny

Ảnh gốc



Ảnh gradient
sau làm mịn



Ảnh LoG



Phương pháp
Canny



Ví dụ về phương pháp Canny

Ảnh gốc



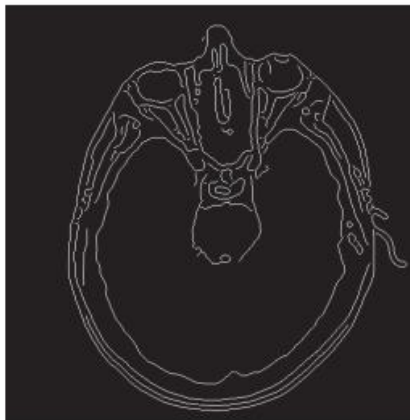
Ảnh gradient
sau làm mịn



Ảnh LoG



Phương
pháp Canny



3

PHÂN NGƯỜI

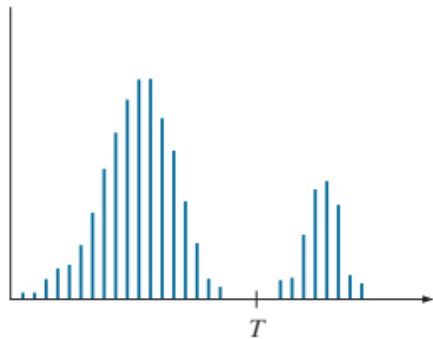
Nội dung

- - Phân ngưỡng là gì?
 - Phân ngưỡng đơn giản
 - Phân ngưỡng thích ứng

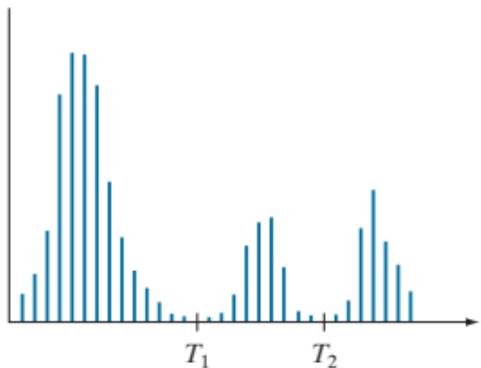
Phân ngưỡng cường độ

- Giả sử ảnh $f(x, y)$ bao gồm các đối tượng sáng trên nền tối, với pixel đối tượng và pixel nền có các giá trị cường độ được nhóm thành hai chế độ nổi trội.
- Để tách đối tượng khỏi nền là chọn một ngưỡng, T , phân tách các chế độ này.

$$g(x, y) = \begin{cases} a & \text{if } f(x, y) > T_2 \\ b & \text{if } T_1 < f(x, y) \leq T_2 \\ c & \text{if } f(x, y) \leq T_1 \end{cases}$$



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

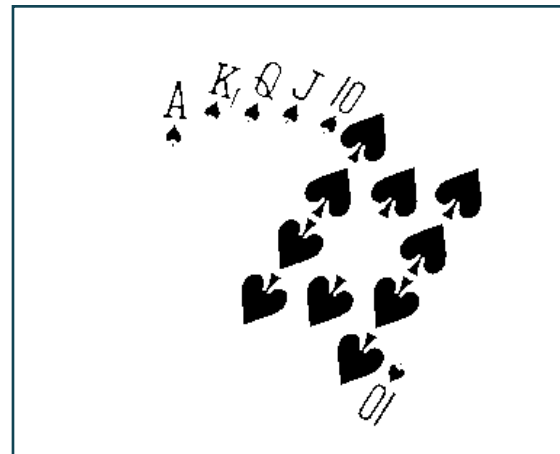


Ví dụ về phân ngưỡng

- Giả sử muốn đọc thông tin trong các quân bài.



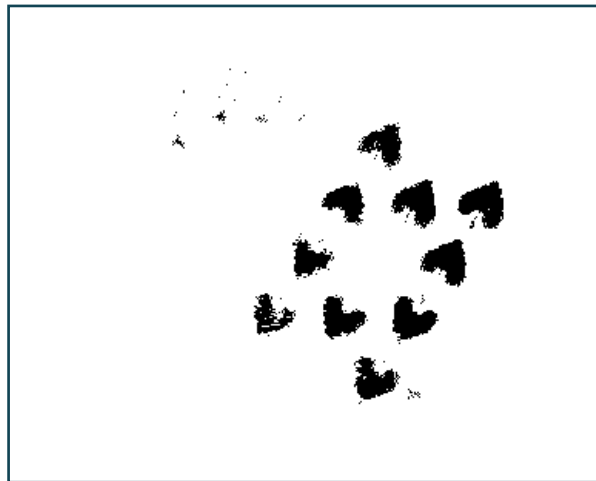
Ảnh gốc



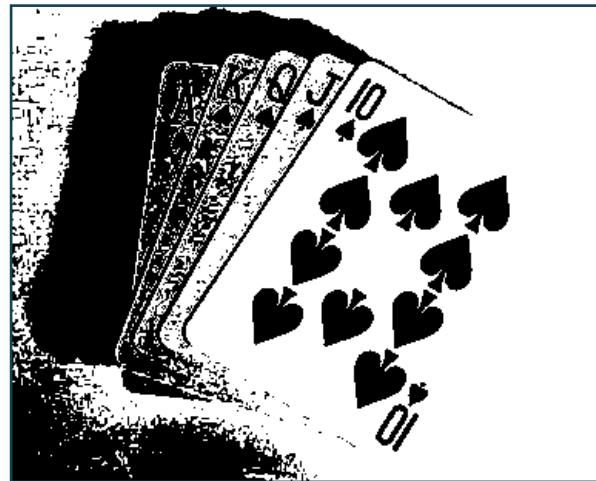
Ảnh phân ngưỡng

Vấn đề khi phân ngưỡng...

- Nếu sử dụng ngưỡng sai, kết quả sẽ rất tệ!



Ngưỡng quá thấp



Ngưỡng quá cao

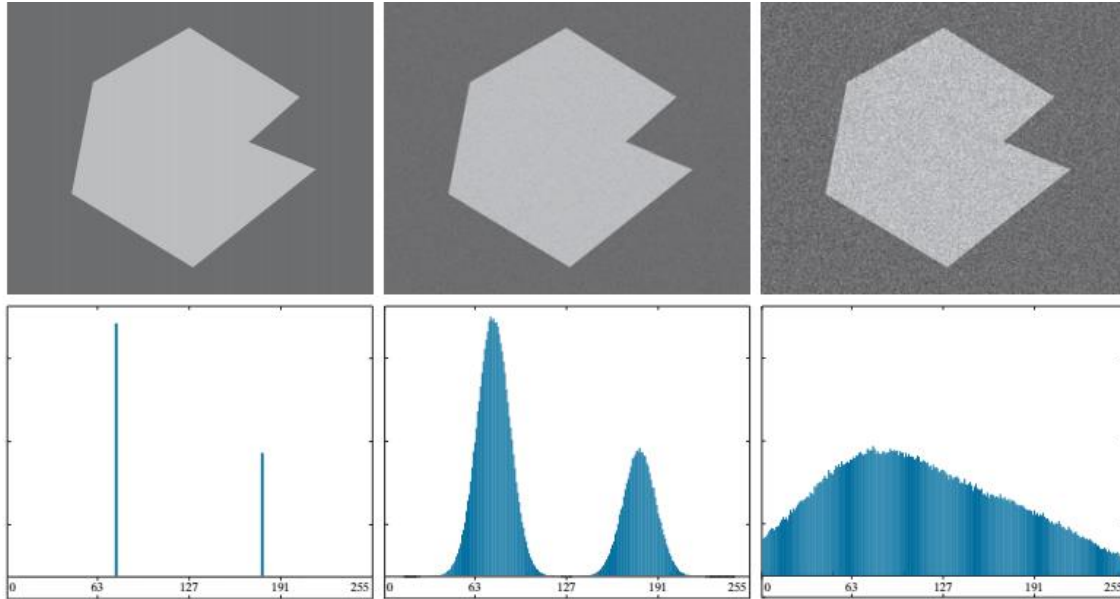
Phân loại ngưỡng

- - Khi T là một hằng số có thể áp dụng cho toàn bộ hình ảnh, ngưỡng này được gọi là **ngưỡng toàn cục (global thresholding)**.
- Khi giá trị của T thay đổi trên một hình ảnh, chúng ta có **ngưỡng thay đổi (variable thresholding)**. Các thuật ngữ phân ngưỡng cục bộ hoặc địa phương đôi khi được sử dụng để biểu thị ngưỡng thay đổi trong đó giá trị của T tại bất kỳ điểm (x,y) nào trong một hình ảnh phụ thuộc vào thuộc tính của vùng lân cận của (x,y) ví dụ: cường độ trung bình của các pixel trong vùng lân cận).
- Nếu T phụ thuộc vào chính các tọa độ không gian (x,y) thì ngưỡng thay đổi thường được gọi là **ngưỡng động hoặc ngưỡng thích ứng**.

Các yếu tố ảnh hưởng đến phân ngưỡng

- (1) Khoảng cách giữa các đỉnh (các đỉnh càng xa nhau thì càng có cơ hội tốt hơn để tách các chế độ);
- (2) Nhiễu trong ảnh;
- (3) Kích thước tương đối của các đối tượng và nền;
- (4) Tính đồng nhất của nguồn chiếu sáng;
- (5) Tính đồng nhất của thuộc tính phản xạ của ảnh.

Vai trò của nhiễu trong phân ngưỡng



Ảnh không nhiễu

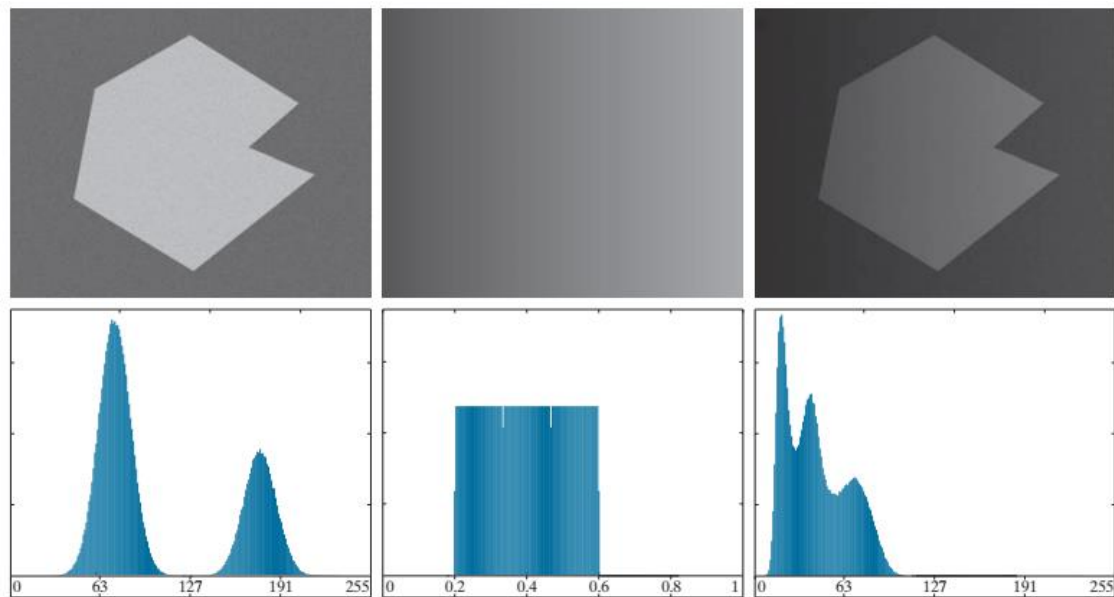
Ảnh có nhiễu
Gaussian

$m = 0; \sigma = 10$

Ảnh có nhiễu
Gaussian

$m = 0; \sigma = 50$

Vai trò của độ chói và phản xạ trong phân ngưỡng hình ảnh



Ảnh nhiễu

Hàm cường độ
dốc thoải

Ảnh tích

Phân ngưỡng toàn cục cơ bản

- - Khi phân bố cường độ của các pixel đối tượng và pixel nền đủ khác biệt, có thể sử dụng một ngưỡng duy nhất (toàn cục) áp dụng cho toàn bộ hình ảnh.
- Trong hầu hết các ứng dụng, khi sử dụng ngưỡng toàn cục thì vẫn cần một thuật toán có khả năng ước tính giá trị ngưỡng cho mỗi hình ảnh.

Thuật toán lặp (thuật toán đẳng liệu)

- 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 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 lặp (thuật toán đẳng liệu)

- 4. Tính giá trị ngưỡng mới:

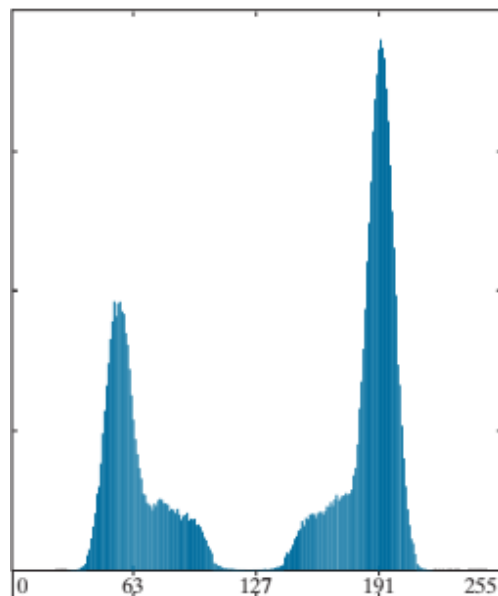
$$t_{k+1} = \frac{\mu_1(t_k) + \mu_2(t_k)}{2} = \frac{1}{2} \left(\frac{\sum_{z=0}^{t_k} z \cdot p(z)}{\sum_{z=0}^{t_k} p(z)} + \frac{\sum_{z=t_k+1}^{L-1} z \cdot p(z)}{\sum_{z=t_k+1}^{L-1} p(z)} \right)$$

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 $|t_{k+1} - t_k| < \Delta T$

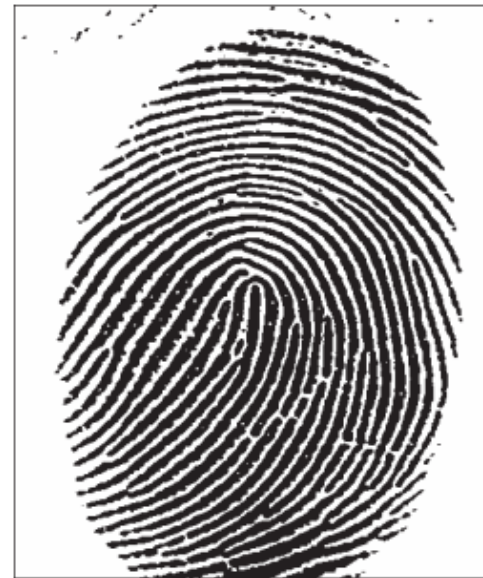
Thuật toán này làm việc hiệu quả trong việc tìm giá trị ngưỡng T khi ảnh có lược đồ xám thích hợp.



Vân tay bị nhiễu



Biểu đồ xám



Sử dụng ngưỡng từ
thuật toán lặp ($T=125,4$
sau 3 lần lặp, $\Delta T=0$)

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 $T_0=5$:

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

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

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
t0=3						
t1=	2.574					
t2=	2.438					
t3=	2.438					
Vậy ngưỡng t = 2						

Phân ngưỡng toàn cục tối ưu sử dụng phương pháp OTSU

- - Phân ngưỡng là bài toán giảm thiểu lỗi trung bình phát sinh khi gán pixel vào một trong các lớp.
 - Phương pháp OTSU tối ưu theo nghĩa là nó tối đa hóa phương sai giữa các lớp và các phép tính được thực hiện hoàn toàn trên biểu đồ của ảnh.
 - Ý tưởng cơ bản là các lớp được phân ngưỡng thích hợp phải phân biệt tương ứng với các giá trị cường độ của các pixel của chúng và ngược lại, nghĩa là **ngưỡng mang lại sự tách biệt tốt nhất giữa các lớp về giá trị cường độ sẽ là ngưỡng tốt nhất (tối ưu).**

Phân ngưỡng toàn cục tối ưu sử dụng phương pháp OTSU

- 1. Tính toán biểu đồ chuẩn hóa của hình ảnh đầu vào. Biểu thị các thành phần của biểu đồ bởi $p_i, i = 0, 1, \dots, L - 1$
- 2. Tính tổng tích lũy $P_1(k), k = 0, 1, \dots, L - 1$, sử dụng phương trình

$$P_1(k) = \sum_{i=0}^k p_i$$

- 3. Tính giá trị trung bình tích lũy, $m(k), k = 0, 1, \dots, L - 1$ sử dụng:

$$m(k) = \sum_{i=0}^k ip_i$$

- 4. Tính giá trị trung bình toàn cục, $m(G)$, sử dụng:

$$m(G) = \sum_{i=0}^{L-1} ip_i$$

Phân ngưỡng toàn cục tối ưu sử dụng phương pháp OTSU

- 5. Tính toán phương sai giữa các lớp, $\sigma_B^2(k)$, $k = 0, 1, \dots, L - 1$ bằng cách sử dụng phương trình:

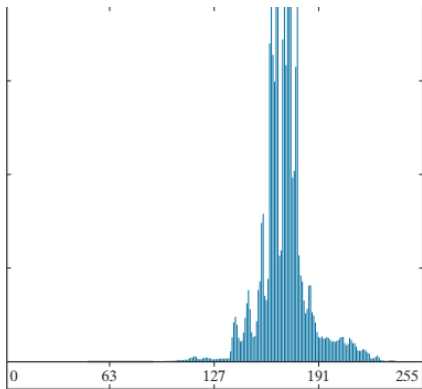
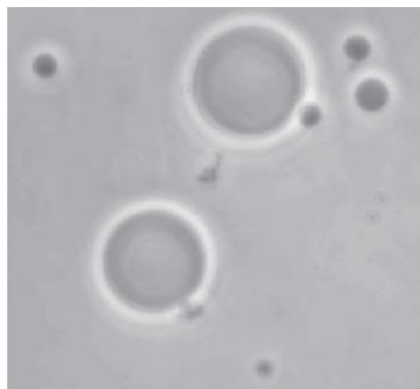
$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

6. Lấy ngưỡng Otsu, k^* , là giá trị của k mà $\sigma_B^2(k)$ đạt giá trị lớn nhất. Nếu cực đại không phải là duy nhất, lấy k^* bằng cách lấy trung bình các giá trị của k tương ứng với các cực đại khác nhau được phát hiện.

7. Tính toán phương sai tổng thể, σ_G^2 , sử dụng $\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$, và sau đó tính thước đo khả năng phân tách, η^* , bằng cách đánh giá phương trình

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \quad \text{với } k = k^*.$$

Ví dụ



Ảnh gốc

Biểu đồ xám

Kết quả phân đoạn
sử dụng ngưỡng
bằng thuật toán lặp

Kết quả phân đoạn
sử dụng ngưỡng
bằng thuật toán Otsu

Bài tập

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

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

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 $\sigma_B^2(k) = 1.68$

Làm mịn hình ảnh để cải thiện phân ngưỡng toàn cục

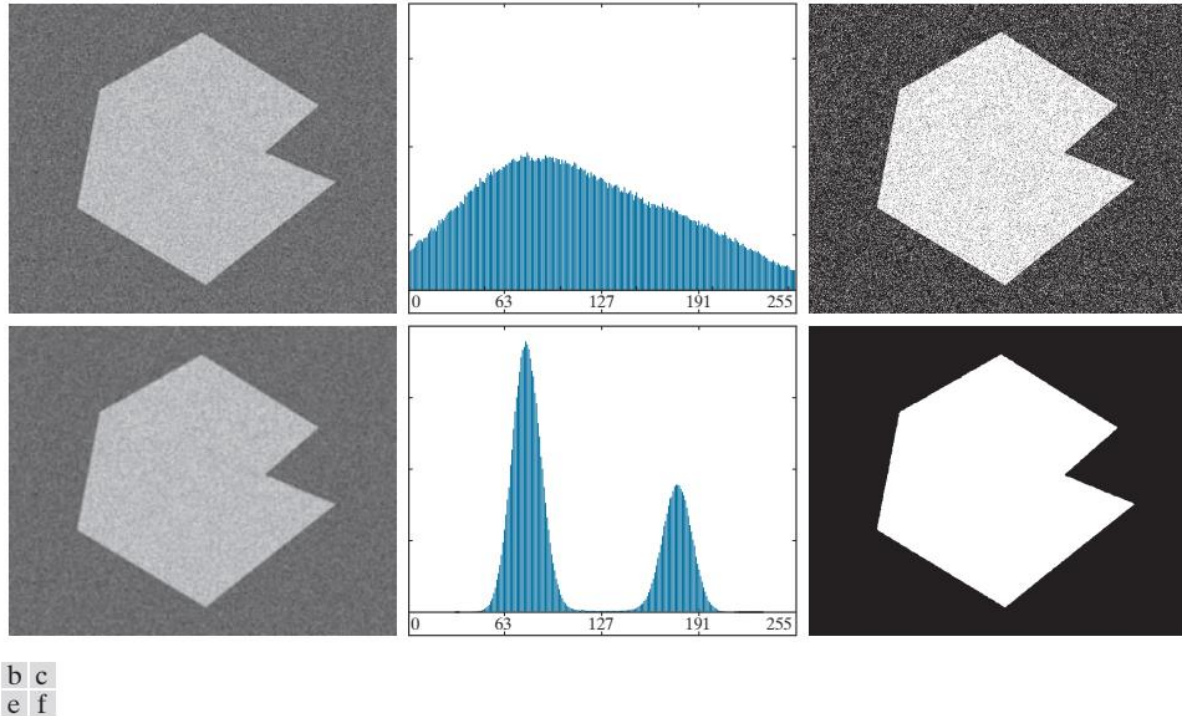
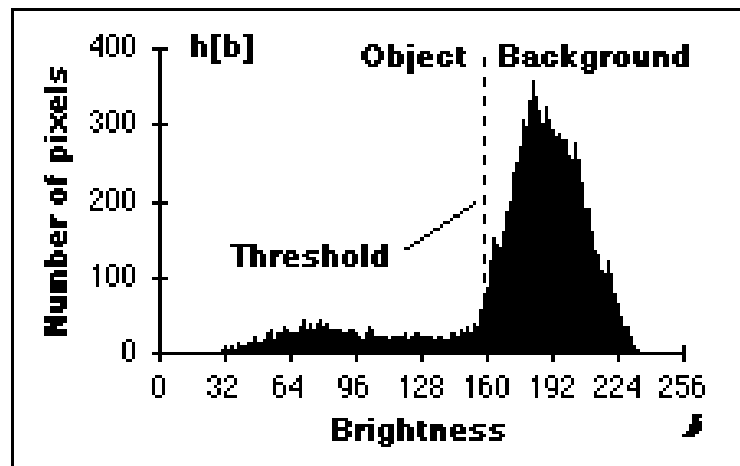


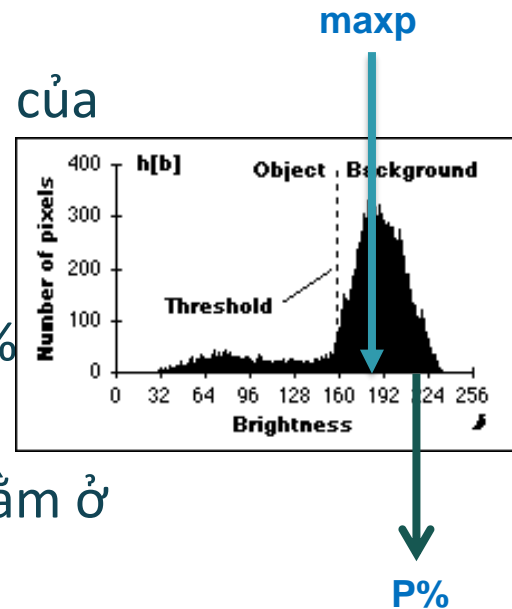
FIGURE 10.37 (a) Noisy image from Fig. 10.33(c) and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging kernel and (e) its histogram. (f) Result of thresholding using Otsu's method.

Tìm ngưỡng bằng thuật toán đối xứng nền

- Giả định trong lược đồ xám có đỉnh nằm ở phần nền và đối xứng quanh giá trị cực đại đó.
- Giá trị đỉnh ($maxp$) 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 $P(a) = p\%$).



- Trong hình trước, các pixel đối tượng nằm ở bên trái của đỉnh nền (đỉnh nền nằm ở mức sáng $\max(p) = 183$).
- 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:
- $\theta = \max p - (p\% - \max p)$
- Ngưỡng: $\theta = 183 - (216 - 183) = 150$



4

PHÂN ĐOẠN BẰNG PHÁT TRIỂN VÙNG VÀ PHÂN ĐOẠN BẰNG TÁCH & HỢP VÙNG

Giới thiệu

Để phân đoạn ảnh chúng ta đã có hai cách tiếp cận:

- Phân đoạn bằng cách tìm ranh giới giữa các vùng dựa trên sự không liên tục về mức cường độ.
- Phân đoạn được thực hiện thông qua các ngưỡng dựa trên sự phân bố các thuộc tính pixel.

Phần này sẽ thảo luận về các kỹ thuật phân đoạn để tìm vùng trực tiếp.

Thuật toán phát triển vùng

- (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 mới nào được thêm vào.

Các vấn đề trong phát triển vùng

- Việc chọn tập hợp một hoặc nhiều điểm xuất phát thường có thể dựa trên bản chất của vấn đề
- Việc lựa chọn các tiêu chí tương tự không chỉ phụ thuộc vào vấn đề đang xem xét mà còn phụ thuộc vào loại dữ liệu hình ảnh có sẵn.
 - Ví dụ, việc phân tích hình ảnh vệ tinh sử dụng đất phụ thuộc nhiều vào việc sử dụng màu sắc. Khi hình ảnh là đơn sắc, phân tích vùng phải được thực hiện với một bộ mô tả dựa trên các mức cường độ và các thuộc tính không gian thuộc tính (chẳng hạn như mô men hoặc kết cấu).
 - Chỉ riêng bộ mô tả có thể mang lại kết quả sai lệch nếu các thuộc tính kết nối không được sử dụng trong quá trình phát triển vùng. Ví dụ: hình dung một sự sắp xếp ngẫu nhiên của pixel có ba giá trị cường độ khác nhau. Nhóm các pixel có cùng giá trị cường độ để tạo thành một "vùng", mà không cần chú ý đến kết nối, sẽ mang lại kết quả phân đoạn là vô nghĩa trong ngữ cảnh này.

Các vấn đề trong phát triển vùng

- Một vấn đề khác trong khu vực đang phát triển là việc **xây dựng quy tắc dừng**. Phát triển vùng sẽ dừng lại khi không còn pixel nào đáp ứng các tiêu chí để đưa vào vùng.
 - Các tiêu chí như giá trị cường độ, kết cấu và màu sắc có bản chất cục bộ và không tính đến “lịch sử” của phát triển vùng. Các tiêu chí bổ sung có thể tăng sức mạnh của thuật toán phát triển theo vùng sử dụng khái niệm về kích thước, độ giống giữa pixel ứng viên và pixel đã phát triển cho đến nay (chẳng hạn như so sánh cường độ của một ứng cử viên và cường độ trung bình của vùng phát triển), và hình dạng của vùng đã được phát triển.

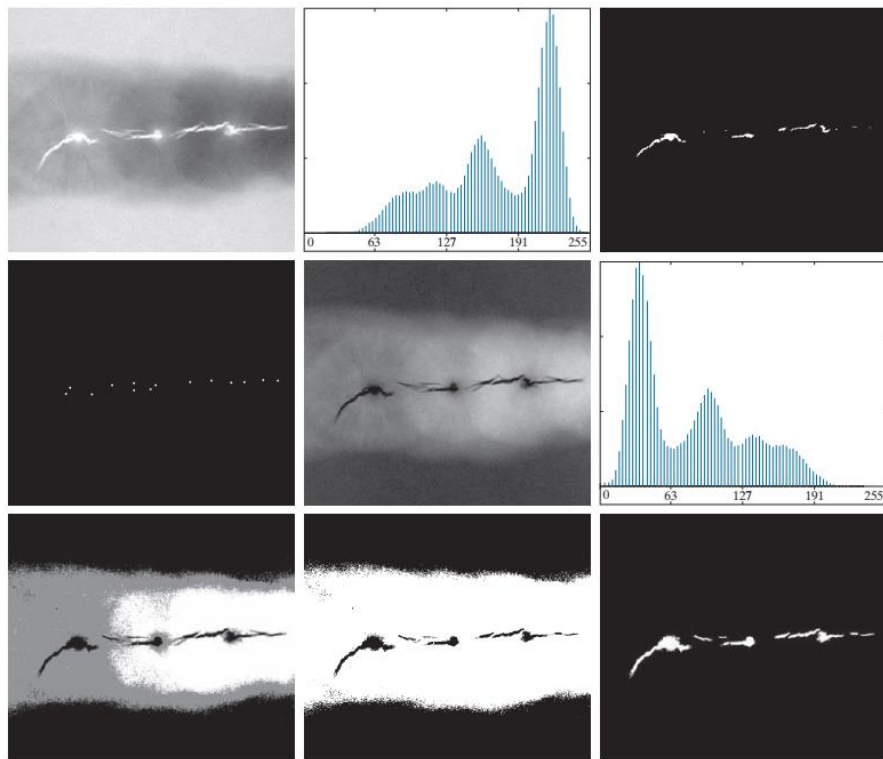
Thuật toán phát triển vùng

$f(x, y)$ biểu thị một ảnh đầu vào; $S(x, y)$ biểu thị một mảng “hạt giống” chứa 1 tại các vị trí của điểm hạt giống và 0 ở những nơi khác; và Q biểu thị một mệnh đề được áp dụng tại mỗi vị trí (x, y) . Mảng f và S có cùng kích thước. Một thuật toán phát triển khu vực cơ bản dựa trên kết nối 8 có thể được phát biểu như sau:

Thuật toán phát triển vùng

1. Tìm tất cả các thành phần được kết nối trong $S(x, y)$ và giảm từng thành phần kết nối xuống một pixel; gán nhãn tất cả các pixel như vậy được tìm thấy là 1. Tất cả các pixel khác trong S được gán nhãn 0.
2. Tạo ảnh f_Q sao cho tại mỗi điểm (x, y) , $f_Q(x, y) = 1$ nếu ảnh đầu vào thỏa mãn một mệnh đề cho trước, Q , tại các tọa độ đó, và ngược lại thì $f_Q(x, y) = 0$.
3. Gọi g là một hình ảnh được tạo thành bằng cách thêm vào mỗi điểm hạt giống trong S tất cả các điểm giá trị 1 trong f_Q có 8 kết nối với điểm hạt giống đó.
4. Gán nhãn từng thành phần được kết nối trong g với nhãn vùng khác nhau (ví dụ: số nguyên hoặc các chữ cái). Đây là hình ảnh phân đoạn thu được theo vùng đang phát triển.

Ví dụ: Thuật toán phát triển vùng




(a) Ảnh X- quang mối hàn lỗi. (b) Lược đồ xám. (c) Ảnh hạt giống ban đầu (sử dụng phân ngưỡng với percentile =99,9%). (d) Ảnh hạt giống cuối cùng (sau phép co). (e) Giá trị tuyệt đối của sự sai khác giữa giá trị hạt giống (255) và ảnh (a). (f) Lược đồ của (e). (g) Ảnh sai khác được phân ngưỡng sử dụng ngưỡng kép $T1 = 68$ và $T2 = 126$. (h) Ảnh sai khác được phân ngưỡng bằng $T1 = 68$. (i) Kết quả sau phân đoạn bằng thuật toán phát triển vùng

VD: Thuật toán phát triển vùng

Mệnh đề Q đo mức độ giống nhau sử dụng sự sai khác cường độ tuyệt đối:

$$Q = \begin{cases} True & \text{nếu sự sai khác cường độ tuyệt đối} \\ & \text{giữa hạt giống và pixel tại (x,y) } \leq T \\ False & \text{nếu khác} \end{cases}$$

Hình (e) cho thấy sự khác biệt giữa giá trị hạt giống (255) và Hình (a). Hình (e) chứa tất cả sự khác biệt cần thiết để tính toán mệnh đề tại mỗi vị trí (x,y). Hình 10.46 (f) cho thấy biểu đồ tương ứng. Chúng ta cần một ngưỡng để sử dụng trong mệnh đề để thiết lập sự giống nhau. Biểu đồ có ba chế độ chính, vì vậy chúng ta có thể sử dụng hai ngưỡng trong trường hợp này là $T1 = 68$ và $T2 = 126$, mà chúng ta thấy tương ứng gần với các thung lũng của biểu đồ.



Hình (g) cho thấy rằng không thể thực hiện phân đoạn các khiếm khuyết bằng cách sử dụng các ngưỡng kép, mặc dù thực tế là các ngưỡng nằm trong thung lũng sâu của biểu đồ. Hình (h) cho thấy kết quả của ngưỡng hình ảnh khác biệt chỉ với T1. Những điểm đen là các pixel mà mệnh đề là TRUE; những điểm khác không thỏa mãn mệnh đề. Kết quả quan trọng ở đây là các điểm trong vùng tốt của mỗi hàn không thỏa mãn mệnh đề, vì vậy chúng sẽ không được đưa vào trong kết quả cuối cùng. Các điểm ở vùng bên ngoài sẽ được thuật toán phát triển vùng coi là các ứng cử viên. Tuy nhiên, Bước 3 sẽ từ chối các điểm bên ngoài vì chúng không có được 8 kết nối với hạt giống. Trên thực tế, như Hình (i) cho thấy, bước này dẫn đến việc phân đoạn chính xác, cho thấy rằng việc sử dụng kết nối là một yêu cầu cơ bản trong trường hợp này.

Ở Bước 4, chúng ta sử dụng cùng một giá trị cho tất cả các vùng được tìm thấy bởi thuật toán. Trong trường hợp này, về mặt hình ảnh bạn nên làm như vậy vì tất cả các vùng đó đều có cùng ý nghĩa vật lý trong ứng dụng này — tất cả chúng đều đại diện cho các trạng thái lỗi

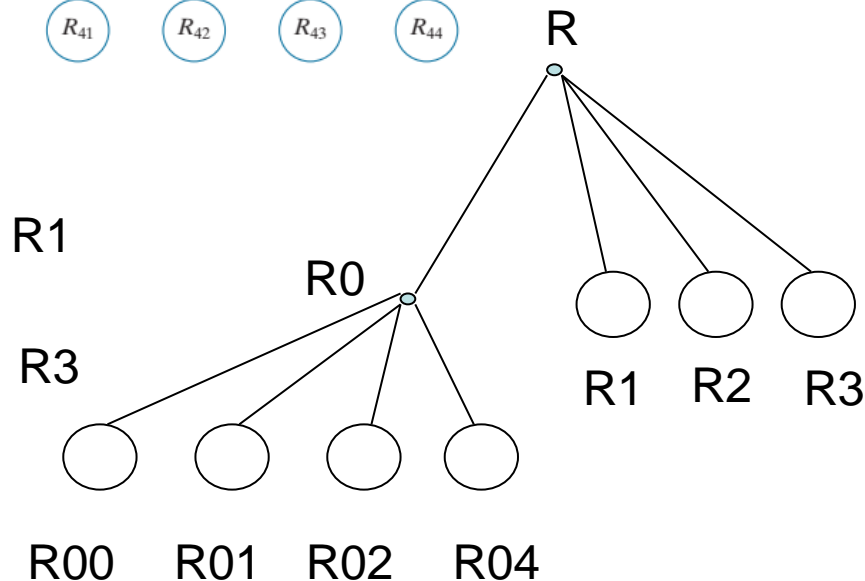
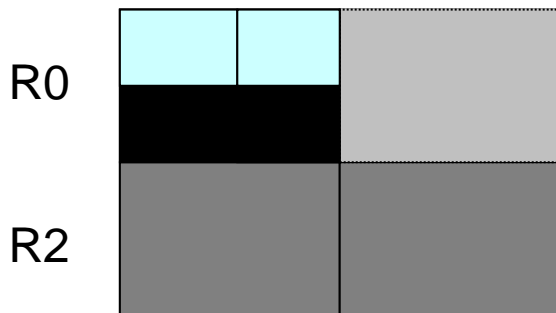
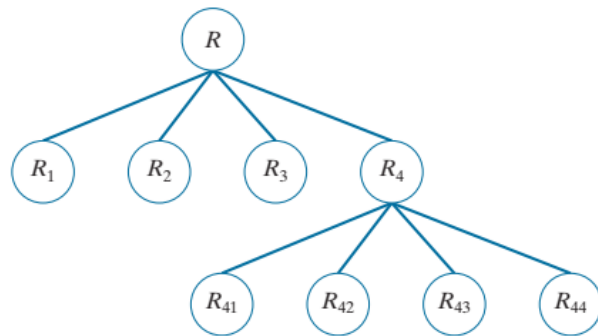
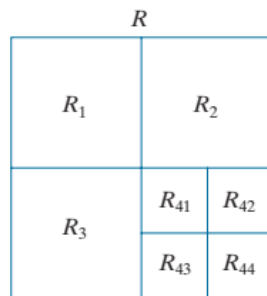
Chia tách và hợp vùng

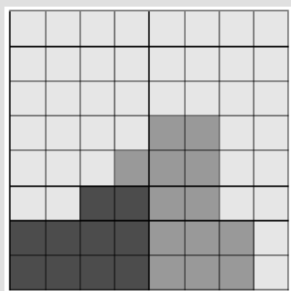
- Phát triển vùng là phát triển từ các điểm hạt giống.
- Một giải pháp thay thế là chia nhỏ một hình ảnh ban đầu thành một tập hợp các vùng rời rạc và sau đó hợp nhất và/hoặc tách các vùng nhằm thỏa mãn các điều kiện phân đoạn được nêu trong slide 5.

Chia tách và hợp vùng

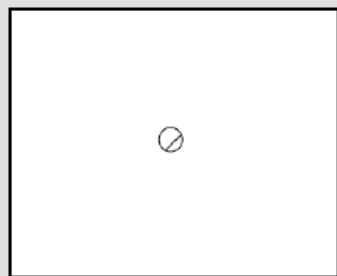
- Gọi R đại diện cho toàn bộ vùng hình ảnh và chọn một mệnh đề Q . Phân đoạn R là chia nhỏ liên tiếp R thành góc phần tư nhỏ hơn và nhỏ hơn vùng sao cho bất kỳ vùng R_i , $Q(R_i) = TRUE$.
- Bắt đầu với R . Nếu $Q(R) = FALSE$, chia ảnh thành các góc phần tư. Nếu Q là $FALSE$ cho bất kỳ góc phần tư, tiếp tục chia nhỏ góc phần tư đó thành các góc phần tư con, v.v...(Kỹ thuật cây tứ phân – quadtrees)
- Phân vùng cuối cùng thường chứa các vùng liền kề với thuộc tính giống hệt nhau. Hạn chế này có thể được khắc phục bằng cách cho phép hợp nhất. Hai vùng lân cận R_j và R_k chỉ được hợp nhất nếu $Q(R_j \cup R_k) = TRUE$.

Chia tách và hợp vùng

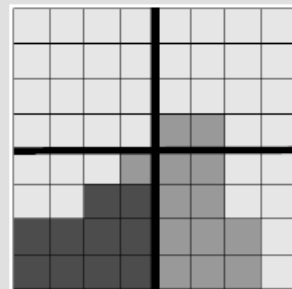




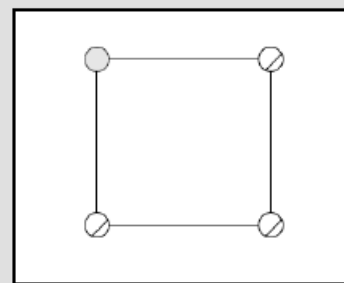
Initial image



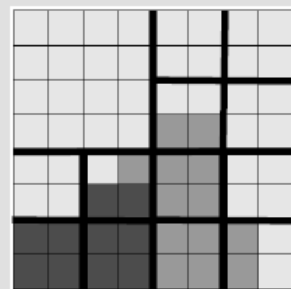
Design of the corresponding adjacency graph



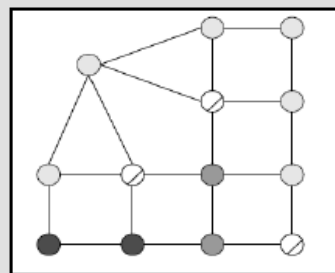
Initial image



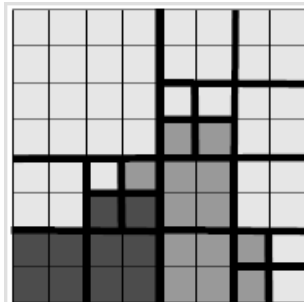
Design of the corresponding adjacency graph



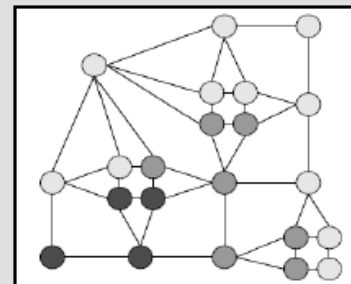
Initial image



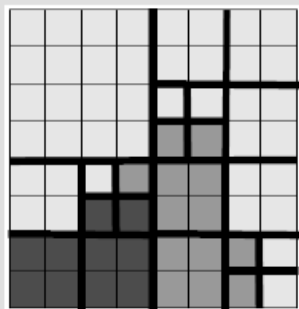
Design of the corresponding adjacency graph



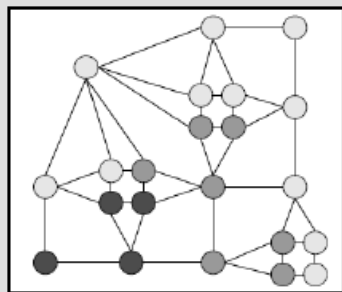
Initial image



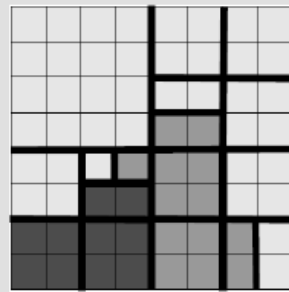
Design of the corresponding adjacency graph



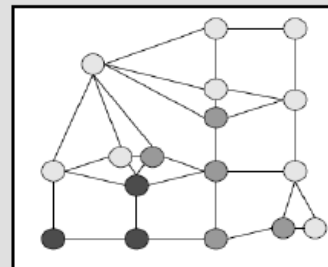
Initial image



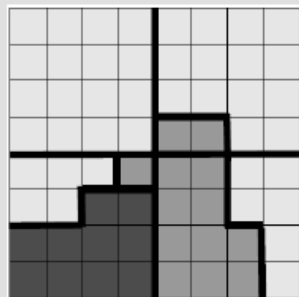
Building of the corresponding region-based segmentation



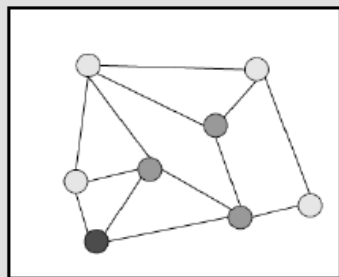
Initial image



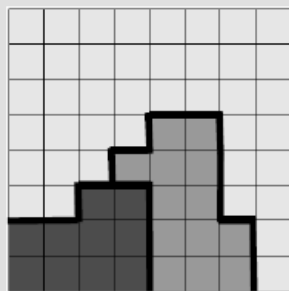
Building of the corresponding region-based segmentation



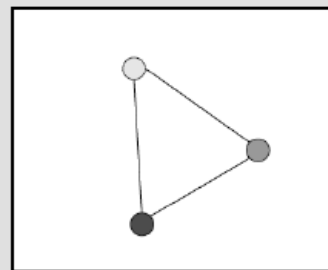
Initial image



Building of the corresponding region-based segmentation



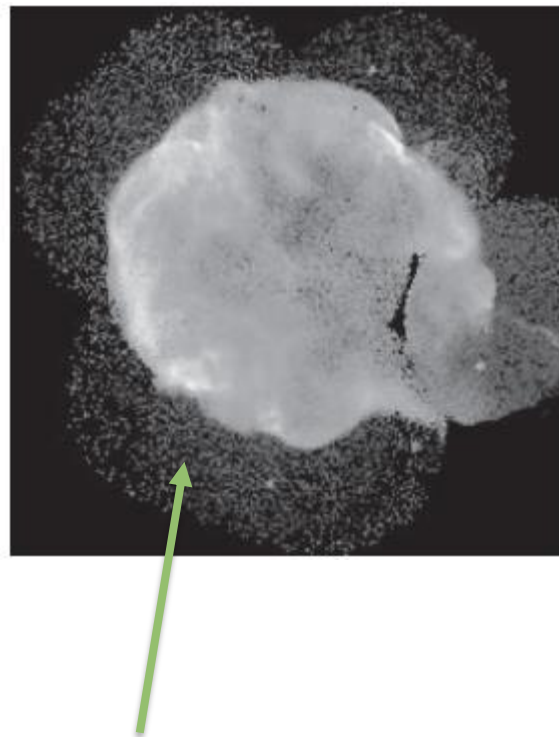
Initial image



Building of the corresponding region-based segmentation

Ví dụ Phân đoạn bằng chia tách và hợp vùng

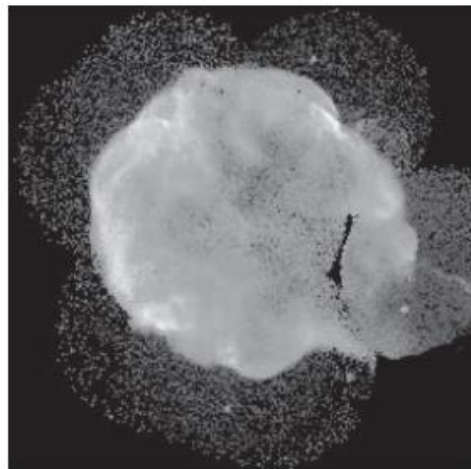
- Hình ảnh tia X 566×566 của siêu tân tinh Cygnus Loop. Mục tiêu của ví dụ này là phân đoạn (trích xuất từ hình ảnh) "vòng" của vật chất ít đặc hơn bao quanh vùng trong.



Nhận xét

- Dữ liệu trong vùng rìa ngoài có tính chất ngẫu nhiên, cho thấy *độ lệch chuẩn* của nó phải lớn hơn độ lệch chuẩn của nền (gần 0) và khu vực trung tâm lớn.
- Tương tự, *giá trị trung bình* (cường độ trung bình) của vùng chứa dữ liệu từ vòng ngoài phải lớn hơn giá trị trung bình của nền tối hơn và nhỏ hơn trung bình của khu vực trung tâm sáng hơn. Do đó, chúng ta có thể phân đoạn khu vực quan tâm bằng cách sử dụng mệnh đề sau:

$$Q(R) = \begin{cases} True & \text{if } \sigma_R > a \text{ AND } 0 < m_R < b \\ False & \text{otherwise} \end{cases}$$

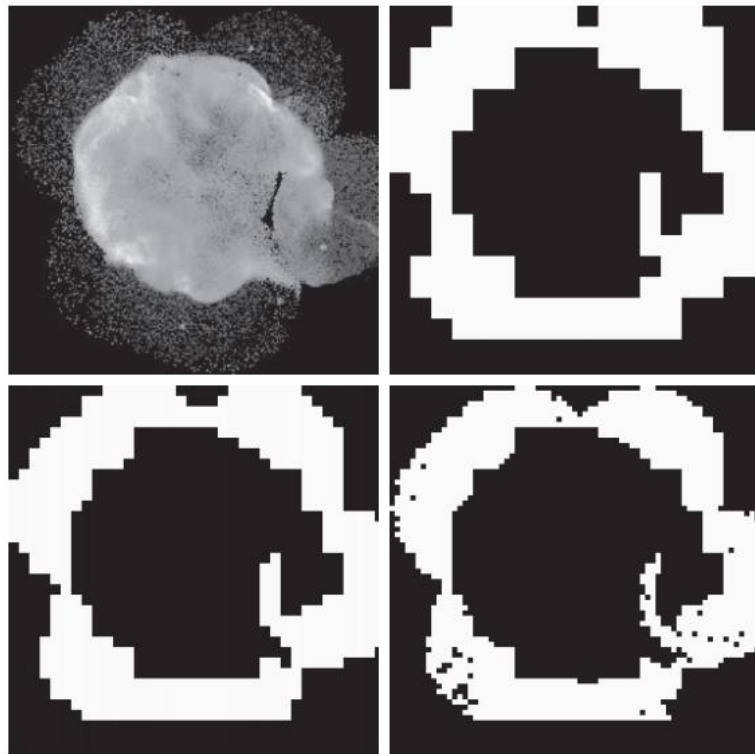


Nhận xét

Phân tích một số vùng trong khu vực quan tâm bên ngoài cho thấy rằng cường độ trung bình của pixel ở những vùng đó không vượt quá 125 và độ lệch chuẩn luôn lớn hơn 10. Do đó chọn $a = 10$ và $b = 125$.

(a) Ảnh gốc

(b) đến (d): kết quả của việc chia tách và hợp vùng với các kích thước 32×32 , 16×16 , và 8×8 .



Một số kết quả



Phát triển vùng



Chia tách vùng



Chia tách và hợp vùng