

Xử lý ảnh

Hoàng Văn Hiệp

Bộ môn Kỹ thuật máy tính

Viện Công nghệ thông tin và Truyền thông

Email: hiephv@soict.hut.edu.vn

Nội dung

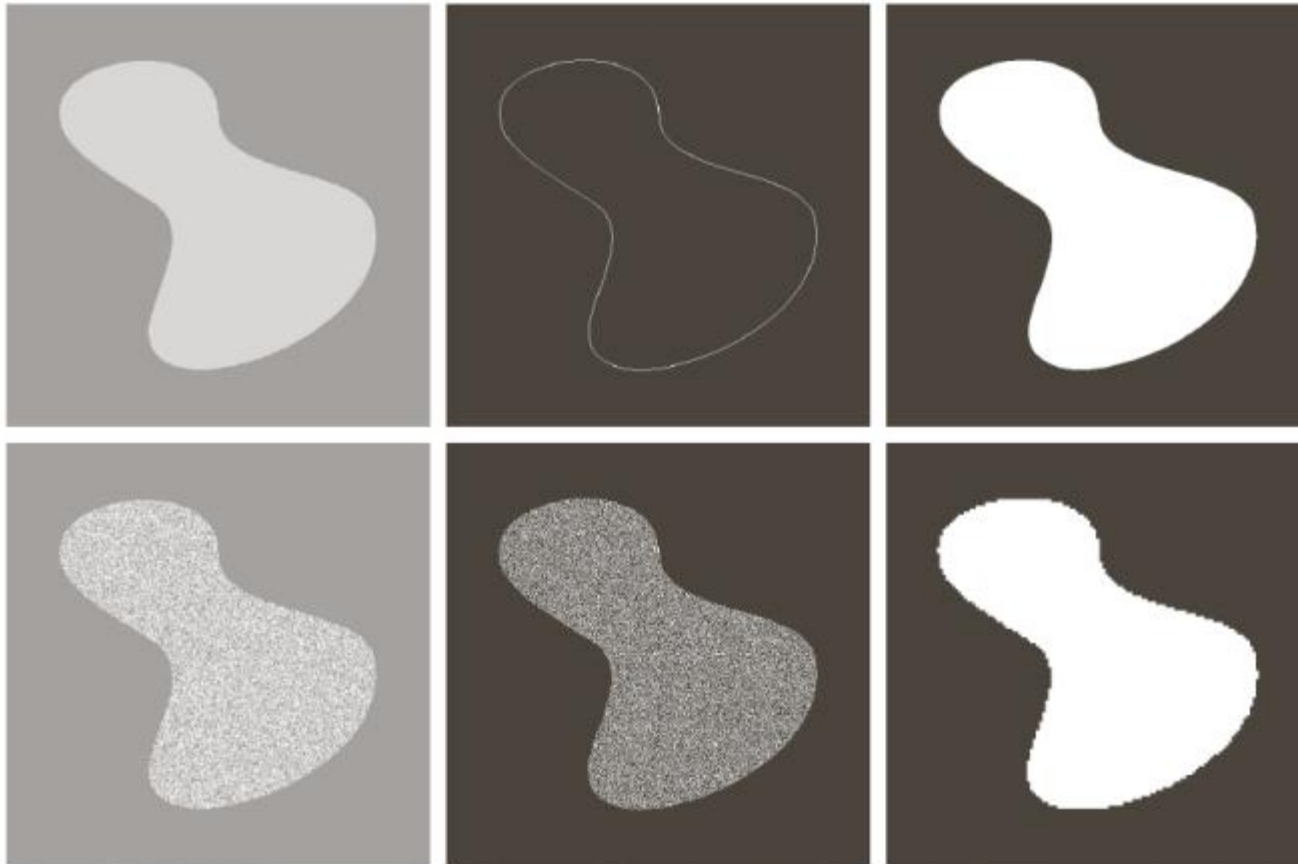
- ❑ Chương 1. Giới thiệu chung
- ❑ Chương 2. Thu nhận & số hóa ảnh
- ❑ Chương 3. Cải thiện & phục hồi ảnh
- ❑ Chương 4. Phát hiện tách biên, phân vùng ảnh
- ❑ Chương 5. Trích chọn các đặc trưng trong ảnh
- ❑ Chương 6. Nén ảnh
- ❑ Chương 7. Lập trình xử lý ảnh bằng Matlab và C

Chương 4. Phân vùng ảnh

□ Hai phương pháp chính áp dụng trong

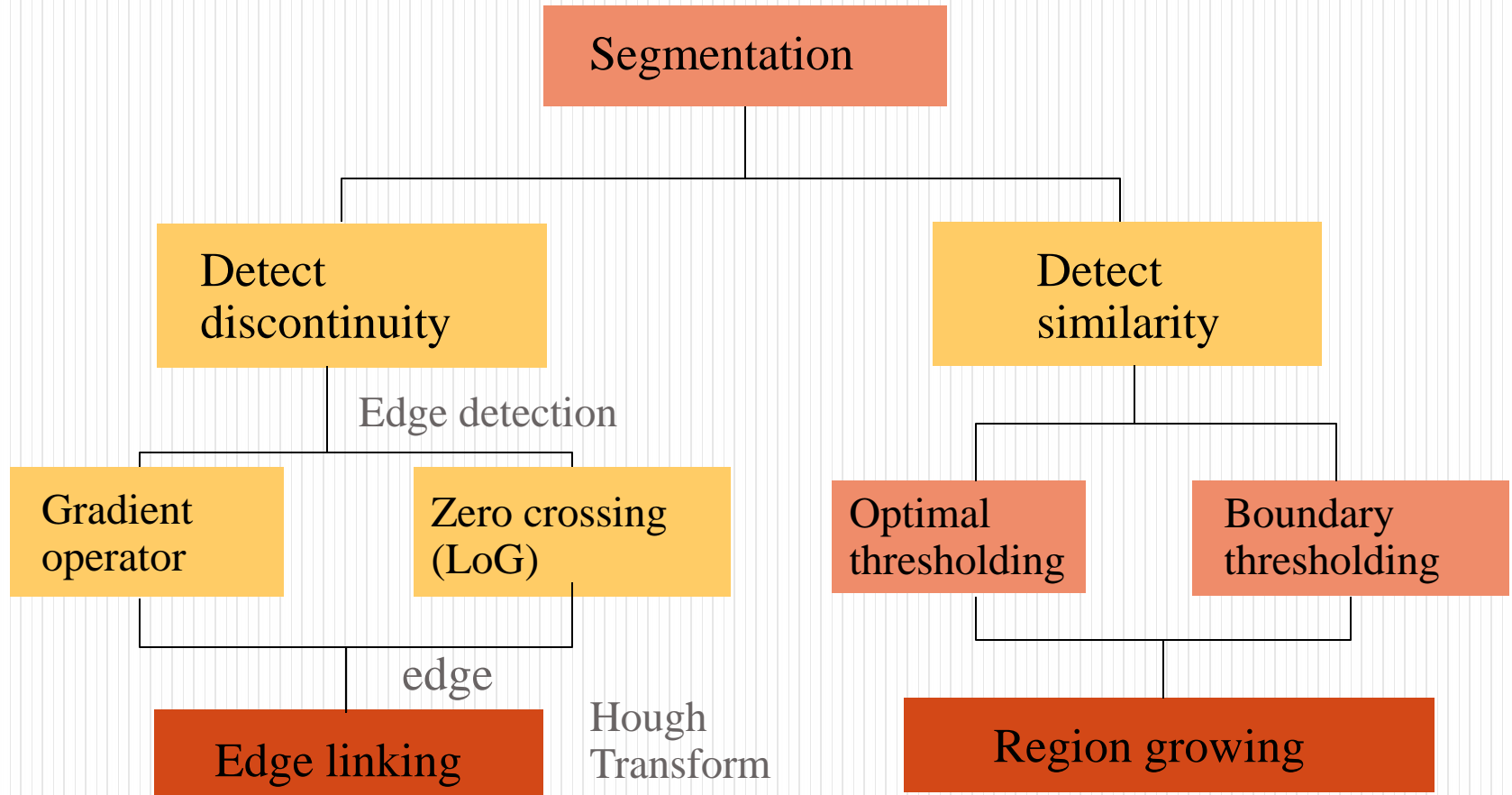
p

-
-



biên

Phân vùng ảnh dựa trên ngưỡng



Phân vùng ảnh dựa trên ngưỡng (tiếp)

□ Cơ sở

- Khi đối tượng và nền được nhóm lại trong các vùng
- Lựa chọn một ngưỡng T có thể phân tách các vùng
- Điểm ảnh $p(x, y)$
 - Nếu $f(x, y) > T \rightarrow p(x, y)$ thuộc đối tượng
 - Nếu $f(x, y) < T \rightarrow p(x, y)$ thuộc nền
- Có thể có nhiều ngưỡng

Phân vùng ảnh dựa trên ngưỡng (tiếp)

□ Lấy ngưỡng có thể coi là bài toán xác định hàm T : $T = T[x, y, p(x, y), f(x, y)]$

- $f(x, y)$: biểu diễn mức xám của điểm ảnh (x, y)
- $p(x, y)$: hàm mô tả thuộc tính cục bộ của ảnh

□ Ảnh sau lấy ngưỡng

- Hai cấp (bi-level)

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

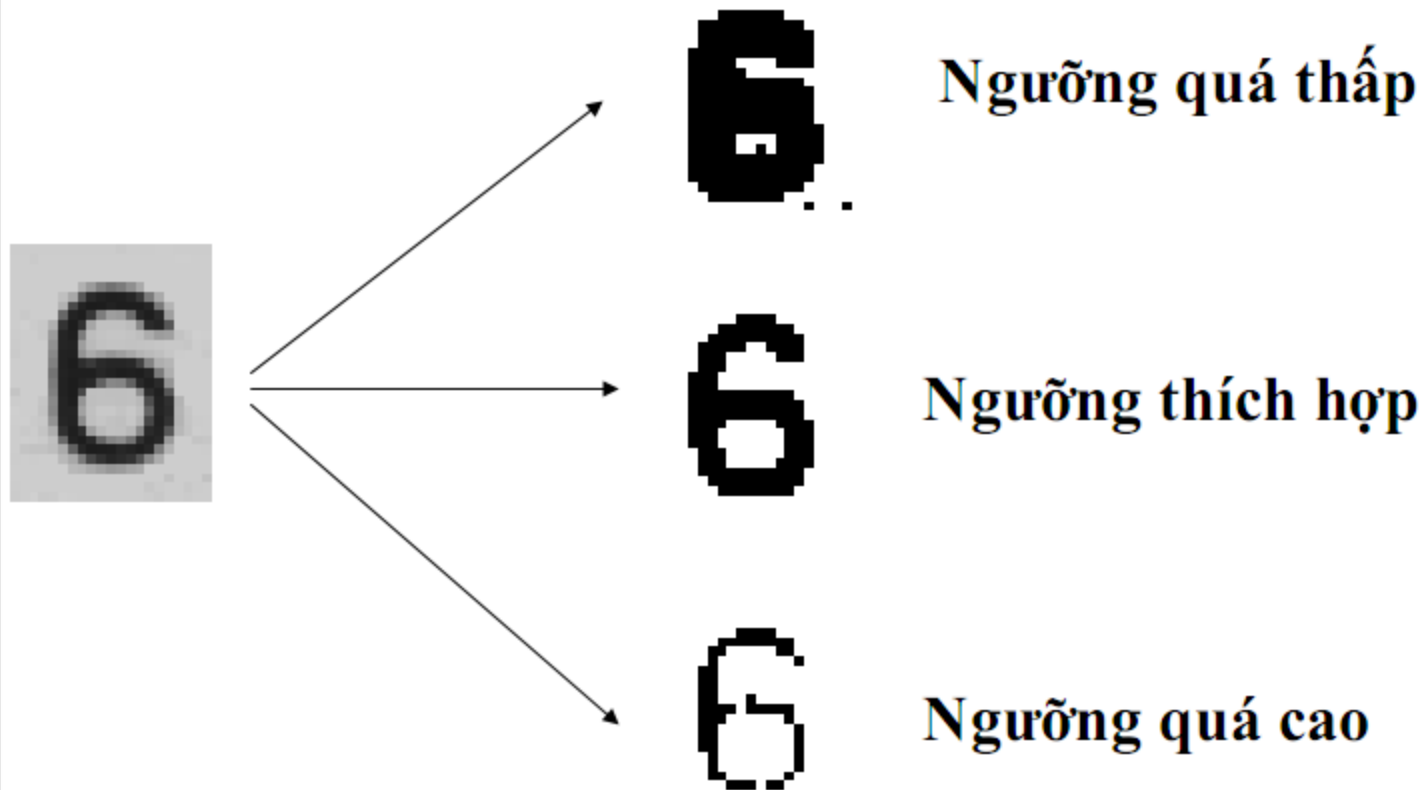
- Đa cấp (multi-level)

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

Phân vùng ảnh dựa trên ngưỡng (tiếp)

- ❑ Vấn đề: làm sao để chọn giá trị ngưỡng T thích hợp
 - Nếu T chỉ phụ thuộc $f(x, y)$: phép lấy ngưỡng toàn cục
 - Nếu T phụ thuộc vào $P(x, y)$ và $f(x, y)$: phép lấy ngưỡng cục bộ
 - Nếu T phụ thuộc x, y : Phép lấy ngưỡng thích nghi (adaptive thresholding)

Phân vùng ảnh dựa trên ngưỡng (tiếp)

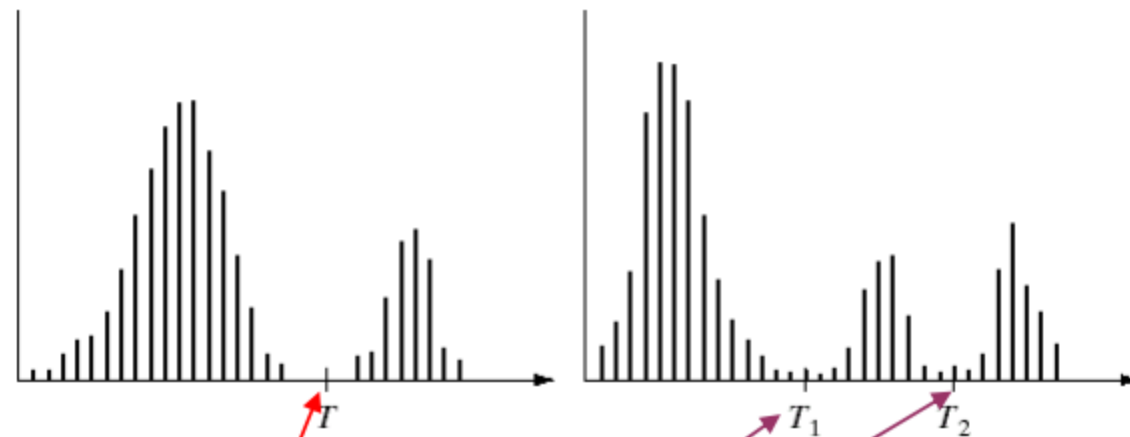


Phân vùng ảnh dựa trên ngưỡng (tiếp)

□ Các phương pháp lấy ngưỡng

- Lấy ngưỡng cứng
- Lấy ngưỡng toàn cục
- Lấy ngưỡng cục bộ
- Lấy ngưỡng thích nghi
- Lấy ngưỡng dựa trên kiểm chứng
- Lấy ngưỡng dựa trên phân nhóm (gom nhóm)

Lấy ngưỡng cứng



Thresholds

Lấy ngưỡng cứng (tiếp)

- ❑ Phụ thuộc chủ quan
 - → phân tích histogram
- ❑ Dễ bị ảnh hưởng bởi nhiễu
- ❑ Ảnh hưởng bởi thay đổi độ sáng

Ảnh hưởng của nhiễu

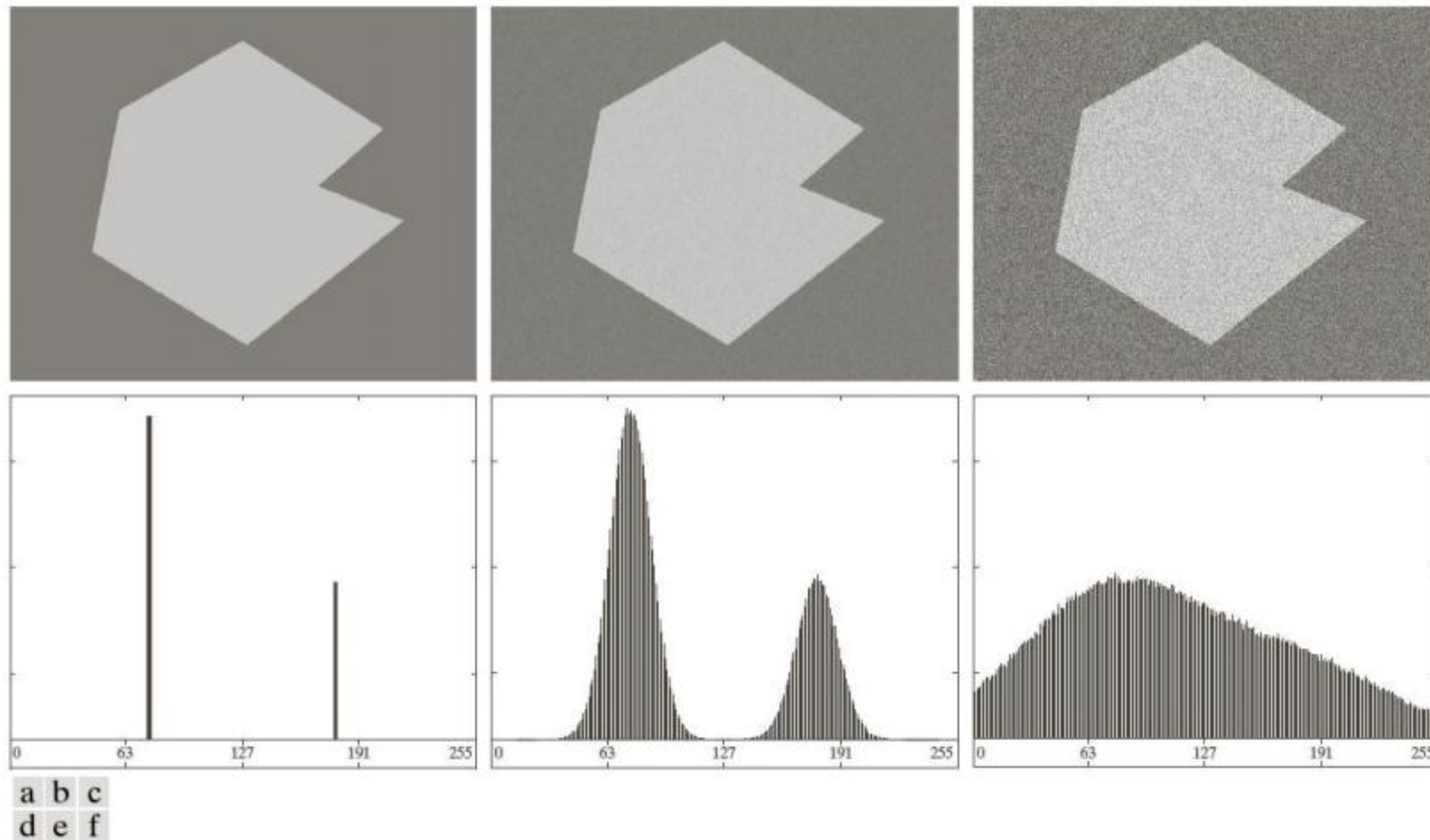


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Ảnh hưởng của độ sáng

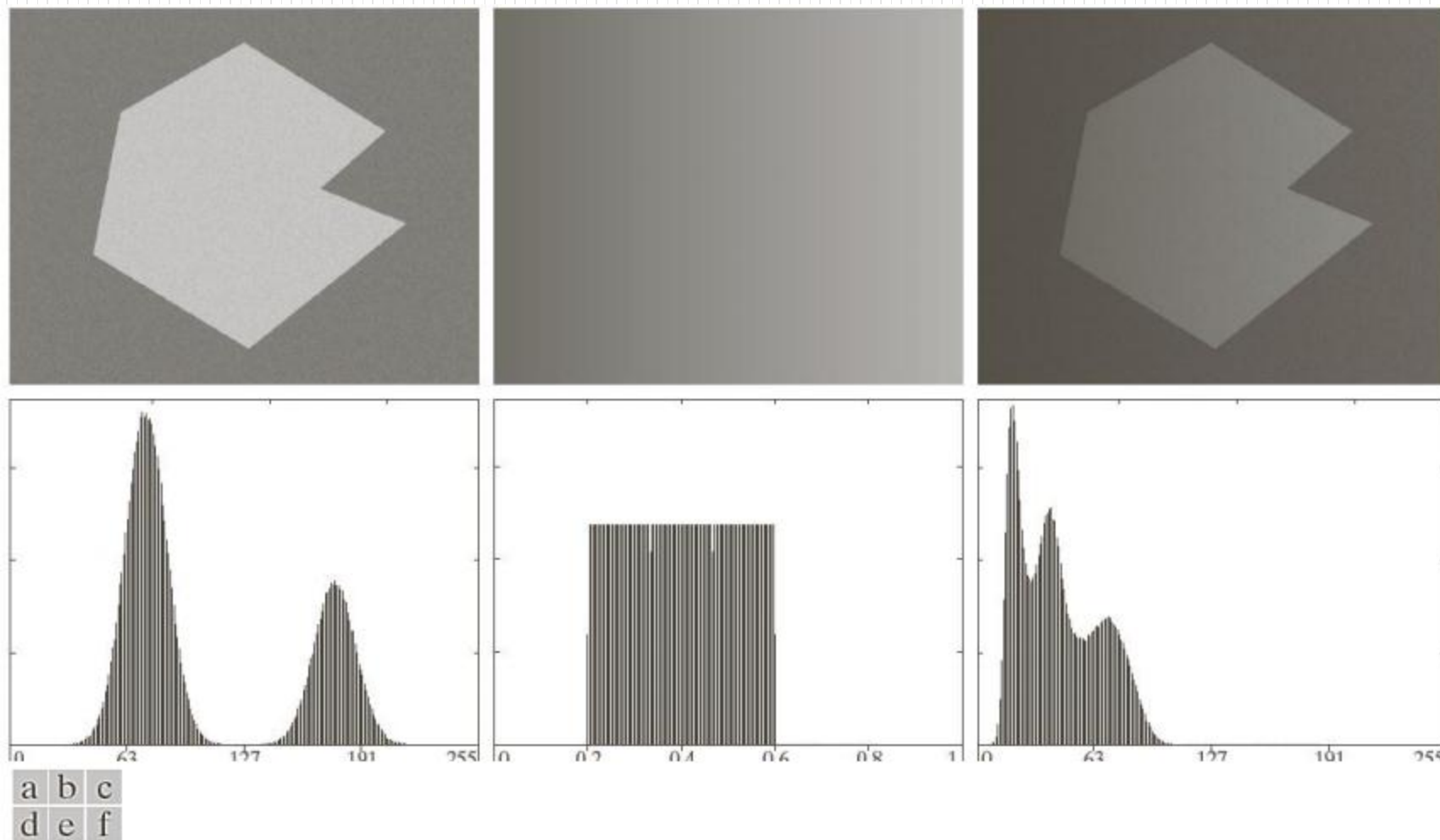


FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range $[0.2, 0.6]$. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Ảnh hưởng của độ sáng

□ Nhận xét

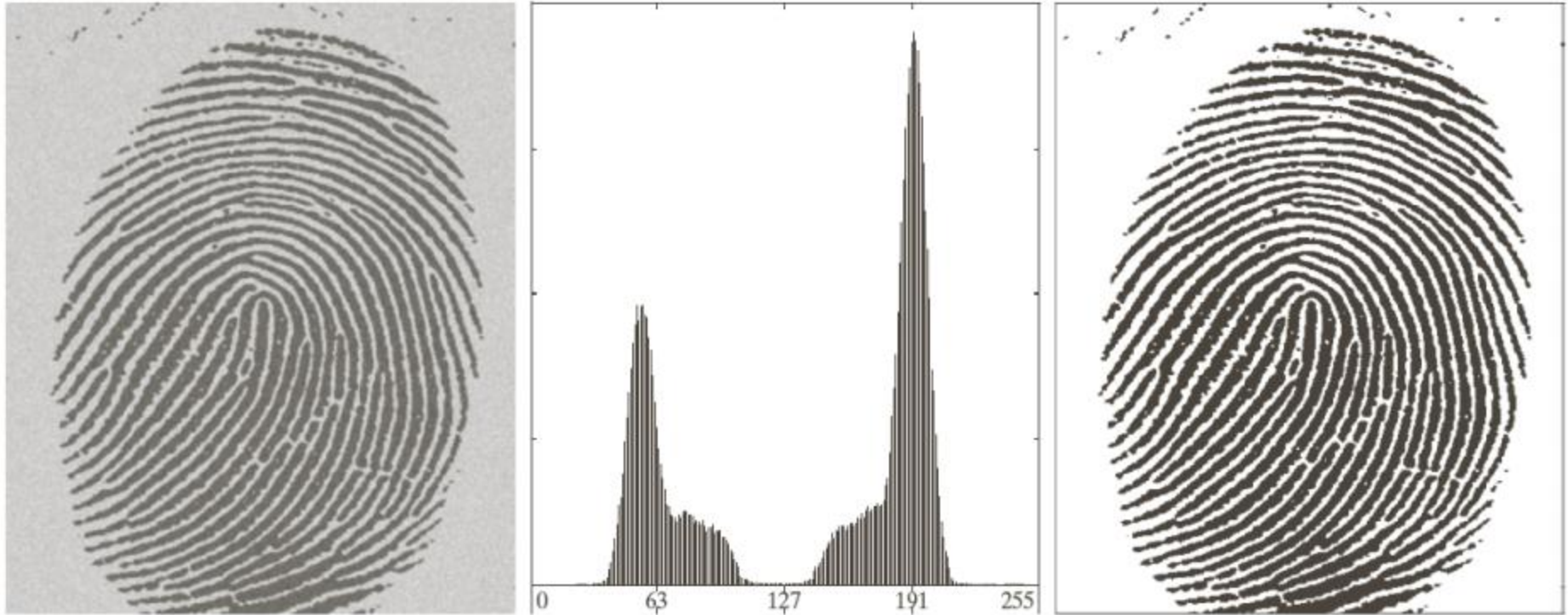
- Ảnh có độ sáng đồng đều tại các vùng sẽ dễ tìm ngưỡng hơn

Lấy ngưỡng toàn cục

□ Cách tiếp cận heuristic

- Bước 1. Xác định giá trị khởi tạo của T (thường là giá trị trung bình mức xám ảnh)
- Bước 2. Chia ảnh thành 2 vùng: $G1$ (gồm các điểm ảnh mức xám $\geq T$) và vùng 2 (gồm những điểm ảnh mức xám $< T$)
- Bước 3. Tính giá trị trung bình mức xám của $G1$ là $m1$, $G2$ là $m2$
- Bước 4. Cập nhật $T = (m1 + m2)/2$
- Bước 5. Quay lại bước 2 đến khi nào $\Delta T \leq \varepsilon$

Lấy ngưỡng toàn cục (tiếp)



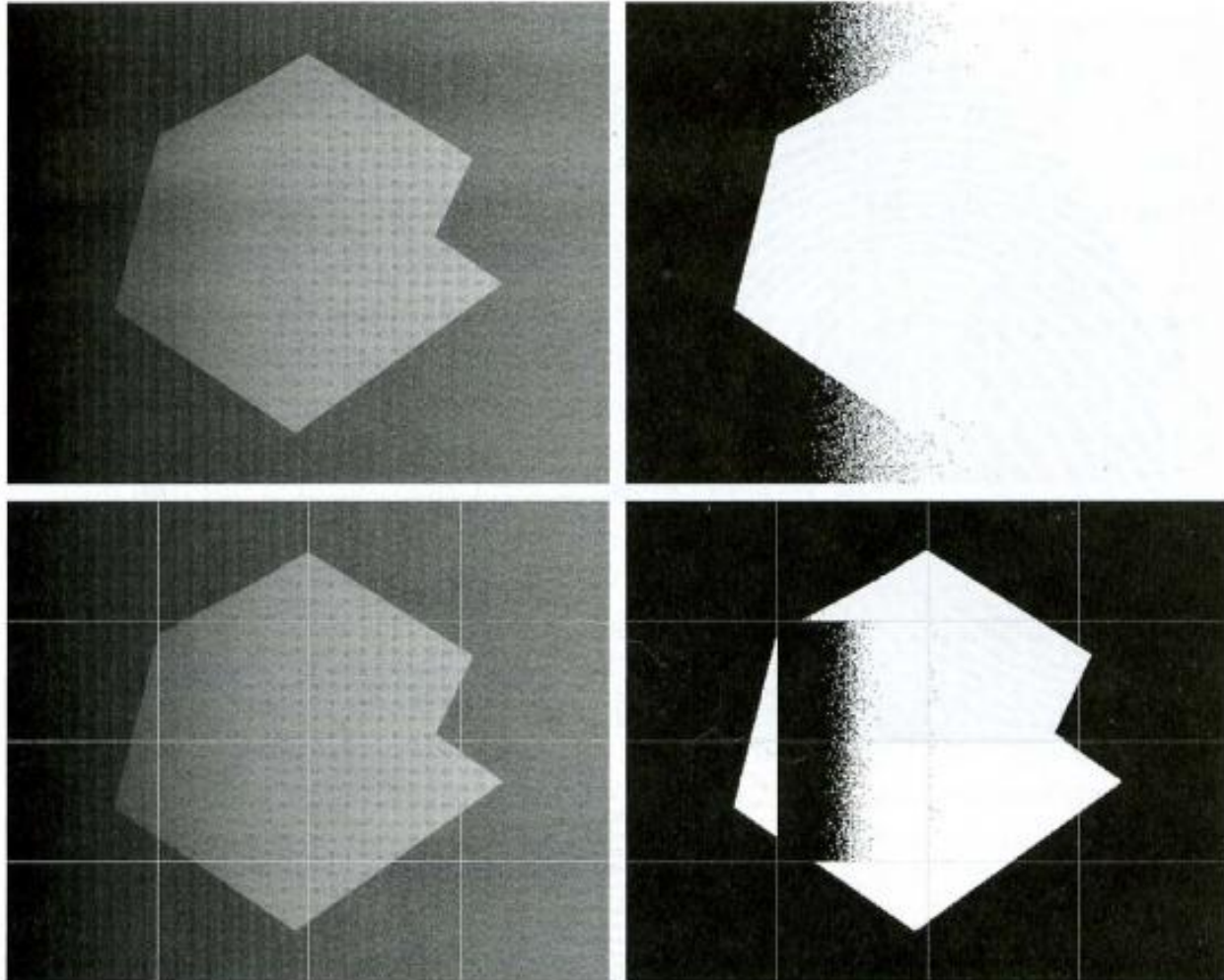
$$T_0 = 0 \xrightarrow[\Delta T = 0]{\text{Convergence}} T_f = 125.4$$

Lấy ngưỡng thích nghi

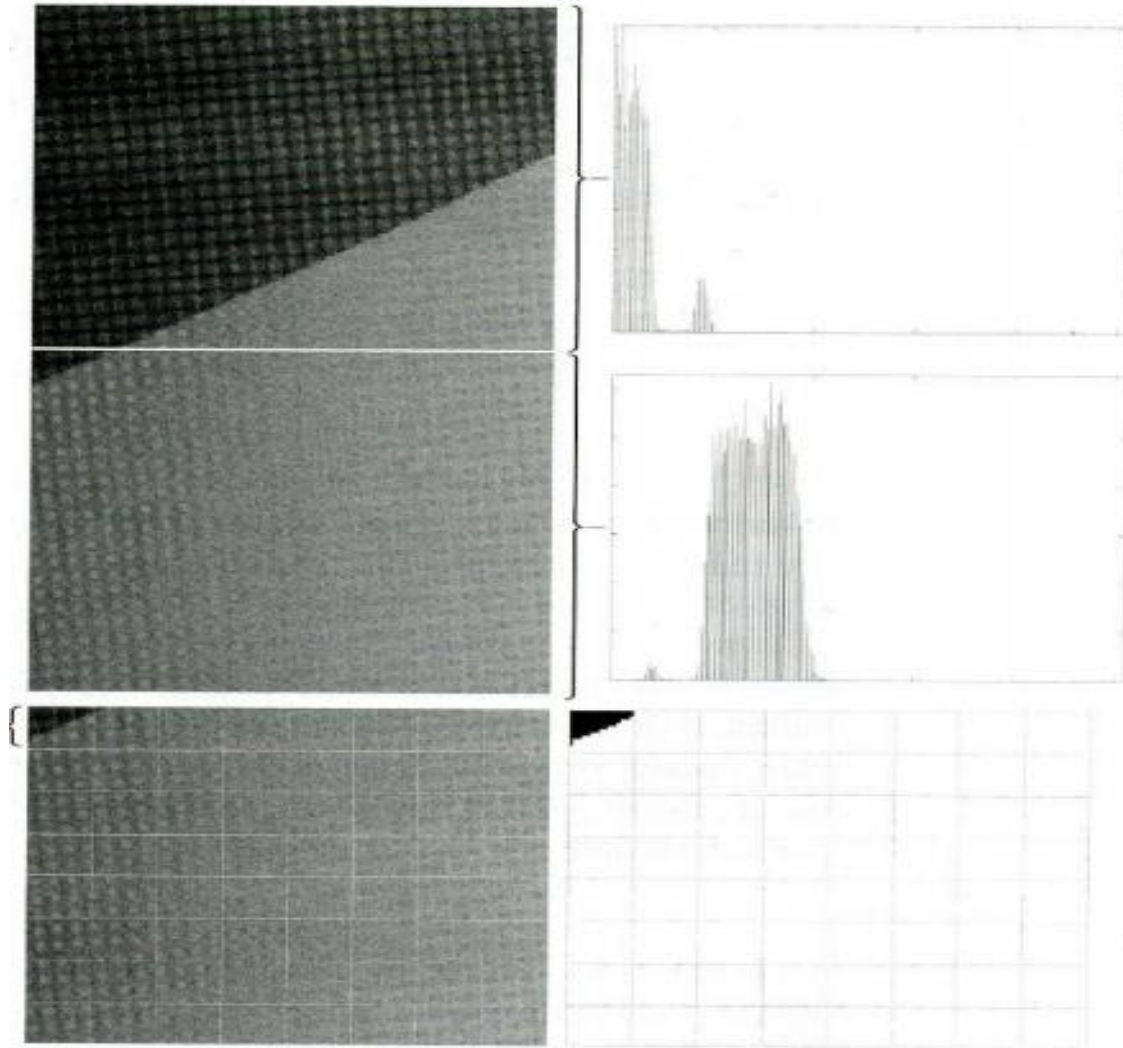
□ Ý tưởng

- Ngưỡng toàn cục bị ảnh hưởng bởi độ sáng, nhiều
- → Chia nhỏ ảnh thành các phần, sau đó áp dụng tìm ngưỡng khác nhau cho từng phần nhỏ
 - Vấn đề:
 - Chia như thế nào là hợp lý
 - Tìm ngưỡng cho từng phần nhỏ

Lấy ngưỡng thích nghi (tiếp)



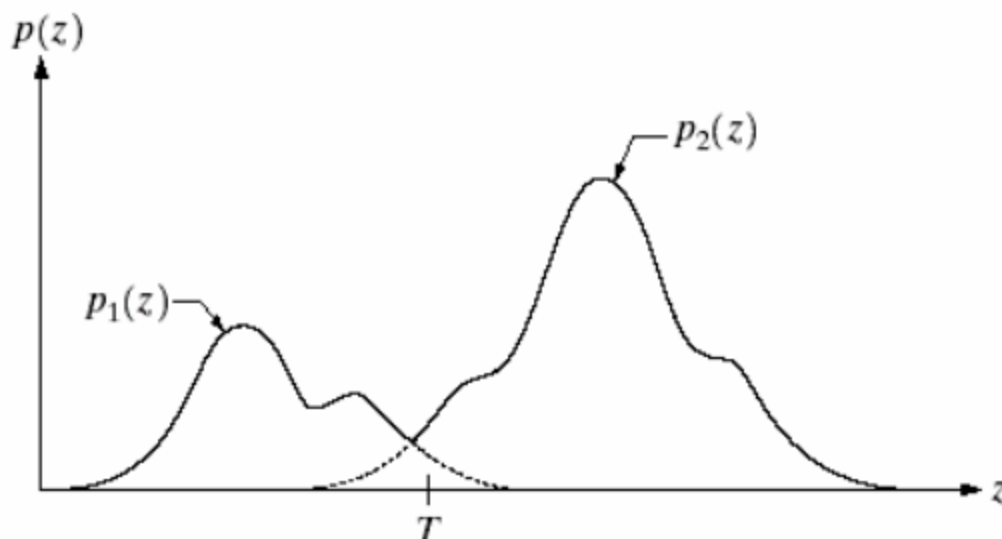
Lấy ngưỡng thích nghi (tiếp)



Lấy ngưỡng tối ưu

- ❑ Giả sử ảnh có 2 vùng chính rõ rệt (vùng đối tượng và nền)
- ❑ Một điểm (x, y) trong ảnh có 2 khả năng
 - H_0 : Không thuộc vùng đối tượng
 - H_1 : Thuộc vùng đối tượng
- ❑ Gọi z là giá trị mức xám trong ảnh (z coi như biến ngẫu nhiên)
- ❑ Các xác suất
 - Xác suất tiên nghiệm: $P_1 = p(z \in H_1)$; $P_2 = p(z \in H_0)$;

Lấy ngưỡng tối ưu (tiếp)



- ❑ $p_1(z)$: hàm mật độ phân bố xác suất của các pixel trên đối tượng
- ❑ $p_2(z)$: hàm mật độ phân bố xác suất của nền (chú ý rằng ta chưa có $p_1(z)$ và $p_2(z)$)
- ❑ Hàm mật độ phân bố xác suất $p(z)$

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

Lấy ngưỡng tối ưu (tiếp)

□ T là ngưỡng được chọn để phân vùng ảnh (pixel > T → nền và ngược lại)

- Xác suất lỗi khi phân vùng các pixel trên nền là đối tượng

$$E_1(T) = \int_{-\infty}^T p_2(z) dz.$$

- Xác suất lỗi khi phân vùng các pixel trên đối tượng là nền

$$E_2(T) = \int_T^{\infty} p_1(z) dz,$$

- Xác suất lỗi tổng:

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

Lấy ngưỡng tối ưu (tiếp)

- Bài toán đặt ra là tìm T để, xác suất lỗi nhỏ nhất

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

- Giải, áp dụng luật leibniz cuối cùng thu được

$$P_1 p_1(T) = P_2 p_2(T)$$

Lấy ngưỡng tối ưu (tiếp)

□ Để giải $P_1 p_1(T) = P_2 p_2(T)$

- Chúng ta cần biết p_1 và p_2 , tuy nhiên thực tế thì p_1 và p_2 là chưa biết
- → giải quyết có 2 cách
 - C1) Giả sử phân bố p_1 và p_2 là các phân bố Gaussian (không có giám sát)
 - C2) Xấp xỉ phân bố $p(z)$ là các phân bố Gaussian từ histogram của ảnh (có giám sát) sao cho tối thiểu hóa:

$$e_{\text{ms}} = \frac{1}{n} \sum_{i=1}^n [p(z_i) - h(z_i)]^2$$

Lấy ngưỡng tối ưu (tiếp)

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(z-\mu_1)^2}{2\sigma_1^2}} + \frac{P_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{(z-\mu_2)^2}{2\sigma_2^2}}$$

- ❑ Phương trình $P_1 p_1(T) = P_2 p_2(T)$
- ❑ Lấy logarit 2 vế đưa phương trình về

$$AT^2 + BT + C = 0$$

$$A = \sigma_1^2 - \sigma_2^2 \qquad B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right)$$

$$\text{Nếu } \sigma_1 = \sigma_2 = \sigma, \qquad T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \cdot \ln\left(\frac{P_2}{P_1}\right)$$

Lấy ngưỡng tối ưu (tiếp)

- Chú ý: có thể lấy ngưỡng tối ưu bằng cách xấp xỉ với các hàm khác Gaussian
 - Raleigh
 - Log-normal
 - ...

Lấy ngưỡng tối ưu Otsu

□ Hàm **graythresh** trong matlab hiện đang cài đặt theo phương pháp này

□ Bài toán

- Cho ảnh đa mức xám $M \times N$
- L mức xám $\{0, 1, 2, \dots, L-1\}$
- n_i : số pixel trong ảnh có mức xám i

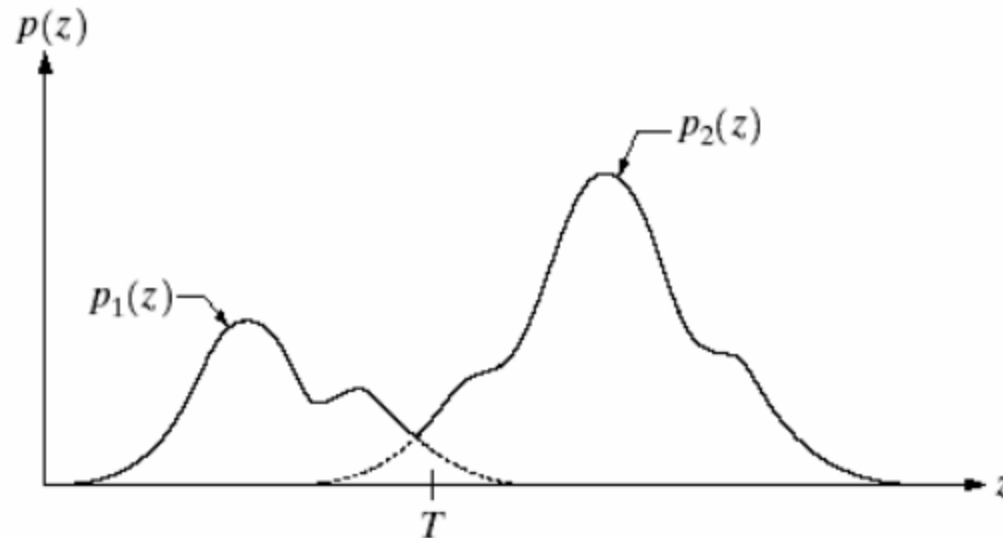
- $MN = n_0 + n_1 + \dots + n_{L-1}$

- Histogram chuẩn hóa:

$$\sum_{i=0}^{L-1} p_i = 1$$

- Tìm ngưỡng t tối ưu

Lấy ngưỡng tối ưu Otsu (tiếp)



Lấy ngưỡng tối ưu Otsu (tiếp)

- Với ngưỡng k , ta có 2 lớp pixel

$$C_1 = \{\text{pixels} | \text{intensity} \in [0, k]\}$$

$$C_2 = \{\text{pixels} | \text{intensity} \in [k+1, L-1]\}$$

- Ý tưởng: Tìm ngưỡng sao cho
minimizes the weighted within-class
variance \rightarrow tương tự với việc
maximizing the between-class variance

Lấy ngưỡng tối ưu Otsu (tiếp)

□ Weighed within-class variance

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

□ Trong đó:

$$q_1(t) = \sum_{i=1}^t P(i)$$

$$q_2(t) = \sum_{i=t+1}^I P(i)$$

Lấy ngưỡng tối ưu Otsu (tiếp)

□ Class mean

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad \mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)}$$

□ Class variance

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Lấy ngưỡng tối ưu Otsu (tiếp)

□ Total variance

$$\sigma^2 = \underbrace{\sigma_w^2(t)}_{\text{Within-class, from before}} + \underbrace{q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2}_{\text{Between-class, } \sigma_B^2(t)}$$

□ Vì total variance = const

→ Minimize within-class tương đương với maximize between-class

Lấy ngưỡng tối ưu Otsu (tiếp)

□ Thuật toán:

- Bước 1. Tính histogram, và xác suất tại mỗi giá trị mức xám
- Bước 2. Khởi tạo $q_1(1) = P(1); \mu_1(0) = 0$
- Bước 3. Duyệt lần lượt các giá trị của t từ 1 đến $L-1$
 - Tính $q_1(t); \mu_1(t)$
 - Tính $\sigma_B^2(t)$
- Bước 4. Cập nhật ngưỡng t ứng với $\sigma_B^2(t)$ lớn nhất

Lấy ngưỡng tối ưu Otsu (tiếp)

□ Chú ý có thể tính $\sigma_B^2(t)$ bằng cách đệ qui

- Khởi tạo: $q_1(1) = P(1)$; $\mu_1(0) = 0$

- Đệ qui:

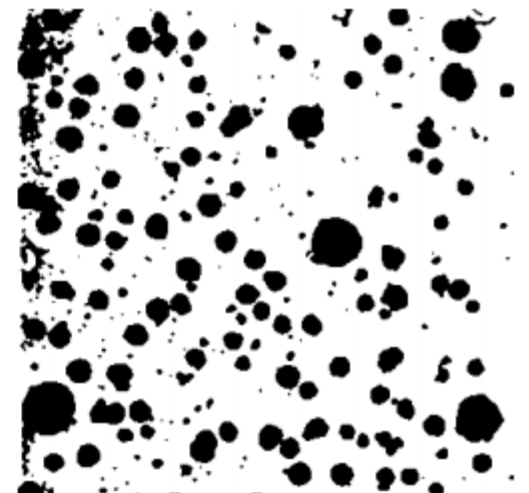
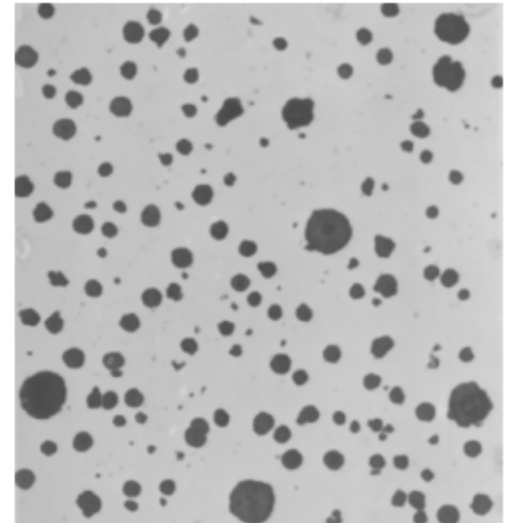
$$q_1(t+1) = q_1(t) + P(t+1)$$

$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)}$$

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$

Lấy ngưỡng tối ưu Otsu (tiếp)

```
>>n=imread('nodules1.tif');  
>> tn=graythresh(n)  
  
tn = 0.5804  
  
>> imshow(im2bw(n,tn))
```



Ảnh hưởng của nhiễu đến lấy ngưỡng Otsu

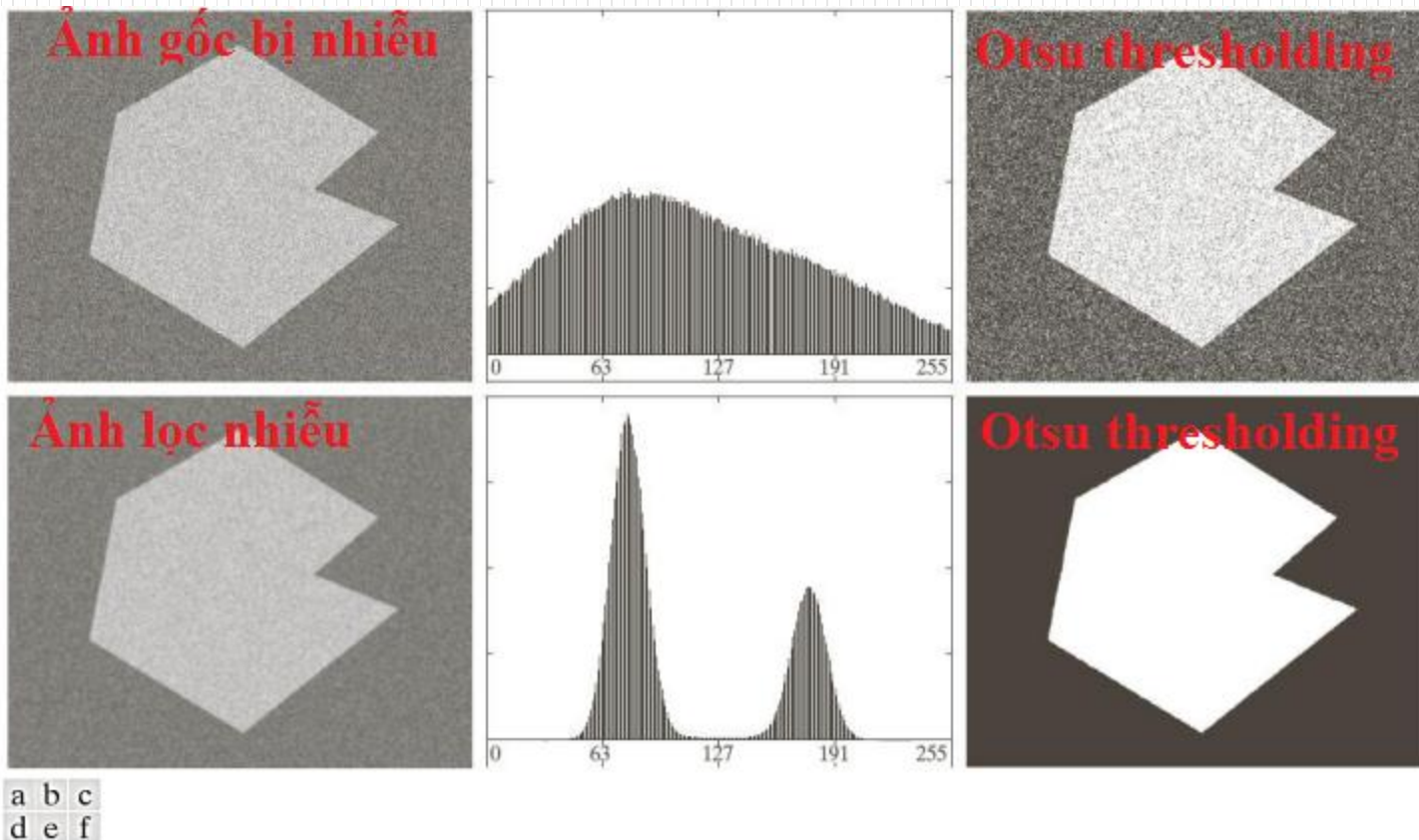


FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Ảnh hưởng kích thước vùng đến lấy ngưỡng Otsu

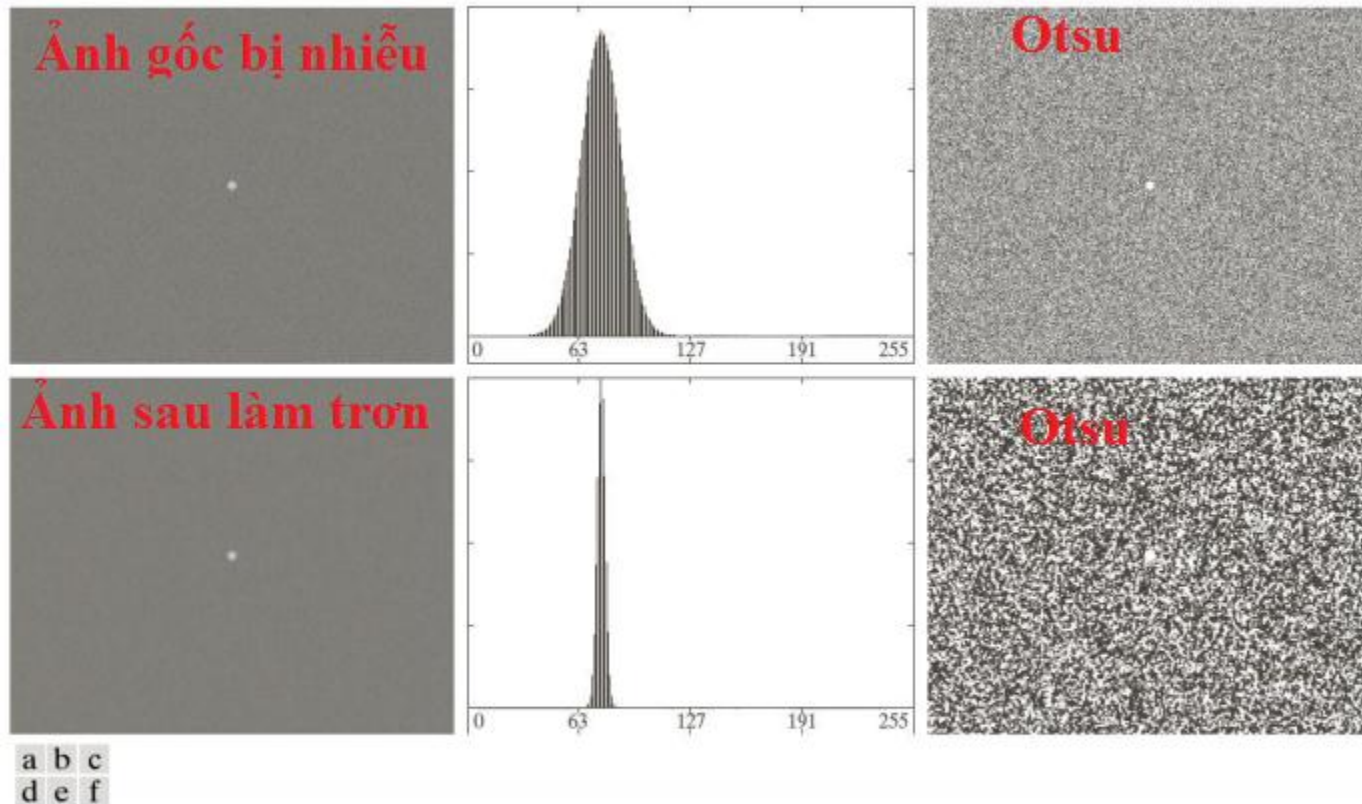


FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

Cải thiện lấy ngưỡng bằng cách kết hợp thông tin biên

□ Thuật toán

- Tính gradient hoặc laplacian của ảnh ban đầu
- Lấy ngưỡng trên ảnh gradient hoặc laplacian để loại bỏ các điểm nhiễu
- Nhân ảnh ban đầu với ảnh gradient hoặc ảnh laplacian → xây dựng histogram
- Lấy ngưỡng Otsu của ảnh ban đầu dựa trên histogram vừa tìm được

Cải thiện lấy ngưỡng bằng cách kết hợp thông tin biên

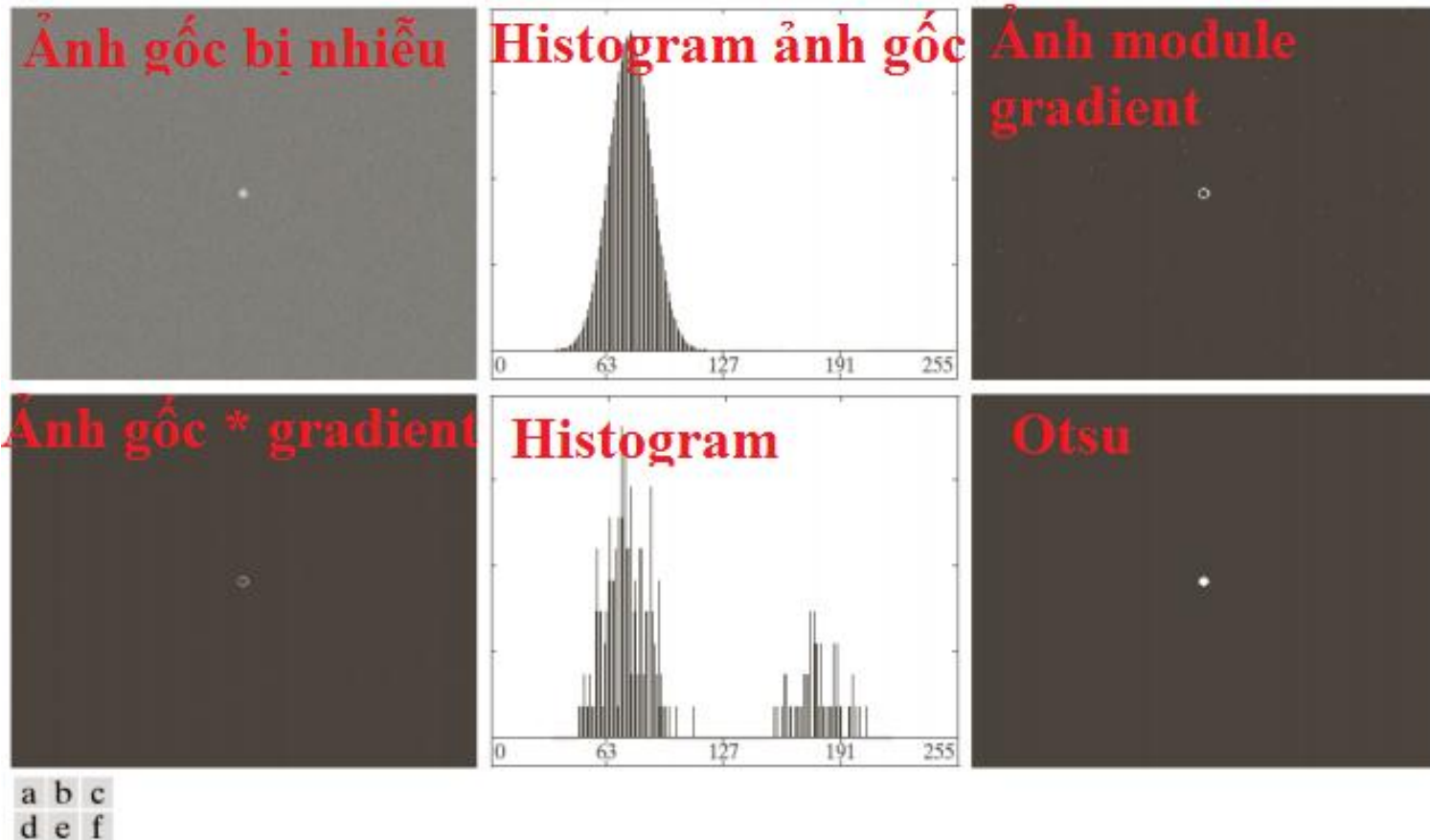


FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.

Phân vùng ảnh dựa trên các thuật toán gom nhóm

- ❑ Mỗi điểm ảnh được đại diện bởi một vector đặc trưng

$$s = [s_1, s_2, \dots, s_n]$$

- ❑ Các đặc trưng có thể là
 - Giá trị mức xám
 - Giá trị thành phần màu sắc
 - Các độ đo các lân cận (ví dụ giá trị trung bình trong cửa sổ chạy)
 - ...
- ❑ Phân nhóm: tiến hành gom các vector giống nhau vào cùng một nhóm

Phân vùng ảnh dựa trên các thuật toán gom nhóm (tiếp)

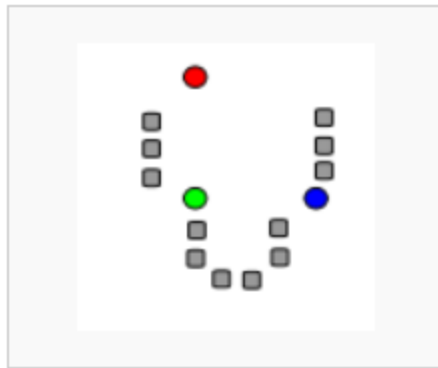
□ Các phương pháp phân nhóm

- K-means
- ISODATA

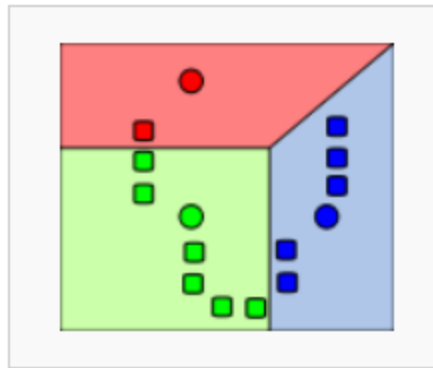
□ Thuật toán K-means

- Bước 1. Khởi tạo k tâm của k nhóm
- Bước 2. Phân loại n điểm vào k nhóm dựa vào khoảng cách đến các tâm
- Bước 3. Tính lại tâm của mỗi nhóm (giá trị trung bình), quay lại bước 2 hoặc sang bước 4
- Bước 4. Thuật toán dừng khi tâm các nhóm ở lần $i + 1$ so với lần thứ i không có thay đổi

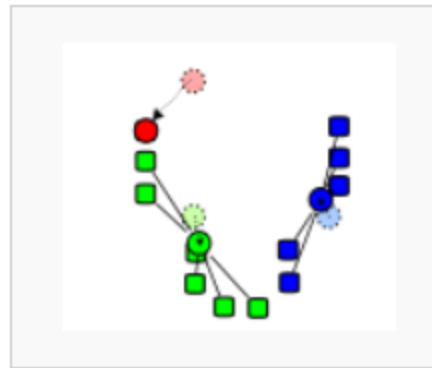
Thuật toán K-means



1) k initial "means" (in this case $k=3$) are randomly selected from the data set (shown in color).



2) k clusters are created by associating every observation with the nearest mean. The partitions here represent the [Voronoi diagram](#) generated by the means.



3) The [centroid](#) of each of the k clusters becomes the new means.



4) Steps 2 and 3 are repeated until convergence has been reached.

Thuật toán gom nhóm ISODATA

□ ISODATA là cải tiến của thuật toán K-means

- Số lượng các nhóm có thể được điều chỉnh tự động
 - Nếu 1 nhóm quá tản mạn → tách làm 2 nhóm
 - Nếu 2 nhóm quá gần nhau → gộp vào một nhóm
- Tính khoảng cách từ tất cả các phần tử đến tất cả các tâm → đưa ra quyết định gom nhóm hay tách nhóm

Phân vùng ảnh trực tiếp dựa trên miền ảnh

- Giả sử R biểu diễn vùng của toàn ảnh, chúng ta có thể chia R ra thành nhiều vùng con khác nhau R_1, R_2, \dots, R_n thỏa điều kiện:

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

(b) R_i là một vùng liên thông, với mọi $i = 1, 2, \dots, n$.

(c) $R_i \cap R_j = \emptyset, \forall i \neq j$.

(d) $P(R_i) = \text{TRUE}$, với mọi $i = 1, 2, \dots, n$.

(e) $P(R_i \cup R_j) = \text{FALSE}$, với mọi $i \neq j$

Phân vùng ảnh trực tiếp dựa trên miền ảnh (tiếp)

- ❑ $P(R_i)$ là một hàm logic được định nghĩa trước trên các điểm ảnh trong tập R_i và \emptyset là tập hợp rỗng.
- ❑ Điều kiện (a) để đảm bảo việc phân vùng là hoàn toàn, mỗi điểm ảnh phải thuộc vào một vùng nào đó.
- ❑ Điều kiện (b) R là một vùng liên thông.
- ❑ Điều kiện (c) để đảm bảo các vùng phải rời nhau.
- ❑ Điều kiện (d) để đảm bảo các điểm ảnh trong vùng phải thỏa một tính chất P nào đó.
- ❑ Điều kiện (e) để đảm bảo hai vùng khác nhau về tính chất P được định nghĩa trước

Phân vùng ảnh trực tiếp dựa trên miền ảnh (tiếp)

- ❑ Áp dụng khi ảnh có nhiều nhiễu → việc phát hiện biên phức tạp hoặc không thể phát hiện chính xác
- ❑ Tiêu chuẩn xác định tính đồng nhất của miền đóng vai trò rất quan trọng
- ❑ Một số tiêu chuẩn tính đồng nhất
 - Theo giá trị mức xám
 - Theo màu sắc, kết cấu ảnh
 - Theo hình dạng, theo mô hình
 - ...

Phân vùng ảnh trực tiếp dựa trên miền ảnh (tiếp)

□ Một số phương pháp

- Phương pháp lan tỏa vùng (gia tăng vùng – region growing)
- Phương pháp phân chia và kết hợp vùng

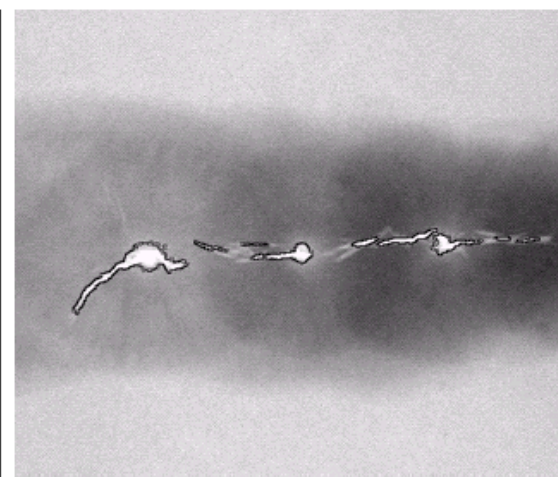
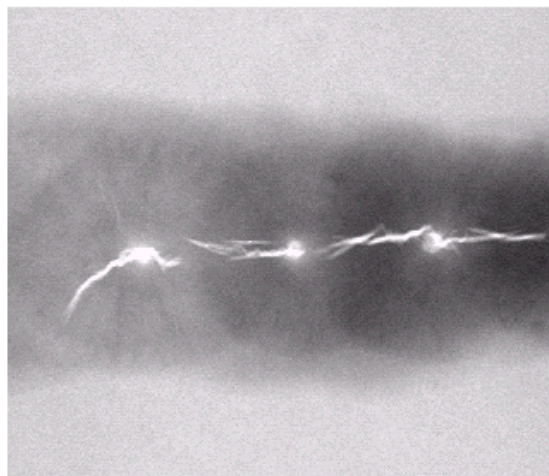
Phương pháp lan tỏa vùng

- ❑ Bắt đầu tại những điểm “hạt giống”
- ❑ Phát triển vùng bằng cách thêm vào tập các điểm “hạt giống” những điểm lân cận thỏa mãn một tính chất cho trước (như cấp xám, màu sắc, kết cấu) – Thỏa mãn hàm P
 - 4 lân cận
 - 8 lân cận

Phương pháp lan tỏa vùng

Tiêu chuẩn:

1. Giá trị sai khác tuyệt đối giữa các điểm ảnh phải nhỏ hơn 65
2. Các điểm ảnh phải là 8 lân cận với nhau và có ít nhất một điểm ảnh nằm trong vùng



Phương pháp lan tỏa vùng (tiếp)

- ❑ Ví dụ: Phân vùng áp dụng lan tỏa vùng cho ảnh sau (sự sai khác < 3 , seed point là những điểm có giá trị lớn nhất)

0	0	1	2	5	7	1	0	1	1	1	1	1	0
0	0	1	6	6	7	1	0	0	0	0	0	0	0
0	1	2	1	2	1	1	0	0	7	7	7	1	1
1	2	1	1	1	2	0	0	0	6	6	7	1	1
1	2	7	6	6	6	5	5	1	6	7	7	1	1
2	3	1	1	1	6	6	1	1	6	6	7	1	1
0	0	0	1	1	1	1	1	1	6	6	7	1	1
0	0	0	0	0	0	1	1	0	0	0	0	1	1

Phương pháp lan tỏa vùng (tiếp)

Các điểm ảnh “hạt giống”

0	0	1	2	5	7	1	0	1	1	1	1	1	0
0	0	1	6	6	7	1	0	0	0	0	0	0	0
0	1	2	1	2	1	1	0	0	7	7	7	1	1
1	2	1	1	1	2	0	0	0	6	6	7	1	1
1	2	7	6	6	6	5	5	1	6	7	7	1	1
2	3	1	1	1	6	6	1	1	6	6	7	1	1
0	0	0	1	1	1	1	1	1	6	6	7	1	1
0	0	0	0	0	0	1	1	0	0	0	0	1	1

Phương pháp lan tỏa vùng (tiếp)

Phát triển vùng.

0	0	1	2	5	7	1	0	1	1	1	1	1	0
0	0	1	6	6	7	1	0	0	0	0	0	0	0
0	1	2	1	2	1	1	0	0	7	7	7	1	1
1	2	1	1	1	2	0	0	0	6	6	7	1	1
1	2	7	6	6	6	5	5	1	6	7	7	1	1
2	3	1	1	1	6	6	1	1	6	6	7	1	1
0	0	0	1	1	1	1	1	1	6	6	7	1	1
0	0	0	0	0	0	1	1	0	0	0	0	1	1

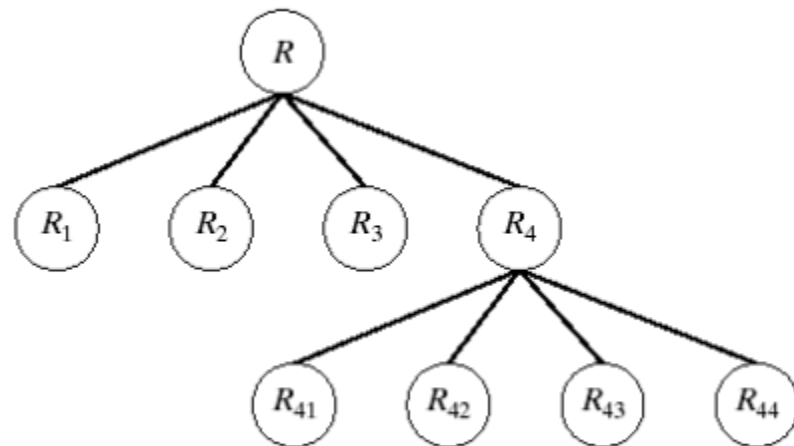
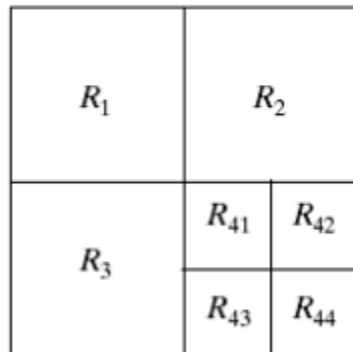
Phương pháp phân chia và kết hợp vùng

□ Ý tưởng:

- Xác định một luật $P(R_i)$ mà mỗi vùng phải thỏa mãn
- Một vùng R_i sẽ được chia thành các vùng nhỏ hơn nếu $P(R_i) = \text{FALSE}$
- Hai vùng R_i và R_j sẽ được gộp vào nhau nếu $P(R_i \cup R_j) = \text{TRUE}$
- Thuật toán dừng khi không chia và gộp được nữa

Phương pháp phân chia và kết hợp vùng (tiếp)

- Có nhiều kỹ thuật tách và hợp vùng
 - Xem xét kỹ thuật tách và hợp vùng theo cấu trúc cây tứ phân



KỸ THUẬT TÁCH VÙNG VÀ HỢP VÙNG TỬ PHÂN

a b c

FIGURE 10.43

(a) Original image. (b) Result of split and merge procedure. (c) Result of thresholding (a).



- $P(R_i) = \text{TRUE}$ nếu có ít nhất 80% các điểm trong R_i có tính chất $|z_j - m| \leq 2\sigma_i$.
- Trong đó:
 - z_j : là cấp xám của điểm ảnh thứ j trong vùng R_i .
 - m : là giá trị trung bình của vùng R_i .
 - σ_i : là độ lệch chuẩn của các cấp xám trong R_i .

TÍNH ĐỘ LỆCH CHUẨN

- Khi các vùng được gộp: tất cả các pixel trong vùng nhận giá trị trung bình của vùng

- Phương sai:
$$\sigma^2 = \frac{\sum_{j=1}^n (z_j - \bar{z})^2}{n-1}$$

- Độ lệch chuẩn:
$$\sigma = \sqrt{\frac{\sum_{j=1}^n (z_j - \bar{z})^2}{n-1}}$$