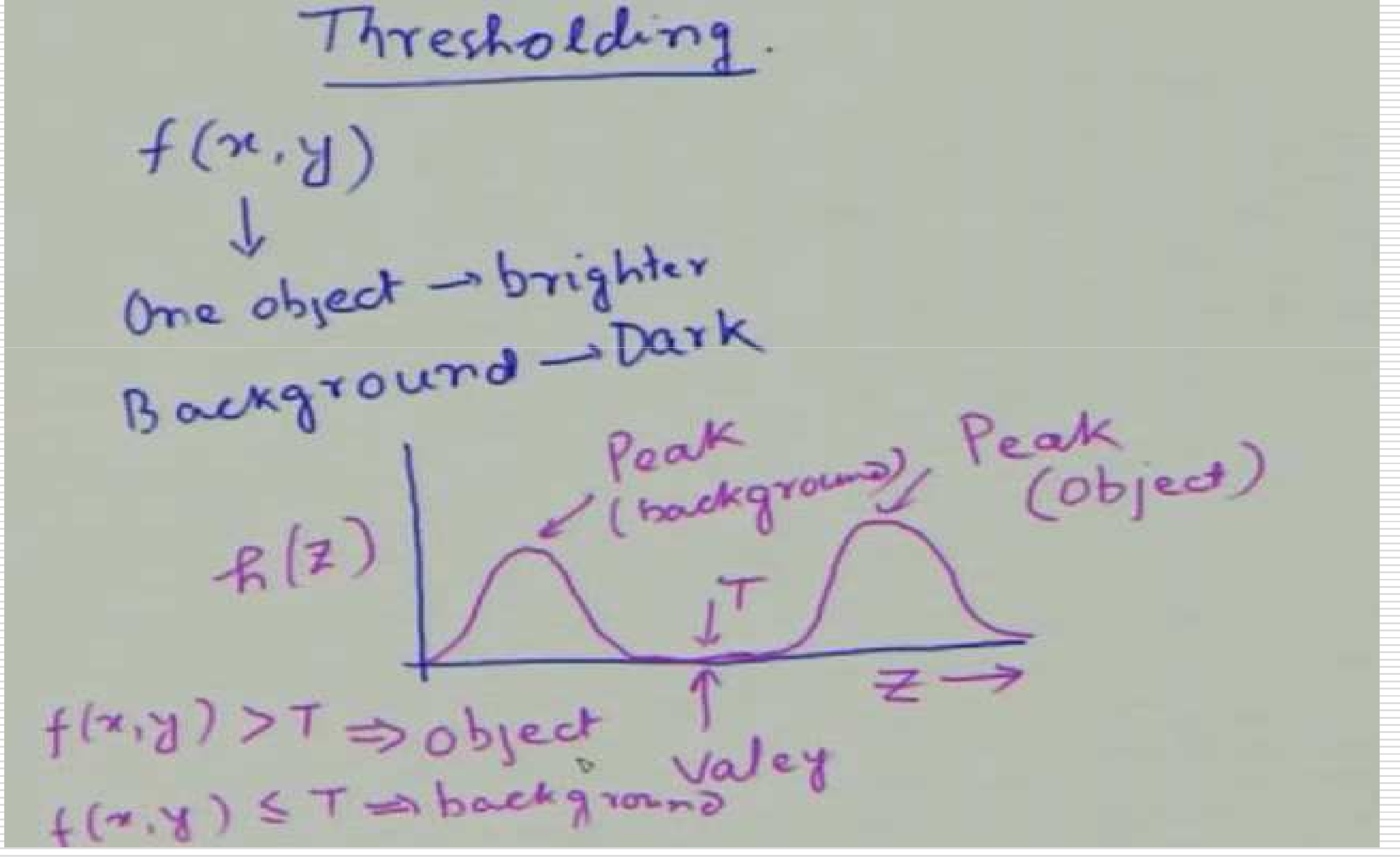
Segementation by Thresholding

Mirunalini
A. CSE

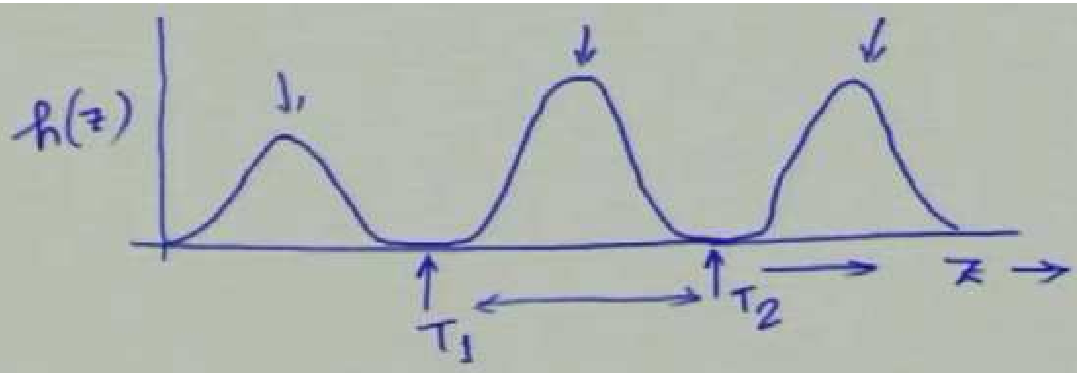
Thresholding

- Global threshold
- Local Threshold
- Adaptive/Dynamic Threshold
- Optimum Threshold

Thresholding



Multiple Thresholding



$$f(x,y) > T_2 \Rightarrow (x,y) \in O_2$$

$$T_1 < f(x,y) \leq T_2 \Rightarrow (x,y) \in O_1$$

$$f(x,y) < T_1 \Rightarrow (x,y) \in \text{Background}$$

Thresholding

$$T = T\left[x, y, p(x,y), f(x,y)\right]$$

$(x,y) \Rightarrow$ pixel location

$f(x,y) \Rightarrow$ pixel intensity at (x,y)

$p(x,y) \Rightarrow$ local property in a neighborhood centered at (x,y)

Thresholding

- T can be viewed as an operation to test the image pixels against a function T
- T = T [x, y, p(x, y) and f (x, y)].
  - (x, y) =>the pixel location in the image,
  - f (x, y)=> Pixel intensity at location (x, y)
  - p (x, y)=>some local property in a neighborhood centered at (x, y).

Thresholding Types

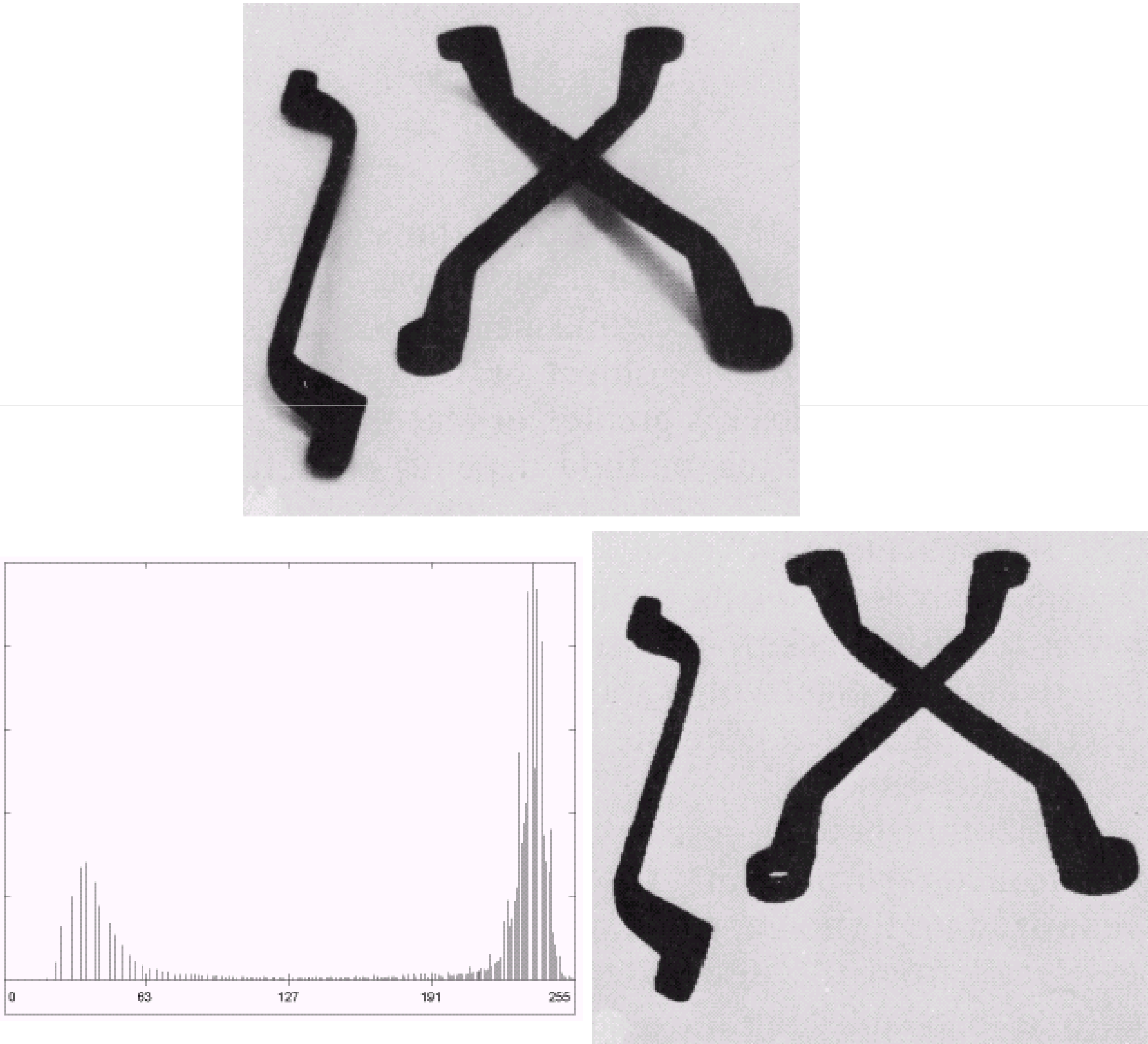
$$T[f(x,y)] \Rightarrow \text{Global}$$

$$T[f(x,y),\, p(x,y)] \Rightarrow \text{Local}$$

$$T[(x,y),\, f(x,y),\, p(x,y)] \Rightarrow \text{Adaptive/Dynamic}$$

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

Thresholding
Basic Global Thresholding



a
b c

FIGURE 10.28 (a) Original image. (b) Image histogram. (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

Basic Global Thresholding

1. Choose initial threshold $T0$

2. Define $f(x,y) > T0$ as background and $f(x,y) < T0$ as foreground

3. Calculate mean for background μbg and foreground μfg

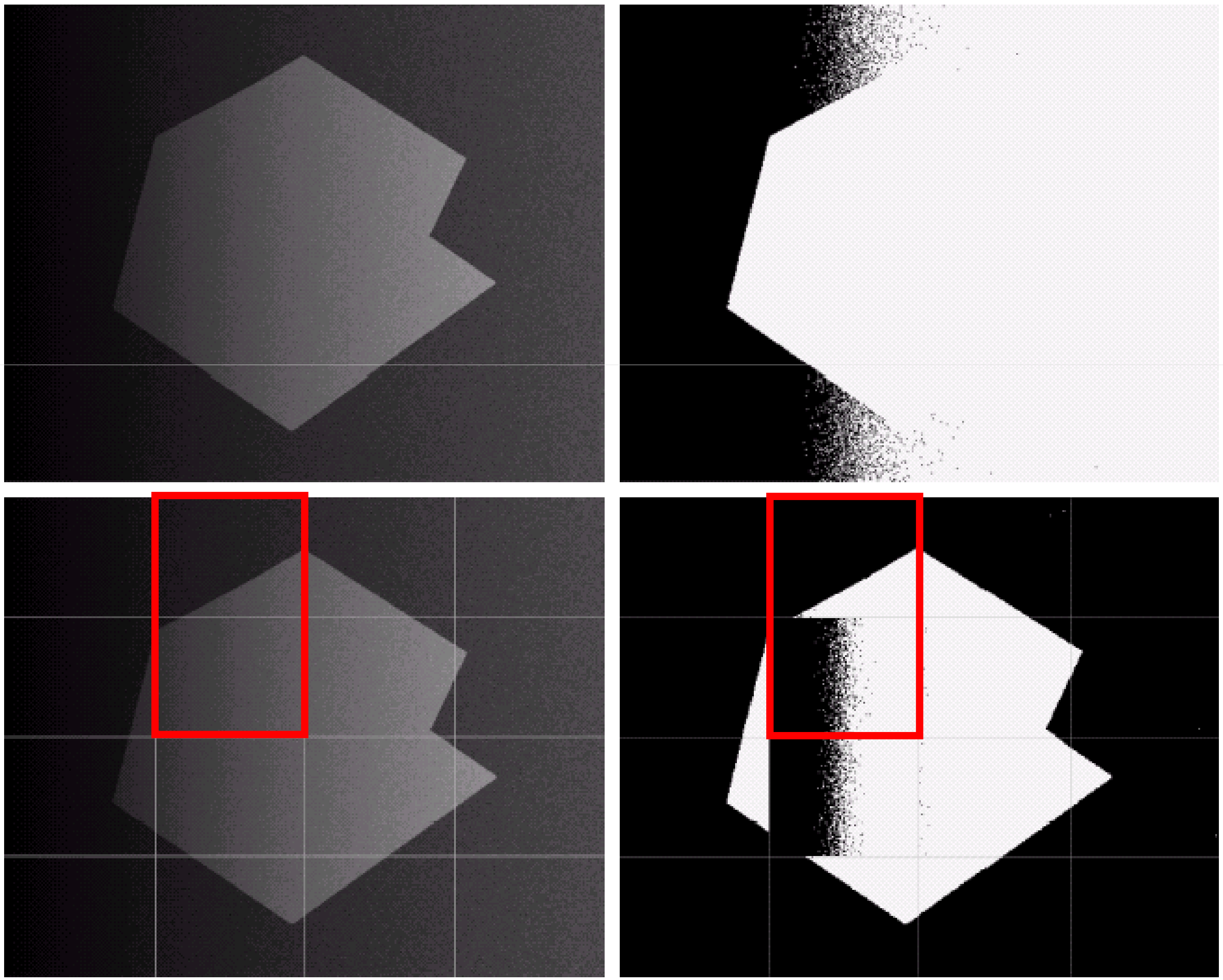
4. Set next threshold $Ti = (\mu bg+\mu fg)/2$

5. Repeat 2.–4. until stopping criteria, $Ti = Ti-1$, is fulfilled

Thresholding
Basic Adaptive Thresholding

a b
c d

FIGURE 10.30
(a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.



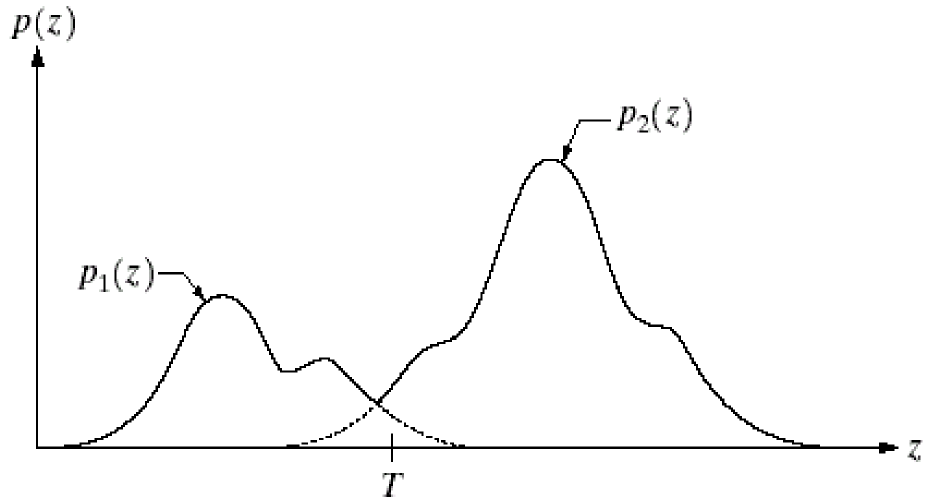
Local or Adaptive Thresholding

- Threshold which is position dependent because every sub image has a particular position;
- Threshold selection is position dependent, it becomes an adaptive thresholding operation

Thresholding - Optimal

- This method treats pixel values as probability density functions.
- The goal of this method is to minimize the probability of misclassifying pixels as either object or background.
- There are two kinds of error:
  - mislabeling an object pixel as background, and
  - mislabeling a background pixel as object.

FIGURE 10.32 Gray-level probability density functions of two regions in an image.



Otsu's method - steps

(1) **Compute normalized histogram of the image,** $p_i = \dfrac{n_i}{MN}, \; i = 0, \ldots, L-1$

(2) **Compute cumulative sums,** $P_1(k) = \sum_{i=0}^{k} p_i, \; k = 0, \ldots, L-1$

(3) **Compute cumulative means,** $m(k) = \sum_{i=0}^{k} i\,p_i, \; k = 0, \ldots, L-1$

(4) **Compute global intensity mean,** $m_G = \sum_{i=0}^{L-1} i\,p_i$

(5) **Compute between-class variance,** $\sigma_B^2(k) = \dfrac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}, \; k = 0, .., L-1$

(6) **Obtain the Otsu threshold,** k^***, that is the value of** k **for which** $\sigma_B^2(k^*)$ **is a maximum – if this maximum is not unique, obtain** k^* **by avaraging the values of** k **that correspond to the various maxima detected**

(7) **Obtain the separability measure** $\eta(k^*) = \dfrac{\sigma_B^2(k^*)}{\sigma_G^2}$

Smoothing to Improve Thresholding

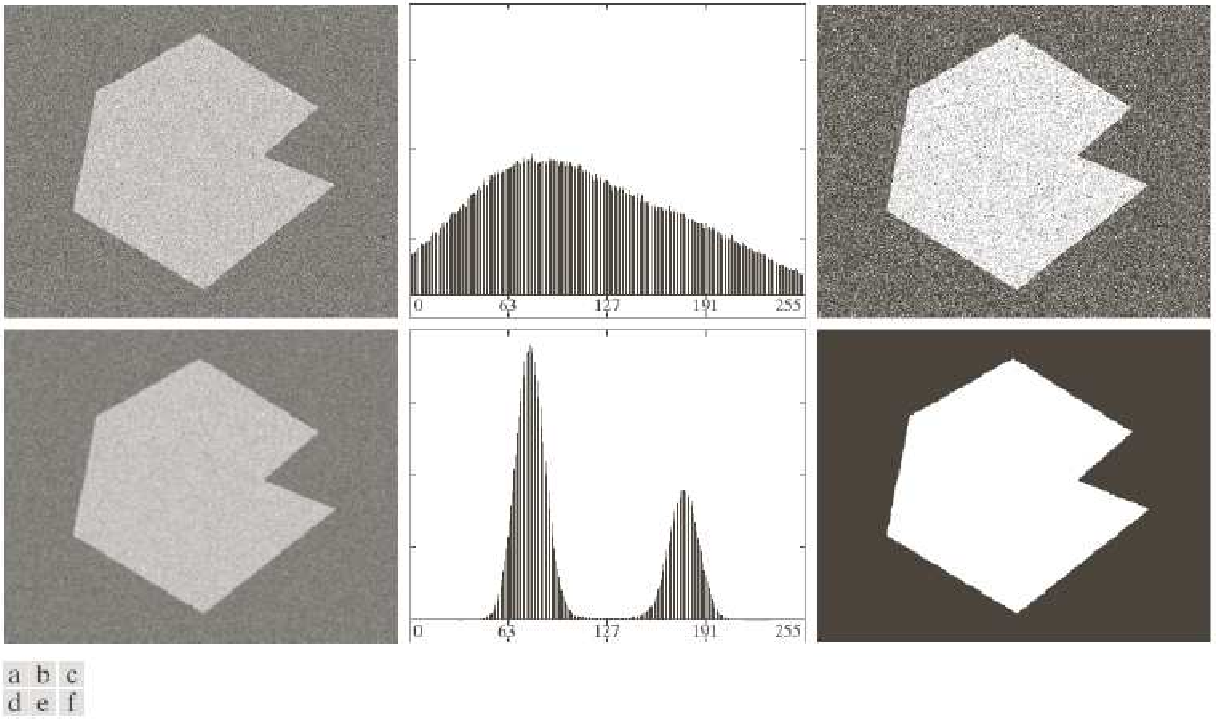


FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Smoothing –Fails for smaller objects

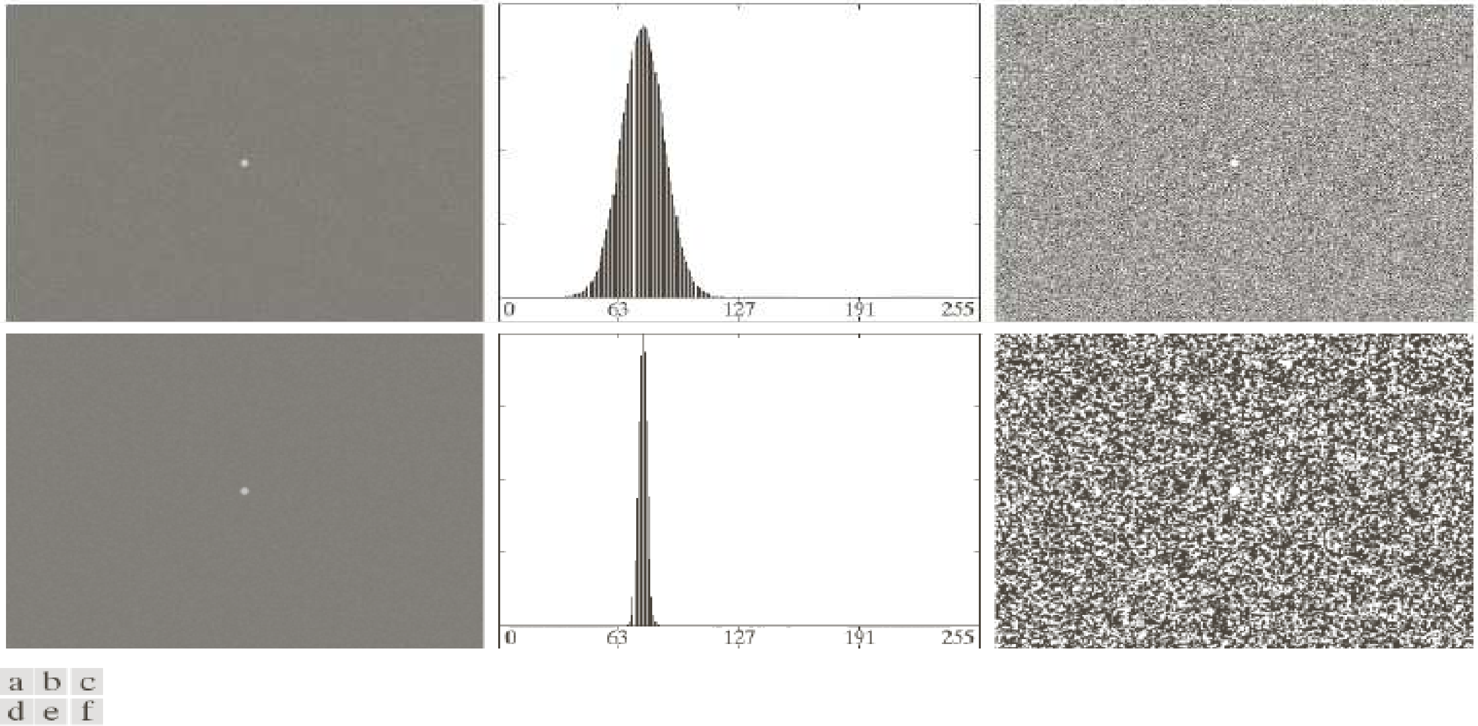
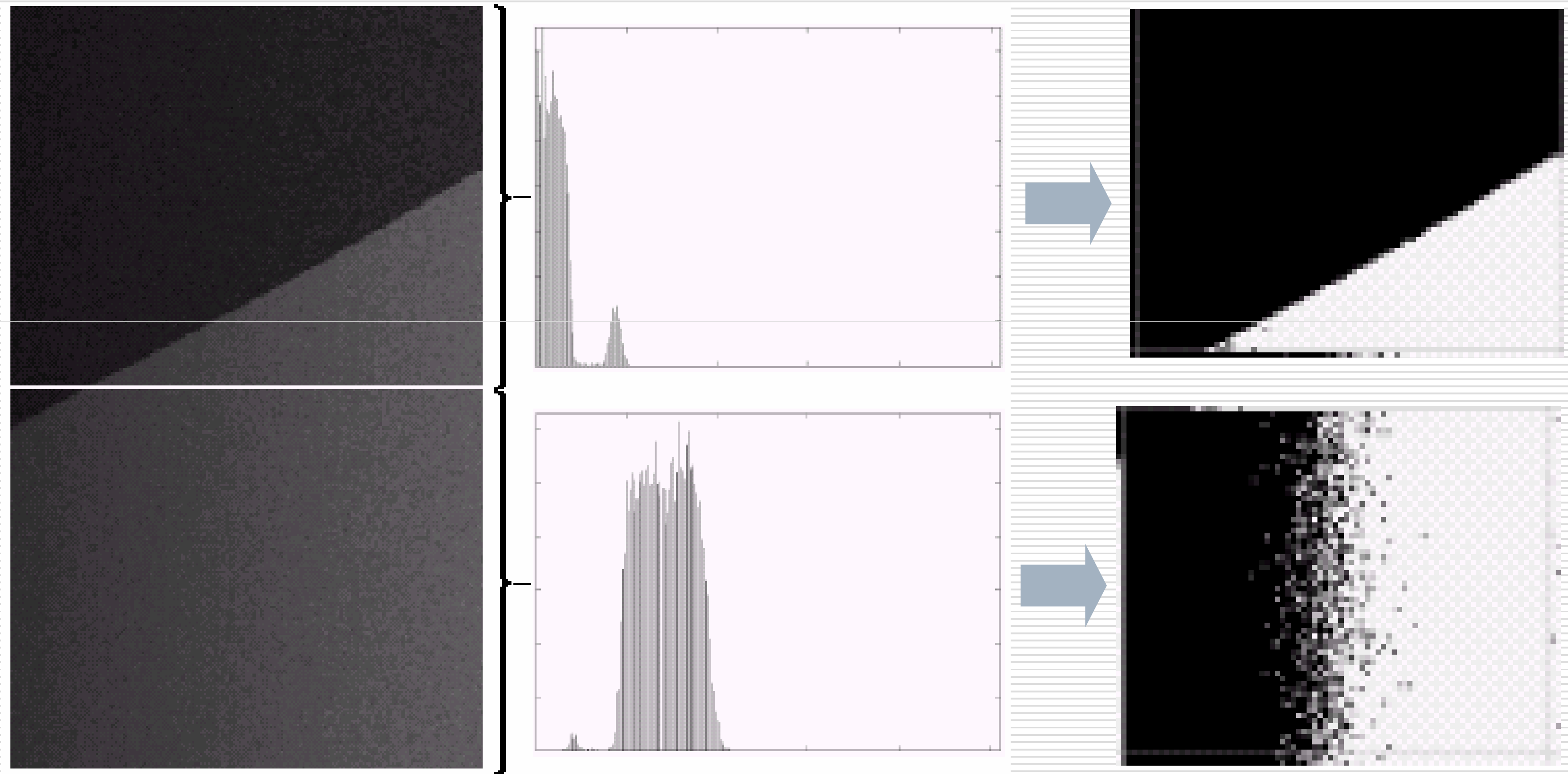


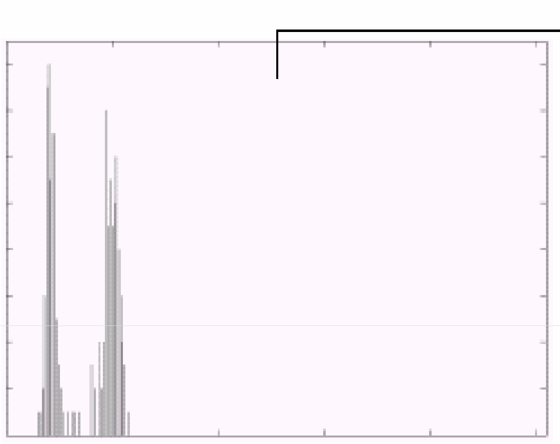
FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

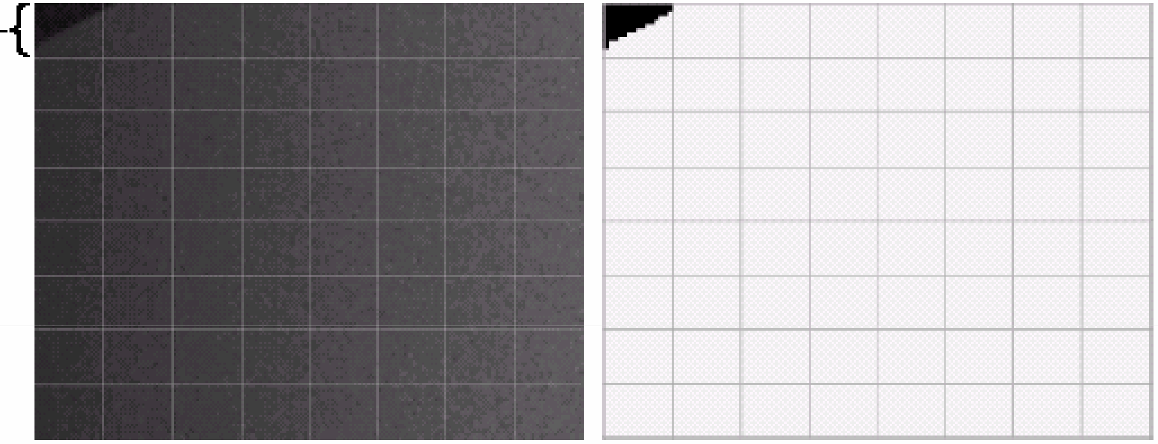
Thresholding
Basic Adaptive Thresholding



How to solve this problem?

Thresholding
Basic Adaptive Thresholding





	a	b
		c
e	d	f

Answer: subdivision

FIGURE 10.31 (a) Properly and improperly segmented subimages from Fig. 10.30. (b)–(c) Corresponding histograms. (d) Further subdivision of the improperly segmented subimage. (e) Histogram of small subimage at top, left. (f) Result of adaptively segmenting (d).

Edges- To improve Global Thresholding

- "Good " Threshold is obtained when histogram peaks tall, narrow, symmetric and separated by deep valleys
- Image of small object on a large background, lead to failure of thresholding
  - Dominated by large peak because of one type of pixels
  - Solution: consider pixels only at edges between object and background
  - Histogram will have peaks of approximately same height
  - Probability of the pixels belonging to object is equal to probability belonging to background
  - Improves the symmetry of histogram modes

Edges- To improve Global Thresholding

(1) Compute an edge image as either the magnitude of the gradient, or the absolute value of the Laplacian, of $f(x,y)$

(2) Specify a threshold value, T

(3) Threshold the image from step (1) using the threshold from step (2) to produce a binary image, $g_T(x,y)$, which is used as a mask image in the following step to select pixels from $f(x,y)$ corresponding to "strong" edge pixels

(4) Compute a histogram using only the pixels in $f(x,y)$ that correspond to the locations of the 1-valued pixels in $g_T(x,y)$

(5) Use the histogram from step (4) to segment $f(x,y)$ globally using, for example, Otsu's method

Edges- To improve Global Thresholding

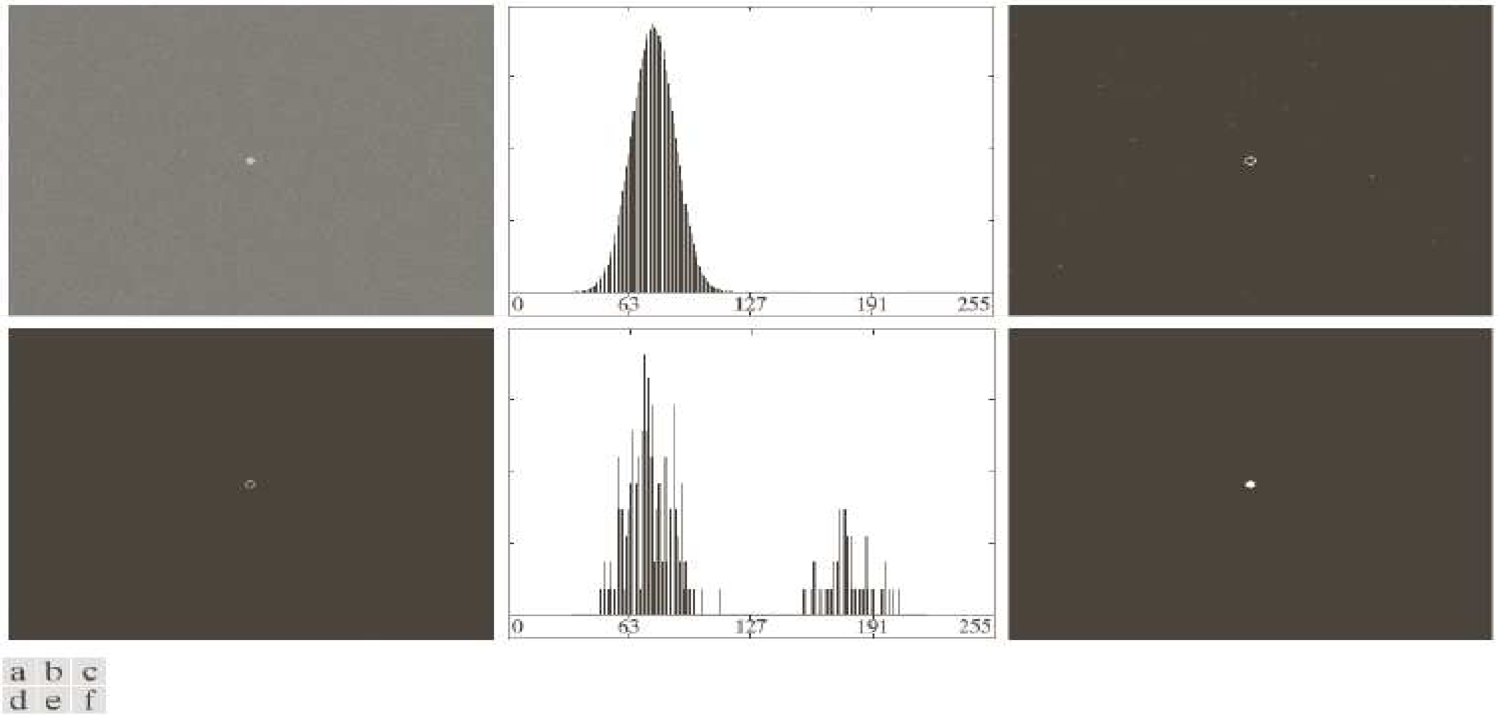


FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based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

Edges- To improve Global Thresholding

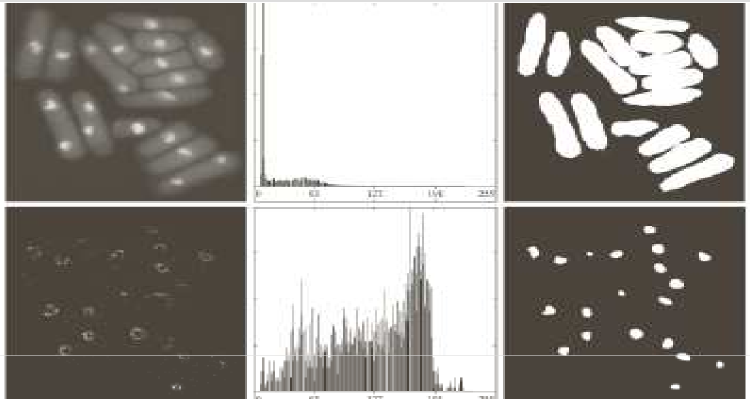


FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

FIGURE 10.44 Image in Fig. 10.43(a) segmented using the same procedure as explained in Figs. 10.43(d)–(f), but using a lower value to threshold the absolute Laplacian image.

Multiple Thresholding

- Otsu's method can be extended to a
- multiple thresholding method
- Between-class variance can be reformulated as

$$\sigma_B^2 = \sum_{k=1}^{K} P_k (m_k - m_G)^2 \qquad (10.3\text{-}21)$$

$$P_k = \sum_{i \in C_k} p_i \qquad (10.3\text{-}22)$$

$$m_k = \frac{1}{P_k} \sum_{i \in C_k} i p_i \qquad (10.3\text{-}23)$$

Multiple Thresholding

- The K classes are separated by K-1 thresholds and these optimal thresholds can be solved by maximizing

$$\sigma_B^2(k_1^*, k_2^*, \ldots, k_{K-1}^*) = \max_{0 \le k_1 \le k_2 \le \ldots k_{n-1} \le L-1} \sigma_B^2(k_1, k_2, \ldots, k_{K-1})$$

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2 \quad (10.3\text{-}25)$$

$$P_1 = \sum_{i=0}^{k_1} p_i, \quad P_2 = \sum_{i=k_1+1}^{k_2} p_i, \quad P_3 = \sum_{i=k_2+1}^{L-1} p_i \quad (10.3\text{-}26)$$

$$m_1 = \frac{1}{P_1}\sum_{i=0}^{k_1} ip_i, \quad m_2 = \frac{1}{P_2}\sum_{i=k_1+1}^{k_2} ip_i, \quad m_3 = \frac{1}{P_3}\sum_{i=k_2+1}^{L-1} ip_i \quad (10.3\text{-}27)$$

Multiple Thresholding

- The following relationships hold:

$$P_1 m_1 + P_2 m_2 + P_3 m_3 = m_G \;, where \;\; P_1 + P_2 + P_3 = 1 \qquad (10.3\text{-}28)$$

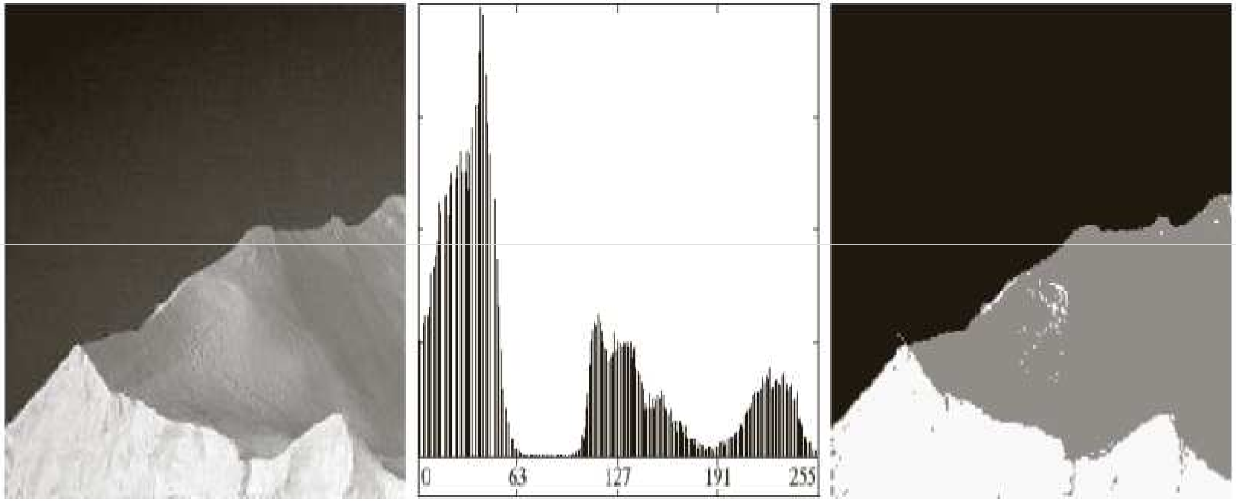
The optimum thresholds can be found by :

$$\sigma_B^2(k_1^*, k_2^*) = \max_{0 \le k_1 \le k_2 \le L-1} \sigma_B^2(k_1, k_2) \qquad (10.3\text{-}30)$$

$$g(x,y) = \begin{cases} a, & \text{if } f(x,y) \le k_1^* \\ b, & \text{if } k_1^* < f(x,y) \le k_2^* \\ c, & \text{if } f(x,y) > k_2^* \end{cases} \qquad \eta(k_1^*, k_2^*) = \frac{\sigma_B^2(k_1^*, k_2^*)}{\sigma_G^2}$$

$$(10.3\text{-}32)$$

Multiple Thresholds



a b c

FIGURE 10.45 (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

Region-Based Segmentation

- Edges and thresholds sometimes do not give good results for segmentation.
- Region-based segmentation is based on the connectivity of similar pixels in a region.
  - Each region must be uniform.
  - Connectivity of the pixels within the region is very important.
- There are two main approaches to region-based segmentation: region growing and region splitting.

Region-Based Segmentation
Basic Formulation

- Let R represent the entire image region.
- Segmentation is a process that partitions R into subregions, $R_1, R_2, \ldots, R_n$, such that

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

$$\text{(b)} \quad R_i \text{ is a connected region, } i = 1, 2, \ldots, n$$

$$\text{(c)} \quad R_i \cap R_j = \phi \text{ for all } i \text{ and } j, i \neq j$$

$$\text{(d)} \quad P(R_i) = \text{TRUE for } i = 1, 2, \ldots, n$$

$$\text{(e)} \quad P(R_i \cup R_j) = \text{FALSE for any adjacent regions } R_i \text{ and } R_j$$

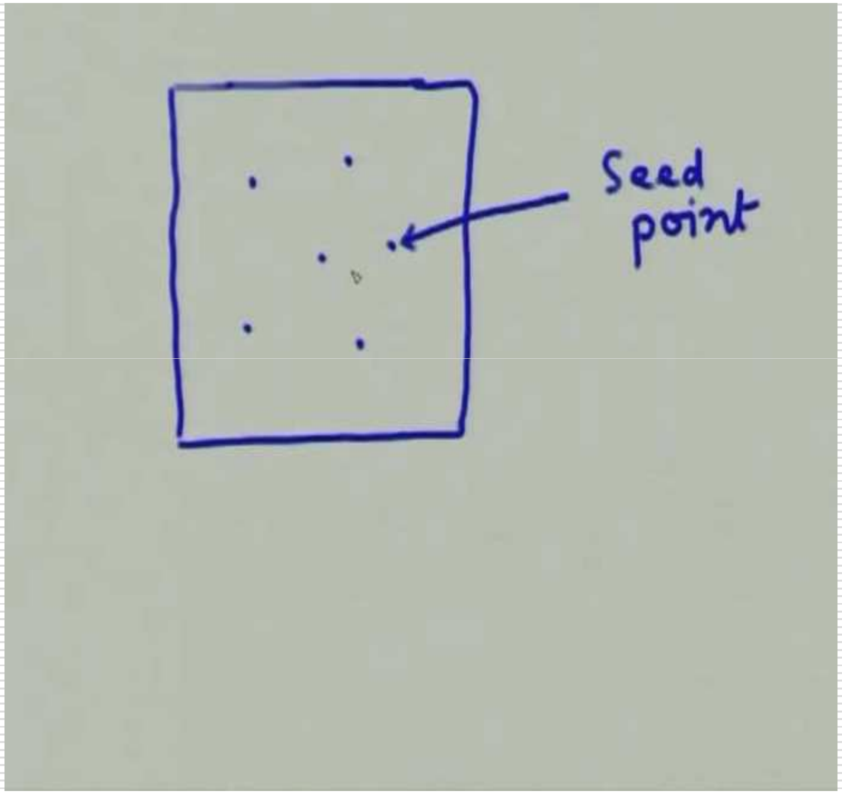
where $P(R_k)$: a logical predicate defined over the points in set R_k

For example: $P(R_k)$=TRUE if all pixels in R_k have the same gray level.

Region-Based Segmentation
Basic Formulation

- *Region growing is a procedure that groups pixels or subregions into larger regions.*
- *The simplest of these approaches is* ***pixel aggregation, which starts with a set of "seed" points and from these grows regions by appending to each seed points those neighboring pixels that have similar properties (such as gray level, texture, color, shape).***
- *Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.*

Region-Based Segmentation
Basic Formulation



- Select a seed point
- Group pixels based on predefined criteria
- Append to each seed neighboring pixels that have predefined properties
- Problems in region growing:
    - Descriptors –misleading use connectivity properties
    - Stopping criteria
    - Non trivial to find good starting points,
    - difficult to automate and needs good criteria for similarity

Region Growing– Algorithm

$f(x, y)$: input image array

$S(x, y)$: seed array containing 1s (seeds) and 0s

$Q(x, y)$: predicate

Region Growing- Algorithm

- Find all connected components in S(x,y) and erode each connected components to one pixel
  - Label all such pixels found as 1.
  - All other pixels in s labeled as 0
- Form an image fq ,let fq(x,y)=1 if the input satisfies the predicate Q other wise fq=0
- Let g be an image formed by appending to each seed point in S all the 1-values points in fq are 8-connected to that seed point
- Label each connected component in g with a different region label thus the segmented image obtained by region growing

Region Growing– Algorithm

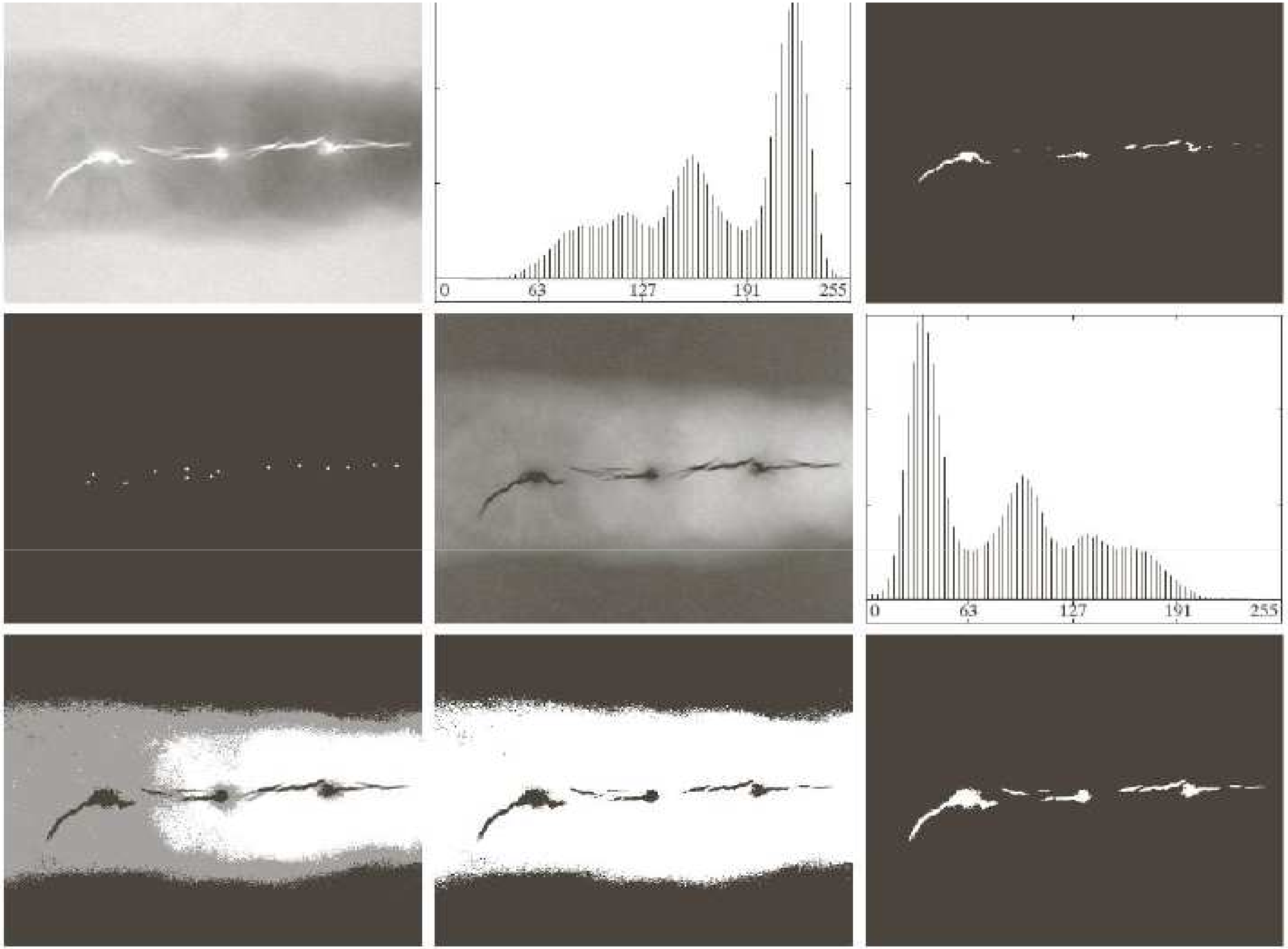
10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

4-connectivity

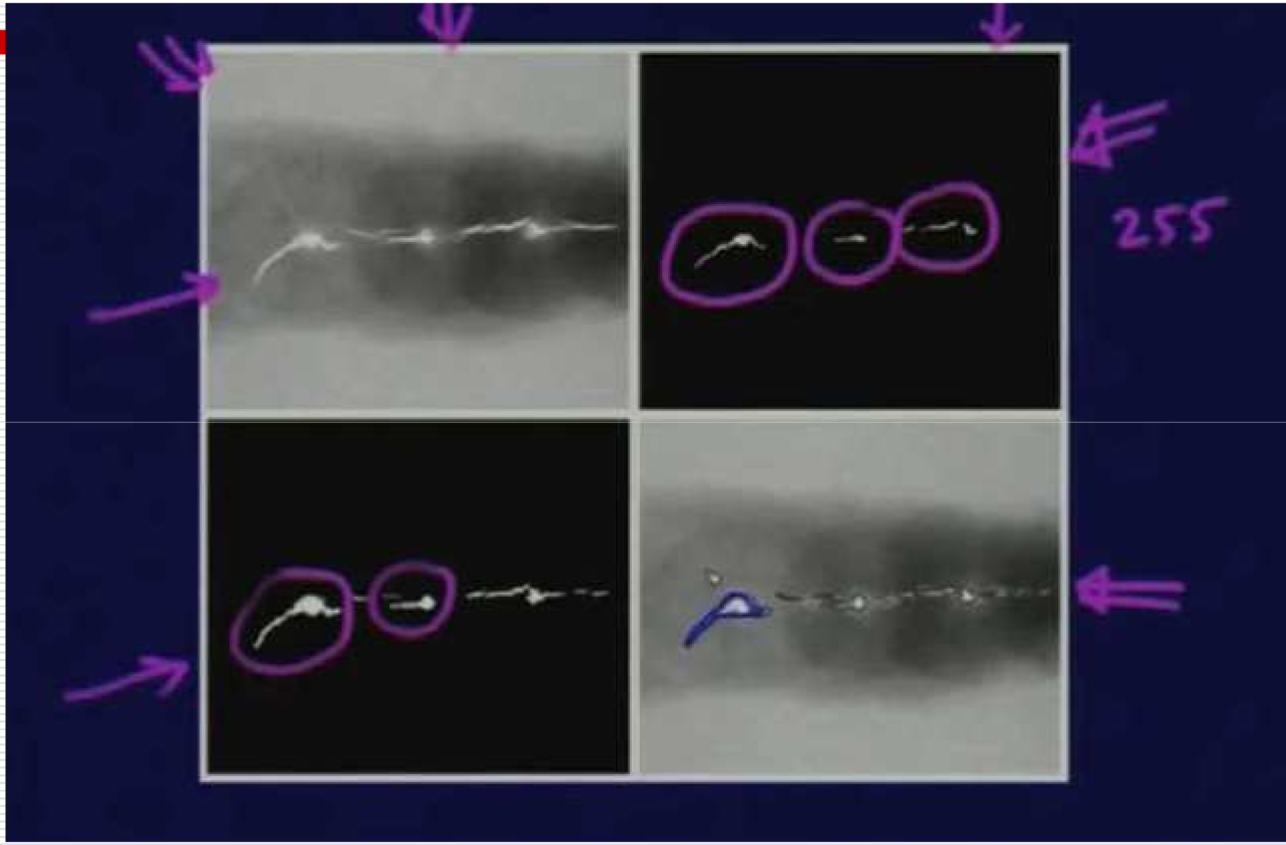
10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

8-connectivity



a b c
d e f
g h i

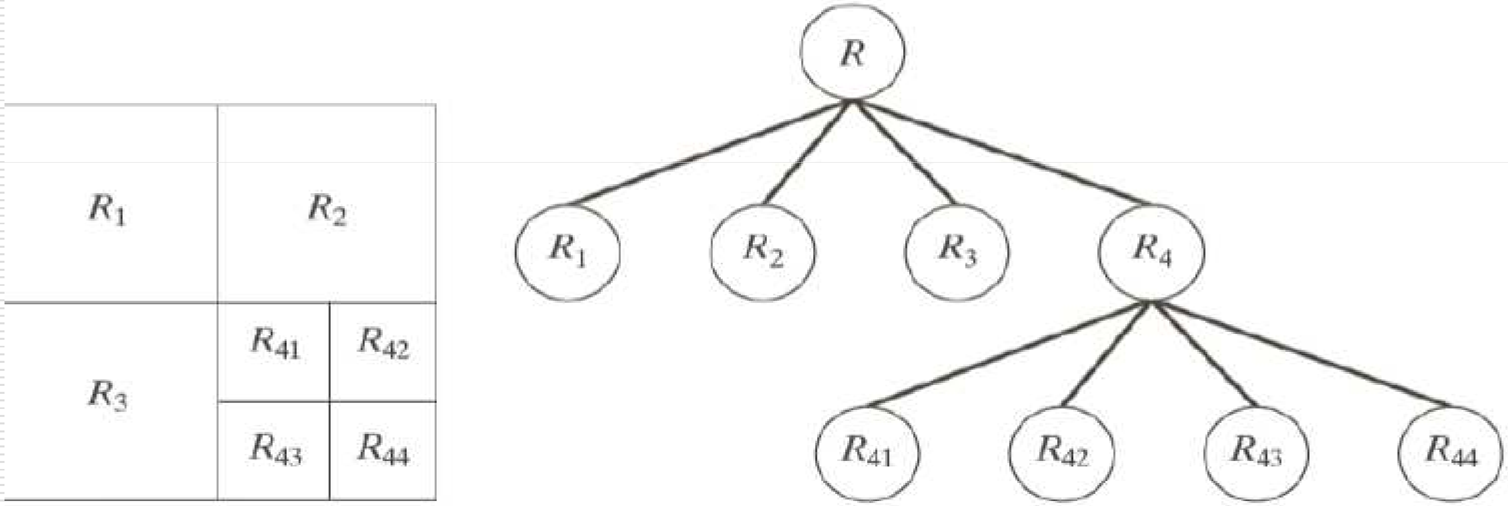
FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference image thresholded with the smallest of the dual thresholds. (i) Segmentation result obtained by region growing. (Original image courtesy of X-TEK Systems, Ltd.)


255

Region splitting and merging

Set up criteria for what is a uniform area (e.g. mean, variance, bimodality
of histogram, texture, etc.).
2. Start with the full image and split it into four sub-images.
3. Check each sub-image. If it is not uniform, split it again into four subimages.
4. Repeat 3. until no more splitting is performed.
5. Compare sub-images with the neighboring regions and merge, if they are
uniform.
6. Repeat 5. until no more merging is performed.
The method is also called quad-tree division

Region splitting and merging



Region splitting and merging