Introduction to Digital Image Processing

Dr. Kundan Kumar
Ph.D. (IIT Kharagpur)
Associate Professor
ECE Department (Cabin - E139)

Institute of Technical Education & Research (ITER)
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India-751030

Image Segmentation: Part II

Edge Linking and Boundary Detection

- Ideally, edge detection yield sets of pixels lying only on edges.
- In practice these pixels seldom characterize edges completely because of noise, non-uniform illumination, etc.
- Edge linking algorithms are used to assemble edge pixels into meaningful edges and/or region boundary.
- Three fundamental approach to edge linking
  - **Local processing**: requires knowledge about edge points in local region
  - **Regional processing**: requires points on the boundary of a region
  - **Global processing**: works with an entire edge image.

Local processing

- Edge linking (local processing)
  - Similarity of two edge pixels at (x, y) and (s, t):

$$|M(s,t) - M(x,y)| \leqslant E$$

$$|\alpha(s,t) - \alpha(x,y)| \leqslant A$$

  - Connect if both condition satisfied.
- Computational expensive.

Local processing

Simple algorithm

1. Compute $M(x,y)$ and $\alpha(x,y)$ for input image.
2. Form a binary image $g(x,y)$:

$$g(x,y) = \begin{cases} 1 & M(x,y) > T_M \text{ AND } \alpha(x,y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

   where T_M: Threshold, A: Specified angle direction, and $\pm T_A$: acceptable direction margin
3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) that do not exceed a specified length, K.
4. To detect gaps in any other direction, θ, rotate g by this angle, and apply the horizontal scanning procedure in Step 3. Rotate the result back by $-\theta$.

Edge linking using local processing

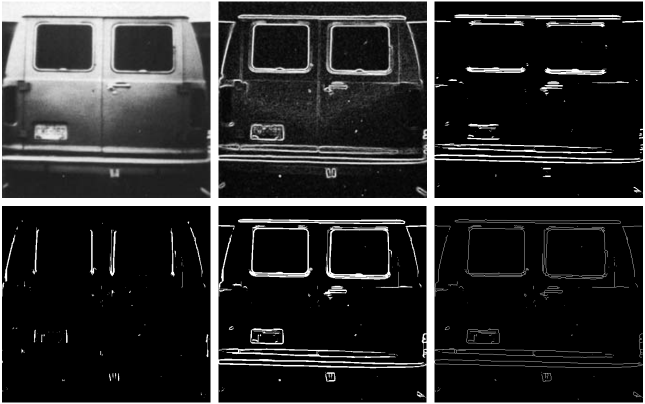


Figure: (a) A 534×566 image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning.

Regional processing: Polygon Fit

- Fit a polygon to a set of points.

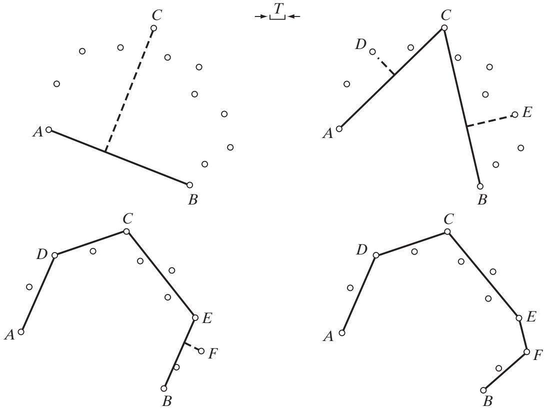


Figure: Iterative polygonal fit algorithm.

Regional processing: Polygon Fit

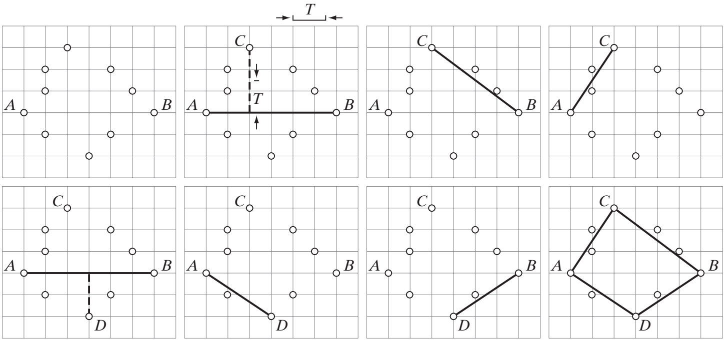
An algorithm for finding a polygonal fit to open and closed curves may be stated as follows:

1. Let P be a sequence of ordered, distinct, 1-valued points of a binary image. Specify two starting points, A and B. These are the two starting vertices of the polygon.
2. Specify a threshold T, and two empty stacks, OPEN and CLOSED.
3. If the points in P correspond to a closed curve, put A into OPEN and put B into OPEN and into CLOSED. If the points correspond to an open curve, put A into OPEN and B into CLOSED.
4. Compute the parameters of the line passing from the last vertex in CLOSED to the last vertex in OPEN.

Regional processing: Polygon Fit

5. Compute the distances from the line in Step 4 to all the points in P whose sequence places them between the vertices from Step 4. Select the point, $V_{\max}$, with the maximum distance, $D_{\max}$ (ties are resolved arbitrarily).
6. If $D_{\max} > T$, place $V_{\max}$ at the end of the OPEN stack as a new vertex. Go to Step 4.
7. Else, remove the last vertex from OPEN and insert it as the last vertex of CLOSED.
8. If OPEN is not empty, go to Step 4.
9. Else, exit. The vertices in CLOSED are the vertices of the polygonal fit to the points in P

Regional Processing: Polygon Fit



CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D
B, C, A	B, D	(AD)	—
B, C, A, D	B	(DB)	—
B, C, A, D, B	Empty	—	—

TABLE 10.1 Step-by-step details of the mechanics in Example 10.11.

Global processing: Hough Transform

- How to find subsets of point that lie on straight line?
  - Find first all lines determined by every pair of points
  - Then find all subsets of points that are close to particular lines
  - This approach involves finding $n(n-1)/2^2$ lines
  - Then performing $(n)(n(n-1)/2)^3$ comparisons of every point to all lines.
- Hough Transform proposed by Hough in 1962.
  - Consider a point (x_i, y_i) in xy-plane
  - General equation of straight line in slope-intercept form $y_i = ax_i + b$
  - Infinite number of line pass through this straight line for varying values of a and b
  - Parameter space: $b = -x_i a + y_i$ in ab-plane
  - Where a is slope and b is intercept.

Global processing: Hough Transform

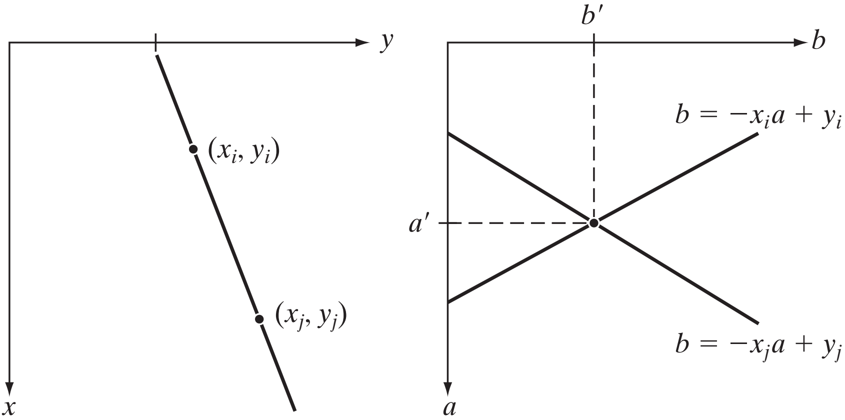
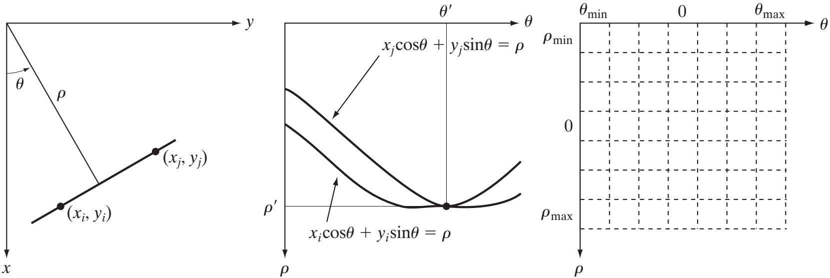


Figure: (a) xy-plane (b) Parameter space

Global processing: Hough Transform

- Practical difficulty:
  - The slope of line approaches infinity as the line approaches the vertical direction.
  - So, use normal representation

$$x\cos\theta + y\sin\theta = \rho$$



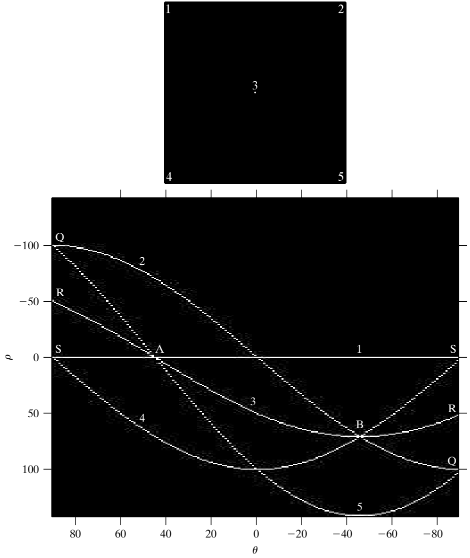
Global processing: Hough Transform

- Computation
  - Subdivide the $\rho\theta$ parameter space into *accumulator cells*.
  - $(\theta_{\min}, \theta_{\max}) \Rightarrow -90^o \leq \theta \leq 90^o$
  - $(\rho_{\min}, \rho_{\max}) \Rightarrow -D \leq \rho \leq D$
  - Where D is the maximum distance between opposite corners in an image.
- Initialize the accumulator value $A(i, j)$ corresponding to cell at coordinate (i, j) (associated with ρ_i, θ_j)
- For every non-background point (x_k, y_k) then find corrsponding value of (ρ_p, θ_q)

$$A(p, q) = A(p, q) + 1$$

- Te the end of procedure, a value of P in $A(i, j)$ means that P points in the xy-plane lie on the line $x \cos\theta_j + y \sin\theta_j = \rho_i$.

Global processing: Hough Transform



a
b

FIGURE 10.33
(a) Image of size 101 × 101 pixels, containing five points.
(b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

Thresholding

- Simple and computationally efficient.
- Intensity Thresholding
  - $f(x,y) > T$ then (x,y) is belong to the object, else (x,y) is belong to the background.
    - Bi-level (T):

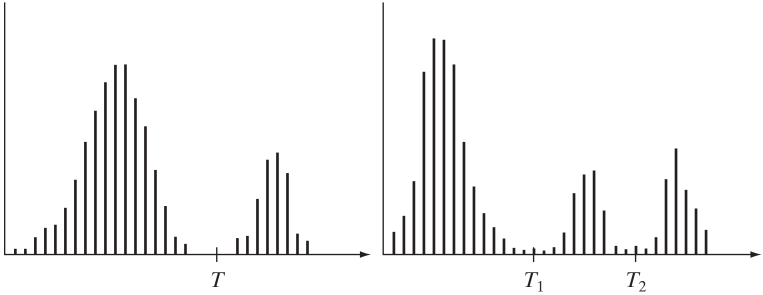
$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leqslant T \end{cases}$$

    - Multi-level $(T_1, T_2, \cdots, T_n)$

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

- Challenge is threshold selection: Histogram

Bi-Modal and Multi-modal Histogram



- If T depends on the spatial coordinate (x, y) themselves, then variable thresholding is often referred to as *dynamic* or *adaptive thresholding*

Noise effect on image Thresholding

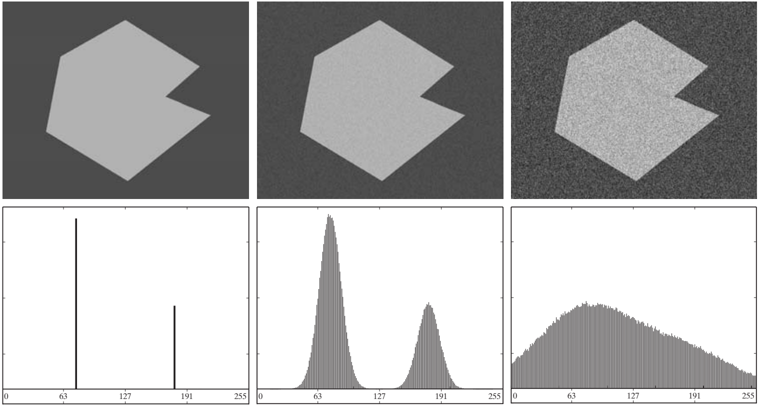


Figure: (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)-(f) Corresponding histograms.

Illumination and Reflectance Effect on image Thresholding

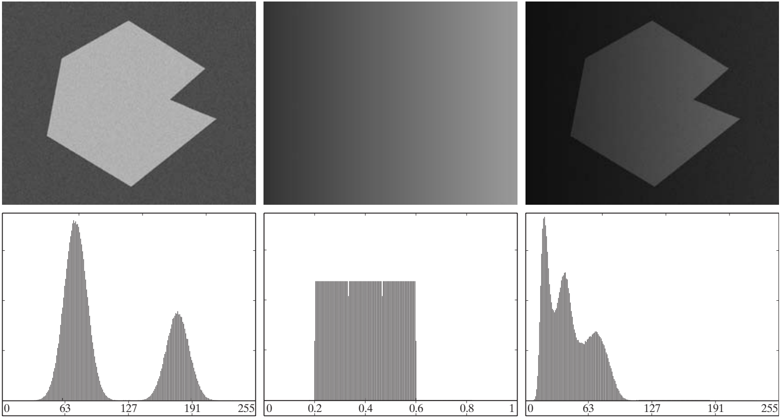


Figure: (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)-(f) Corresponding histograms.

Basic Global Thresholding

A Heuristic approach:

1. Select an initial estimate for the global threshold T.
2. Segment image using T which will produce two groups of pixels: G_1 consisting of all pixels with intensity values $> T$, and G_2 consisting of pixels with values $\leq T$.
3. Compute the average (mean) intensity values m_1 and m_2 for the pixels in G_1 and G_2, respectively.
4. Compute a new threshold value

$$T = \frac{1}{2}(m_1 + m_2)$$

5. Repeat Steps 2-4 until small changed in successive T values (predefined ΔT).

Basic Global Thresholding: Example

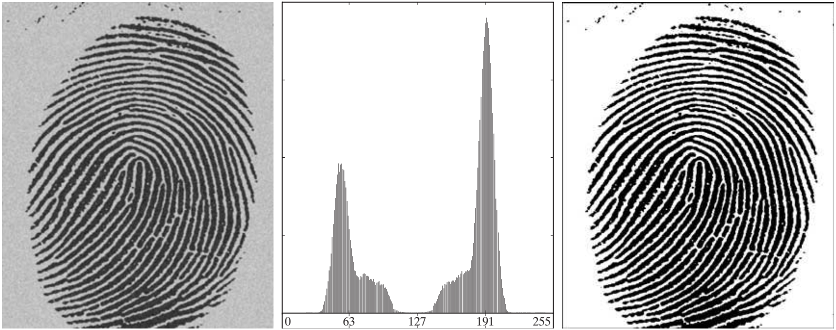


Figure: (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity)

Optimal Global Thresholding: Otsu's Methods

- Basic idea:
  - Separability of two class of data:
    - The method is optimum in the sense that it maximizes the *between-class variance*.
    - Between-class variance is a measure of separability between classes
    - Small distance between each class samples (Within-class distance)
    - Global variance to Between-Class variance ratio
  - Entirely based on computations performed on the histogram of an image, an easily obtainable 1-D array.

Otsu's algorithm

1. Compute the normalized histogram of the input image. Denote the components of the histogram by $p_i, i = 0, 1, 2, \ldots, L-1$.
2. Compute the cumulative sums $P_1(k)$, for $k = 0, 1, \ldots, L-1$ using
$$P_1(k) = \sum_{i=0}^{k} p_i$$
3. Compute the cumulative means, $m(k)$, for $k = 0, 1, \ldots, L-1$, using
$$m(k) = \sum_{i=0}^{k} ip_i$$
4. Compute the global intensity mean, m_G, using
$$m_G = \sum_{i=0}^{L-1} ip_i$$

Otsu's algorithm

2. 3. Compute the cumulative means, for using Eq. (10.3-8). 4. Compute the global intensity mean, using (10.3-9). 5. Compute the between-class variance, for using Eq. (10.3-17). 6. Obtain the Otsu threshold, as the value of for which is maxi- mum. If the maximum is not unique, obtain by averaging the values of corresponding to the various maxima detected. 7. Obtain the separability measure, by evaluating Eq. (10.3-16) at

Thresholding Example

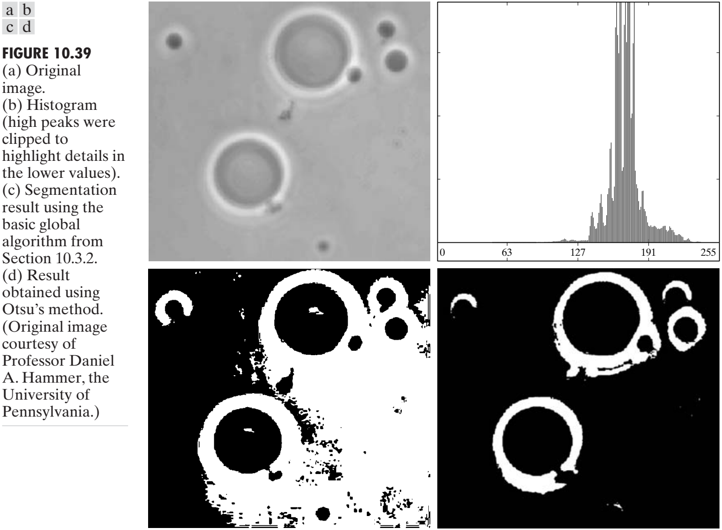
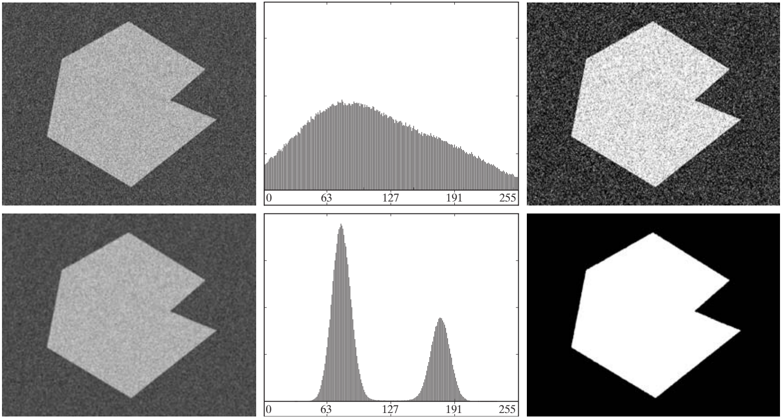


FIGURE 10.39 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

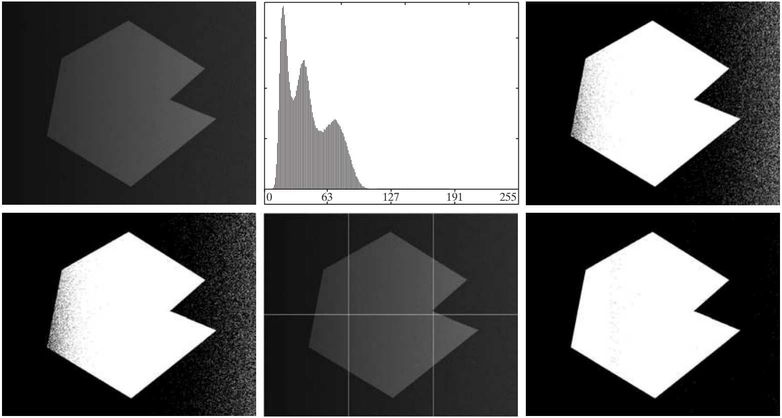
Otsu's Algorithm: Noise Effect



a b c
d e f

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.

Variable Thresholding: Image partitioning

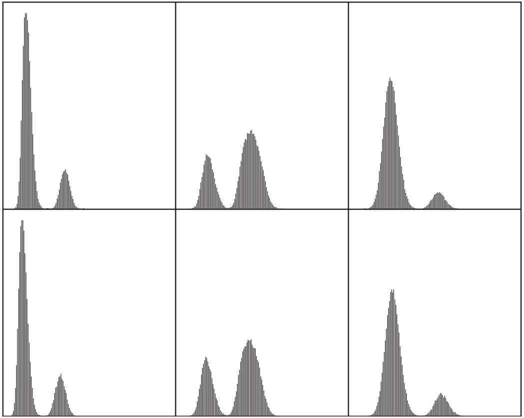


a	b	c
d	e	f

FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

Variable Thresholding: Image partitioning

FIGURE 10.47
Histograms of the six subimages in Fig. 10.46(e).

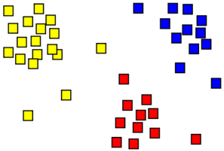


Multivariate Segmentation

- A multiple sensor (R/G/B, Multi-Band, and etc.)
- Image data: $z \in \Re^N$
  - Threshold:

$$g = \begin{cases} 1 & dist(z,a) < T \\ 0 & otherwise \end{cases}$$

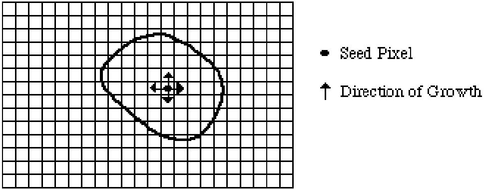
  - a: A specific color
- Segmentation
  - A cluster task



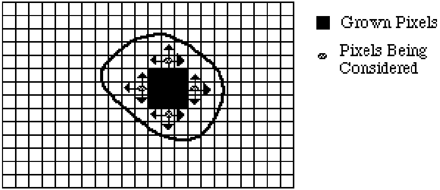
Region Growing

- Region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria for growth.
- Basic steps:
  - Select a start (seed) point
  - Grow the point based on a predefined property (e.g. connectivity) similar to the seed.
- Seed point can be one or more number of point depending to type of problem.
- For cluster of pixels, the centroid of the cluster can be used as seed.

Region Growing: Basic steps



(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

How to choose the seed points

- It depends on the nature of the problem.
- If targets need to be detected using infrared images for example, choose the brightest pixel(s).
- Without a-priori knowledge, compute the histogram and choose the gray-level values corresponding to the strongest peaks.

Similarity criteria/ Growing criteria

- Gray level value difference (with respect to S.P.) less than a threshold.
- Each candidate pixel should be 8-neighbor of region.

Region splitting and merging

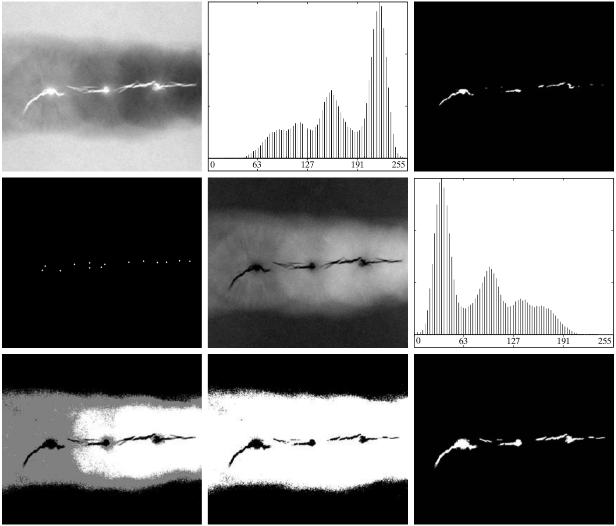
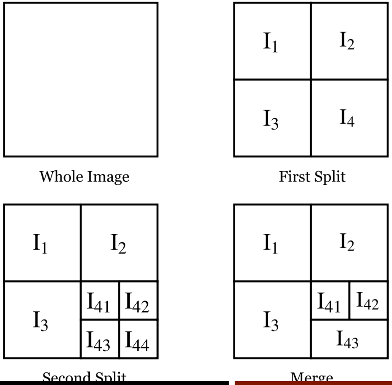


Figure: (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference image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

Region splitting and merging

- Define a criteria for each region to be a valid segment.
- Split each region which is not satisfy the criteria.
- Merge two neighbor region based on criteria.
- Split until a minimum size quadregions



Splitting and Merging

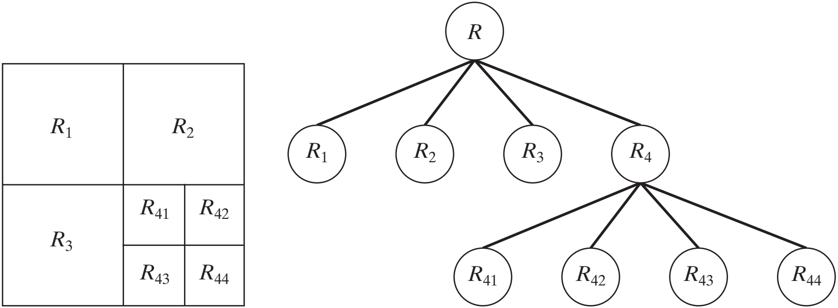
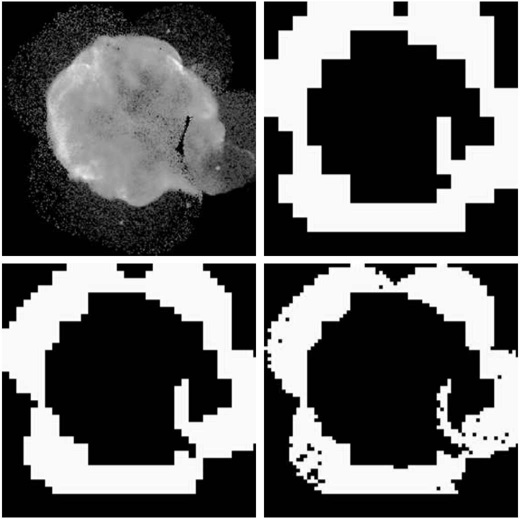


Figure: (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

Splitting and Merging Criteria

Splitting-Merging Criteria



a	b
c	d

FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

Segmentation using morphological watersheds

Thank You

Queries?