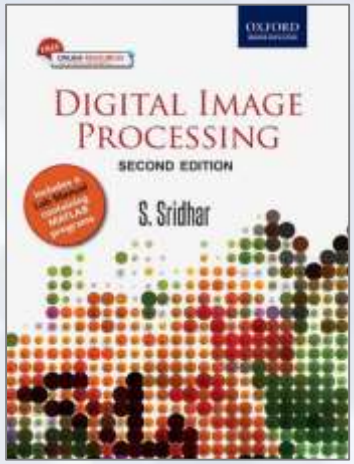
OXFORD
HIGHER EDUCATION

Digital Image Processing

2nd Edition

S. Sridhar



Chapter 9

Image Segmentation

Image Segmentation

- Segmentation is the process of partitioning a digital image into multiple regions and extracting meaningful regions known as regions of interest (ROI) for further image

Formal Definition of Image Segmentation

1. If the subregions are combined, the original region can be obtained. Mathematically, it can be stated that $\bigcup R_i = R$ for $i = 1, 2, \ldots, n$. For example, if the three regions of Fig. 7.1(c) R_1, R_2, and R_3 are combined, the whole region R is obtained.
2. The subregions R_i should be connected. In other words, the region cannot be open-ended during the tracing process.
3. The regions $R_1, R_2, \ldots, R_n$ do not share any common property. Mathematically, it can be stated as $R_i \bigcap R_j = \varphi$ for all i and j where $i \neq j$. Otherwise, there is no justification for the region to exist separately.
4. Each region satisfies a predicate or a set of predicates such as intensity or other image statistics, that is, the predicate (P) can be colour, grey scale value, texture, or any other image statistic. Mathematically, this is stated as $P(R_i) = \text{True}$.

CLASSIFICATION OF ALGORITHMS

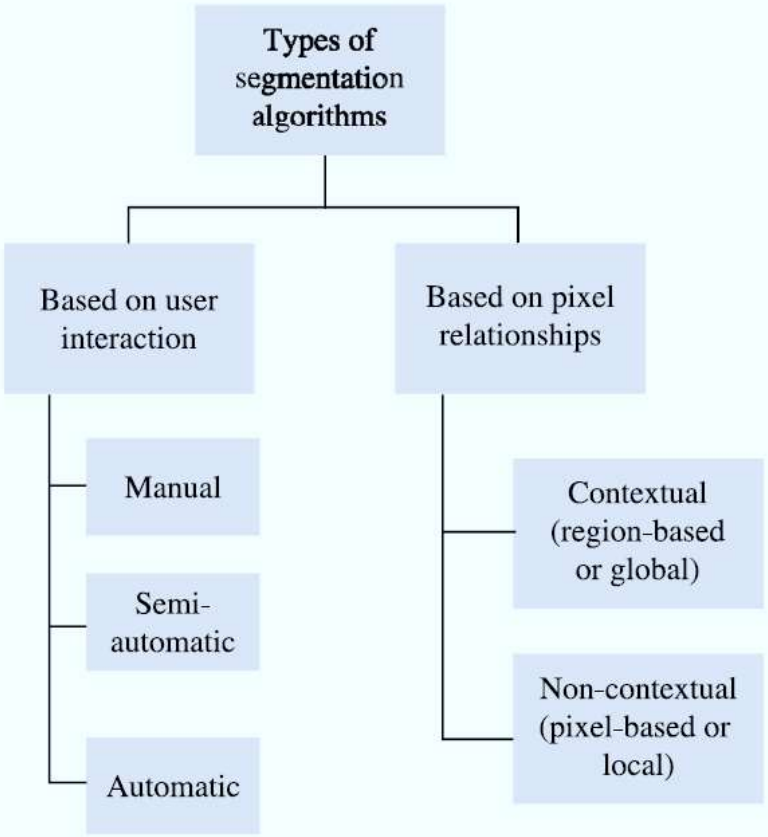


Fig. 9.2 Classification of segmentation algorithms

Types of Segmentation Algorithms

- Contextual (region-based or global) algorithms
- Non-contextual (pixel-based or local) algorithms

DETECTION OF DISCONTINUITIES- Point Detection

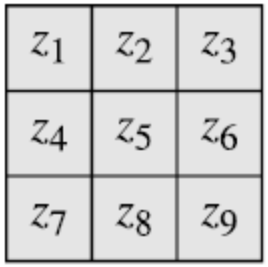


Fig. 9.3 Generic 3 × 3 spatial mask

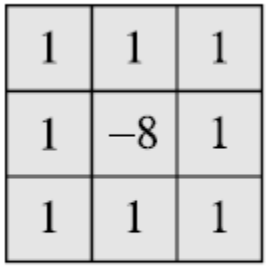


Fig. 9.4 Point detection mask

Point Detection

The mask is superimposed onto an image and the convolution process is applied. The response of the mask is given as

$$R = \sum_{i=1}^{9} z_k f_k$$

where the f_k values are the grey level values of the pixels associated with the image. A threshold value T is used to identify the points. A point is said to be detected at the location on which the mask is centred if $|R| \geq T$,

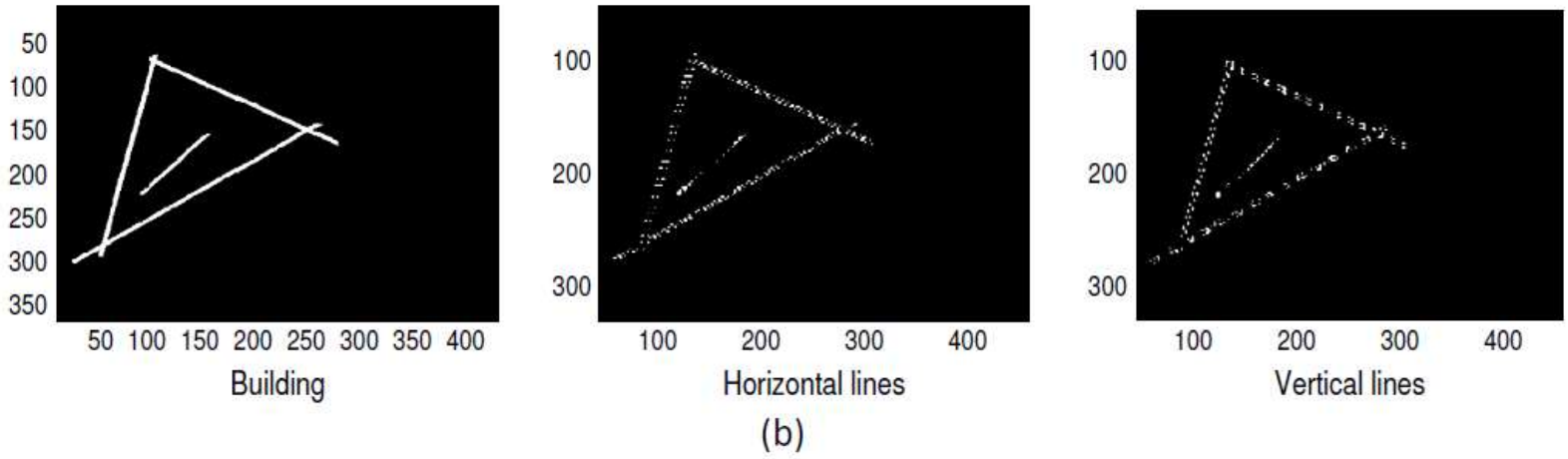
Line Detection

$$M_1 = \begin{pmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{pmatrix}, M_2 = \begin{pmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{pmatrix}, M_3 = \begin{pmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{pmatrix}, M_4 = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$$

(a)

Fig. 9.5 Line detection (a) Mask for line detection

Illustration



(b)

Fig. 9.5 (b) Original image and detected lines

EDGE DETECTION

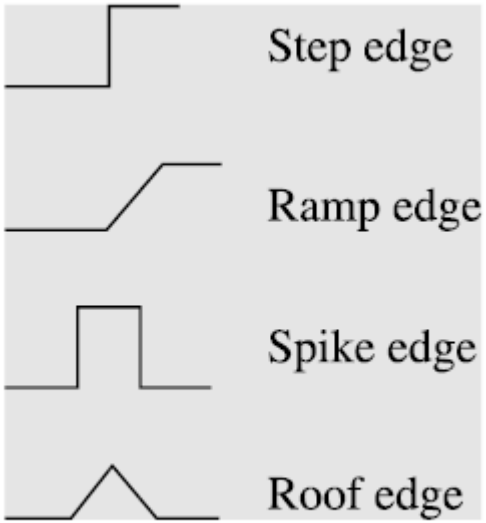


Fig. 9.7 Types of edges

Stages in Edge Detection

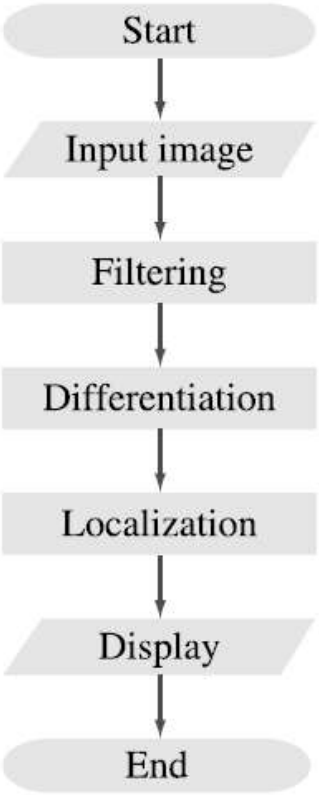


Fig. 9.8 Edge detection process

Edge Detection

$$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

where $$g_x = \left[\frac{\partial f(x,y)}{\partial x}\right] \text{ and } g_y = \left[\frac{\partial f(x,y)}{\partial y}\right]$$

The magnitude of this vector, generally referred to as the gradient ∇f, is

$$\nabla f(x,y) = \text{mag}(\nabla f(x,y)) = \left[(g_x)^2 + (g_y)^2\right]^{1/2}$$

Types of Edge Detectors

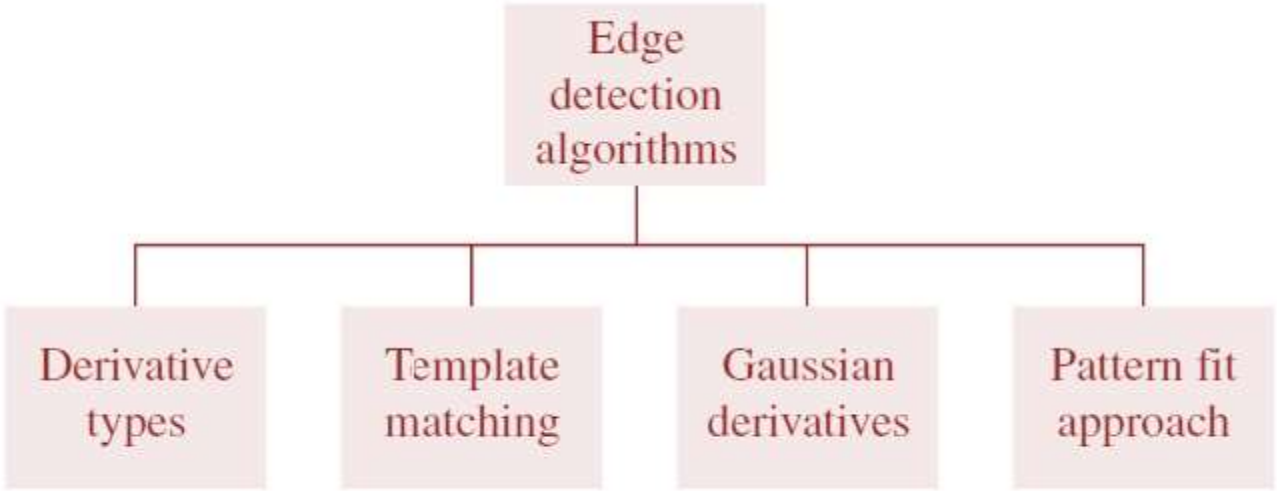


Fig. 9.9 Types of edge detectors

First-order Edge Detection Operators

$$\text{Backward difference} = \frac{f(x) - f(x - \Delta x)}{\Delta x}$$

$$\text{Forward difference} = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$$\text{Central difference} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

Roberts operator

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5)$$

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6)$$

Roberts masks of the for the given cross difference is

$$g_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } g_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Prewitt

$$\nabla f \cong \left|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)\right| + \left|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)\right|$$

This approximation is known as the Prewitt operator. Its masks are as follows:

$$M_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \text{ and } M_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Sobel

$$\nabla f \equiv \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

The masks are as follows:

$$M_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \text{ and } M_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

(a) (b) (c) (d)

Fig. 9.10 Edge detection using first-order operators (a) Original image (b) Roberts edge detection (b) Prewitts edge detection (c) Sobel edge detection

Kirsch

$$K_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} K_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} K_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} K_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$K_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} K_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} K_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} K_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

Robinson compass mask The spatial masks for the Robinson edge operator for all the directions are as follows:

$$R_0 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} R_1 = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix} R_2 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix},$$

$$R_3 = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{pmatrix} R_4 = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} R_5 = \begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{pmatrix},$$

$$R_6 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} R_7 = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

Frei-Chen

$$F_1=\frac{1}{2\sqrt{2}}\begin{pmatrix}1 & \sqrt{2} & 1\\ 0 & 0 & 0\\ -1 & -\sqrt{2} & -1\end{pmatrix}\qquad F_2=\frac{1}{2\sqrt{2}}\begin{pmatrix}1 & 0 & -1\\ \sqrt{2} & 0 & -\sqrt{2}\\ 1 & 0 & -1\end{pmatrix}$$

$$F_3=\frac{1}{2\sqrt{2}}\begin{pmatrix}0 & -1 & \sqrt{2}\\ 1 & 0 & -1\\ \sqrt{2} & 1 & 0\end{pmatrix}\qquad F_4=\frac{1}{2\sqrt{2}}\begin{pmatrix}\sqrt{2} & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & \sqrt{2}\end{pmatrix}$$

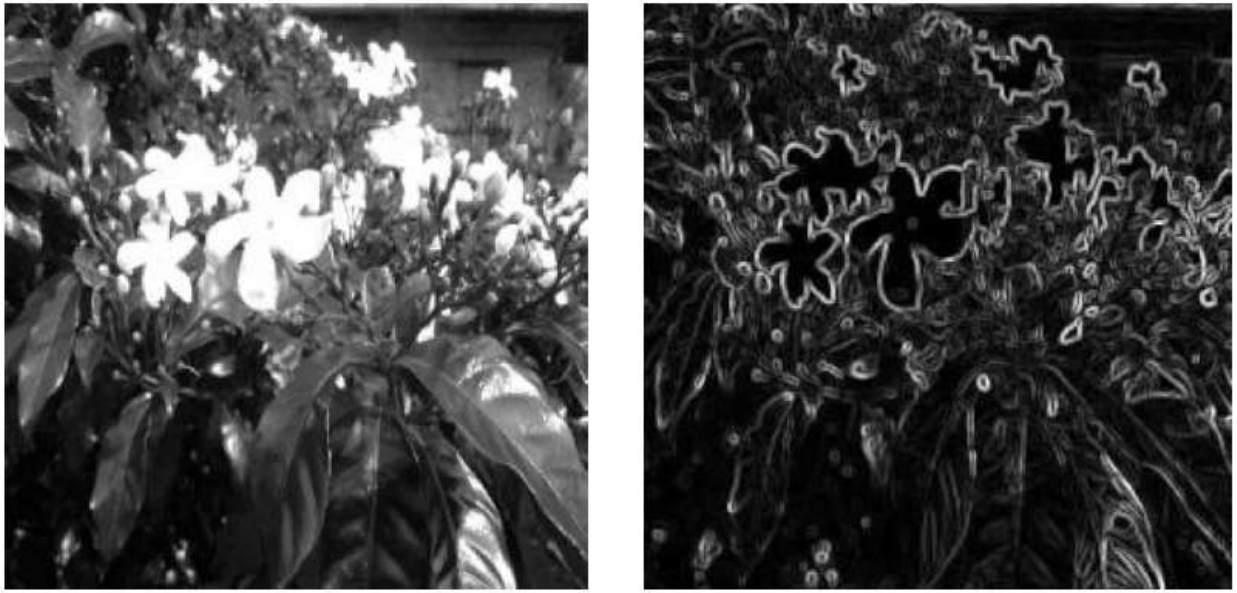
$$F_5=\frac{1}{2}\begin{pmatrix}0 & 1 & 0\\ -1 & 0 & -1\\ 0 & 1 & 0\end{pmatrix}\qquad F_6=\frac{1}{2}\begin{pmatrix}-1 & 0 & -1\\ 0 & 0 & 0\\ 1 & 0 & -1\end{pmatrix}$$

Frei-Chen

$$F_7 = \frac{1}{6}\begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

$$F_8 = \frac{1}{6}\begin{pmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{pmatrix}$$

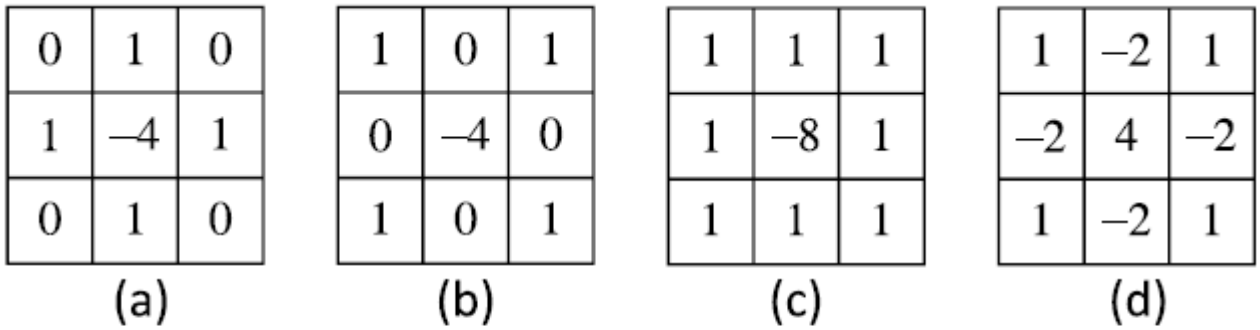
$$F_9 = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



(a) (b)

Fig. 9.11 Template matching masks (a) Original image (b) Image obtained using Krisch mask

Second-order Derivative Filters



0	1	0
1	–4	1
0	1	0

(a)

1	0	1
0	–4	0
1	0	1

(b)

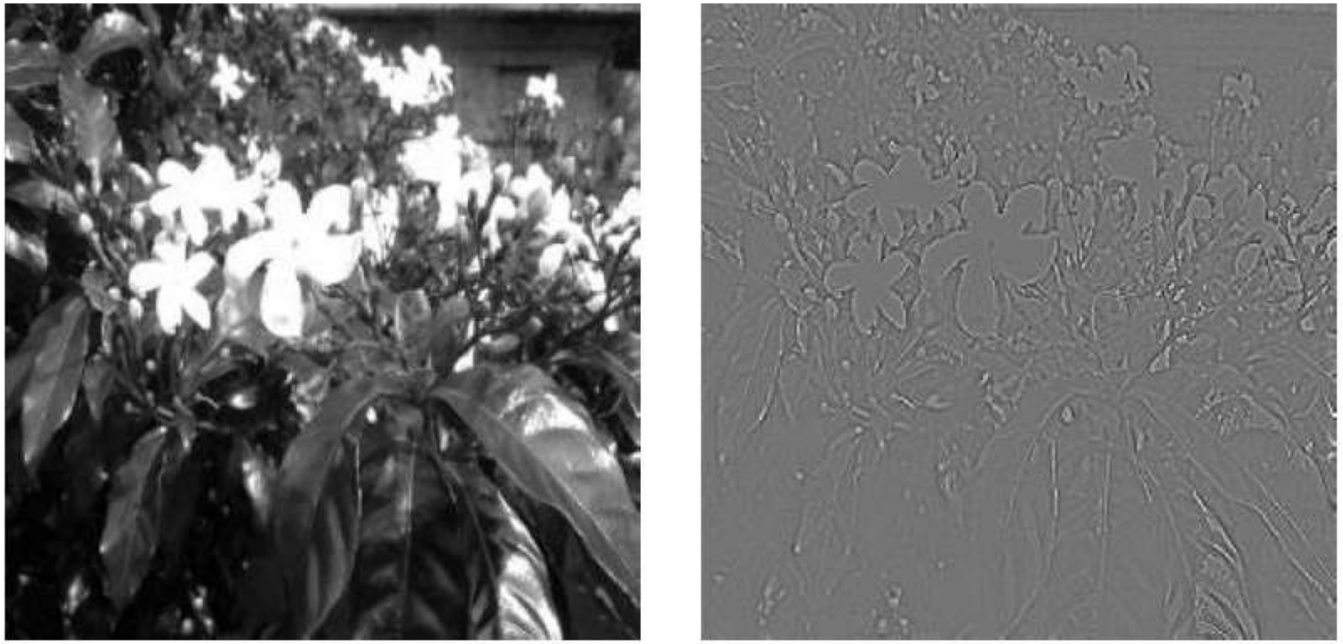
1	1	1
1	–8	1
1	1	1

(c)

1	–2	1
–2	4	–2
1	–2	1

(d)

Fig. 9.12 Different Laplacian masks (a) Laplacian filter (b) 45° rotated mask (c) Variant 1 (d) Variant 2



(a) (b)

Fig. 9.13 Laplacian method (a) Original image (b) Result of applying Laplacian mask

Laplacian of Gaussian (Marr–Hildrith) operator

The LoG kernel can now be described as

$$LoG \triangleq \frac{\partial^2}{\partial x^2} G_\sigma(x,y) + \frac{\partial^2}{\partial y^2} G_\sigma(x,y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

Difference of Gaussian filter

$$G_{\sigma_1}(x,y) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{x^2+y^2}{2\sigma_1^2}\right) \text{ with Gaussian of width } \boldsymbol{\sigma}_1$$

The width of the Gaussian is changed and a new kernel is obtained as

$$G_{\sigma_2}(x,y) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{x^2+y^2}{2\sigma_2^2}\right) \text{ with Gaussian of width } \boldsymbol{\sigma}_2$$

The DoG is expressed as the difference between these two Gaussian kernels:

$$\begin{aligned} G_{\sigma_1}(x,y) - G_{\sigma_2}(x,y) &= (G_{\sigma_1} - G_{\sigma_2}) * f(x,y) \\ &= \text{DoG} * f(x,y) \end{aligned}$$

The DoG as a kernel can now be stated as

$$\text{DoG} = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}}\left[\frac{1}{\sigma_1} e^{-\frac{(x^2+y^2)}{2\sigma_1^2}} - \frac{1}{\sigma_2} e^{-\frac{(x^2+y^2)}{2\sigma_2^2}}\right]$$

Canny edge detection

1. Good edge detection—The algorithm should detect only the real edge points and discard all false edge points.
2. Good edge localization—The algorithm should have the ability to produce edge points that are closer to the real edges.
3. Only one response to each edge—The algorithm should not produce any false, double, or spurious edges.

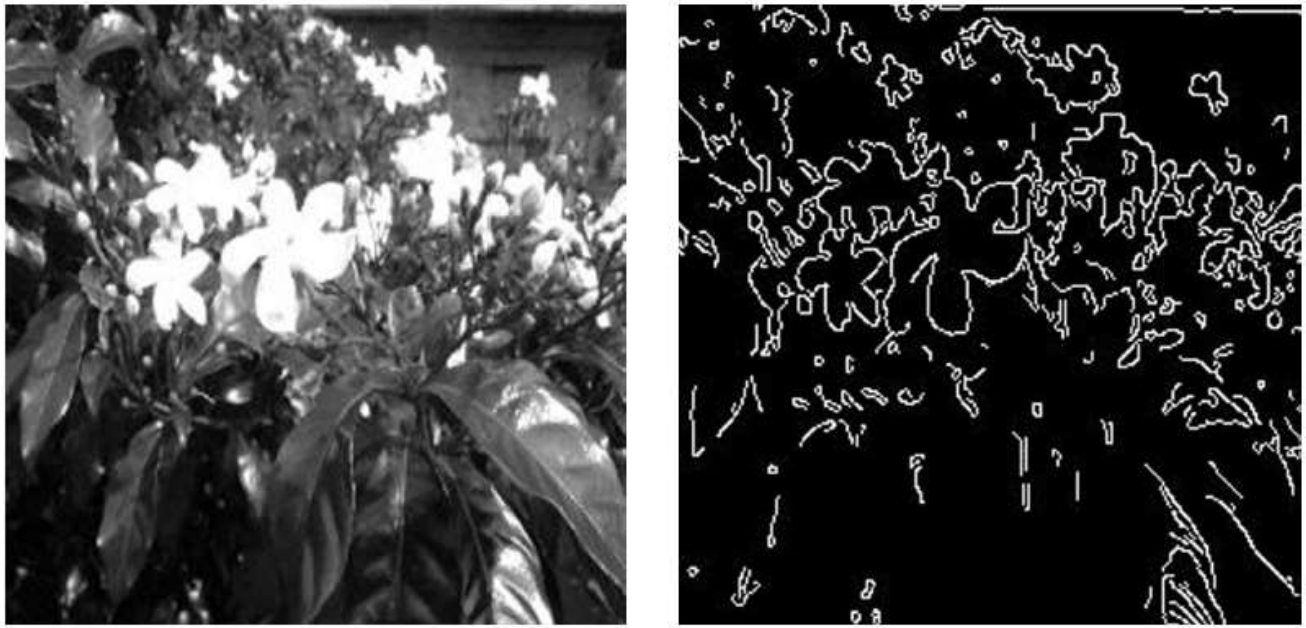
Algorithm

- First convolve the image with the Gaussian filter. Compute the gradient of the resultant smooth image. Store the edge magnitude and edge orientation separately in two arrays, $M(x, y)$ and $\alpha(,)xy$, respectively.

- The next step is to thin the edges. This is done using a process called *non-maxima suppression*.

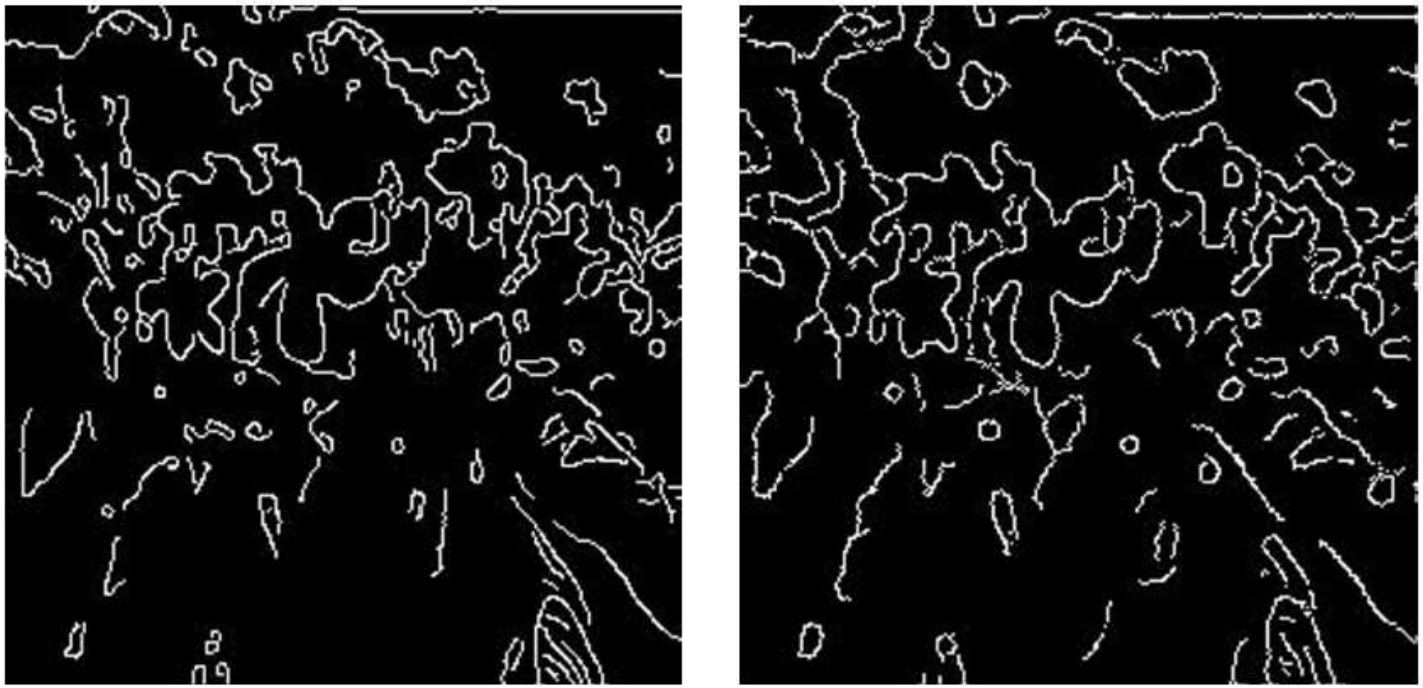
Algorithm

- Apply hysteresis thresholding. The idea behind hysteresis thresholding is that only a large amount of change in the gradient magnitude matters in edge detection and small changes do not affect the quality of edge detection.



(a) (b)

Fig. 9.15 Canny edge detection (a) Original image (b) Canny edge detection at $\sigma = 1$



(c) (d)

Fig. 9.15 (c) Canny edge detection at $\sigma = 1.5$ (d) Canny edge detection at $\sigma = 2$

Pattern fit algorithm

$$v = \alpha_1 x + \alpha_2 y + \alpha_3$$

Let the error be a squared difference between the original and the approximated and can be calculated using a least square fit. The error is given as

$$\varepsilon = [a_1 x + a_2 y + a_3 - g_0]^2 + [a_1 (x-1) + a_2 y + a_3 - g_1]^2$$
$$+ [a_1 (x-1) + a_2 (y-1) + a_3 - g_2]^2 + [a_1 x + a_2 (y-1) + a_3 - g_3]^2$$

$$\alpha_1 = \frac{g_0 + g_3}{2} - \frac{g_1 + g_2}{2}$$

$$\alpha_2 = \frac{g_0 + g_1}{2} - \frac{g_2 + g_3}{2}$$

This is equivalent to the Roberts filter mask. Here, the edge gradient is given as

$$g'(x, y) = \sqrt{\alpha_1^2 + \alpha_2^2}$$

This approximates the Roberts gradient in the root mean square (RMS) way.

If the plane is estimated with grey scale values $g_i (i = 1, 2, ..., 8)$, then

$$\alpha_1 = \frac{1}{2}\left(\frac{g_4 + g_5 + g_6}{3} - \frac{g_1 + g_2 + g_3}{3}\right)$$

$$\alpha_2 = \frac{1}{2}\left(\frac{g_6 + g_7 + g_8}{3} - \frac{g_1 + g_2 + g_3}{3}\right)$$

This is equivalent to the Prewitt mask. This approximates the Prewitt mask by a factor of $\frac{1}{\sqrt{2}}$ in the RMS sense.

If the grey scale values g_1, g_3, g_5, and g_7 are weighted by 2 and the remaining coefficients by 1, we get

$$\alpha_1 = \frac{1}{2}\left(\frac{g_4 + 2g_5 + g_6}{6} - \frac{g_2 + 2g_1 + g_8}{6}\right)$$

$$\alpha_2 = \frac{1}{2}\left(\frac{g_6 + g_7 + g_8}{6} - \frac{g_2 + 2g_3 + g_4}{6}\right)$$

This is proved to be similar to the Sobel operator. This approximates the Sobel mask by a factor of $\frac{2\sqrt{2}}{3}$ in the RMS sense.

Edge Operator Performance

1. Missing valid edge points
2. Classifying the noise points as valid edge points
3. Smearing edges

Pratt figure of merit

The Pratt figure of merit rating factor is one such objective evaluation procedure and is given as

$$R = \frac{1}{A}\sum_{i=1}^{D}\frac{1}{1+\alpha d^2}$$

D is the number of edge points detected by the edge operator. A is the maximum of ideal edge points present in the image and amongst the detected edge points (D). The distance

Edge Linking Algorithms

1. Similar gradient magnitude

$$\left\|\nabla f(x,y)\right\| - \left\|\nabla f(x',y')\right\| \leq T$$

2. Similar gradient orientation

$$\left|\phi(\nabla f(x,y) - \phi(\nabla f(x',y')\right| \leq A$$

where A is the angular threshold. Edge linking is a post-processing technique that is used to link edges.

Edge relaxation

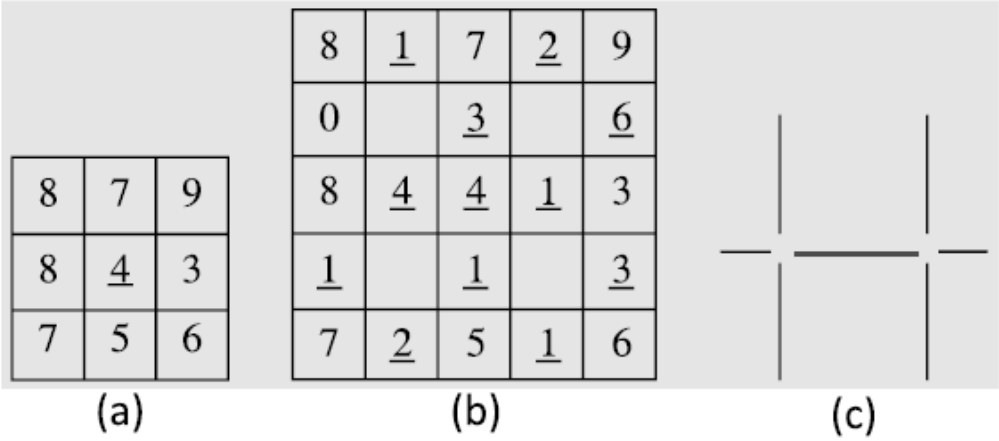


Fig. 9.16 Edge relaxation (a) Original image (b) Cracks with respect to the centre (c) The 3–3 crack edge

1. Assume an initial confidence level for the edge.
2. Classify the neighbouring edge crack types. It is called based on the value of m, a, b, and c. The edge type is type 0 if $(m - a)(m - b)(m - c)$, type 1 if $a(m - b)(m - c)$, type 2 if $ab(m - c)$, and type 4 if abc.
3. Update the confidence of the edge. Increment if the cracks are of the type (1, 1), (1, 2), (2, 1), (1, 3), or (3, 1), decrement if the cracks are of the type (0, 0), (0, 1), (0, 3), (2, 0), or (3, 0) and do nothing for other cracks.
4. Repeat steps 2–3 until the edge confidence becomes 0 or 1.

Graph theoretic algorithms

1. Forming the graph
2. Assigning cost functions
3. Identifying the start and end points
4. Finding the minimum cost path

Algorithms

1. Start from the start node. For this situation, the edge is assumed to flow from top to bottom. So edges *a* and *c* are the only possibilities. Place the nodes in the open list.
2. Let *n* be the first node of the open list.
3. If the open list is empty then exit, otherwise extract the node *n*.
4. If node *n* is the goal, then exit.
5. Otherwise,
   (a) Expand *n*
   (b) Evaluate *f*(*n*) for all the children
   (c) Sort them in the order of minimum cost
   (d) Place them in the open list and preserve the order
6. Repeat steps 2–5 till the open list becomes empty.

HOUGH TRANSFORMS AND SHAPE DETECTION

$$y = m'x + c'$$

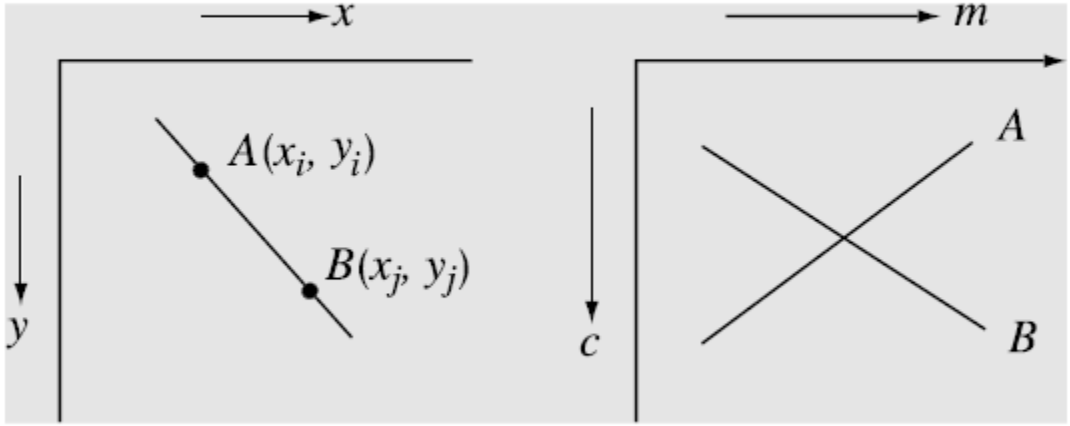


Fig. 9.18 Line in Hough transform

Modified Algorithm

1. Load the image.
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space P.
4. Repeat the following for all the pixels of the image:

   If the pixel is an edge pixel, then for all θ

   (a) Calculate ρ for the pixel (x, y) and θ.

   (b) Increment the position (ρ, θ) in the accumulator array P.
5. Show the Hough space.
6. Find the local maxima in the parameter space.
7. Draw the line using the local maxima.

Circle Detection

1. Load the image.
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space P.
4. Repeat the following for all the pixels of the image:

   If the pixel is an edge pixel, then for all values of r, calculate

   (a) $x_0 = x - r\cos\theta$

   (b) $y_0 = y - r\sin\theta$

   (c) $P(x_0, y_0, r) = P(x_0, y_0, r) + 1$
5. Show the Hough space.
6. Find the local maxima in the parameter space.
7. Draw the circle using the local maxima.

CORNER DETECTION

$$c(x,y) = [\Delta x \Delta y]\begin{pmatrix} A & B \\ B & C \end{pmatrix}\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

where $A(x,y) = \sum_{W} I_x^{\,2}(x_i, y_i)$, $B(x,y) = \sum_{W} I_x(x,y) I_y(x,y)$, and $C(x,y) = \sum_{W} I_y^2(x,y)$.

The matrix C captures the intensity structure of the neighbourhood. In addition, it is smoothed by the linear Gaussian filter. The Gaussian-smoothed matrix R is represented as

$$R = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$$

The Eigen values of this matrix R characterize edge strength and the Eigen vectors represent the edge orientation. Let λ_1 and λ_2 be the Eigen values, then

Circle Detection Algorithm

1. If the values of λ_1 and λ_2 are too small, it implies that the window region is flat.
2. If λ_1 is high and λ_2 is small or vice versa, it is an edge.
3. If λ_1 and λ_2 are both high, it indicates the presence of a corner.

Harris Corner

1. Select an image location (x, y).
2. Identify it as a corner point when $Q(u, v) >$ threshold value.
3. Insert into a list of corner points sorted by the corner strength.
4. Eliminate the false corners when a weaker corner point lies near a stronger corner.
5. Display all the corner points.
6. Exit.

PRINCIPLE OF THRESHOLDING

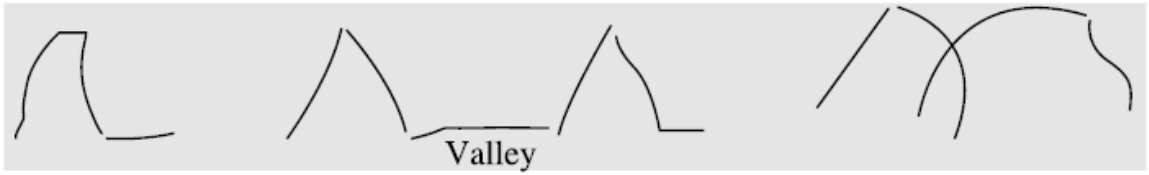


Fig. 9.22 Some examples of histograms (a) Unimodal histogram (b) Bimodal histogram (c) Histogram showing overlapping regions between foreground and background objects

Effect of noise over threshold process and peakiness test

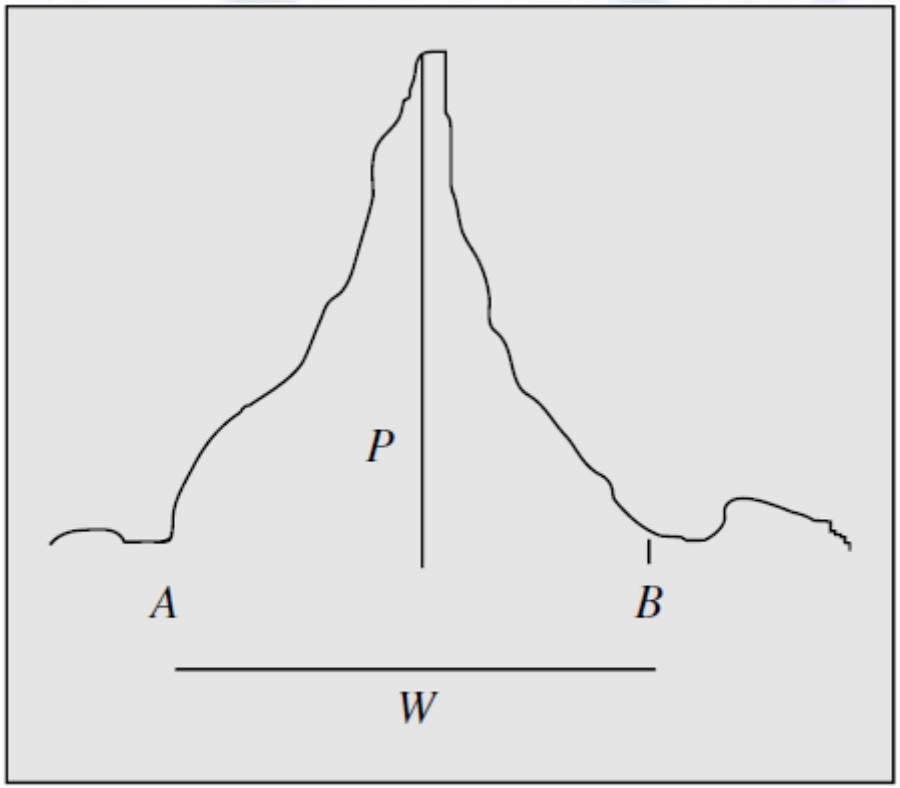


Fig. 9.23 A sample histogram peak

$$\text{Peakiness} = \left(1 - \frac{A+B}{2P}\right) \times (1 - \text{Peak sharpness})$$

Thresholding Algorithm

1. Compute the histogram.
2. Smooth the histogram by averaging. This step will remove the small peaks.
3. Identify the candidate peaks and valleys in the histogram. Detect the good peaks by applying the peakiness test. Remove all false peaks created by noise. Then segment the image using the thresholds at the valleys.

Global Thresholding Algorithms

1. Choose an initial threshold $T = T_0$, where T_0 is the mean of the two peaks or the mean pixel value.
2. Calculate the mean value of the pixel below the threshold (μ_1) and the mean value of the pixel above this threshold (μ_2).
3. Compute a new threshold as

$$T_i = \mu_1 + \mu_2/2$$

4. Repeat steps 2 and 3 until there is no change in T.

A variation of Global thresholding Algorithm

1. Choose an initial threshold value T randomly, say, 128.
2. Find the mean (m_1) of the pixels that lie below T in the histogram.
3. Find the mean (m_2) of the pixels that lie above T in the histogram.
4. The new threshold $T_{new} = (m1 + m2)/2$.
5. Repeat steps 2–4 till T_{new} no longer changes.

Global Thresholding

(a)

(b)

Fig. 9.24 Global thresholding algorithm (a) Original image (b) Threshold image

Multiple Thresholding

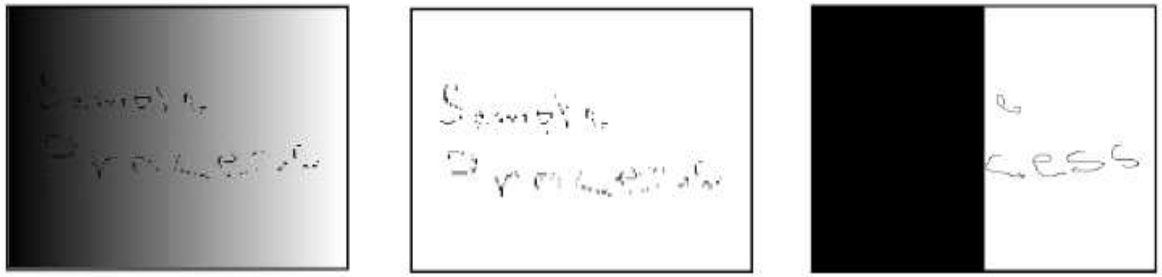
$$\text{Output image} = \begin{cases} g_1 \text{ if } 0 \le f_i \le t_1 \\ g_2 \text{ if } t_1 < f_i \le t_2 \\ \quad \cdots \\ g_n \text{ if } t_{n-1} < f_i \le 255 \end{cases}$$

Adaptive Thresholding Algorithm

1. $\text{Mean} + c$

2. $\text{Median} + c$

3. $\frac{\text{Max} + \text{Min}}{2}$



(a) (b) (c)

Fig. 9.25 Comparison of thresholding algorithms (a) Original image (b) Result of adaptive algorithm (c) Result of global thresholding algorithm

Optimal Thresholding Algorithms

$$\alpha T^2 + \beta T + \gamma = 0$$

where $\alpha = \sigma_1^2 - \sigma_2^2$

$$\beta = 2 \times (\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)$$

$$\gamma = \sigma_1^2 \mu_2^2 - \sigma_2^2 \mu_1^2 + 2\sigma_1^2 \sigma_2^2 \ln \frac{\sigma_2 P_1}{\sigma_1 P_2}$$

There are two cases, detailed as follows:

Case 1: If the variances are equal, that is, $\sigma^2 = \sigma_1^2 = \sigma_2^2$, then a single threshold is sufficient and can be given as

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_1}{P_2}\right)$$

Case 2: If the prior probabilities $P_1 = P_2$, then the optimal threshold can be described as

$$T = \frac{\mu_1 + \mu_2}{2}$$

Non-Parametric – Otsu Algorithm

The algorithm can be stated as follows:

1. Load the image.
2. Form the histogram.
3. Calculate the probability of every level.
4. Initialize $w_i(0)$ and $\mu_i(0)$.
5. For every threshold value ranging from one to the maximum intensity, update the w_i and μ_i values and compute $\sigma_b^2(t)$.
6. Find the maximum $\sigma_b^2(t)$. The corresponding threshold value is the optimal threshold value.

PRINCIPLE OF REGION-GROWING

Selection of the initial seed This is a very important issue. The initial seed that represents the ROI should be given typically by the user. The initial seed can also be chosen automatically. The seeds can be either single or multiple.

Seed growing criteria The initial seed grows by the similarity criterion if the seed is similar to its neighbourhood pixels. Similarity criterion denotes the minimum difference in the grey levels or the average of the set of pixels and can apply to grey level, texture, or colour. Thus, the initial seed 'grows' by adding the neighbours if they share the same properties as the initial seed.

Termination of the segmentation process The rule for stopping the growing process is also very important as the region-growing algorithms are computationally very intensive operations and may not terminate easily.

Split-and-merge Algorithm

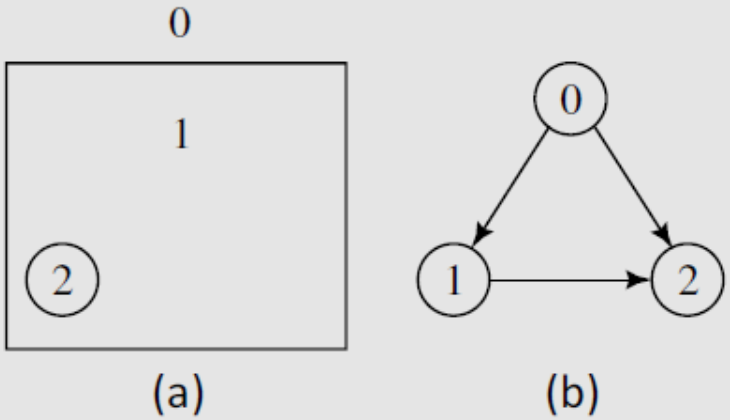


Fig. 9.29 Region adjacency graph (a) Sample image (b) RAG for the sample image

Algorithm

1. Segment the image into regions $R_1, R_2, \ldots, R_n$ using a set of thresholds.
2. Create a RAG. Use a similarity measure and formulate a homogeneity test.
3. For the regions R_i and R_j of the RAG, apply the homogeneity test. The homogeneity test is designed based on the similarity criteria such as intensity or any image statistics. If the similarity measure (S_{ij}) is greater than the threshold T, merge R_i and R_j. The threshold value can be determined using any known technique.
4. Repeat step 3 until no further region exists that requires merging.

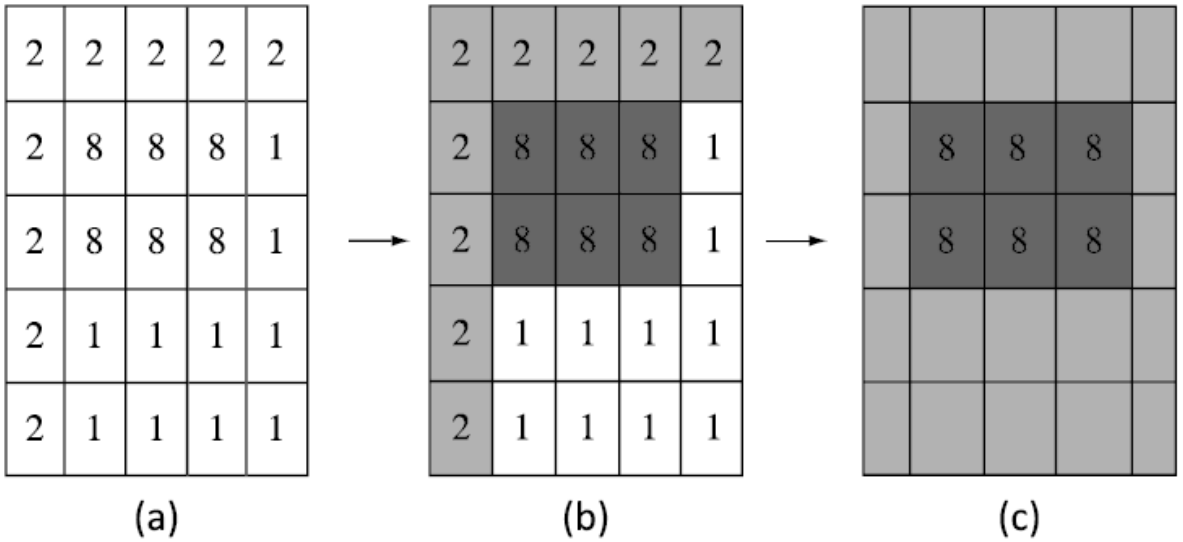


Fig. 9.31 Split-and-merge technique (a) Original image (b) Image showing separate regions (c) Final result

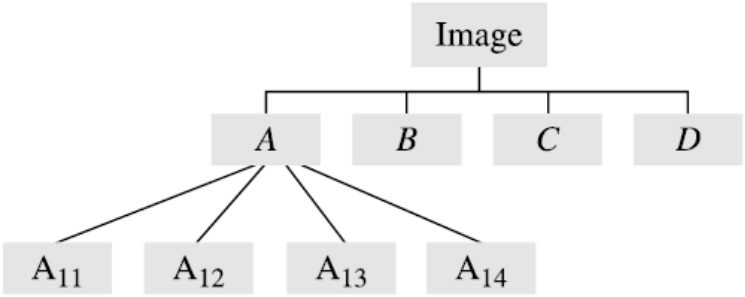
Split-and-merge Algorithm using Pyramid Quadtree

The algorithm is stated in two phases as follows:

Phase 1: Split and continue the subdivision process until some stopping criteria is fulfilled. The stopping criteria often occur at a stage where no further splitting is possible.

Phase 2: Merge adjacent regions if the regions share any common criteria. Stop the process when no further merging is possible.

(a)



(b)

Fig. 9.33 Split-and-merge algorithm using pyramid quadtree (a) Original image with the shaded region (b) Quadtree representation

DYNAMIC SEGMENTATION APPROACHES

$$d_{i,j} = \begin{cases} 1 \text{ if } \left| f(x,y,t_i) - f(x,y,t_j) \right| > T \\ 0 \text{ otherwise} \end{cases}$$

The logic can be extended for multiple images. There will be multiple frames $f(x, y, t_1), f(x, y, t_2), \ldots, f(x, y, t_n)$. The first image $f(x, y, t_1)$ is called the reference image.

Active contour models

1. Internal energy
2. External energy

Internal energy depends on intrinsic properties such as the boundary length or curvature. The minimization of these forces leads to shrinkage of the contour. To counteract the shrinkage of snakes, external energy is used. The *external energy* is derived from the image structures and it determines the relationship between the snake and the image. This

$$E_{\text{total}} = \alpha E_{\text{length}} + \beta E_{\text{curvature}} - \gamma E_{\text{image}}$$

E_{length} is the total length of the snake, $E_{\text{curvature}}$ is its curvature, and E_{image} is the counteracting or opposing force. The other parameters α, β, and γ are used to control the energy functions of the snake. The values of these parameters are estimated empirically and can be adjusted.

Fig. 9.34 Active contour model (a) Original image—contour is shown in yellow (b) Final contour shown in red (c) Segmented image (Refer to the OUPI website for colour images)

VALIDATION OF SEGMENTATION ALGORITHMS

Table 9.1 Contingency table

Human expert	Detected by the algorithm (computer system)	Missed by the algorithm (computer system)
ROI present	True positive (TP)	False negative (FN)
ROI absent	False positive (FP)	True negative (TN)

Metrics

$$\text{Sensitivity} = \frac{TP}{TP + TN} \times 100\%$$

Specificity is a measure of the ability of a test to give a negative result in extracting the ROI.

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100\%$$

The efficiency of the segmentation is given as

$$\text{Efficiency} = \frac{TN + TP}{TN + TP + FN + FP}$$

The efficiency or success rate is the ability of the algorithm to extract the ROI.

SUMMARY

- Segmentation is the process of partitioning the digital image into multiple regions. The main aim of segmentation is to extract the ROI for image analysis.
- Segmentation algorithms can be classified into three categories—manual segmentation, semi-automatic segmentation, and automatic segmentation algorithms. They can also be classified based on the technique, as contextual and non-contextual.
- An edge is a set of connected pixels that lie on the boundary between two regions. The pixels on an edge are called edge points.
- There are three stages of edge detection—filtering, differentiation, and localization.
- The objective of the sharpening filter is to highlight the intensity transitions.
- Edge detectors often do not produce continuous edges. Hence, edge linking algorithms are required to make them continuous.
- The Hough transform is used to fit points as plane curves.
- Thresholding produces uniform regions based on the threshold criterion T.
- Region oriented algorithms use the principle of similarity to produce coherent regions.
- A snake is a contour which, like an elastic band, grows or shrinks to fit the object of interest. Snakes can be controlled by the energy function.