Chapter 7 Image Segmentation

Introduction

Image Segmentation refers to the decomposition or partition of an image into meaningful regions. Segmentation subdivides an image into its constituent regions or objects. Most of the image analysis algorithms perform segmentation as a first step towards producing the description. It is typically used to locate objects and boundaries (lines and curves) in an image.

Some of practical applications of image segmentations are as follows

- Identifying objects in a moving scene for object-based video compression (MPEG4),
- Identifying objects which are at different distances from a sensor using dept measurements from a laser range finder enabling path planning for mobile robots (Stereo Vision).
- Face Recognition, Medical Imaging, Machine Vision, Iris Recognition, etc.

Image segmentation algorithms are based on two basic properties of intensity values:

(a) Discontinuity and (b) Similarity (Continuity)

(a) Discontinuity: To partition an image based on abrupt change in intensity such as edge in image. (Edge detection, Point or spots detection, Line detection)

(b) Similarity: To partition an image into regions that is similar according to a set of predefined criteria. (Thresholding, Region growing, Region splitting etc.)

7.1 Detection of Discontinuities

The most common way to look for discontinuities is to run a mask through the image. For 3 x 3 mask the response (R) of the mask at any given point in the image is given by:

$R = w_1z_1 + w_2z_2 + \ldots\ldots\ldots\ldots + w_9z_9$

$R = \sum_{i=1}^{9} w_i z_i$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

- Where $Z_i$is the gray level of the pixel associated with the mask coefficient $w_{i..}$The response of the mask is defined with respect to its center location.

7.1.1 Point Detection

It is principle of detection of isolated points in an image. We can say that a point has been detected at the location on which the mask is centered if $|R| \geq T$ (Where T is a non-negative threshold and R is response of the mask).The point can be detected using the mask shown below.

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

Figure 7.1: Point Detection Mask

- This formulation measures the weighted difference between the center point and its neighbors. The idea is that an *isolated point* (a point whose gray level is significantly different from its background and which is located in homogeneous area) will be quite different from its surroundings and thus be easily detectably by this type of mask.

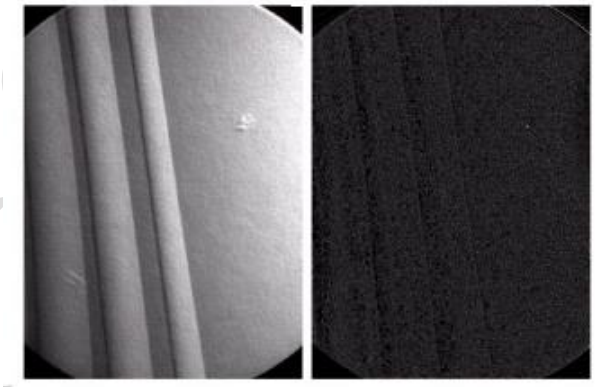


Figure 7.2 (a) Original Image (b) Point Detection Result

7.1.2 Line Detection

Line Detection is possible via various masks. The horizontal mask, +45° mask , Vertical mask and -45° mask is given below.

-1	-1	-1
2	2	2
-1	-1	-1

Figure 7.3: Horizontal Line Detection Mask

-1	-1	2
-1	2	-1
2	-1	-1

Figure 7.4 : + 45° Line Detection Mask

-1	2	-1
-1	2	-1
-1	2	-1

Figure 7.5: Vertical Line Detection Mask

2	-1	-1
-1	2	-1
-1	-1	2

Figure 7.6: -45° Line Detection Mask

- Suppose that the four masks (R_1 to R_4) are run individually through an image. If at a certain point of image, $|R_i| > |R_j|$ for all $i \neq j$, that point is said to be more likely associated with a line in the direction of mask i.For example, if at a point in the image, |R1| > |Rj| for j=2,3,4, that particular point is said to be more likely associated with a horizontal line.

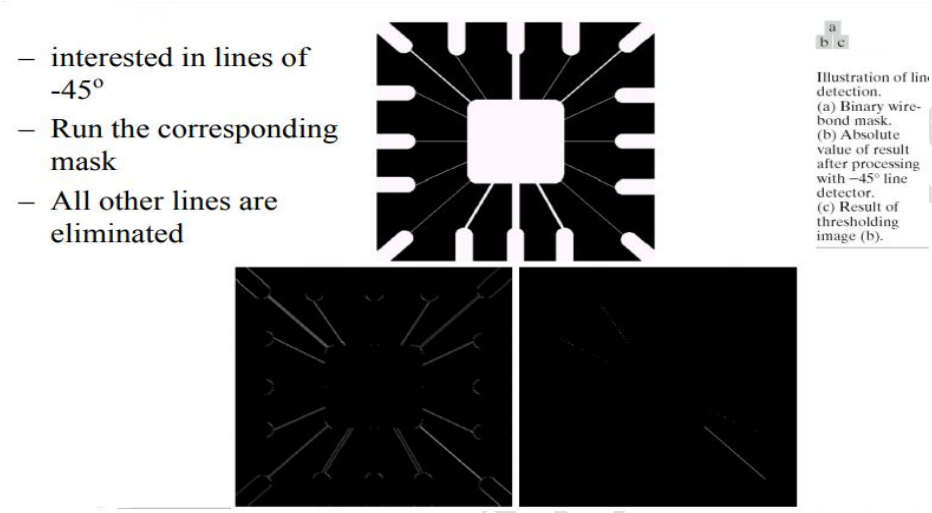


Figure 7.7 Illustration of Lines Detection

7.1.3 Edge Detection

Edge detection is a technique of image processing used to identify points in a digital image with discontinuities, simply to say, sharp changes in the image brightness. These points where the image brightness varies sharply are called the edges (or boundaries) of the image.

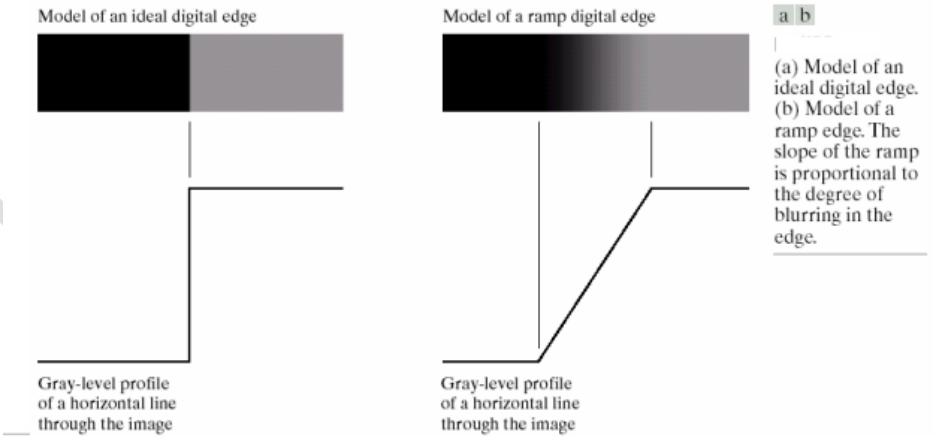


Figure 7.8 Different models of edges.

- It is most common approach for detecting meaningful discontinuities in gray level. Model of ideal digital edge, mode of a ramp digital edge. The "thickness" of the edge is determined by the length of the ramp and its transitions from an initial to a final gray level. This length is determined by the slope, which, in turn, is determined by the degree of blurring. Blurred edges tend to be thick and sharp edges tend to be thin.

First and Second Derivative of Gray Level Profile

- The first derivative is positive at the points of transition into and out of the ramp as we move from left to right along the profile. It is constant for points in the ramp and is zero in areas of constant gray level.
- The second derivative is positive at the transition associated with dark side and negative at the transition associated with the light side of the edge. The value is zero along the ramp and in the areas of constant gray level. Observations are given below :
- The magnitude of first derivative can be used to detect the presence of an edge at a point in an image. The sign of second derivative can be used to determine whether an edge pixel lies on dark or light side of an edge.

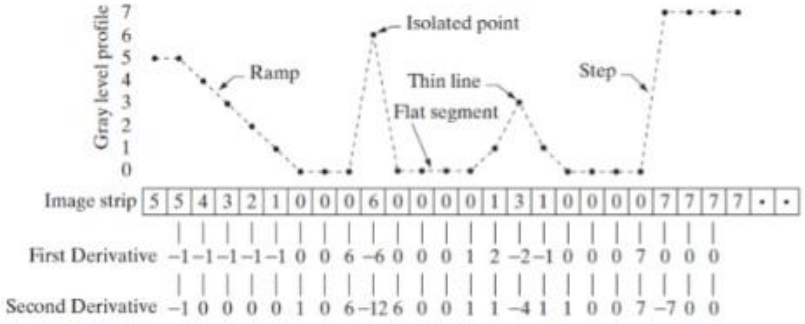


Figure 7.9 First and Second Derivatives of Gray Level Profile .

(a)First Order Derivative

First order derivative can be achieved by using Gradient operator (Roberts, Prewitt and Sobel operators).A 3 X 3 region of an image is shown below.

Gradient Operator

The gradient are finding edge strength and direction at location (x, y) of an image f, is the gradient denoted by Delta .This vector has the important geometrical property that it points in the direction of the greatest rate of change of location (x, y).

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

The magnitude (length) of a vector is denoted by,

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}$$

Where is the value of the rate of a change in the direction of the gradient vector gx, gy? The direction of the gradient vector is given by the angle measured with respect to the x-axis.

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

The direction of an edge at an arbitrary point (x, y) is orthogonal to the direction, α(x ,y), of the gradient vector at the point.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Figure 7.10 3 x 3 Region of an Image

- Computation of the gradient of an image is based on obtaining the partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ at every pixel location.

  $\partial f/\partial x = f(x+1,y) - f(x,y)$

  $\partial f/\partial y = f(x,y+1) - f(x,y)$

Robert Operator: The Roberts operator performs an easy, fast to calculate, 2-D special gradient activity on a picture. It therefore highlights regions of high special gradient which frequently correspond to edges. In its commonest usage, the input to the operator could be a grayscale image, as is that the output. Component values at every purpose within the output represent the calculable magnitude of the special gradient of the input image at that time.

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Figure 7.11 Roberts operator: $G_x=(Z_9-Z_5)$ and $G_y = (Z_8-Z_6)$

Prewitt Operator:

Prewitt operator is used for edge detection in an image. It detects two types of edges

- Horizontal edges
- Vertical Edges

Edges are calculated by using difference between corresponding pixel intensities of an image. All the masks that are used for edge detection are also known as derivative masks.All the derivative masks should have the following properties:

- Opposite sign should be present in the mask.
- Sum of mask should be equal to zero.
- More weight means more edge detection.

Prewitt operator provides us two masks one for detecting edges in horizontal direction and another for detecting edges in a vertical direction.

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Figure 7.12 Horizontal and Vertical Operator

Gx = (z3+z6+z9) – (z1+z4+z7) Gy = (z7+z8+z9) – (z1+z2+z3)

Sobel Opeartor:

Sobel mask is same as that of the Prewitt mask .There is just one distinction, Sobel operator has '2' and '-2' values in center of 1st, third column of horizontal mask and 1st, third rows of vertical mask. This offers additional weight age to the element values round the edge region.

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

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

Sobel operator convolution masks

Gx = (z3+2z6+z9) – (z1+2z4+z7)
Gy = (z7+2z8+z9) – (z1+2z2+z3)

Figure 7.13 Sobel Convolution Masks

Second order derivative (Laplacian Operator)

It is also possible to use *second order derivatives* to detect edges.

A very popular second order operator is the *Laplacian* operator.

The Laplacian of two dimensional image f(x,y) is given by

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Once more we can use discrete difference approximations to estimate the derivatives and represent the Laplacian operator with 3×3 the convolution mask

0	1	0
1	-4	1
0	1	0

Figure 7.14 Laplacian Operator Convolution Masks

- This equation needs to be expressed in discrete form.

  $\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$ (Partial Second order derivative in x direction)

  $\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$ (Partial Second order derivative in y direction)

  $\nabla^2 f = [\, f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \,]$
- Construct filter mask using $\nabla^2 f$.

  Second-order derivative of a 2-D function

  – Digital approximations by proper masks

  – Complementary use for edge detection

  • Cons: Laplacian is very sensible to noise; double edges

  • Pros: Dark or light side of the edge; zero crossings are of better use

  • Laplacian of Gaussian (LoG): preliminary smoothing to find edges through

zero crossings

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

7.2 Edge Linking and Boundary Detection Local processing

Ideally, edge detection should yield sets of pixels lying only on edges. In practice, these pixels seldom characterize edges completely because of noise or breaks in the edges. Therefore, edge detection typically is followed by linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries. There are usually two linking techniques

(a) Local Processing

(b) Global Processing

(a) Local Processing

All points that are similar according to predefined criteria are linked, forming an edge of pixels that share common properties. Similarity according to: 1. Strength (magnitude) 2. Direction of the gradient vector .

Strength of the gradient vector response $\nabla f \approx |G_x| + |G_y|$; $|\nabla f(x,y) - \nabla f(x_0, y_0)| \le E$

Gradient vector direction $\alpha(x,y) = \tan^{-1}(G_x / G_y)$ $\quad |\alpha(x,y) - \alpha(x_0, y_0)| < A$

Both magnitude and angle criteria should be satisfied

1. Compute the gradient magnitude and angle arrays, M(x,y) and α(x,y), of the input image, f(x,y) .
2. Form a binary image, g, whose value at any pair of coordinates (x,y) is given by:

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

where T_M is a threshold, A is a specified angle direction, and $\pm T_A$ defines a "band" of acceptable directions about A

3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length, K. Note that, by definition, a gap is bounded at both ends by one or more 1s. The rows are processed individually with no memory between them.

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 -θ

Example: find rectangular shapes similar to license plate

– Find gradients

– Connect edge points

– Check horizontal-verticalproportion

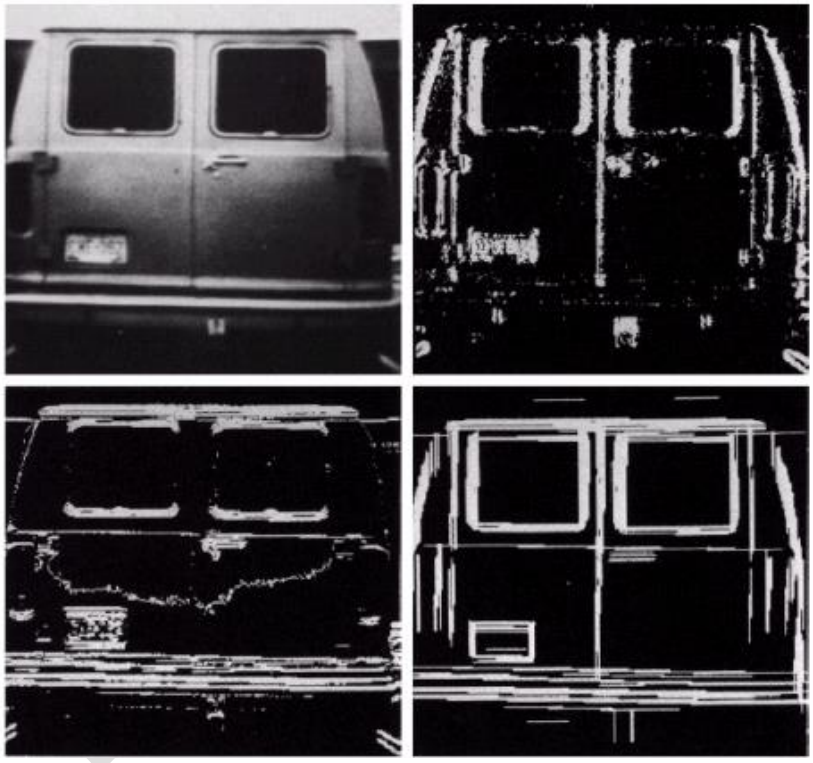


Figure 7.15 Local Processing of License Plate

Global Processing

- Good for unstructured environments.
- All pixels are candidate for linking.

- Need for predefined global properties.
- What we are looking for?
- Decide whether sets of pixels lie on curves of a specified shape.

Global processing via the Hough Transform

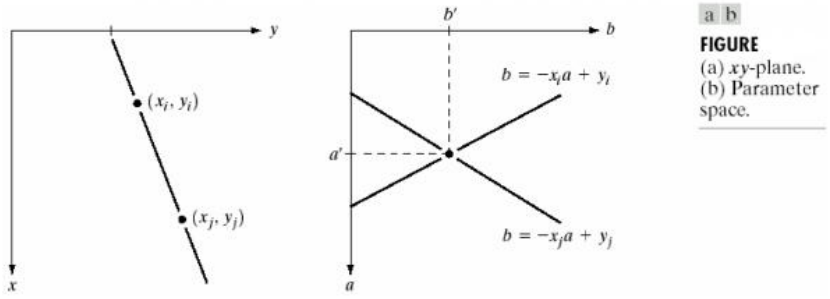
– Determine if points lie on a curve of specified shape

– Consider a point (x_i,y_i) and the general line equation $y_i = ax_i + b$

– Write the equation with respect to ab-plane (parametric space) $b = -x\, a_i + y_i$

– Write the equation for a second point (x_j,y_j) and find the intersection point (a',b') on the parametric space

– All points on a line intersect at the same parametric point



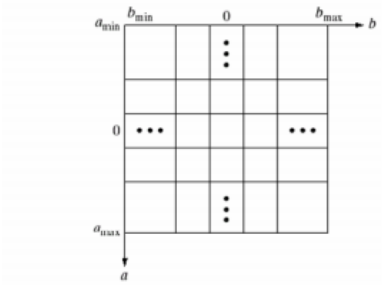
a b
FIGURE
(a) *xy*-plane.
(b) Parameter space.

Figure 7.16 Global processing via the Hough Transform

Computational aspects of the Hough transform

– Subdivision of the parametric space into accumulator cells

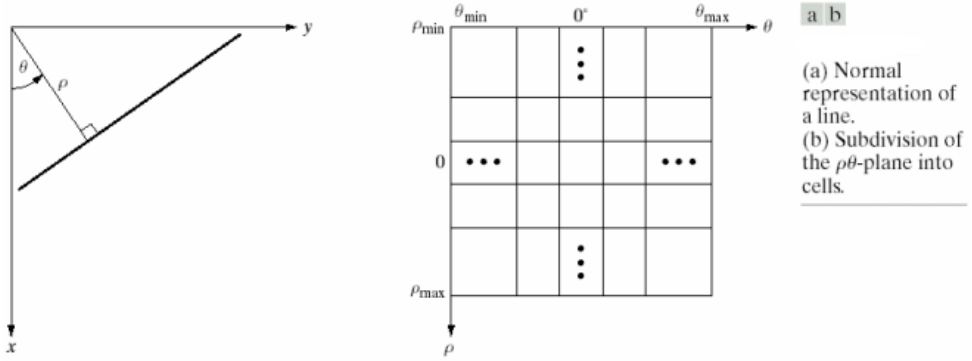
- The cell at (i,j) with accumulator values $A(i,j)$ corresponds to (a_i,b_j)
- For every point (x_k,y_k) vary a from cell to cell and solve for b: $b = -x_k a + y_k$
- If a_p generates b_q, then increment the accumulator $A(p,q)=A(p,q)+1$
- At the end of the procedure, a value of Q in $A(i,j)$ corresponds to Q points in the xy-plane lying on the line $y = a_i x + b_j$
- K different incremens of a generate K different values of b; for n different image point, the method involves nK computations (linear complexity)



Hough transform: handling the vertical lines

– Through normal representationx cosθ + y sinθ = ρ

Instead of straight lines, there are sinusoidal curves in the parameter space. The number or intersecting sinusoids is accumulated and then the value Q in the accumulator A(i,j) shows the number of collinear points lying on a linex $\cos\theta_j$ + y $\sin\theta_j$ = ρ

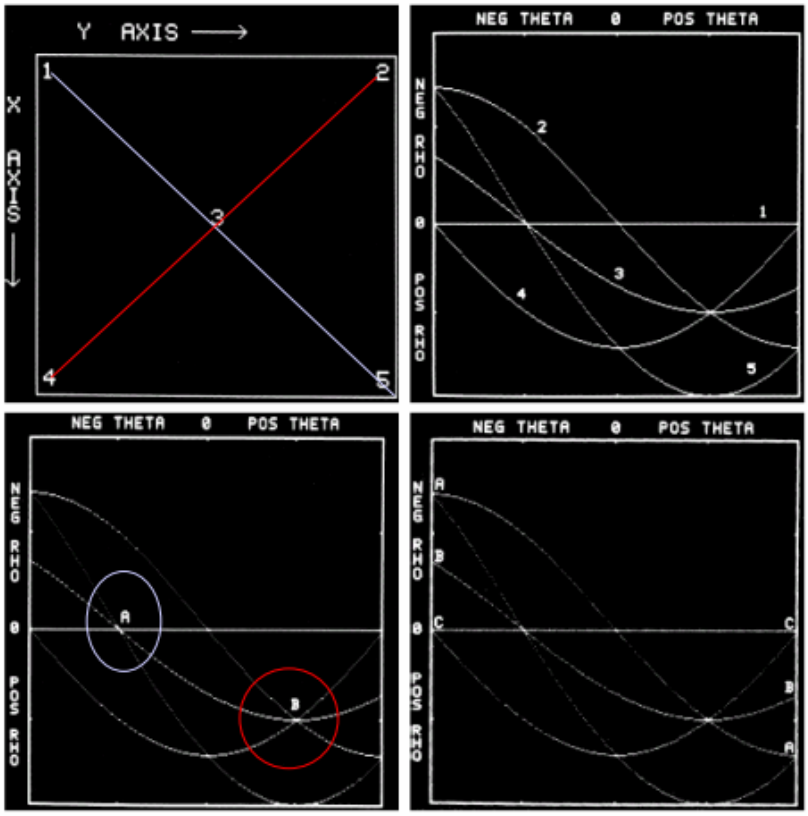


a b

(a) Normal representation of a line.
(b) Subdivision of the $\rho\theta$-plane into cells.

Example: two lines connecting three points each indicates that the Hough transform exhibits a reflective adjacency relationship

Summary of Hough transform for edge linking – Compute the gradient – Specify subdivisions in the parametric plane – Examine the counts of the accumulator cells – Examine the continuity relationship between pixels in a chosen cell

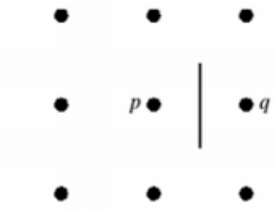


Global processing via Graph-Theoretic Techniques

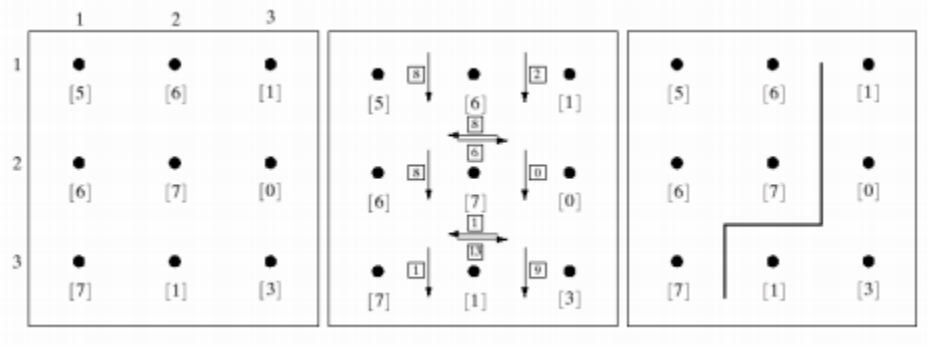
– Graph G=(N,U): a set of nodes N and a set U of arcs (n_i,n_j).

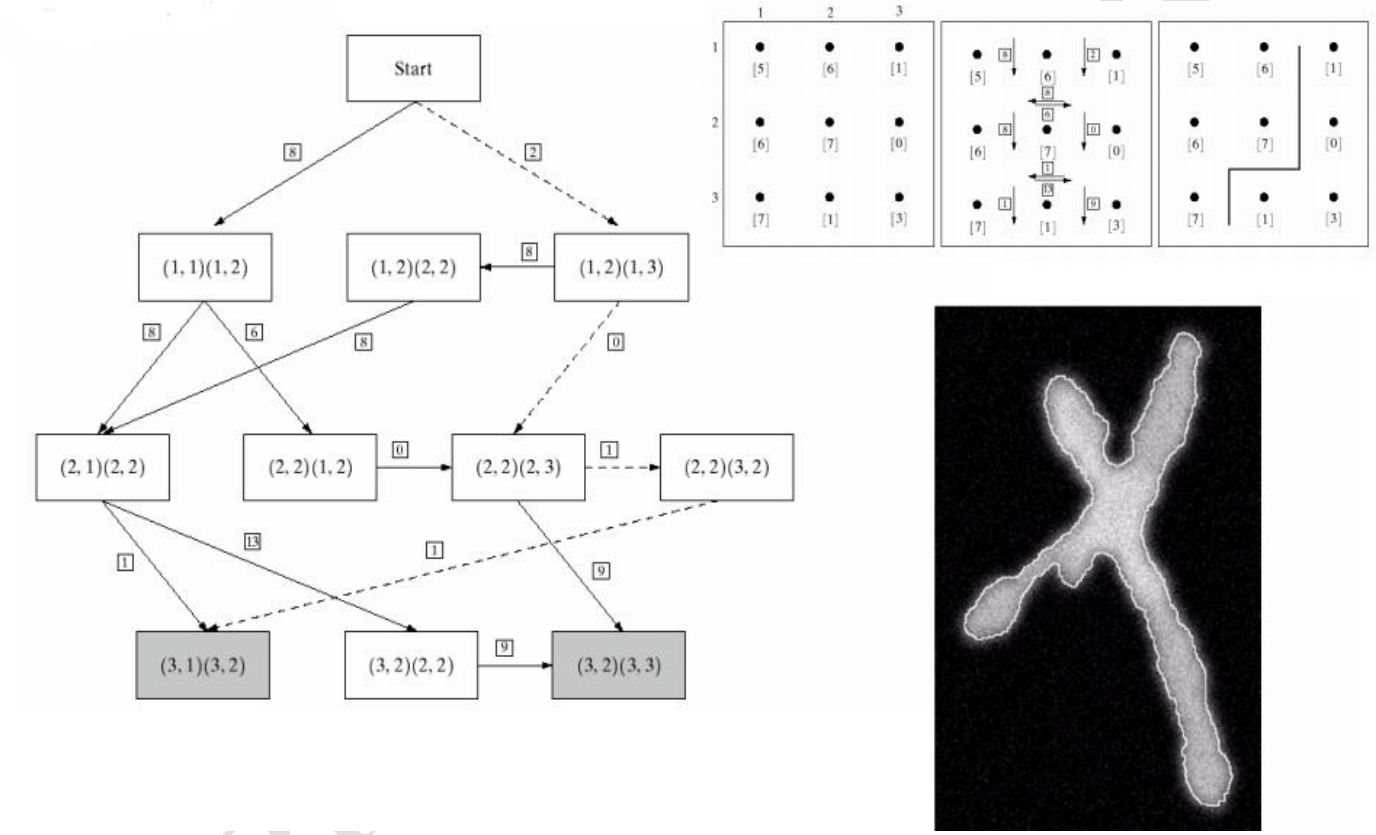
• In a directed arc ni is called parent and nj is called successor• Expansion: identifying successors of a node.• Starting (0 or root) level, last (goal) level.Cost c(n_i, n_j) associated with an arc;

• Path n1, n2,…,nk, with each n_i, being a successor of n_{j-1}. Cost of a path is the sum ofcosts of the arcs constituting the path.



- Edge element defined between two neighboor pixels p and q (x_p, y_p) (x_q, y_q)
- Associate cost with an edge elementc(p,q) = H −[f (p) − f (q)]





7.3 Thresholding

Thresholding is usually the first step in any segmentation approach and plays central role in segmentation. This makes a difference between the object and background of the image. It is carried out with an assumption that the range of intensity levels covered by objects of interest is different from the background. The segmented image g(x,y) is given by :

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

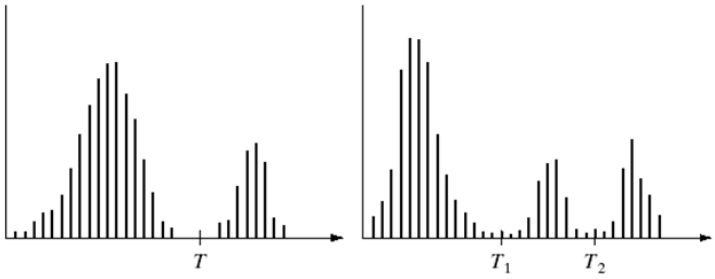


Figure 7.17 a b Single threshold Multiple threshold

Thresholding may be defined as:

(i)Single Level Thresholding

(ii) Multi Level Thresholding

(i)Single Level Thresholding

Suppose that an image f(x,y) has light object on a dark background. If we draw the histogram of this type of image it will have two dominant modes of gray levels. One group near to zero point of histogram showing dark background and one group near the maximum point of histogram shows light objects as shown in above (a) is single threshold because we have separated information of an image by single threshold .

f(x,y)>T ,is called object

f(x,y)<T it is called background

(ii) Multi Level Thresholding

Now let us consider that three dominant mode characterize the histogram for two types of light objects and black background. This is called multilevel thresholding because we have used more than one threshold for separation of image information as shown above in figure (b).

If $T_1<f(x,y)\le T_2$ One object class

If $f(x,y)> T_2$ another object class.

Types of Thresholding:

Thresholding may be categorized as follows:

(a)Global Thresholding: If the thresholding "T" depends upon only the gray levels of an image.

(b)Local Thresholding: If the thresholding "T" depends upon the gray levels of an image and some local properties also.

(c) Adaptive Thresholding: When the thresholding "T" depends upon the gray levels of an image, local properties and spatial coordinates of pixels. This is called dynamic thresholding.

Global Thresholding

In this approach, first partition the image histogram by using a single global threshold 'T'. Segmentation is then accomplished by scanning the image pixel by pixel and labeling each pixel as object or background depending whether the grey level of the pixel is greater or less than the value of T. In global thresholding there are two dominant modes light and dark region.

The pixels are assigned and labeled are also assigned .This technique is specially used in industrial areas where control of illumination is possible. How to obtain T using global thresholding as follows:

(ii) Select an initial estimate for T.

(iii) Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels ≤ T

(iv) Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

(v) Compute a new threshold value:

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

(vi) Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit.

Adaptive Thresholding: It changes the threshold dynamically over the image which is also known as dynamic thresholding. Adaptive thresholding works well in situation where image is affected by non-uniform illumination problem.

When there are more than two dominant modes in an image we can't use global thresholding. In global thresholding single threshold is used. But in adaptive thresholding we use more than one threshold value. Adaptive thresholding is the method where the threshold value is calculated for

smaller regions or blocks and therefore, there will be different threshold values for different regions and on each region we perform global thresholding. We keep diving the image into the regions until we get two dominant modes.

An approach to handling situations in which single value thresholding will not work is to divide an image into sub images and threshold these individually. Since the threshold for each pixel depends on its location within an image this technique is said to *adaptive*

Procedure for Adaptive Thresholding

1. Convolve the image with suitable statistical operator, .i.e. mean or median.
2. Subtract the original from the convolved image
3. Threshold the difference image with C.
4. Invert the threshold image.

7.4. Region Based Segmentation

A region can be classified as a group of connected pixels exhibiting similar properties. The similarity between pixels can be in terms of intensity, color, etc. In this type of segmentation, some predefined rules are present which have to be obeyed by a pixel in order to be classified into similar pixel regions. Region-based segmentation methods are preferred over edge-based segmentation methods in case of a noisy image. Region-Based techniques are further classified into 2 types based on the approaches they follow.

1. Region growing method
2. Region splitting and merging method

The objectives of image segmentation are to partition an image into multiple regions. These techniques are based on finding the regions directly. To apply this technique following basic formulation condition must be applied. Let R represent the entire image region, segmentation partition R into n-sub regions as R_1,R_2,R_3,R_4…..Rn .

7.4.1 Region Extraction

Suppose the spatial domain on which the image is defined is denoted by V. The image segmentation techniques divides V into say n regions denoted by R_i (i=1,2,……n) such that

1. $\bigcup_{i=1}^{n} R_i$ = V
2. $R_i \cap R_j = \emptyset$
3. Prop (R_i) = True
4. Prop ($R_i \cup R_j$) = False

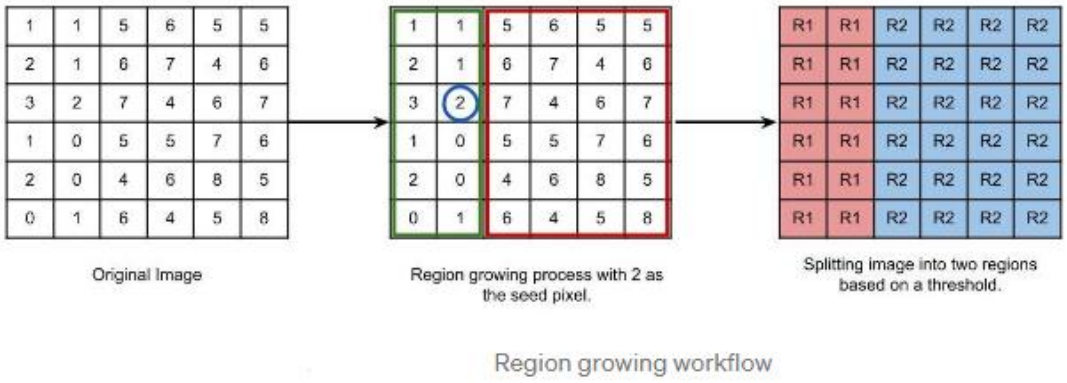
The first (1.) condition says that every pixel of the image domain is mapped to one region or the other. The second (2.) condition ensures that a pixel is mapped to only one region i.e. It indicates that multiple regions must be disjoint. Third condition (3.) indicates that the regions are defined on some property i.e. It indicates that properties must be satisified by all the pixels in segmented regions. Finally maximality of each region is assured by the fourth condition (4.) i.e. It indicates that properties of region Ri and Rj are different.

7.4.2 Region Growing

- ✓ Region Growing is a technique that group's pixels or sub-regions into larger region based on some pre-defined criteria. Let us pick an arbitrary pixel (r,c) from the domain of the image to be segmented. This pixel is called *seed pixel* and this pixel belongs to some other regions.
- ✓ Examine the nearest neighbors (4 or 8-neighbors depending on the connectivity) of (r,c) one by one and a neighboring pixel is accepted to belong to the same region as (r,c) if they together satisfy the homogeneity property of a region.
- ✓ Once a new pixel is accepted as a member of the current region, the nearest neighbors of this new pixel are examined. This pixel goes on recursively until no more pixels is accepted. All pixels of the current regions are marked with a unique label.
- ✓ Another seed pixel is picked up and the same procedure is repeated. Region labeling is done until every pixel is assigned to some region or the other. Main problem of this approach in addition to large execution time are: Selection of property to be satisfied and Selection of seed point.
- ✓ Consider a seed pixel of 2 in the given image and a threshold value of 3, if a pixel has a value greater than 3 then it will be considered inside the seed pixel region.

Otherwise, it will be considered in another region. Hence 2 regions are formed in the following image based on a threshold value of 3.

Figure 7.18 Region Growing Workflow



7.4.3 Region Splitting

- ✓ Suppose we try to satisfy the homogeneity property over a rectangular region. [g(r,c)≤ th].If the gray level present in the region do not satisfy the property ,Divide into four equal quadrants. If the property is satisfied => Leave as it is.
- ✓ Graph theory => Region [node] .Then a region is split into four children if it does not satisfy the given property. If the property is satisfied => Node is left unaffected.
- ✓ This method is applicable to images whose number of rows and columns are integer power of 2. Example: Consider 8 x 8 Image with gray level ranging from 0 to 7. Threshold value (th) = 3.

5	6	6	6	7	7	6	6
6	7	6	7	5	5	4	7
6	6	4	4	3	2	5	6
5	4	5	4	2	3	4	6
1	3	2	3	3	2	4	7
0	0	1	0	2	2	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	6

Figure 7.19 Region Splitting

✓ Perform the split operation. => Start from 4 x 4. [Splitting process to create m/2 x m/2].

7.4.4 Region Merging

✓ This method is exactly opposite to the region splitting method. Region splitting method => Top-down method. Region merging method => Bottom-up method.

✓ This method is also applicable to images whose number of rows and number of columns is some integer power of 2. At any level of merging, check if four adjacent regions arranged in 2 x 2 together satisfies the homogeneity property. If yes => Merge regions to a single homogeneous region => Otherwise leave as they are.In terms of graph theory => Child nodes are removed if the parent node satisfies homogeneity.

5	6	6	6	7	7	6	6
6	7	6	7	5	5	4	7
6	6	4	4	3	2	5	6
5	4	5	4	2	3	4	6
1	3	2	3	3	2	4	7
0	0	1	0	2	2	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	6

Figure 7.20 Region Merging

- Perform the merge operation => Start from 2 x 2 [Merging process to create 2m x 2m]

7.4.5 Split and Merge

- If most of the homogeneous regions are small =>takes more time. (Split Operation).No a priori knowledge available about the size of regions => More Time. It is a hybrid approach which combines both splitting and merging technique. Suppose we start with rectangular regions of size m x m pixels.
- To each region homogeneity property is tested. If test fails, the regions is split into four quadrants each of size m/2x m/2. If the region satisfies the homogeneity property => Merging process is followed to form a region of size (2m) x (2m).

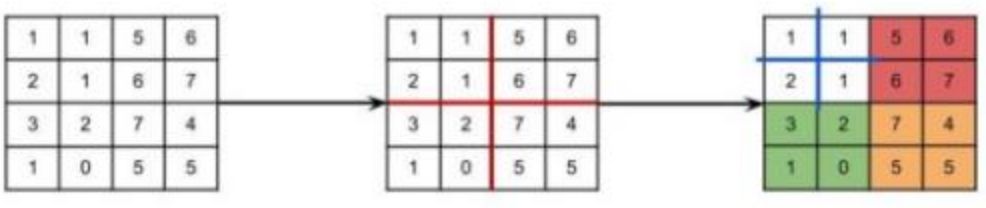


Figure 7.21 :Region Spliting and Merging