Image Segmentation

Segmentation was used to identify the object of image that we are interested. We have three approaches to do it. The first is Edge detection. The second is to use threshold. The third is the region-based segmentation. It does not mean that these three of that method can solve all of the problems that we met, but these approaches are the basic methods in segmentation.

1. Introduction

We first discuss from the case of the monochrome and static images. The fundamental problem in segmentation is to partition an image into regions. Segmentation algorithms for monochrome images are generally based on one of the following two basic categories. The first one is *Edge-based segmentation*. The second one is *Region-based segmentation*. Another method is to use the threshold. It belongs to *Edge-based segmentation.*

There are three goals that we want to achieve:

1. The first one is speed. Because we need to save the time from segmentation and give complicate compression more time to process.
2. The second, one is to have a good shape matching even under less computation time.
3. The third, one is that the result of segmenting shape will be intact but not fragmentary which means that we want to have good connectivity.

2. Edge-Based Segmentation

The focus of this section is on the segmentation methods based on detection in sharp, local changes in intensity. The three types of image feature in which we are interested are isolated points, lines , and edges. Edge pixels are pixels at which the intensity of an image function changes abruptly.

2.1. Fundamental

We know local changes in intensity can be detected using derivatives. For

reasons that will become evident , first- and second-order derivatives are particularly well suited for this purpose.

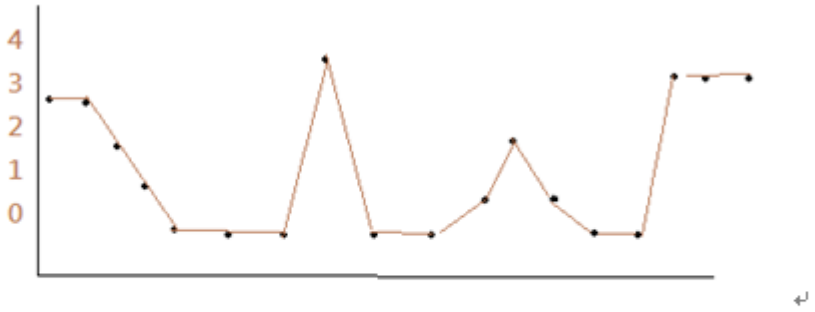


Image strip	3	3	2	1	0	0	4	0	0	1	2	1	0	0	4	4	4
1st. derivative	0	-1	-1	-1	0	4	-4	0	1	1	-1	-1	0	4	0	0	0
2nd. derivative	0	-1	-0	0	1	4	-8	4	1	0	-2	0	1	4	-4	0	0

Figure 2.1 The intensity histogram of image

We have following conclusions from Figure 2.1 The intensity histogram of image Figure 2.1:

First-order derivatives generally produce thicker edges in an image.

1. Second-order derivative have a stronger response to fine detail, such as thin lines, isolated points, and noise.
2. Second-derivatives produce a double-edge response at ramp and step transitions in intensity.
3. The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light.

2.2. Isolated Points

It is based on the conclusions reached in the preceding section. We know that point detection should be based on the second derivative, so we expect Laplacian mask.

1	1	1
1	-8	1
1	1	1

Figure 2.2 The mask of isolation point

We can use the mask to scan the all point of image , and count the response of every point, if the response of point is greater than T(the threshold we set), we can define the point to 1(light), if not, set to 0(dark).

$$G(x,y)=\begin{cases}1 & \text{if } |R(x,y)|\geq T\\ 0 & \text{otherwise}\end{cases} \tag{2.1}$$

2.3. Line Detection

As discussion in section2.1,we know the second order derivative have stronger response and to produce thinner lines than 1st derivative. We can get four different direction of mask.

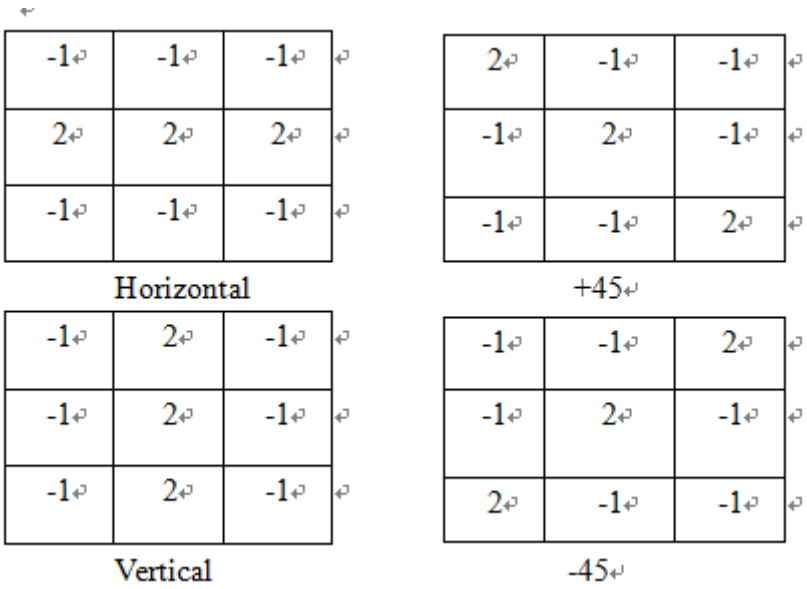


Figure 2.3 Line detection masks.

Let talk about how to use the four masks to decide which direction of mask is better than others. Let R_1, R_2, R_3 and R_4 denote the response of the masks in Figure 2.3. If at a point in the image , $|R_k|>|R_j|$,for all j ≠ k. $|R_1|>|R_j|$ for j=2,3,4,that point is said to be more likely associated with a line in the direction of mask k.

2.4. Edge detection

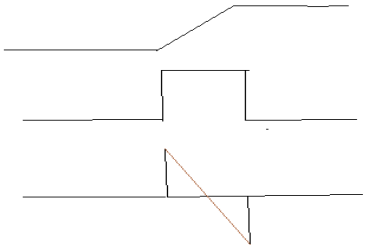


Figure 2.4 (a) Two region of constant intensity separated by an ideal vertical ramp edge.(b)Detail near the edge, showing a horizontal intensity profile.

We conclude from the observation that the magnitude of 1st derivative can be used to detect the presence of an edge at a point in an image. The 2nd derivative have two properties : (1) it produces two values for every edge in an image.(2)its zero crossings can be used for locating the center of thick edges.

2.4.1. Basic Edge Detection (gradient)

The image gradient is to find edge strength and direction at location (x,y) of image, and defines as the vector.

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{2.2}$$

The magnitude (length) of vector ∇f , denoted as M(x,y)

$$\text{mag}(\nabla f) = \sqrt{g_x + g_y} \tag{2.3}$$

The direction of the gradient vector is given by the angle

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] \tag{2.4}$$

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

We are dealing with digital quantities,so a digital approximation of the partial derivatives over a neighborhood about a point is required.

1. Roberts cross-gradient operators. Roberts [1965].

-1	0
0	1

(a)

0	-1
1	0

(b)

Figure 2.5 Roberts mask

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5) \tag{2.5}$$

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6) \tag{2.6}$$

2. *Prewitt operator*

-1	-1	-1
0	0	0
1	1	1

(c)

-1	0	1
-1	0	1
-1	0	1

(d)

0	1	1
-1	0	1
-1	-1	0

(e)

-1	-1	0
-1	0	1
0	1	1

(f)

Figure 2.6 Prewitt's mask

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \tag{2.7}$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \tag{2.8}$$

3. *Sobel operator*

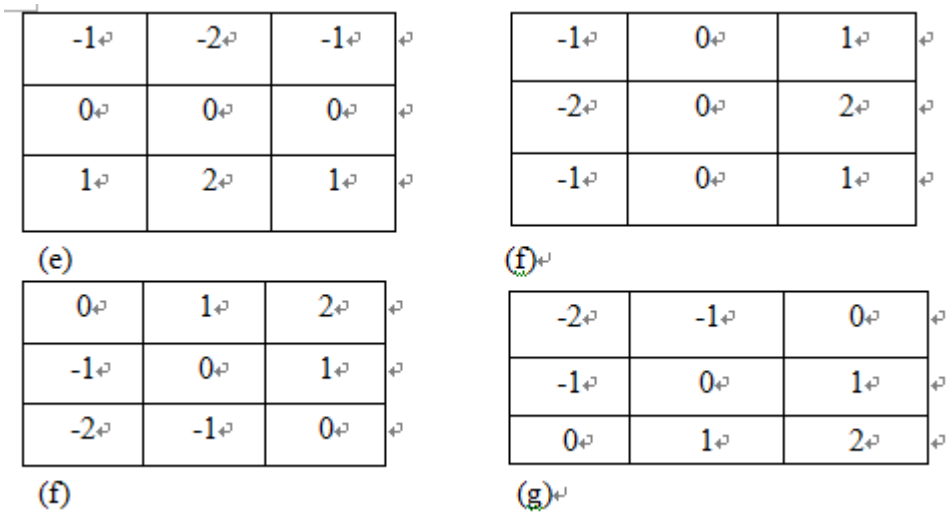


Figure 2.7 (a)~(g) are region of an image and various masks used to compute the gradient at the point labeled z_5

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{2.9}$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{2.10}$$

The Sobel mask uses 2 in the center location for image smoothing. The Prewitt masks are simpler to implement than Sobel masks, but the Sobel masks have better noise-suppression(smoothing) characteristics makes them preferable.

In the previous discussion, we just discuss to obtain the g_x and g_y. However, this implementation is not always desirable ,so an approach used frequently is to approximately the magnitude of the gradient by absolute values:

$$M(x, y) \approx |g_x| + |g_y| \tag{2.11}$$

2.4.2. The Marr-Hildreth edge detector(LoG)

This method in use at the time were based on using small operators ,and we discuss previously the 2nd derivative is better than 1st derivative in small operator.
Then we use Laplacian to make it.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Figure 2.8 5x5mask of LOG

The Marr-Hildreth algorithm consists of convolving the LoG filter with an input image, $f(x, y)$.

$$g(x, y) = [\nabla^2 G(x, y)] \otimes f(x, y) \tag{2.12}$$

Because these are linear process, Eq(2.4-9) can be written also as

$$g(x, y) = \nabla^2 [G(x, y) \otimes f(x, y)] \tag{2.13}$$

It's edge-detection algorithm may be summarized as follow:

1. Filter the input image with an $n \times n$ Gaussian lowpass filter(It can smooth the large numbers of small spatial details).
2. Compute the Laplacian of the image resulting from Step1 using.
3. Finding the zero crossings of the image from Step2.

To specify the size of Gaussian filter, recall that about 99.7% of the volume under a 2-D Gaussian surface lies between $\pm 3\sigma$ about the mean. So $n \geq 6\sigma$.

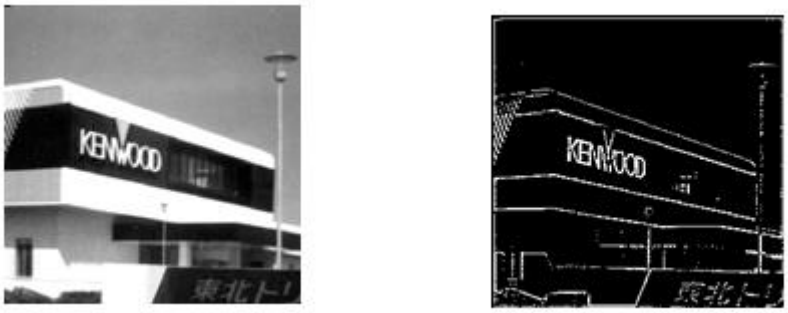


Figure 2.9 (a) Image of input. (b) After using the LoG with threshold 200.

2.5. Edge Linking and Boundary Detection

Edge detection should yield sets of pixel lying on edge, but noise would breaks in the edges due to nonuniform illumination. Therefore, edge detection typically is followed by linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries.

2.5.1. Local processing

This edge linking is to analyze the characteristics of pixels in a small neighborhood about every point (x,y) that has been declared an edge point by previous techniques.

The two principle properties used for establishing similarity of edge pixels are

1. the strength(magnitude)

$$\left|M(s,t)-M(x,y)\right| \le E \tag{2.14}$$

S_{xy} denote the set of coordinates of a neighborhood centered at point (x,y).

E is a positive threshold.

2. The direction of the gradient vector

$$\left|\alpha(s,t)-\alpha(x,y)\right| \le A \tag{2.15}$$

A is a positive angle threshold.

2.5.2. Regional processing

Often, the location of regions of interest in an image are known or can be determined. In such situations ,we can use techniques for linking pixels on a regional basis, with the desired result being an approximation to the boundary of the region.
We discuss the mechanics of the procedure using the following Fig2.6

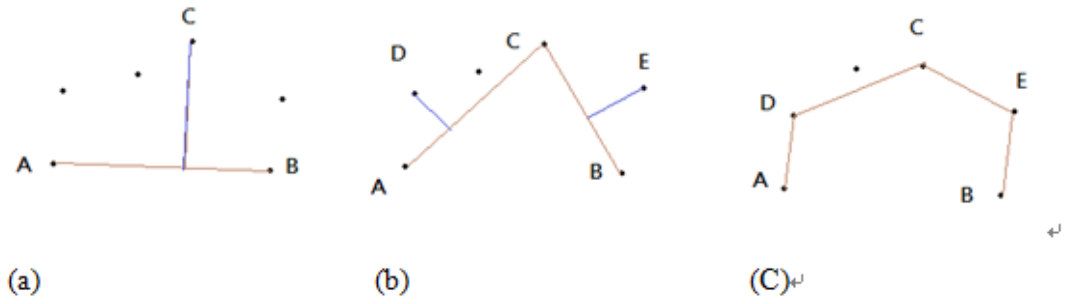


Figure 2.10 illustration of the iterative polygonal fit algorithm

An algorithm for finding a polygonal fit may be stated as follows:

1. Specify two starting point A and B.
2. Connected A and B ,and compare which points are defined vertices of the polygon max and larger than T(threshold).
3. Then connect all the point we have, thus compare it like step 2 until distance between every point and connected lines vertices is smaller than T.

2.5.3. Global processing using the Hough transform

In regional processing, it makes sense to link a given set of pixels only if we know that they are part of the boundary of a meaningful region. Often, we have to work with unstructured environments in which all we have.

We can use Hough transform to use coordinate transition to find out the similar point in other place.

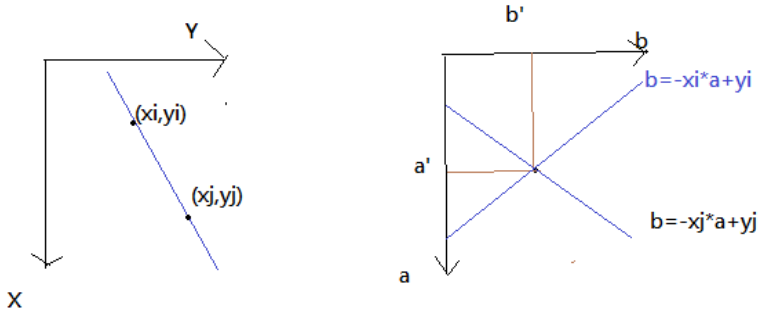


Figure 2.11 (a)xy-plane. (b)Parameter space

Consider a point (x_i, y_i) in the xy-plane and the general equation of a straight line in slope-intercept from, $y_i = ax_i + b$,a second point (x_j, y_j) also has a line in parameter space associated with it .but a practical difficulty with the approach, is that a (slope of a line)approaches infinity as the line approaches the vertical direction. One way to use the normal representation of a line:

$$x\cos\theta + y\sin\theta = \rho \qquad (2.5\text{-}3)$$

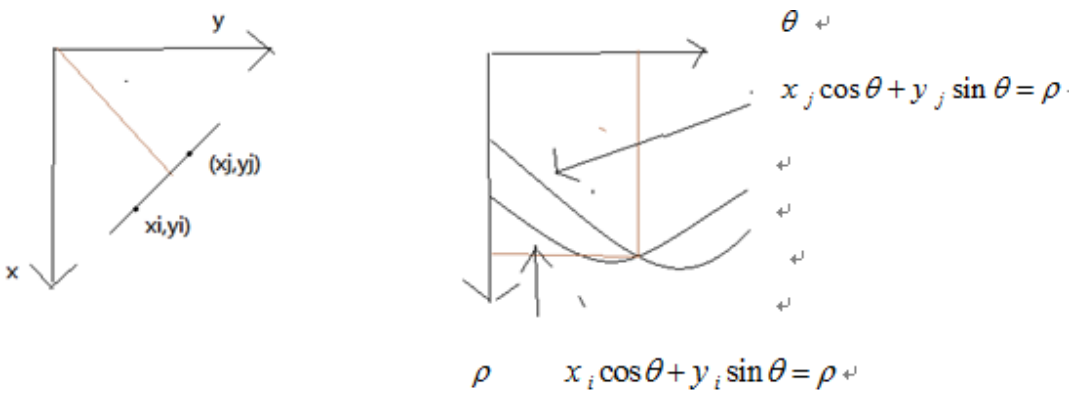


Figure 2.12 (a) A line in the xy-plane. (b)Sinusoidal curves in the *ρθ*-plane; the point of the intersection *(ρ,θ)*corresponds to the line passing through point *(xi,yi)* and *(xj,yj)*in the *xy*-plane

2.6. Segmentation Using Morphological Watersheds

2.6.1. Background

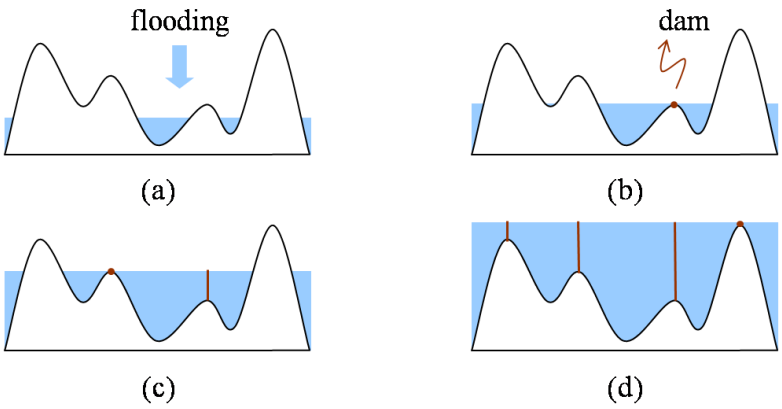
The concept of watershed is based on visualizing an image in three dimensions: two spatial coordinates and intensity. We consider three types of points:

1. The points belonging to the local minimum.
2. The points where a drop of water, if placed at the locations of these points, would fall to a single local minimum. It is called *catchment basin* or *watershed*.
3. The points where water would be equally likely to fall to more than one local minimum. They are similar to the crest lines on the topographic surface and are termed *divide lines* or *watershed lines*.

The two main properties of watershed segmentation result are continuous boundaries and over-segmentations. As we know, the boundaries that made by the watershed algorithm are exact the watershed lines in the image. Therefore, the numbers of region basically will be equal to the numbers of minima in the image. There are two steps to achieve the solution using marker:

1. Preprocessing
2. Defining the criteria that the markers have to be satisfy.

The following figures are the mechanism to construct dam.



Figur2.10(a)~(d) Watershed algorithm.

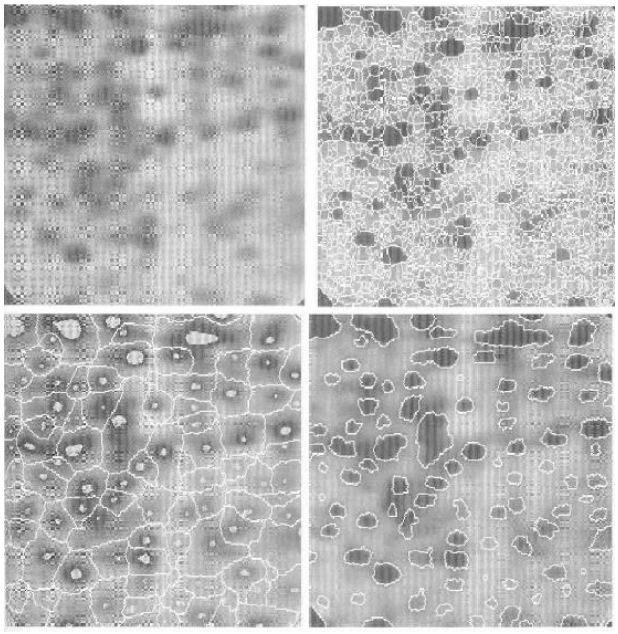
Supposed that figure2.10 are the image of input , and the height of the "mountain" is proportional to intensity values input image. We start to flood water from below by letting water rise through the holes at a uniform rate. Figure (b) we see that water now has risen into the first and second catchment basins. So we will construct a dam to stop it to overflowing ,and do the same motion step by step.

2.6.2. The Use of Markers

Direct application of the watershed segmentation algorithm in the form discussed in the previous section generally leads to *oversegmentation* due to noise and other local irregularities of gradient.

An approach used to control oversegmentation is based on the concept of markers. Then we have markers. We have *internal* markers, associated with objects of interest, and *external* markers. A procedure for markers selection typically will consist of two principal steps: (1)preprocessing (usually smoothing) (2)definition of a set of criteria that markers must satisfy.(to do edge detection for every small region)

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



Thresholding

2.7. Basic Global Thresholding

As the fact that we need only the histogram of the image to segment it, segmenting images with Threshold Technique does not involve the spatial information of the images. Therefore, some problem may be caused by noise, blurred edges, or outlier in the image. That is why we say this method is the simplest concept to segment images.

When the intensity distributions of objects and background pixels are sufficiently distinct, it is possible to use a single(global) threshold applicable over the entire image. The following iterative algorithm can be used for this purpose:

1. Select an initial estimate for the global threshold, T.
2. Segment the image using T as

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \qquad (2.16)$$

   This 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.
4. Compute a new threshold values:

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

5. Repeats Step2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter.

2.8. Optimum Global Thresholding Using Otsu's Method

Thresholding may be viewed as a statistical-decision theory problem whose objective is to minimize the average error incurred in assigning pixels to two or more groups.

Let $\{0,1,2,\ldots,L\text{-}1\}$ denote the L distinct intensity levels in a digital image of size $M \times N$ pixels, and let n_i denote the number of pixels with intensity i. The total number, MN, of pixels in the image is $MN = n_0 + n_1 + n_2 + \ldots + n_{L-1}$. The normalized

histogram has components $p_i = n_i / MN$, from which it follows that

$$\sum_{i=0}^{L-1} p_i = 1, p_i \geq 0 \tag{2.17}$$

Now, we select a threshold $T(k) = k, 0 < k < L-1$, and use it to threshold the input image into two classes, C_1 and C_2, where C_1 consist with intensity in the range $[0, k]$ and C_2 consist with $[k+1, L-1]$.

Using this threshold , $P_1(k)$,that is assigned to C_1 and given by the cumulative sum.

$$P_1(k) = \sum_{i=0}^{k} p_i \tag{2.18}$$

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \tag{2.19}$$

The validity of the following two equations can be verified by direct substitution of the preceding result:

$$P_1 m_1 + P_2 m_2 = m_G \tag{2.20}$$

$$P_1 + P_2 = 1 \tag{2.21}$$

In order to evaluate the "goodness" of the threshold at level k we use the normalized, dimensionless metric

$$\eta = \frac{\sigma_B^2(k)}{\sigma_G^2} \tag{2.22}$$

Where σ_G^2 is the *global variance*

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i \tag{2.23}$$

And σ_B^2 is the *between-class variance*, define as :

$$\sigma_B^2 = P_1 (m_1 - m_G)^2 + P_2 (m_2 - m_G)^2 \tag{2.24}$$

$$\sigma_B^2(k) = P_1 P_2 (m_1 - m_2)^2 = \frac{(m_G P_1(k) - m(k))^2}{P_1(k)(1 - P_1(k))} \tag{2.25}$$

Indicating that the *between-class variance* and η is a measure of *separability*

between class.

Then, the optimum threshold is the value, k^* ,that maximizes $\sigma_B^2(k)$

$$\sigma_B^2(k^*) = \max_{o \le k \le L-1} \sigma_B^2(k) \tag{2.26}$$

In other word, to find k^* we simply evaluate (2.7-11) for all *integer values* of k
Once k^* has been obtain, the input image $f(x, y)$ is segmented as before:

$$g(x, y) = \begin{cases} 1 & \text{if f(x,y)} \ge \text{k*} \\ 0 & \text{if f(x,y)} \le \text{k*} \end{cases} \tag{2.27}$$

For x = 0,1,2,…,M-1 and y = 0,1,2…,N-1. This measure has values in the range

$$0 \le \eta(k^*) \le 1 \tag{2.28}$$

2.8.1. Using image Smoothing/Edge to improve Global Threshold

Compare the difference between preprocess of smoothing and Edge detection.

	Smoothing	Edge detection
What situation is more suitable for the method	Large object we are interested.	Small object we are interested

2.9. Multiple Threshold

For three classes consisting of three intensity intervals, the between-class variance is given by:

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2 \tag{2.29}$$

The following relationships hold:

$$P_1 m_1 + P_2 m_2 + P_3 m_3 = m_G \tag{2.30}$$

$$P_1 + P_2 + P_3 = 1 \tag{2.31}$$

The optimum threshold values,

$$\sigma_B^2(k_1^*, k_2^*) = \max_{o < k_1 < k_2 < L-1} \sigma_B^2(k_1, k_2) \tag{2.32}$$

Finally, we note that the separability measure defined in section 2.7.2 for one threshold extends directly to multiple thresholds:

$$\eta(k_1^*, k_2^*) = \frac{\sigma_B^2(k_1^*, k_2^*)}{\sigma_G^2} \tag{2.33}$$

2.10. Variable Thresholding

Image partitioning

One of the simplest approaches to variable threshold is to subdivide an image into nonoverlapping rectangles. This approach is used to compensate for non-uniformities in illumination and/or reflection.

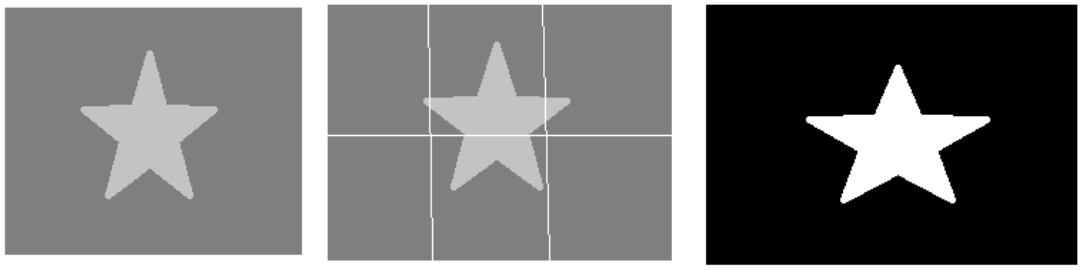


Figure 2.13 (a)Noisy, shaded image (b) Image subdivide into six subimages. (c)Result of applying Otsu's method to each subimage individually.

Image subdivision generally works well when the objects of interest and the background occupy regions of reasonably comparable size. When this is not the case, the method typically fail.

Variable thresholding based on local image properties

We illustrate the basic approach to local thresholding using the standard deviation and mean of the pixels in a neighborhood of every point in an image. Let σ_{xy} and m_{xy} denote the standard deviation and mean value of the set of pixels contained in a neighborhood, S_{xy}.

$$g(x,y) = \begin{cases} 1 & \text{if Q(local parameters) is true} \\ 0 & \text{if Q(local parameters) is true} \end{cases} \tag{2.34}$$

Where Q is a predicate based on parameter computes using the pixels in neighborhood.

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} true & \text{if } f(x,y) > a\sigma_{xy} \; AND \; f(x,y) > bm_{xy} \\ false & \text{otherwise} \end{cases} \tag{2.35}$$

Using moving average

Computing a moving average along scan lines of an image. This implementation is quite useful in document processing, where speed is a fundamental requirement. The scanning typically is carried out line by line in a zigzag pattern to reduce illumination bias.

$$m(k+1) = \frac{1}{n}\sum_{i=k+2-n}^{k+1} z_i = m(k) + \frac{1}{n}(z_{k+1} - z_{k-n}) \tag{2.36}$$

Let z_{k+1} denote the intensity of the point encountered in the scanning sequence at step k+1. Where n denote the number of point used in computing the average.

$m(1) = z_1/n$ is the initial value. Segmentation is implemented using Eq(2.7-1) with $T_{xy} = bm_{xy}$,where b is constant and m_{xy} is the moving average at point *(x,y)* in the input Image.

Multivariable Thresholding

We have been concerned with thresholding based on a single variable: gray-scale intensity. A notable example is color imaging, where red(R),greed(G), and blue(B) components are used to form a composite color image. It can be represented as a 3-D vector, **z**= $(z_1, z_2, z_3)^T$,whose component are the RGB colors at a point.

Let **a** denote the average reddish color in which we are interested, D(**z,a**) is a distance measure between an arbitrary color point, **z**, then we segment the input image as follows:

$$g(x,y) = \begin{cases} 1 \text{ if } D(z,a)<T \\ 0 \text{ otherwise} \end{cases} \quad (2.37)$$

Note that the inequalities in this equation are the opposite of the equation we used before. The reason is that the equation D(**z**,**a)**=T defines a volume.

$$D(z,a) = \|z-a\| = [(z-a)^T(z-a)]^{\frac{1}{2}} \quad (2.38)$$

$$D(z,a) = \|z-a\| = [(z-a)^T C^{-1}(z-a)]^{\frac{1}{2}} \quad (2.39)$$

A more powerful distance measure is the so-called *Mahalanobis distance.*
Where **C** is the covariance matrix of the **z**s, when **C**=**I**, the identity matrix.

3. Region-Based Segmentation

3.1. Region Growing

Region growing segmentation is an approach to examine the neighboring pixels of the initial “seed points” and determine if the pixels are added to the seed point or

not.

Step1. Selecting a set of one or more starting point (seed) often can be based on the nature of the problem.

Step2. The region are grown from these seed points to adjacent point depending on a threshold or criteria(8-connected) we make.

Step3. Region growth should stop when no more pixels satisfy the criteria for inclusion in that region

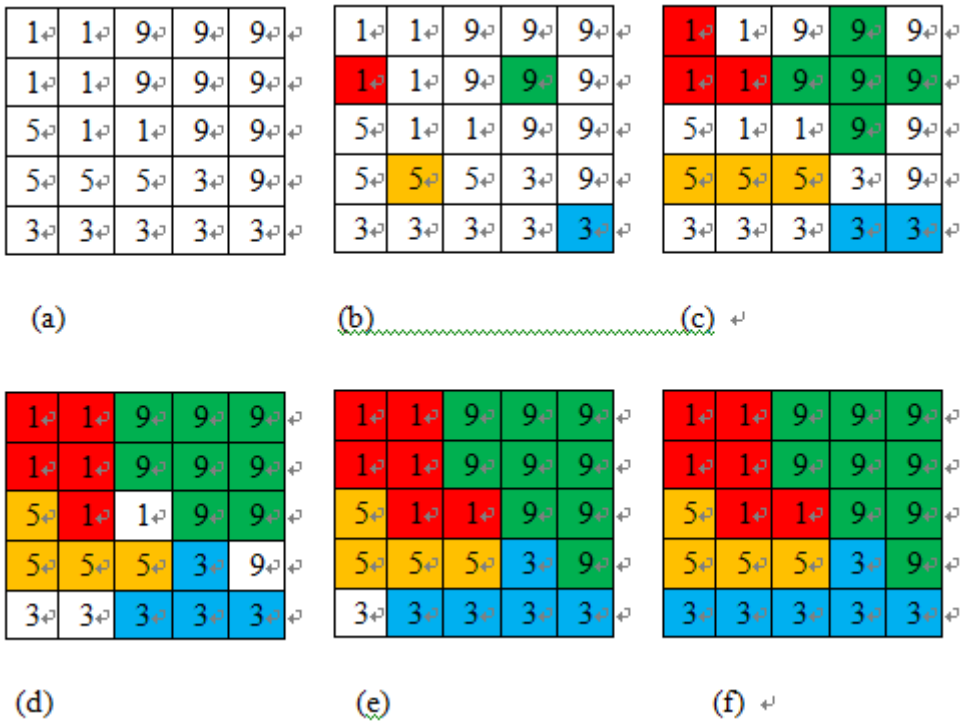


Figure 3.1 (a)Original image (b)Use step 1 to find seed based on the nature problem.(c) Use Step 2(4-connected here) to growing the region and finding the similar point. (d)(e) repeat Step 2. Until no more pixels satisfy the criteria. (f) The final image.

Then we can conclude several important issues about region growing：

1. The suitable selection of seed points is important. The selection of seed points is depending on the users.
2. More information of the image is better. Obviously, the connectivity or pixel adjacent information is helpful for us to determine the threshold and seed points.
3. The value, “minimum area threshold”. No region in region growing method result will be smaller than this threshold in the segmented image.
4. The value, “Similarity threshold value“. If the difference of pixel-value or the difference value of average gray level of a set of pixels less than “Similarity threshold value”, the regions will be considered as a same region.
5. The result of an image after region growing still have point’s gray-level higher

than the threshold but not connected with the object in image.

We briefly conclude the advantages and disadvantages of region growing.

Advantages：

1. Region growing methods can correctly separate the regions that have the same properties we define.
2. Region growing methods can provide the original images which have clear edges the good segmentation results.
3. The concept is simple. We only need a small numbers of seed point to represent t he property we want, then grow the region.
4. We can choose the multiple criteria at the same time.

It performs well with respect to noise, which means it has a good shape matching of its result.

Disadvantage：

1. The computation is consuming, no matter the time or power.
2. This method may not distinguish the shading of the real images.

In conclusion, the region growing method has a good performance with the good shape matching and connectivity. The most serious problem of region growing method is the time consuming.

3.2. Simulation of Region Growing Using C++.

Figure 3.2 Lena image after using region growing, there are 90% pixels have been classified. Threshold/second: 20/4.7 seconds.

The method have connected region, but it need more time to process.

3.3. Region Splitting and Merging

An alternative method is to subdivide an image initially into a set of arbitrary, disjoint regions and then merge and/or split the region.

The quadtrees means that we subdivide that quadrant into subquadrants, and it is the following as:

1. Split into four disjoint quadrants any region R_i for which $Q(R_i) = FALSE$ (means the region don't satisfy same logic in R_i) .
2. When no further splitting is possible, merge any adjacent region R_j and R_k for which $Q(R_j \cup R_k) = TRUE$ (means that R_j and R_k have similarity we define in some where) .
3. Stop when no further merging is possible.

Advantage of region splitting and merging：

We can split the image by choosing the criteria we want, such as segment variance or mean of the pixel-value. And the splitting criteria can be different from the merging criteria.

Disadvantage of it：

1. Computation is intensive.
2. Probably producing the blocky segments.

The blocky segment problem effect can be reduce by splitting for higher resolution, but at the same time, the computational problem will be more serious.