

Dam Construction

Dam construction is based on binary images, which are members of 2-D integer space Z^2 . The simplest way to construct dams separating sets of binary points is to use morphological dilation.

The basics of how to construct dams using dilation are illustrated in Fig. Figure (a) shows portions of two catchment basins at flooding step $n - 1$ and Fig. (b) shows the result at the next flooding step, n . The water has spilled from one basin to the other and, therefore, a dam must be built to keep this from happening. In order to be consistent with notation to be introduced shortly,

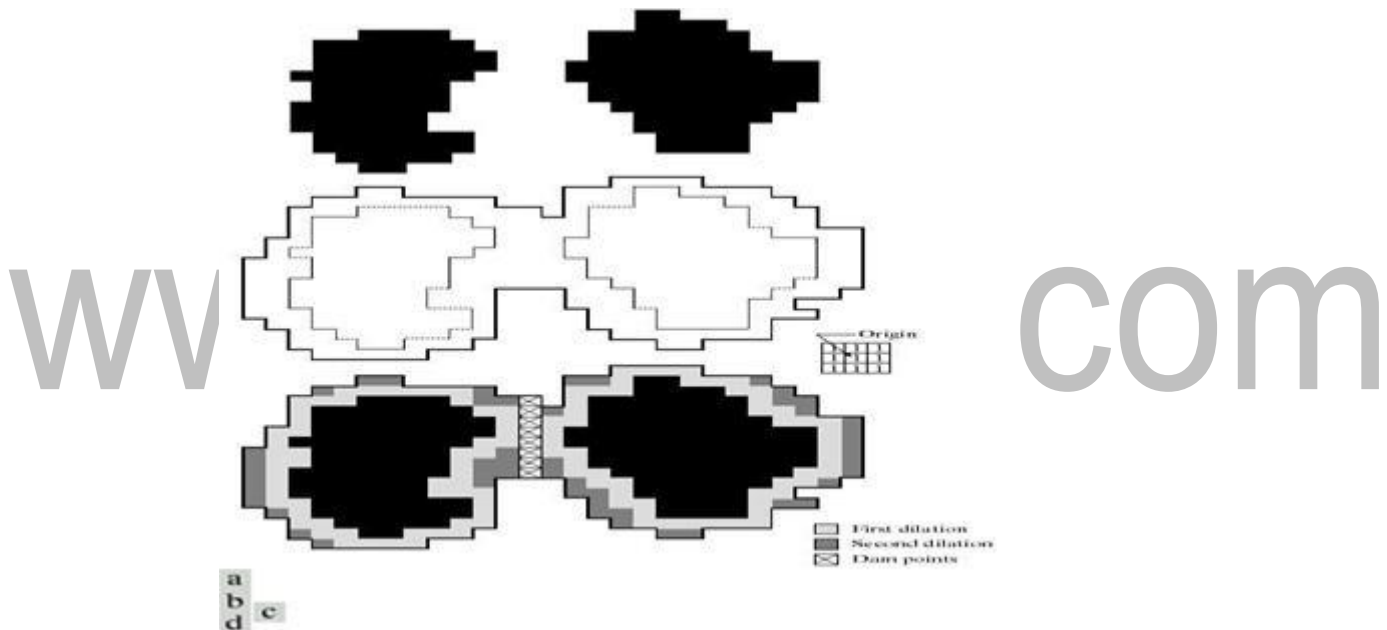


FIGURE 4.5.1(a) Two partially flooded catchment basins at stage of flooding. (b) Flooding at stage showing that water has spilled between basins. (c) Structuring element used for dilation. (d) Result of dilation and dam construction.)

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010 - Page-773)

let M_1 and M_2 denote the sets of coordinates of points in two regional minima. Then let the set of coordinates of points in the *catchment basin* associated with these two minima at stage $n - 1$ of flooding be denoted by and respectively.

These are the two black regions shown in Fig. (a). Suppose that each of the connected components in Fig.(a) is dilated by the structuring element shown in Fig. (c), subject to two conditions:

- (1) The dilation has to be constrained to q (this means that the center of the structuring element can be located only at points in q during dilation),and
- (2) The dilation cannot be performed on points that would cause the sets being dilated to merge (become a single connected component).

Figure (d) shows that a first dilation pass (in light gray) expanded the boundary of each original connected component. Note that condition (1) was satisfied by every point during dilation, and condition (2) did not apply to any point during the dilation process; thus the boundary of each region was expanded uniformly.

In the second dilation (shown in medium gray), several points failed condition (1) while meeting condition (2), resulting in the broken perimeter shown in the figure. It also is evident that the only points in q that satisfy the two conditions under consideration describe the one-pixel-thick connected path shown crossed-hatched in Fig. (d). This path constitutes the desired separating dam at stage n of flooding. Construction of the dam at this level of flooding is completed by setting all the points in the path just determined to a value greater than the maximum gray-level value of the image. The height of all dams is generally set at 1 plus the maximum allowed value in the image. This will prevent water from crossing over the part of the completed dam as the level of flooding is increased. It is important to note that dams built by this procedure, which are the desired segmentation boundaries, are connected components. In other words, this method eliminates the problems of broken segmentation lines,

Although the procedure just described is based on a simple example, the method used for more complex situations is exactly the same, including the use of the 3

X 3 symmetric structuring elements shown in Fig.(c).

1. At each step of the algorithm, the binary image is obtained in the following manner

1. Initially, the set of pixels with minimum gray level are 1, others 0.

2. In each subsequent step, we flood the 3D topography from below and the pixels covered by the rising water are 1s and others 0s. M_1, M_2 :

– Sets of coordinates of points in the two regional minima $C_{n-1}(M_1), C_{n-1}(M_2)$ -

– Sets of coordinates of points in the catchment basins associated with M_1, M_2 at stage $n-1$ of flooding (catchment basins up to the flooding level)

3. $C[n-1]$ - Union of $C_{n-1}(M_1), C_{n-1}(M_2)$

4. At flooding step $n-1$, there are two connected components. At flooding step n , there is only one connected component

– This indicates that the water between the two catchment basins has merged at flooding step n

– Use “ q ” to denote the single connected component

5. Steps - Repeatedly dilate $C_{n-1}(M_1), C_{n-1}(M_2)$ by the 3×3 structuring element shown, subject to the following condition

• Constrained to q (center of the structuring element can not go beyond q during dilation)

6. The dam is constructed by the points on which the dilation would cause the sets being dilated to merge.

-Resulting one-pixel thick connected path

7. Setting the gray level at each point in the resultant path to a value greater than the maximum gray value of the image. Usually $\max + 1$

Edge Detection Algorithm

Edge detection algorithm is followed by linking procedures to assemble edge pixels into meaningful edges.

Basic approaches

Local Processing

2. Global Processing via the Hough Transform
3. Global Processing via Graph-Theoretic Techniques

1. Local processing

Analyze the characteristics of pixels in a small neighborhood (say, 3x3, 5x5) about every edge pixels (x,y) in an image. All points that are similar according to a set of predefined criteria are linked, forming an edge of pixels that share those criteria

Criteria

1. The strength of the response of the gradient operator used to produce the edge pixel an edge pixel with coordinates (x₀,y₀) in a predefined neighborhood of (x,y) is similar in magnitude to the pixel at (x,y) if

$$|f(x,y) - f(x_0,y_0)| < E$$

2. The direction of the gradient vector an edge pixel with coordinates (x₀,y₀) in a predefined neighborhood of (x,y) is similar in angle to the pixel at (x,y) if

$$|\angle(x,y) - \angle(x_0,y_0)| < A$$

3. A Point in the predefined neighborhood of (x,y) is linked to the pixel at (x,y) if **both magnitude and direction criteria are satisfied.**

4. The process is repeated at every location in the image

5. A record must be kept

6. Simply by assigning a different gray level to each set of linked edge pixels.

7. find rectangles whose sizes makes them suitable candidates for license plates

Use horizontal and vertical Sobel operators , eliminate isolated short segments

Link conditions: gradient value > 25 , gradient direction differs < 15°

2. GLOBAL PROCESSING VIA THE HOUGHTRANSFORM

Suppose that we have an image consisting of several samples of a straight line, Hough [1962] proposed a method (commonly referred as *Hough transform*) for finding the line (or lines) among these samples (or edge pixels). Consider a point (x_i, y_i) . There is an infinite number of lines passing through this point, however, they all can be described by $y_i = a x_i + b$

This means that all the lines passing (x_i, y_i) are defined by two parameters, a and b .

This equation can be rewritten as

$$b = y_i - a x_i$$

Now, if we consider b as a function of a , where x_i and y_i are two constant the parameters can be represented by a single line in the ab -plane. The Hough transform is a process where each pixel sample in the original xy -plane is transformed to a line in the ab - plane. Consider two pixels (x_1, y_1) and (x_2, y_2) .

Suppose that we draw a line L across these points. The transformed lines of (x_1, y_1) and (x_2, y_2) in the ab -plane intersect each other in the point (a', b') , which is the description of the line L , see Figure . In general, the more lines cross point (a, b) the stronger indication that is that there is a line $y = a x + b$ image

To implement the Hough transform a two-dimensional matrix is needed, see Figure. In each cell of the matrix there is a counter of how many lines cross that point. Each line in the ab - plane increases the counter of the cells that are along its way. A problem in this implementation is that both the slope (a) and intercept (b) approach infinity as the line approaches the vertical. One way around this difficulty is to use the normal representation of a line

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

Here d represents the shortest distance between origin and the line. θ represents the angle of the shortest path in respect to the x -axis. Their corresponding ranges are $[0, 2D]$, and $[-90, 90]$, where D is the distance between corners in the image. Although

the focus has been on straight lines, the Hough transform is applicable to any other shape. For example, the points lying on the circle

$$x^2 + y^2 = c^2$$

can be detected by using the approach just discussed.

The basic difference is the presence of three parameters (c_1, c_2, c_3), which results in a 3-dimensional parameter space.

HOUGH TRANSFORM STEPS:

1. Compute the gradient of an image and threshold it to obtain a binary image.
2. Specify subdivisions in the θ -plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.
5. based on computing the distance between disconnected pixels identified during traversal of the set of pixels corresponding to a given accumulator cell.
6. a gap at any point is significant if the distance between that point and its closet neighbor exceeds a certain threshold.
7. Link criteria:
 - 1). the pixels belonged to one of the set of pixels linked according to the highest count
 - 2). no gaps were longer than 5pixels

Edge Detection

Detection of discontinuities

Discontinuities such as isolated points, thin lines and edges in the image can be detected by using similar masks as in the low- and high-pass filtering. The absolute value of the weighted sum given by equation (2.7) indicates how strongly that particular pixel corresponds to the property described by the mask; the greater the absolute value, the stronger the response.

1. Point detection

The mask for detecting *isolated points* is given in the Figure. A point can be defined to be isolated I if the response by the masking exceeds a predefined threshold:

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

Figure 4.1.1: Mask detecting isolated points.

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010.-Page-312)

2. Line Detection

- Line detection is identical to point detection.
- Instead of one mask, four different masks must be used to cover the four primary directions namely, horizontal, vertical and two diagonals.
- Thus lines are detected using the following masks

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

Figure 4.1.2 Masks for line detection

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010.-Page-699)

- Horizontal mask will result with max response when a line passed through the middle row of the mask with a constant background.
- Thus a similar idea is used with other masks.
- The preferred direction of each mask is weighted with a larger coefficient (i.e., 2) than other possible directions.
- Consider the following diagrams:



Fig4.1.3: Model of an Ideal digital edge, Model of an Ramp digital edge.

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010.-Page-312)

THICK EDGE:

- The slope of the ramp is inversely proportional to the degree of blurring in the edge.
- We no longer have a thin (one pixel thick) path.
- Instead, an edge point now is any point contained in the ramp, and an edge would then be a set of such points that are connected.
- The thickness is determined by the length of the ramp.
- The length is determined by the slope, which is in turn determined by the

degree of blurring.

- **Blurred edges tend to be thick and sharp edges tend to be thin**

Steps for Line detection:

- Apply every masks on the image
- let R_1, R_2, R_3, R_4 denotes the response of the horizontal, +45 degree, vertical and - 45 degree masks, respectively.
- if, at a certain point in the image $|R_i| > |R_j|$,
- for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i .
- Alternatively, to detect all lines in an image in the direction defined by a given mask, just run the mask through the image and threshold the absolute value of the result.
- The points that are left are the strongest responses, which, for lines one pixel thick, correspond closest to the direction defined by the mask.

2.Edge Detection

- The most important operation to detect discontinuities is *edgedetection*.
- **Edge detection is used for detecting discontinuities in gray level. First and second order digital derivatives are implemented to detect the edges in an image.**
- Edge is defined as the boundary between two regions with relatively distinct gray- level properties.
- An edge is a set of connected pixels that lie on the boundary between two regions.
- Approaches for implementing edge detection

first-order derivative (Gradient operator)

Second-order derivative (Laplacian operator)

Detection of discontinuities

Discontinuities such as isolated points, thin lines and edges in the image can be detected by using similar masks as in the low- and high-pass filtering. The absolute value of the weighted sum given by equation (2.7) indicates how strongly that particular pixel corresponds to the property described by the mask; the greater the absolute value, the stronger the response.

www.binils.com

EROSION AND DILATION IN MORPHOLOGICAL PROCESSING.

These operations are fundamental to morphological processing.

Erosion:

With A and B as sets in Z^2 , the erosion of A by B , denoted $A \ominus B$, is defined as

$$A \ominus B = \{z | (B)_z \subseteq A\}$$

In words, this equation indicates that the erosion of A by B is the set of all points z such that B , translated by z , is contained in A . In the following discussion, set B is assumed to be a structuring element. The statement that B has to be contained in A is equivalent to B not sharing any common elements with the background; we can express erosion in the following equivalent form:

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\}$$

where, A^c is the complement of A and \emptyset is the empty set.

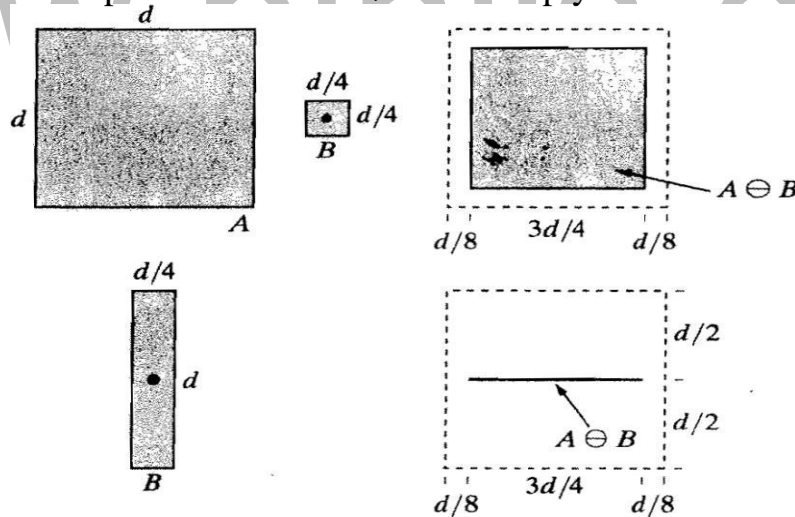


Fig4.1.4: a) Set (b) Square structuring element, (c) Erosion of by shown shaded. (d) Elongated structuring element. (e) Erosion of by using this element. The dotted border in (c) and (e) is the boundary of set A , shown only for reference.

(Source: Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010.-Page-634)

The elements of A and B are shown shaded and the background is white. The solid boundary in Fig. (c) is the limit beyond which further displacements of the origin of B would cause the structuring element to cease being completely contained in A .

Thus, the locus of points (locations of the origin of B) within (and including) this boundary, constitutes the erosion of A by B . We show the erosion shaded in Fig. (c). The boundary of set A is shown dashed in Figs. (c) and (e) only as a reference; it is not part of the erosion operation. Figure (d) shows an elongated structuring element, and Fig. (e) shows the erosion of A by this element. Note that the original set was eroded to a line. However, these equations have the distinct advantage over other formulations in that they are more intuitive when the structuring element B is viewed as a spatial mask

Thus erosion shrinks or thins objects in a binary image. In fact, we can view erosion as a *morphological filtering* operation in which image details smaller than the structuring element are filtered (re-moved) from the image

(i) Dilation

However, the preceding definitions have a distinct advantage over other formulations in that they are more intuitive when the structuring element B is viewed as a convolution mask. The basic process of flipping (rotating) B about its origin and then successively displacing it so that it slides over set (image) A is analogous to spatial convolution. Keep in mind, however, that dilation is based on set operations and therefore is a nonlinear operation, whereas convolution is a linear operation. Unlike erosion, which is a shrinking or thinning operation, dilation "grows" or "thickens" objects in a binary image. The specific manner and extent of this thickening is controlled by the shape of the structuring element used.

In the following Figure (b) shows a structuring element (in this case $B = B$ because the SE is symmetric about its origin). The dashed line in Fig. (c) shows the

original set for reference, and the solid line shows the limit beyond which any further displacements of the origin of B by z would cause the intersection of B and A to be empty. Therefore, all points on and inside this boundary constitute the dilation of A by B . Figure (d) shows a structuring element designed to achieve more dilation vertically than horizontally, and Fig. (e) shows the dilation achieved with this element

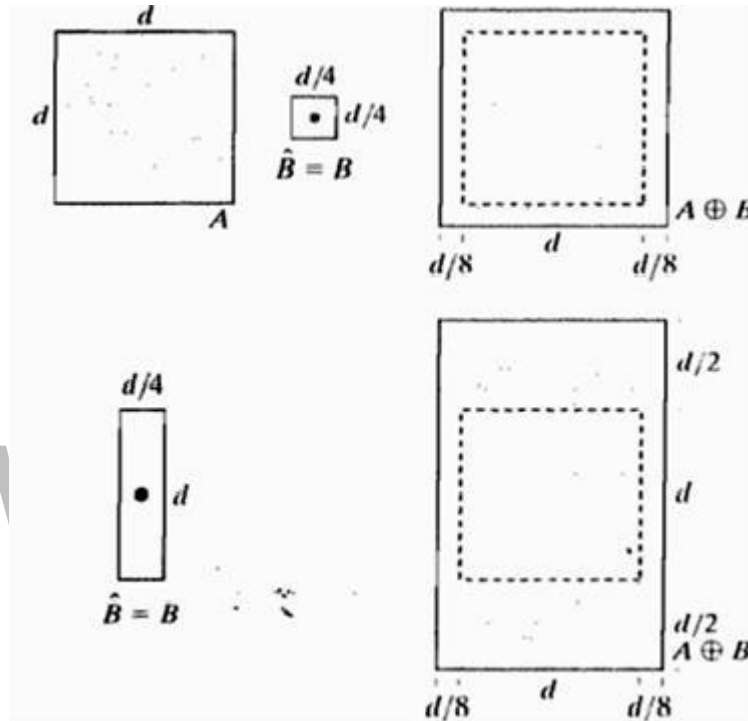


FIG:4.1.5 (a) Set (b) Square structuring element (the dot denotes the origin). (c) Dilation of by shown shaded. (d) Elongated structuring element. (e) Dilation of using this element. The dotted border in (c) and (e) is the boundary of set shown only for reference.

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010.-Page-634)

Region Growing Segmentation

Region growing is a procedure that groups pixels or sub regions in to layer regions based on predefined criteria. The basic approach is to start with a set of seed points and from there grow regions by appending to each seed these neighboring pixels that have properties similar to the seed.

The objective of segmentation is to partition an image into regions. We approached this problem by finding boundaries between regions based on discontinuities in gray levels, whereas segmentation was accomplished via thresholds based on the distribution of pixel properties, such as gray-level values or color.

Basic Formulation:

The growing continues until a *stopping rule* takes effect. The region growing has several practical problems:

how to select the seed points growing rule (or uniform criterion) stopping rule.

Let R represent the entire image region. We may view segmentation as a process that partitions R into n subregions, R_1, R_2, \dots, R_n , such that The predicate $P(R_i)$ is used to check the condition. In any region, if $P(R_i) = \text{true}$, then image is subdivided into various subimages. If $P(R_i) = \text{false}$, then divide the image into quadrants. If $P(R_i) = \text{false}$, then further divide the quadrants into sub quadrants

Region Growing:

As its name implies, region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria. The basic approach is to start with a set of "seed" points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed (such as specific ranges of gray level or color). When a priori information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations shows clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds.

The selection of similarity criteria depends not only on the problem under consideration, but also on the type of image data available. For example, the analysis of land-use satellite imagery depends heavily on the use of color. This problem would be significantly more difficult, or even impossible, to handle without the inherent information available in color images. When the images are monochrome, region analysis must be carried out with a set of descriptors based on gray levels and spatial properties (such as moments or texture).

Basically, growing a region should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as gray level, texture, and color, are local in nature and do not take into account the "history" of region growth. Additional criteria that increase the power of a region growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far (such as a comparison of the gray level of a candidate and the average gray level of the grown region), and the shape

of the region being grown. The use of these types of descriptors is based on the assumption that a model of expected results is at least partially available.

SPLITTING AND MERGING

- The quad tree segmentation is a simple but still powerful tool for image decomposition.
- In this technique, an image is divided into various sub images of disjoint regions and then merge the connected regions together. The procedure just discussed grows regions from a set of seed points. An alternative is to subdivide an image initially into a set of arbitrary, disjointed regions and then merge and/or split the regions in an attempt to satisfy the conditions. A split and merge algorithm that iteratively works toward satisfying these constraints is developed.

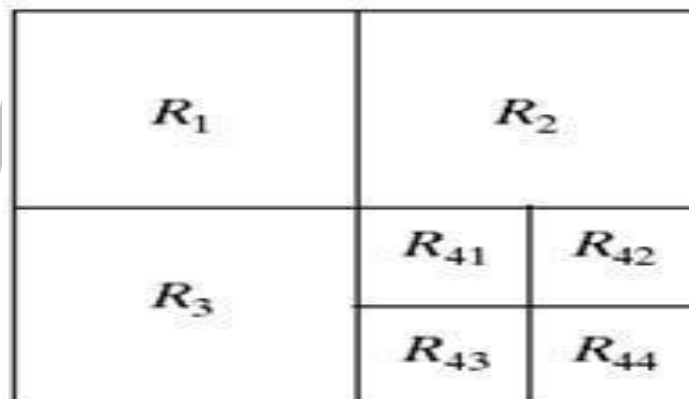
Let R represent the entire image region and select a predicate P . One approach for segmenting R is to subdivide it successively into smaller and smaller quadrant regions so that, for any region R_i , $P(R_i) = \text{TRUE}$. We start with the entire region. If

$P(R) = \text{FALSE}$, we divide the image into quadrants. If P is FALSE for any quadrant, we subdivide that quadrant into sub quadrants, and so on. This particular splitting technique has a convenient representation in the form of a so-called quad tree (that is, a tree in which nodes have exactly four descendants) Note that the root of the tree corresponds to the entire image and that each node corresponds to a subdivision. In this case, only R_4 was sub divided further.

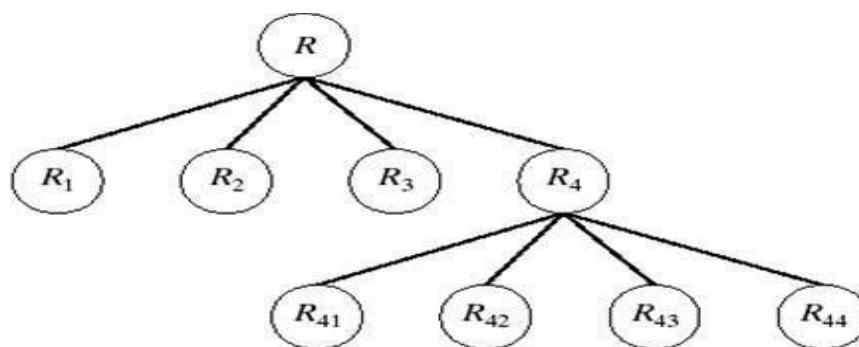
Steps involved in splitting and merging

- Split into 4 disjoint quadrants any region R_i for which $P(R_i) = \text{FALSE}$.
- Merge any adjacent regions R_j and R_k for which $P(R_j \cup R_k) = \text{TRUE}$.
- Stop when no further merging or splitting is possible.

The following shows the partitioned image



(A)



(B)

Fig:4.3.1 (a) Partitioned image. (b) Corresponding quadtree. Represents the entire image region.

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010-Page- 767)

An **example of quad tree-based** split and merge is shown below

In the first phase the image is decomposed into four sub blocks (a), which are then further divided, except the top leftmost block (b).

- At the third phase the splitting phase is completed(c).
- At the merging phase the neighboring blocks that are uniform and have the same color are merged together.

The segmentation results to only two regions (d), which is ideal in this favorable example.

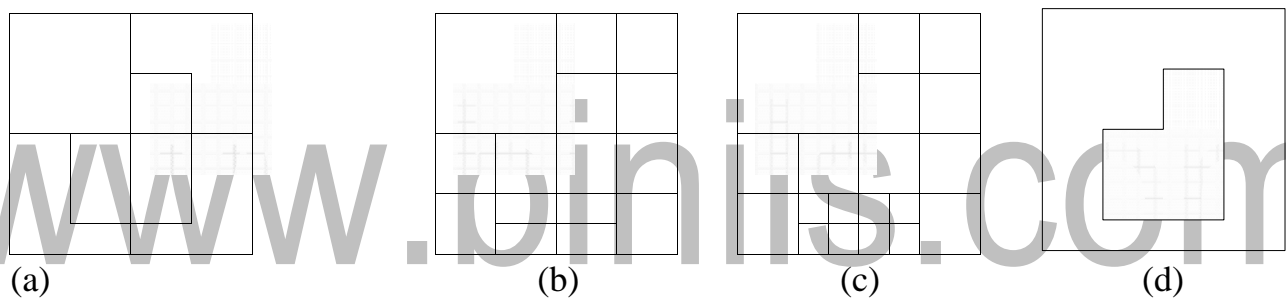


Figure :4.3.2 Example of quad tree based splitting and merging.

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010.-Page-367)

If only splitting were used, the final partition likely would contain adjacent regions with identical properties. This drawback may be remedied by allowing merging, as well as splitting. Satisfying the constraints, requires merging only adjacent regions whose Several variations of the preceding basic theme are possible. For example, one possibility is to split the image initially into a set of blocks. Further splitting is carried out as described previously, but merging is initially limited to groups of four blocks that are descendants in the quadtree representation and that satisfy the predicate P. When no further mergings of this type are possible, the procedure is terminated by one final merging of regions satisfying step 2. At this point, the merged regions may be of different sizes. The principal advantage of this approach is that it uses the same quadtree for splitting and merging, until the final merging step.

www.binils.com

Segmentation By Morphological Watersheds

The aim of Segmentation is to separate regions wrt brightness, color, reflectivity, texture, etc. Segmentation based on three principal concepts: (a) detection of discontinuities, (b) thresholding, and (c) region processing.

1. Basic Concepts

The concept of watersheds is based on visualizing an image in three dimensions: two spatial coordinates versus gray levels. In such a "topographic" interpretation, we consider three types of points:

a. Points belonging to a regional minimum
b. Catchment basin / watershed of a regional minimum

Points at which a drop of water will certainly fall to a single minimum

c. Divide lines / Watershed lines
Points at which a drop of water will be equally likely to fall to more than one minimum. Crest lines on the topographic surface

This technique is to identify all the third type of points for segmentation

1. Piercing holes in each regional minimum of I
2. The 3D topography is flooded from below gradually
3. When the rising water in distinct catchment basins is about to merge, a dam is built to prevent the merging
4. The dam boundaries correspond to the watershed lines to be extracted by a Watershed segmentation algorithm

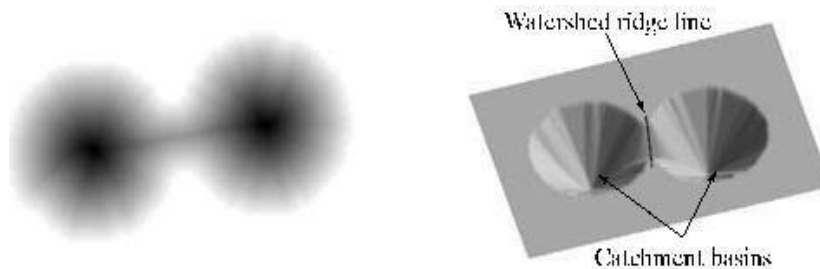


Fig4.4.1: Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final Watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/ Ecole des Mines de Paris.)

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010.-Page-770)

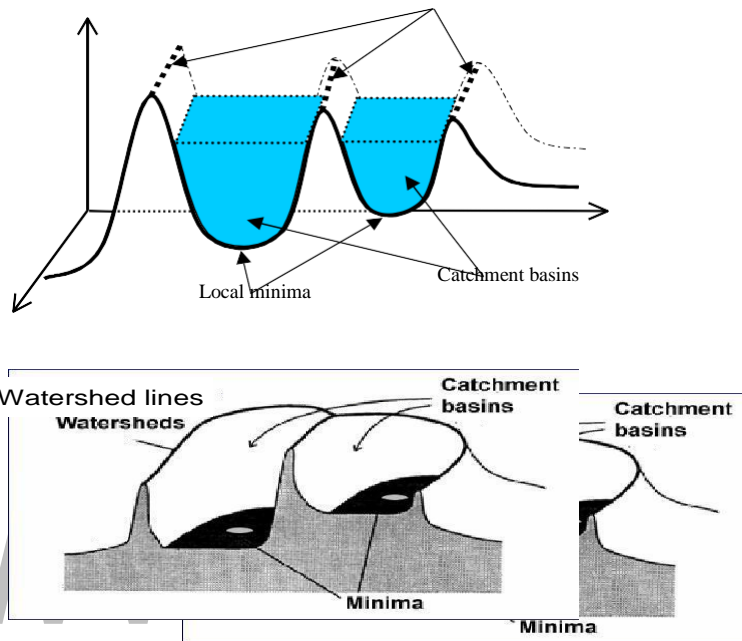


Fig4.4.2: (a) Original image. (b) Topographic view. (c)– (d) Two stages of flooding.

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010.-Page-770.)

The principal objective of segmentation algorithms based on these concepts is to find the watershed lines. The basic idea is simple: Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate. When the rising water in distinct catchment basins is about to merge, a dam is built to prevent Merging.

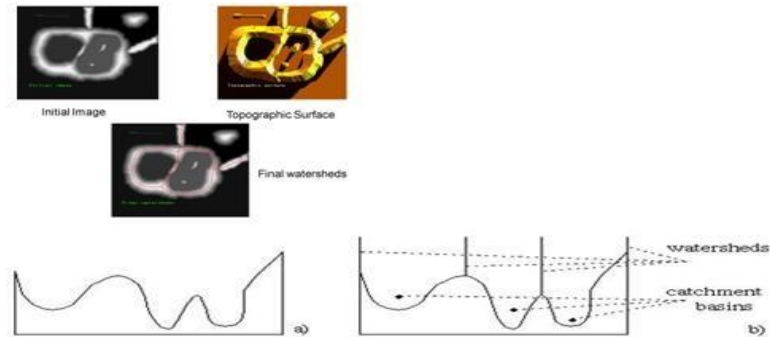
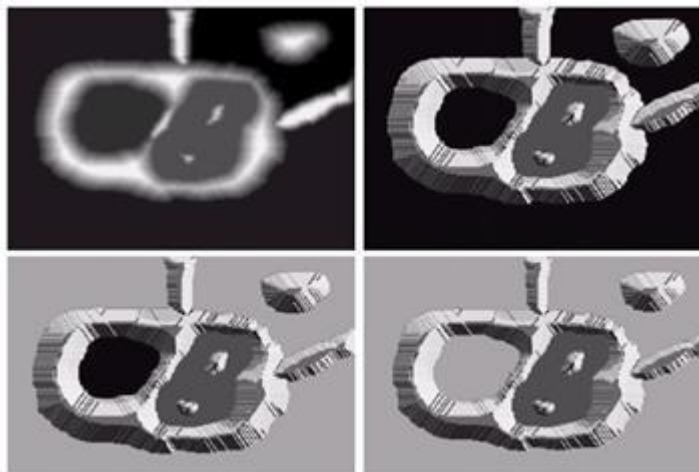


Fig4.4.3: One dimensional example of Watershed Segmentation

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010 --Page-771)

The flooding will eventually reach a stage when only the tops of the dams are visible above the water line. These dam boundaries correspond to the divide lines of the watersheds.

Therefore, they are the (continuous) boundaries extracted by a watershed segmentation algorithm. These ideas can be explained further with the aid of Fig.



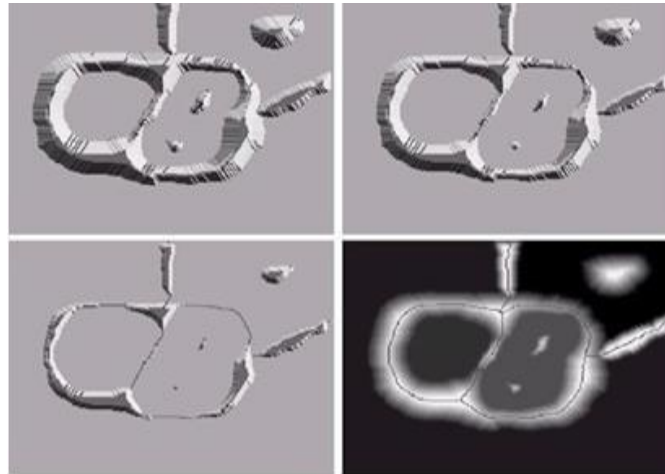


Figure: 4.4.4 shows a simple gray-scale image and Fig. (b) is a topographic view, in which the height of the "mountains" is proportional to gray-level values in the input image.

(Source: Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Third Edition, 2010-Page 773)

For ease of interpretation, the backsides of structures are shaded. This is not to be confused with gray-level values; only the general topography of the three-dimensional representation is of interest. In order to prevent the rising water from spilling out through the edges of the structure, we imagine the perimeter of the entire topography (image) being enclosed by dams of height greater than the highest possible mountain, whose value is determined by the highest possible gray-level value in the input image.

Suppose that a *hole* is punched in each regional minimum [shown as dark areas in Fig. (b)] and that the entire topography is flooded from below by letting water rise through the holes *at* a uniform rate. Figure (c) shows the first stage of flooding, where the "water" shown in light gray, has covered only areas that correspond to the very dark background in the image. In Figs. (d) and (e) we see that the water now has risen into the first and second catchment basins, respectively. As the water continues to rise, it will eventually overflow

from one catchment basin into another. The first indication of this is shown in (f). Here, water from the left basin actually overflowed into the basin on the right and a short "dam" (consisting of single pixels) was built to prevent water from merging at that level of flooding (the details of dam building are discussed in the following section). The effect is more pronounced as water continues to rise, as shown in Fig. (g). This figure shows a longer dam between the two catchment basins and another dam in the top part of the right basin. The latter dam was built to prevent merging of water from that basin with water from areas corresponding to the background. This process is continued until the maximum level of flooding (corresponding to the highest gray-level value in the image) is reached. The final dams correspond to the watershed lines, which are the desired segmentation result. The result for this example is shown in Fig. (h) as a dark, one-pixel-thick path superimposed on the original image. Note the important property that the watershed lines form a connected path, thus giving continuous boundaries between regions.

One of the principal applications of watershed segmentation is in the extraction of nearly uniform (blob like) objects from the background. Regions characterized by small variations in gray levels have small gradient values. Thus, in practice, we often see watershed segmentation applied to the gradient of an image, rather than to the image itself. In this formulation, the regional minima of catchment basins correlate nicely with the small value of the gradient corresponding to the objects of interest. From one catchment basin into another. The first indication of this is shown in (f). Here, water from the left basin actually overflowed into the basin on the right and a short "dam"

(consisting of single pixels) was built to prevent water from merging at that level of flooding (the details of dam building are discussed in the following section), The effect is more pronounced as water continues to rise, as shown in Fig. (g). This figure shows a longer dam between the two catchment basins and another dam in the top part of the right basin. The latter dam was built to prevent merging of water from that basin with water from areas corresponding to the background. This process is continued until the maximum level of flooding (corresponding to the highest gray-level value in the image) is reached. The final dams correspond to the watershed lines, which are the desired segmentation result. The result for this example is shown in Fig. (h) as a dark, one-pixel-thick path superimposed on the original image. Note the important property that the watershed lines form a connected path, thus giving continuous boundaries between regions.

One of the principal applications of watershed segmentation is in the extraction of nearly uniform (blob like) objects from the background. Regions characterized by small variations in gray levels have small gradient values. Thus, in practice, we often see watershed segmentation applied to the gradient of an image, rather than to the image itself. In this formulation, the regional minima of catchment basins correlate nicely with the small value of the gradient corresponding to the objects of interest.