Segmentation using Similarity Based Approach

- Similarity based segmentation
- Thresholding

  1. Global thresholding
  2. Dynamic or adaptive threshold
  3. Optimal threshold
  4. Local thresholding

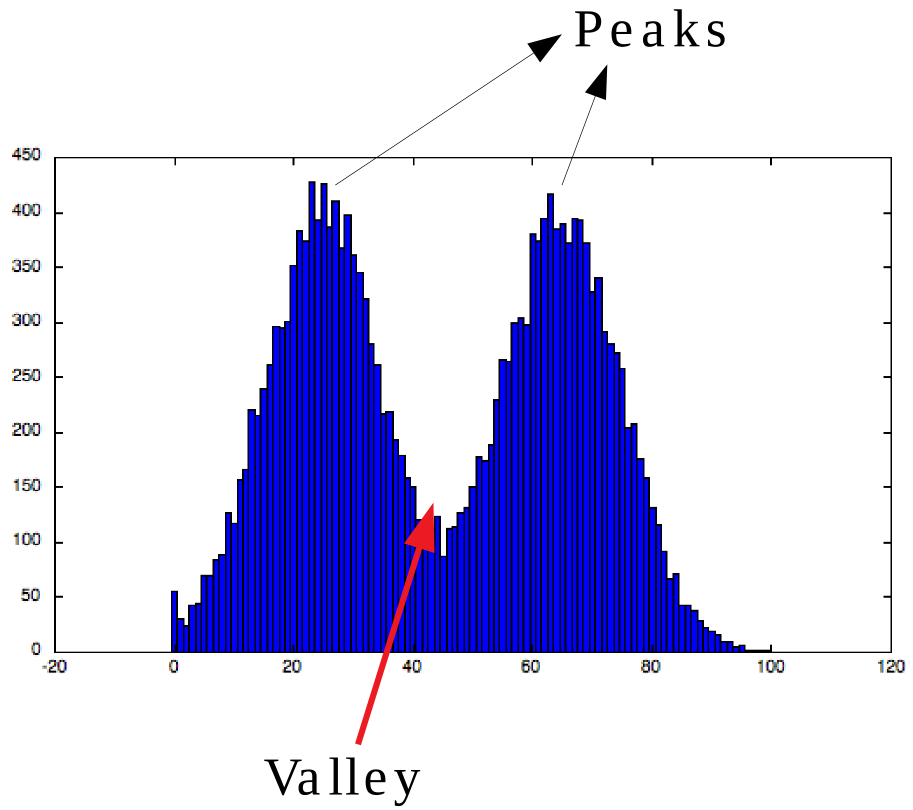
- Region growing technique
- Region splitting and merging technique

Only local and global are to be reviewed

Similarity based Segmentation

Thresholding

- Suppose an image f(x,y) is having a dark object against bright background
- Such image generate bimodal histogram



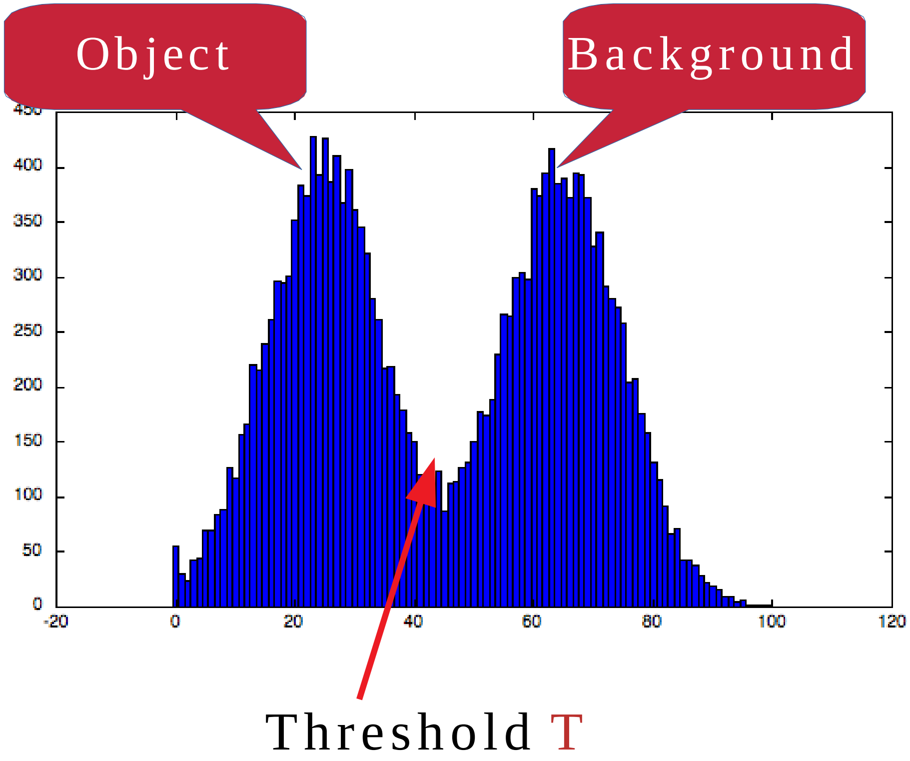
Similarity based Segmentation

Thresholding

- Suppose an image f(x,y) is having a dark object against bright background
- Such image generate bimodal histogram

$$f(x,y) < T \quad \text{implies} \quad object$$

$$f(x,y) \geq T \quad \text{implies} \quad background$$



Similarity based Segmentation

- Thresholding example

Input image

Segmented output image

Similarity based Segmentation

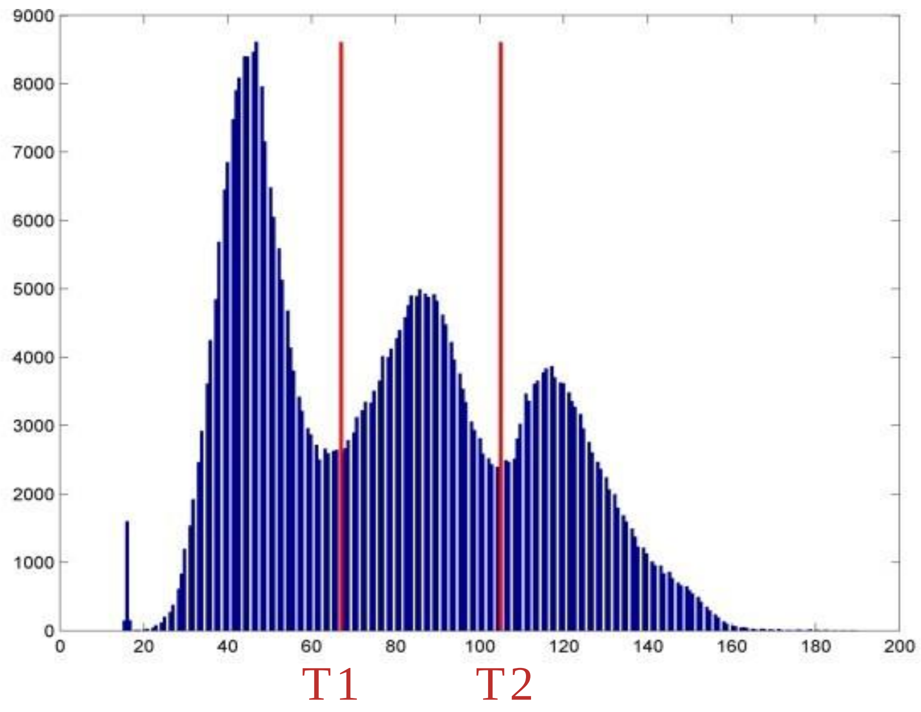
Thresholding

Trimodal histogram

$$f(x,y) > T_2 \quad implies \quad object\,2$$

$$T_1 < f(x,y) \le T_2 \quad implies \quad object\,1$$

$$f(x,y) < T_1 \quad implies \quad background$$



Similarity based Segmentation

- Selection of threshold value
- Thresholding function which test the image against threshold value

$$T = T[x, y, p(x, y), f(x, y)]$$

(x,y) = coordinates of the pixels

f(x,y) = intensity value of the pixels

p(x,y) = local property in neighborhood
Centered at (x,y)

Similarity based Segmentation

- Selection of threshold value
- Thresholding function which test the image against threshold value

$$T = T[x, y, p(x, y), f(x, y)]$$

Depending on combination of these three values Thresholding can be
1. Local
2. Global
3. Adaptive
4. Optimal

(x,y) = coordinates of the pixels

f(x,y) = intensity value of the pixels

p(x,y) = local property in neighborhood Centered at (x,y)

Similarity based Segmentation

- If T[f(x,y)] then it is global thresholding
- If T[f(x,y), p(x,y)] then it is local thresholding
- If T[(x,y), f(x,y), p(x,y)] then it is Adaptive thresholding

$$g(x,y) = 0 \ \ if \ \ f(x,y) > T$$ Object pixel

$$g(x,y) = 1 \ \ if \ \ f(x,y) \leq T$$ Background pixel

Similarity based Segmentation

- How to find threshold value?
- Every time looking at the histogram and deciding is not feasible
- Need some automated process for threshold selection

Similarity based Segmentation

- Automatic global thresholding:

1. Initialize value of threshold T
2. Perform segmentation using threshold value T to get two regions G1 and G2

Pixel intensity values from group G1 and G2 are similar but two groups different

Similarity based Segmentation

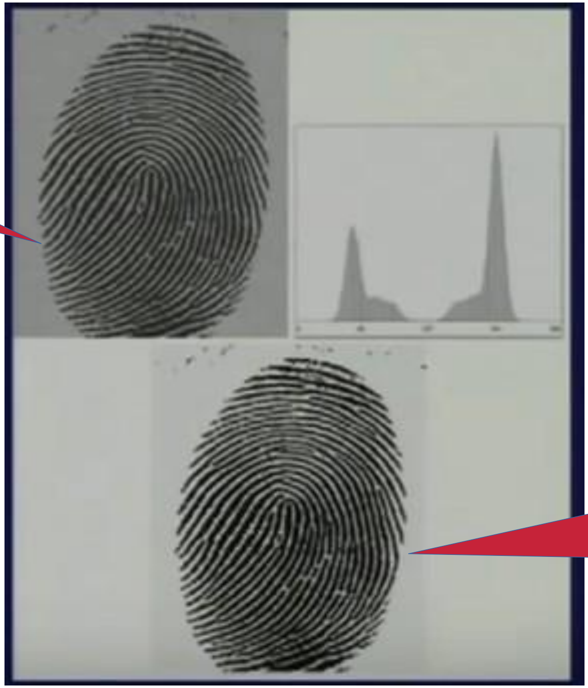
- Automatic global thresholding:

1. Initialize value of threshold T
2. Perform segmentation using threshold value T to get two regions G1 and G2
3. Compute mean M1 and M2 using pixel intensity values of G1 and G2
4. New threshold value T = (M1 + M2)/2
5. If (|Ti – T(i+1)| <= T') then STOP
6. else goto step 2

T' is some tolarance value

Similarity based Segmentation

- Automatic global thresholding example:



Similarity based Segmentation

- Automatic global thresholding:

  Gobal thresholding gives very good results if intensity of illumination is uniform

- However, getting a global threshold value is difficult

Similarity based Segmentation

Automatic global thresholding for poor illuminated image

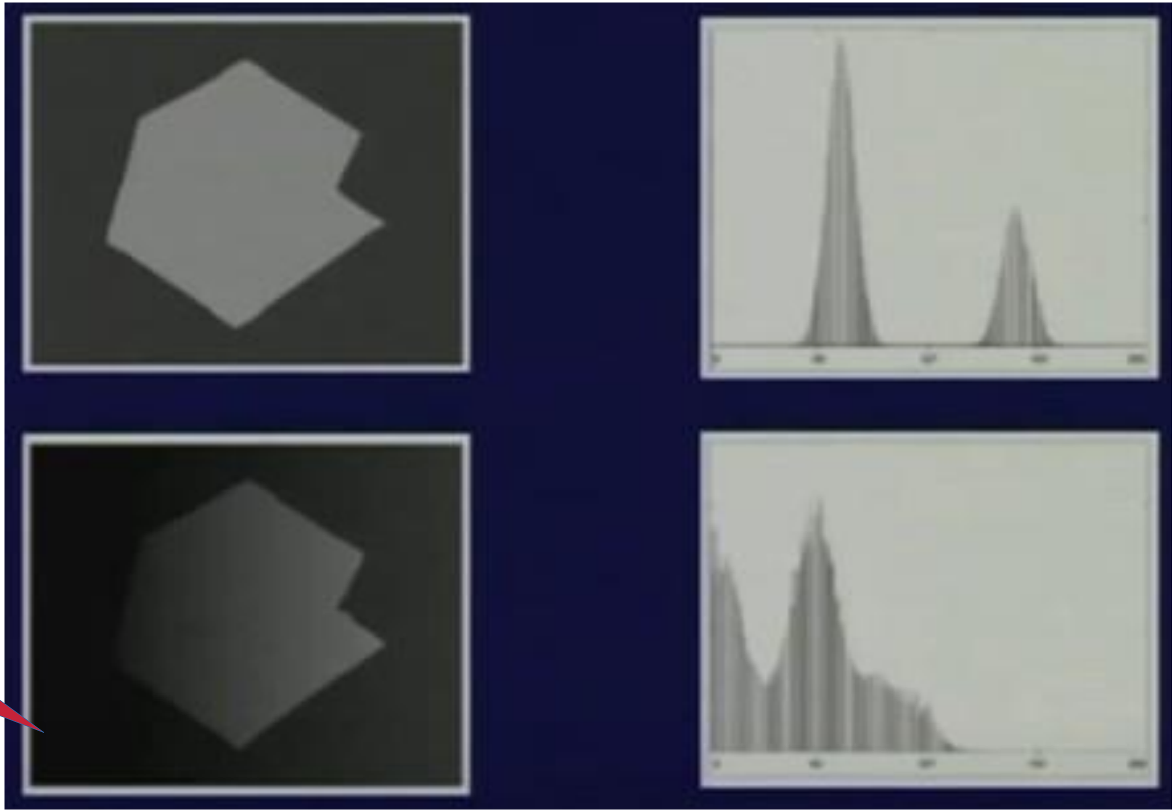


Image in non uniform illumination source

Similarity based Segmentation

Automatic global thresholding for poor illuminated image

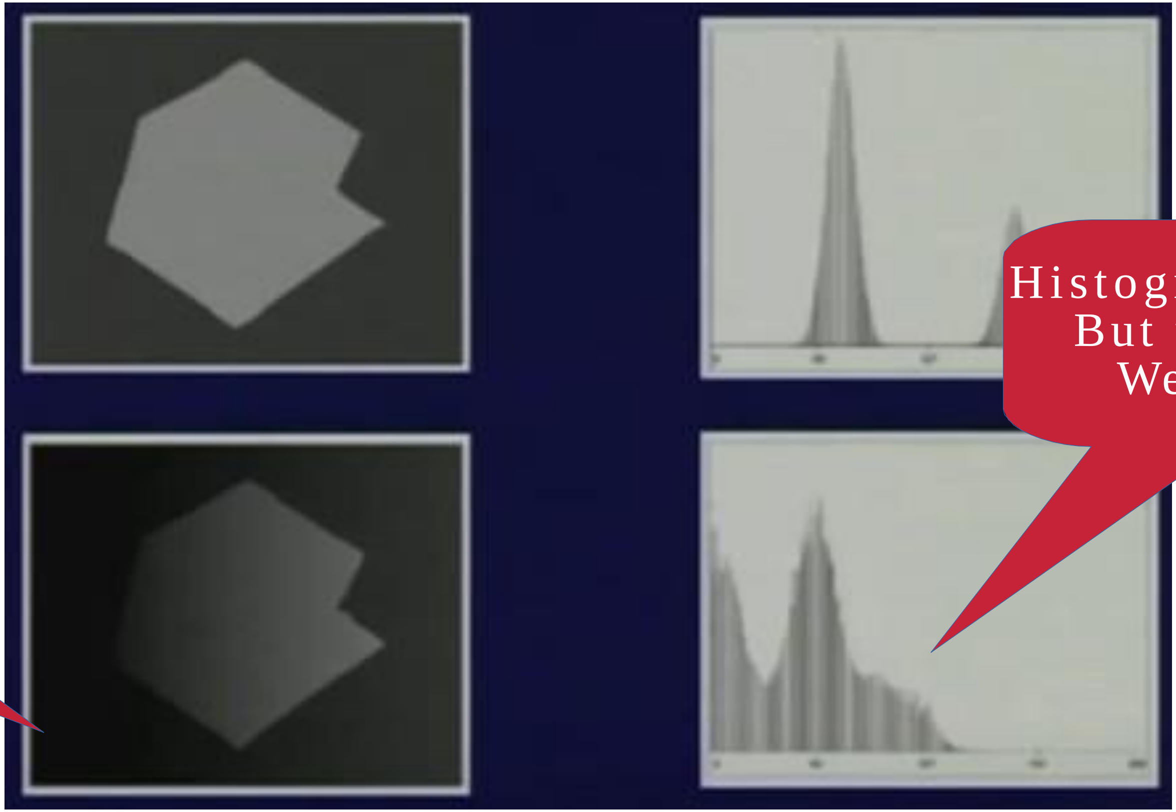
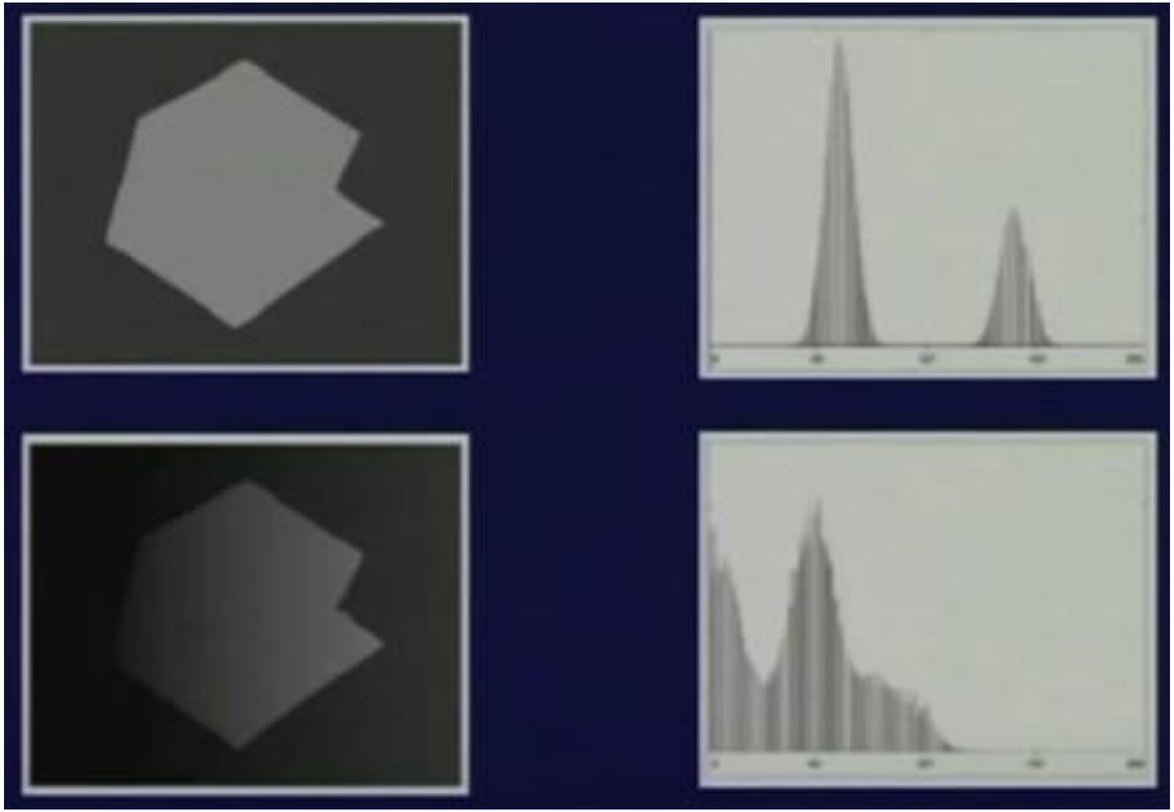


Image in non uniform illumination source

Histogram is bimodal But valley is not Well defined

Similarity based Segmentation

Automatic global thresholding for poor illuminated image



Automatic global thresholding
is likely to fail!!!

Similarity based Segmentation

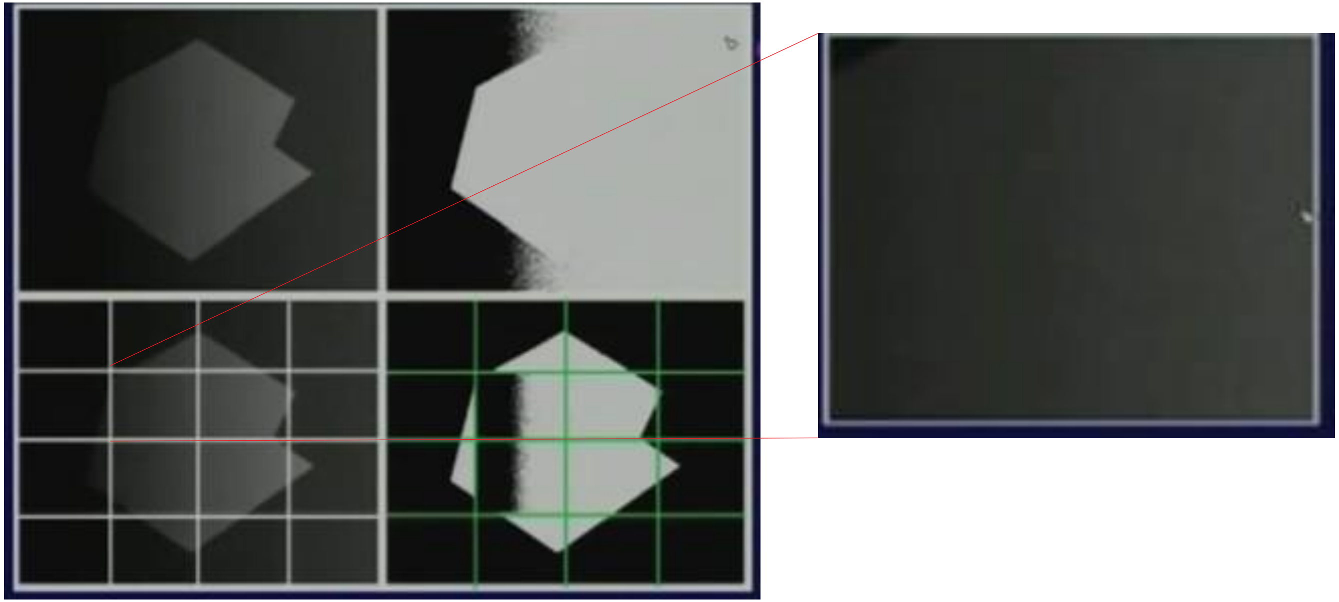
- Solution for poor illumination problem of global thresholding:

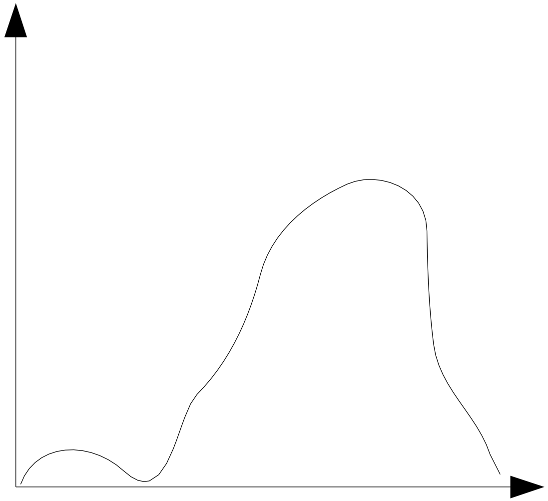
- Divide the image into number of sub-images so that illumination can be uniform
- Find the global threshold value for sub-images
- Union of all threshold values gives the final threshold

This is called as an adaptive thresholding as the threshold value is depending on the location in image

Similarity based Segmentation

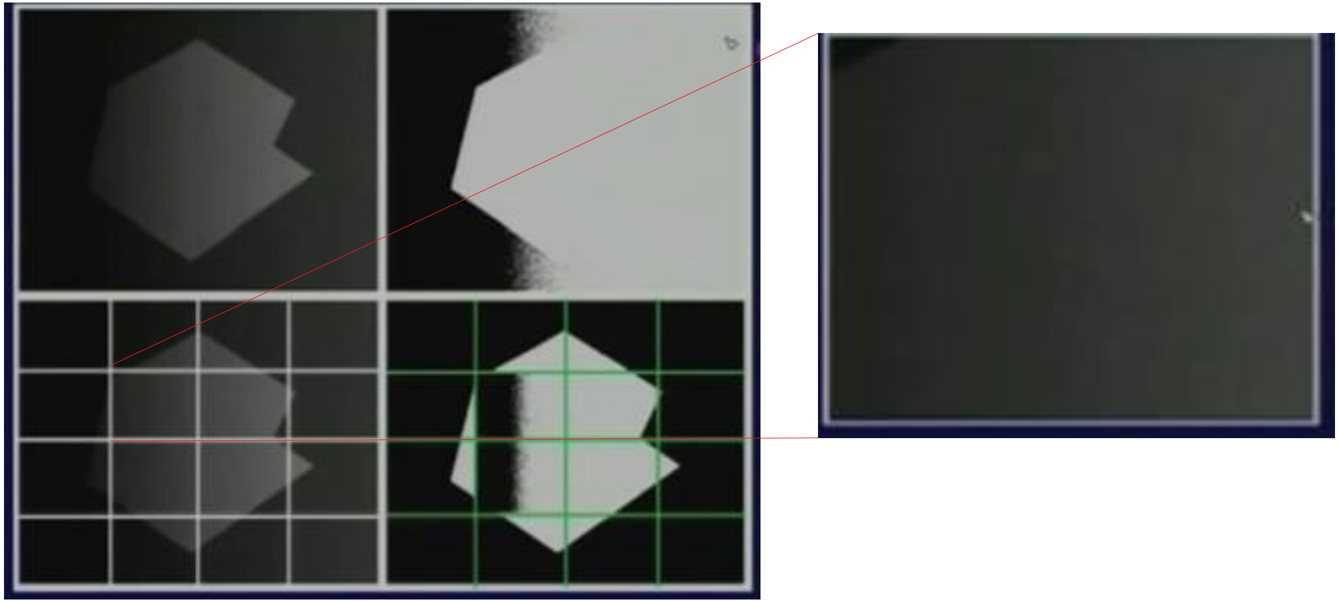
- Solution for poor illumination problem of global thresholding

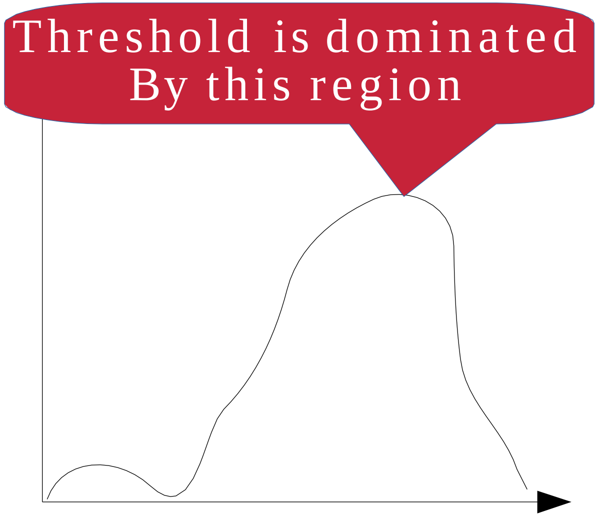




Similarity based Segmentation

- Solution for poor illumination problem of global thresholding





Similarity based Segmentation

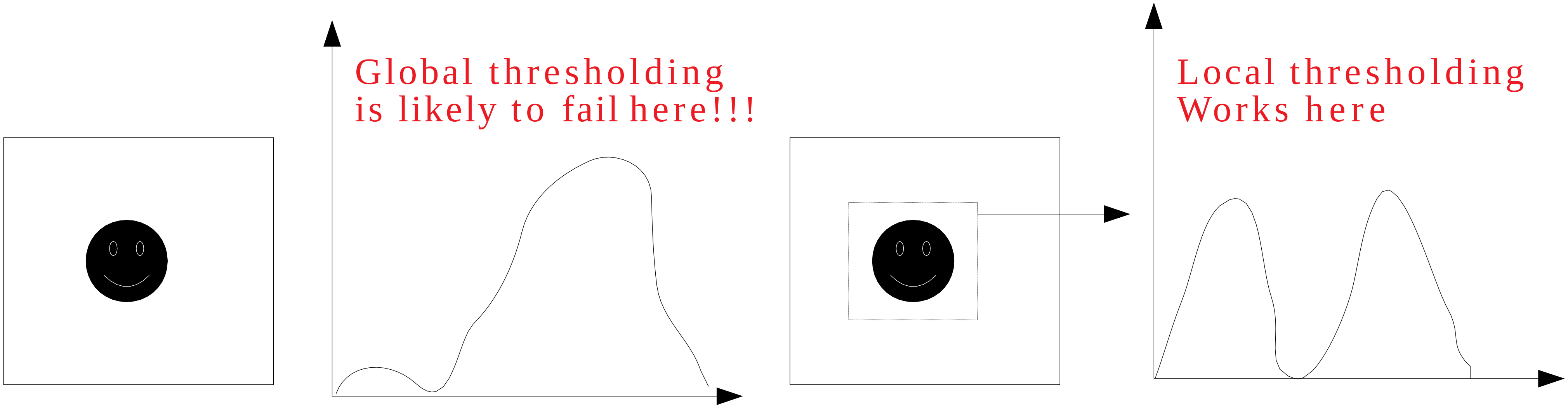
- Local Thresholding

Solution:
Consider local image and apply the thresholding

This converts the histogram into bimodal
and equally distributed in object and background
region

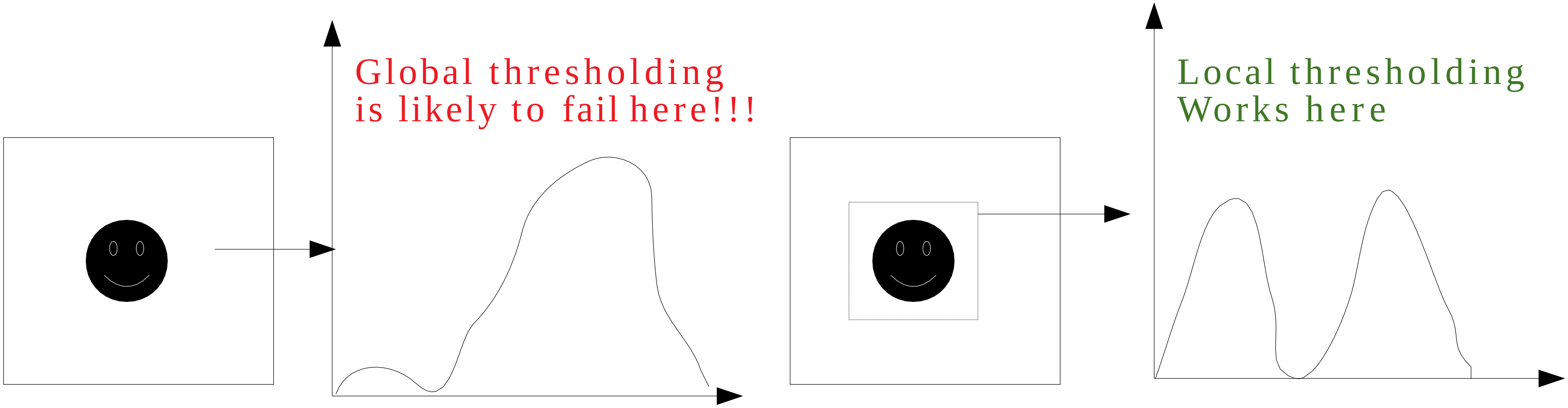
Similarity based Segmentation

- Local Thresholding



Similarity based Segmentation

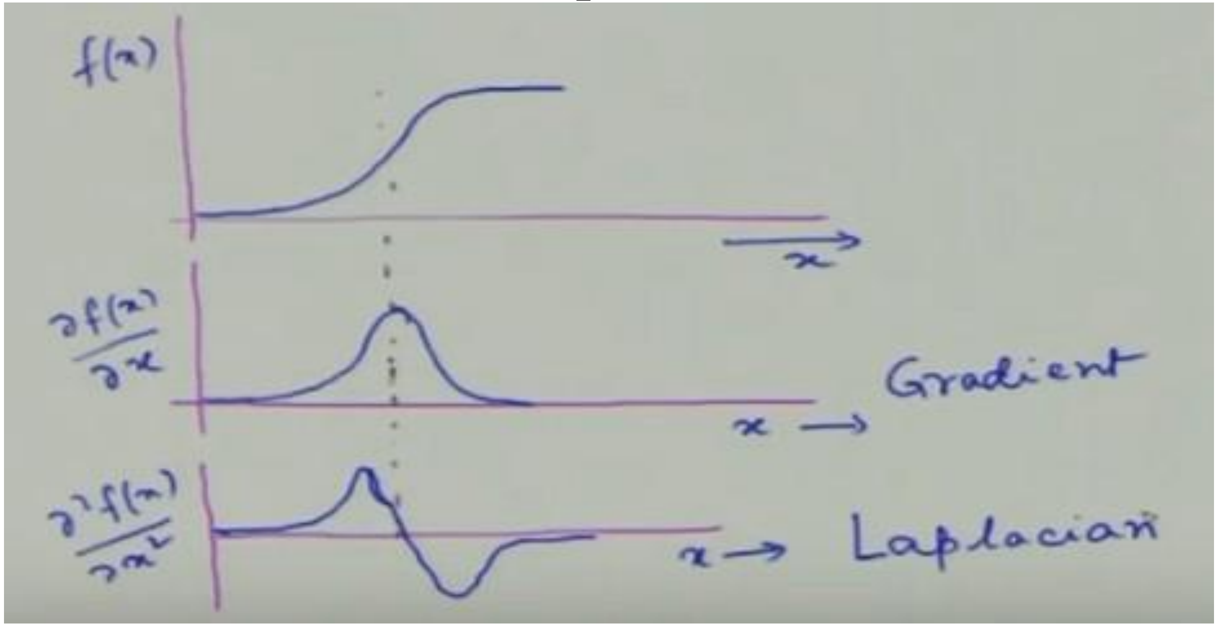
- Local Thresholding



BUT HOW TO DETECT LOCAL BOUNDARY
BETWEEN OBJECT AND BACKGROUND?

Similarity based Segmentation

- Local Thresholding – Boundary detection using Gradient and Laplacian

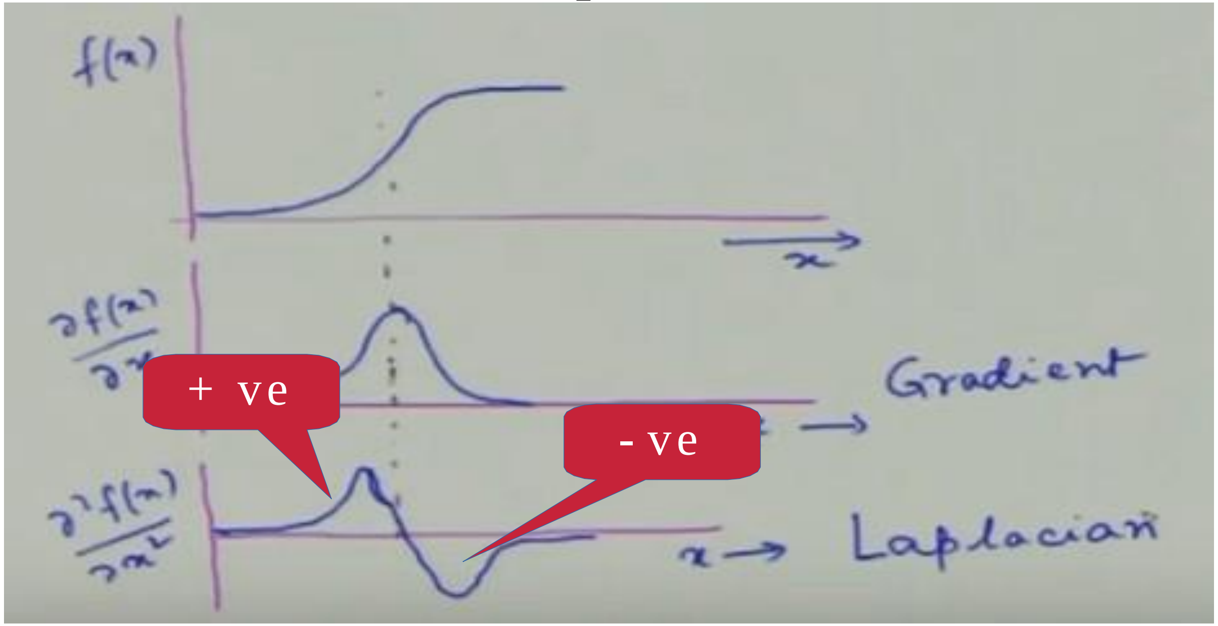


Gradient gives position of The edge whereas

Laplacian Determines whether point Lies on darker side or brighter Side of the edge

Similarity based Segmentation

- Local Thresholding – Boundary detection using Gradient and Laplacian



Gradient gives position of
The edge whereas

Laplacian Determines
whether point Lies on
darker side or brighter
Side of the edge

Similarity based Segmentation

- Local Thresholding – Boundary detection using Gradient and Laplacian

$$\text{using three properties } f(x,y) \;\; \nabla f(x) \;\; \nabla^2 f(x,y)$$

$$s(x,y) = 0 \;\; \text{if} \;\; \nabla f(x,y) < T$$ Not belong to boundary

$$= +ve \;\; \text{if} \;\; \nabla f(x,y) \geq T \quad \nabla^2 f(x,y) \geq 0$$ Belongs to Object

$$= -ve \;\; \text{if} \;\; \nabla f(x,y) \geq T \quad \nabla^2 f(x,y) < 0$$ Belongs to Background

Similarity based Segmentation

Local Thresholding –
Boundary detection using Gradient and Laplacian

This image is used to
Findout object region
And background image

Similarity based Segmentation

Local Thresholding – Boundary detection using Gradient and Laplacian

Scan each row of the image from left to right

- (- , +) indicates transition from background to object
- (+, -) indicates transition from object to background

- (......***)(-, +)(0 or +)(+, -)(***.....)

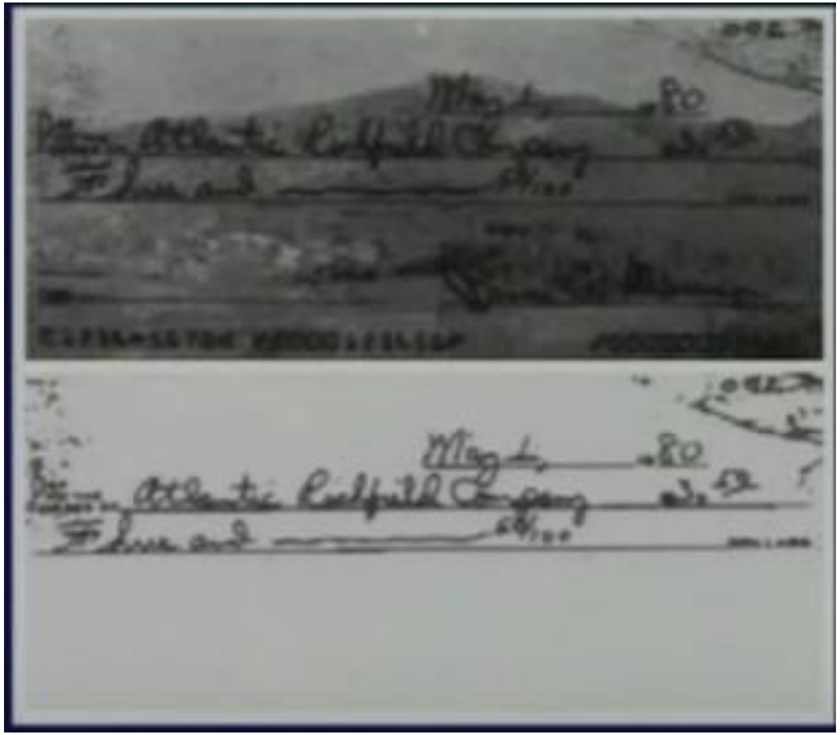
Similarity based Segmentation

Local Thresholding –
Boundary detection using Gradient and Laplacian

(……***)(-, +)(0 or +)(+, -)(***…..)

Object

Any combination of
-, + or 0



Region-Based Segmentation

- Edges and thresholds sometimes do not give good results for segmentation.
- Region-based segmentation is based on the connectivity of similar pixels in a region.
  - Each region must be uniform.
  - Connectivity of the pixels within the region is very important.
- There are two main approaches to region-based segmentation: region growing and region splitting.

Region-Based Segmentation
Basic Formulation

Let R represent the entire image region.

- Segmentation is a process that partitions R into subregions,
- $R_1, R_2, \ldots, R_n$, such that

(a) $\bigcup_{i=1}^{n} R_i = R$

(b) R_i is a connected region, $i = 1, 2, \ldots, n$

(c) $R_i \cap R_j = \phi$ for all i and $j, i \neq j$

(d) $P(R_i) = \text{TRUE}$ for $i = 1, 2, \ldots, n$

(e) $P(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions R_i and R_j

where $P(R_k)$: a logical predicate defined over the points in set R_k

For example: $P(R_k)$=TRUE if all pixels in R_k have the same gray level.

Region Growing

- Thresholding still produces isolated image
- Region growing algorithms work on **principle of similarity**

•**It states that a region is coherent if all the pixels of that region are homogeneous with respect to some characteristics such as colour, intensity, texture, or other statistical properties**

•Thus idea is to pick a pixel inside a region of interest as a starting point (also known as a **seed point**) and allowing it to grow

•**Seed point is compared with its neighbours, and if the properties match , they are merged together**

•This process is repeated till the regions converge to an extent that no further merging is possible

Region Growing Algorithm

- It is a process of grouping the pixels or subregions to get a bigger region present in an image

- **Selection of the initial seed:** Initial seed that represent the ROI should be given typically by the user. Can be chosen automatically. The seeds can be either single or multiple

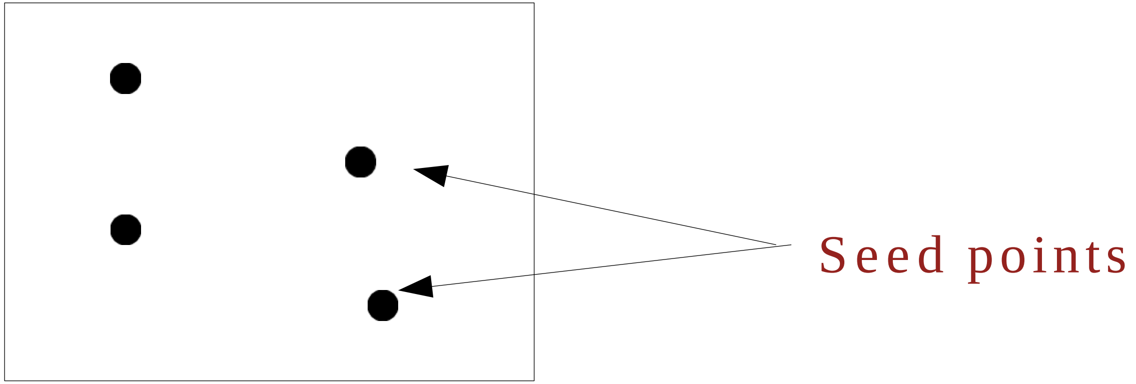
- **Seed growing criteria:** Similarity criterion denotes the minimum difference in the grey levels or the average of the set of pixels. Thus, the initial seed ‘grows’ by adding the neighbours if they share the same properties as the initial seed

- **Terminate process:** If further growing is not possible then terminate region growing process

Similarity based Segmentation

- Region growing segmentation

  Grouping the pixels to make larger subregions such that properties of newly added pixels must be same

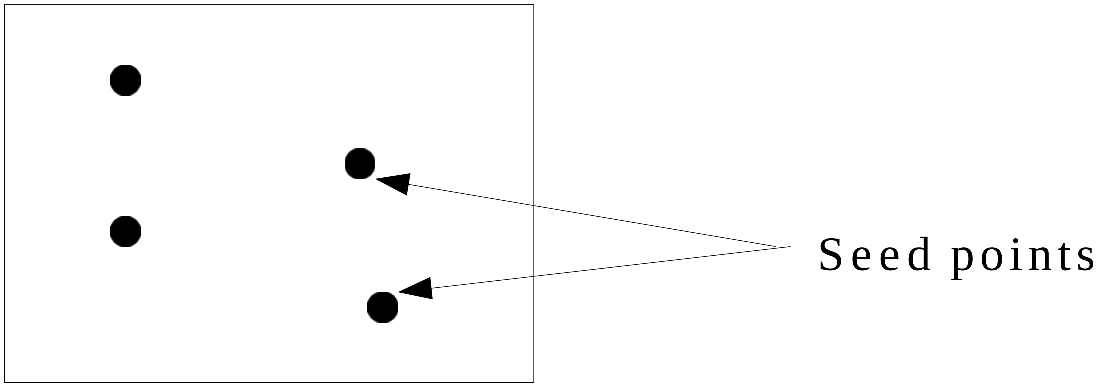


Similarity based Segmentation

- Region growing segmentation

  Make 3X3 neighborhood (4, 8 or m-connected) around seed point and check for the intensity values

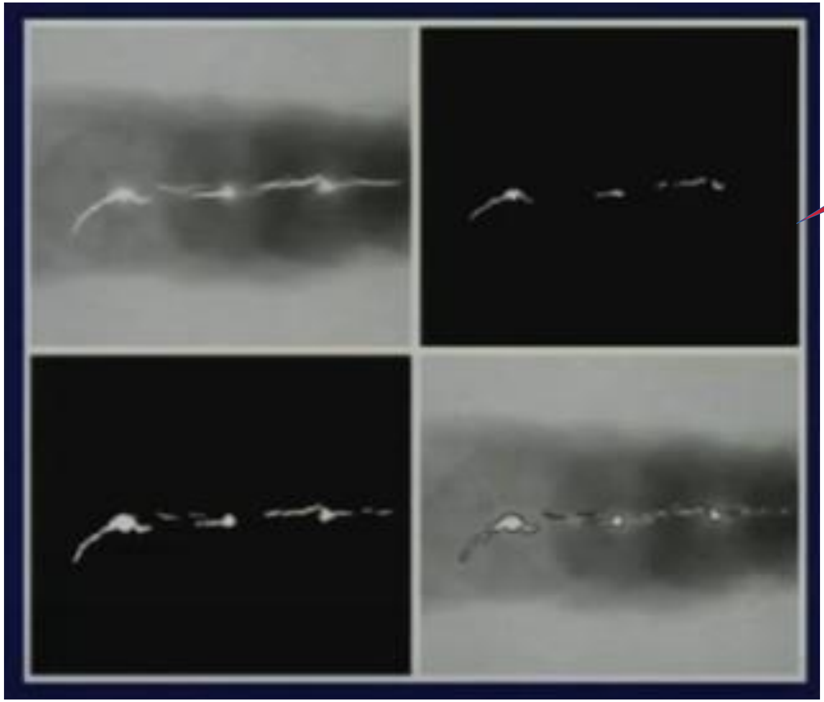
If difference in intensity values is very large then Don't include in the set



Similarity based Segmentation

- Region growing segmentation

X-Ray image of
Welded part



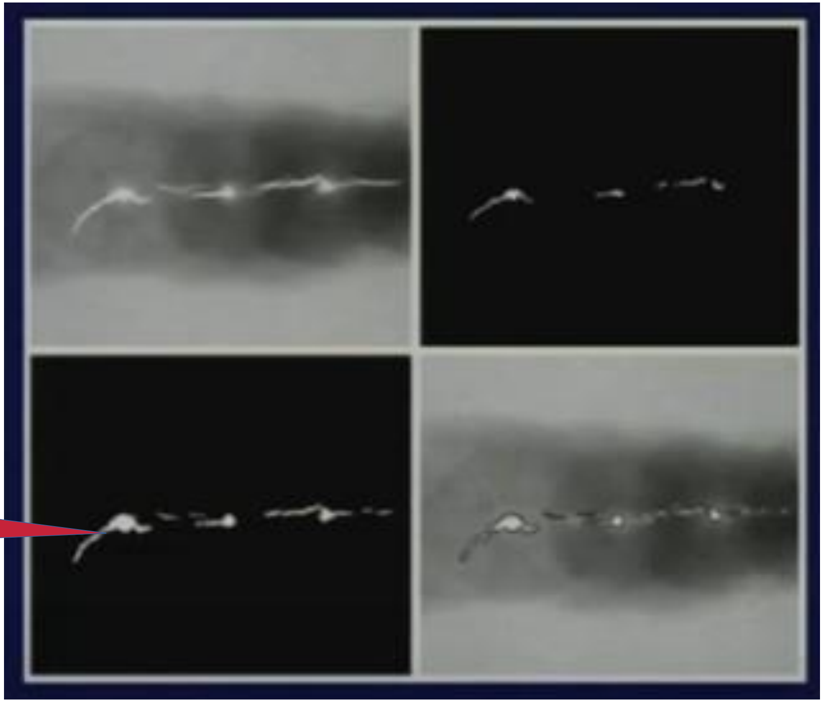
Thresholding
Output

Cracks are indicated
Near 255 value

Similarity based Segmentation

- Region growing segmentation

X-Ray image of
Welded part



Select seed points
With value 255

Perform region
Growing operations
On each seed point
In Original Image

Region Growing Algorithm

- Consider image shown in figure:

1	0	7	8	7
0	1	8	9	8
0	0	7	9	8
0	1	8	8	9
1	2	8	8	9

•Assume seed point indicated by underlines. Let the seed pixels 1 and 9 represent the regions C and D, respectively

•Subtract pixel from seed value

•If the difference is less than or equal to 4 (i.e. T=4), merge the pixel with that region. Otherwise, merge the pixel with the other region.

Split and Merge Algorithm

- Region growing algorithm is slow
- So seed point can be extended to a seed region
- Instead of a single pixel, a node of a Regional adjacency graph (RAG) a region itself is now considered as a starting point.
- The split process can be stated as follows:

1)Segment the image into regions R1, R2,….Rn using a set of thresholds
2)Create RAG. Use a similarity measure and formulate a homogeneity test
3)The homogeneity test is designed based on the similarity criteria such as intensity or any image statistics
4)Repeat step 3 until no further region exits that requires merging

Split and Merge Algorithm

-

2	2	2	2	2
2	8	8	8	1
2	8	8	8	1
2	1	1	1	1
2	1	1	1	1

2	2	2	2	2
2	8	8	8	1
2	8	8	8	1
2	1	1	1	1
2	1	1	1	1

8	8	8
8	8	8

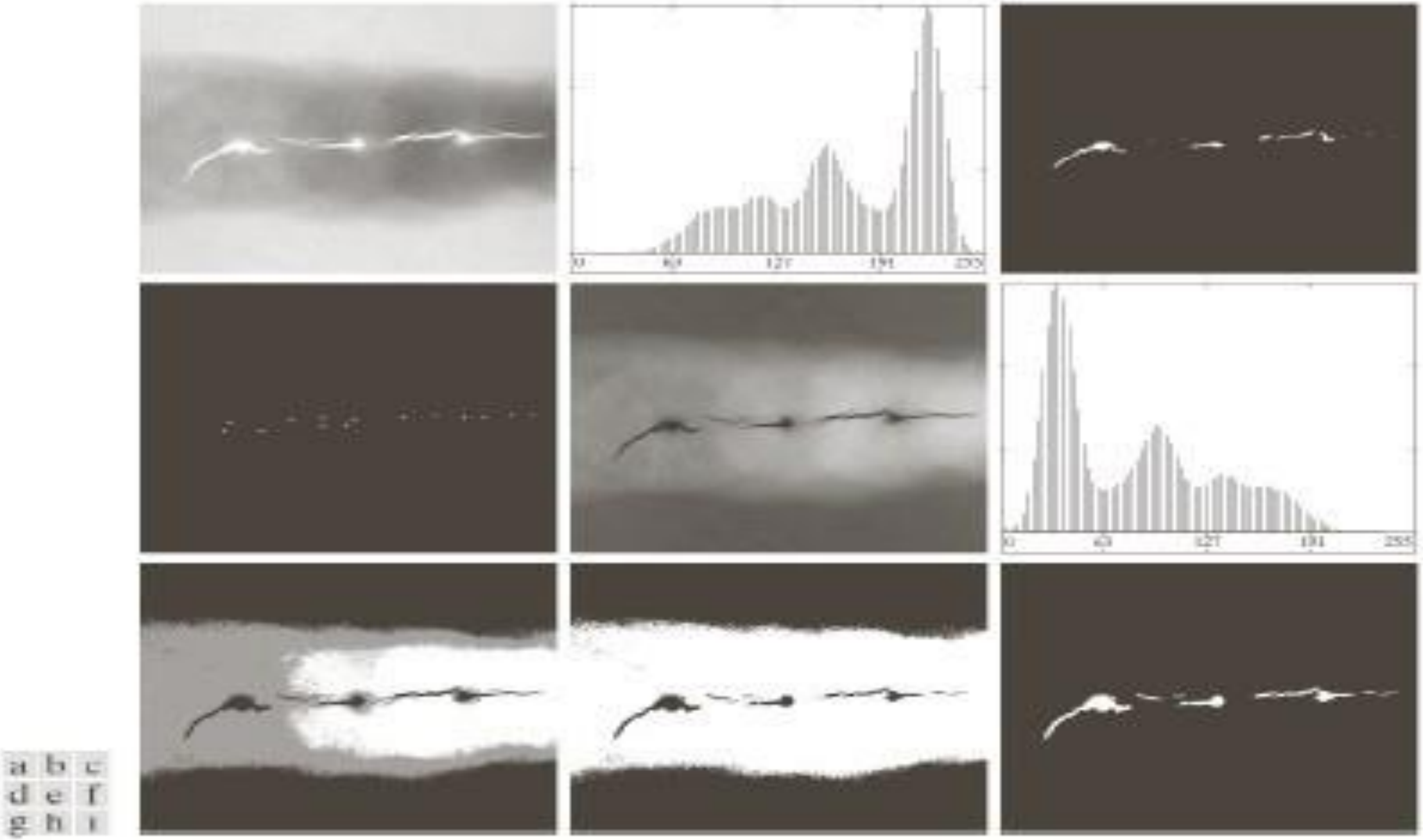
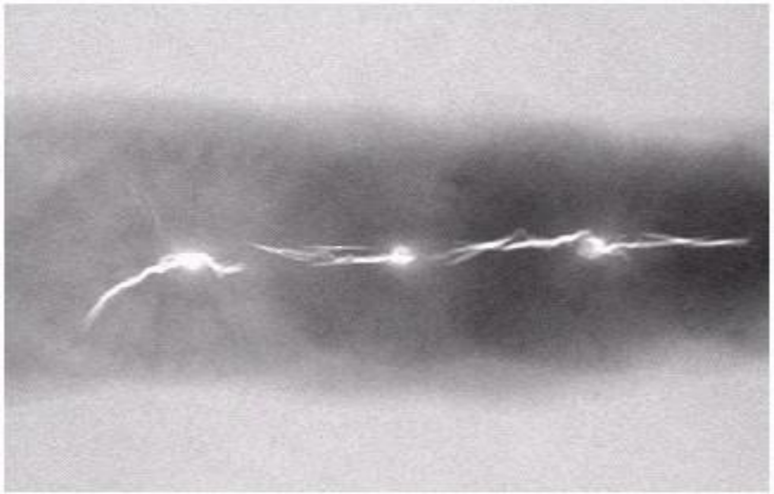
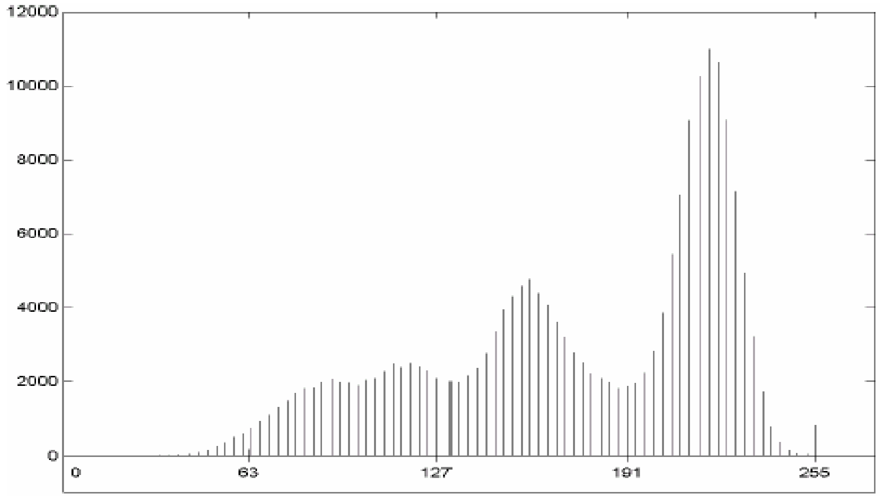


FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference image thresholded with the smallest of the dual thresholds. (i) Segmentation result obtained by region growing. (Original image courtesy of X-TEK Systems, Ltd.)

Region-Based Segmentation
Region Growing

- (a). It is difficult to segment the defects by thresholding methods. (Applying region growing methods are better in this case.)





Split and Merge using Quadtree

- Entire image is assumed as a single region. Then the homogeneity test is applied. If the conditions are not met, then the regions are split into four quadrants.

- This process is repeated for each quadrant until all the regions meet the required homogeneity criteria. If the regions are too small, then the division process is stopped.

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

Region-Based Segmentation
Region Splitting and Merging

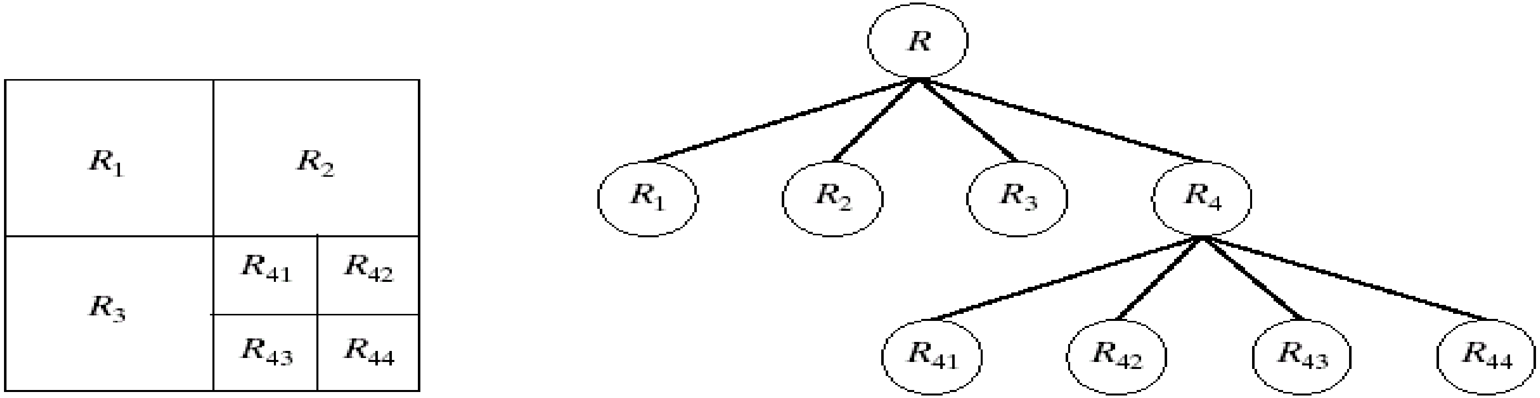
- Region splitting is the opposite of region growing.
  - First there is a large region (possible the entire image).
  - Then a predicate (measurement) is used to determine if the region is uniform.
  - If not, then the method requires that the region be split into two regions.
  - Then each of these two regions is independently tested by the predicate (measurement).

Region-Based Segmentation
Region Splitting

- The main problem with region splitting is determining where to split a region.
- One method to divide a region is to use a quadtree structure.
- Quadtree: a tree in which nodes have exactly four descendants.

a b

FIGURE 10.42
(a) Partitioned image.
(b) Corresponding quadtree.



Region-Based Segmentation
Region Splitting and Merging

The split and merge procedure:

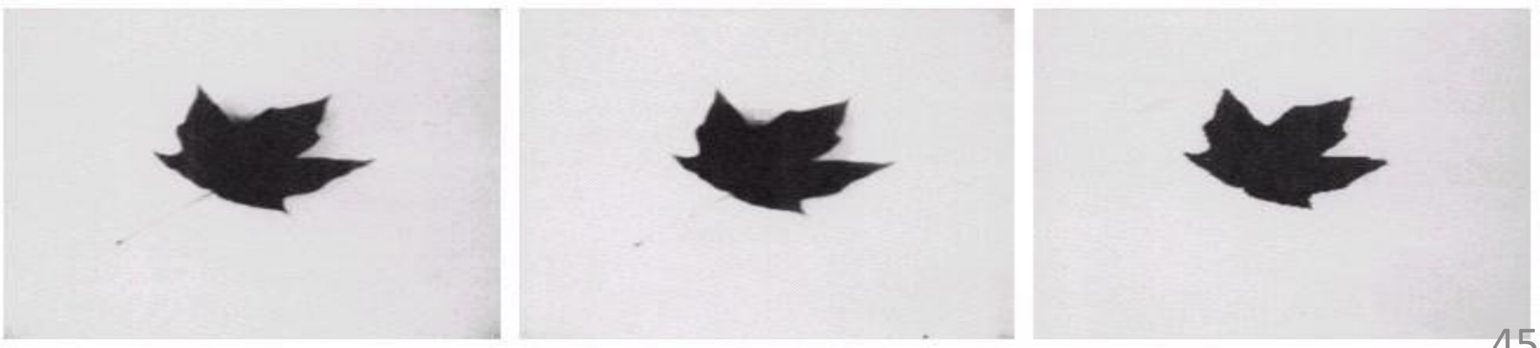
– Split into four disjoint quadrants any region *Ri* which $P(R_i)$ = FALSE.

– Merge any adjacent regions *Rj* and *Rk* for which $P(R_j \cup R_k)$ = TRUE. (the quadtree structure may not be preserved)

– Stop when no further merging or splitting is possible.

a b c

FIGURE 10.43
(a) Original image. (b) Result of split and merge procedure.
(c) Result of thresholding (a).



Technique	Type	Characteristics	Drawbacks
Robert	1st -order derivative	Detect edges and orientation	Thick edges, Sensitive to noise, Inaccurate
Prewitt	1st -order derivative	Detect edges and orientation	Thick edges, Sensitive to noise, Inaccurate
Sobel	1st -order derivative	Detect edges and orientation	Thick edges, Sensitive to noise, Inaccurate
LoG	2nd -order derivative	Thin edges & Rotationally invariant	Unable to find the orientation of edges
Simple thresholding	Global thresholding	Useful in case of bimodal images	Selection of proper threshold
Multiple thresholding	Global thresholding	Used when more than two classes of objects present in the image	Selection of proper threshold & Space and computational complexity is high
Local thresholding	Local thresholding	Useful in situation where background illumination is highly nonuniform	Detect edges and orientation
Otsu	Optimal global thresholding	Applied to the image where boundary (between foreground and background) is not clear & Select threshold optimally based on the characteristics of image data	Detect edges and orientation
Canny edge detection	Combined	Produce perfect contours, Low error rate & Single edge point response	Thin edges & Rotationally invariant

Otsu: iterate through all possible values of threshold and measure the spread of background and foreground pixels