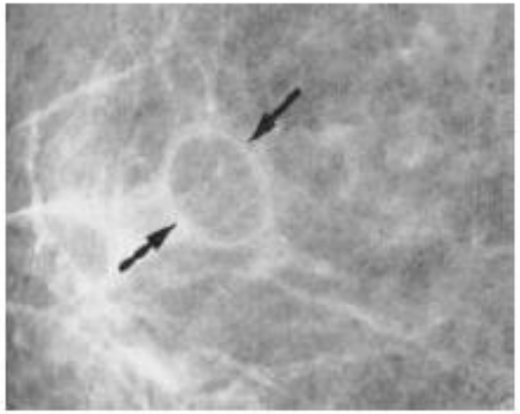
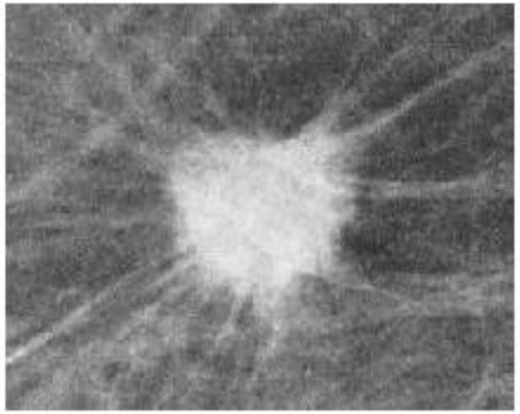
Image Segmentation

Image Segmentation: Context

- Segmentation decomposes the image into parts for further analysis
  - *Example:* background subtraction in human motion analysis
- Once the region of interest is segmented, the representation space can be changed (from image-space to feature space)



Circumscribed (benign)
Lesions in digital mammography



Spiculated (Maignant)
Lesions in digital mammography

Image Segmentation

Divides an image into regions that are **connected** and have some **similarity** within the region and some **difference** between adjacent regions.

The goal is usually to find individual objects in an image.

Image Segmentation

There are fundamentally two kinds of approaches to segmentation: discontinuity and similarity.

- Similarity may be due to pixel intensity, color or texture.
- Differences are sudden changes (discontinuities)
- Sudden changes in intensity along a boundary line is called an edge.

Image Segmentation: Discontinuity

- This approach is to partition an image, based upon abrupt changes in the gray-scale levels.
- The principal approaches are detection of isolated points, lines and edges in an image.
- Boundary estimation using edge detection
  - Boundaries produced are necessarily closed.
  - Computation of regions are based on differences (discontinuities)

Image Segmentation: Similarity

- The principal approaches are based on thresholding, region growing and region splitting / merging.
- Region-based boundaries
  - Closed boundaries
  - Computation of regions is based on similarity.

Classification of Segmentation Techniques

- Intensity-based Segmentation (Thresholding)
- Edge-based Segmentation
- Region-based Segmentation

Image Segmentation: In pictures
Edge Segmentation

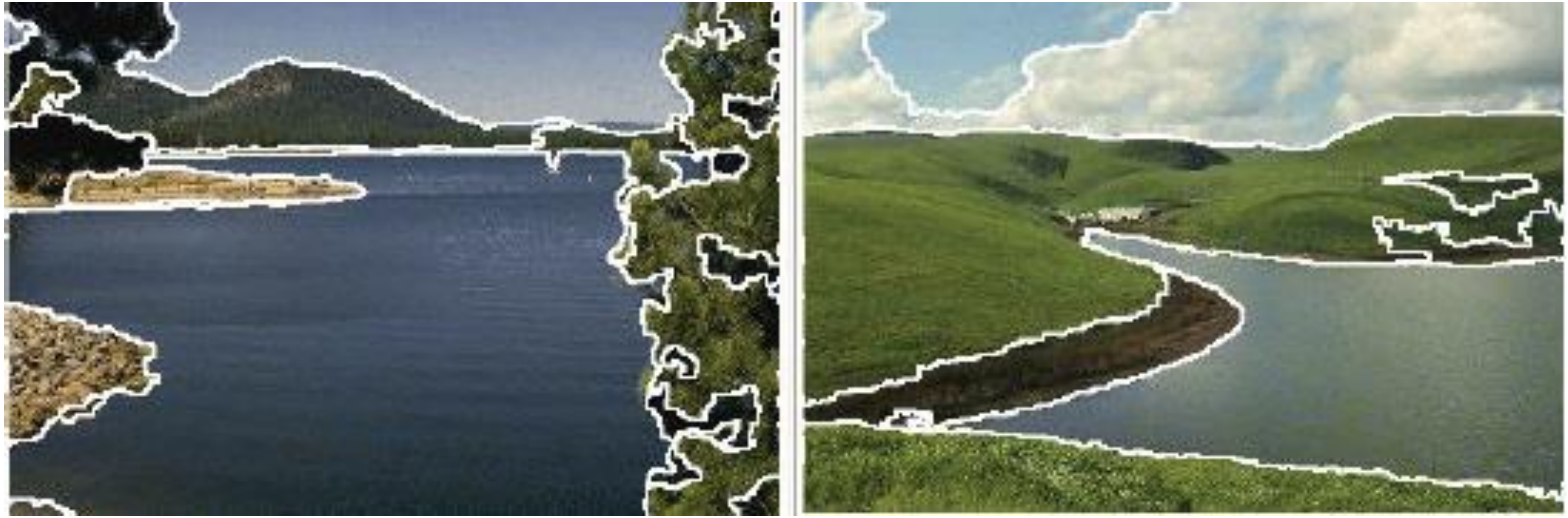


Image Segmentation: In pictures
Region Segmentation

Detection of Discontinuities

- There are three kinds of discontinuities of intensity: points, lines and edges.
- The most common way to look for discontinuities is to scan a small mask over the image. The mask determines which kind of discontinuity to look for.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$

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

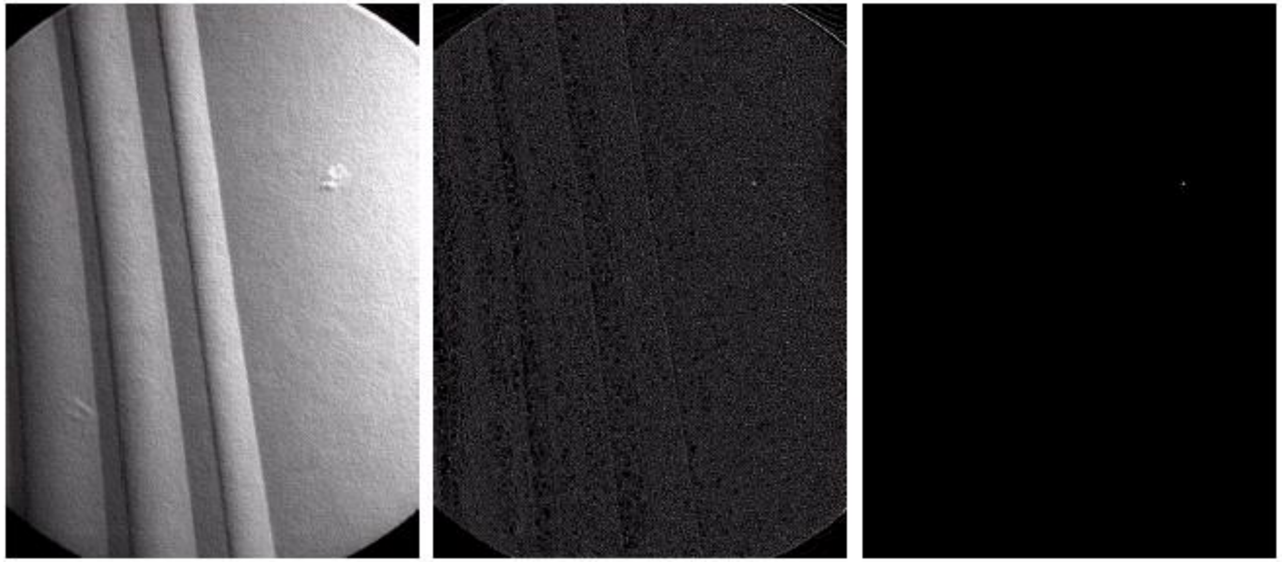
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Detection of Discontinuities
Point Detection

$$|R| \geq T$$

where T : a nonnegative threshold

–1	–1	–1
–1	8	–1
–1	–1	–1



Detection of Discontinuities
Line Detection

- Slightly more common than point detection is to find a one pixel wide line in an image.
- For digital images the three point straight lines are only horizontal, vertical, or diagonal (+ or –45°).

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

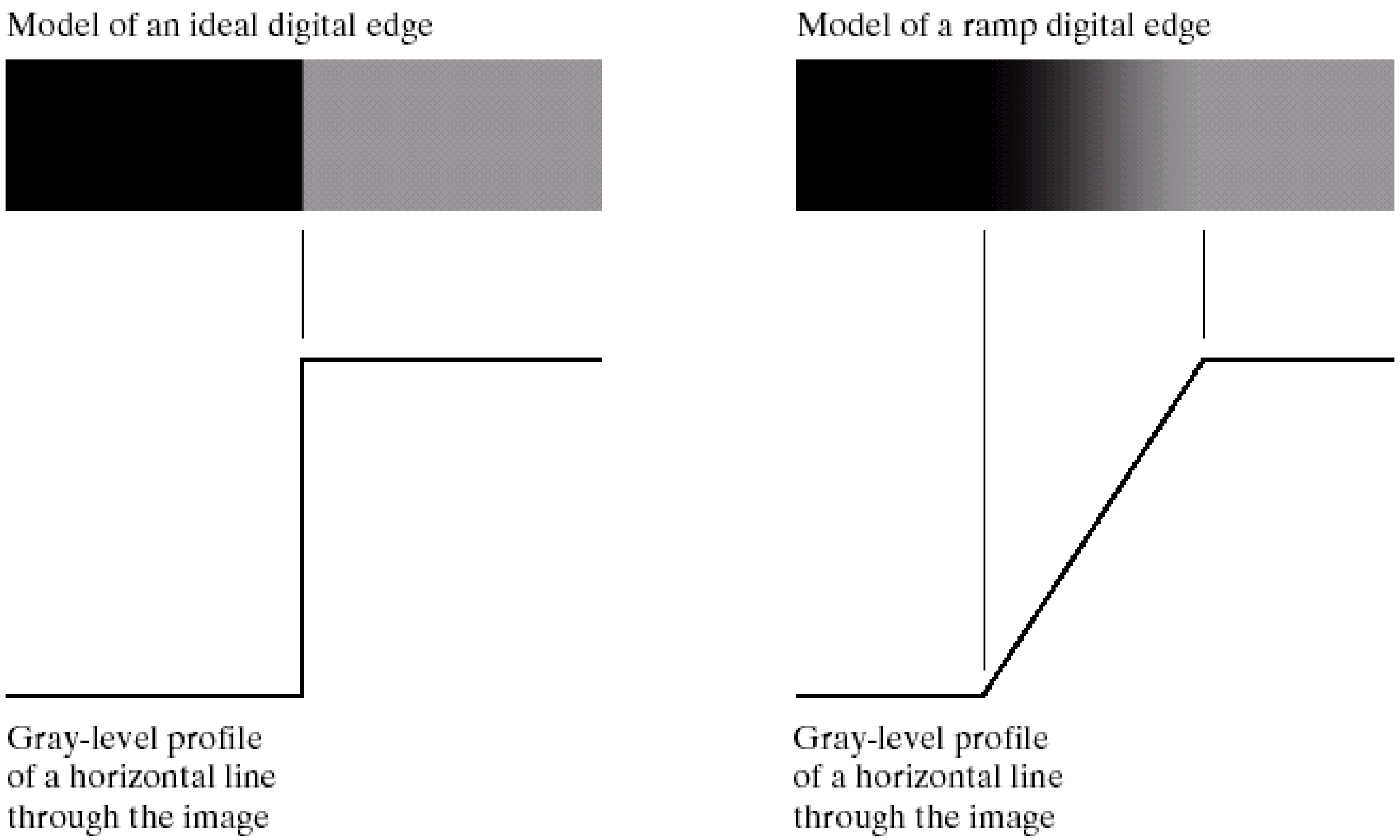
−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

−45°

Detection of Discontinuities
Edge Detection



Detection of Discontinuities
Gradient Operators

First-order derivatives:

- The gradient of an image *f*(*x*,*y*) at location (*x*,*y*) is defined as the vector:

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

- The magnitude of this vector: $\nabla f = \text{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2}$
- The direction of this vector: $\alpha(x, y) = \tan^{-1}\left(\frac{G_x}{G_y}\right)$

Detection of Discontinuities
Gradient Operators

Roberts cross-gradient operators →

−1	0
0	1

0	−1
1	0

Prewitt operators →

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Sobel operators →

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Detection of Discontinuities
Gradient Operators

Prewitt masks for detecting diagonal edges →

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

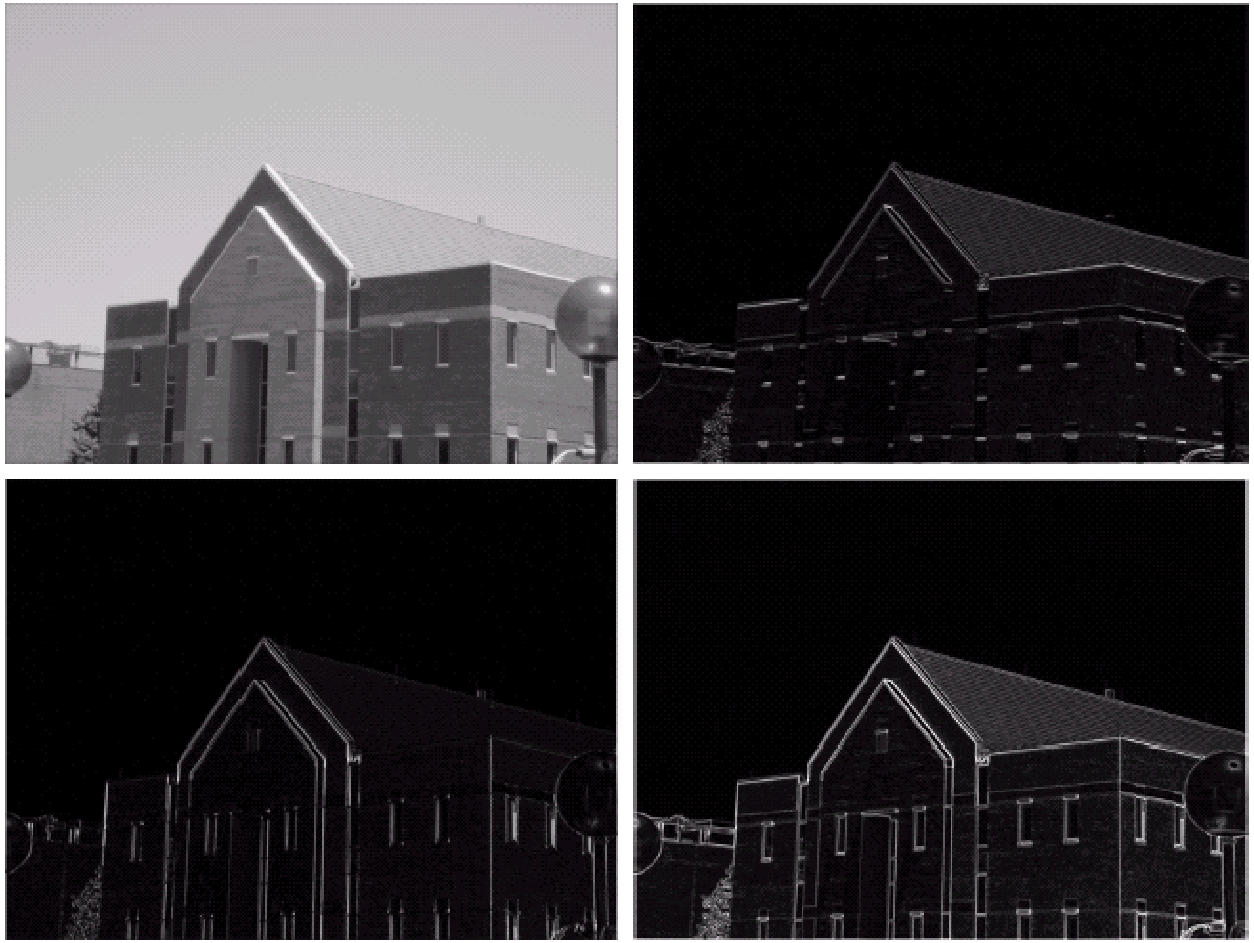
Sobel masks for detecting diagonal edges →

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Detection of Discontinuities
Gradient Operators: Examples

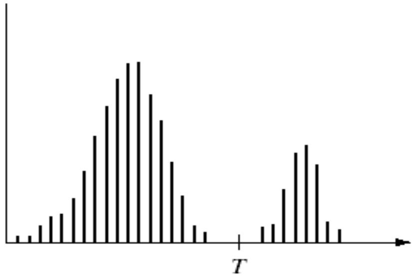
Detection of Discontinuities
Gradient Operators: Examples



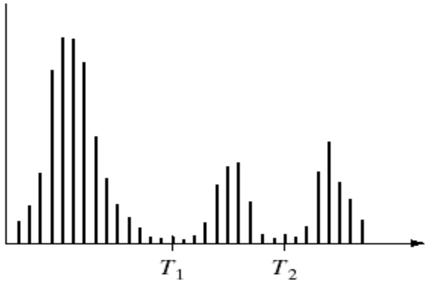
Smoothened with (5 X 5) Average Filter

Thresholding: Assumptions

- The intensity values are different in different regions
- Within each region, the intensity values are similar.

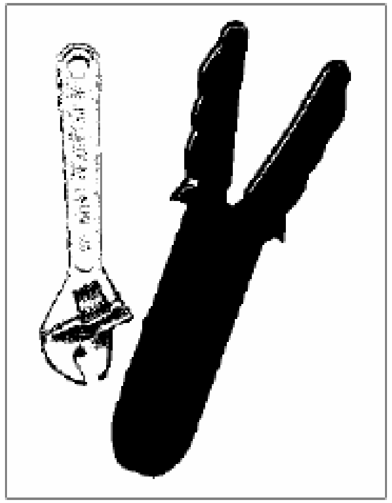


Single threshold

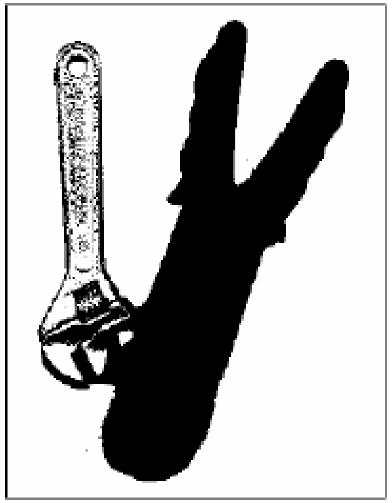


Multiple threshold

Original

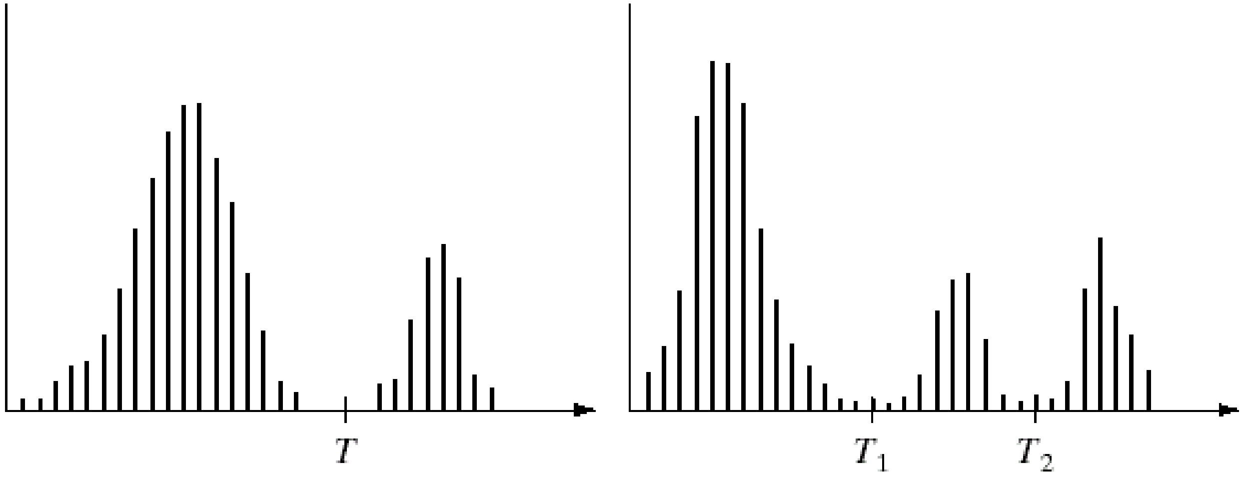


Threshold: 50



Threshold: 75

Thresholding



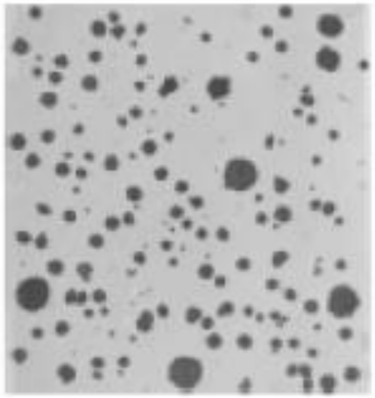
Single threshold   Multiple threshold

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

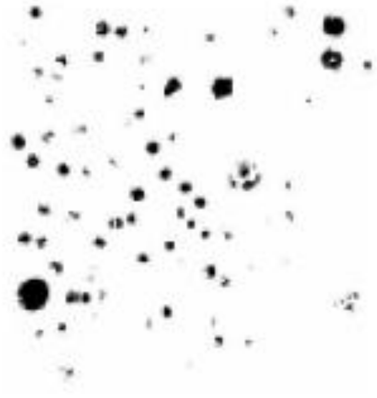
$$g(x,y)=\begin{cases}a, & \text{if } f(x,y)>T_2\\ b, & \text{if } T_1<f(x,y)\le T_2\\ c, & \text{if } f(x,y)\le T_1\end{cases}$$

Thresholding: Assumptions

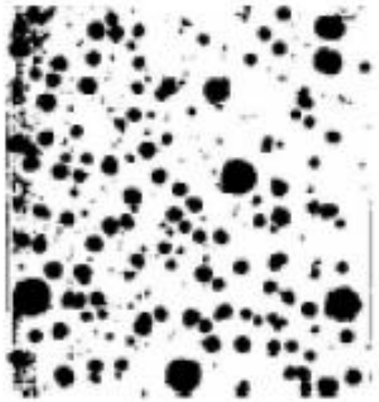
- Choosing a threshold is a critical task



Original



Too Low



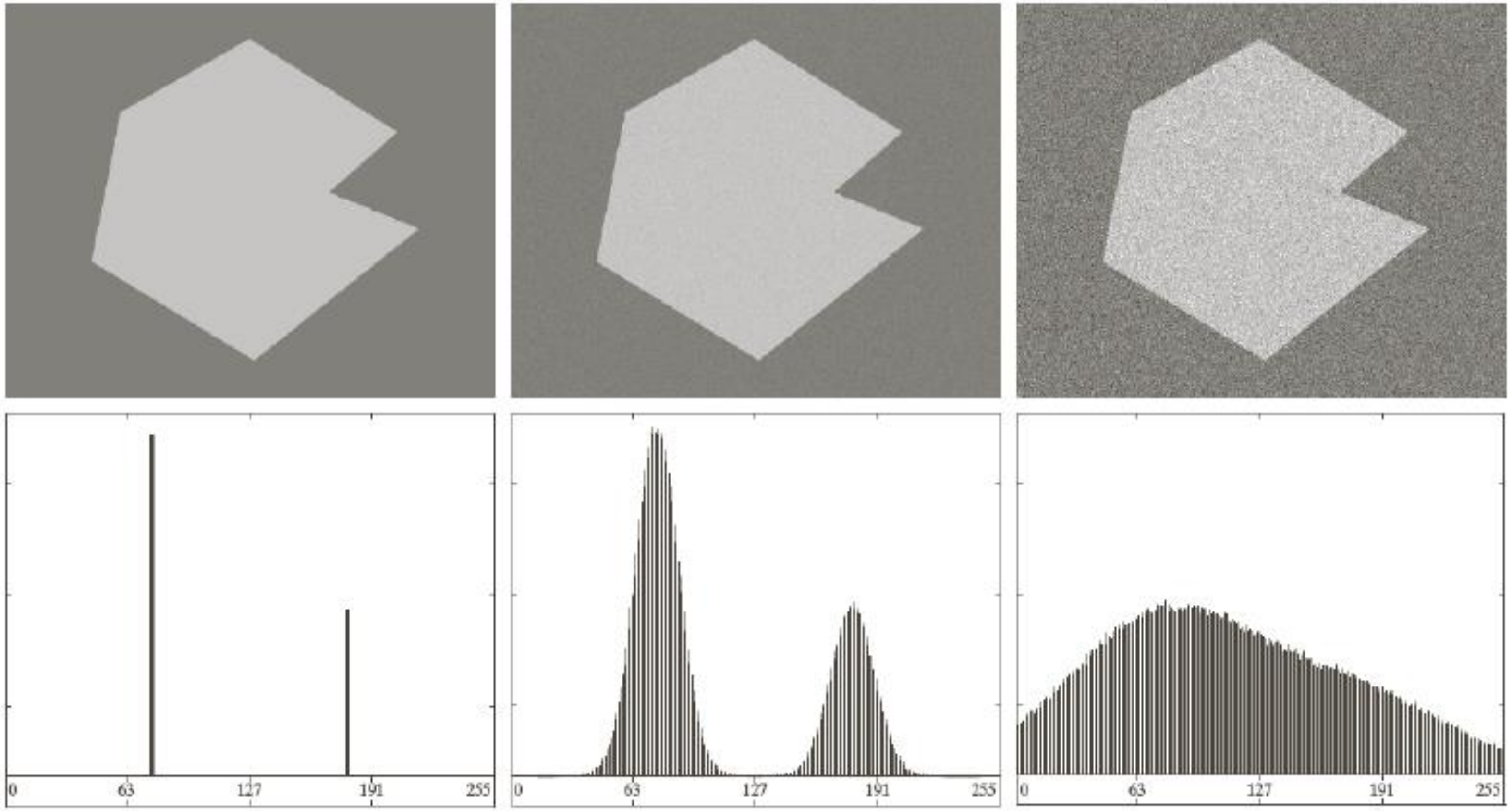
Too High

```
n=imread('nodules1.jpg');
figure(1); imshow(n);
n1=im2bw(n,0.35);
n2=im2bw(n,0.75);
figure(2), imshow(n1);
figure(3), imshow(n2);
```

Choosing the Threshold Value

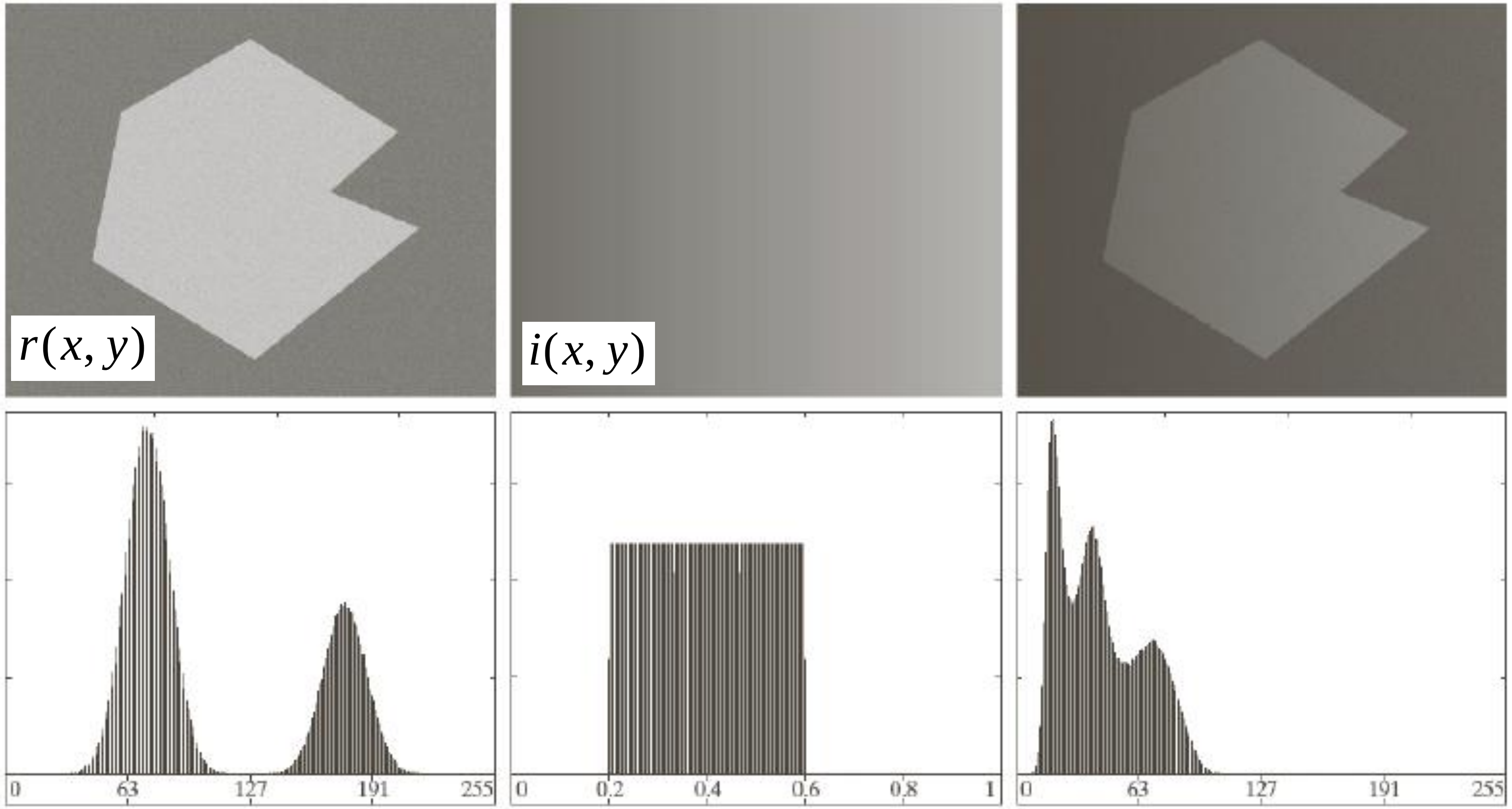
- Peaks and valleys of the image histogram can help in choosing the appropriate value for the threshold(s).
- Some factors affects the suitability of the histogram for guiding the choice of the threshold:
  - the separation between peaks;
  - the noise content in the image;
  - the relative size of objects and background;
  - the uniformity of the illumination;
  - the uniformity of the reflectance.

Thresholding
The Role of Noise



No Noise　　10 % Noise　　50 % Noise

Thresholding
The Role of Illumination and Reflectance



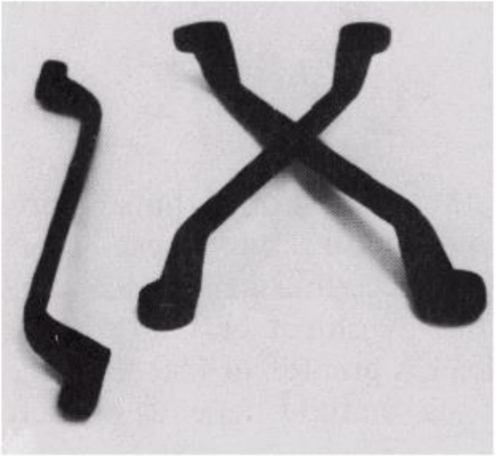
$$f(x,y) = i(x,y)r(x,y)$$

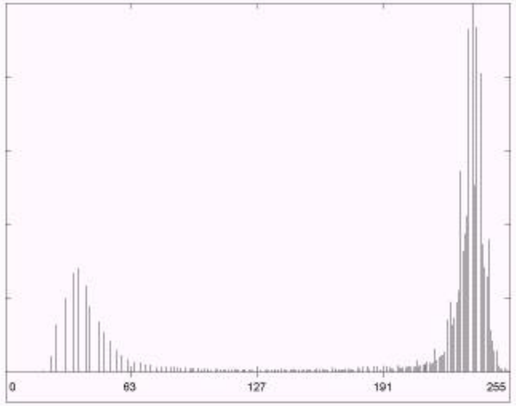
A B A . B

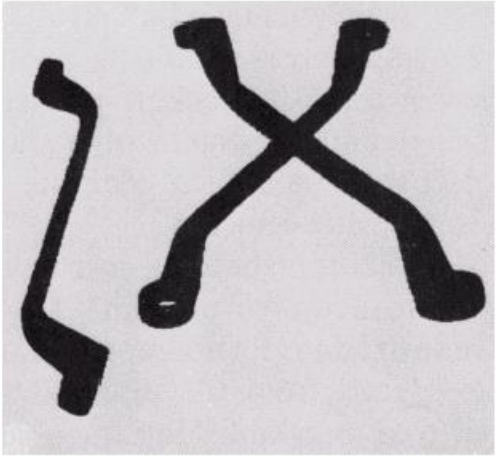
Global Thresholding
Automatically obtaining threshold, T

1. Select an initial estimate for *T*.
2. Segment the image using *T*. This well produce two groups of pixels: **G1** consisting of all pixels with gray level values > T and **G2** consisting of pixels with values < T.
3. Compute the average gray level values $\mu 1$ and $\mu 2$ for the pixels in regions *G1* and *G2*.
4. Compute a new threshold value: $T = \frac{1}{2}[\mu 1 + \mu 2]$
5. Repeat step 2 through 4 until the difference in *T* in successive iterations is smaller than a predefined parameter ***To***.

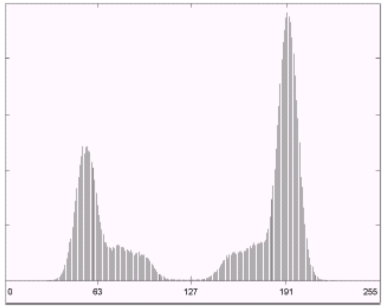
Global Thresholding
Automatically obtaining threshold, T: Example

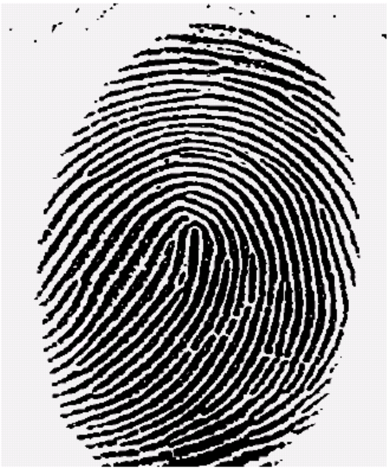






Global Thresholding
Automatically obtaining threshold, T: Example

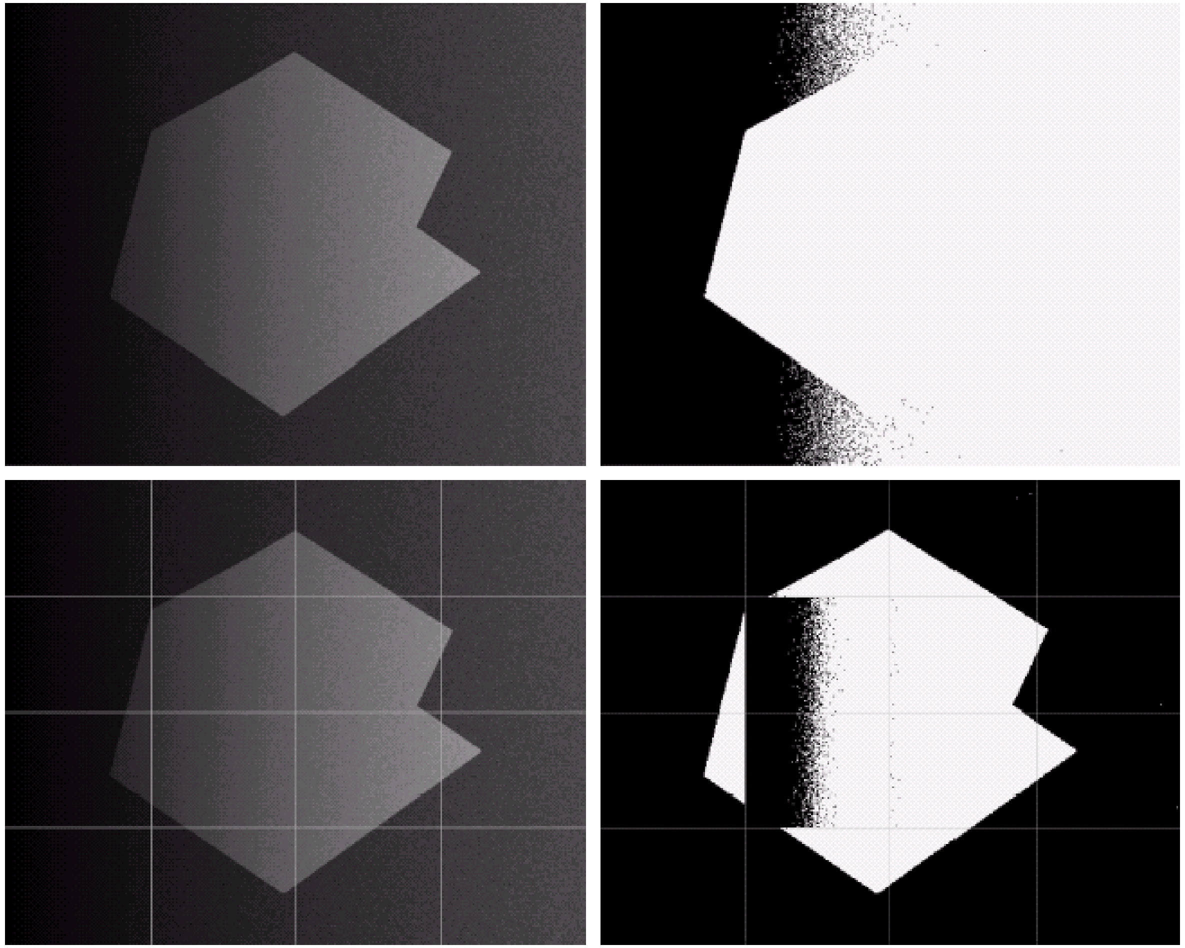


Adaptive Thresholding

You can simulate the effect with the following steps:

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

Adaptive Thresholding



Global Thresholding

Adaptive Thresholding

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

Definition

A group of connected pixels with similar properties

Image Interpretation

Region is an important concept in interpreting an image since regions may correspond to objects in the scene.

Region-Based Segmentation

Idea: Those pixels that correspond to an object are grouped together and marked.

Principles

- Similarity
  - Gray-value differences
  - Gray-value variances
- Spatial Proximity
  - Euclidean distance
  - Compactness of a region

Region-Based Segmentation
Basic Definition & Formulation

Definition

- As any other type of image segmentation, its main goal is to partition an image, I into regions Ri.
- Unlike edge-based segmentation, which returns boundaries between regions, region-based segmentation is a technique that allows us to determine the region directly.

Region-Based Segmentation
Basic Definition & Formulation

Formulation

- **Completeness:** The segmentation must be complete. (i.e) every pixel must be in a region. $\bigcup_{i=1}^{n} R_i = I$
- **Connectedness:** The points of a region must be connected in some sense.
- **Disjointness:** Regions must be disjoint. $R_i \bigcap R_j = \varnothing \quad \forall i = 1,2,...n$

Region-Based Segmentation
Basic Definition & Formulation

Formulation

- Satisfiability: Pixels of a region must satisfy one common property, P atleast. (i.e) any region must satisfy a homogeneity predicate, P. $P(R_i) = TRUE, \quad \forall i$
- Segmentability: Different regions satisfy different properties. (i.e) any two adjacent regions cannot be merged into a single region.

$$P\left(R_i \bigcup R_j\right) = FALSE$$

Region-Based Segmentation
Region Growing by pixel Aggregation

Input Seed Growing Final Region

Region-Based Segmentation
Region Growing by pixel Aggregation

Algorithm (for a single region):

- Let R be the region to extract.
  - Initially, the region R only contains its seed point p.
- Let F be a FIFO (First In, First Out) that contains the boundary points of R.
  - Initially, F contains the 8-neighborhood of the seed point p.
- **while** F is not empty
  - **for** each neighbor pixel p* of p in F
    - **if** p* is similar to p
      - p* is added to R
      - neighbor pixels of p* (not in R) are added to F
    - **else**
      - Set p* as non-similar

Region-Based Segmentation
Advantages and Disadvantages

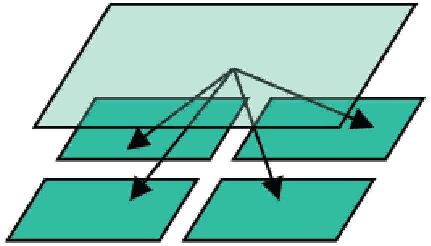
Advantages

- Fast method
- Conceptually simple.

Disadvantages

- Local Method: No global view of the problem.
- Gradient problem: In practice, there is almost always a continuous points related to color close that connects two points of an image.
- Thus unless we use a pre-defined variance (threshold), this will lead to the gradient problem.

Region Splitting and Merging Segmentation



Region Splitting

- Region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.

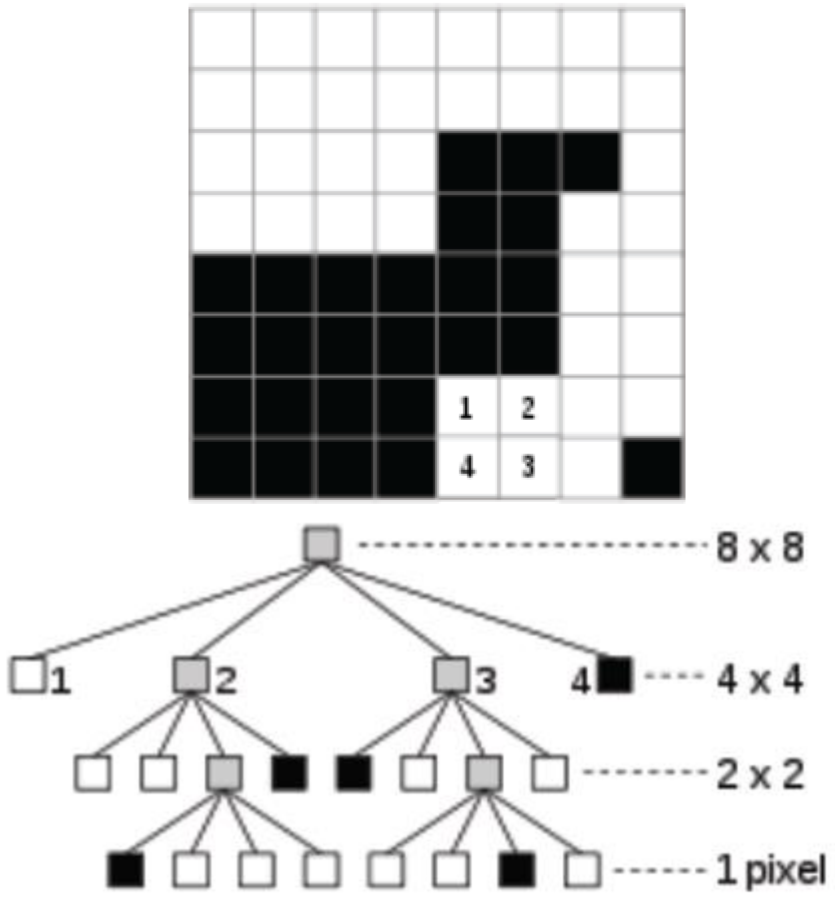
Region Merging

- Region merging is the opposite of region splitting and avoids over-segmentation.
- Start with small regions(eg. 2X2 or 4X4 regions) and merge the regions that have similar characteristics (gray level, variance).

Splitting and Merging: Data Structures

Quadtree for Splitting

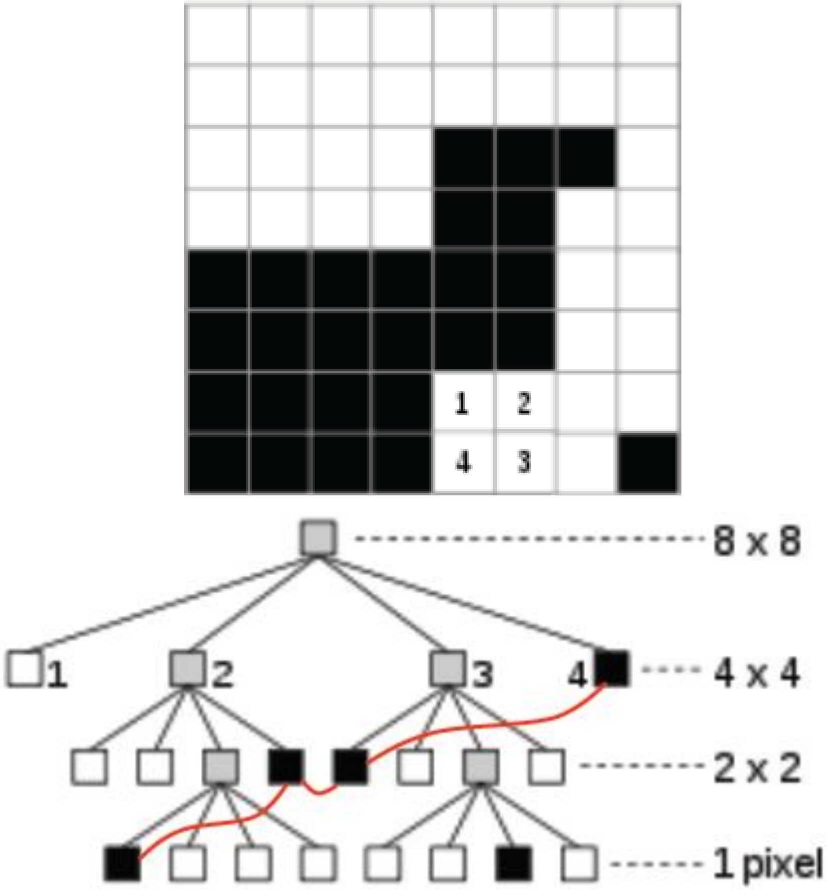
Splitting is a top-down procedure that creates regions that may be adjacent and homogeneous, but not ,merged



Splitting and Merging: Data Structures

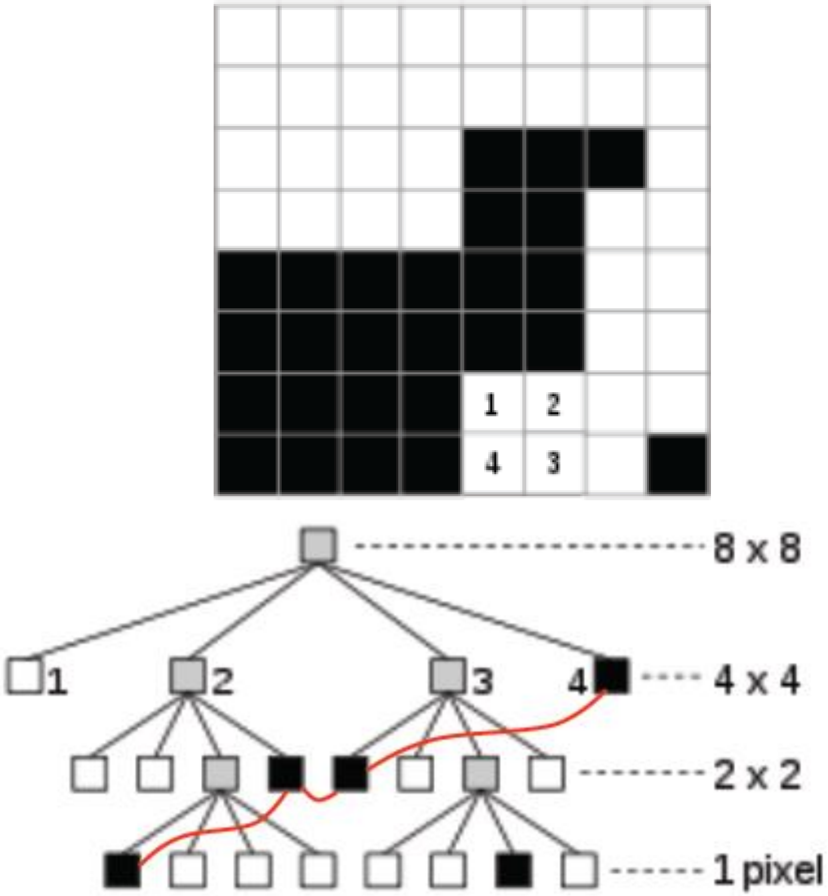
RAG (Region Adjacency Graph)

Splitting and merging work together iteratively.



Splitting and Merging Segmentation Algorithm

- If a region R is inhomogeneous [P(R)=FALSE], then R is split into four sub-regions.
- If two adjacent regions R_i, R_j are homogeneous [$P(R_i \cup R_j)$=TRUE], they are then merged.
- Stops when no further splitting or merging is possible.



Region Splitting: Example

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

original image

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

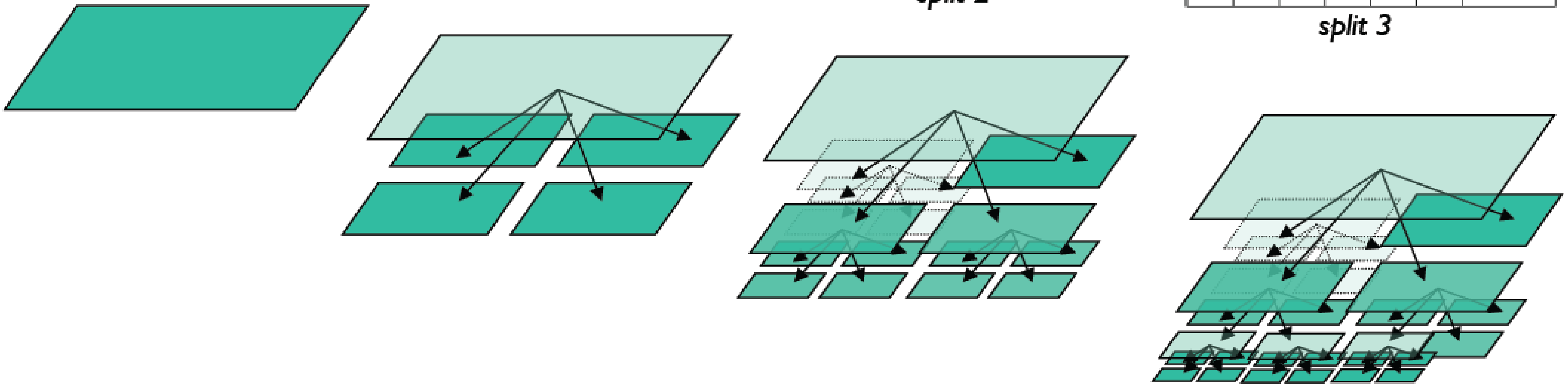
split 1

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

split 2

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

split 3



Region-Based Segmentation
Region Growing