Dr. Pushpa Mamoria
Senior Lecturer
Dept. of Computer Application
UIET, CSJM University, Kanpur
http://csjmu.ac.in/all-faculty/dr-pushpa-mamoria/
https://vidwan.inflibnet.ac.in/profile/226676

Feature Extraction:

1. Edge linking and Boundary detection
2. Thresholding
3. Edge based segmentation and region based segmentation
4. Introduction to Image Morphology

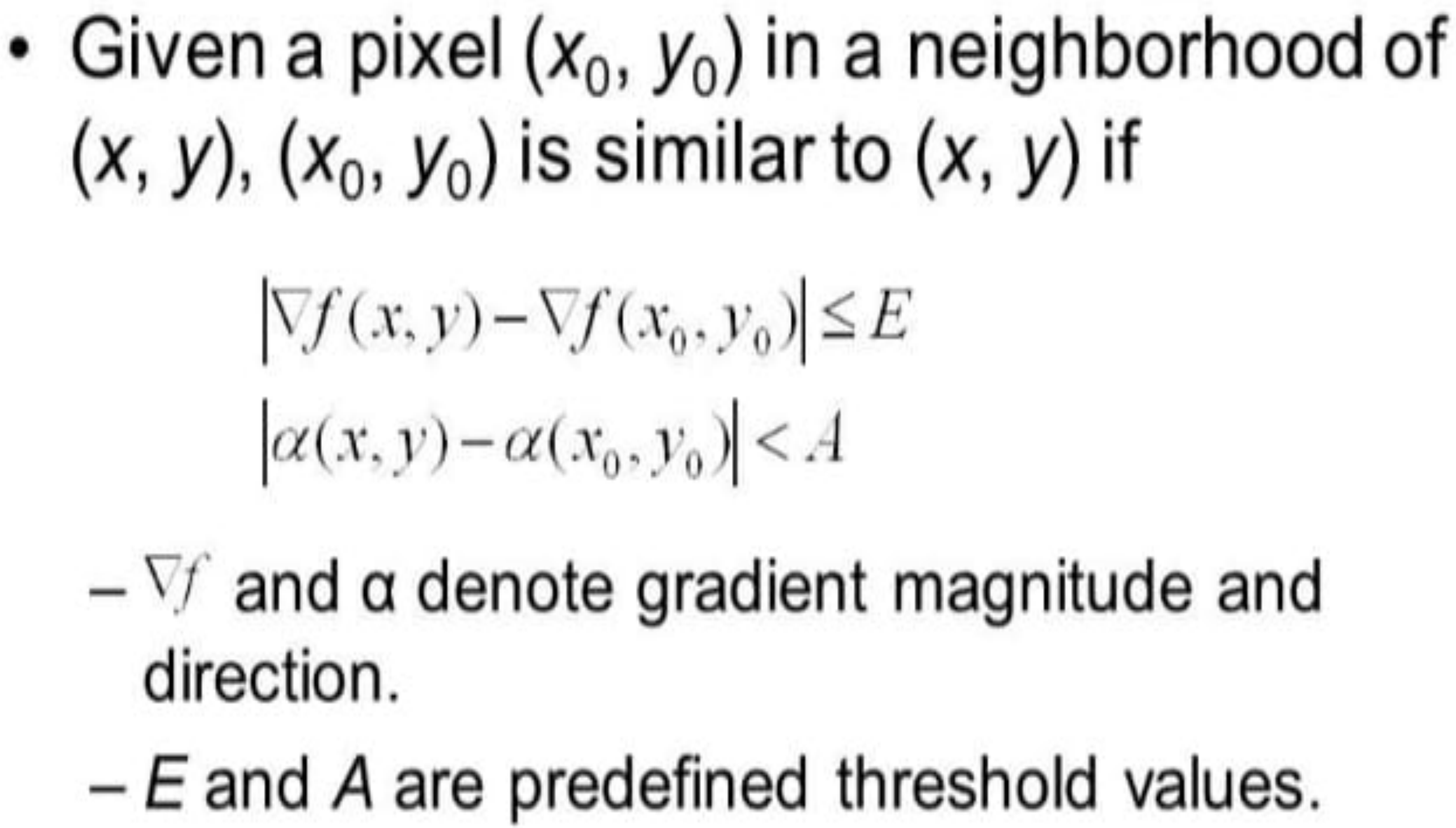
Detection of Discontinuities - Edge Linking

- Set of pixels from edge detecting algorithms, seldom define a boundary completely because of noise, breaks in the boundary etc.
- Edge detection is always followed by edge linking
- This section discusses edge algorithms that link edge pixels into meaningful forms

Local Processing

- Analysing the characteristics of edge pixels in a small neighbourhood
- → Gradient magnitude
- → Gradient direction
- All points that are similar according to a set of criteria are linked

Local Processing



- Given a pixel (x_0, y_0) in a neighborhood of (x, y), (x_0, y_0) is similar to (x, y) if

$$\left|\nabla f(x, y) - \nabla f(x_0, y_0)\right| \le E$$

$$\left|\alpha(x, y) - \alpha(x_0, y_0)\right| < A$$

  - ∇f and α denote gradient magnitude and direction.
  - E and A are predefined threshold values.

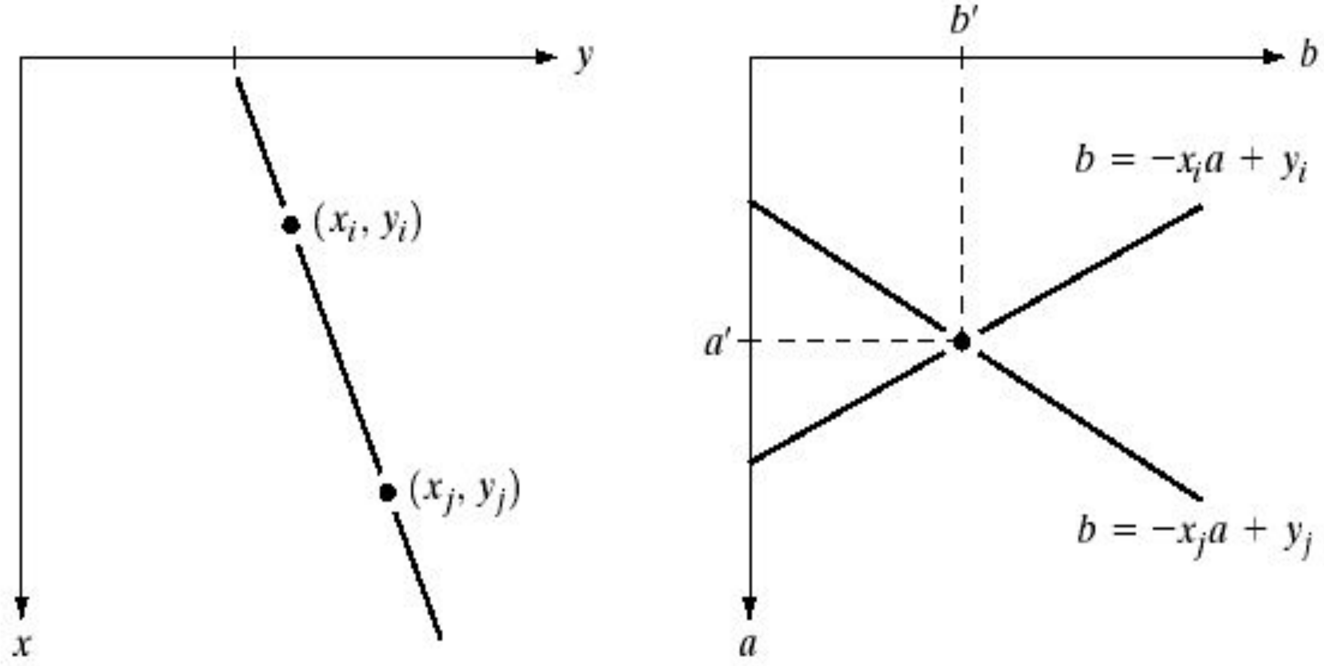
Contin….

- Two neighbouring pixels are linked if both magnitude and direction criteria are satisfied
- Linked points are assigned a different gray level.

Global processing via the Hough transform

- Motivation
  - Given a point (x_i , y_i), many lines pass through this point as $y_i = ax_i + b$ with different a and b.
  - Find all lines determined by every pair of points and find all subsets of points that are close to particular lines.
    - exhausted!

The Parameter Space



a b

FIGURE 10.17
(a) xy-plane.
(b) Parameter space.

Hough Transform

- Hough transform is a process that converts *xy*-plane to *ab*-plane (parameter space). Considering $b = -ax_i + y_i$ in *ab*-plane:
  - A point (x_i , y_i) in the image space is mapped to many points $\{(a, b)\}$ in the parameter space which are on line.
  - (x_j, y_j) is mapped to many points $\{(a, b)\}$ in the parameter space which are on line: $b=-ax_j+y_j$.
- These two points are collinear if $b = -ax_i + y_i$ and $b=-ax_j+y_j$ in the parameter space intersect at (a', b').

The Procedures of Hough Transform

Step 1:

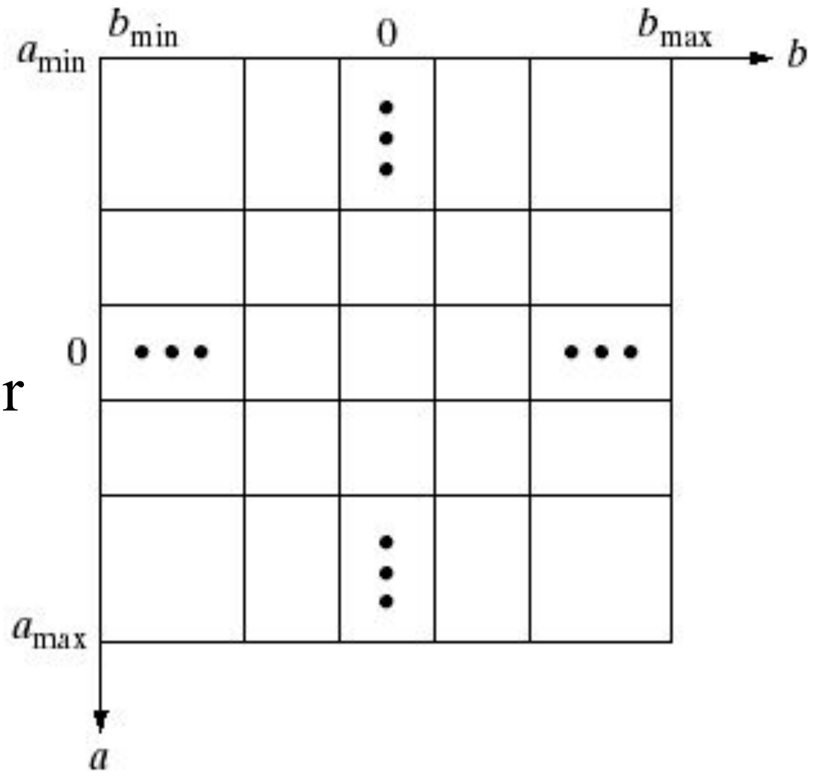
Subdivide *ab*-plane to accumulator cells. Let $A(i, j)$ be the cell at (i, j) where $a_{\min} \le a_i \le a_{\max}$, $b_{\min} \le b_j \le b_{\max}$ and $A(i, j)=0$.

Step 2:

For every (x_k, y_k), find $b=-x_k a_p+y_k$ for each allowed p.

Step 3:

Round off b to the nearest allowed value b_q. Let $A(p,q)=A(p,q)+1$.



Performance and Limitation

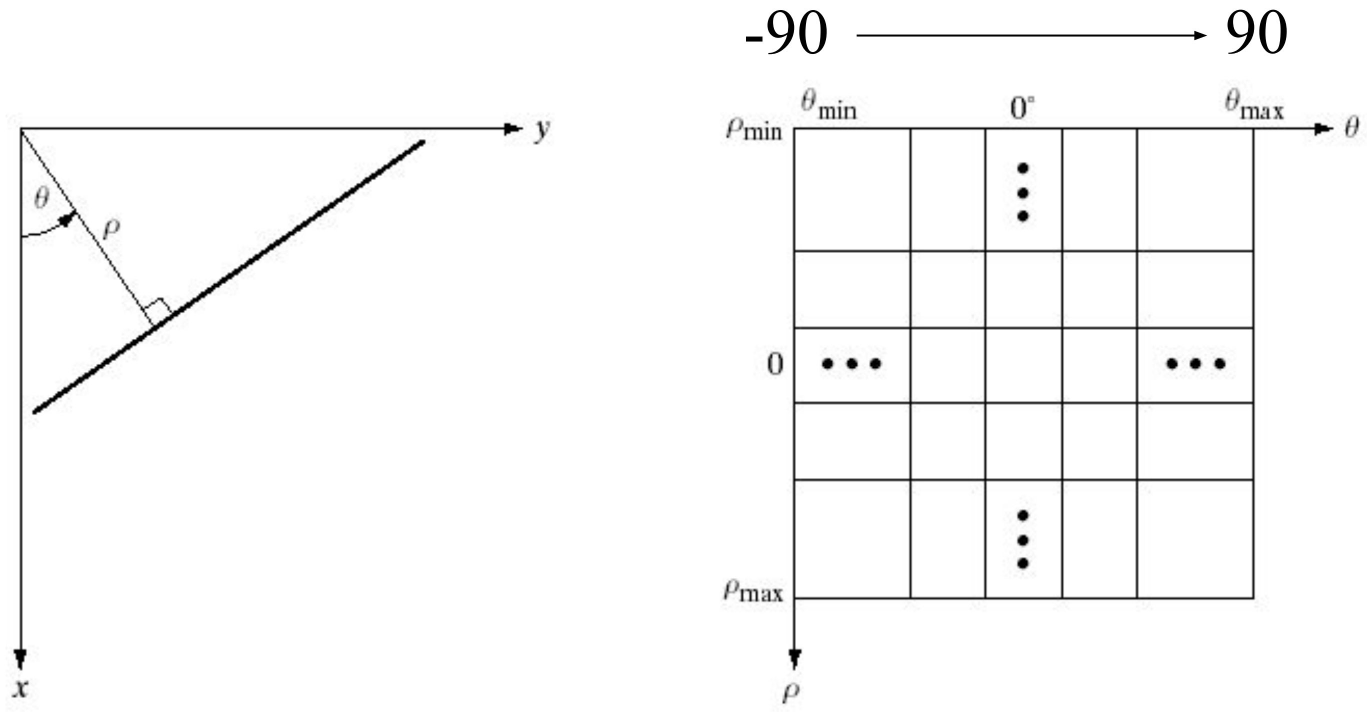
- Performance
  - With *n* image points and *K* accumulator cells, there are *nK* computation involved.
- Limitation
  - The slope approaches infinitely as the line approaches the vertival.
  - Solution:
    - Rewrite the normal representation of a line to

      $x_i \cos\theta + y_i \sin\theta = \rho$

Rewriting a line as $x\cos\vartheta + y\sin\vartheta = \rho$

- This implementation is identical to the method using slope-intercept representation.
- Instead of straight lines, the loci are sinusoidal curves in $\rho\vartheta$ -plane. □

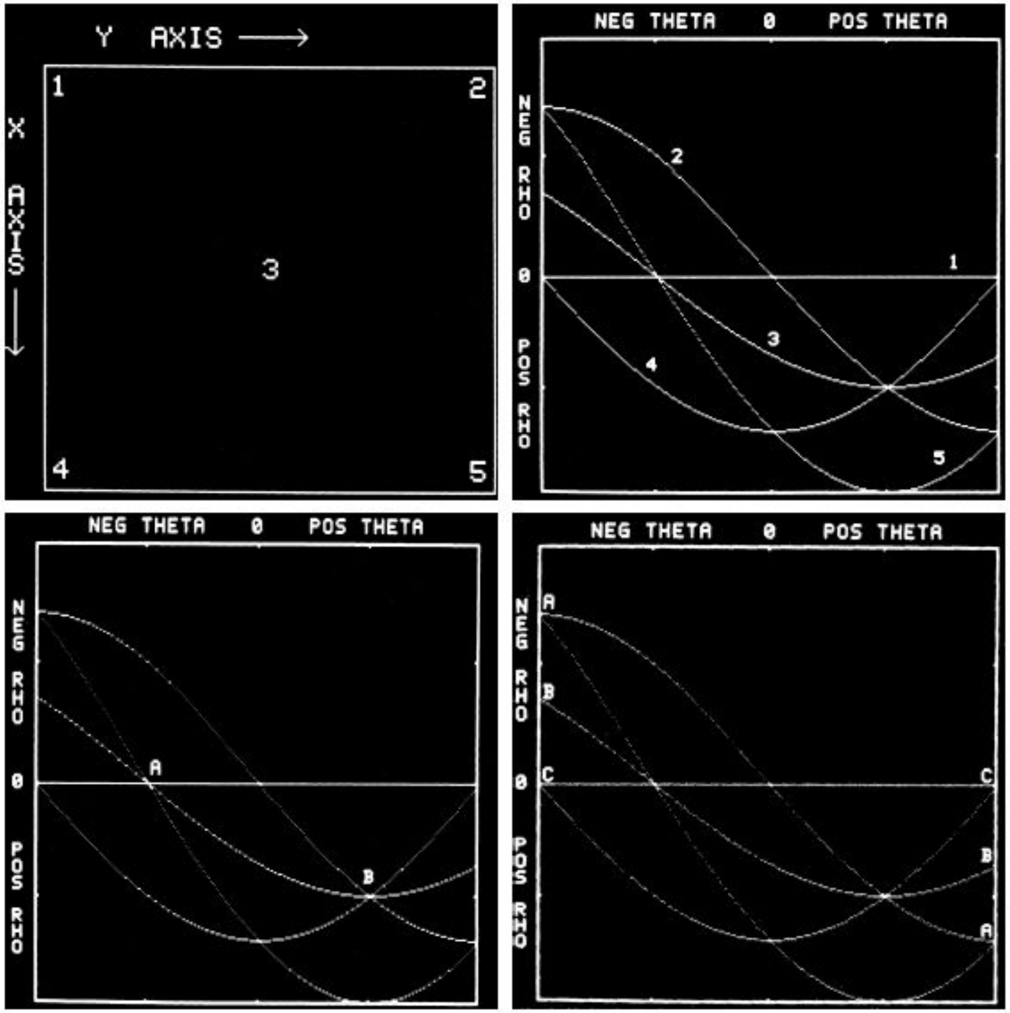
Rewriting a line as $x \cos\theta + y \sin\theta = \rho$



a b

FIGURE 10.19 (a) Normal representation of a line. (b) Subdivision of the $\rho\theta$-plane into cells.

Example 10.7



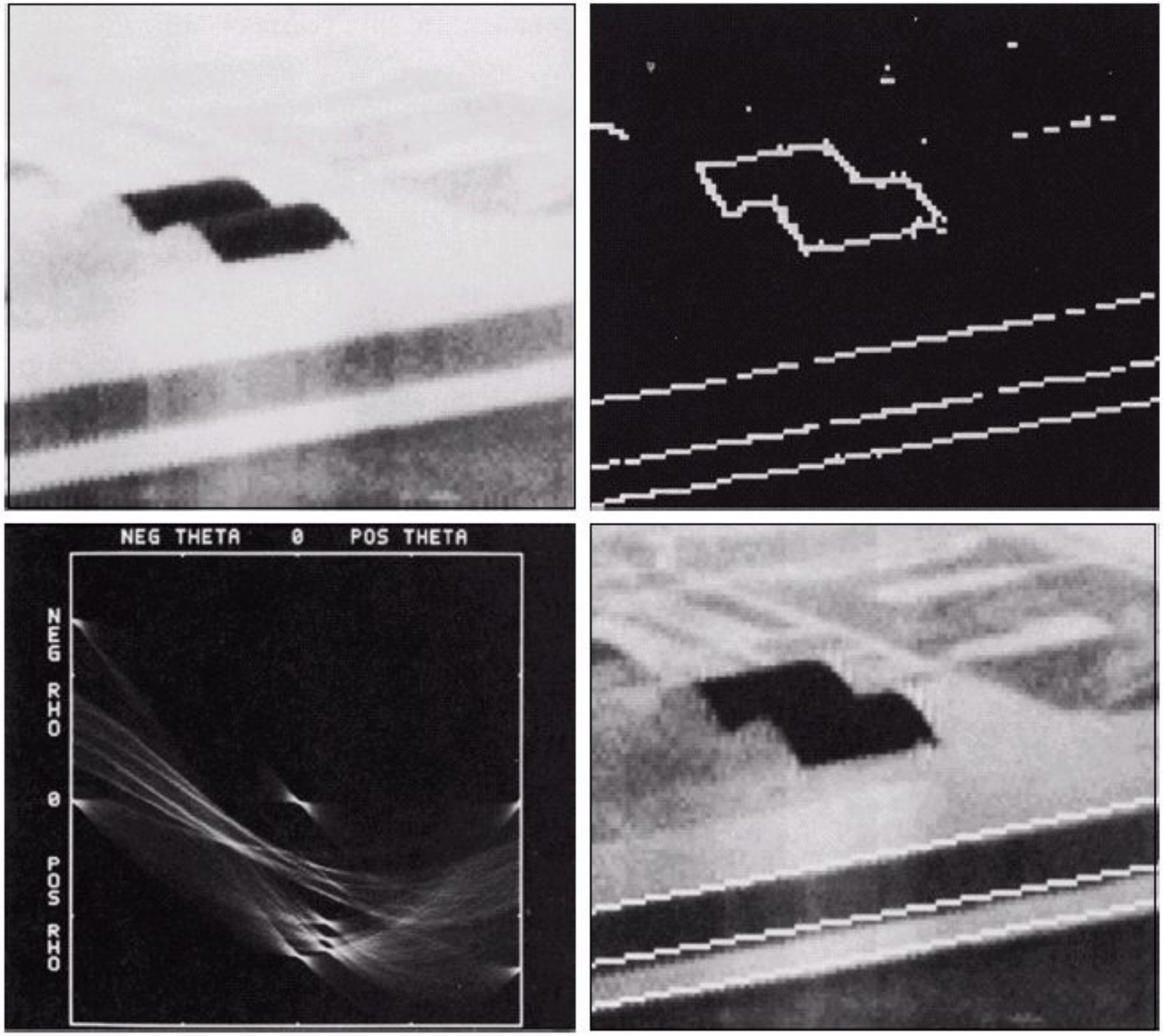
a b
c d

FIGURE 10.20 Illustration of the Hough transform. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

Edge-Linking Using Hough Transform

1. Compute the gradient of an image and threshold it to obtain a binary image.
2. Specify subdivisions in the $\rho\vartheta$ -plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (for continuity) between pixels in a chosen cell.□

Example 10.8



a b
c d

FIGURE 10.21 (a) Infrared image. (b) Thresholded gradient image. (c) Hough transform. (d) Linked pixels. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

Edge Linking and Boundary Detection

- Set of pixels from edge detecting algorithms, seldom define a boundary completely because of noise, breaks in the boundary etc.

- Therefore, Edge detecting algorithms are usually followed by linking and other detection procedures, designed to assemble edge pixels into meaningful boundaries

Two Methods

► Local Processing (Local edge linker)

►Global Processing (Global edge linker)

Local Processing

Two principal properties for establishing similarity of edge pixels:

- Strength of the response of the gradient operator used to produce the edge pixel

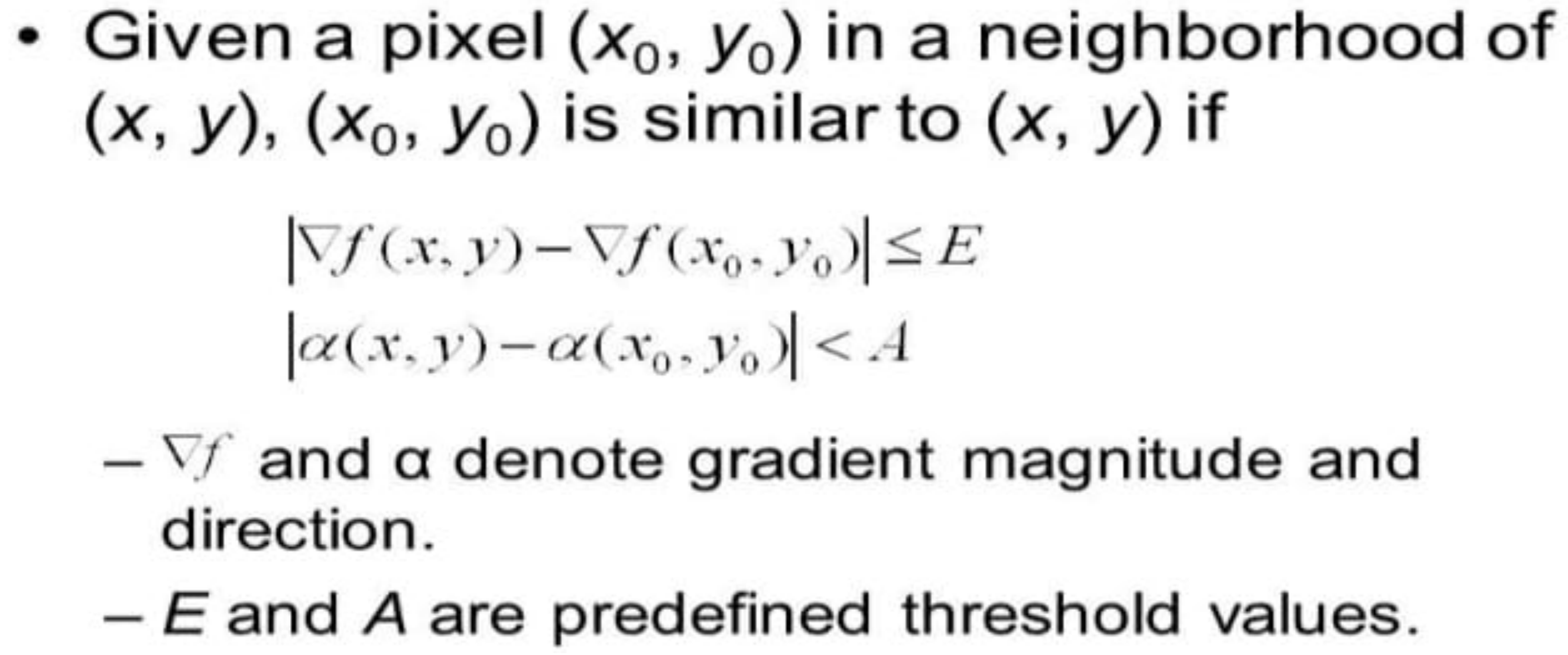
- Direction of the gradient vector.

Local Processing

- Analyzing the characteristics of edge pixels in a small neighborhood.

  - Gradient magnitude.

  - Gradient direction.



- Given a pixel (x_0, y_0) in a neighborhood of (x, y), (x_0, y_0) is similar to (x, y) if

$$\left|\nabla f(x,y) - \nabla f(x_0,y_0)\right| \le E$$

$$\left|\alpha(x,y) - \alpha(x_0,y_0)\right| < A$$

  – ∇f and α denote gradient magnitude and direction.

  – E and A are predefined threshold values.

Local Processing

- In other words (cont.):

2.
$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

(x’,y’) and (x,y) are similar if:

$$\left|\alpha(x, y) - \alpha(x', y')\right| < A$$

where A is an angle threshold.

Global processing: Hough transform

- Now consider global relationships between pixels.
- Suppose that, for n points in an image, we want to find all subsets of these points that lie on straight lines

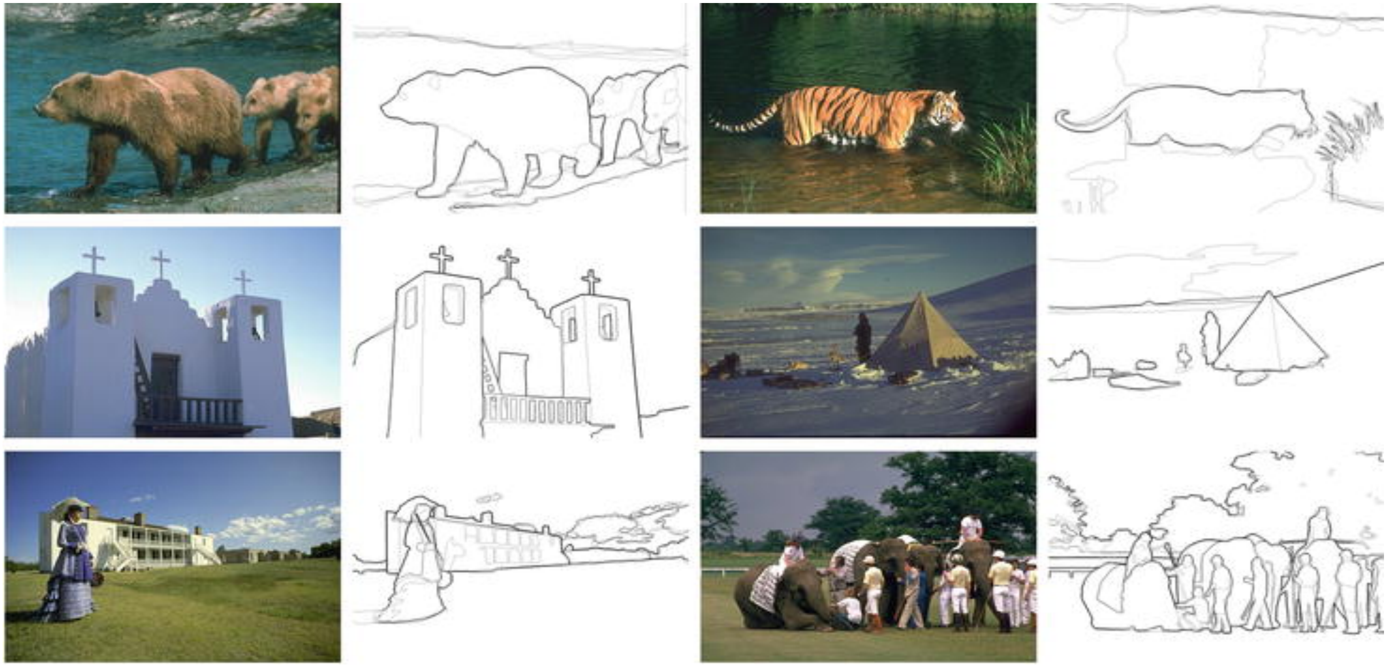
Global Processing via the Hough Transform

- Since a,b approach infinity as a line approaches the vertical, we can use the normal representation of a line:

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

Boundary Detection

- *Boundary detection* is the process of detecting and localizing salient boundaries between objects in a scene.



- The *Berkeley Segmentation Data set* (BSDS) was established in 2001 and quickly became the standard benchmark for **Boundary detection**.

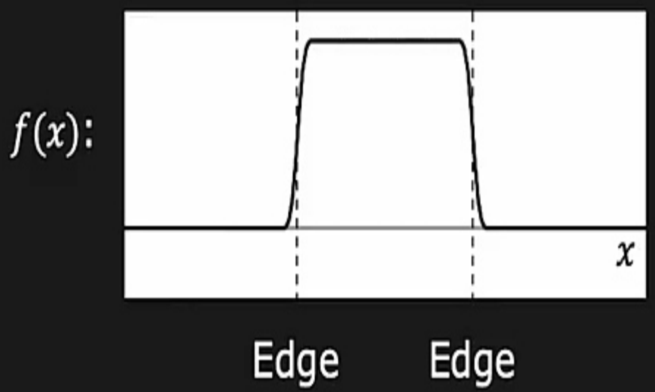
- In comparison, boundary detection is usually viewed as a mid-level process of finding boundaries of (and between) objects in scenes, thus having close ties with both grouping/segmentation and object shape. A large-scale data set of natural images with human-marked groundtruth boundaries

Boundary Detection is a vital part of extracting information encoded in images, allowing for the computation of quantities of interest including density, velocity, pressure, etc.

- Boundary detection is closely related to, but not identical with, edge detection. Edge detection is a classical problem in computer vision which aims at finding brightness discontinuities. Edge detection is usually viewed as a low-level process of feature extraction that works under the assumption of ideal edge models (such as step and ridge edges).

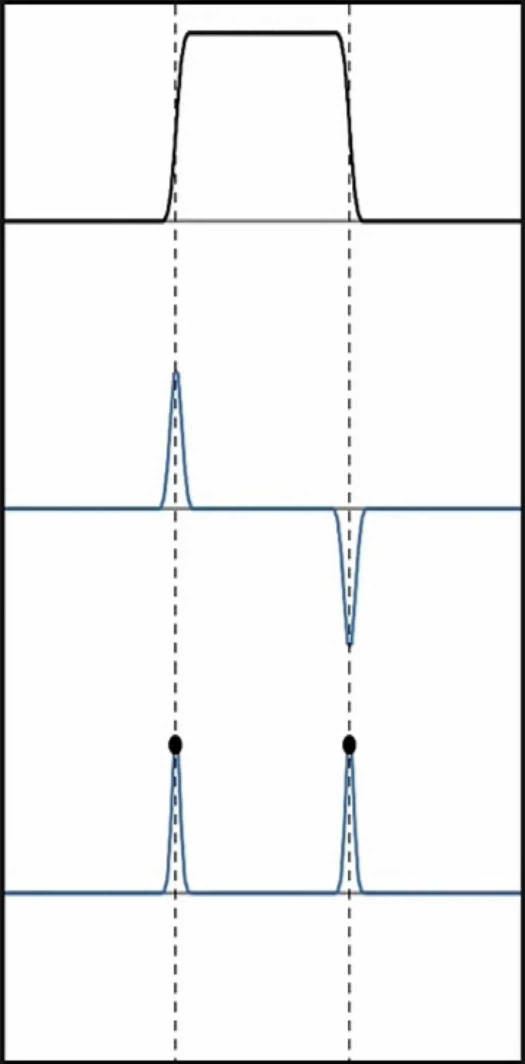
1D Edge Detection

Edge is a rapid change in image intensity in a small region.



Basic Calculus: Derivative of a continuous function represents the amount of change in the function.

Edge Detection Using 1st Derivative



$f(x)$

First Derivative: $\frac{\partial f}{\partial x}$

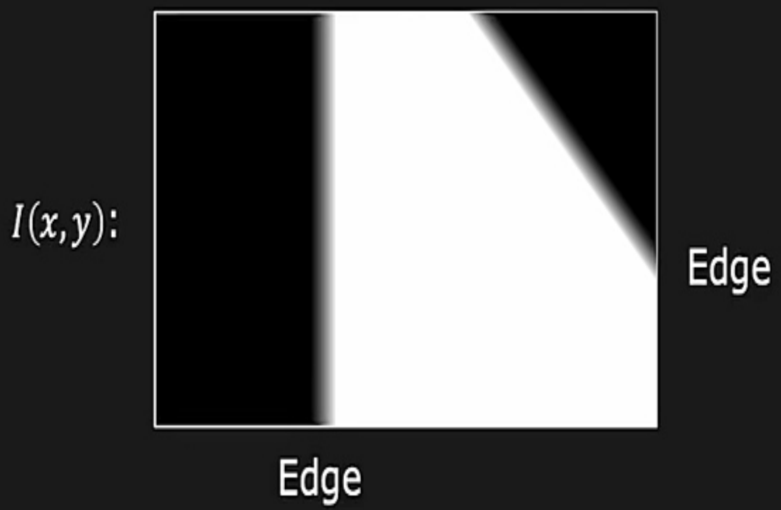
First Derivative Absolute Value: $\left|\frac{\partial f}{\partial x}\right|$

Local Extrema Indicate Edges

Local Maxima Indicate Edges

Provides Both Location and Strength of an Edge

2D Edge Detection



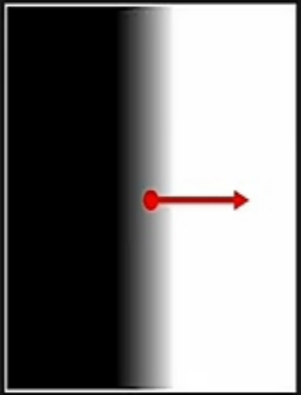
Basic Calculus: Partial Derivatives of a 2D continuous fun[illegible] represents the amount of change along each dimension.

Gradient (∇)

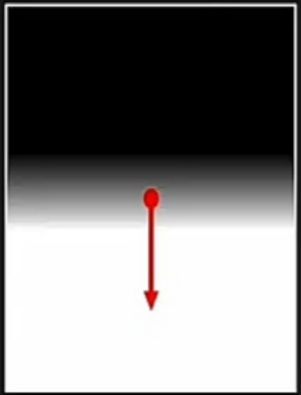
Gradient (Partial Derivatives) represents the direction of most rapid change in intensity

$$\boxed{\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]}$$

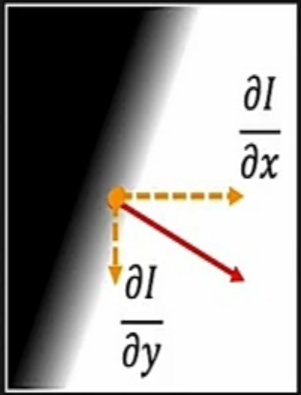
Pronounced as "Del I"



$$\nabla I = \left[\frac{\partial I}{\partial x}, 0\right]$$



$$\nabla I = \left[0, \frac{\partial I}{\partial y}\right]$$

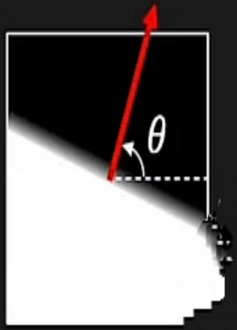

$\frac{\partial I}{\partial x}$
$\frac{\partial I}{\partial y}$

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

Gradient (∇) as Edge Detector

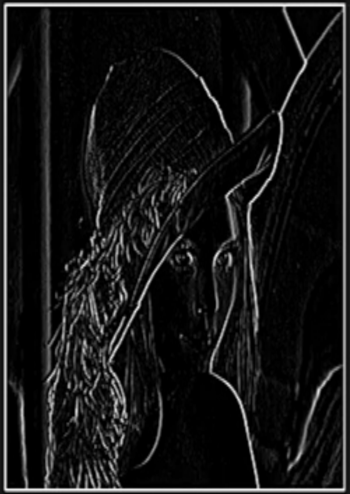
Gradient Magnitude $S = \|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

Gradient Orientation $\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$

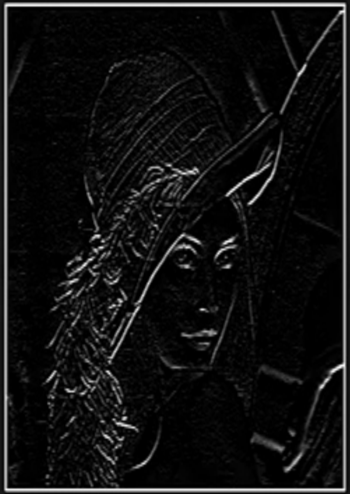


Gradient (∇) Using Sobel Filter

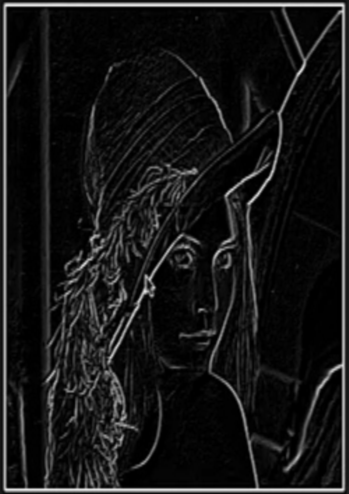
Image (I)



$\partial I/\partial x$



$\partial I/\partial y$



Gradient Magnitude

Edge Thresholding

Standard: (Single Threshold T)

$\|\nabla I(x,y)\| < T$	Definitely Not an Edge
$\|\nabla I(x,y)\| \geq T$	Definitely an Edge

Sobel Edge Detector

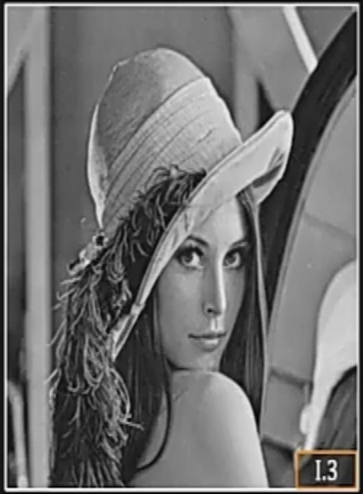
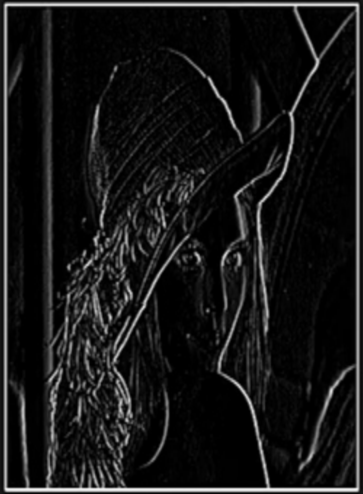
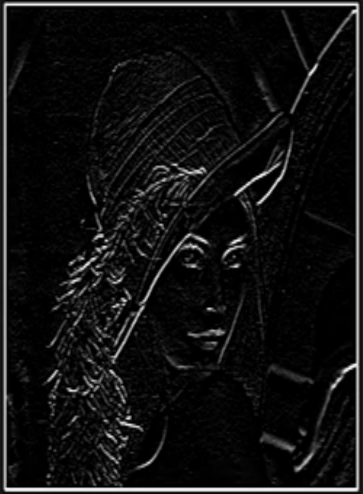


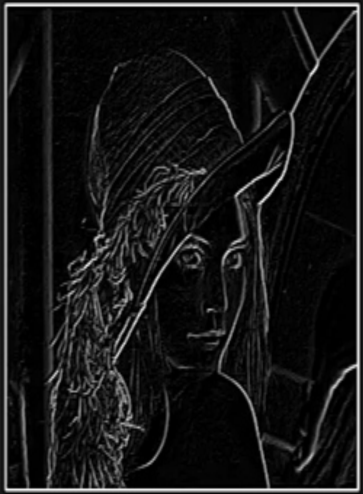
Image (I)



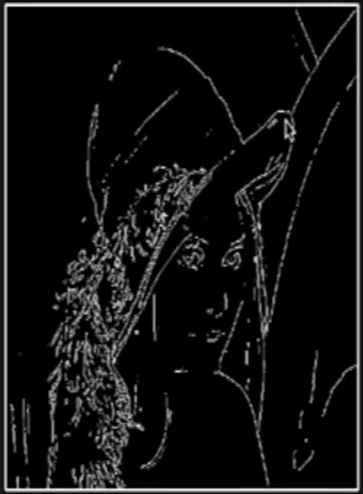
$\partial I / \partial x$



$\partial I / \partial y$



Gradient Magnitude

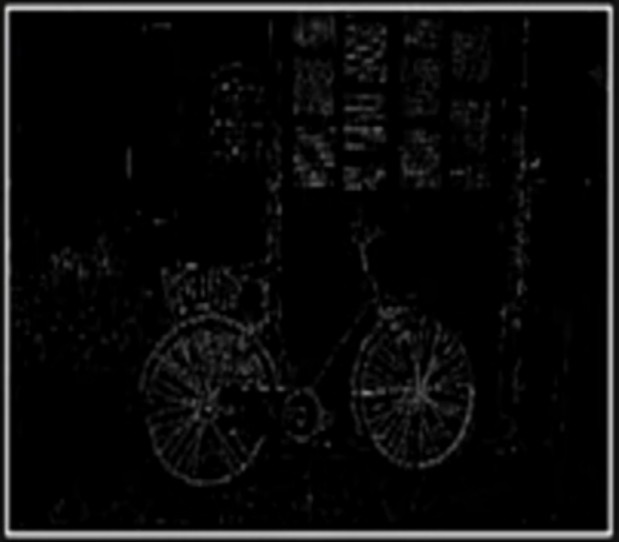


Thresholded Edge

Global Processing - Hough Transform

- Hough transformation is a features extraction method for detecting simple shapes such as circle, lines etc. in an image.
- Hough transformation takes the images created by edge detection operations but most of the time edge map is disconnected.
- Therefore Hough transformation is used to connect the disjoined edge points.

Difficulties for the Fitting Approach

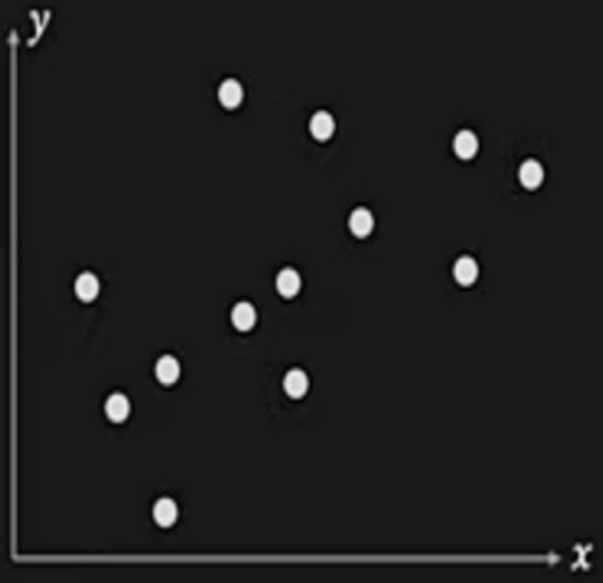


- Extraneous Data: Which points to fit to?
- Incomplete Data: Only part of the model is visible.
- Noise

Solution: Hough Transform

Hough Transform: Line Detection

Given: Edge Points (x_i, y_i)

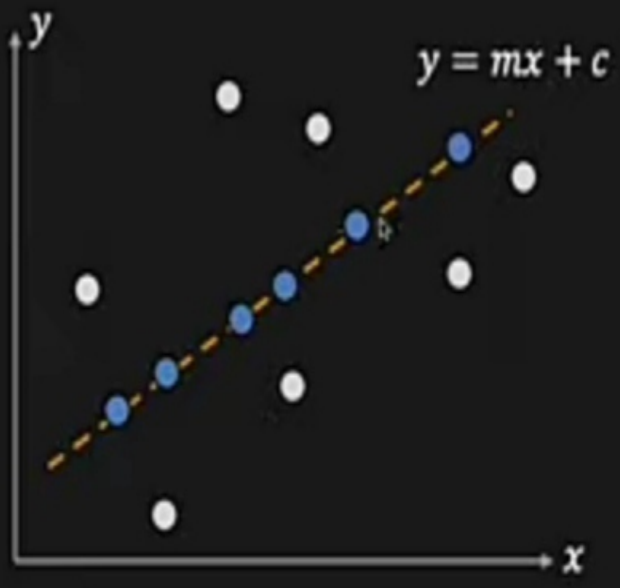


Hough Transform: Line Detection

Given: Edge Points (x_i, y_i)

Task: Detect line

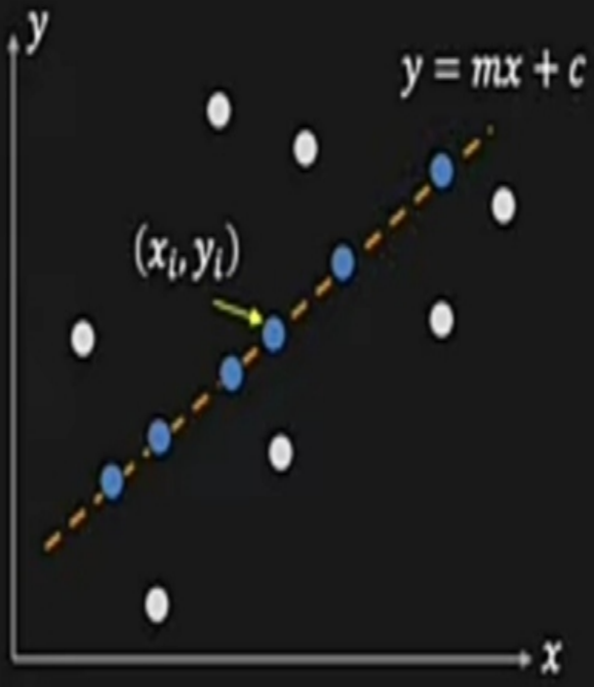
$$y = mx + c$$



Given: Edge Points (x_i, y_i)

Task: Detect line

$$y = mx + c$$



Consider point (x_i, y_i)

$$y_i = mx_i + c \quad \Longleftrightarrow \quad c = -mx_i + y_i$$

Hough Transform: Concept

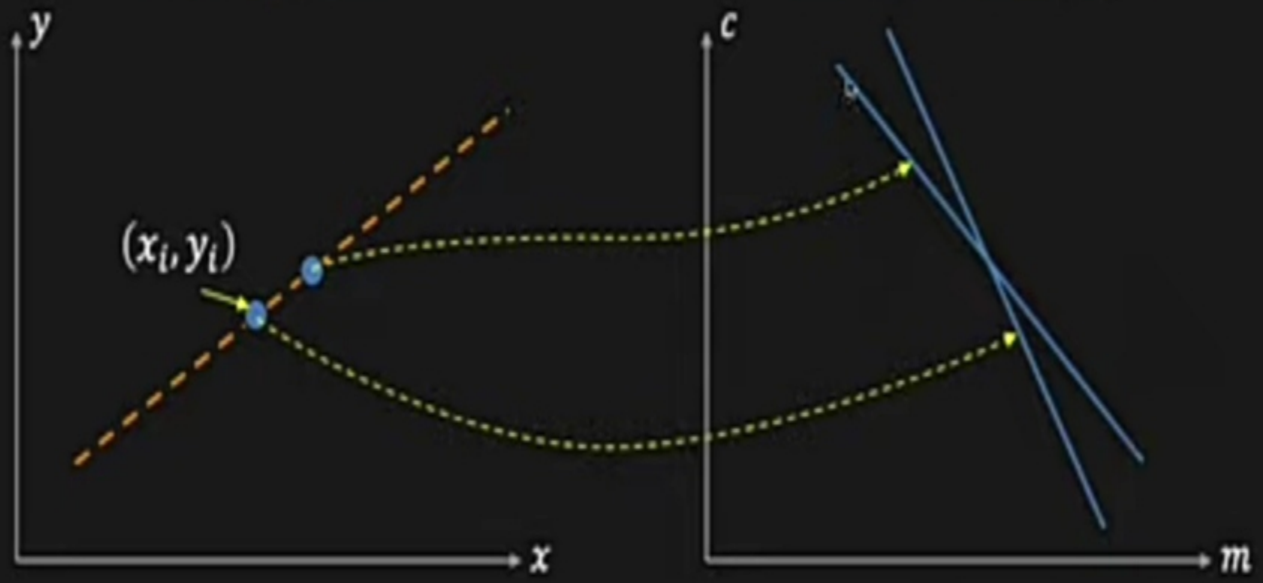


Image Space

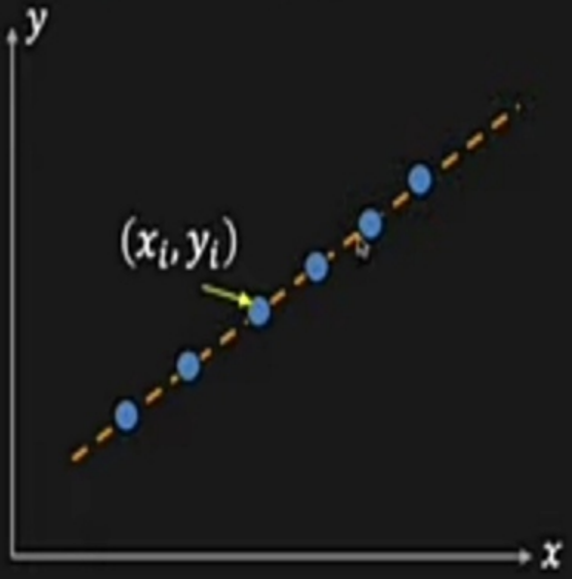
Parameter Space

$y_i = mx_i + c$

$c = -mx_i + y_i$

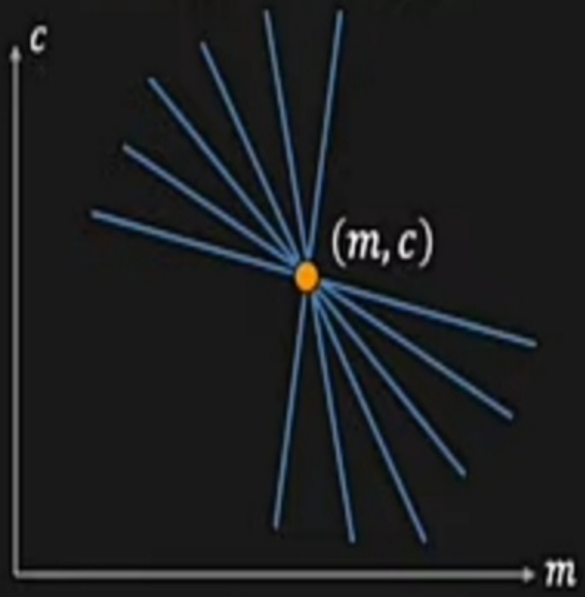
Hough Transform: Concept

Image Space



$$y_i = mx_i + c$$

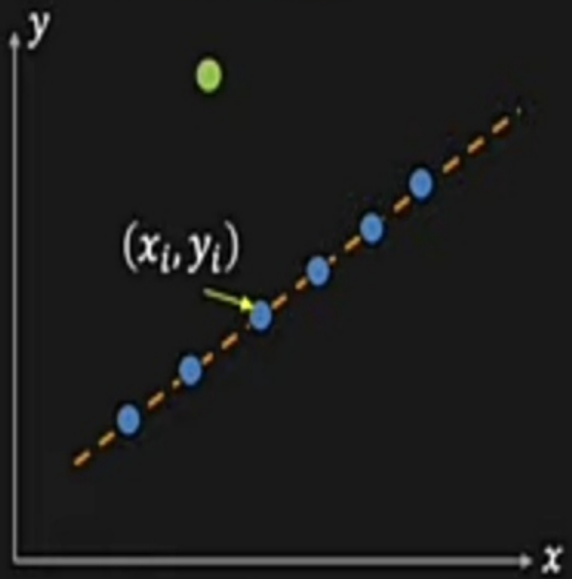
Parameter Space



$$c = -mx_i + y_i$$

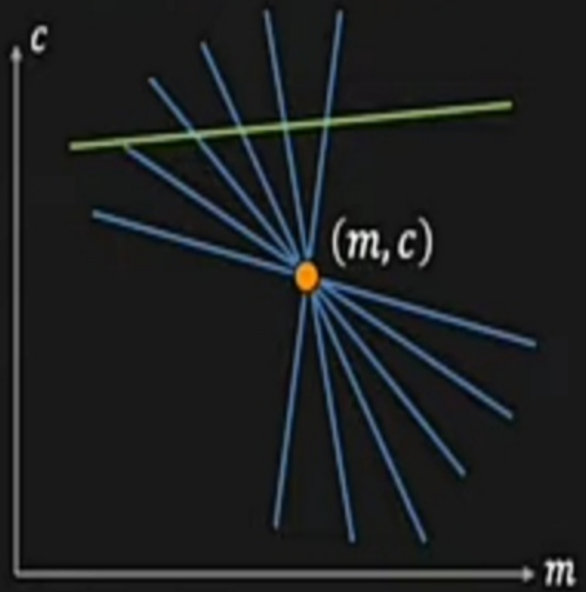
Hough Transform: Concept

Image Space



$y_i = mx_i + c$

Parameter Space



$c = -mx_i + y_i$

Point ⟷ Line

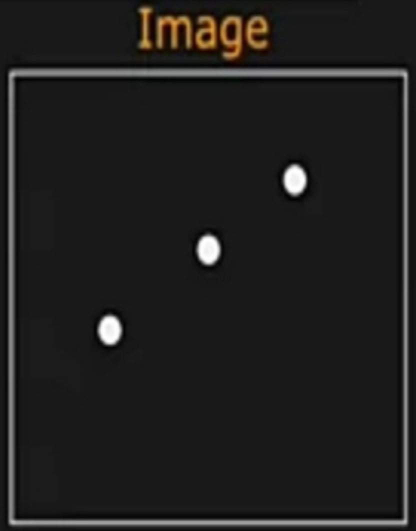
Line ⟷ Point

Line Detection Algorithm

Step 1. Quantize parameter space (m, c)

Step 2. Create accumulator array $A(m, c)$

Step 3. Set $A(m, c) = 0$ for all (m, c)



Image

$A(m,c)$

c	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0

Line Detection Algorithm

Step 1. Quantize parameter space (m, c)

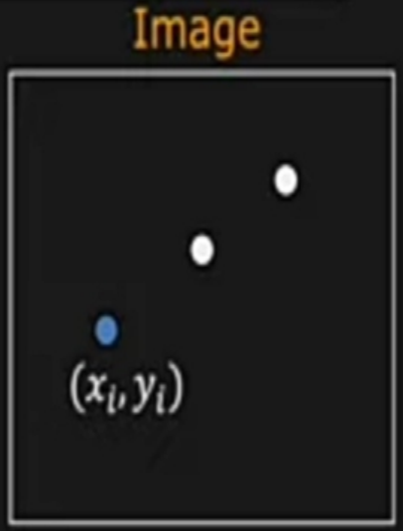
Step 2. Create accumulator array $A(m, c)$

Step 3. Set $A(m, c) = 0$ for all (m, c)

Step 4. For each edge point (x_i, y_i),

$$A(m, c) = A(m, c) + 1$$

if (m, c) lies on the line: $c = -m x_i + y_i$



$A(m, c)$

c

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Line Detection Algorithm

Step 1. Quantize parameter space (m, c)

Step 2. Create accumulator array $A(m, c)$

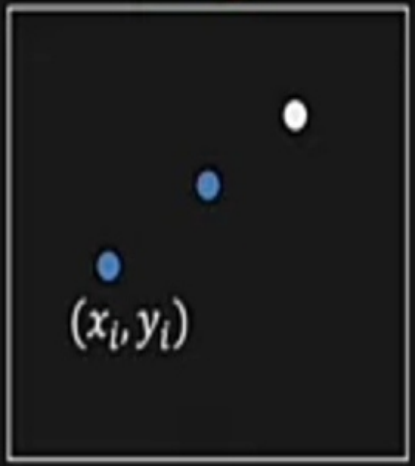
Step 3. Set $A(m, c) = 0$ for all (m, c)

Step 4. For each edge point (x_i, y_i),

$$A(m, c) = A(m, c) + 1$$

if (m, c) lies on the line: $c = -mx_i + y_i$

Image



$A(m, c)$

c

1	0	0	0	1
0	1	0	1	0
0	0	2	0	0
0	1	0	1	0
1	0	0	0	1

Line Detection Algorithm

Step 1. Quantize parameter space (m, c)

Step 2. Create accumulator array $A(m, c)$

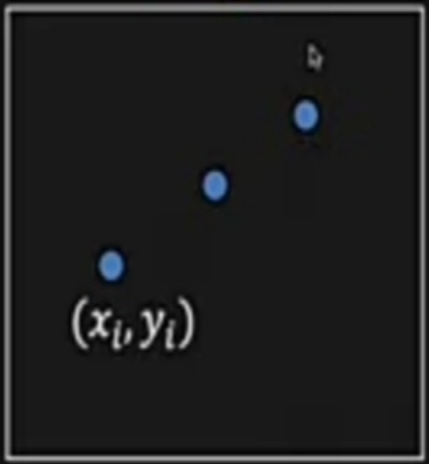
Step 3. Set $A(m, c) = 0$ for all (m, c)

Step 4. For each edge point (x_i, y_i),

$$A(m, c) = A(m, c) + 1$$

if (m, c) lies on the line: $c = -mx_i + y_i$

Image



$A(m, c)$

c

1	0	0	0	1
0	1	0	1	0
1	1	3	1	1
0	1	0	1	0
1	0	0	0	1

Multiple Line Detection

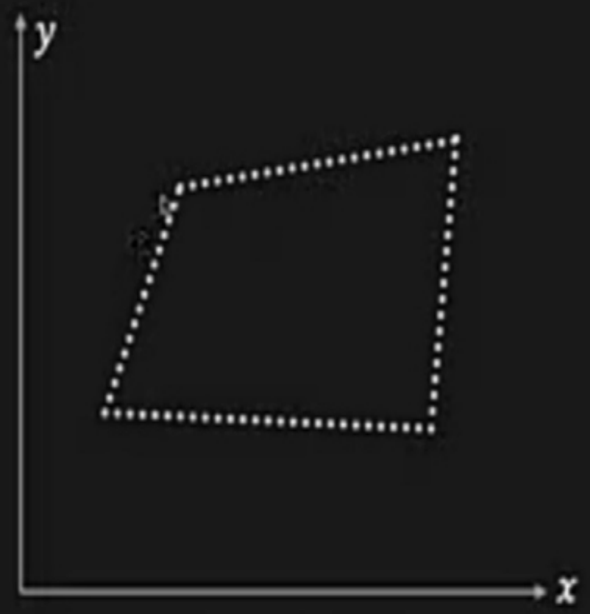
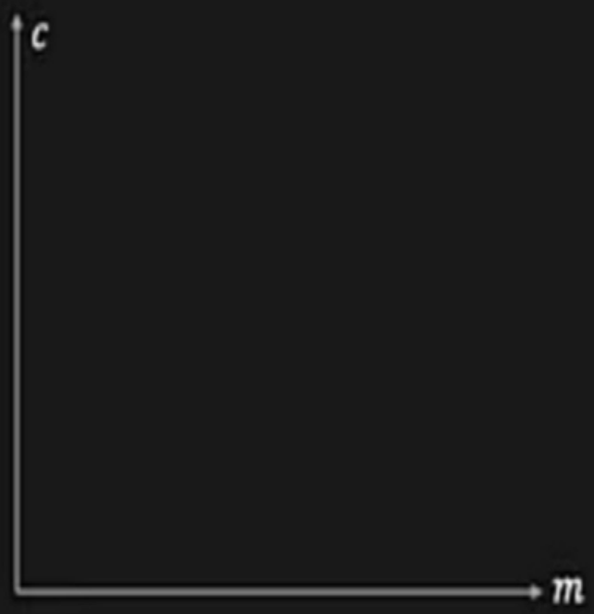


Image Space



Parameter Space

Multiple Line Detection

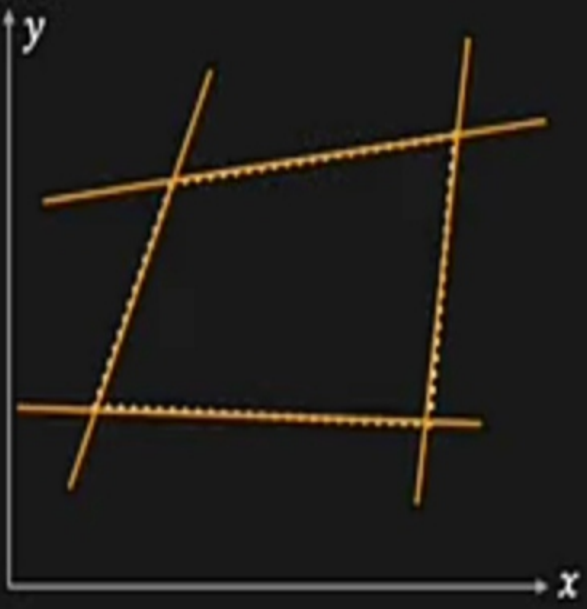
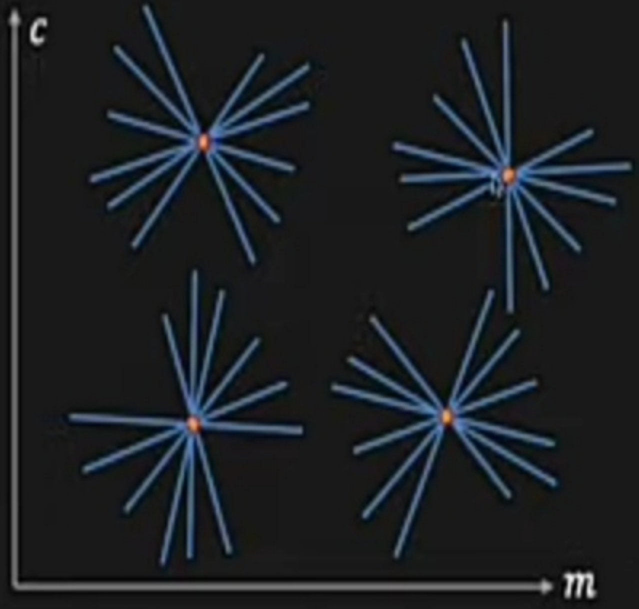


Image Space



Parameter Space

Better Parameterization

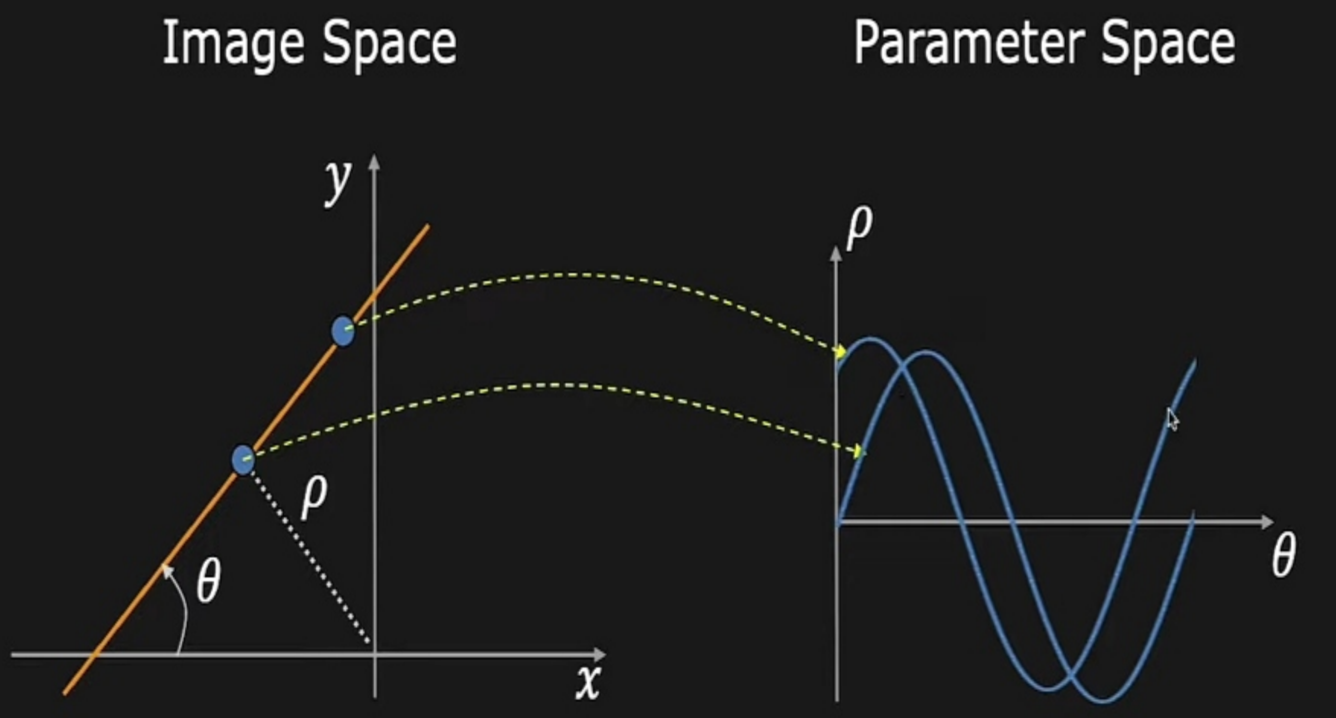
Issue: Slope of the line $-\infty \leq m \leq \infty$

- Large Accumulator
- More Memory and Computation

Solution: Use $x \sin\theta - y \cos\theta + \rho = 0$

- Orientation θ is finite: $0 \leq \theta < \pi$
- Distance ρ is finite

Better Parameterization



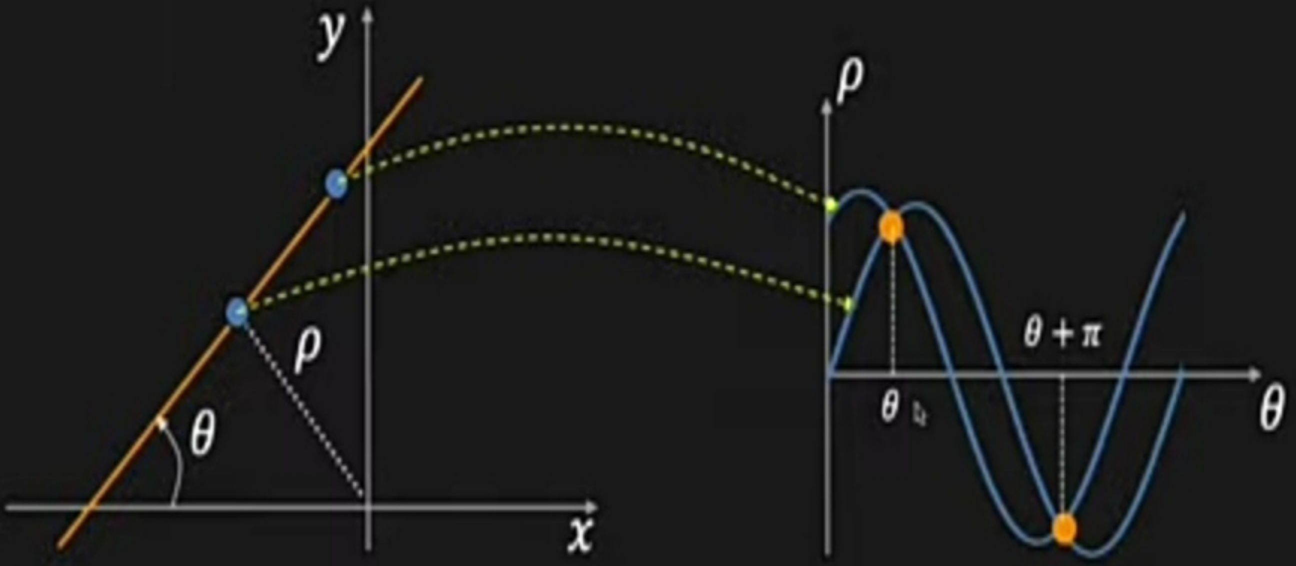
$$x \sin\theta - y\cos\theta + \rho = 0$$

$$x \sin\theta - y\cos\theta + \rho = 0$$

Better Parameterization

Image Space

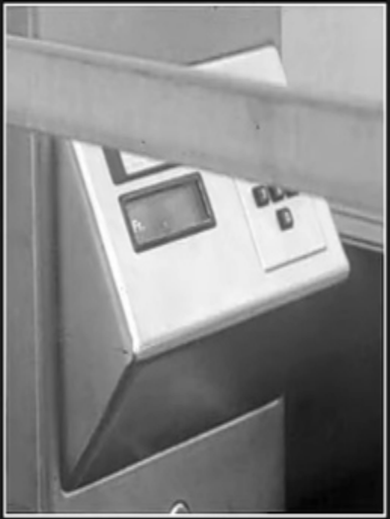
Parameter Space



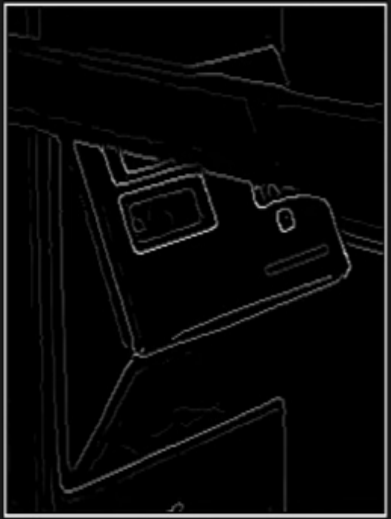
$x \sin\theta - y\cos\theta + \rho = 0$

$x \sin\theta - y\cos\theta + \rho = 0$

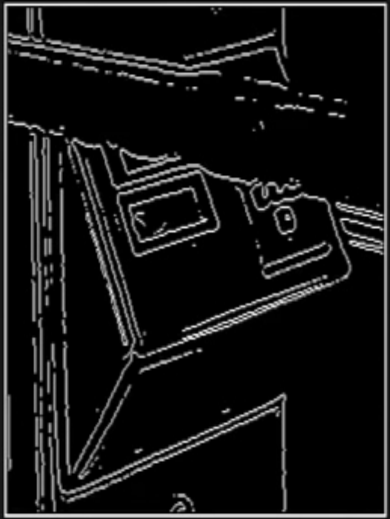
Line Detection Results



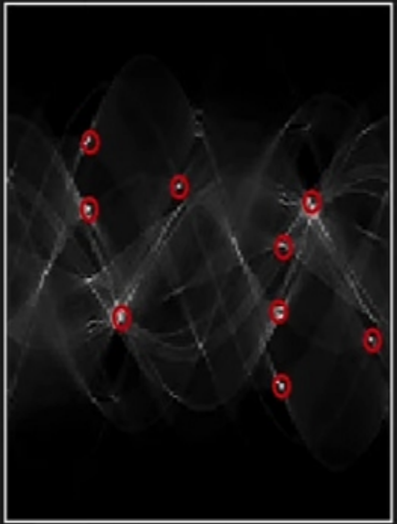
Original Image



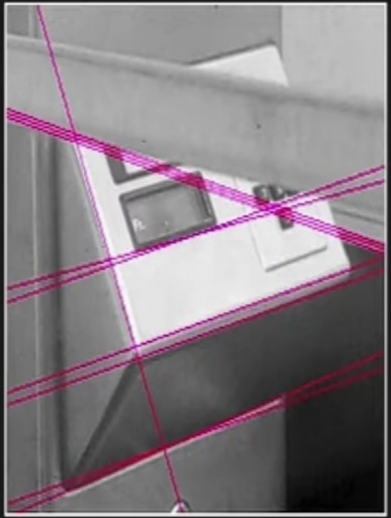
Gradient



Edge (Threshold)



Hough Transform $A(\rho,\theta)$

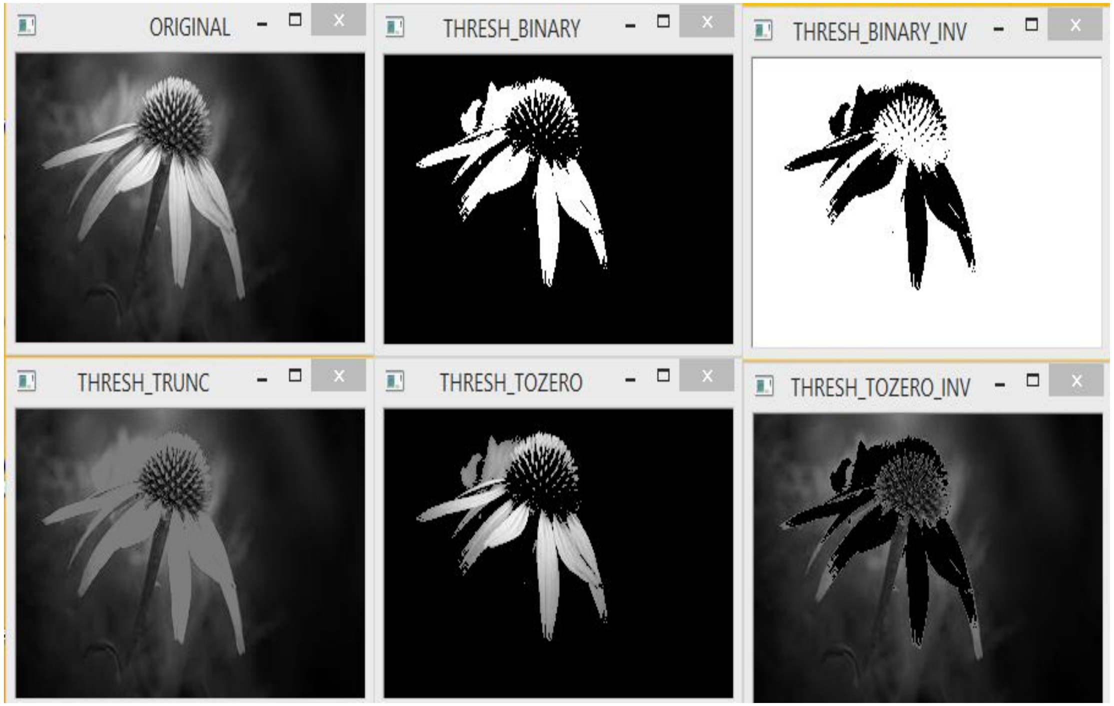


Detected Lines

THRESHOLDING

THRESHOLDING

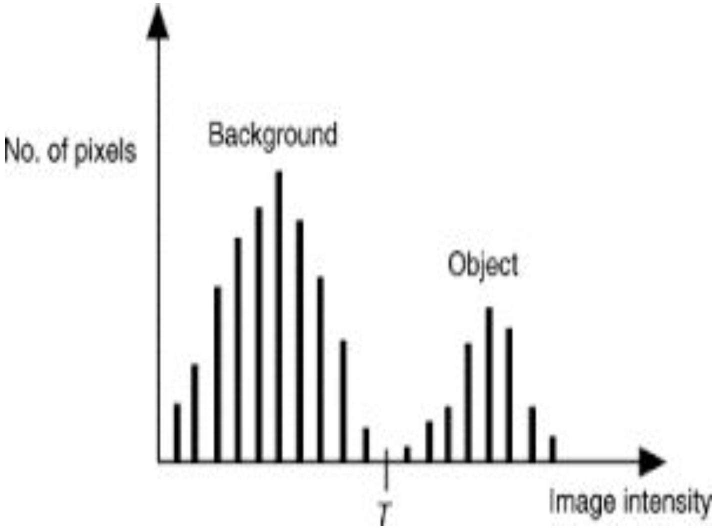
It is carried out with the assumption that the range of intensity level covered by object of interest is different from the background.



STEPS

I. A threshold T is selected
II. Any point (x,y) in the image at which f(x,y)> T is called an object point
III. The segmented image denoted by g(x,y) is given by

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



TYPES OF THRESHOLDING

1 - Global thresholding

where T is constant ,which is called global thresholding.

2- Variable thresholding

T changes over an image, In variable thresholding more than one T value are exist.

3- Local or Regional thresholding

In variable thresholding if the value of T at any point of (x,y) in an image depends on the properties of a neighbourhood of (x,y)

4- Dynamic or adoptive thresholding

In variable thresholding if the value of T depends upon the spatial coordinate of (x,y)

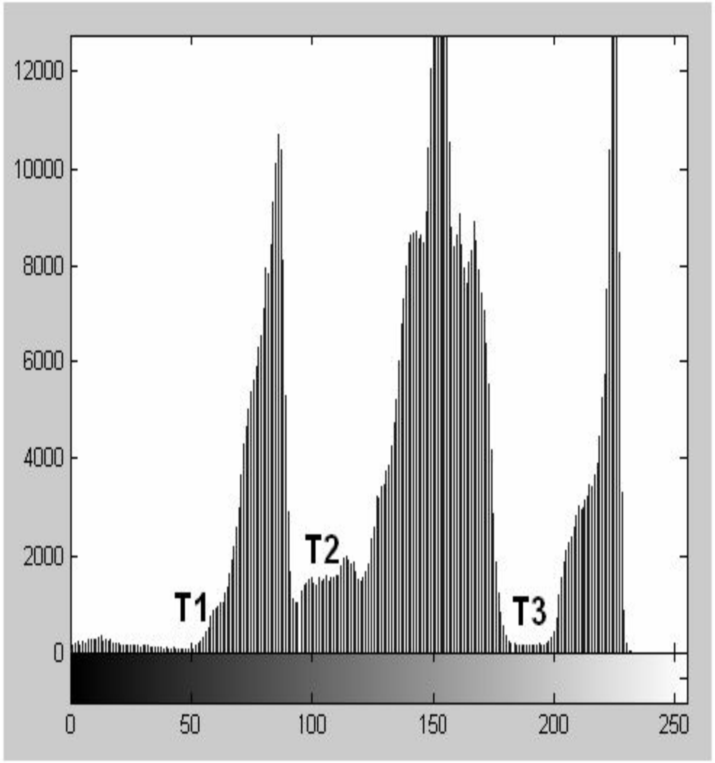


Figure 1. Threshold value multiples

III. Proposed Method

The proposed multi-level thresholding sytem

PROCEDURE FOR GLOBAL THRESHOLDING TO OBTAIN T

1. Select an initial estimated for T.(This value should be greater than the minimum and less then the maximum intensity level in the image .It is better to choose the average intensity of an image).
2. Segment the image using T. This will produce two group of pixel G1 consist of all pixel with grey level values >T and G2 consisting of all pixel with values<=T.
3. Compute the average grey level values $\mu 1$ and $\mu 2$ for the pixels in the region G1 and G2
4. Compute the average grey level values $\mu 1$ and $\mu 2$ for the pixels in the region G1 and G2
5. Compute a new threshold value

$$T = 1/2(\mu 1 + \mu 2)$$

6. Repeat step 2 through 4 until the difference in T in successive iterations is smaller then a predefine parameter T .

Example-

5	3	9
6	2	7
8	4	2

Let T=(5+3+9+2+1+7+8+4+2)/9

T=41/9

=4.55 equivalent to 5

Segmenting the image using T we would get

G1={9,7,8} G2={5,3,2,1,4,2}

μ1=(9+7+8)/3

μ2=(5+3+2+1+4+2)/6

=8 =2.83 equivalent to 3

T=1/2 (8+3)

=5.5 equivalent to 5

Region and Edge Based Segmentation

Segmentation

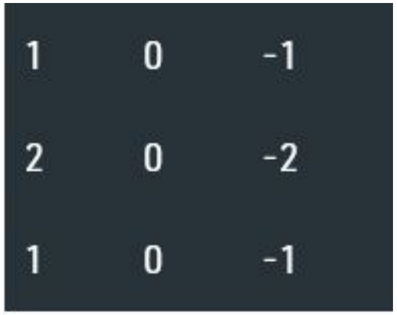
Segmentation is the separation of one or more regions or objects in an image based on a discontinuity or a similarity criterion. A region in an image can be defined by its border (edge) or its interior, and the two representations are equal. There are prominently three methods of performing segmentation:

- Pixel Based Segmentation
- Region-Based Segmentation
- Edges based segmentation

Edges based segmentation

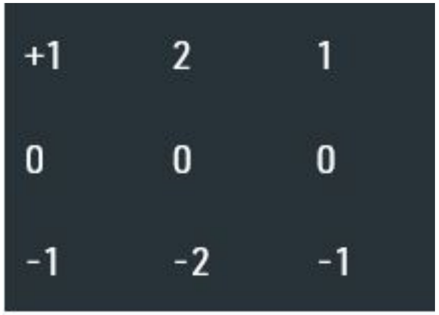
Edge-based segmentation contains 2 steps:

Edge Detection: In edge detection, we need to find the pixels that are edge pixels of an object. There are many object detection methods such as Sobel operator, Laplace operator, Canny, etc.



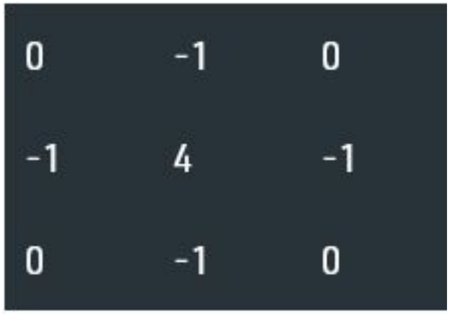
1	0	-1
2	0	-2
1	0	-1

Sobel vertical Operator



+1	2	1
0	0	0
-1	-2	-1

Sobel Horizontal Operator



0	-1	0
-1	4	-1
0	-1	0

Negative Laplace Operator

Edge Linking:

In this step, we try to refine the edge detection by linking the adjacent edges and combine to form the whole object. The edge linking can be performed using any of the two methods below:

- Local Processing: In this method, we used gradient and direction to link the neighborhood edges. If two edges have a similar direction vector then they can be linked.
- Global processing: This method can be done using HOG transformation

Pros :

- This approach is similar to how the humans brain approaches the segmentation task.
- Works well in images with good contrast between object and background.

Limitations:

- Does not work well on images with smooth transitions and low contrast.
- Sensitive to noise.
- Robust edge linking is not trivial and easy to perform.

Region-Based Segmentation

The objective of segmentation is to partition an image into regions. We approached this problem by attempting to find boundaries between regions based on discontinuities in intensity levels, whereas in pixel based segmentation was accomplished via thresholds based on the distribution of pixel properties, such as intensity values or color. There are two variants of region-based segmentation:

- **Top-down approach**
  – First, we need to define the predefined seed pixel. Either we can define all pixels as seed pixels or randomly chosen pixels. Grow regions until all pixels in the image belongs to the region.
- **Bottom-Up approach**
  – Select seed only from objects of interest. Grow regions only if the similarity criterion is fulfilled.

Segmentation techniques that are based on finding the regions directly:-

1. Region Growing is a procedure that groups pixels or sub-regions into larger regions based on predefined criteria for growth. 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 predefined properties similar to the seed (such as specific ranges of intensity or color).

   Selecting a set of one or more starting points often can be based on the nature of the problem. 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.

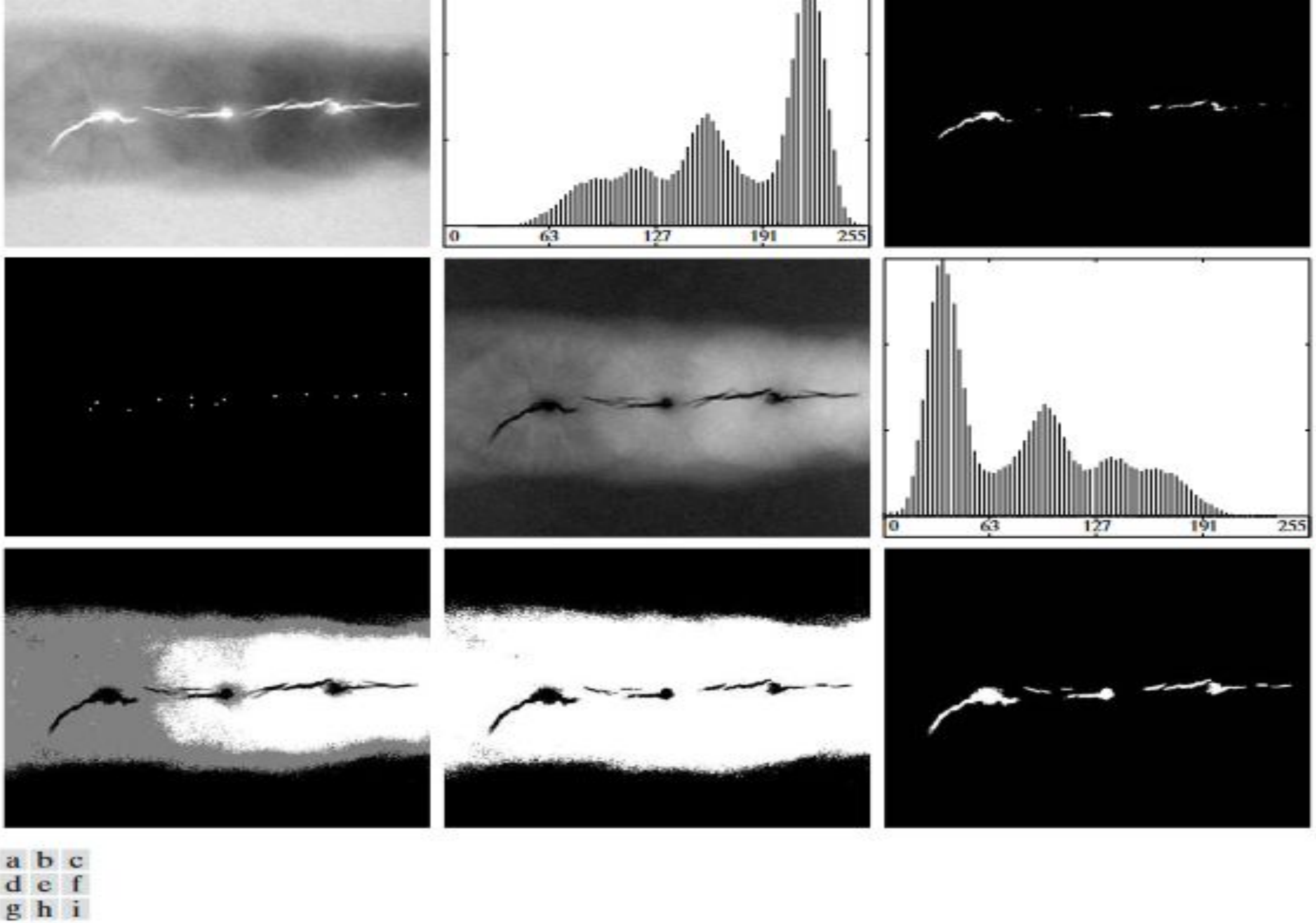
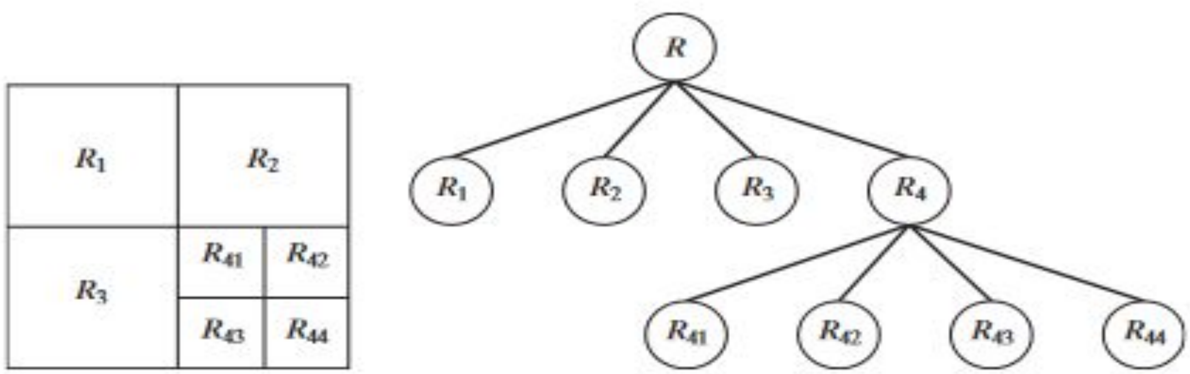


FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference 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 Splitting and Merging

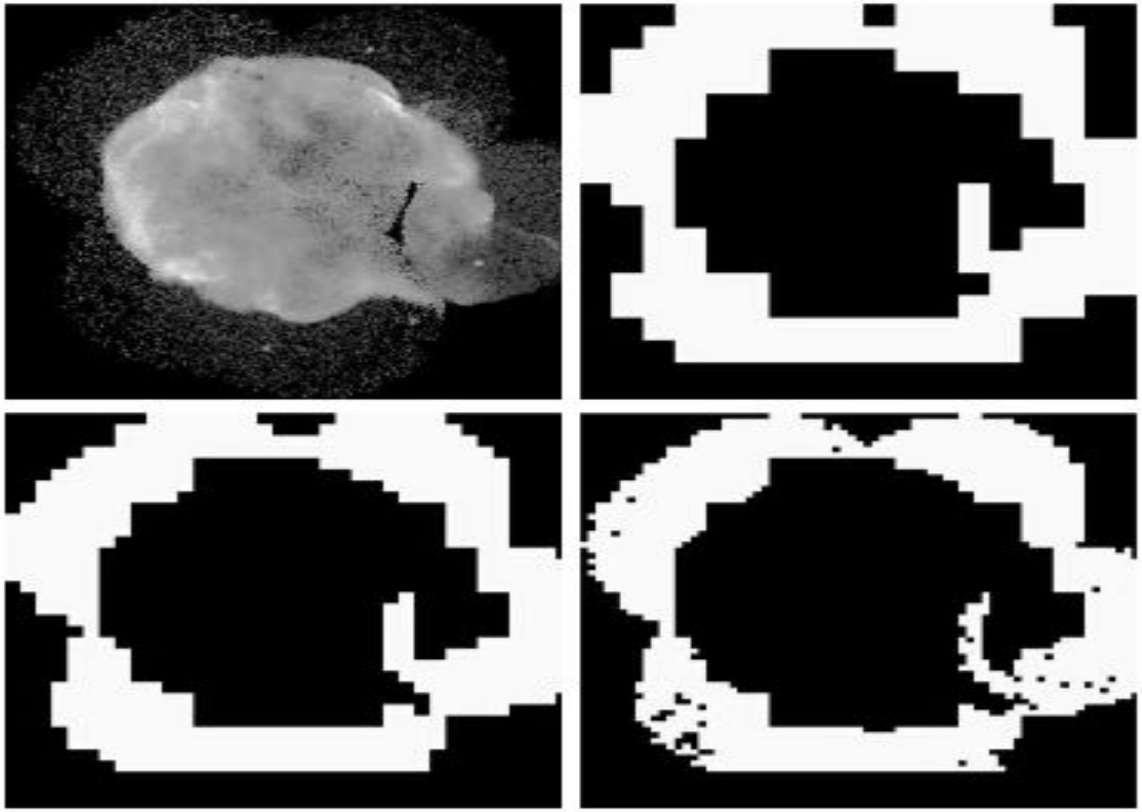
The procedure discussed in the last section grows regions from a set of seed points. An alternative is to subdivide an image initially into a set of arbitrary, disjoint regions and then merge and/or split the regions in an attempt to satisfy the conditions of segmentation.



a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. *R* represents the entire image region.

Figure 10.53(a) shows a X-ray band image of the Cygnus Loop. The objective of this example is to segment out of the image the “ring” of less dense matter surrounding the dense center. The region of interest has some obvious characteristics that should help in its segmentation.



a b
c d

FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA’s Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32 × 32, 16 × 16, and 8 × 8 pixels, respectively. (Original image courtesy of NASA.)

- **Similarity Measures:** Similarity measures can be of different types: For the grayscale image the similarity measure can be the different textures and other spatial properties, intensity difference within a region or the distance b/w mean value of the region.
- **Region merging techniques:** In the region merging technique, we try to combine the regions that contain the single object and separate it from the background.. There are many regions merging techniques such as Watershed algorithm, Split and merge algorithm, etc**.**
- **Pros:** Since it performs simple threshold calculation, it is faster to perform.
- Region-based segmentation works better when the object and background have high contrast.
- **Limitations:** It did not produce many accurate segmentation results when there are no significant differences b/w pixel values of the object and the background.

INTRODUCTION TO IMAGE MORPHOLOGY

- Image morphology is a branch of science which deals with form and structure of images.
- Morphological image processing or morphology describes a range of image processing techniques that deal with the shape(or morphology) of features in an image.

Uses of Morphology

- Morphological image processing is used to extract image components for representation and description of region shape, such as boundaries, skeletons, and the convex hull.
- Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on bi-level images.

Operations of Morphology

Two basic operations-

- Erosion
- Dilation

Dilation: Adds pixels to the boundaries of objects in an image.

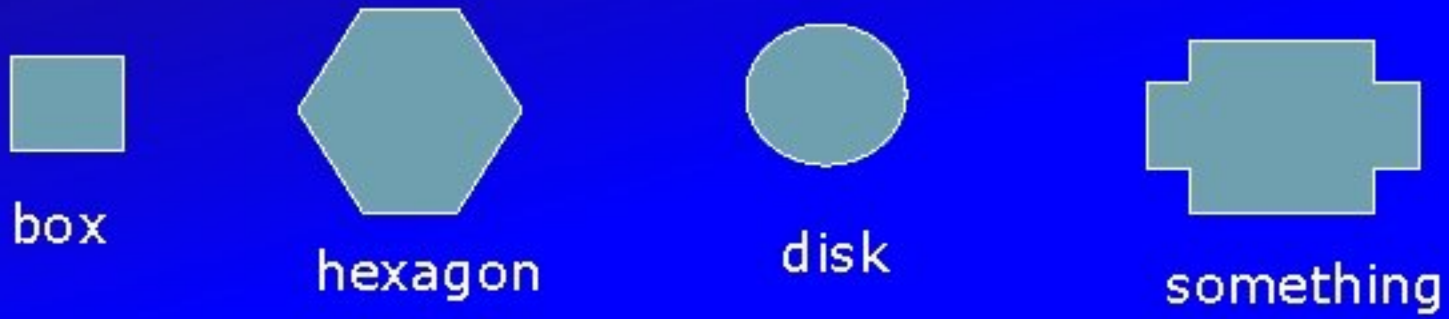
Erosion: Removes pixels on object boundaries.

Structuring element- The no of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image.

Structuring Elements

A **structuring element** is a shape mask used in the basic morphological operations.

They can be any shape and size that is digitally representable, and each has an **origin**.

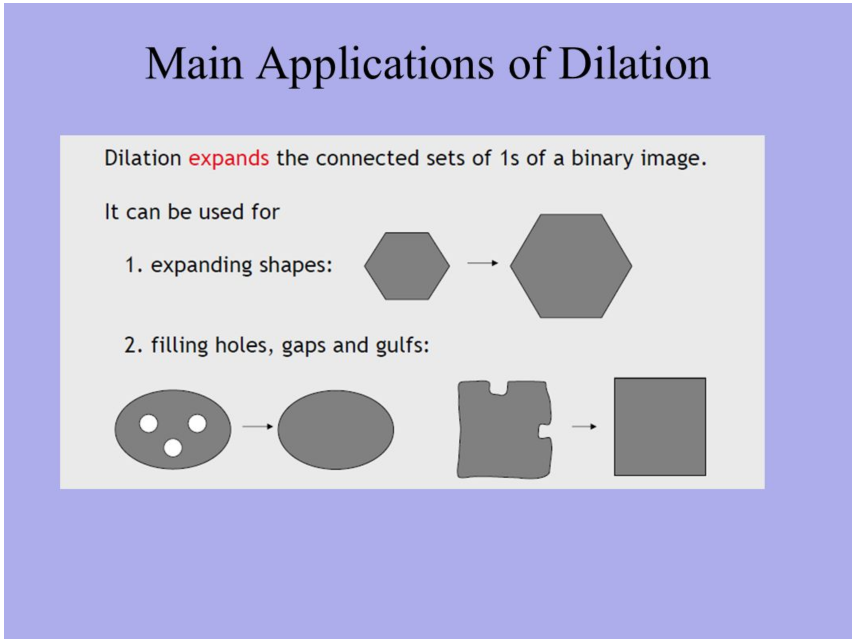


box(length,width)

disk(diameter)

Dilation

- Fills in holes.
- Smoothes object boundaries.
- Adds an extra outer ring of pixels onto object boundary, i.e., object becomes slightly larger.



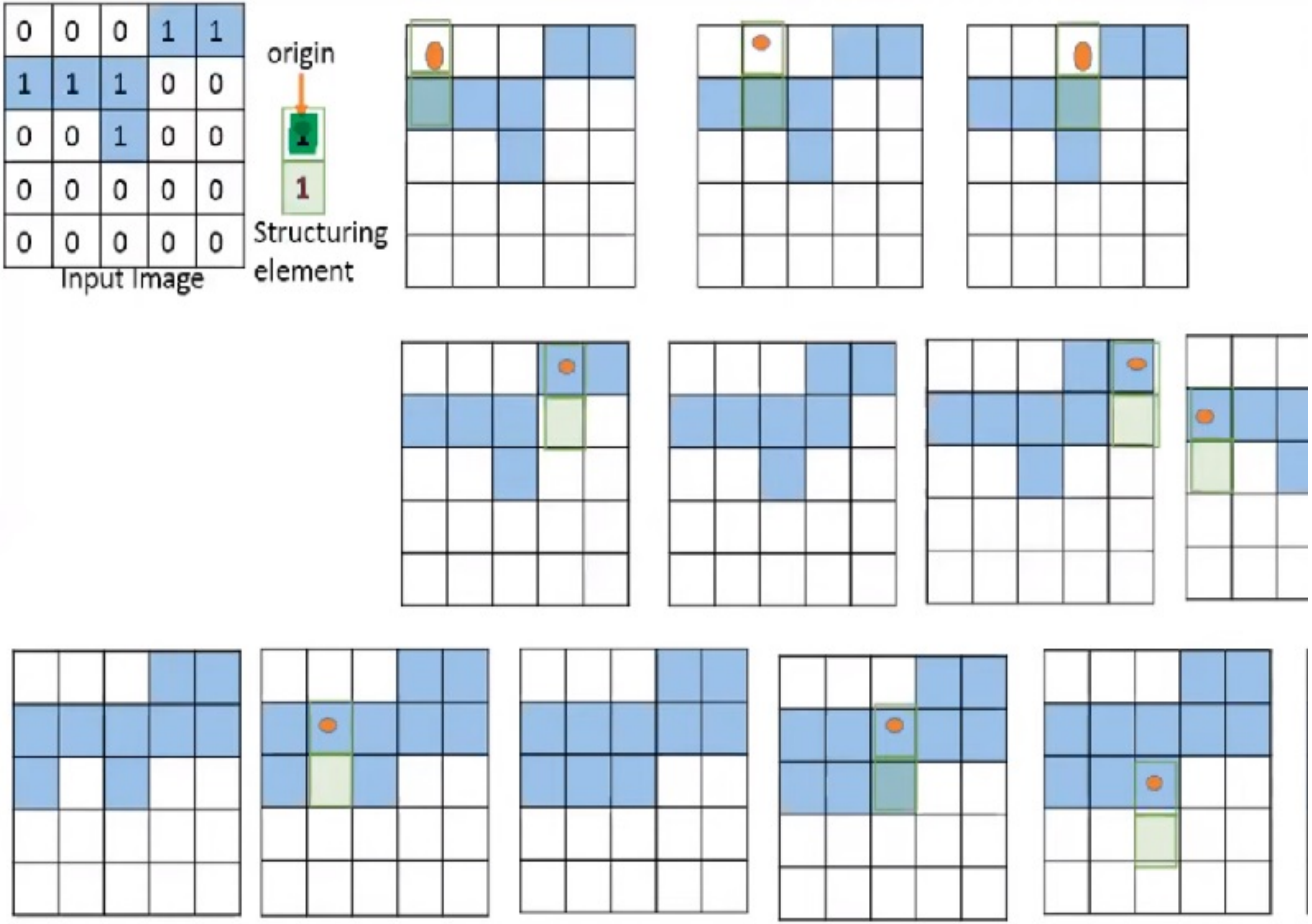
Main Applications of Dilation

Dilation expands the connected sets of 1s of a binary image.

It can be used for

1. expanding shapes:

2. filling holes, gaps and gulfs:

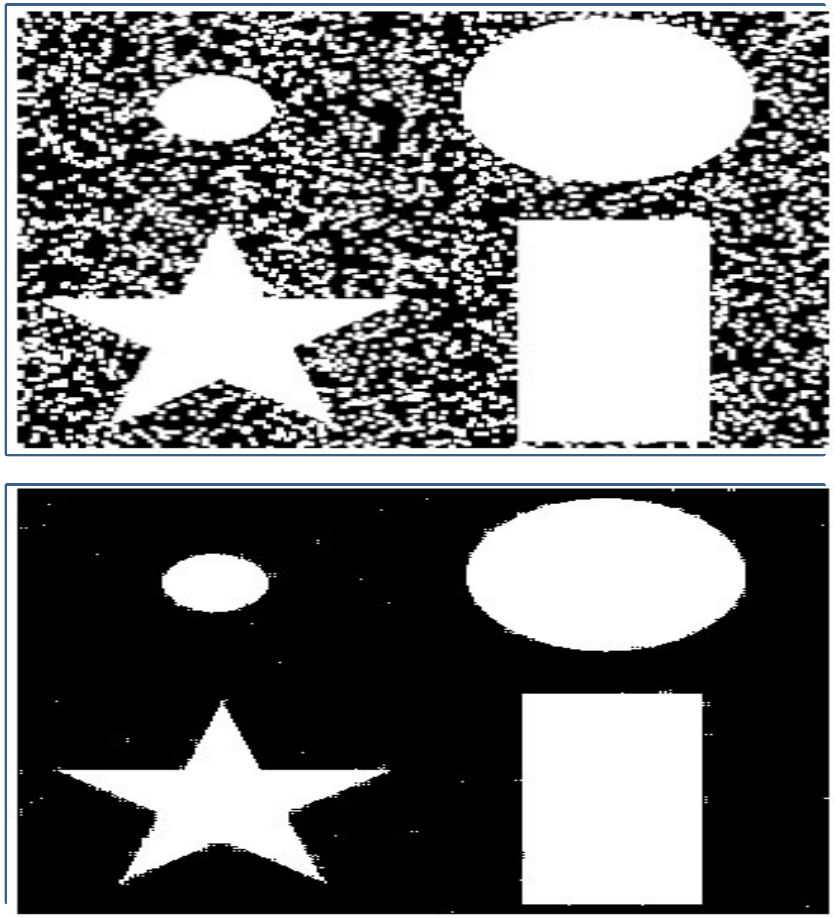


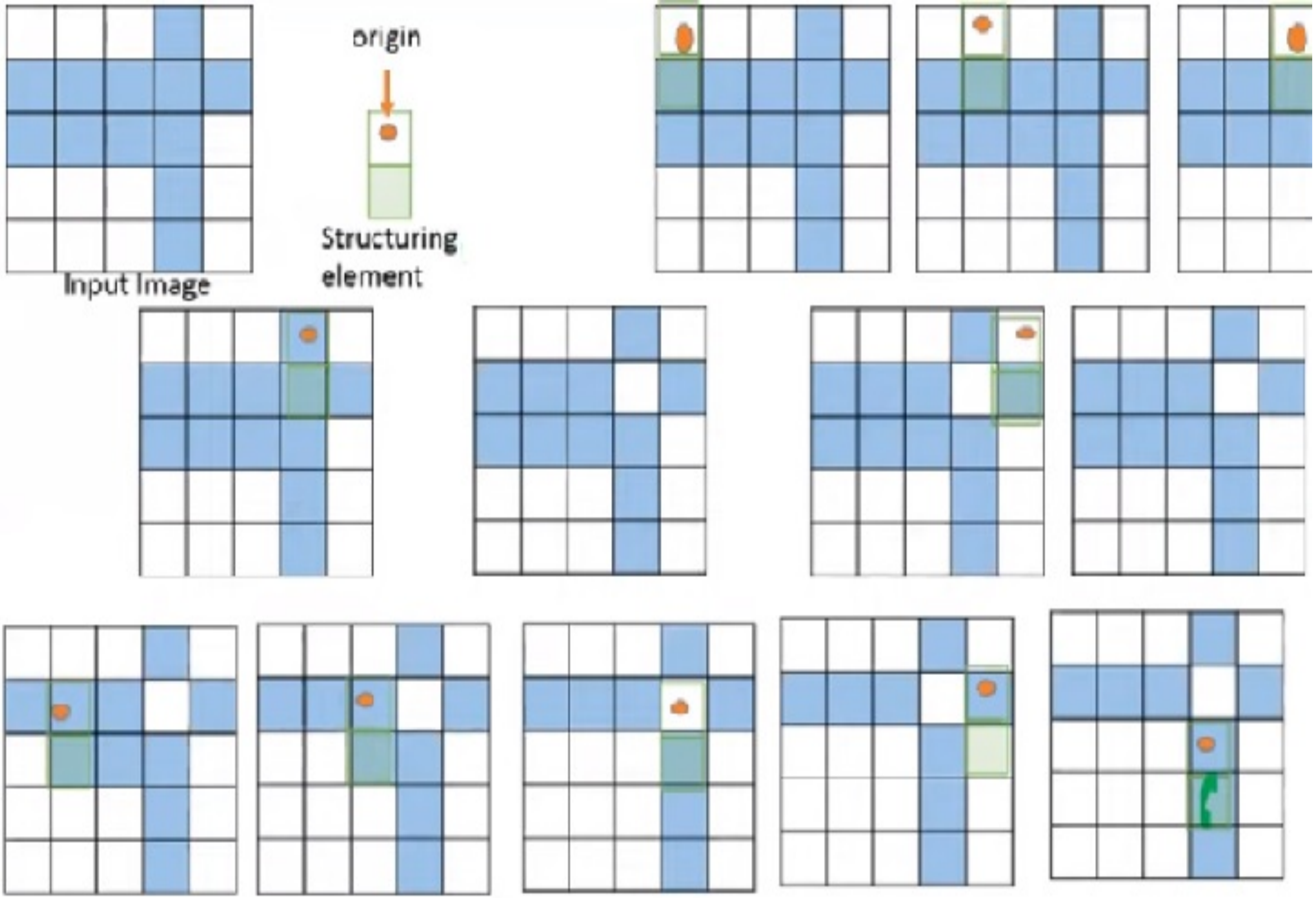
0	0	0	1	1
1	1	1	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input Image

Erosion

- Removes isolated noisy pixels.
- Smoothes object boundary.
- Removes the outer layer of the object pixels, i.e., object becomes slightly smaller.




origin
Structuring element
Input Image

Opening and Closing Process

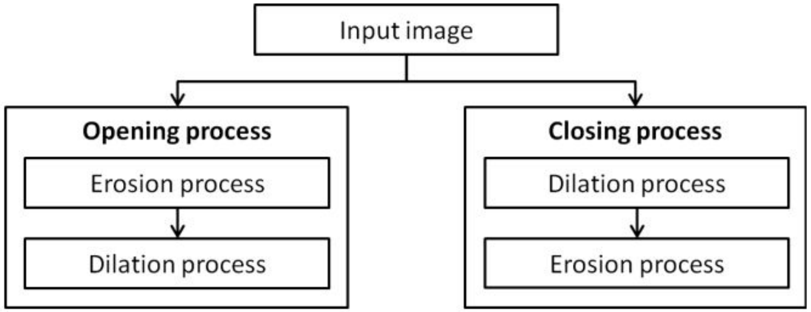


Figure 2: Flowchart of the opening and closing processes

- Opening: It serves to eliminate noise.
- Closing: It serves to close up cracks in the object and holes due to pepper noise

References: 1. DIP by Gozalez & Woods
2. Other websites