DIGITAL IMAGE PROCESSING MODULE-IV

PARVATHY LAKSHMY

FACULTY

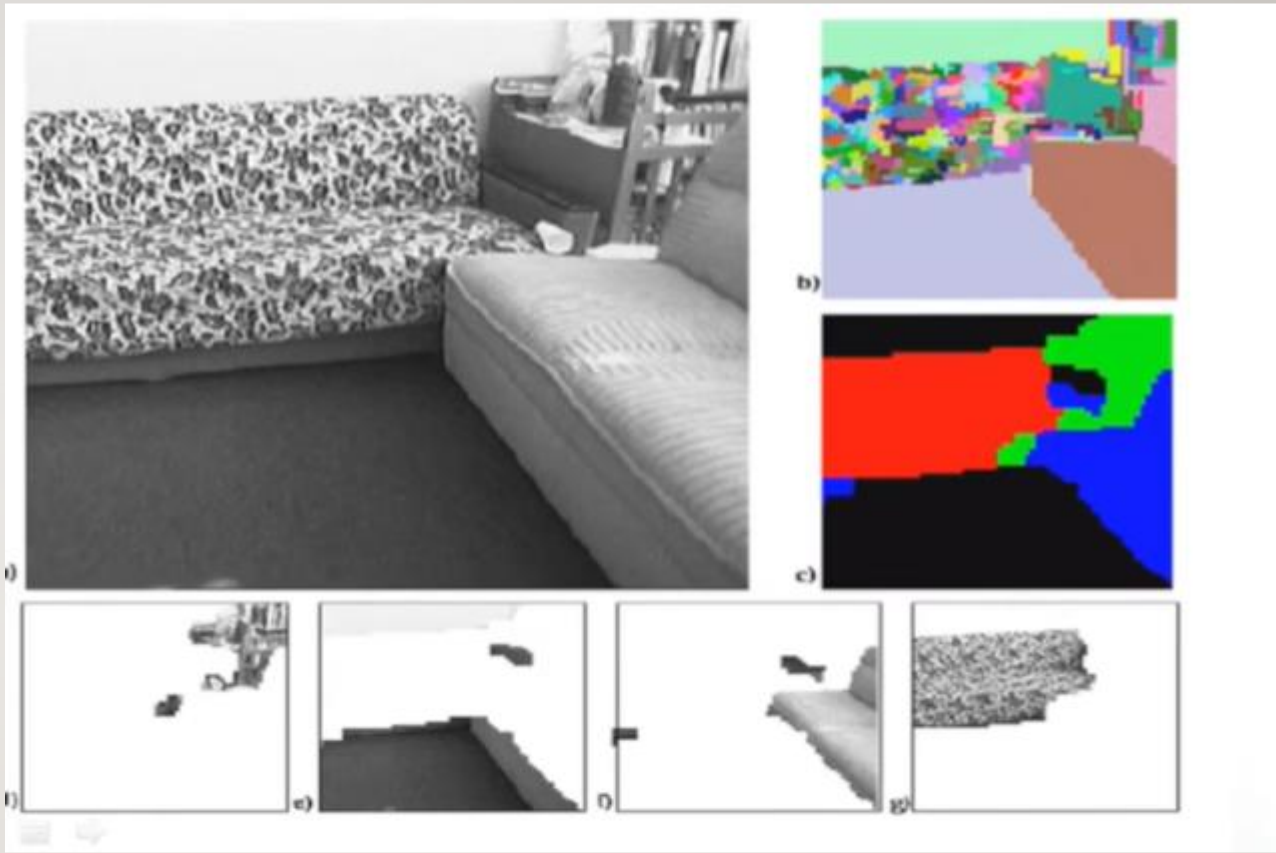
ELECTRICAL DEPARTMENT

VJTI

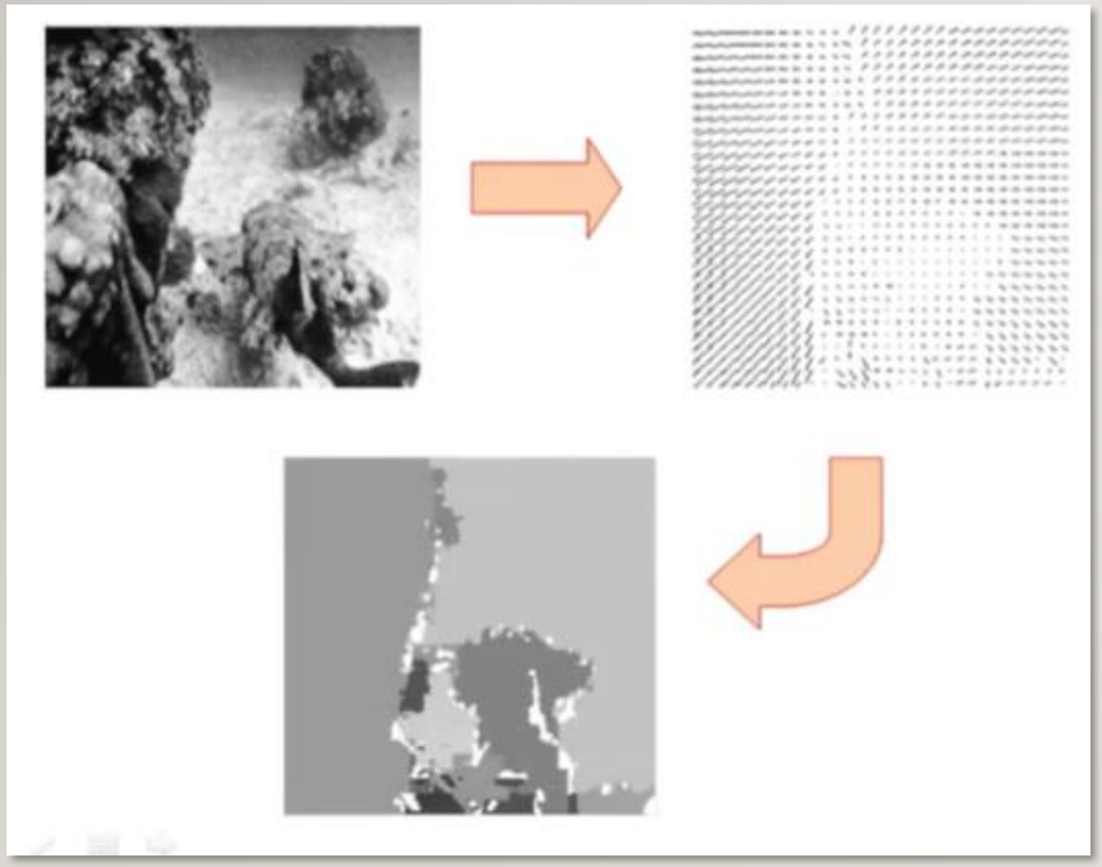
IMAGE SEGMENTATION

- Transition from:
- ➢ Image processing methods whose input and output are images to
- ➢ Image processing methods in which inputs are images but outputs are attributes extracted from those images.
- Segmentation subdivides an image into its constituent regions or objects

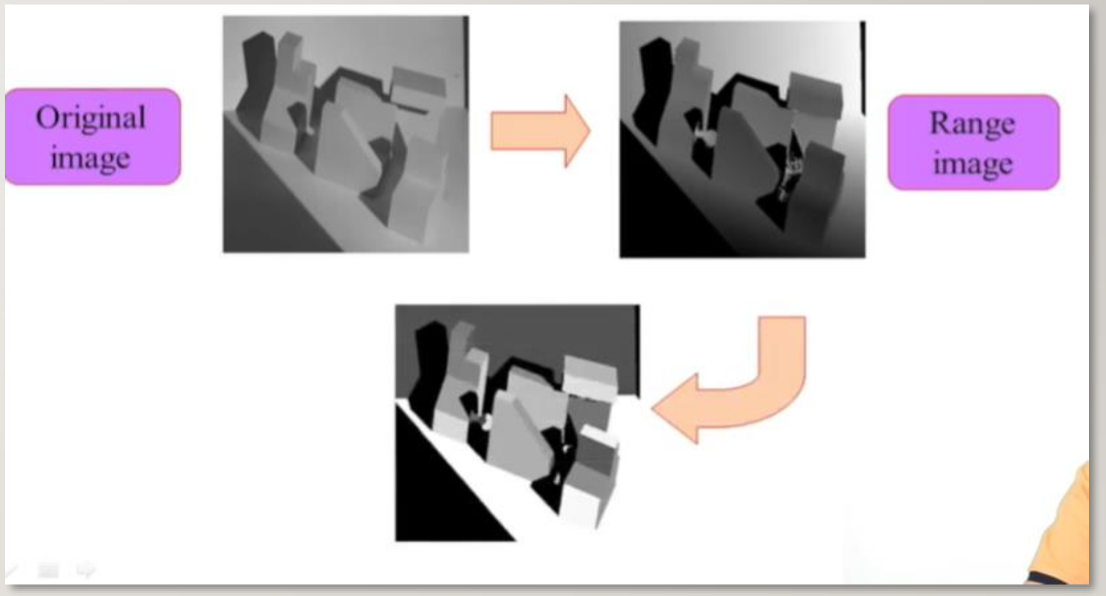
Segmentaion based on grayscale



Segmentation based on texture



Segmentaion based on motion



Segmentaion based on depth

IMAGE SEGMENTATION

- The level to which subdivision is carried depends on the problem being solved
- Segmentation should stop when objects of interest in application is isolated

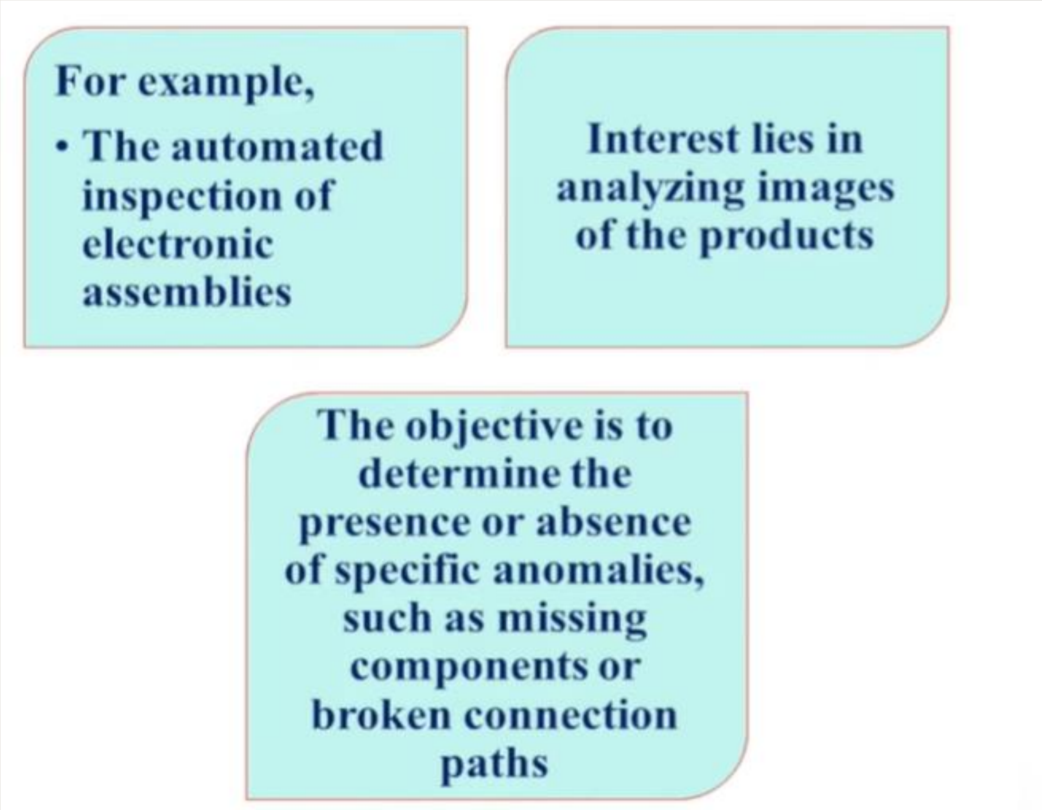
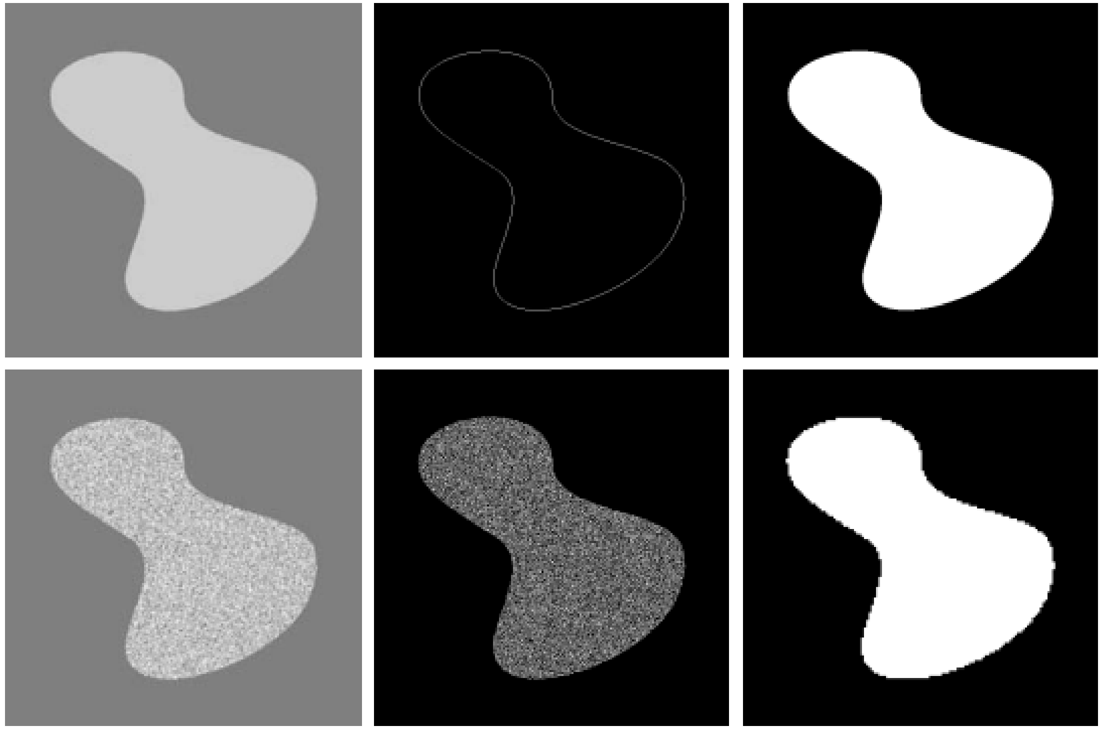


IMAGE SEGMENTATION

- There is no point in carrying segmentation past the level of detail required to identify the elements.
- Segmentation of non trivial images is one of the most difficult tasks in image processing
- Segmentation accuracy determines the eventual success or failure of computerized analysis procedures.
- Image segmentation algorithms generally are based on one of the two basic properties of intensity values-**Discontinuity and Similarity**

IMAGE SEGMENTAION

- **Discontinuity**-Approach is to partition an image based on abrupt changes in intensity such as edges in an image
- **Similarity-**Partitioning an image into regions that are similar according to a set of predefined criteria
- Eg: Thesholding, Region Growing, Region Splitting, Merging



a b c
d e f

FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

IMAGE SEGMENTATION

- Let R represent the entire spatial region occupied by an image.
- We may view image segmentation as a process that partitions R into n subregions, such that

➢ $\bigcup_{i=1}^{n} R_i = \mathrm{R}$

➢ R_i is a connected set i=1,2,3…..n

➢ $R_i \cap R_j = \emptyset$ for all i and j,i $\neq j$

➢ $Q(R_i\ )$=True for i=1,2…….n

➢ $Q(R_i \cup R_j)$ =False for any adjacent regions R_i and R_j

IMAGE SEGEMENTATION

- Condition (a) indicates that the segmentation must be complete; that is every pixel must be in a region.
- Condition (b) requires that points in a region be connected in some predefined sense (e.g., the points must be 4- or 8-connected.
- Condition (c) indicates that the regions must be disjoint.
- Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region—for example, $Q(R_i)$=True if all pixels in have the same intensity level.
- Finally, condition (e) indicates that two adjacent regions Ri and Rj must be different in the sense of predicate Q.

POINT LINE AND EDGE DETECTION

- The focus of this section is on segmentation methods that are based on detecting sharp, *local* changes in intensity.
- The three types of image features in which we are interested are isolated **points**, **lines**, and **edges**.
- ***Edge pixels*** are pixels at which the intensity of an image function changes abruptly, and *edges* (or *edge*
- *segments*) are sets of connected edge pixels.
- ***Edge detectors*** are local image processing methods designed to detect edge pixels.
- A line may be viewed as an edge segment in which the intensity of the background on either side of the line is either much higher or much lower than the intensity of the line pixels.

POINT LINE AND EDGE DETECTION

- Local averaging smooths an image.
- Given that averaging is analogous to integration, local changes in intensity can be detected using derivatives.
- First- and second-order derivatives are well suited for this purpose.
- Derivatives of a digital function are defined in terms of differences.

POINT LINE AND EDGE DETECTION

- There are various ways to approximate these differences, but we require that any approximation used for a first derivative

➢ Must be zero in areas of constant intensity

➢ Must be nonzero at the onset of an intensity step or ramp

➢ Must be nonzero at points along an intensity ramp

POINT LINE AND EDGE DETECTION

- Similarly, we require that an approximation used for a second derivative

➢ Must be zero in areas of constant intensity

➢ Must be nonzero at the onset and end of an intensity step or ramp

➢ Must be zero along intensity ramps.

POINT LINE AND EDGE DETECTION

- We obtain an approximation to the first-order derivative at point x of a one-dimensional function f(x) by expanding the function f(x+Δx) into a Taylor series about x letting $\Delta x = 1$ and keeping only the linear terms.
- The result is the digital difference

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

POINT LINE AND EDGE DETECTION

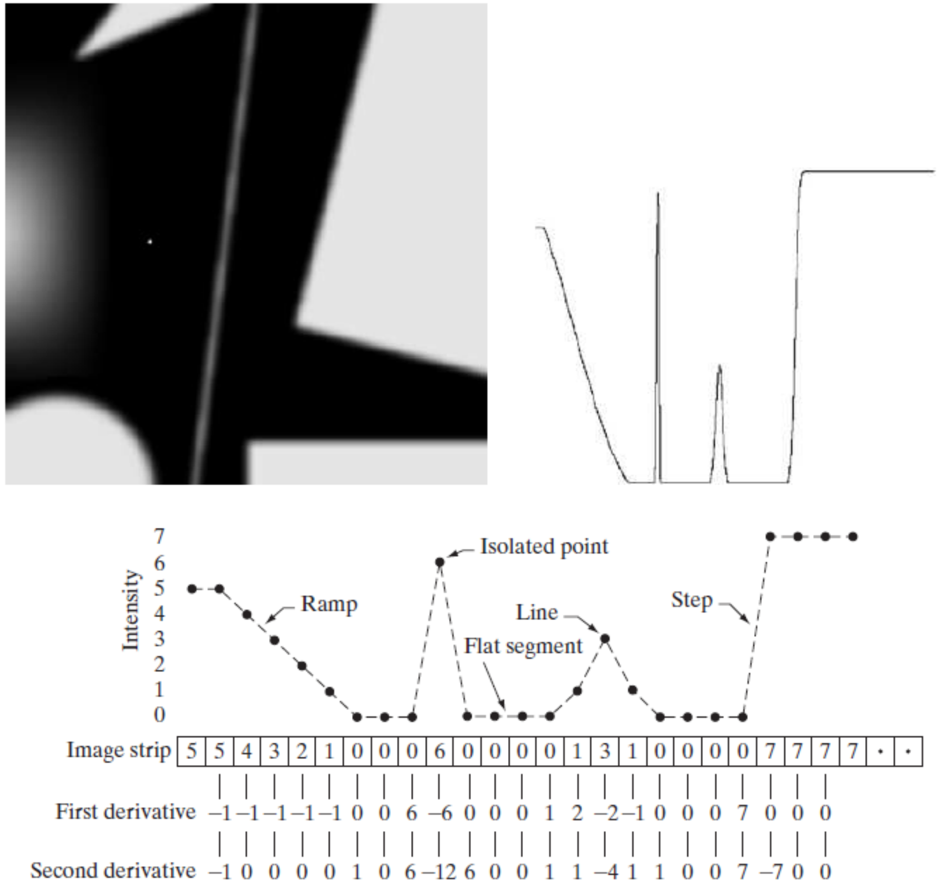
- We obtain an expression for the second derivative by differentiating the above equation with respect to x

$$
\begin{aligned}
\frac{\partial^2 f}{\partial x^2} = \frac{\partial f'(x)}{\partial x} &= f'(x+1) - f'(x) \\
&= f(x+2) - f(x+1) - f(x+1) + f(x) \\
&= f(x+2) - 2f(x+1) + f(x)
\end{aligned}
$$

POINT LINE AND EDGE DETECTION

- This expansion is about point x+1.
- Our interest is on the second derivative about point x, so we subtract 1 from the arguments in the preceding expression and obtain the result

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x+1) + f(x-1) - 2f(x)$$



a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

POINT LINE AND EDGE DETECTION

- Initially, we note that the first-order derivative is nonzero at the onset and along the entire intensity ramp, while the second order derivative is nonzero only at the onset and end of the ramp.
- Because edges of digital images resemble this type of transition; we conclude that first order derivatives produce "thick" edges and second-order derivatives much finer ones.
- Next, we encounter the isolated noise point. Here, the magnitude of the response at the point is much stronger for the second than for the first-orderderivative.

POINT LINE AND EDGE DETECTION

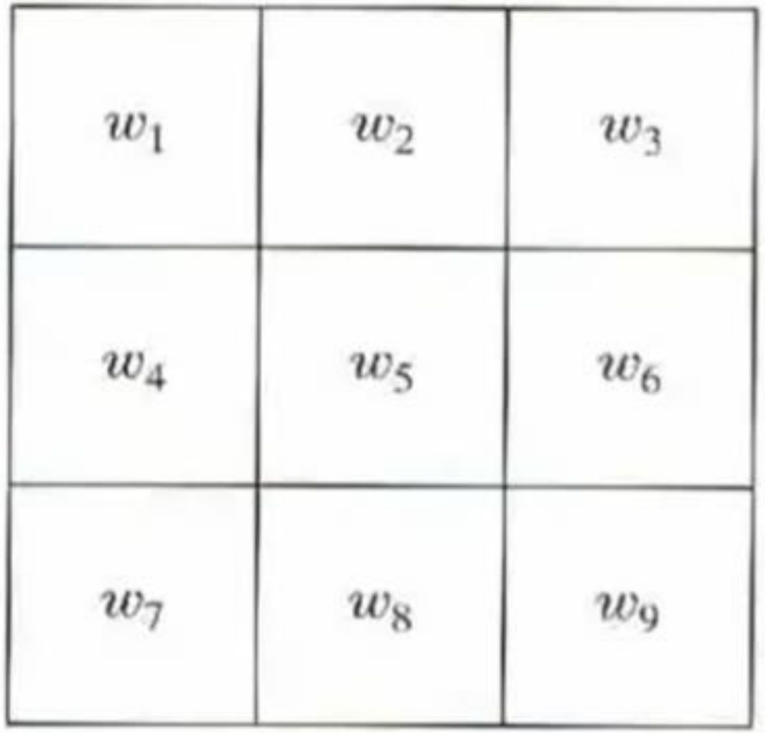
- A Second-order derivative is much more aggressive than a first-order derivative in enhancing sharp changes.
- Thus, we can expect second-order derivatives to enhance fine detail (including noise) much more than first-order derivatives
- Finally, note in both the ramp and step edges that the second derivative has opposite signs (negative to positive or positive to negative) as it transitions into and out of an edge.
- This "double-edge" effect is an important characteristic that can be used to locate edges.
- The sign of the second derivative is used also to determine whether an edge is a transition from light to dark (negative second derivative) or from dark to light (positive second derivative), where the sign is observed as we move *into* the edge.

POINT LINE AND EDGE DETECTION

- In summary, we arrive at the following conclusions: (1) First-order derivatives generally, produce thicker edges in an image.
- Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise.
- Second-order derivatives produce a double-edge response at ramp and step transitions in intensity.
- The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light.

DETECTION OF ISOLATED POINTS

- Techniques for detecting the three basic types of gray level discontinuities
- ➢ Points
- ➢ Lines
- ➢ Edges
- Most common way to look for discontinuities is to run a mask through the image



A general 3 X 3 mask

DETECTION OF ISOLATED POINTS

- For 3X3 mask shown, procedure involves computing the sum of products of the coefficients with the gray levels contained in the region encompassed by the mask
- The response of the mask at any point in the image is given by

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

- where Z_i is the gray level of the pixel associated with coefficient W_i
- Response of the mask is defined with respect to the centre location

POINT DETECTION

- Using the mask shown in previous figure, a point has been detected at the location on which the mask is centred if

$$|R| \geq T$$

- Where T is a non negative threshold and R is given by the above equation.
- This formulation measures the weighted differences between the centre point and its neighbours.
- Idea-Isolated point will be different from its surroundings and be easily detectable by this mask

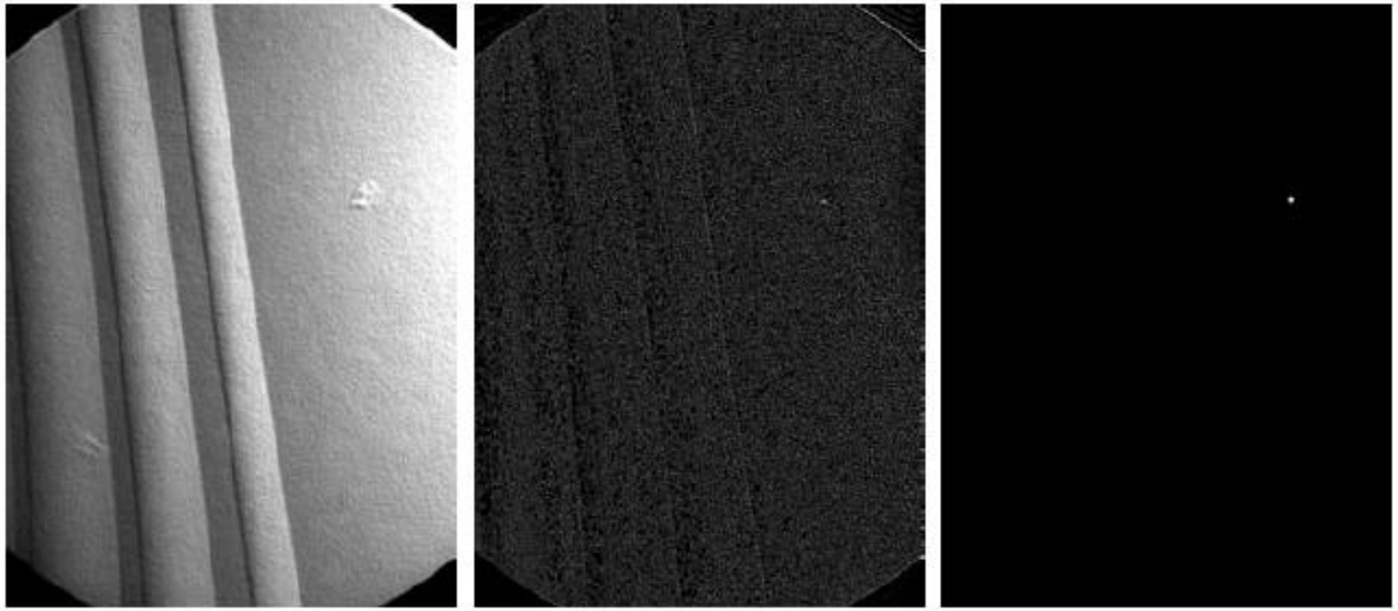
POINT DETECTION

- Emphasis here is strictly on the detection of points
- Only differences that are considered of interest are those large enough(as determined by T)to be considered isolated points.
- Mask coefficients sum to zero indicating the mask response will be zero in areas of constant Gray level

−1	−1	−1
−1	8	−1
−1	−1	−1

(a) A point detection mask

1	1	1
1	−8	1
1	1	1



a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Consider the masks shown in figure

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

Line Masks

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

−45°

Line Masks

LINE DETECTION

LINE DETECTION

- If the first mask is moved around an image, it will respond more strongly to lines(one pixel thick) oriented horizontally
- With a constant background, the maximum response will result when the line is passed through the middle of the mask
- The second mask responds best to lines oriented at $+45^0$
- The third mask responds best to vertical lines.
- The fourth mask responds to lines in -45^0 direction.

LINE DETECTION

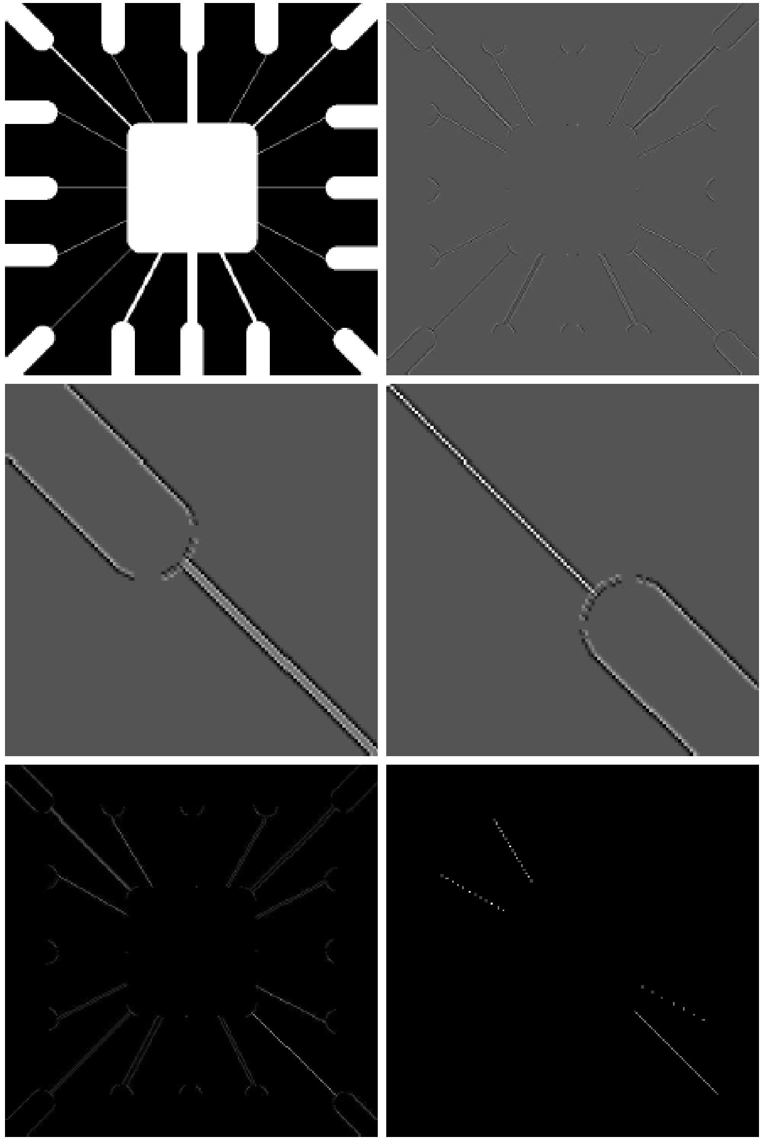
- Preferred direction of each mask is weighted with a large coefficient(i.e. 2) than other possible directions.
- Coefficients in each mask sum to zero indicating a zero response from masks in areas of constant Gray level.
- Response of the mask at any point in the image is given by:

$$R = w_1 z_1 + w_2 z_2 + \cdots \ldots \ldots \ldots w_9 z_9 = \sum_{i=1}^{9} w_i \, z_i$$

- Let R_1, R_2, R_3 and R_4 denote responses of the masks.

LINE DETECTION

- If we are interested in detecting lines in a specified direction, we would use the mask associated with that direction and threshold its output
- If we are interested in detecting all lines in an image in the direction defined by the given mask, we simply run the mask through the image and threshold the absolute value of the result.
- Points that are left are the strongest responses which for lines, one pixel thick correspond closest to the direction defined by the mask



a b
c d
e f

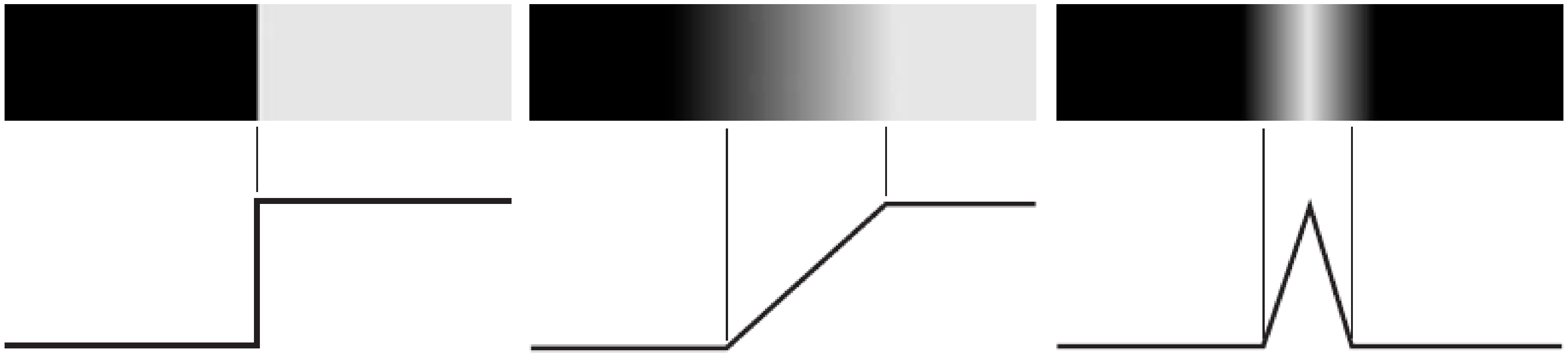
FIGURE 10.7
(a) Image of a wire-bond template. (b) Result of processing with the +45° line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

EDGE MODELS

- To understand Edge Detection
- Approaches for implementing first and second order digital derivatives for the detection of edges in an image
- An edge is a set of connected pixels that lie on the boundary between two regions.
- Difference between Edge and Boundary.
- Edge-local concept Region Boundary-Global idea
- Edge- Ability to measure gray-level transitions in a meaningful way

a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.



really nothing more than a 1 pixel thick line running through a region in an

EDGE MODELS

- Ideal Edge-Set of connected pixels, each of which is located at an orthogonal step transition in gray level
- Degree of blurring derermined by the factors

➢ Quality of Image System

➢ Sampling Rate

➢ Illumination conditions

Thus edges have a ramp like profile

EDGE MODELS

- Slope determined by degree of blurring. Blurred edges-thick Sharp edges-thin
-

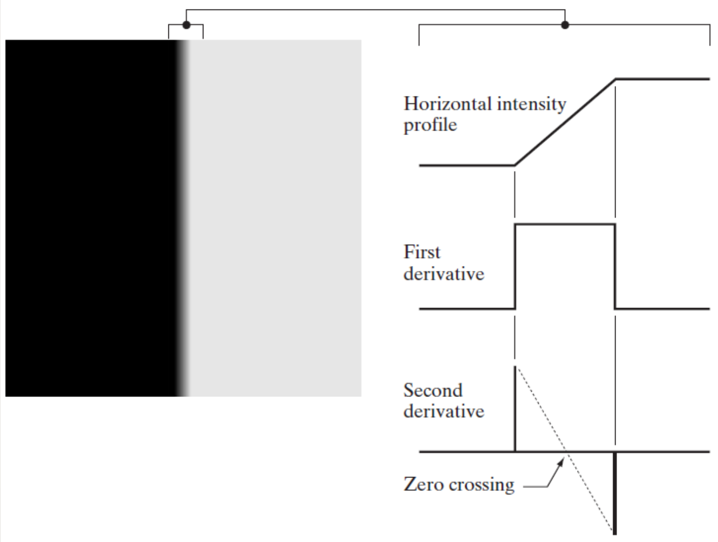


FIGURE 10.10 (a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

EDGE MODELS

- Magnitude of first derivative-Used to detect the presence of an edge at a point in the image
- Sign of second derivative-determine whether edge pixels lie on dark or light side of the edge.
- First order derivatives in an image are computed using a gradient
- Second order derivatives are obtained using a Laplacian

EDGE MODELS

Conclusion:

- To classify as a meaningful edge point, the transition in the gray level associated with that point has to be significantly stronger than the background at that point
- The method of choice to determine whether the value is "significant" or not is to use a threshold
- Edge points in an image can be called as zero crossing points of its second derivatives

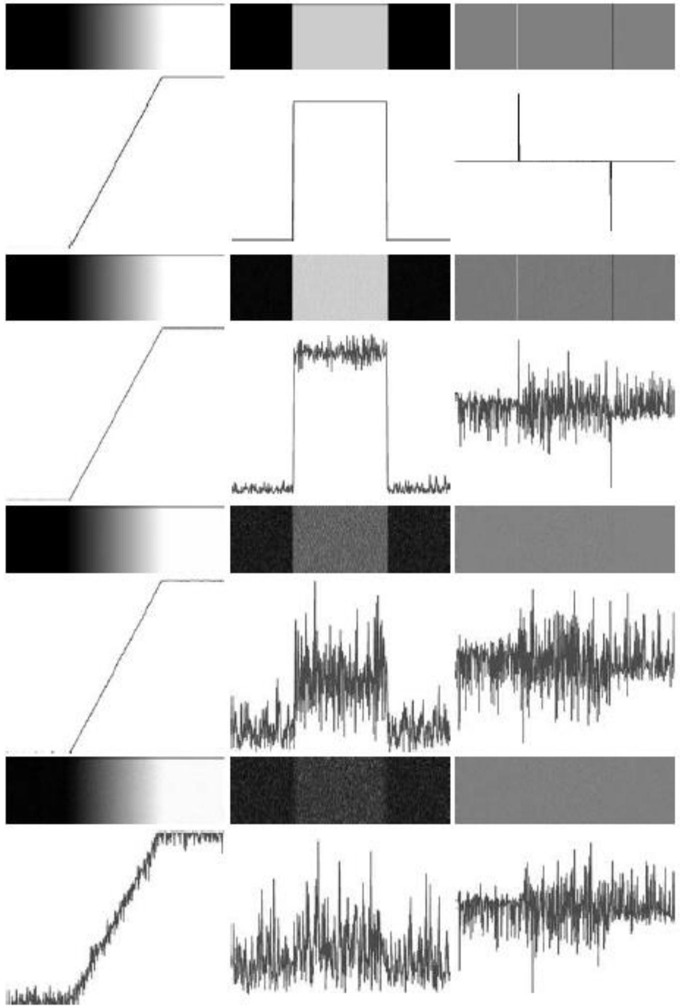


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

EDGE DETECTION

Gradient operators:

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

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

- Gradient vector points in the direction of maximum rate of change of f at coordinates (x,y)
- Magnitude of the vector is given by $M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$
- Value of rate of change in the direction of the gradient vector.

EDGE DETECTION

- ∇f is called the gradient
- Let $\alpha(x, y)$ represent the direction angle of the vector ∇f at (x,y)
- The direction of the gradient vector is given by the angle

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

- Obtaining the gradient of an image requires computing the partial derivatives $\partial f/\partial x$ $\partial f/\partial y$ and at every pixel location in the image

EDGE DETECTION

- We know that

$$g_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y)$$

$$g_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y)$$

- Corresponding 1D masks to represent above equations will be

−1
1

−1	1

EDGE DETECTION

- Consider the mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- A 3X3 region of an image (the z's are gray level values) and various masks used to compute the gradient at point labelled z5

EDGE DETECTION

- When diagonal edge direction is of interest, we need a 2-D mask. The ***Roberts cross-gradient operators*** are one of the earliest attempts to use 2-D masks with a diagonal preference
- The Roberts operators are based on implementing the diagonal differences

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

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

a	
b	c
d	e
f	g

FIGURE 10.14
A 3×3 region of an image (the z's are intensity values) and various masks used to compute the gradient at the point labeled z_5.

Filter masks used to compute the derivatives needed for the gradient are often called *gradient operators, difference operators, edge operators*, or *edge detectors*.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

EDGE DETECTION

- Masks of size 2X2 are simple conceptually, but they are not as useful for computing edge direction as masks that are symmetric about the center point, the smallest of which are of size 3X3.
- These masks take into account the nature of the data on opposite sides of the center point and thus carry more information regarding the direction of an edge.
- The simplest digital approximations to the partial derivatives using masks of size 3X3 are given by

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

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

EDGE DETECTION

- In these formulations, the difference between the third and first *rows* of the 3X3 region approximates the derivative in the *x*-direction, and the difference between the third and first columns approximate the derivate in the *y*-direction.
- It can be implemented over an entire image by filtering with the two masks shown in above figure.
- These masks are called the ***Prewitt operators***

EDGE DETECTION

- A slight variation of the preceding two equations uses a weight of 2 in the center coefficient:

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

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

- Masks used to implement these equations are called the ***Sobel operators***

EDGE DETECTION

- The Prewitt masks are simpler to implement than the Sobel masks, but, the slight computational difference between them typically is not an issue.
- The fact that the Sobel masks have better noise-suppression (smoothing) characteristics makes them preferable because, noise suppression is an important issue when dealing with derivatives.
- Coefficients of all the masks sum to zero, thus, giving a response of zero in areas of constant intensity, as expected of a derivative operator.

a	b
c	d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

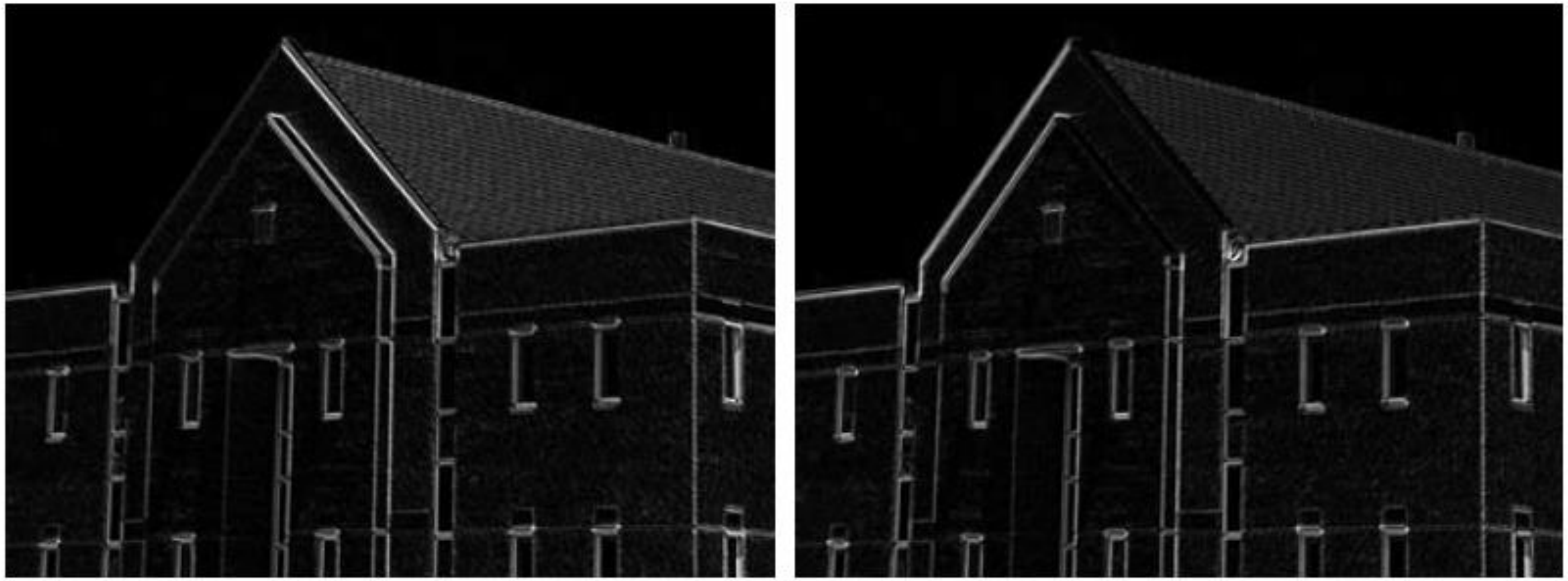
Sobel

a b
c d

FIGURE 10.16
(a) Original image of size 834×1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.



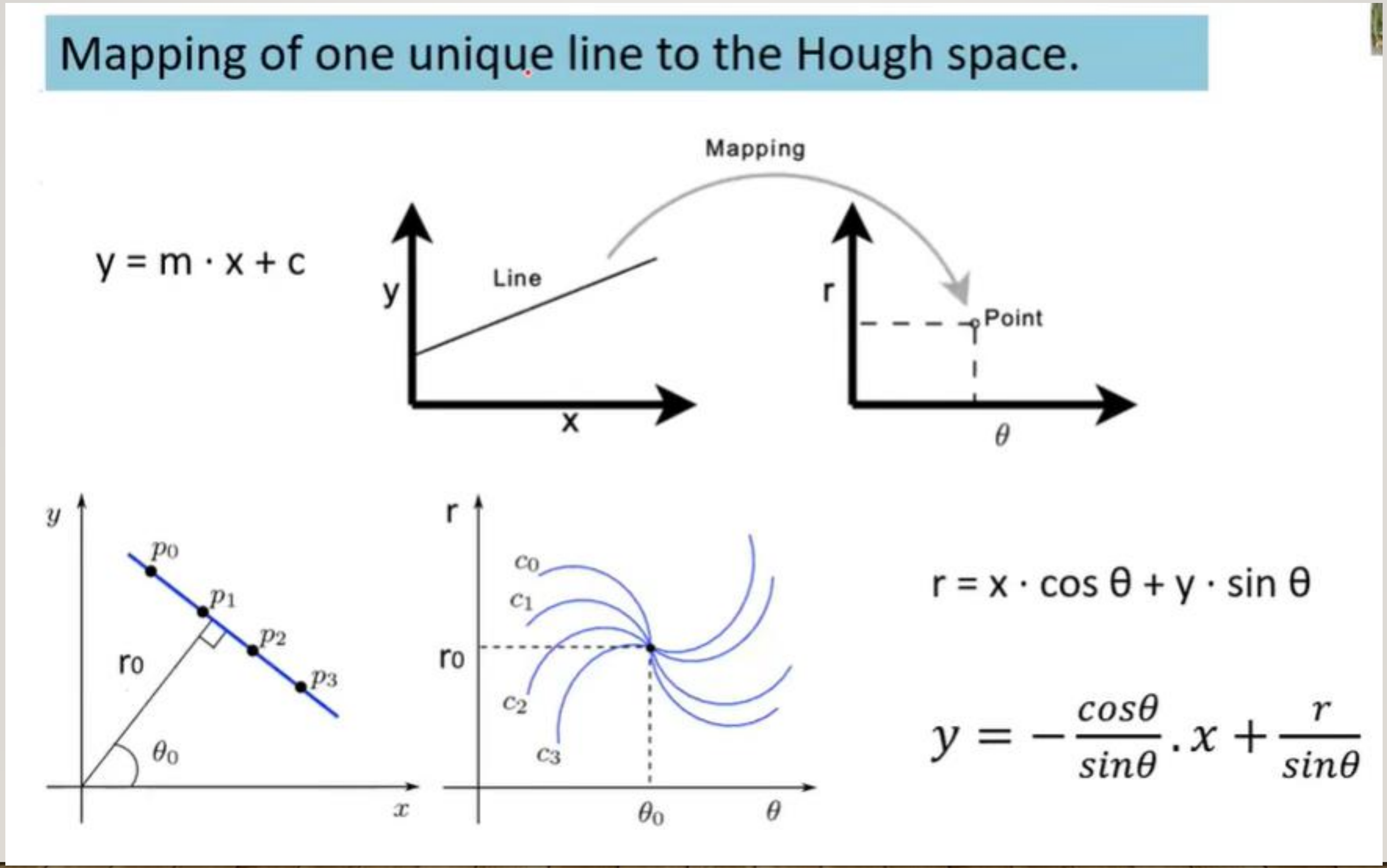
a b

FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

EDGE LINKING USING HOUGH TRANSFORM

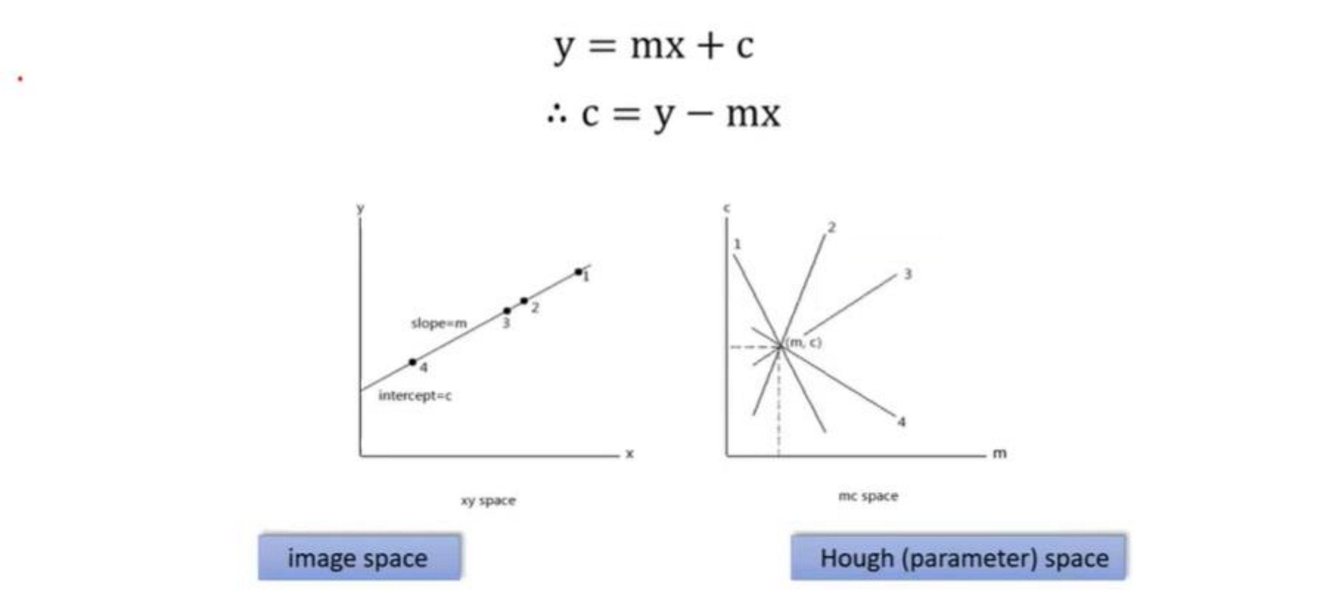
- Hough transform is a feature extraction technique
- Used to detect lines, circles or dots
- Idea of Hough transform-Every edge point in the edge map is transformed to all possible lines that could pass through that point.

REPRESENTATION OF LINES IN HOUGH SPACE



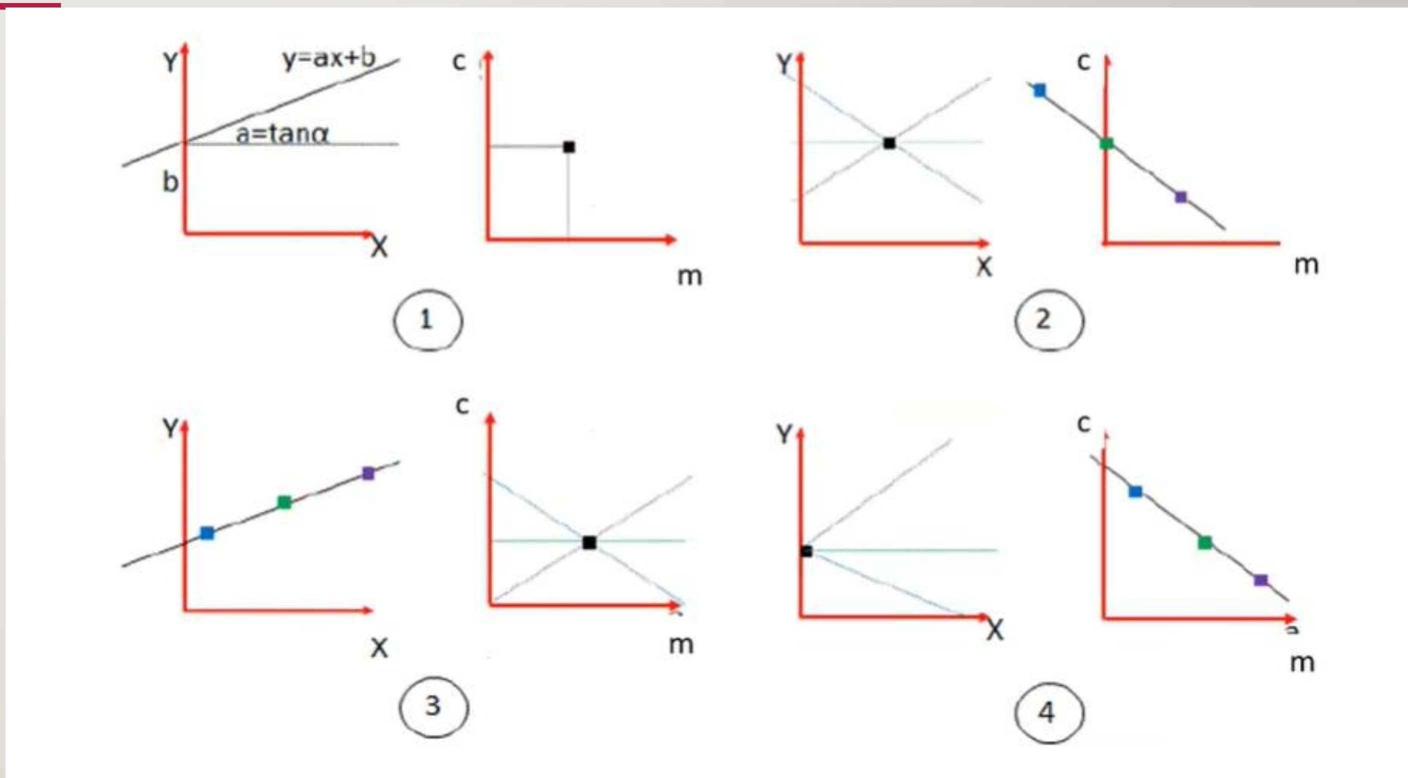
HOUGH TRANSFORM

- Hough's transform is all about converting points in xy space to lines in mc space
- Straight line y=mx+c can be represented as a point (c,m) in the parameter space.



DIFFERENT CASES OF HOUGH TRANSFORM APPLIED TO STRAIGHT LINES

- Image Space is represented at the left and the Parameter Space is represented at the right.



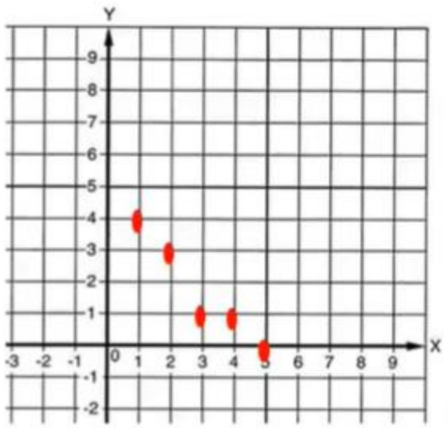
HOUGH TRANSFORM

- Given 5 points use Hough transform to draw a line joining these points

(1,4), (2,3), (3,1), (4,1), (5,0)

(1,4) (2,3) (3,1) (4,1) (5,0)

X	1	2	3	4	5
y	4	3	1	1	0



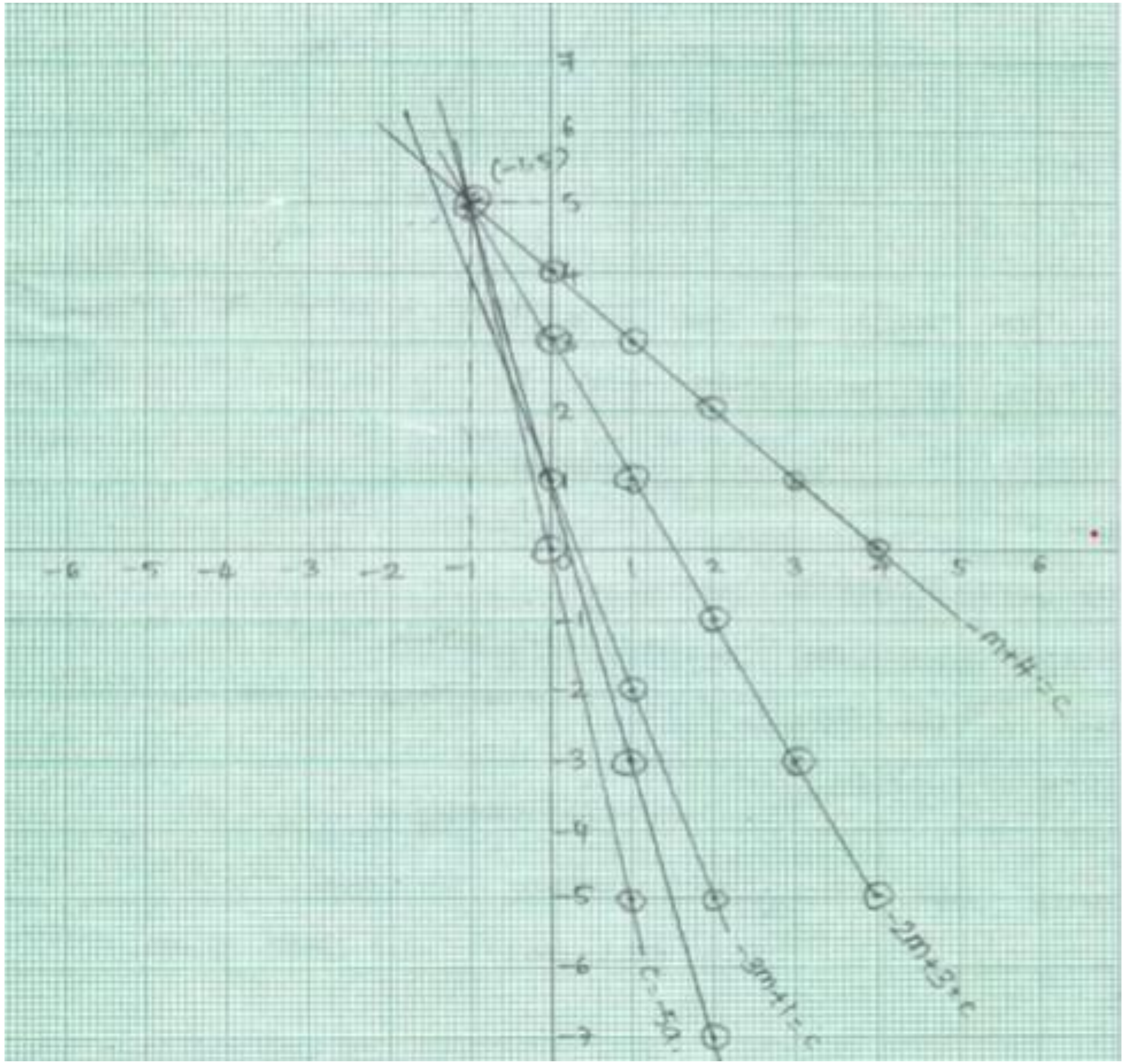
HOUGH TRANSFORM

- Equation of straight line: y=mx+c   c=y-mx

x	1	2	3	4	5
y	4	3	1	1	0
Eq.n of lines	c = 4-m	c = 3-2m	c = 1-3m	c = 1-4m	c =0 -5m

Eq.n of lines	c = 4-m	c = 3-2m	c = 1-3m	c = 1-4m	c =0 -5m

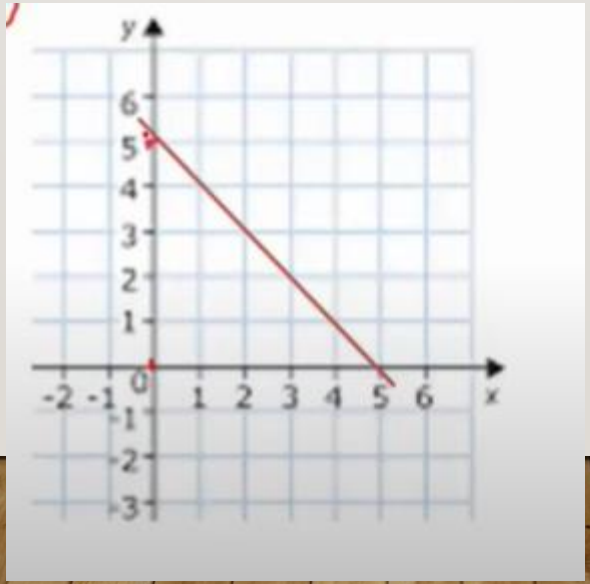
- Draw these lines on graph paper in parameter space(m-c plane).



(-1,5)
-m+4=c
-2m+3=c
-3m+1=c
c=5a

HOUGH TRANSFORM

- Except (3,1) all the remaining points meet at a common point (-1,5) where m=-1,c=5.
- Hence this value of m and c are used to draw a line ie: **y=-x+5.**
- Hence using Hough transform, the line is drawn joining only 4 points that is (1,4) (2,3), (4,1), (5,0) from given data except (3,1)



HOUGH TRANSFORM

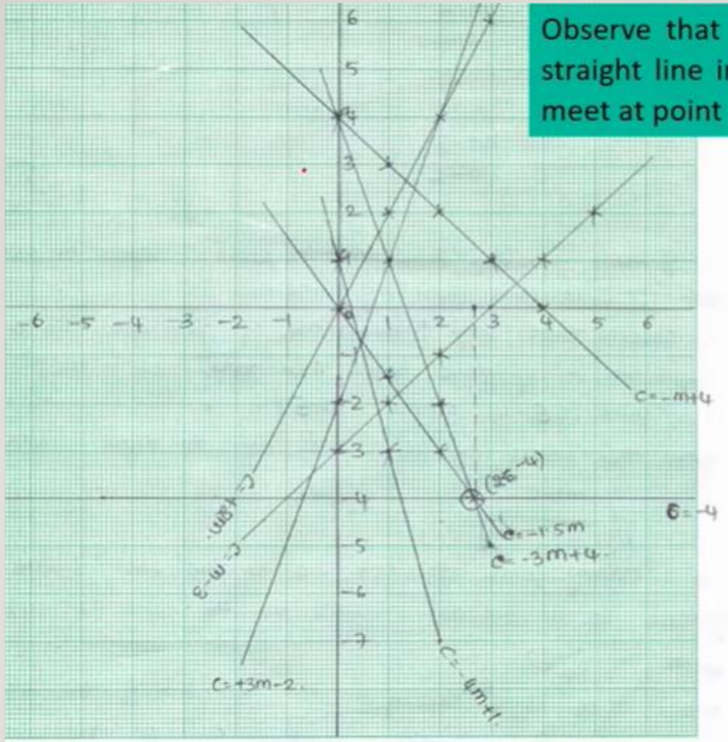
- Using Hough's transform find line passing maximum number of points

  (3,4), (0,-4), (1,4), (6,12),(4,1), (1.5,0), (-2,0), (-1,-3), (3,-2)

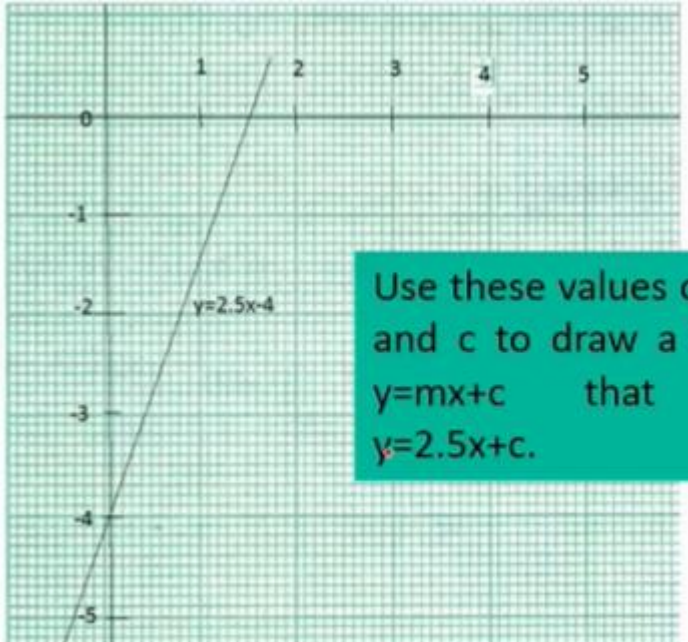
X	3	0	1	6	4	1.5	-2	-1	3
y	4	-4	4	12	1	0	0	-3	-2

- Equation of the line: y=mx+c   c=y-mx

x	y	Equation of line
3	4	$c = 4 - 3m$
0	-4	$c = -4$
1	4	$c = 4 - m$
6	12	$c = 12 - 6m$
4	1	$c = 1 - 4m$
1.5	0	$c = -1.5m$
-2	0	$c = +2m$
-1	-3	$c = m - 3$
3	-2	$c = 3m - 2$



Observe that only three points of given data lie on the straight line in x-y plane, hence three lines in m-c plane meet at point (2.5, -4) where x=2.5 and y=-4



Use these values of m and c to draw a line y=mx+c that is y=2.5x+c.

IMAGE SEGMENTATION BASED ON THRESHOLDING

- Thresholding techniques produce segments having pixels with similar intensities.
- Thresholding is a useful technique for establishing boundaries in images that contain solid objects resting on a contrasting background.
- The thresholding technique requires that an object has homogenous intensity and a background with a different intensity level.
- Such an image can be segmented into two regions by simple thresholding.

GLOBAL THRESHOLDING

- Global thresholding is the simplest and most widely used of all possible segmentation methods.
- In global thresholding, a threshold value of θ is chosen and the following condition is imposed

$$f(m, n) = \begin{cases} 1 & \text{if } f(m, n) \geq \theta \\ 0 & \text{else} \end{cases}$$

GLOBAL THRESHOLDING

- Equation is a complete description of a binarisation algorithm;
- It contains no indication on how to select the value of the threshold parameter θ.
- The value of θ has to be selected in an optimal manner.
- Global thresholding will suffer when pixels from different segments overlap in their use of intensities
- If the overlap is due to variation in illumination across the image, variable thresholding could be used.
- This can be visualised as a form of local segmentation.

ADAPTIVE THRESHOLDING

- Global thresholding, or fixed thresholding, works well if the objects of interest have a reasonably uniform interior gray level and rest on a background of unequal but a relatively uniform gray level.
- In many cases, the background gray level is not constant, and object contrast varies within an image.
- In such cases, a threshold that works well in one area might not work well in other areas of the image.
- In these cases, it is convenient to use a threshold gray level that is a slowly varying function of position in the image

LIMITATIONS OF THRESHOLDING

- Thresholding is often used as an initial step in a sequence of image-processing operations.
- The main limitation of thresholding techniques is that in its simplest form, only two classes are generated and it cannot be applied to multi-channel images.
- A thresholding technique does not take into account the spatial characteristics of an image.
- This causes it to be sensitive to noise and intensity inhomogeneities.

REGION BASED SEGMENTATION

- Objective-Partition an image into regions
- Finding boundaries between regions based on discontinuities in Gray levels
- Accomplished via thresholds based on distribution of pixel properties such as Gray level values or colour.

Region Growing:

- Procedure that groups pixels or subregions into larger regions based on predefined criteria

REGION BASED SEGMENTATION

Region Growing:

- Start with a set of seed points and from these grow regions by appending to each seed those neighbouring pixels that have properties similar to the seed
- 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
- When the images are monochrome, region analysis must be carried out with a set of descriptors based on intensity levels and spatial properties (such as moments or texture).

REGION BASED SEGMENTATION

- Descriptors alone can yield misleading results if connectivity properties are not used in the region-growing process.
- For example, visualize a random arrangement of pixels with only three distinct intensity values.
- Grouping pixels with the same intensity level to form a "region" without paying attention to connectivity would yield a segmentation result that is meaningless.

REGION BASED SEGMENTATION

- Another problem in region growing is the formulation of a stopping rule.
- Region growth should stop when no more pixels satisfy the criteria for inclusion in that region.
- Criteria such as intensity values, 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 and the shape of the region being grown.

REGION BASED SEGMENTATION

- Another problem in region growing is the formulation of a stopping rule.
- Region growth should stop when no more pixels satisfy the criteria for inclusion in that region.
- Criteria such as intensity values, 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 and the shape of the region being grown.

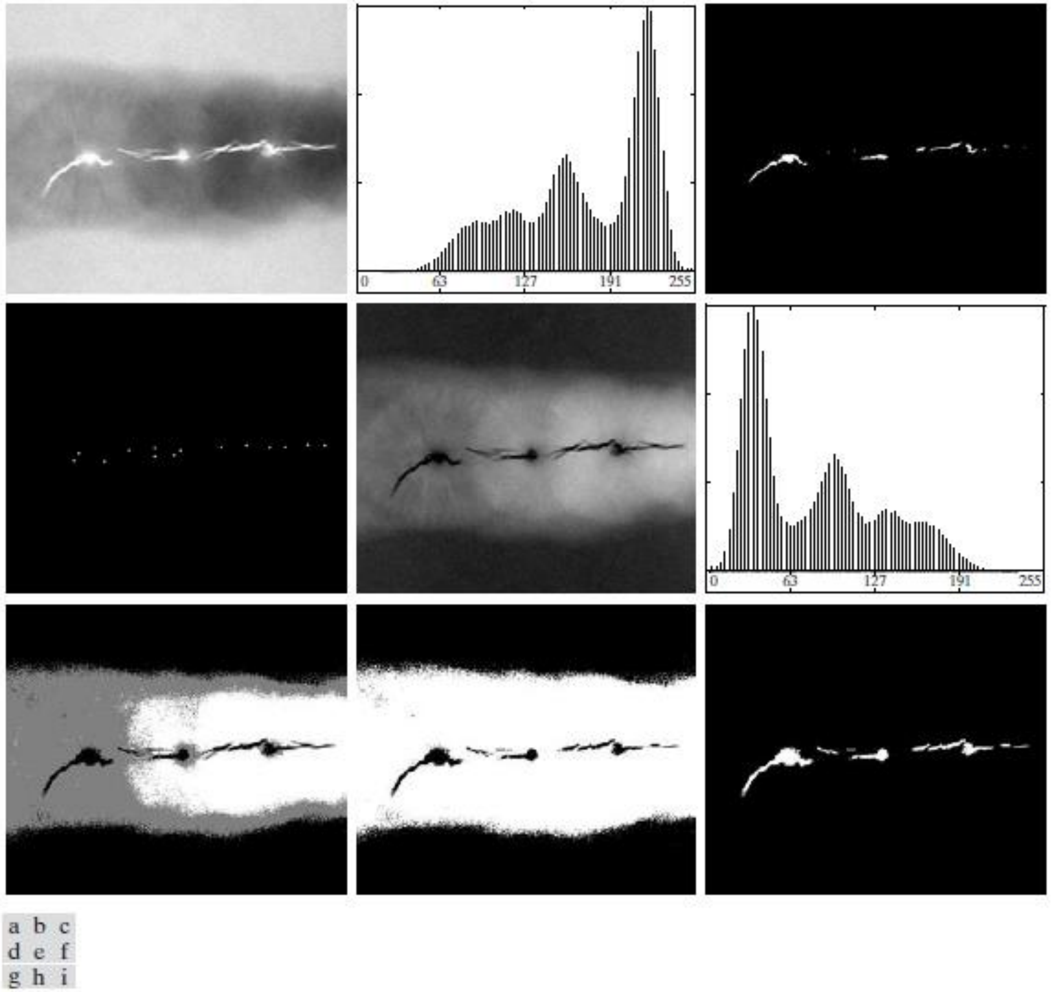


FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference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

Region Splitting and Merging:

- Let R represent the entire image region and select a predicate Q.
- One approach for segmenting R is to subdivide it successively into smaller and smaller quadrant regions so that, for any region Ri, Q(Ri)=True.
- If Q(R)=False, we divide the image into quadrants.
- If Q is FALSE for any quadrant, we subdivide that quadrant into subquadrants, and so on.
- This particular splitting technique has a convenient representation in the form called ***quadtrees.***
- Root of the tree corresponds to the entire image and that each node corresponds to the subdivision of a node into four descendant nodes

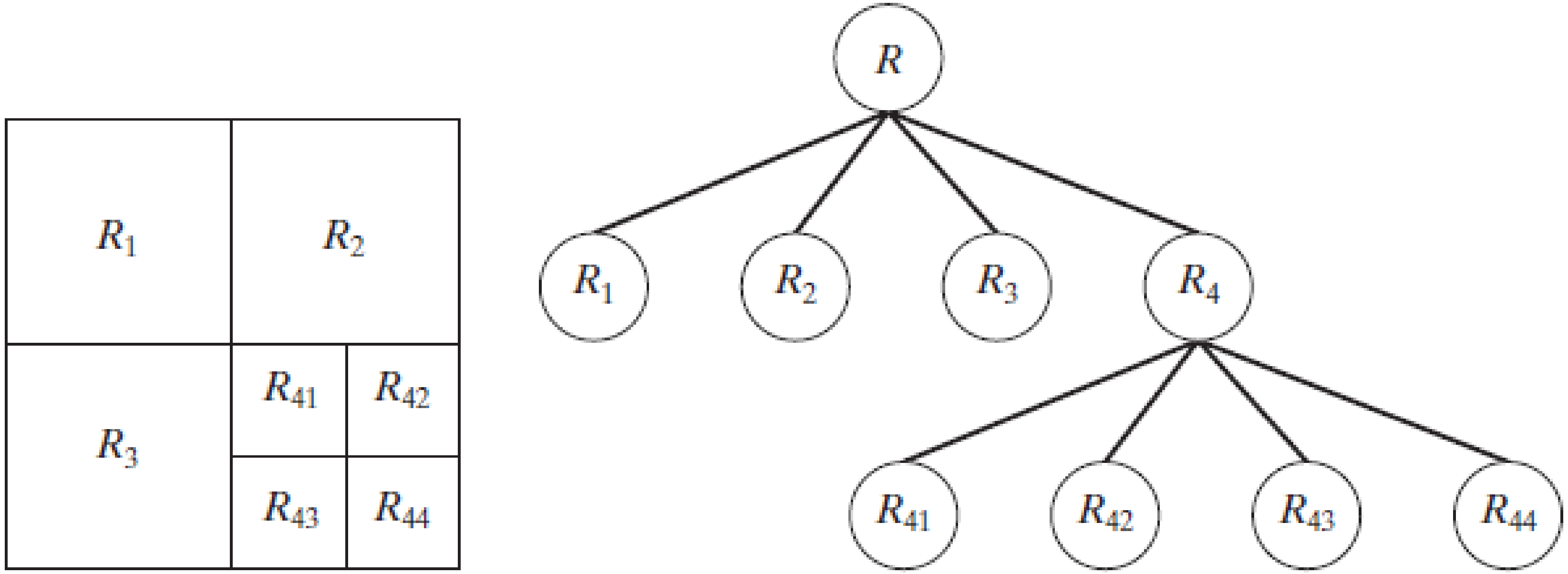


FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

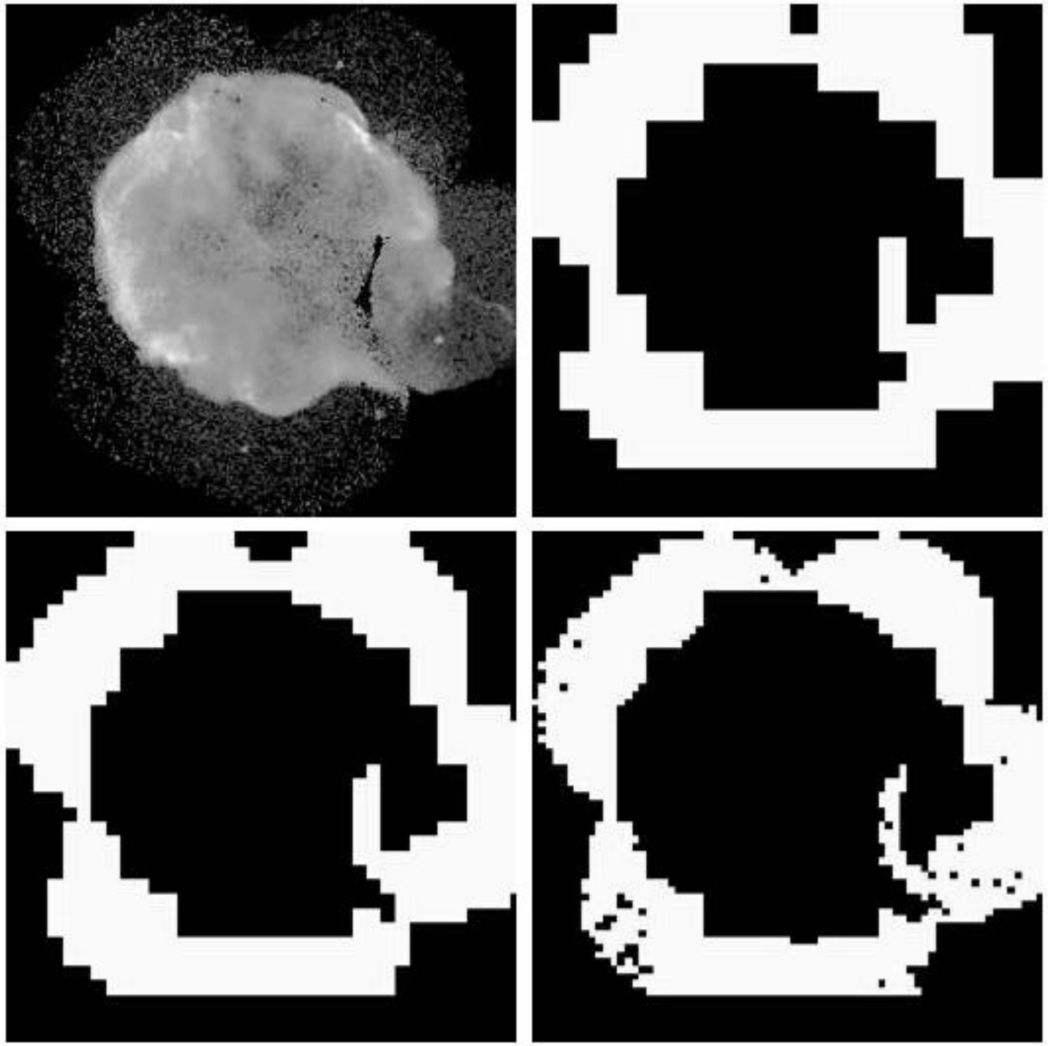
REGION BASED SEGMENTATION

- If only splitting is used, the final partition normally contains adjacent regions with identical properties.
- This drawback can be remedied by allowing merging as well as splitting
- That is, two adjacent regions Ri and Rk and are merged only if $Q(R_j \cup R_k)$= True.

REGION SEGMENTATION

Region Splitting and Merging-Summary:

- Split into four disjoint quadrants any region R_i for which $Q(R_i)=False$
- When no further splitting is possible, merge any adjacent regions R_j and R_k for which $Q(R_j \cup R_k)=True$
- Stop when no further merging is possible.



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.)

IMAGE SEGMENTATION BY CLUSTERING

- Image Segmentation is the process of partitioning an image into multiple regions based on the characteristics of the pixels in the original image.
- Clustering is a technique to group similar entities and label them.
- Thus, for image segmentation using clustering, we can cluster similar pixels using a clustering algorithm and group a particular cluster pixel as a single segment.
- The process of image segmentation by clustering can be carried out using two methods.

➢ **Agglomerative** clustering

➢ **Divisive** clustering

IMAGE SEGMENTATION BY CLUSTERING

- In **Agglomerative clustering**, we label a pixel to a close cluster and then increase the size of the clusters iteratively. The following steps explain the process of Agglomerative clustering.
- Each pixel is considered to be an individual cluster
- Similar clusters with smaller inter-cluster distances are merged.
- The steps are repeated.

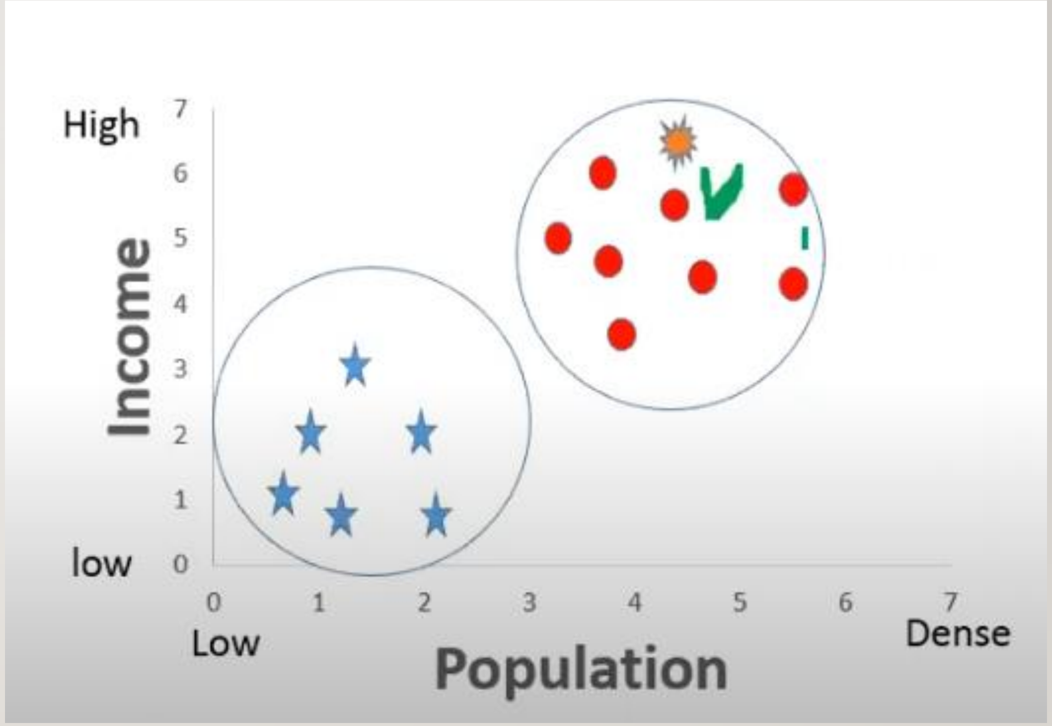
IMAGE SEGMENTATION BY CLUSTERING

- In **Divisive clustering**, the following process is followed.
- All the pixels are assigned to a single cluster.
- The cluster is split into two with large inter-cluster distance over some epochs.
- The steps are repeated until the optimal number of clusters is reached.

CLUSTERING

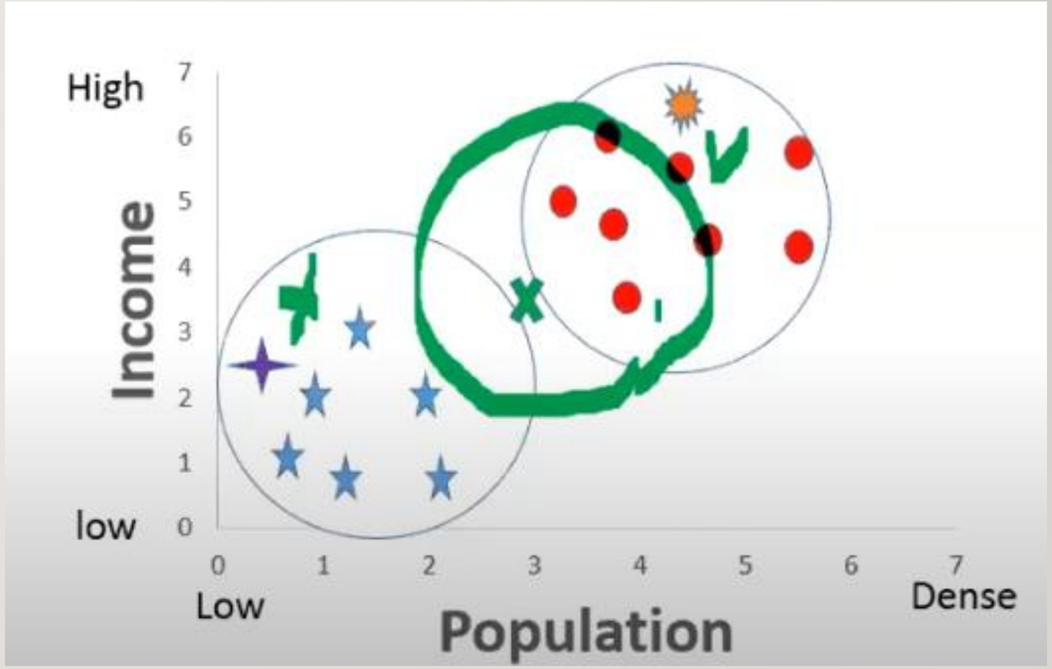
Clustering with Real Life Example :

- Bob wants to open a pizza outlet and wants to select location for targeted customers
- His selection on a location will be based on-**Income** and **Population**



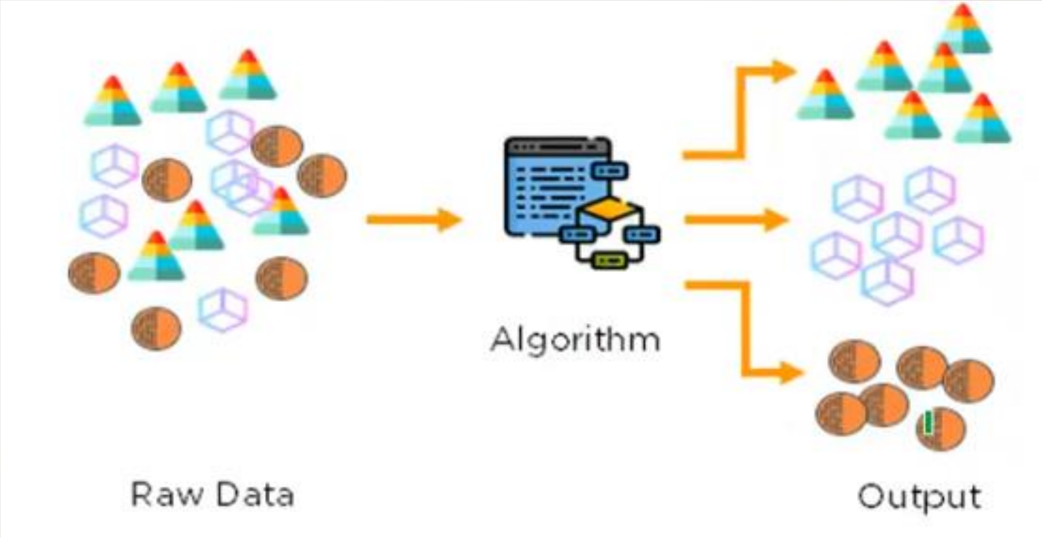
CLUSTERING

- Location A-Dense and High Income
- Location B-Low Population and Low Income
- Location C-Average Population and Average Income

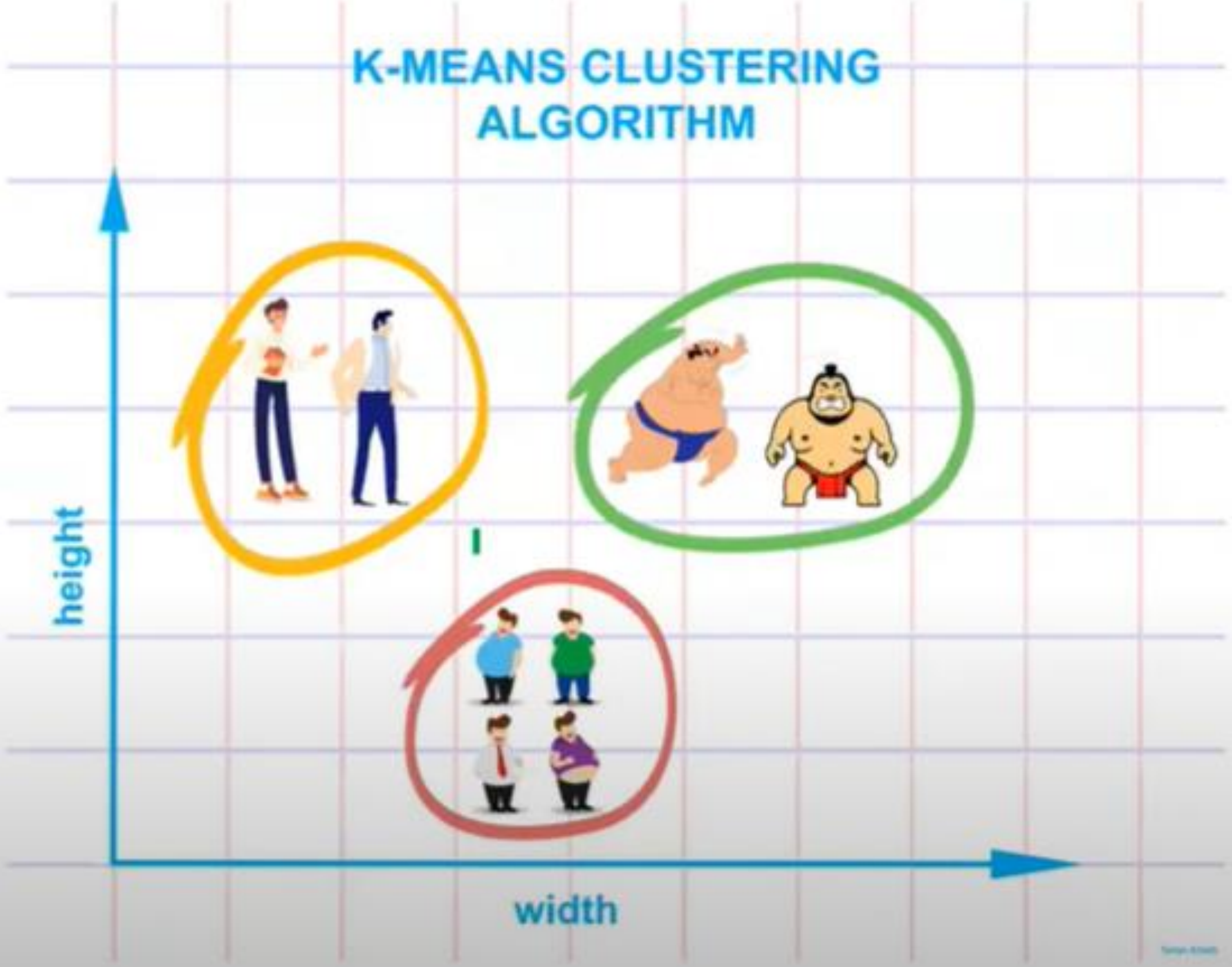


K MEANS CLUSTERING

- Simplest Unsupervised learning algorithm that solves the clustering problem
- K represents the number of clusters



S. N	Height	Weight
1	6	50
2	5	100
3	6.5	57
4	4.5	90
5	4.8	67
6	5	59
7	5.1	55
8	4.5	60



Example

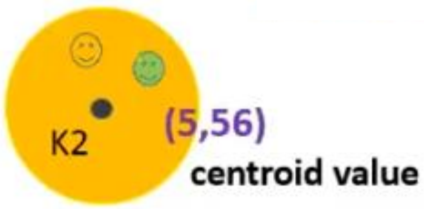
Boys	Height	Weight
1	4	50
2	5	56
3	5.5 xo	83 yo
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

observed values

Let us make two clusters ie k=2, K1 and k2

centroid for K1 =(4,50)

centroid for K2 =(5,56



Select centroid for K1 and k2

Let row 1 is centroid of K1, and row 2 is centroid of k2

Now lets assign other entries to k1 or k2 using Euclidian distance formula

$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 – *observed value*
x_c – centroid value

For row 3

ED for K1 , x_0=5.5, $x_c = 4, y_0 = 83, yc = 50$ **ED for K1 =33.0341**

ED for K2 , x_0=5.5, $x_c = 5, y_0 = 83, yc = 56$ **ED for K2 =27.0046**

K1 = {1, } **k2={2, 3 }**

Centroid for K1 remain same =(4,50)

Calculate new centroid for K2

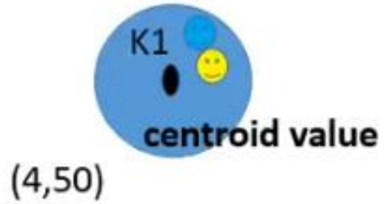
5.5+5/2=5.25

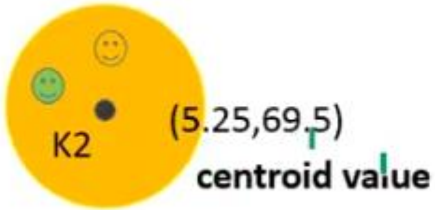
83+56/2=69.5

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5 xo	45 yo
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

observed values

centroid for K1 =(4,50) centroid for K2 =(5.25,69.5)





Now lets assign other entries to k1 or k2 using Euclidian distance formula

$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 – *observed value*

x_c – centroid value

For row 4

ED for K1 , x_0**=4.5,** $x_c = 4, y_0 = 45, yc = 50$ **ED for K1 =5.0249**

ED for K2 , x_0**=4.5,** $x_c = 5.25, y_0 = 45, yc =$ **69.5** **ED for K2 =24.5115**

K1 = {1, **4** }

k2={2, 3}

centroid for K2 remain same

Calculate new centroid for K1

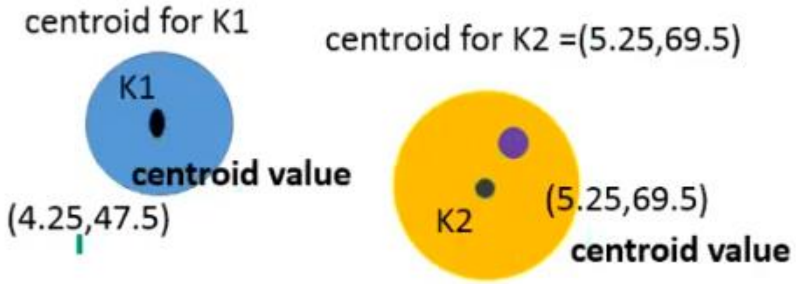
4.5+4/2=4.25

45+50/2=47.5

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

xo *yo*

observed values



Now lets assign other entries to k1 or k2 using Euclidian distance formula

$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 – *observed value*

x_c – **centroid value**

For row 5

ED for K1 , x_0**=5.8** $x_c = 4.25, y_0 = 67, yc =$**47.5** **ED for K1 = 19.5615**

ED for K2 , x_0**=5.8,** $x_c = 5.25, y_0 = 67, yc =$**69.5** **ED for K2 =2.5598**

K1 = {1, 4 }

k2={2, 3, 5 }

centroid for K1 remain same

Calculate new centroid for K2

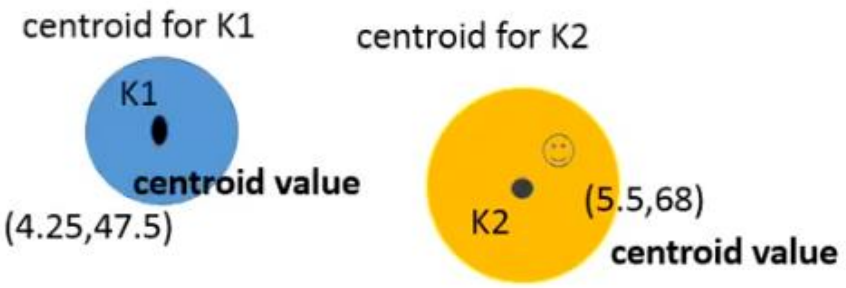
5.8+5.25/2=5.5

67+69.5/2=68

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

xo *yo*

observed values



Now lets assign other entries to k1 or k2 using Euclidian distance formula

$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 – *observed value*

x_c – **centroid value**

For row 6

ED for K1 , x_0**=6** $x_c = 4.25, y_0 = 90, yc$ **=47.5** **ED for K1 =42.536**

ED for K2 , x_0**=6,** $x_c = 5.5.6, y_0 = 90, yc$ **=68** **ED for K2 =22**

K1 = {1,4 }

k2={2, 3, 5, 6 }

centroid for K1 remain same

Calculate new centroid for K2

6+5.5/2=5.7

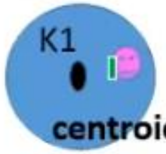
90+68/2=79

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

xo *yo*

observed values

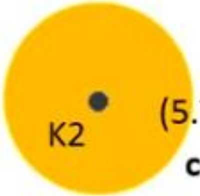
centroid for K1



centroid value

(4.25,47.5)

centroid for K2



(5.7,79)

centroid value

Now lets assign other entries to k1 or k2 using Euclidian distance formula

$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 – *observed value*

x_c – **centroid value**

For row 7

ED for K1 , x_0**=5.4** $x_c = 4.25, y_0 = 55, yc =$**47.5** **ED for K1 =7.5877**

ED for K2 , x_0**=5.4,** $x_c = 5.7, y_0 = 55, yc =$**79** **ED for K2 = 24.0019**

K1 = {1,4 , 7}

k2={2, 3,5,6}

centroid for K2 remain same

Calculate new centroid for k1

4.25+5.4/2=4.8

47.5+55/2=51

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

xo *yo*

observed values

centroid for K1 centroid for K2

$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 – *observed value*

x_c – **centroid value**

For row 8

ED for K1 , x_0**=5.6** $x_c = 4.8, y_0 = 60, yc$ **=51** **ED for K1 = 9**

ED for K2 , x_0**=5.6,** $x_c = 5.7, y_0 = 60, yc$ **=79** **ED for K2 = 19**

K1 = {1,4 , 7, **8** }

k2={2, 3,5,6}

centroid for K2 remain same

Calculate new centroid for k1

4.8+5.6/2=5.2

51+60/2=55.5

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

xo *yo* *observed values*

centroid for K1 centroid for K2)

$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

x_0 – *observed value*

x_c – **centroid value**

For row 9

ED for K1 , x_0**=4.5** $x_c = 5.2, y_0 = 43, yc =$**55.5** **ED for K1 =12.5196**

ED for K2 , x_0**=4.5,** $x_c = 5.7, y_0 = 43, yc =$**79** **ED for K2 = 36**

K1 = {1,4 , 7,8, **9** }

k2={2, 3,5,6}

centroid for K2 remain same

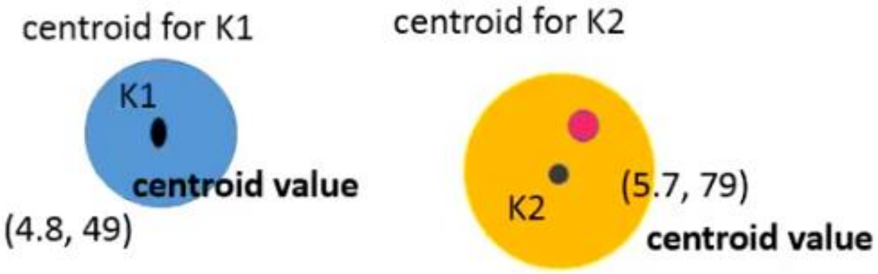
Calculate new centroid for k1

5.2+4.5/2=4.8

55.5+43/2=49

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5 ***xo***	80 ***yo***

observed values



$$\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

$\boldsymbol{x_0 - observed\ value}$

$\boldsymbol{x_c -}$ **centroid value**

For row 10

ED for K1 , x_0**=5** $x_c = 4.8, y_0 = 80, yc =$**49**

ED for K1 = 31

ED for K2 , x_0**=5,** $x_c = 5.7, y_0 = 80, yc =$**79**

ED for K2 = 1.2207

K1 = {1,4 , 7,8,9}

k2={2, 3,5,6, **10** }

Boys	Height	Weight
1	4	50
2	5	56
3	5.5	83
4	4.5	45
5	5.8	67
6	6	90
7	5.4	55
8	5.6	60
9	4.5	43
10	5	80

K1 = {1,4 , 7,8,9}

k2={2, 3,5,6, 10 }

IMAGE MORPHOLOGY

Image Morphology:

- Image morphology is a set of image processing techniques that analyses and modifies the shape and structure of objects in an image by using a small kernel called a **structuring element**
- Used for Segmentation and Feature Extraction
- Two Basic operations we use in image morphology are:

➢ Erosion

➢ Dilation

DILATION AND EROSION

Dilation:

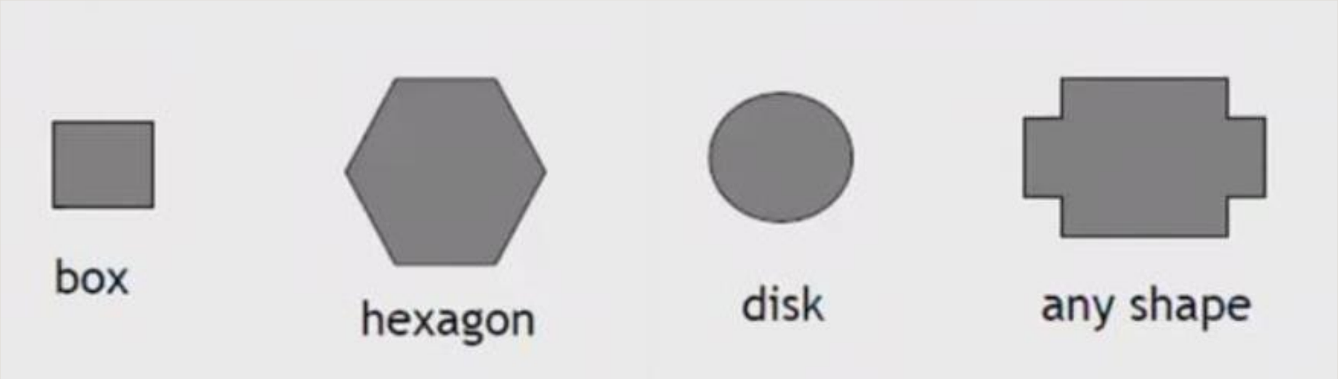
- Process that adds pixels to the boundaries of objects in an image
- If A represents the original image and B represents the structural element then dilation is represented as $A \oplus B$

Erosion:

- Process that removes pixels on object boundaries
- If A represents the original image and B represents the structural element then erosion is represented as $A \ominus B$

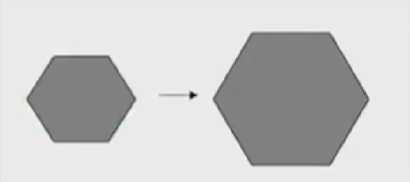
DILATION AND EROSION

- The number 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.
- Structural element is a shape mask used in the basic morphological operations.
- They can be of any shape and size that is digitally representable and each has an origin.



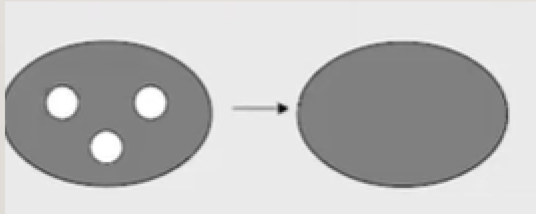
DILATION AND EROSION

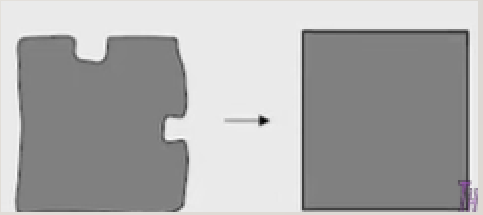
- Dilation fills in holes, smoothes object boundaries and adds an extra outer ring of pixels into the object boundary ie, object becomes slightly larger.
- Can be used to

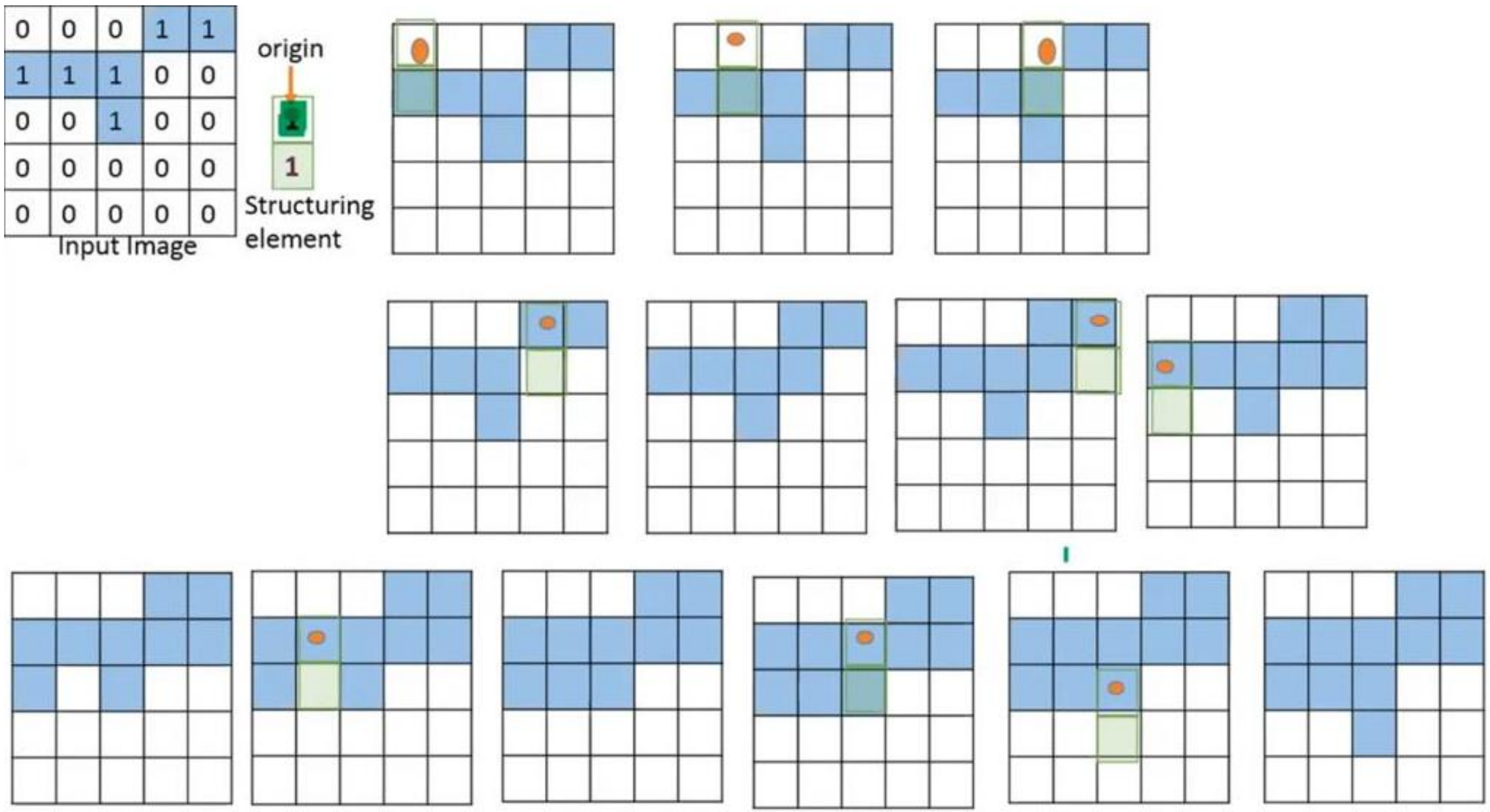


➢ Expand shapes

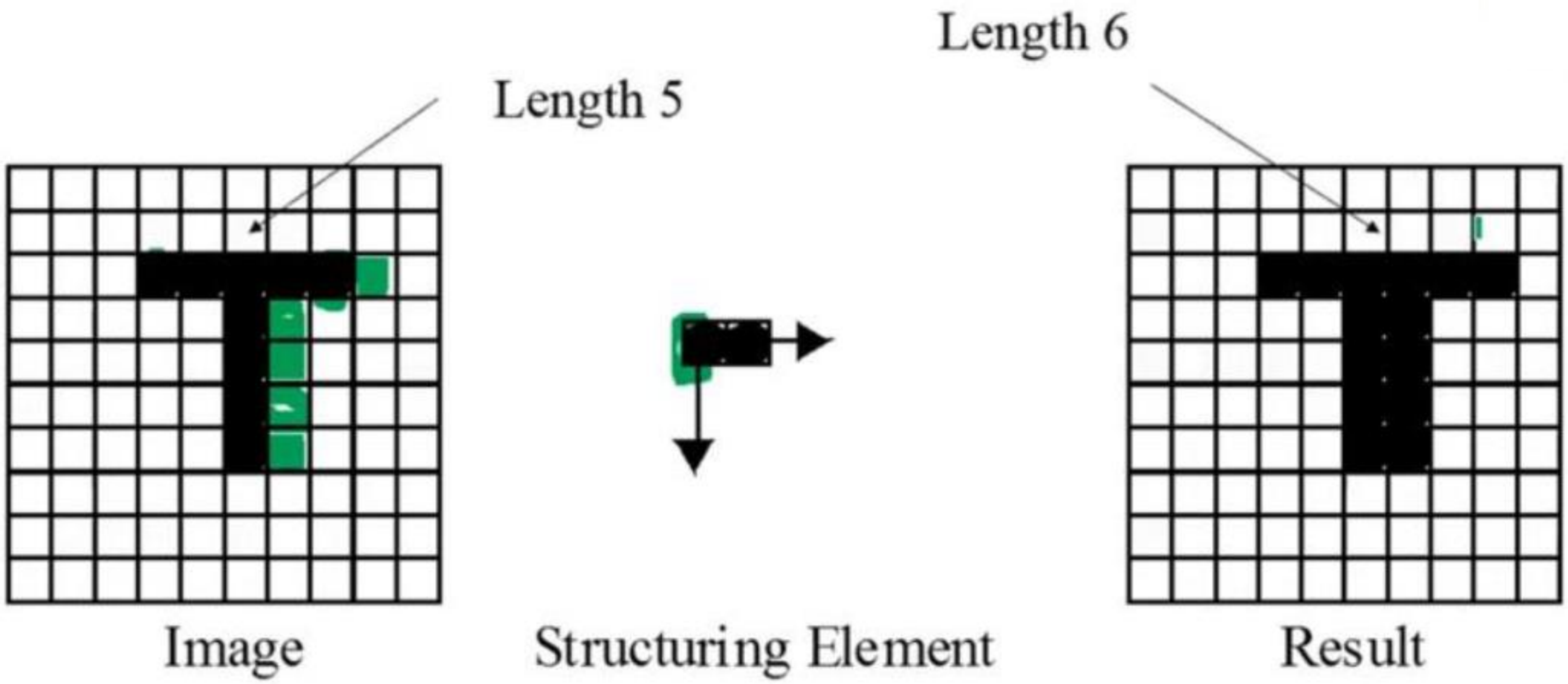
➢ Filling holes



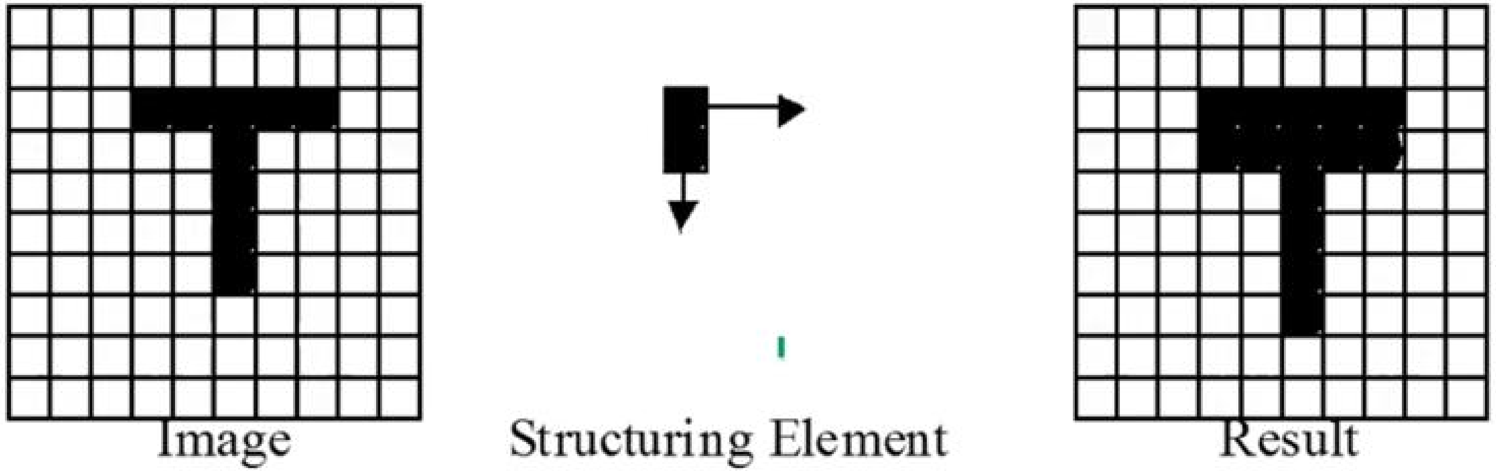



0 0 0 1 1
1 1 1 0 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0
Input Image
origin
1
Structuring element

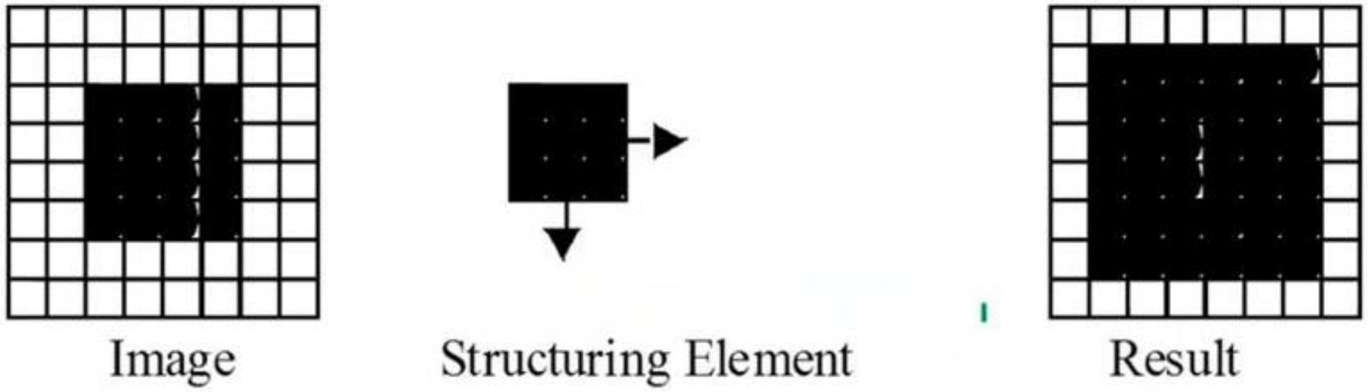
Structuring Element for Dilation



Structuring Element for Dilation



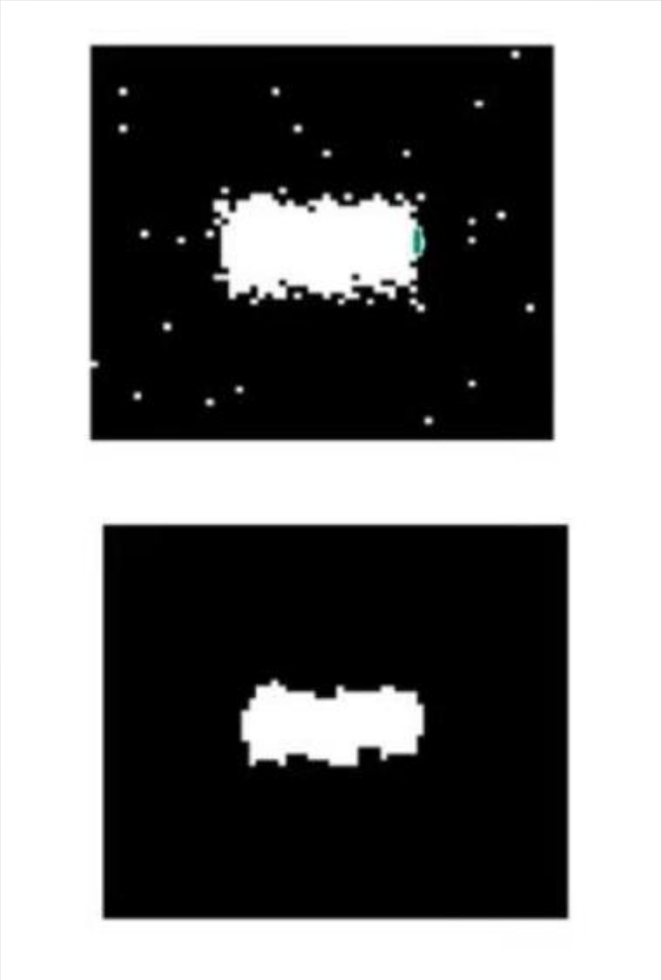
Structuring Element for Dilation

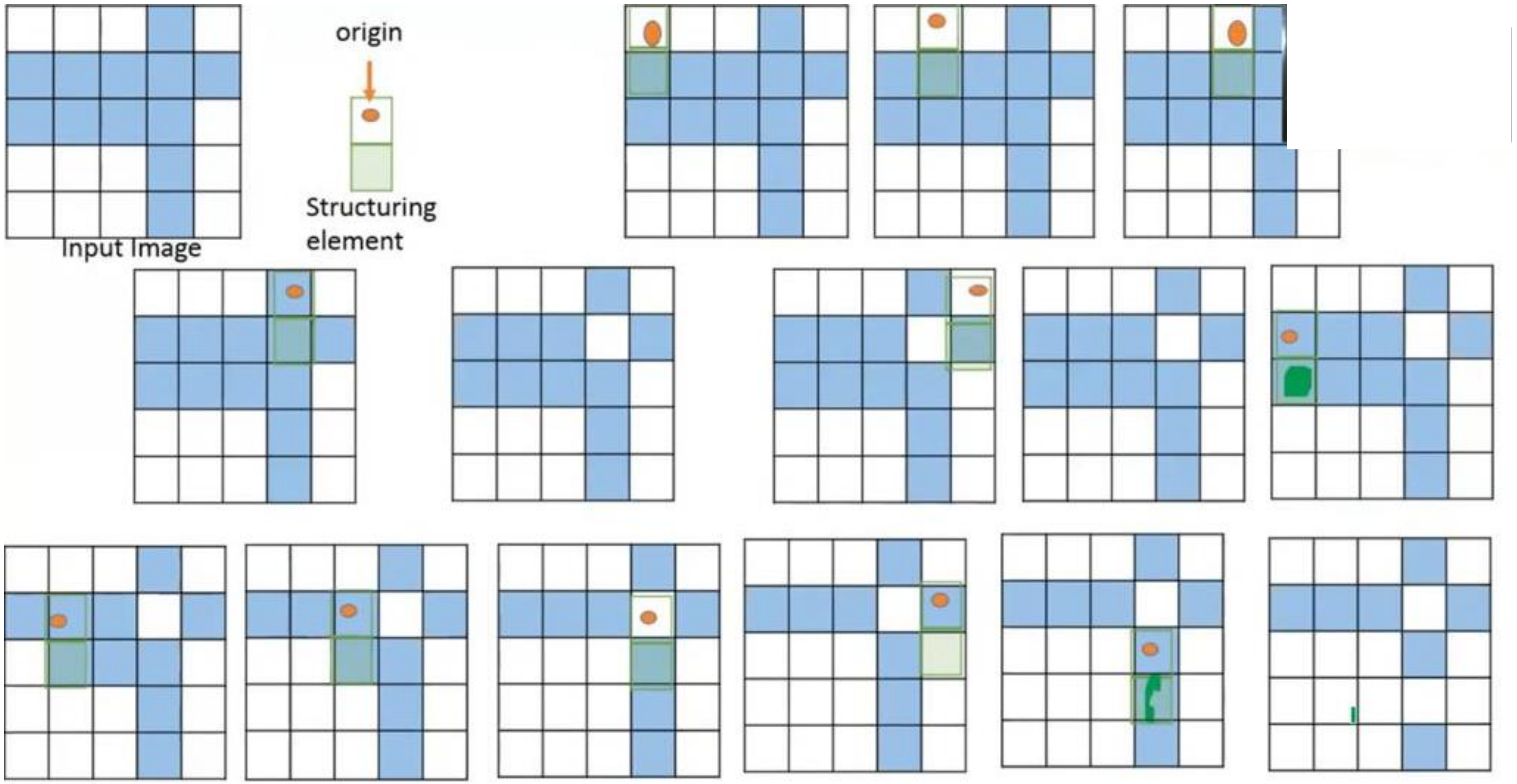


Dilation example

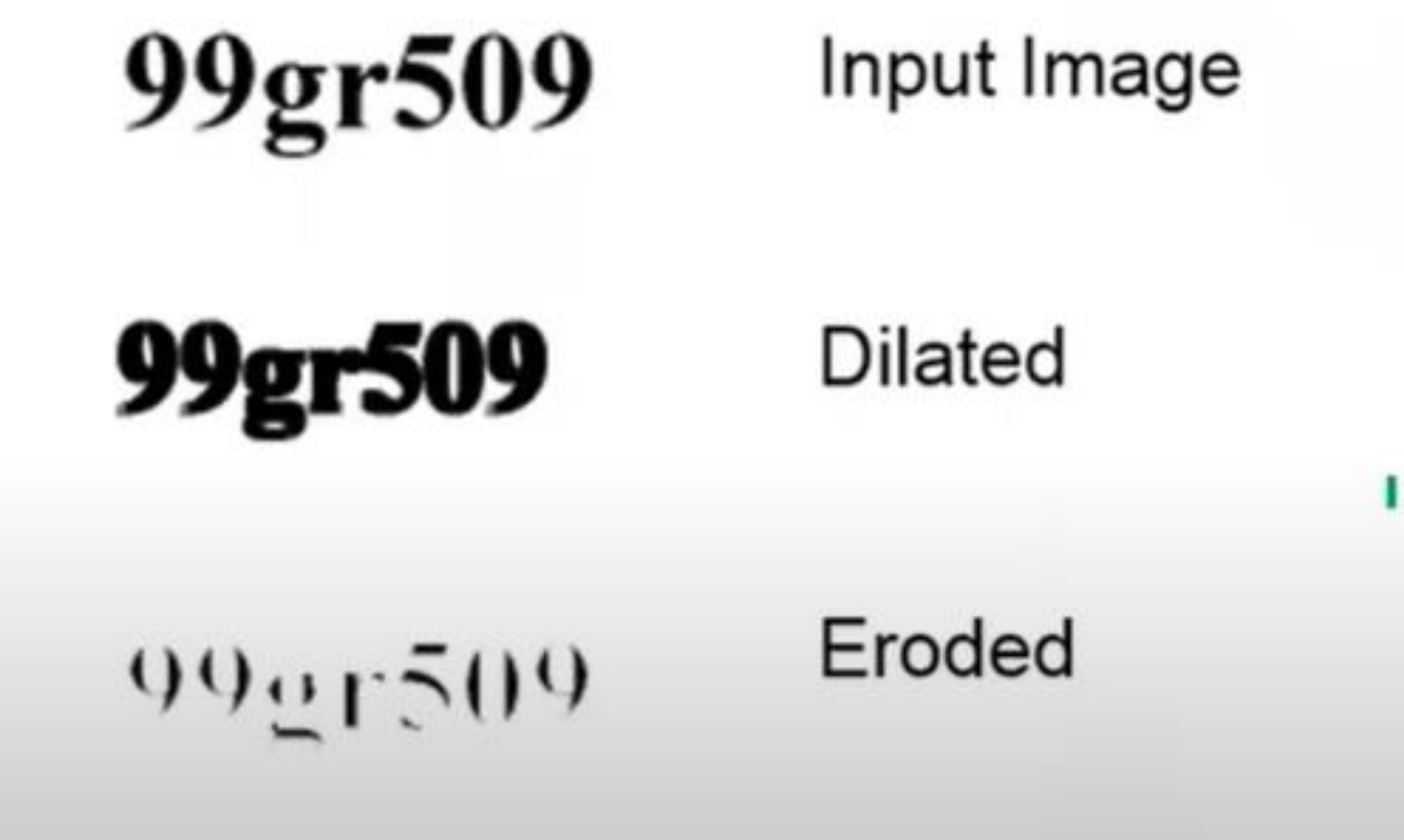
EROSION

- Erosion removes isolated noisy pixels
- Smoothes object boundary
- Removes the outer layer of object pixels ie, object becomes slightly smaller



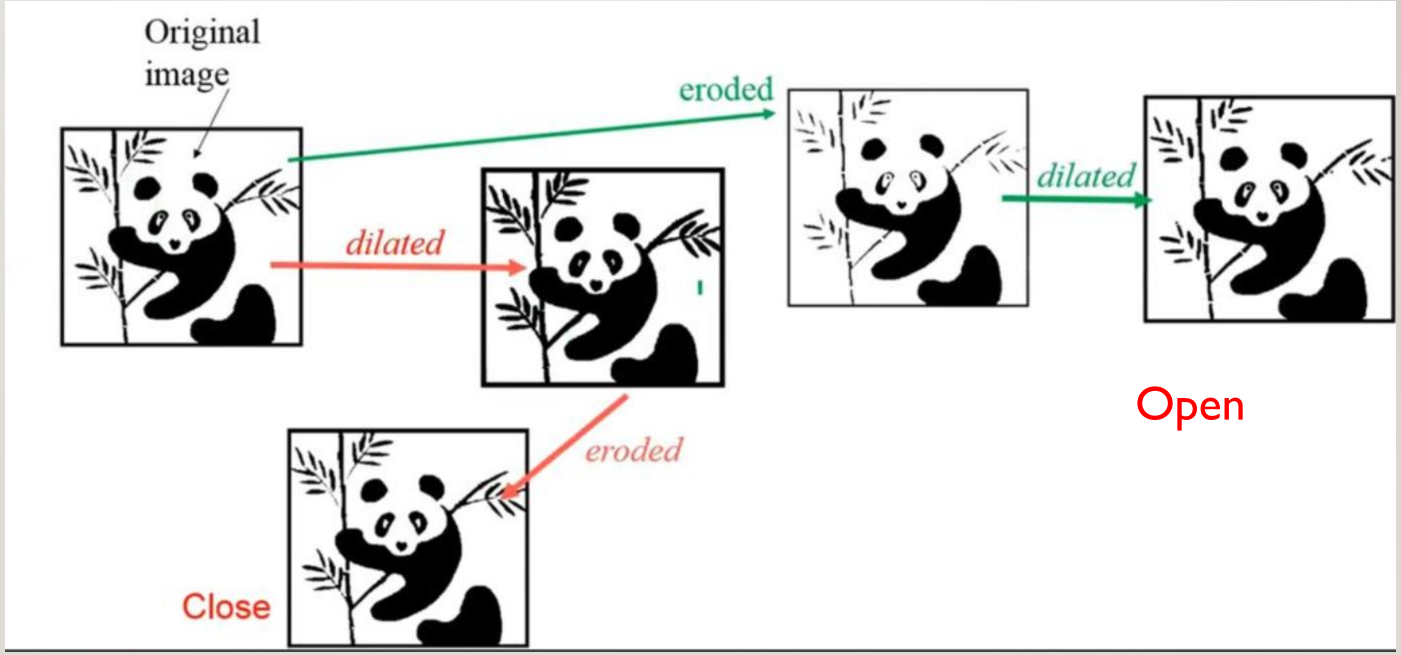

origin
Structuring element
Input Image

Erosion Example


99gr509
Input Image
99gr509
Dilated
99gr509
Eroded

OPENING AND CLOSING OPERATION

- **Opening**=Erosion followed by Dilation
- **Closing**=Dilation followed by Erosion

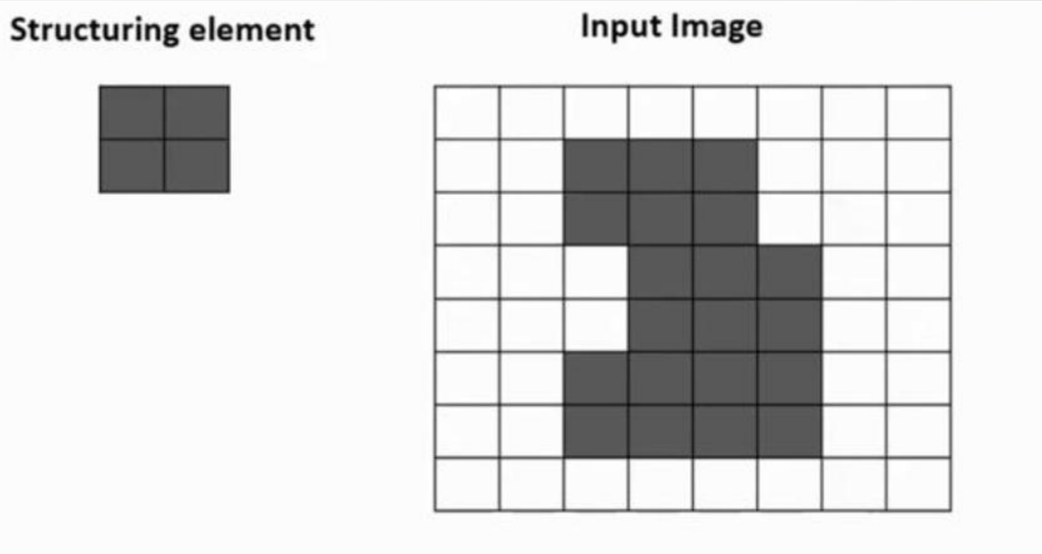


DIFFERENCE

- Erosion and Dilation clean image but leaves objects either smaller or larger than their original size
- Opening and closing perform same functions as erosion and dilation, but object size remains same.

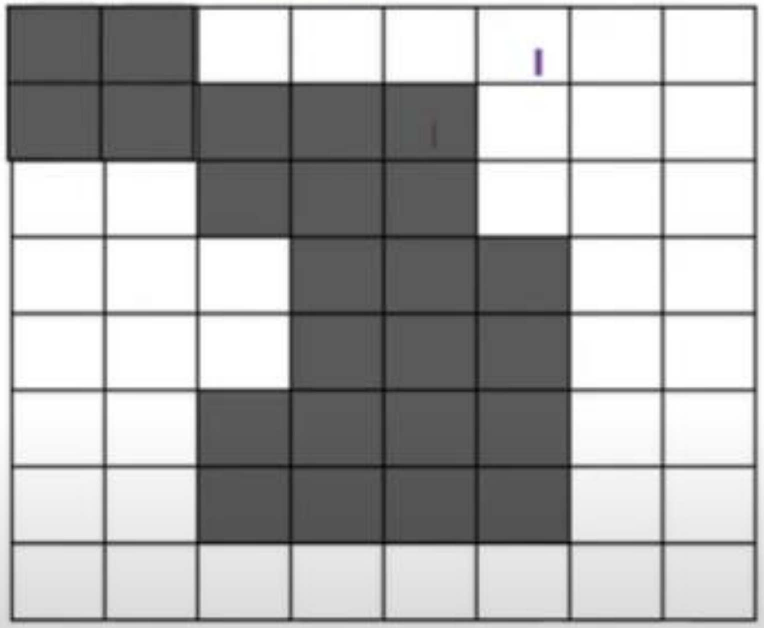
HIT MISS AND FIT TRANSFORMS

- In Image morphology, **Hit or miss** transform is an operation that detects a given pattern in a binary image using a structuring element containing 1's, 0's and blanks for don't cares.

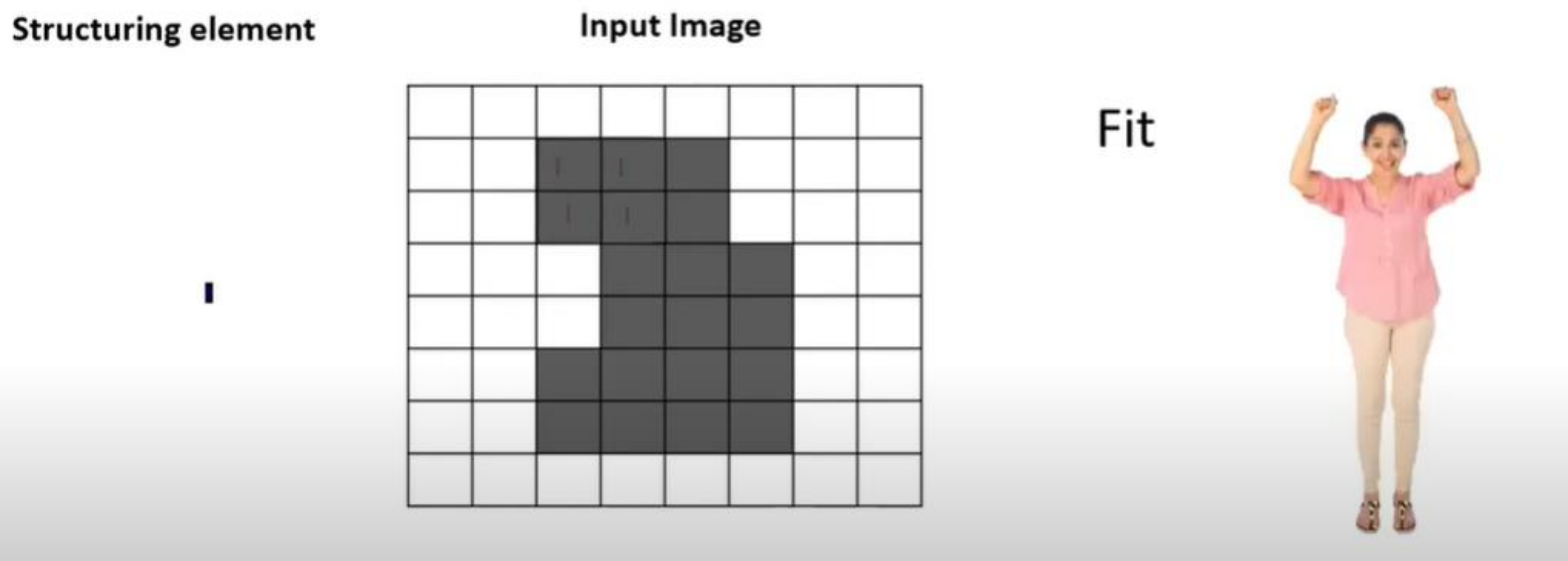


Structuring element

Input Image

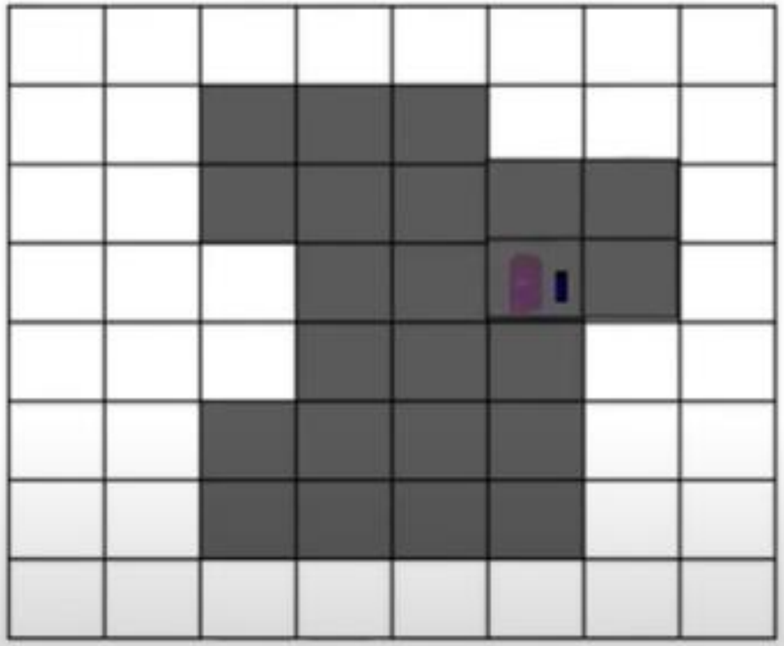


Miss


Structuring element
Input Image
Fit

Structuring element

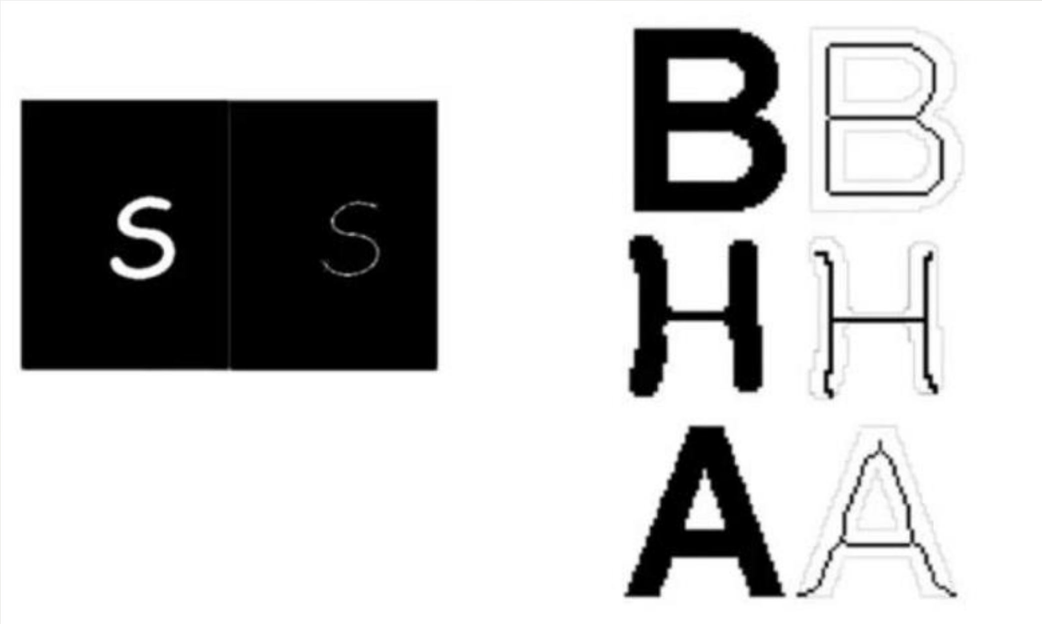
Input Image



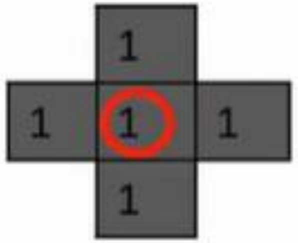
Hit

HIT AND MISS TRANSFORM

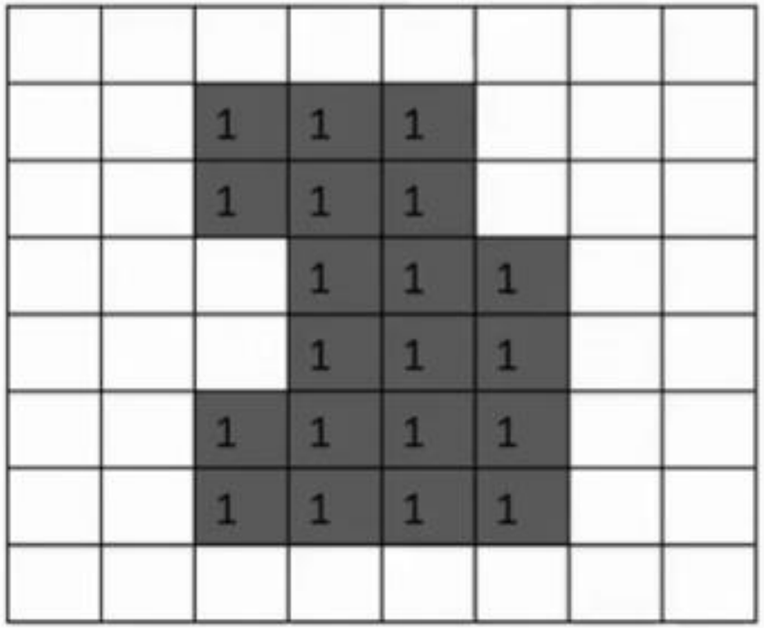
- Hit and Miss alogorithm can be used to thin and skeletonize a shape in a binary image

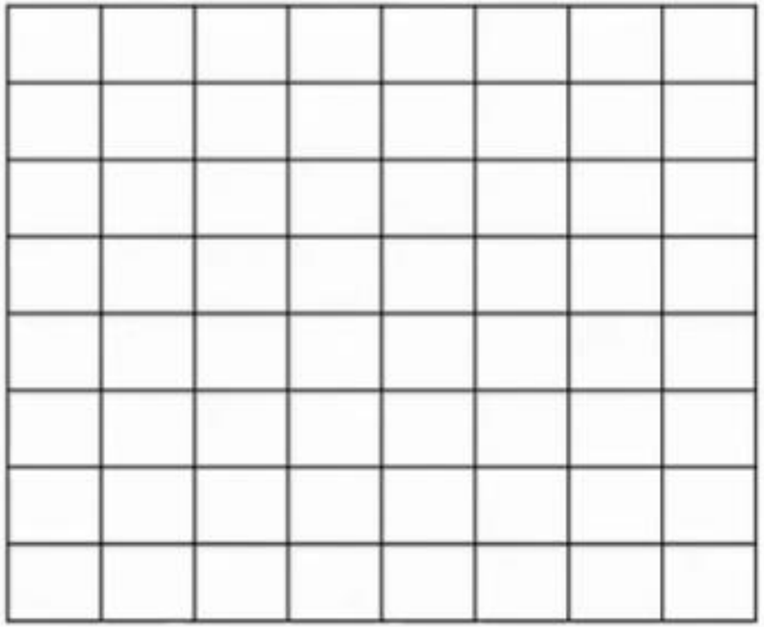


Structuring element



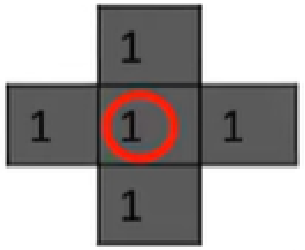
Input Image



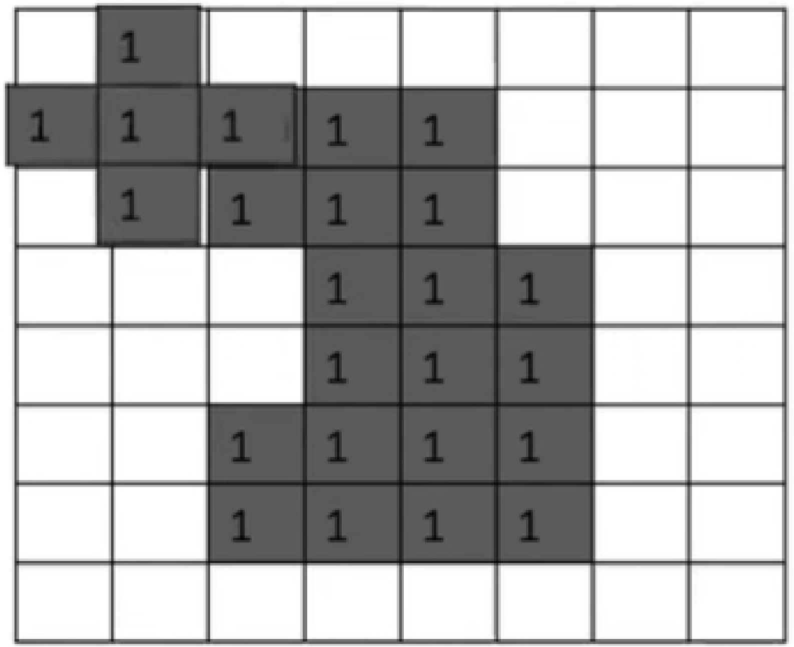


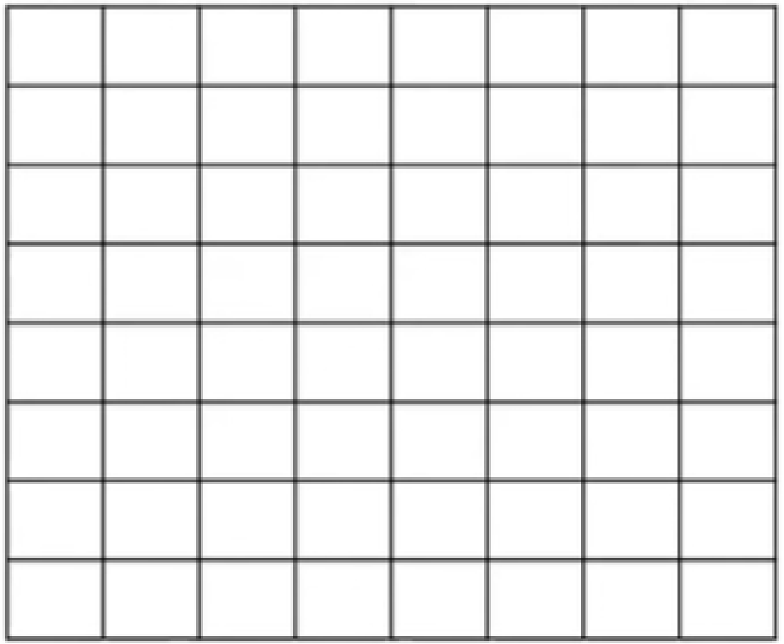
Input Image

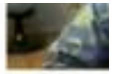
Structuring element



Hit

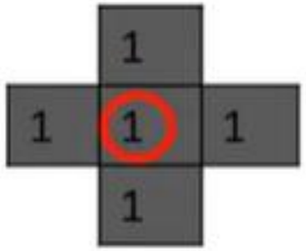




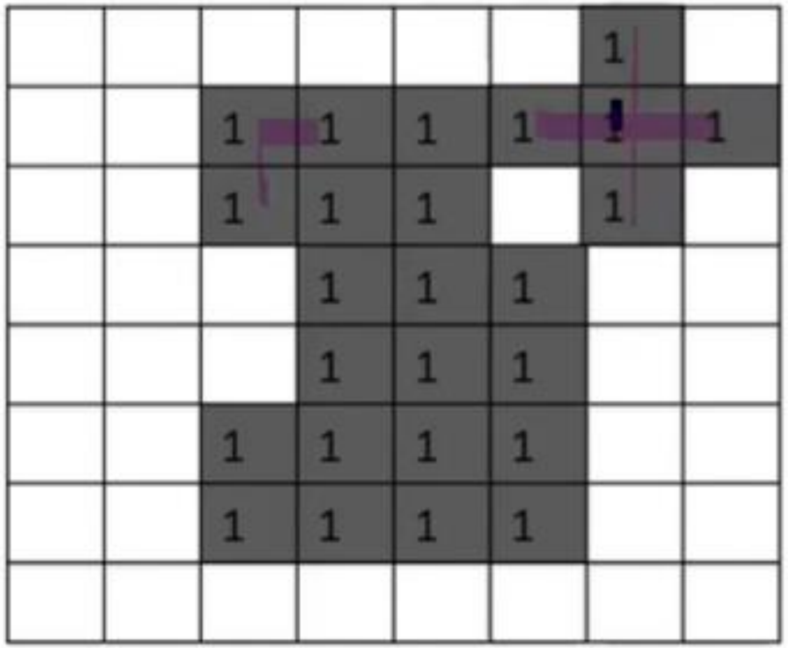


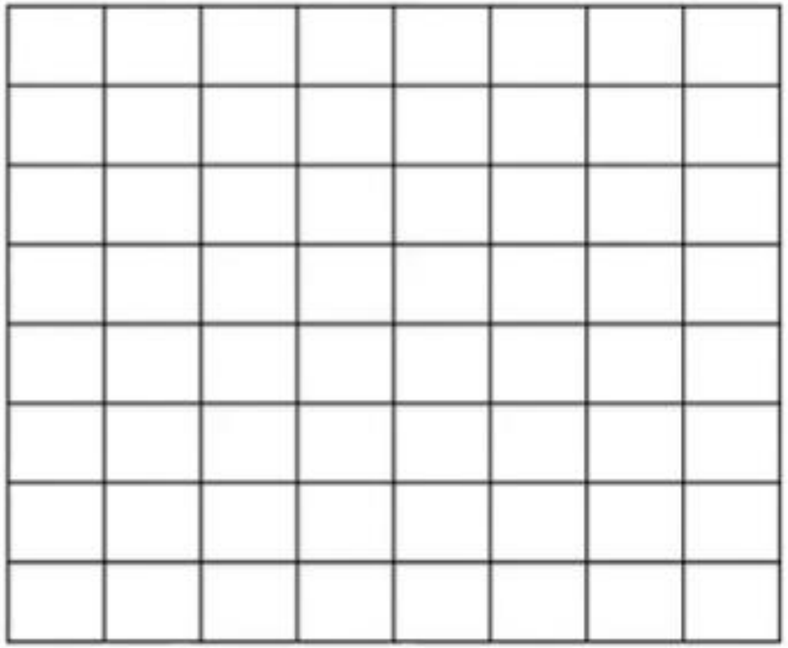
Input Image

Structuring element



Hit miss





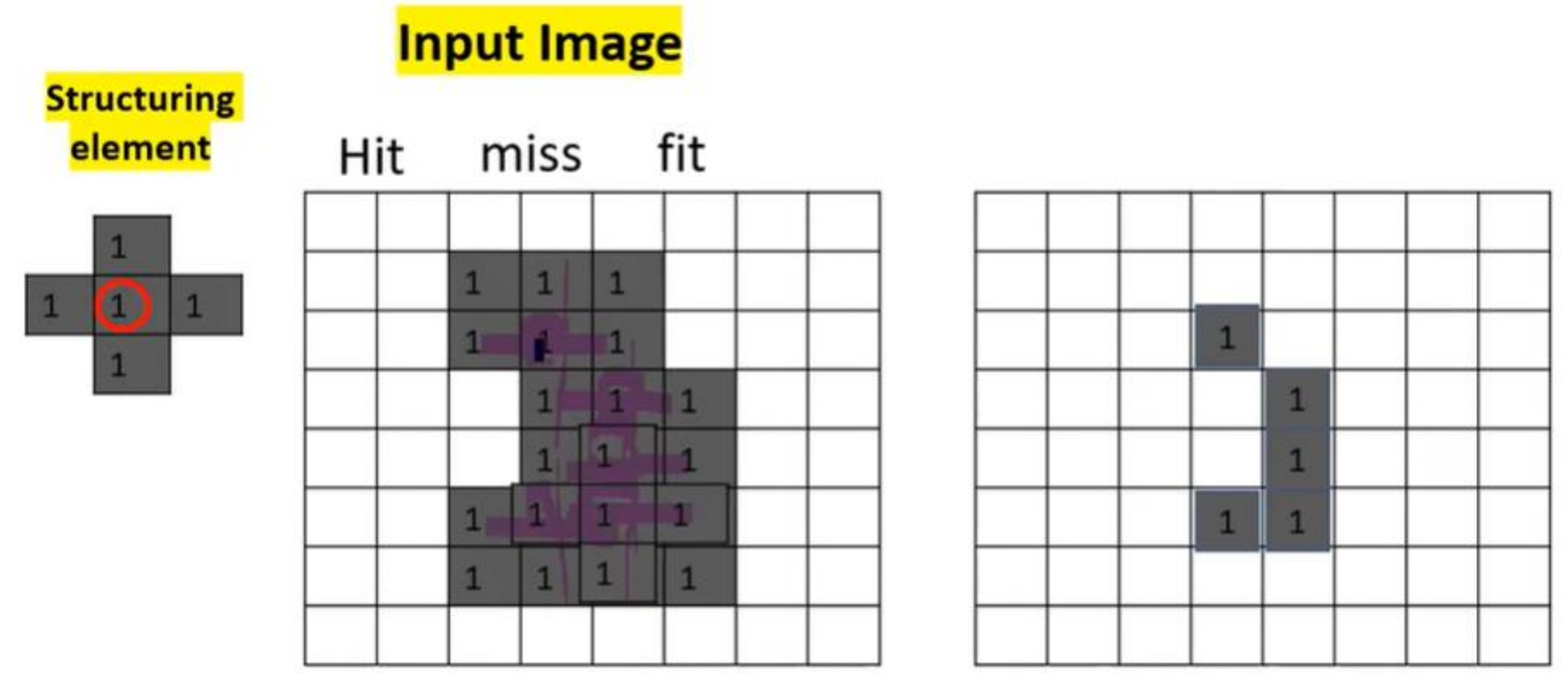
Input Image

Structuring element

	1	
1	1	1
	1	

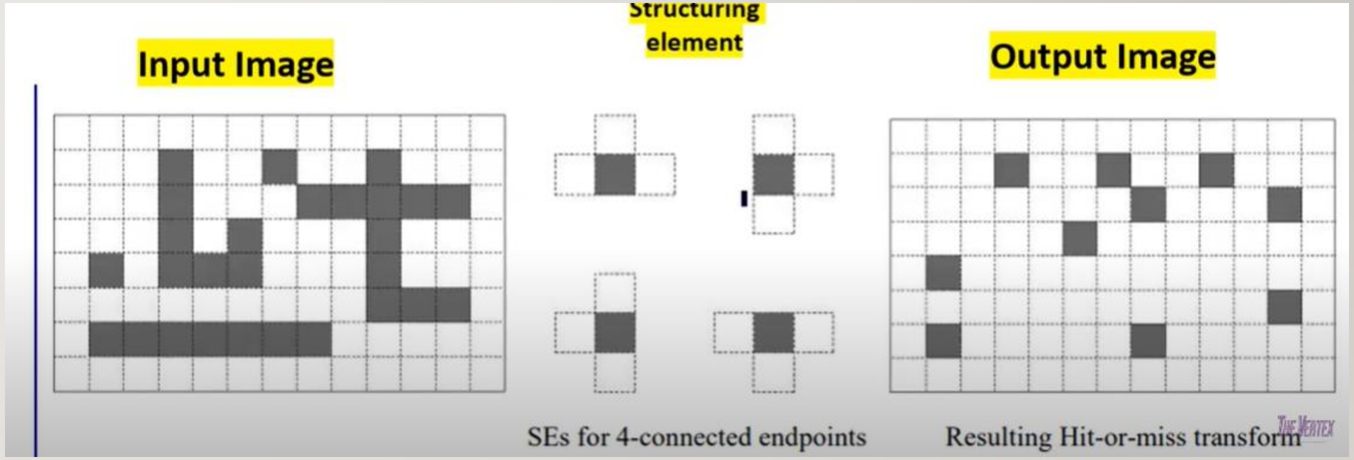
Hit miss fit

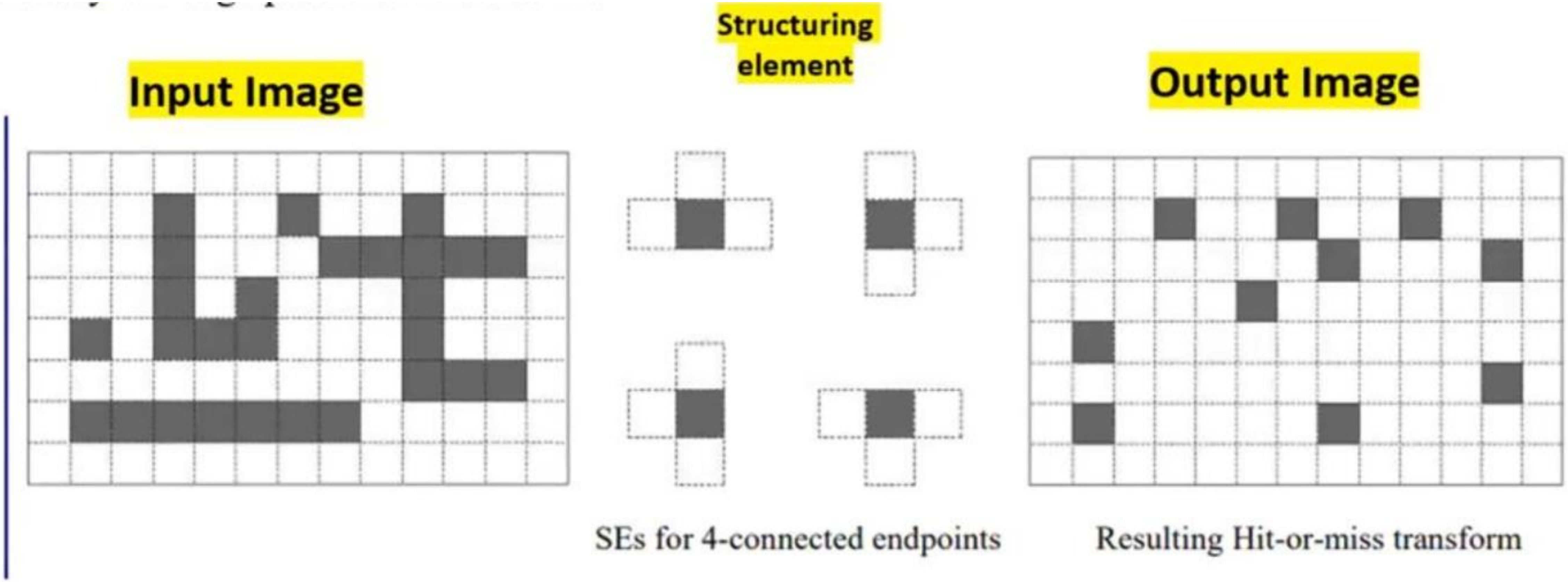
		1	1	1			
		1	1	1			
			1	1	1		
			1	1	1		
		1	1	1	1		
		1	1	1	1		


Input Image
Structuring element
Hit
miss
fit

HIT AND MISS TRANSFORM

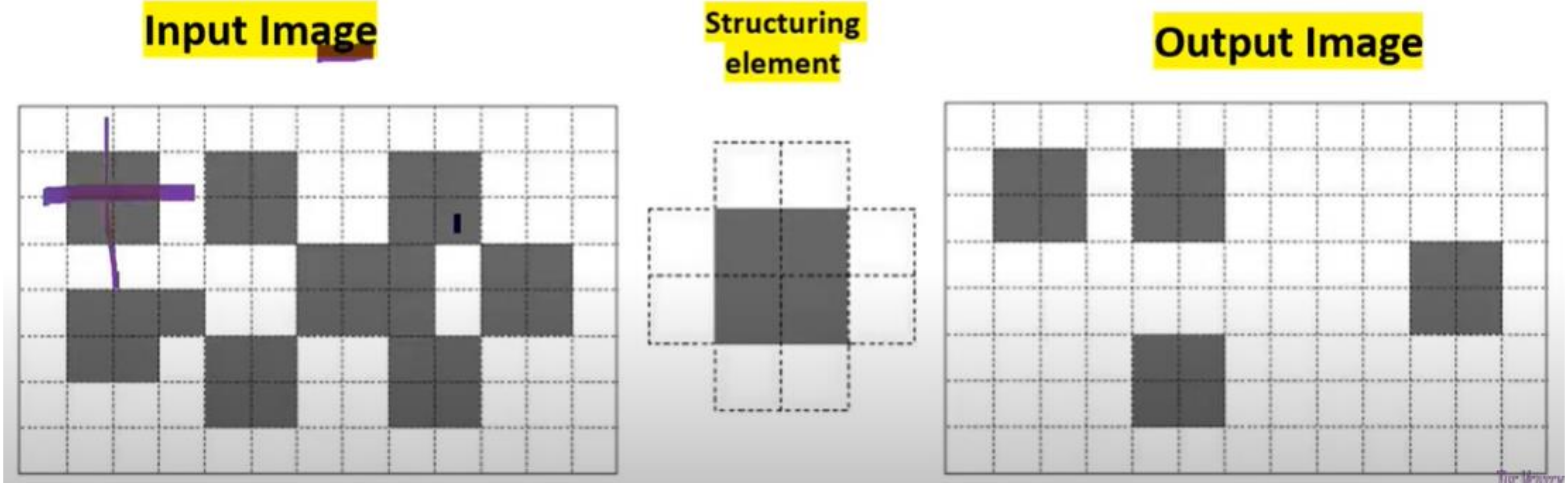
- Hit and Miss transform is an iterative process containing repeated steps to thin the shape by using hit- and- miss method.
- In each iteration, some different structuring elements are used to identify the edge pixels to be removed.
- **Example 1**



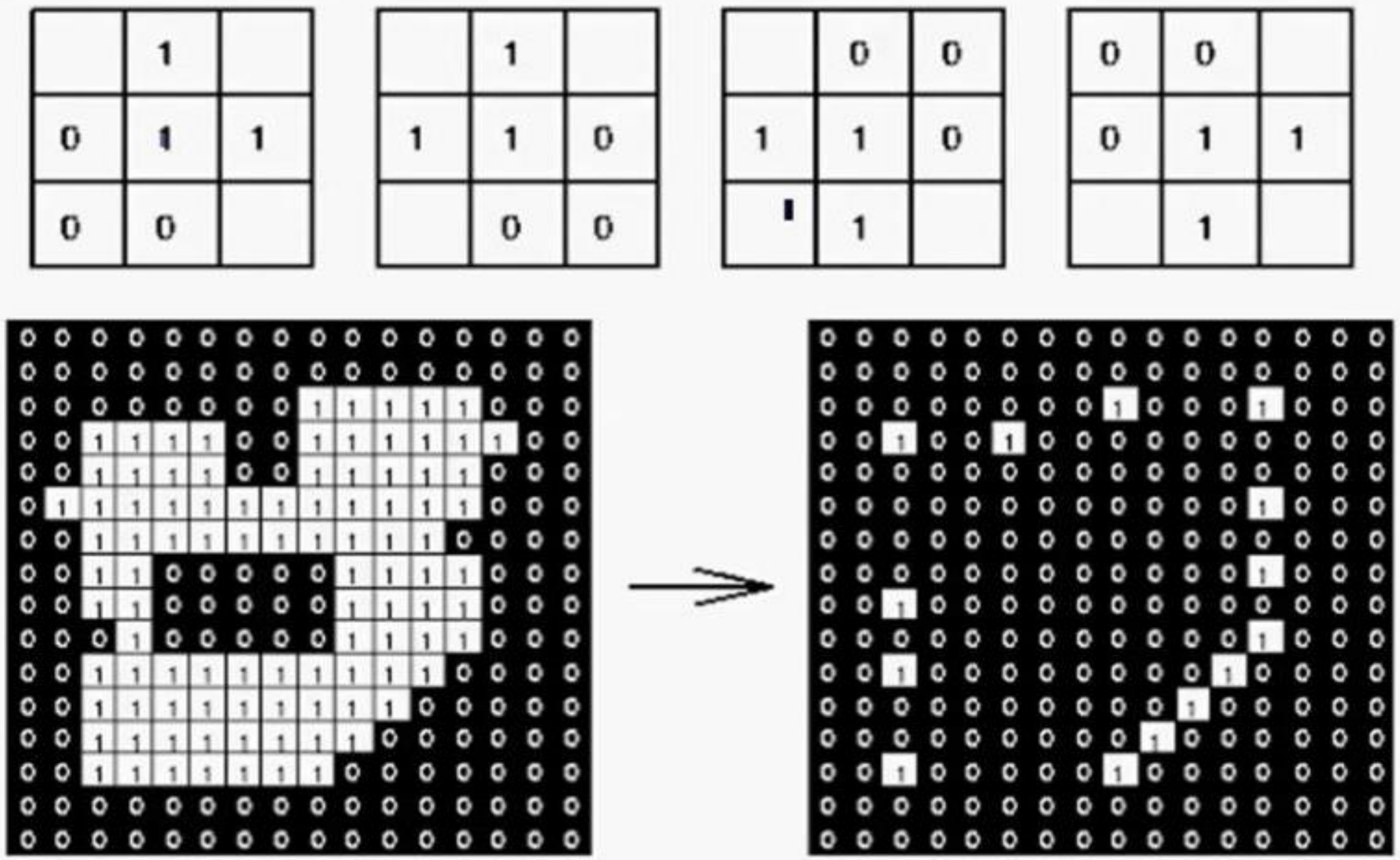


SEs for 4-connected endpoints

Resulting Hit-or-miss transform


Input Image
Structuring element
Output Image

Example 3



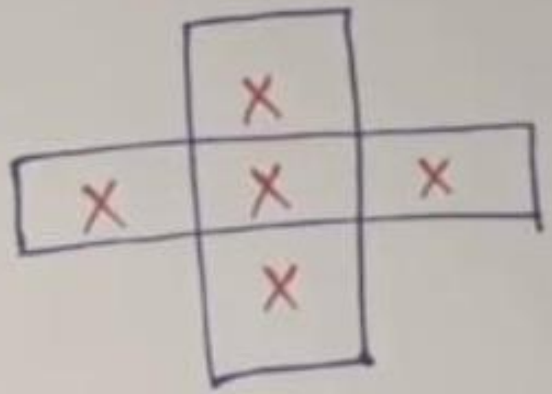
BOUNDARY EXTRACTION ON BINARY IMAGES

- Boundary extraction in image morphology is performed by taking the set difference between an original image and its eroded version,.

$$\mathbf{I-(I \ominus S)}$$

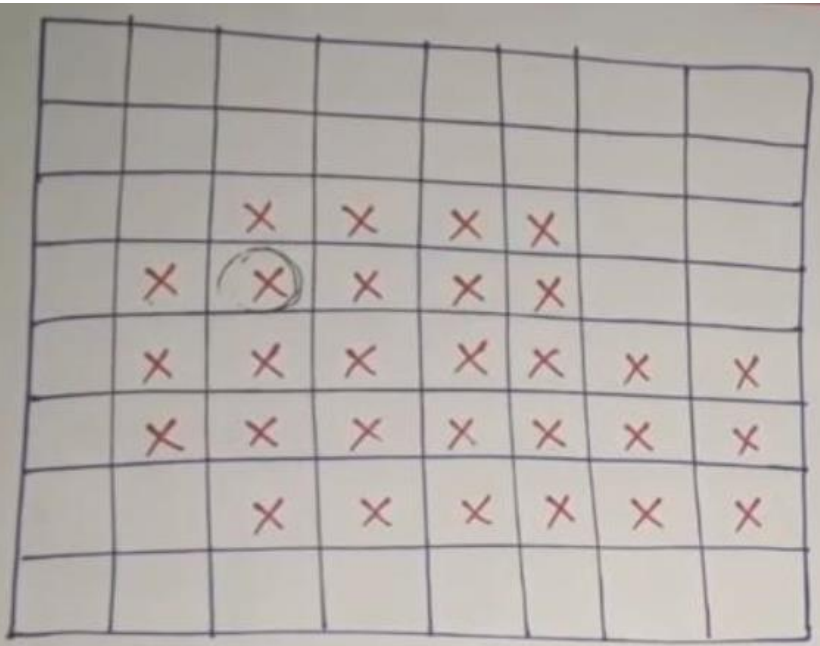
- Where **I** is the original image and **S** is the structuring element.
- This subtractive step leaves only the pixels on the edges of the objects in the original image.
- Larger structuring elements result in thicker boundaries.

Original Image

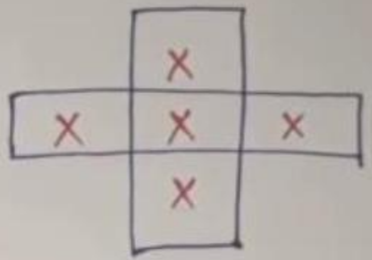


Structuring Element

A

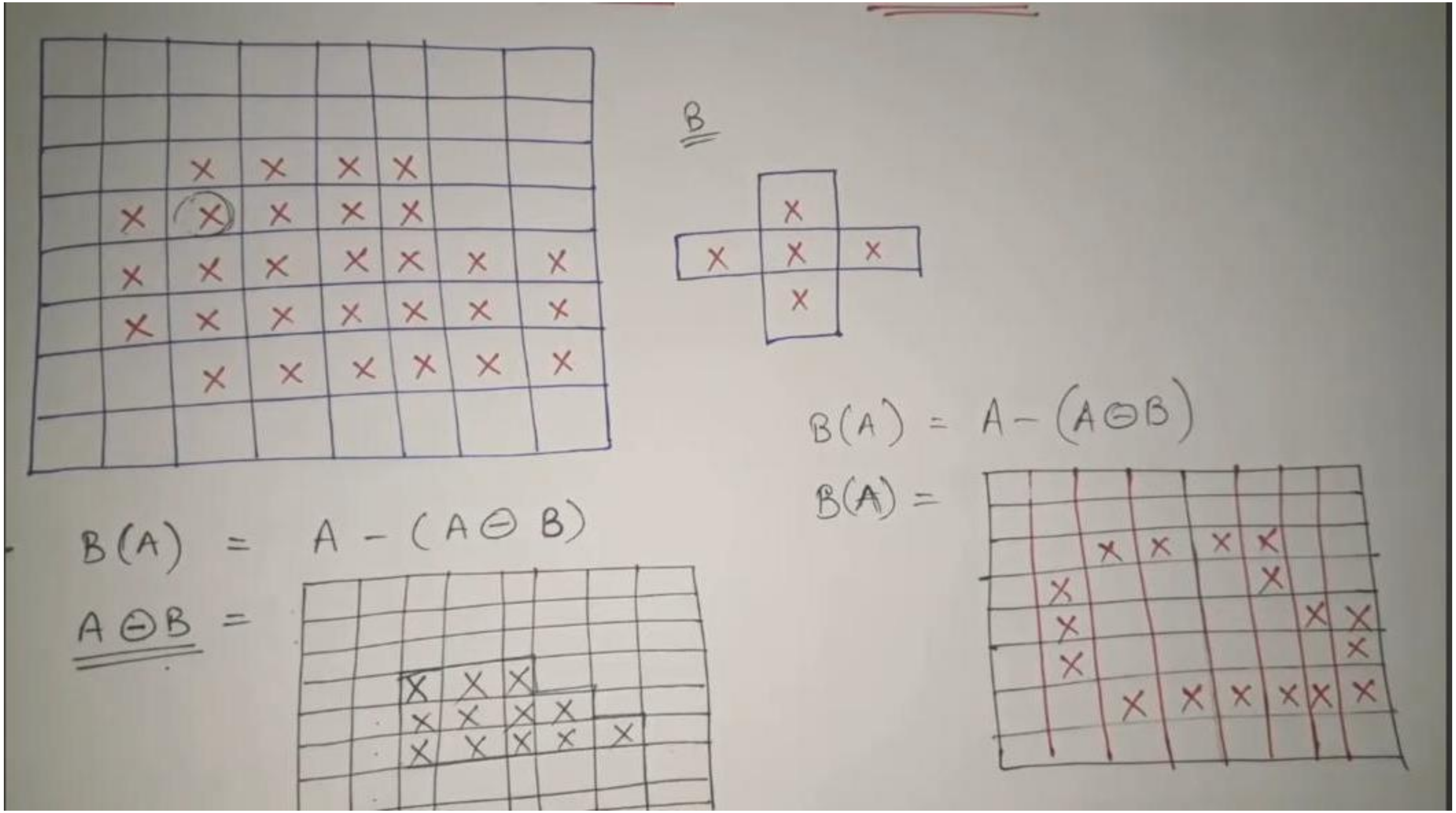


B



$$\Rightarrow \quad B(A) = A - (A \ominus B)$$

$$A \ominus B =$$


B
B(A) = A − (A ⊖ B)
B(A) =
B(A) = A − (A ⊖ B)
A ⊖ B =