Robot Vision

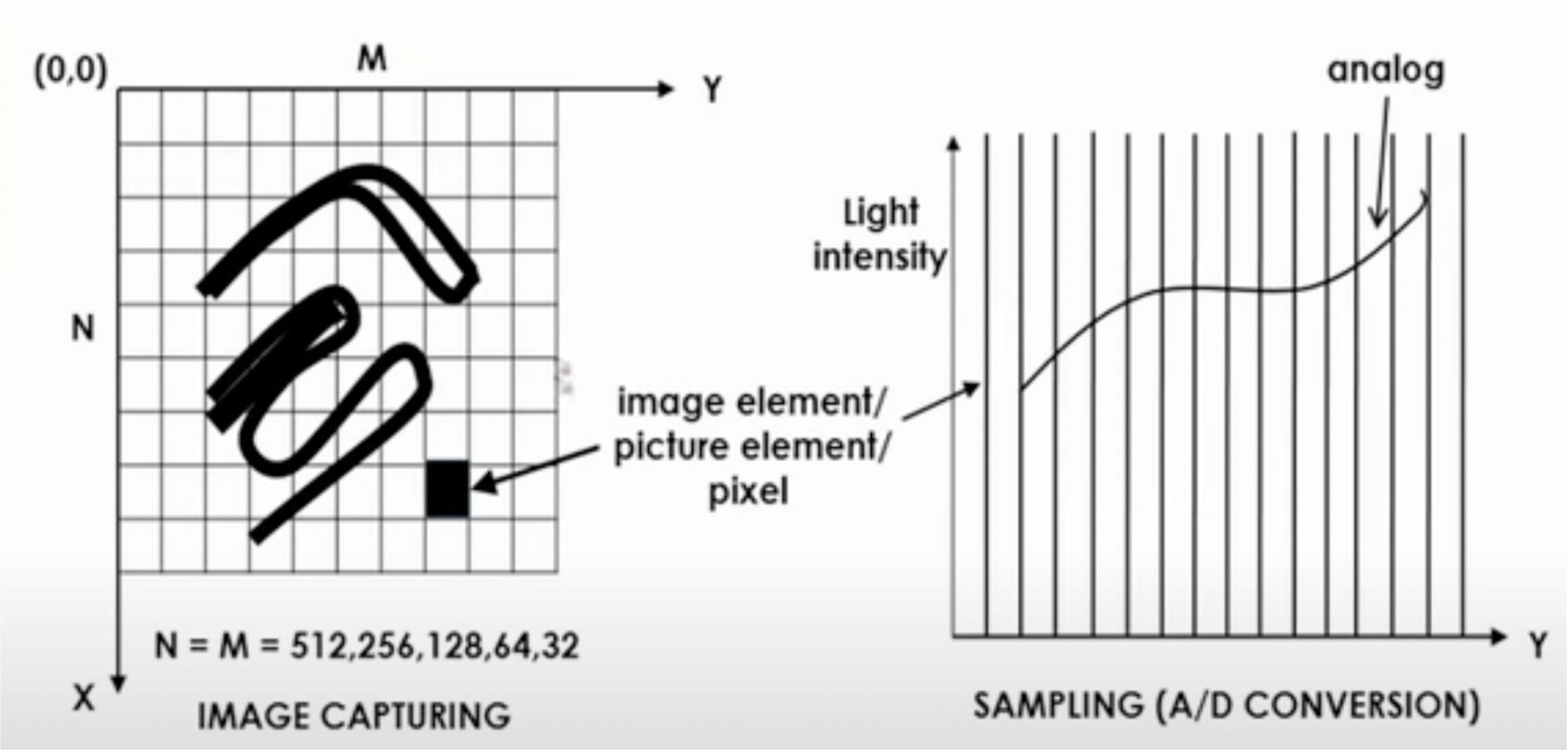
Robot Vision

- To extract, characterize and interpret objects present in an image/photograph captured using a camera.

Steps to be followed

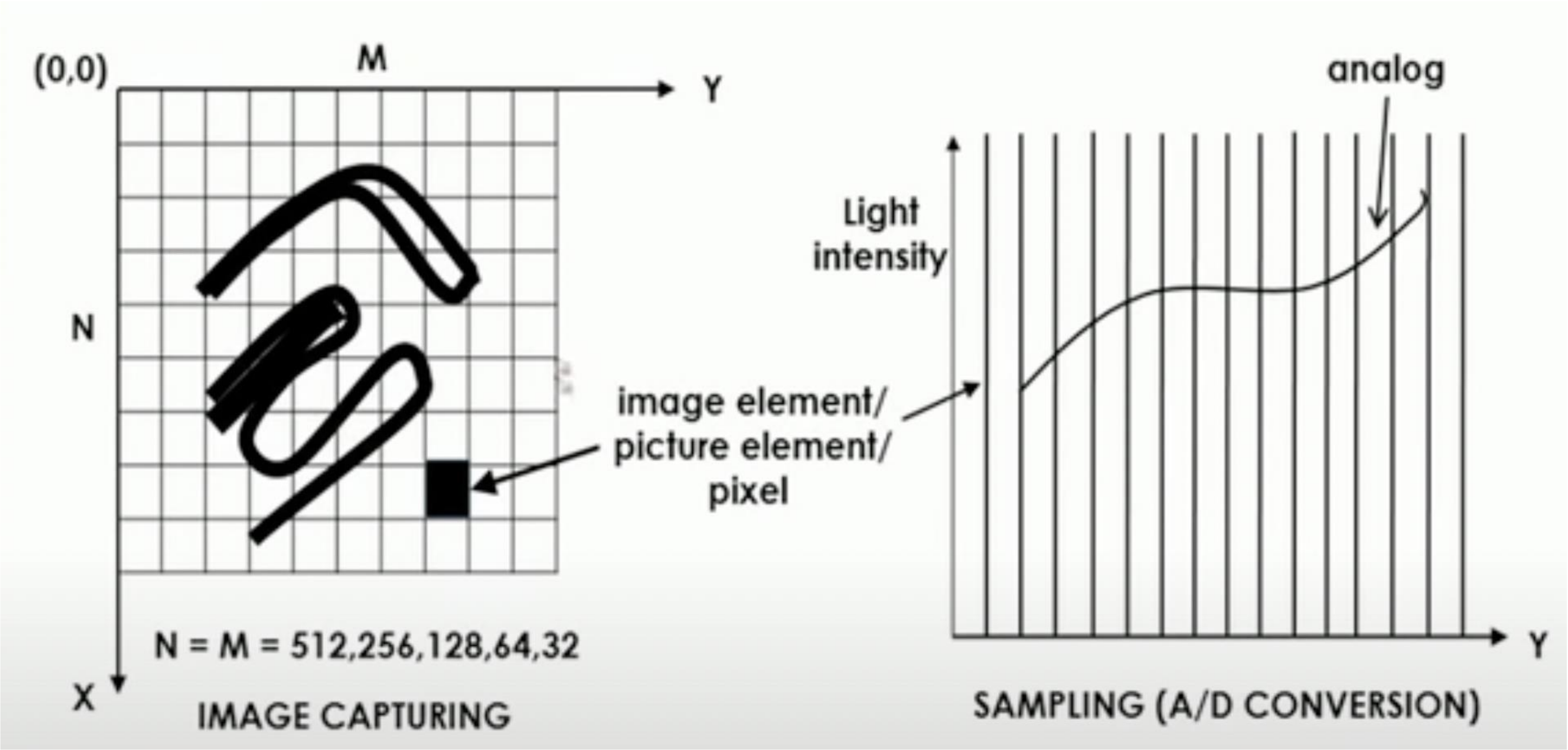
- Step 1: Capturing image of the environment using CCD camera.
-
- Step 2: Light intensity is measured along a particular direction say Y using Electron Beam Scanner (in which the charge accumulated in photo-sites is proportional to light intensity). Analog plot of light intensity is digitized and it is known as A/D conversion or digitizing

Robot vision



Robot vision

	photo-sit es					



Steps to be followed

- Step 3: Image is stored as an array of pixels (each pixel may have different light intensity values). It is known as frame grabbing
- Step 4: Preprocessing of the data collected in Step 3 is done for noise reduction, restoration of lost information etc.

Frame Grabbing

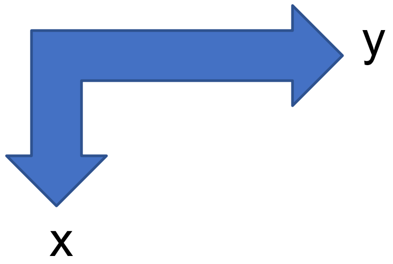
$$\begin{bmatrix} f(0,0) & f(0,1) & ------- & f(0,M-1) \\ f(1,0) & f(1,1) & ------- & f(1,M-1) \\ . & . & ------- & . \\ . & . & ------- & . \\ . & . & ------- & . \\ f(N-1,0) & f(N-1,1) & ------- & f(N-1,M-1) \end{bmatrix}$$

$f(x, y)$: Light intensity of image at the point (x,y)

Methods of Pre-processing

a. Masking

$$\underbrace{P}_{\text{Pre-processed Intensity}}(x, y) = \underbrace{O}_{\text{Operator}}[\underbrace{f(x, y)}_{\text{Input intensity}}]$$



$f(x, y)$:Light intensity value at pixel Q

$f(x-1, y-1)$	$f(x-1, y)$	$f(x-1, y+1)$
$f(x, y-1)$	$Q : f(x, y)$	$f(x, y+1)$
$f(x+1, y-1)$	$f(x+1, y)$	$f(x+1, y+1)$

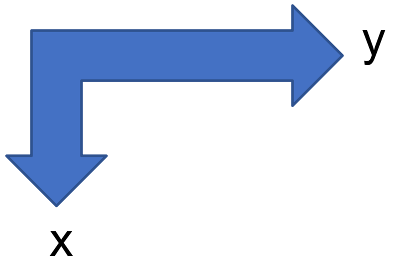
Let us consider the pixel Q having the coordinates (x,y). It has two horizontal and two vertical and four diagonal neighbors

- Let us consider a 3*3 mask with coefficients W_1, W_2, W_3, W_4, W_5, W_6, W_7, and W_8

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

-1	-1	-1
-1	$+8$	$-$
-1	-1	-1

Example of a 3*3 mask



$f(x, y)$:Light intensity value at pixel Q

$f(x-1, y-1)$	$f(x-1, y)$	$f(x-1, y+1)$
$f(x, y-1)$	$Q : f(x, y)$	$f(x, y+1)$
$f(x+1, y-1)$	$f(x+1, y)$	$f(x+1, y+1)$

x

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Let us consider the pixel Q having the coordinates (x,y). It has two horizontal and two vertical and four diagonal neighbors

$$P(x,y) = O[f(x,y)]$$

$$\begin{aligned}
&= W_1 f(x-1,y-1) + W_2 f(x-1,y) + W_3 f(x-1,y+1) + W_4 f(x,y-1) \\
&\quad + W_5 f(x,y) + W_6 f(x,y+1) + W_7 f(x+1,y-1) + W_8 f(x+1,y) + W_9 f(x+1,y+1)
\end{aligned}$$

• b. Neighborhood Averaging

Here $P(x, y)$ is calculated by averaging the intensity values of the pixels contained in a pre-defined neighborhood of $f(x, y)$.

$$P(x, y) = \frac{1}{R} \sum_{(n,m) \in S} f(n, m)$$

Where, S is the set of pixels lying in the neighbourhood of (x,y) including itself and R is the total number of neighbourhood pixels including itself

- **C. Median Filtering**

To determine pre-processed light intensity value of a pixel Q, we consider light intensity values of all its neighboring pixels including itself. We sort light intensity values in the ascending order say, and then determine the median value. This median value is going to replace the intensity value at Q.

- **Step 5: Thresholding**

To get clear distinction between objects and the background, let T be the threshold intensity

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

For the black background and white object, 1 corresponds to objects and 0 indicates the background

- **Step 6**: Edge detection
- To detect the edge of an object
- Gradient operator

$$G[p(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix}$$

- **Masks used for Gradient operator**

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

$$G_x = \frac{\partial p}{\partial x}$$

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

$$G_y = \frac{\partial p}{\partial y}$$

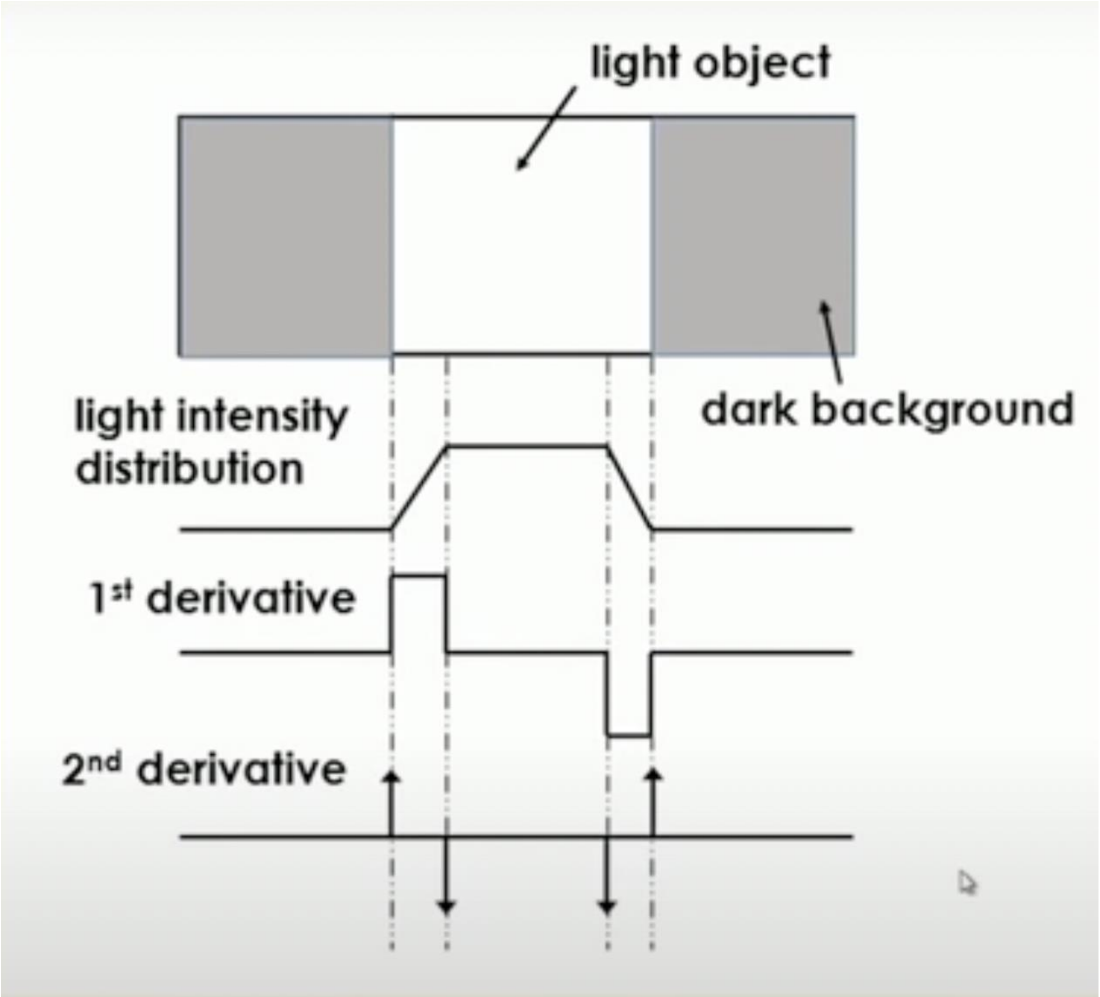
- **Laplace operator**

$$L[p(x,y)] = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}$$

0	1	0
1	-4	1
0	1	0

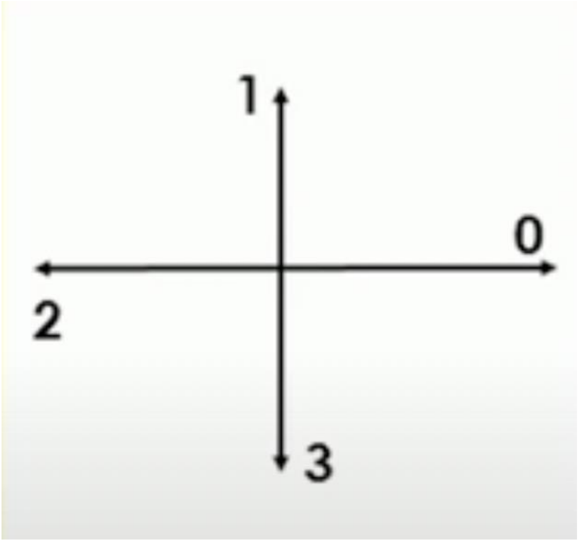
- **Edge Detection(Contd.)**

- **Light object on dark background**

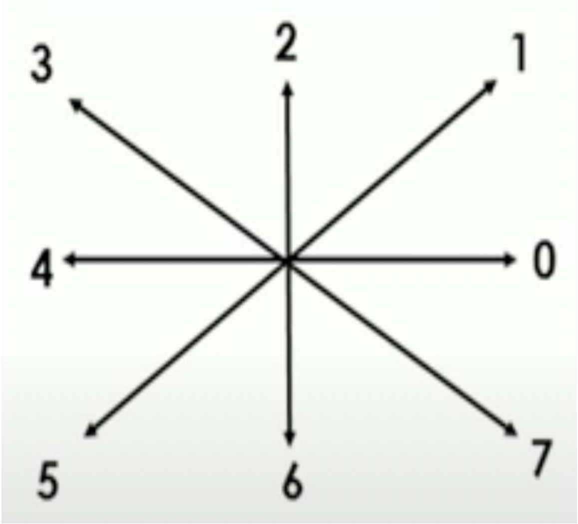


Boundary Descriptors

- **Chain Codes:**-used to represent the boundary of an object by a set of straight line segments of specified length and direction

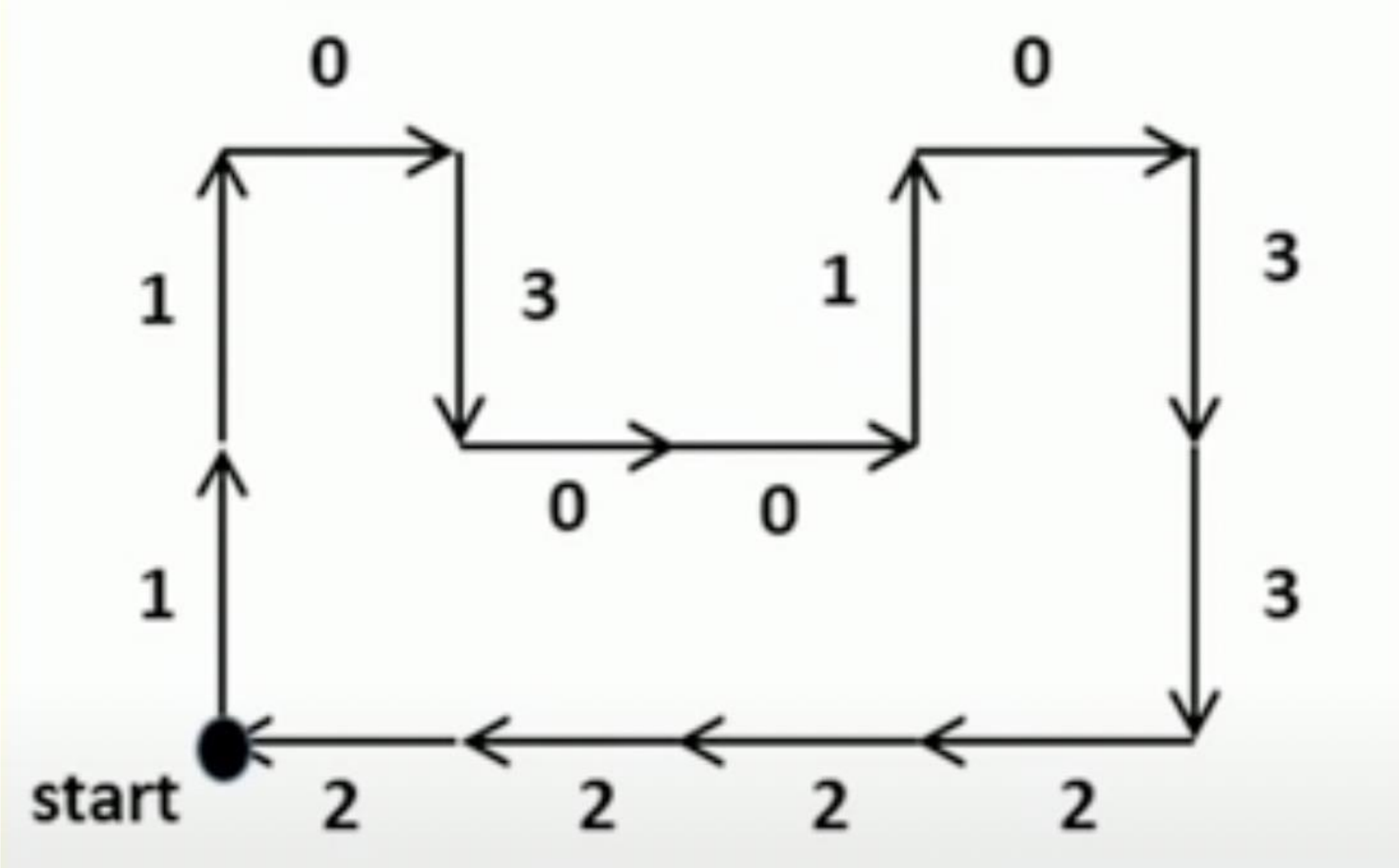


4-directional chain code



8-directional chain code

Example:

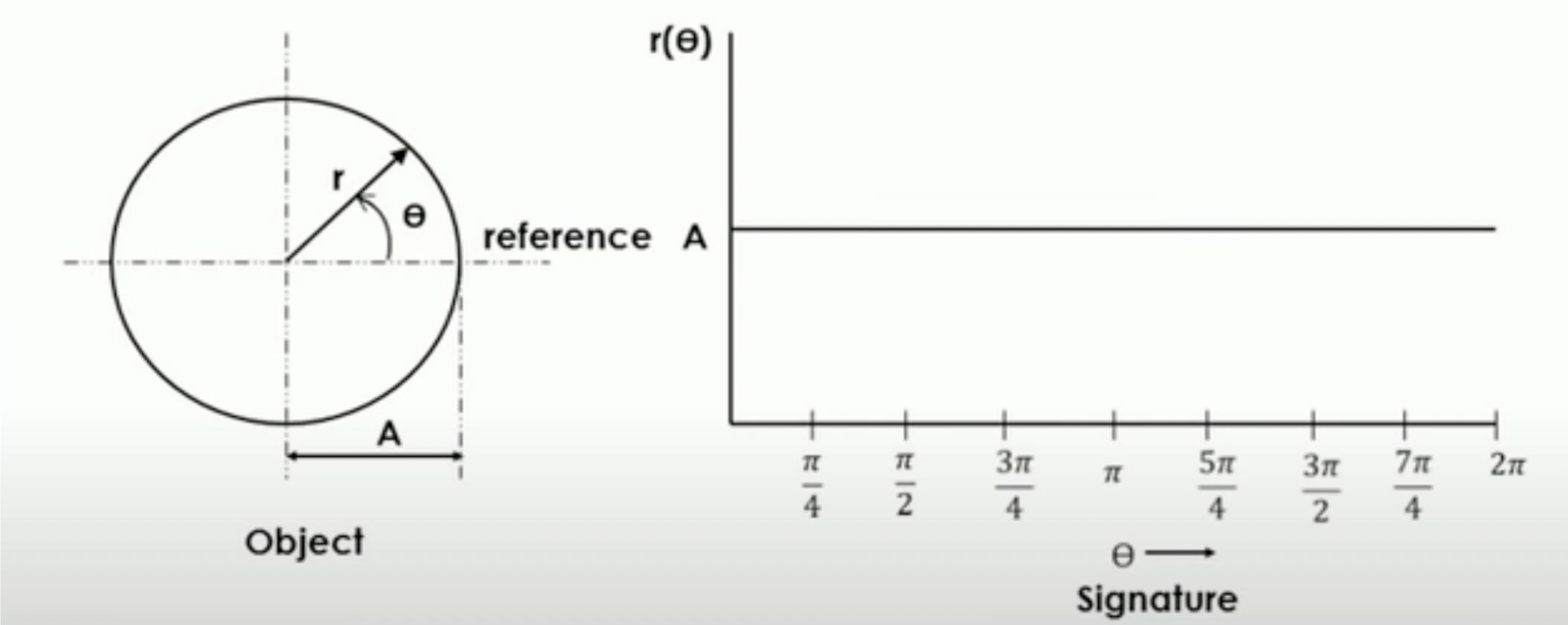


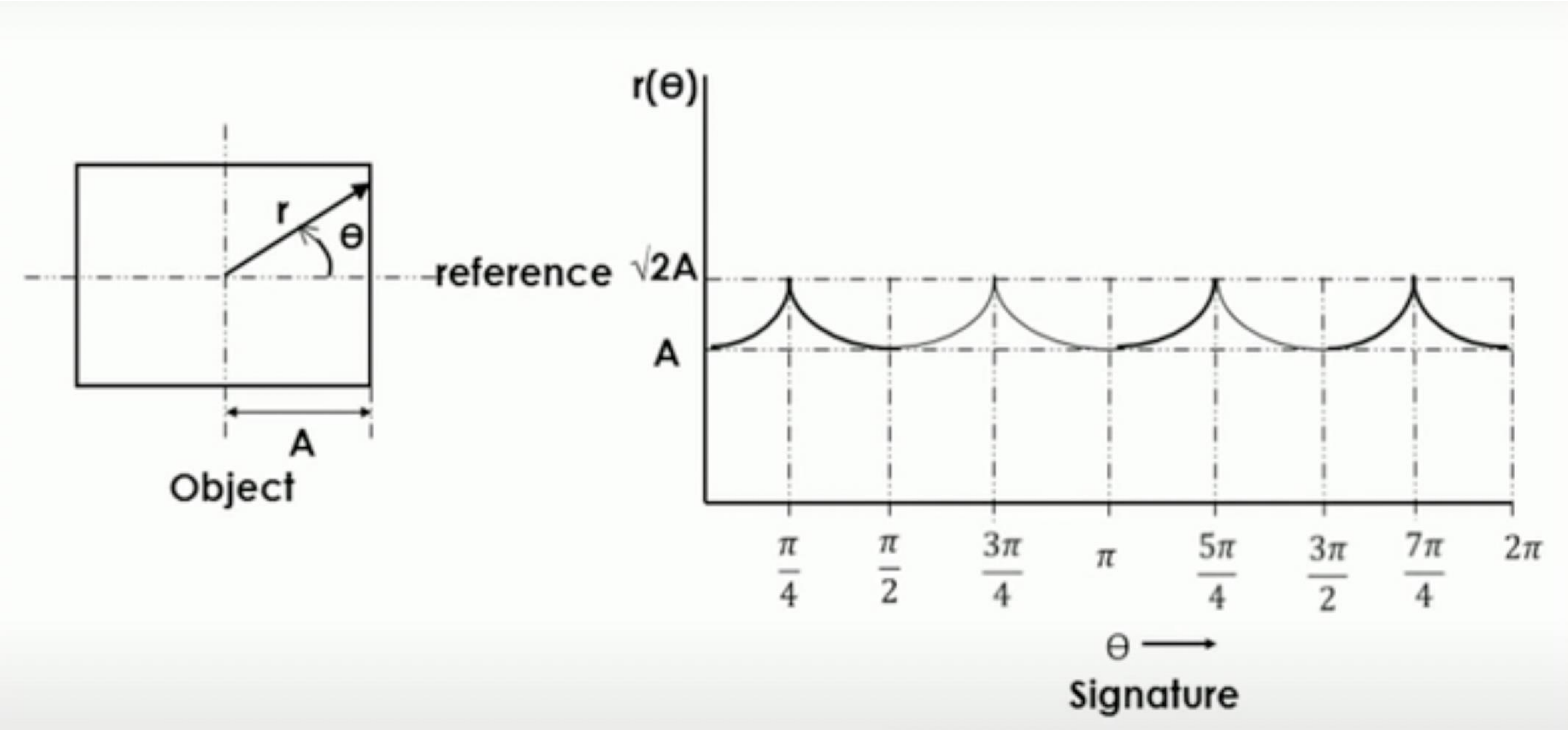
Chain code: 11030010332222

Signature

- One dimensional functional representation of a boundary

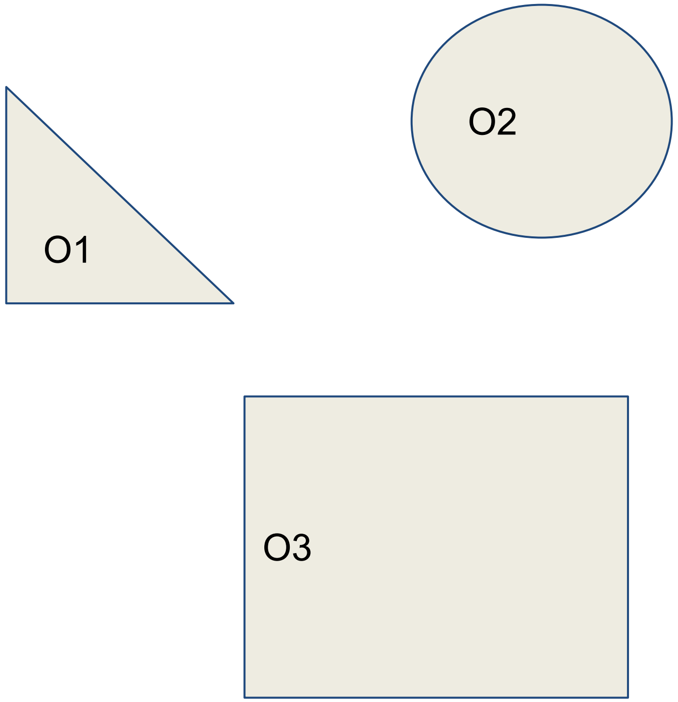
Examples:





Note: To identify multiple objects present in an image, we consider

$$\text{Compactness} = (\text{perimeters}^2 / area)$$



$$\text{Compactness} = (\text{perimeters}^2 / area)$$

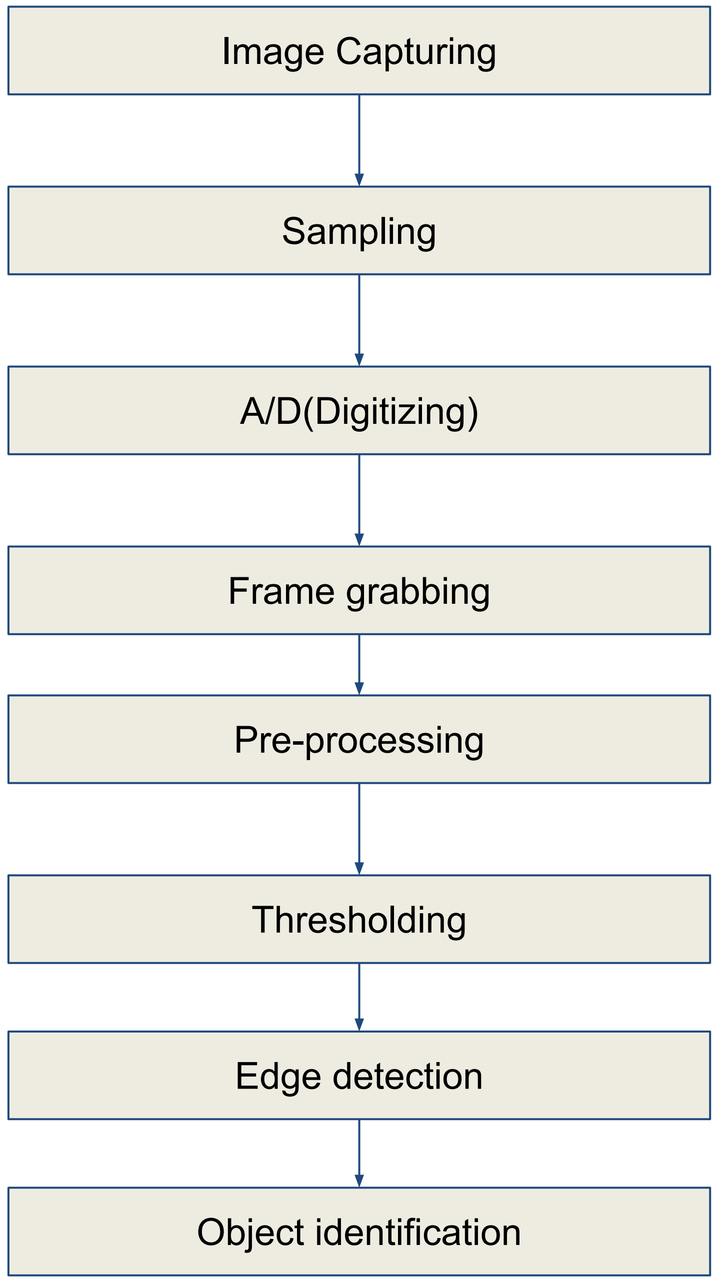


Image Capturing
Sampling
A/D(Digitizing)
Frame grabbing
Pre-processing
Thresholding
Edge detection
Object identification