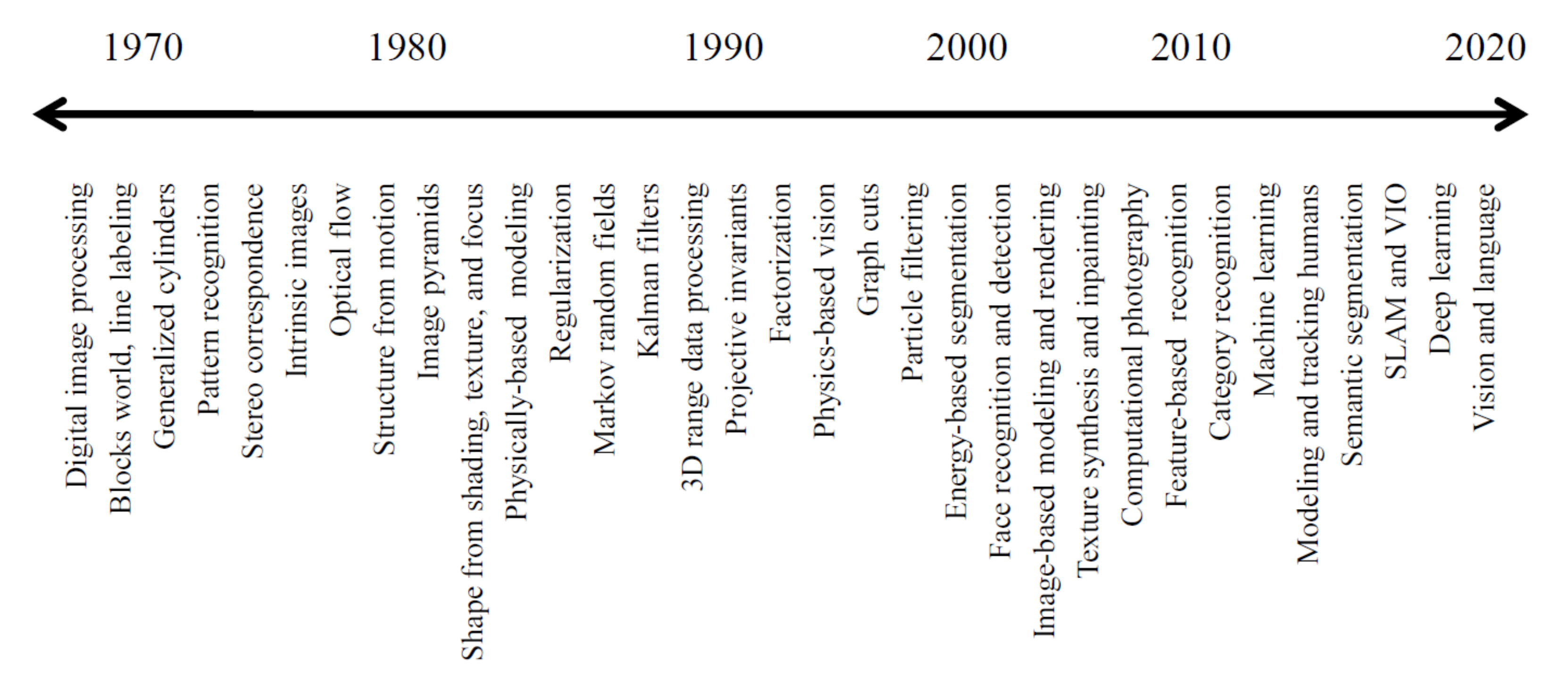
# 22AIE313 Computer Vision & Image Understanding (2-1-3-4)

Dr. Aiswarya S Kumar
Dept. of Computer Science & Engineering (AI)
Amrita School of Computing

# What is computer vision?

- The human visual system has no problem interpreting the subtle variations in this photograph.
- As humans, we can correctly segment the object from its background.
- “Computer vision is a field of artificial intelligence (AI) that allows computers to see and understand the visual world.”




*A rough timeline of some of the most active topics of research in computer vision.*

# Geometric primitives

- Geometric primitives form the basic building blocks used to describe 3D shapes.

## 2D points

- 2D points (pixel coordinates in an image) can be denoted using a pair of values,

  x = ($x$, $y$) ϵ $R^2$
- 2D points can also be represented using homogeneous coordinates,

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in \mathcal{P}^2$$

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\mathbf{x}}$$

Projective Geometry

# Geometric primitives

## 2D lines

2D lines can also be represented using homogeneous coordinates $\tilde{\mathbf{l}} = (a, b, c)$

The corresponding line equation is

$$\bar{\mathbf{x}} \cdot \tilde{\mathbf{l}} = ax + by + c = 0$$

We can normalize the line equation vector so that

$$\mathbf{l} = (\hat{n}_x, \hat{n}_y, d) = (\hat{\mathbf{n}}, d) \qquad \|\hat{\mathbf{n}}\| = 1$$

n ^ is the normal vector perpendicular to the line and d is its distance to the origin.

# Geometric primitives

When using homogeneous coordinates, we can compute the intersection of two lines as

$$\tilde{\mathbf{x}} = \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2$$

Similarly, the line joining two points can be written as

$$\tilde{\mathbf{l}} = \tilde{\mathbf{x}}_1 \times \tilde{\mathbf{x}}_2$$

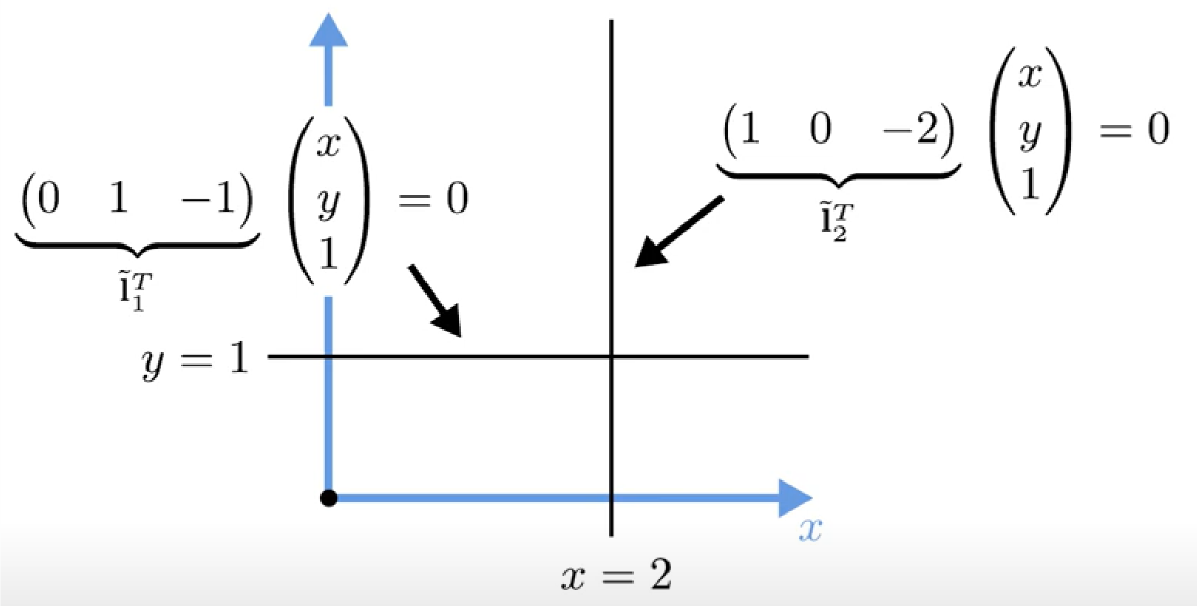
where  is the cross product operator

# Geometric primitives

Cross product expressed as product of skew-symmetric matrix and a vector

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

Note: We have used square brackets to represent matrix [ ] and normal brackets ( ) to represent vectors.



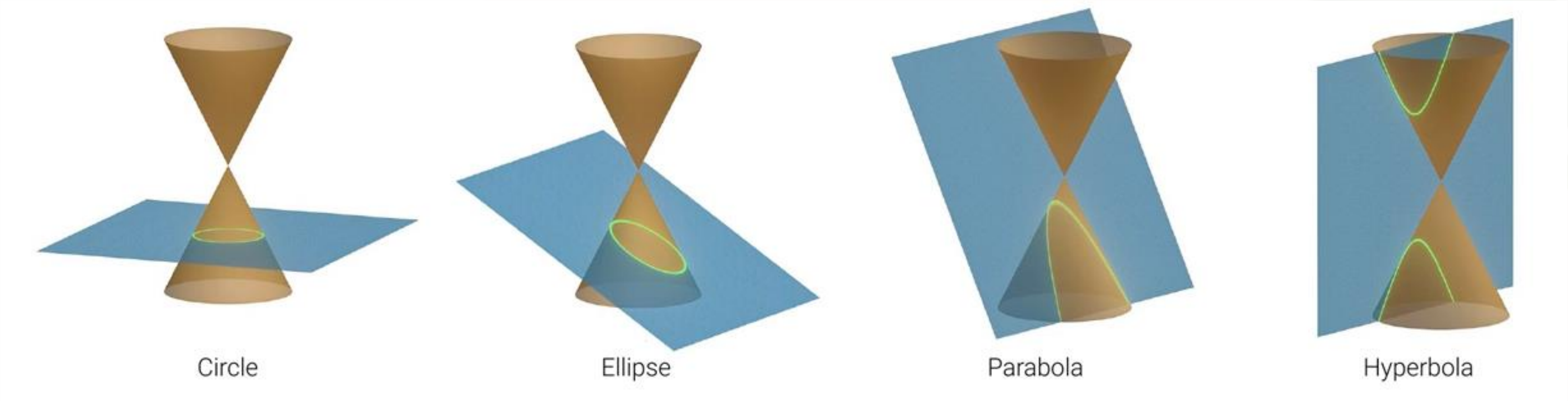

$$[\mathbf{x}]_\times = \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix}$$

$$\tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2 = [\tilde{\mathbf{l}}_1]_\times \tilde{\mathbf{l}}_2 = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

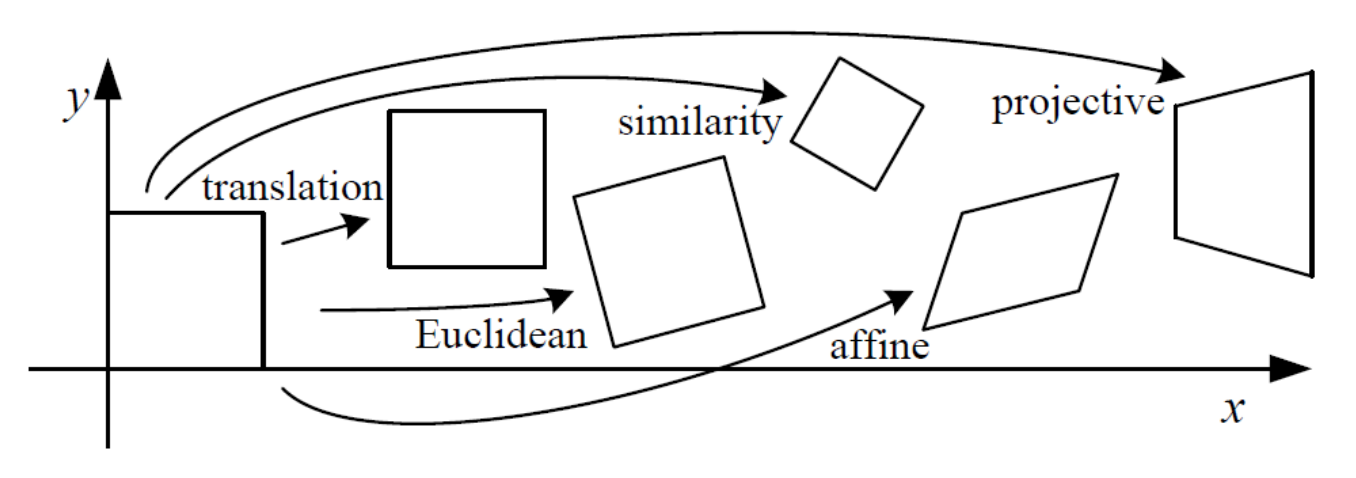
# Geometric primitives

## 2D conics

There are other algebraic curves that can be expressed with simple polynomial homogeneous equations.



# 2D transformations



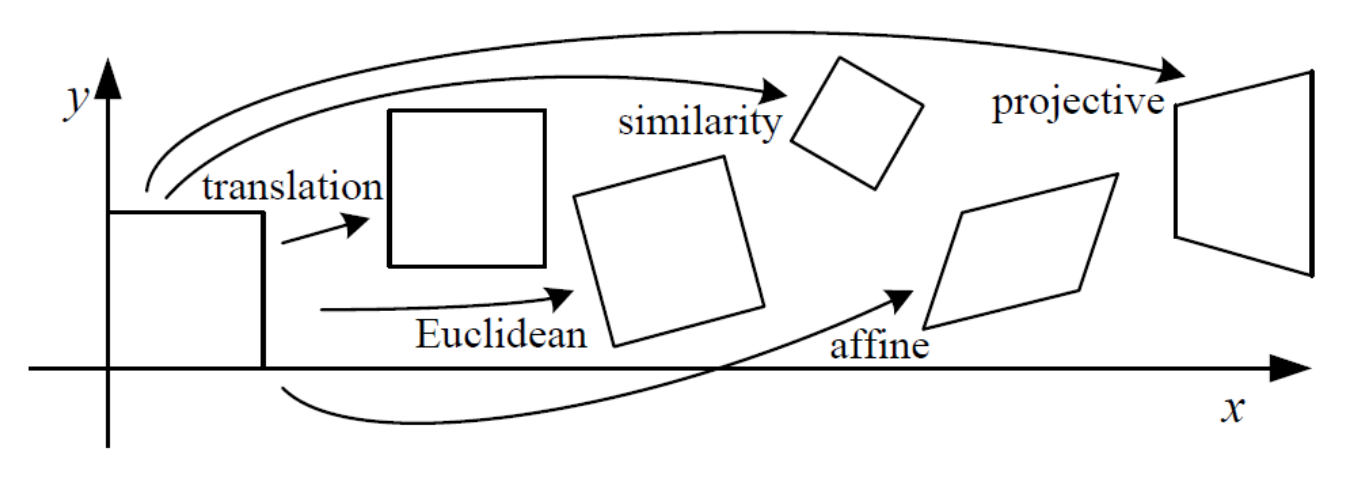

**Translation (2 DoF)**:
2D translations can be written as x' = x + t

$$\bar{\mathbf{x}}' = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \bar{\mathbf{x}}$$

where I is the (2 X 2) identity matrix.

OR $$\mathbf{x}' = \begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix} \bar{\mathbf{x}}$$

# 2D transformations



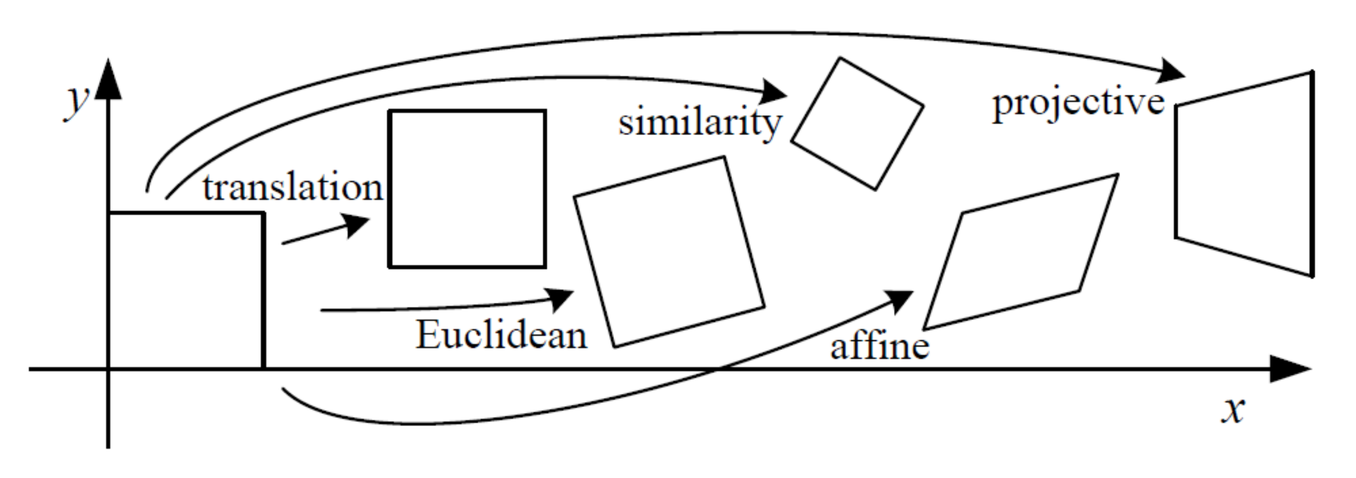

**Euclidean (3 DoF)**:
Rotation + translation
x' = Rx + t

$$\bar{\mathbf{x}}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \bar{\mathbf{x}}$$

where R is the (2 X 2) orthonormal rotation matrix.

OR $$\mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \bar{\mathbf{x}}$$

# 2D transformations



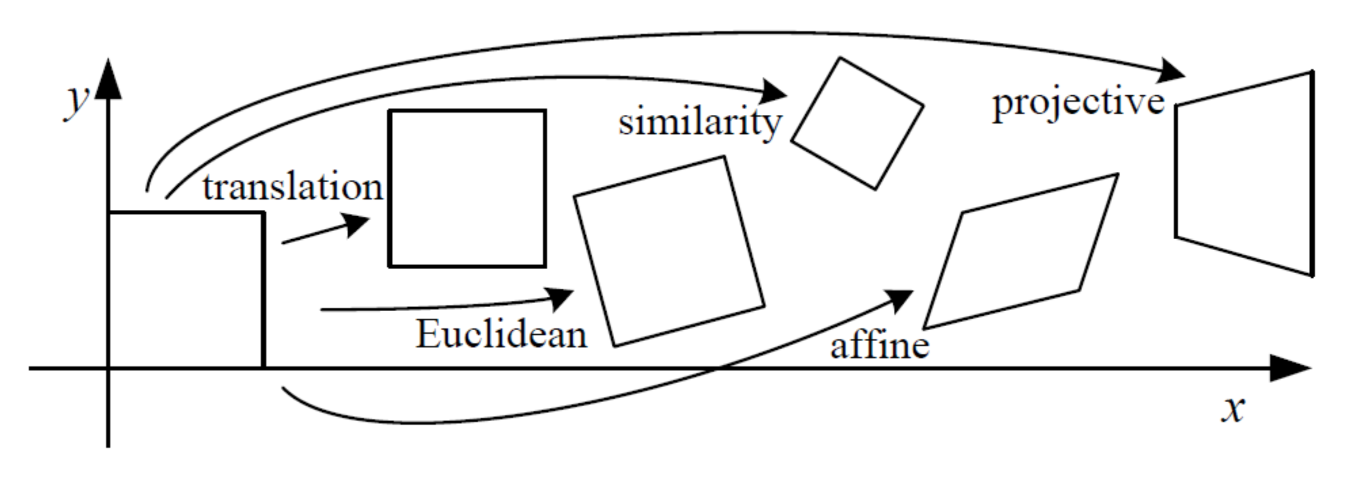

**Similarity transform (4 DoF)**:
Scaled rotation + translation
x' = sRx + t

OR $\mathbf{x}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix} \bar{\mathbf{x}}$

$$\bar{\mathbf{x}}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \bar{\mathbf{x}}$$

where R is the (2 X 2) orthonormal rotation matrix and s is an arbitrary scale factor.

# 2D transformations




Affine transformation (6 DoF):

$\mathbf{x}' = \mathbf{A}\bar{\mathbf{x}}.$ where A is an arbitrary 2 X 3 matrix.

$$\mathbf{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{\mathbf{x}}.$$

# Affine Transformation

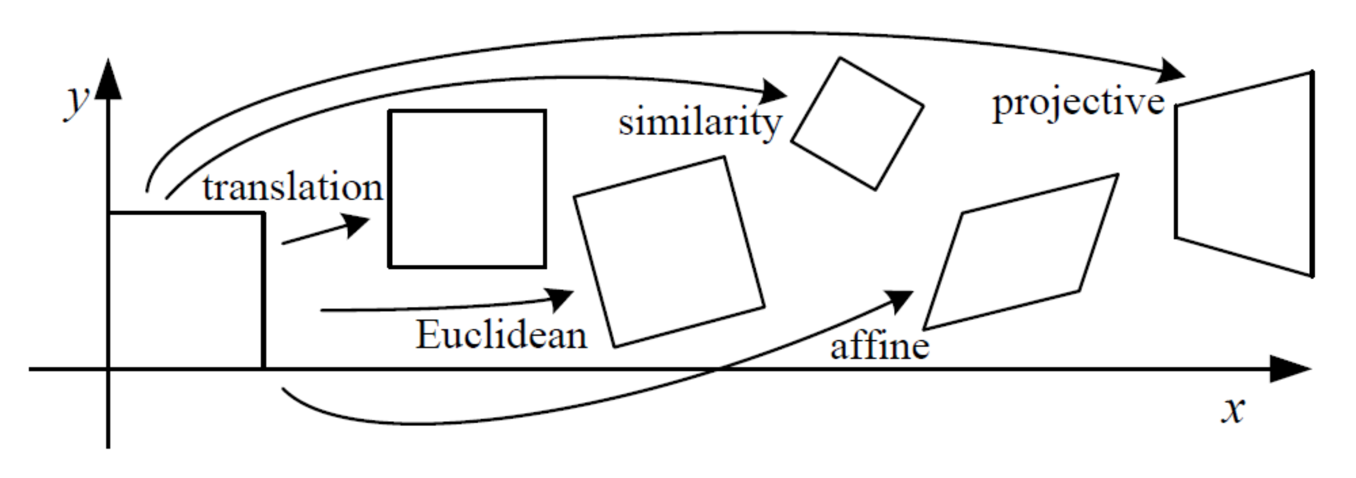
It preserves parallelism but may change distance and angles. It includes

- Translation
- Scaling
- Rotation
- Shearing

$$T = \begin{bmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{bmatrix}$$

where a,b,c,d define linear transformations (rotation, scaling, shear). tx,ty represent translation and the last row [0,0,1] ensures that they are in homogeneous coordinates.

# 2D transformations




**Projective transformation /Homography (8 DoF)**:
Operates on homogeneous coordinates

$\tilde{\mathbf{x}}' = \tilde{\mathbf{H}}\tilde{\mathbf{x}}$ Where $\tilde{\mathbf{H}}$ is an arbitrary 3 X 3 homogeneous matrix.

# Projective Transformation

It allows perspective distortion.

Ex: Converging parallel lines (like how a road appears to narrow in the distance).

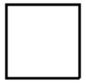
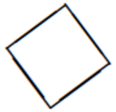
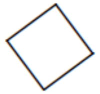
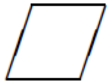
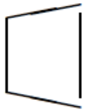
A Projective Transformation Matrix, also called a Homography Matrix, is a 3×3 matrix that represents a perspective transformation.

$$H = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ p & q & 1 \end{bmatrix}$$

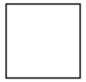
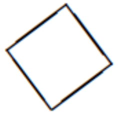
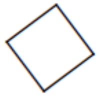
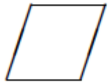
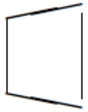
where

- a,b,c,d define the linear transformation
- tx,ty define translation and
- p,q represent perspective distortion (if p=q=0, the transformation is affine)

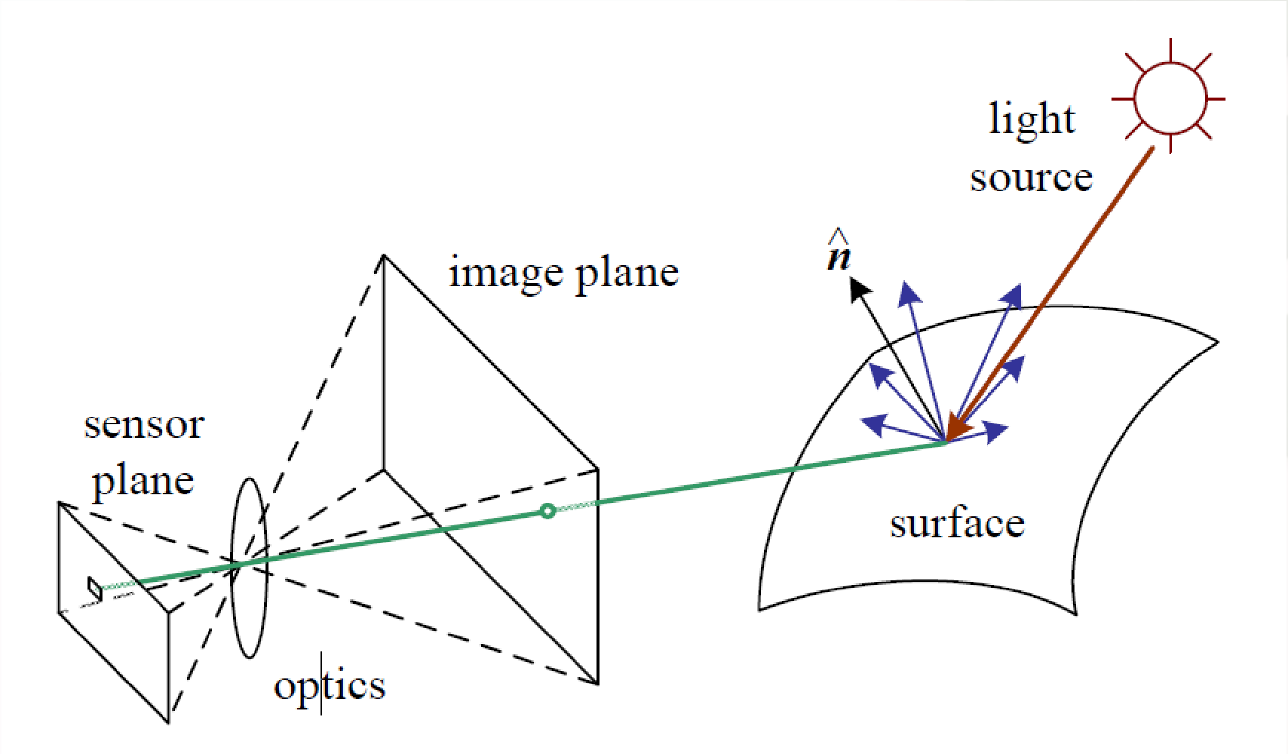
# Hierarchy of 2D transformations

| Transformation | Matrix | # DoF | Preserves | Icon |
|---|---|---|---|---|
| translation | $\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2\times 3}$ | 2 | orientation |  |
| rigid (Euclidean) | $\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2\times 3}$ | 3 | lengths |  |
| similarity | $\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2\times 3}$ | 4 | angles |  |
| affine | $\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2\times 3}$ | 6 | parallelism |  |
| projective | $\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3\times 3}$ | 8 | straight lines |  |

# Hierarchy of 3D transformations

| Transformation | Matrix | # DoF | Preserves | Icon |
|---|---|---|---|---|
| translation | $\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{3\times 4}$ | 3 | orientation |  |
| rigid (Euclidean) | $\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{3\times 4}$ | 6 | lengths |  |
| similarity | $\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{3\times 4}$ | 7 | angles |  |
| affine | $\begin{bmatrix} \mathbf{A} \end{bmatrix}_{3\times 4}$ | 12 | parallelism |  |
| projective | $\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{4\times 4}$ | 15 | straight lines |  |

# Photometric image formation

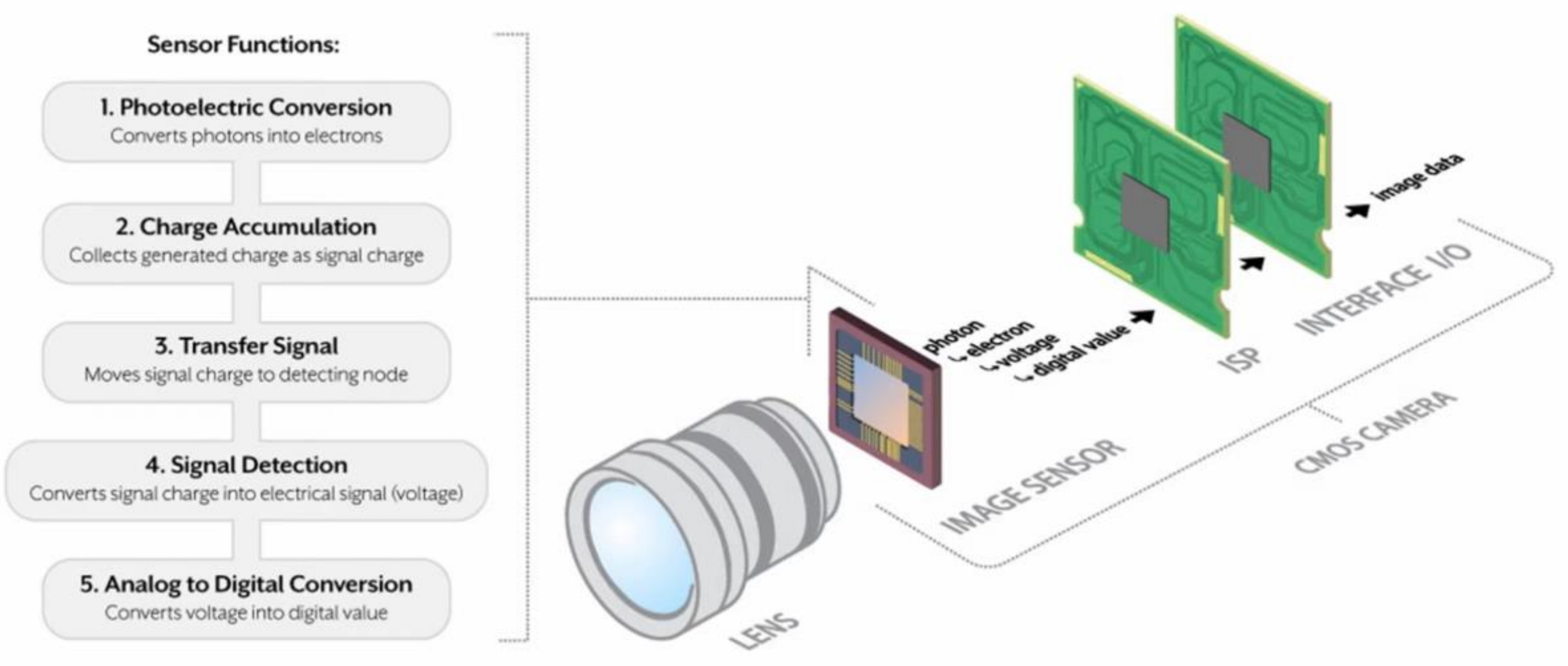


# Photometric image formation

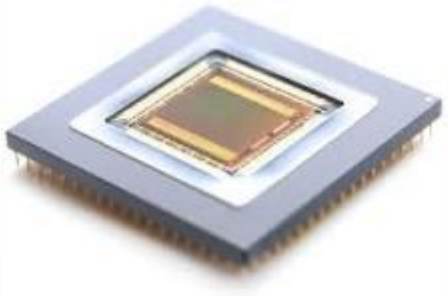
**Some factors that affect image formation are:**

- The strength and direction of the light emitted from the source.
- The material and surface geometry along with other nearby surfaces.
- Sensor capture properties

# Digital Camera



# Digital Camera



A digital camera has a sensor that converts light into electrical charges.

Once the sensor converts the light into electrons, it reads the value (accumulated charge) of each cell in the image.

A CCD transports the charge across the chip and reads it at one corner of the array.

An analog-to-digital converter (ADC) then turns each pixel's value into a digital value by measuring the amount of charge at each photosite and converting that measurement to binary form.

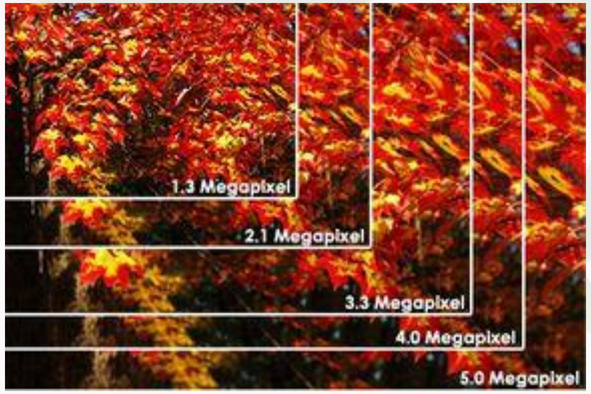
# Image Resolution

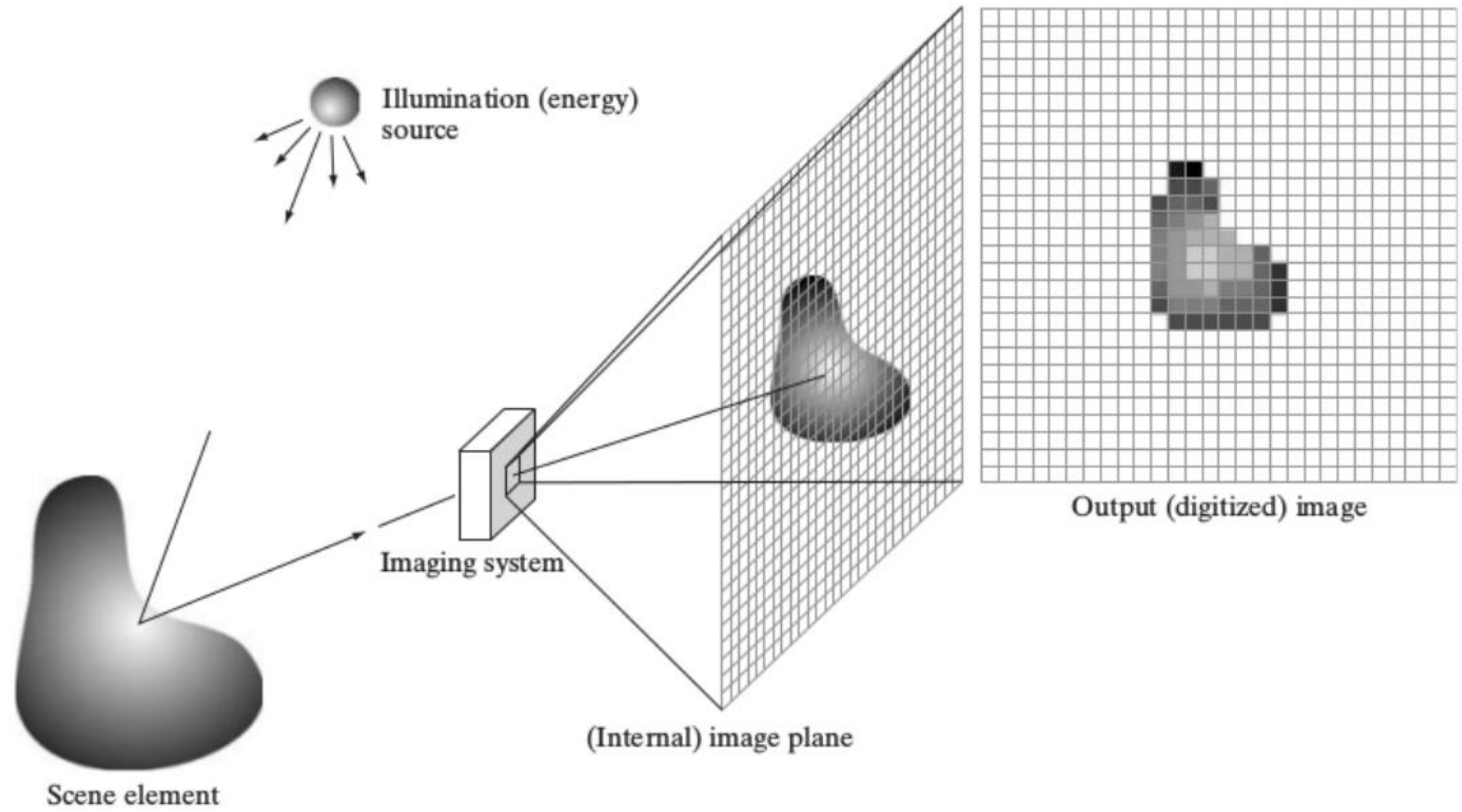
The amount of detail that the camera can capture is called the resolution, and it is measured in pixels.

The more pixels a camera has, the more detail it can capture and the larger pictures can be without becoming blurry or "grainy.“

High-end consumer cameras can capture over 12 million pixels.

Some professional cameras support over 16 million pixels, or 20 million pixels for large-format cameras.




Illumination (energy)
source
Output (digitized) image
Imaging system
(Internal) image plane
Scene element

# Sampling and Quantization

In Digital Image Processing, signals captured from the physical world need to be translated into digital form by “Digitization” Process.

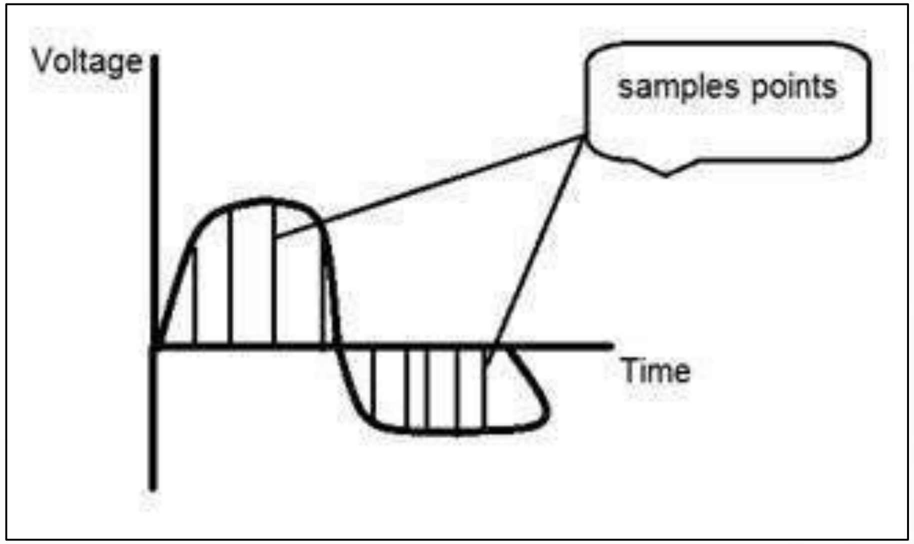
In order to become suitable for digital processing, an image function f(x,y) must be digitized both spatially and in amplitude.

This digitization process involves two main processes called

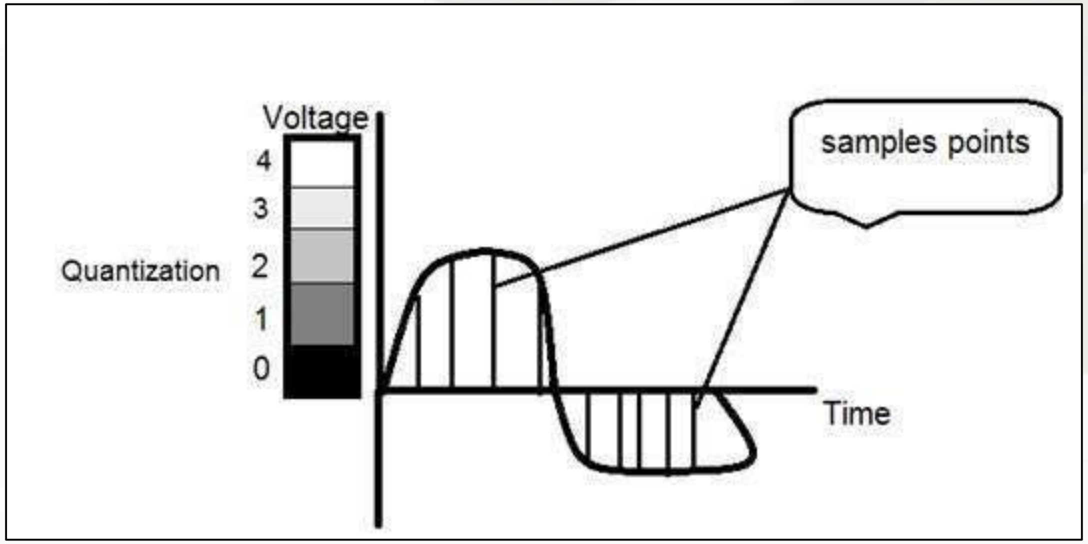
**Sampling:** Digitizing the co-ordinate value is called sampling.

**Quantization:** Digitizing the amplitude value is called quantization

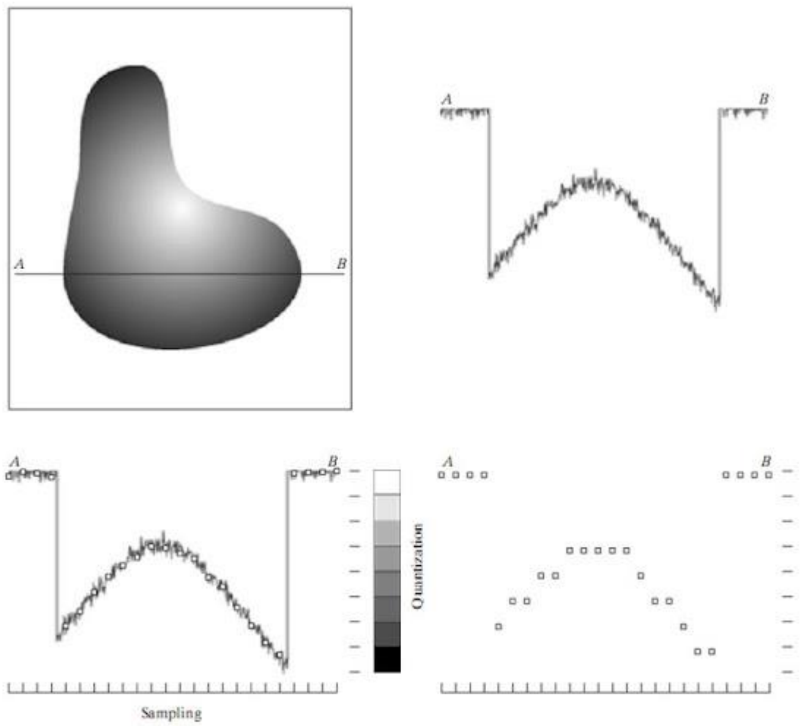
# Sampling and Quantization

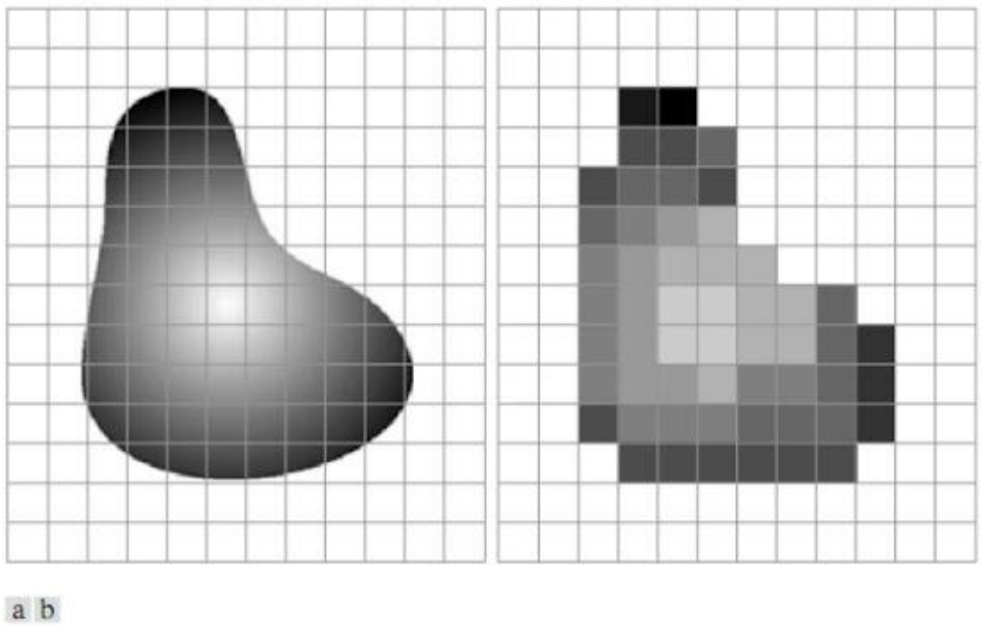



Sampling




Quantization


A
B
A
B
A
B
A
B
Quantization
Sampling


a b

# Thank you