COMPUTER VISION

UNIT - I

Introduction to Computer Vision

In Computer Vision a camera is linked to a computer. The Computer interprets images of a real scene to obtain information useful for tasks such as navigation, manipulation and recognition.

In computer vision, we are trying to do the inverse, i.e. to describe the world that we see in one or more images and to reconstruct its properties, such as shape, illumination and color distributions.

Computer vision is being used today in a wide variety of real-world applications, which include

* Optical character recognition
* Machine inspection.
* Retail
* 3D model building
* Medical imaging

* Automotive Safety
* Match move
* Motion Capture
* Surveillance
* Fingerprint recognition & biometrics

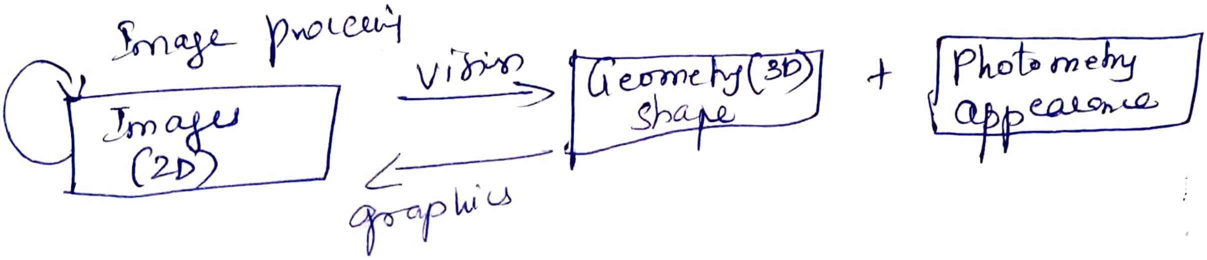
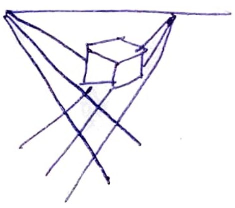
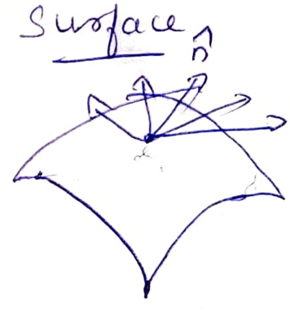


Image Formation

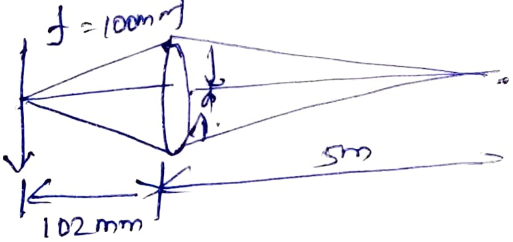
Perspective projection



light scattering when hitting a surface



Lens optics



Bayer color filter array

G	R	G	R
B	G	B	G
G	R	G	R
B	G	B	G

Geometric Primitives

Geometric primitives form the basic building blocks used to describe three-dimensional shapes.

2D points

2D points can be denoted using a pair of values, $x = (x, y) \in R^2$

$$x = \begin{bmatrix} x \\ y \end{bmatrix}$$

A homogeneous vector $\tilde{x}$ can be converted back into an inhomogeneous vector x by dividing through by the last element $\tilde{w}$, i.e.,

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

where $\bar{x} = (x, y, 1)$ is the augmented vector.

2D lines

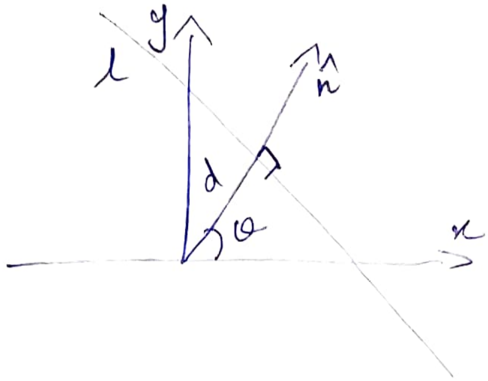
2D lines can also be represented using homogeneous coordinate $\tilde{l} = (a, b, c)$. The corresponding line equation is

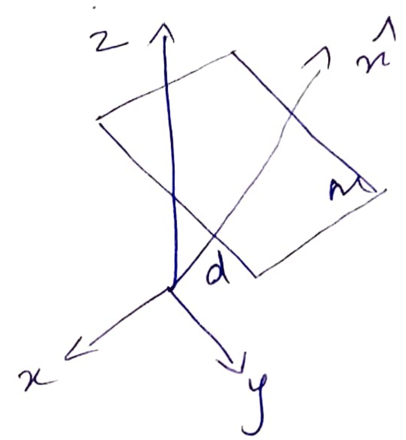
$$\bar{x} \cdot \tilde{l} = ax + by + c = 0$$

We can normalize the line equation vector so that $l = (\hat{n}_x, \hat{n}_y, d) = (\hat{n}, d)$ with $\|\hat{n}\| = 1$. In this case, $\hat{n}$ is the normal vector perpendicular to the line and d is distance to the origin.

2D line equation

3D plane equation





The combination (θ, d) is also known as polar coordinates.

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

$$\bar{x} = \bar{l}_1 \times \bar{l}_2$$

where $\times$ is the cross product operator. Similarly, the line joining two points can be written as

$$\bar{l} = \bar{x}_1 \times \bar{x}_2$$

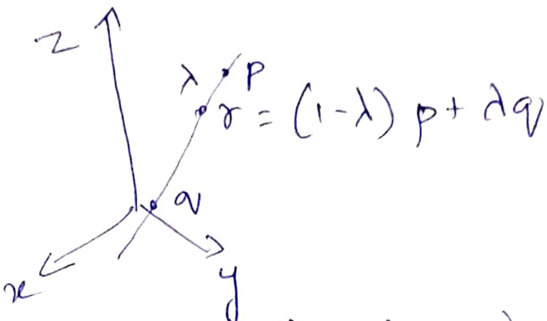
2D Conics

There are other algebraic curves that can be expressed with simple polynomial homogeneous equations. For example, the conic sections can be written using a quadric equation

$$\bar{x}^T Q \bar{x} = 0$$

3D Points

Point Coordinates in three dimensions can be written using inhomogeneous coordinates $x = (x, y, z) \in \mathbb{R}^3$



3D ~~plan~~ line equation, $r = (1-\lambda)p + \lambda q$

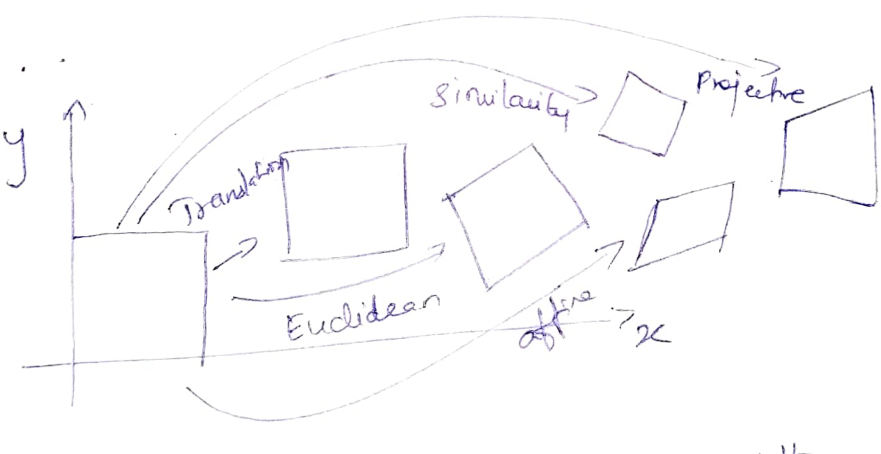
3D Planes

3D planes can also be represented as homogeneous coordinates $\bar{m} = (a, b, c, d)$ with a corresponding plane equation

$$\bar{x} \cdot \bar{m} = ax + by + cz + d = 0$$

2D Transformations

The simplest transformation occur in the 2D Plane. .



Translation : 2D translations can be written as

$$x' = x + t$$

Where I is the (2×2) Identity matrix

Rotation + Translation

This transformation is also known as 2D rigid body motion or the 2D Euclidean transformation. It can be written as

$$x' = Rx + t$$

where

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

is an orthonormal rotation matrix with $RR^T = I$ and $|R| = 1$.

Scaled Rotation.

Also known as the similarity transform, this transformation can be expressed as $x' = sRx + t$ where s is an arbitrary scale factor. It can also be written as

$$x' = \begin{bmatrix} sR & t \end{bmatrix} \bar{x} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \bar{x},$$

where we no longer require that $a^2 + b^2 = 1$.

Affine

The affine transformation is written as $x' = A\bar{x}$, where A is an arbitrary 2×3 matrix, ie.

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

Projective

This transformation, also known as perspective transform or homography, operates on homogeneous coordinates,

$$\bar{x}' = \tilde{H}\bar{x},$$

where $\tilde{H}$ is an arbitrary 3x3 matrix. The resulting homogeneous coordinate $\bar{x}'$ must be normalized in order to obtain an inhomogeneous result x, ie.

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \quad \text{and} \quad y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$

Hierarchy of 2D transformations

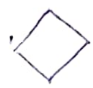
The above transformation form a nested set of groups, ie, they are closed under composition and have an inverse that is a member of the same group. Each group is a subset of the more complex group below it.

Co-vectors

While the above transformation can be used to transform points in 2D plane, can they also be used directly to transform a line equation.

$$\bar{l}' \cdot \bar{x}' = \bar{l}^T \tilde{H} \bar{x} = (\tilde{H}^T l^{-1})^T \bar{x} = \bar{l} \cdot \bar{x} = 0$$

ie $\bar{l}' = \tilde{H}^{-T} \bar{l}$.

Transformation	Matrix	# DOF	Preserves	Icon
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{2\times 3}$	3	lengths	
Similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times 3}$	6	parallelism	
Projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times 3}$	8	Straight line	

3D Transformations

The set of three-dimensional coordinate transformations is very similar to that available for 2D transformations. As in 2D, these transformation form a nested set of groups.

Translation. 3D translations can be written as

$$x' = x + t \quad \text{or}$$

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

where I is the (3×3) identity matrix and 0 is the zero vector

Rotation + translation

Also known as 3D rigid body motion or the 3D Euclidean transformation, it can be written as $x' = Rx + t$ ~~or~~

where R is a 3×3 orthonormal rotation matrix with $RR^T = I$ and $|R| = 1$. Note that sometimes it is more convenient to describe a rigid motion way

$$x' = R(x - c) = Rx - Rc,$$

where c is the center of rotation

Scaled rotation

The 3D similarity transform can be expressed as $x' = sRx + t$ where s is an arbitrary scale factor. It can also be written as

$$x' = \begin{bmatrix} sR & t \end{bmatrix} \bar{x}$$

Affine

The affine transform is written as $x' = A\bar{x}$, where A is an arbitrary 3×4 matrix, is

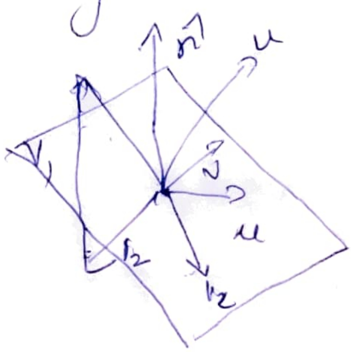
$$x' = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{bmatrix} \bar{x}$$

Projective

This transformation, variously known as 3D perspective transform, homography, or collineation, operates on homogeneous coordinates,

$$\tilde{x}' = \tilde{H}\tilde{x},$$

where $\tilde{H}$ is an arbitrary 4x4 homogeneous matrix. As in 2D, the resulting homogeneous coordinate $\tilde{x}'$ must be normalized in order to obtain an inhomogeneous result x. Perspective transformations preserve straight lines.



3D rotation

The biggest difference between 2D and 3D coordinate transformations is that the parameterization of 3D rotation matrix R is not as straight forward but several possibilities exist.

Euler angles

A rotation matrix can be formed as the product of three rotations around three cardinal axes, e.g., x, y and z, or x, y, and x.

Axis/angle

A rotation can be represented by a rotation axis $\hat{n}$ and an angle θ, or equivalently by a 3D vector $\omega = \theta\hat{n}$.

$$v_{\parallel} = \hat{n}(\hat{n} \cdot v) = (\hat{n}\hat{n}^T) \cdot v$$

Next, we compute the perpendicular residual of v from $\hat{n}$,

$$v_{\perp} = v - v_{\parallel} = (I - \hat{n}\hat{n}^T)v$$

3D to 2D projections

The simplest model is orthography, which requires no division to get the final result. The more commonly used model is perspective, since this more accurately models the behaviour of real cameras.

Orthography and para-perspective

~~An~~ An orthographic projection simply drops the z component of the three-dimensional coordinate p to obtain the 2D point x. This can be written as

$$x = \begin{bmatrix} I_{2\times2} \mid 0 \end{bmatrix} \cdot p$$

If we are using homogeneous coordinate, we can write

$$\tilde{x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{p},$$

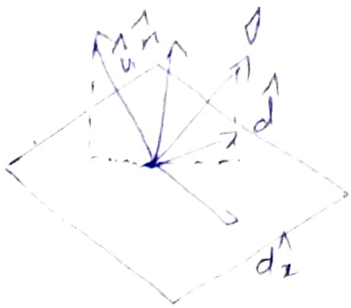
Photometric image formation

Lighting

Images cannot exist without light. To produce the image, the scene must be illuminated with one or more light sources. Light sources can generally be divided into point and area light sources.

A point light source originates at a single location in space, potentially at infinity. In addition to its location, a point light source has an intensity and a color spectrum. The intensity of a light source falls off with the square of the distance between the source and the object being lit, because the same light is being spread over a larger (spherical) area.

Light scatter when it hits a surface



The bidirectional reflectance distribution function (BRDF) $f(\theta_i, \phi_i, \theta_r, \phi_r)$.

Reflectance and Shading

When light hits an object's surface, it is scattered and reflected.

The Bidirectional Reflectance Distribution function (BRDF)

The most general model of light scattering is the BRDF. Relative to some local coordinate frame on the surface, the BRDF is a four dimensional function that describes how much of each wavelength arriving at an incident direction $\hat{v}_i$ is emitted in a reflected direction v_r. The function can be written in terms of angles of the incident.

$$f_r(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)$$

The BRDF is reciprocal.

Most surfaces are isotropic, i.e., there are no preferred directions on the surface as far as light transport is concerned.

Diffuse reflection

The diffuse component scatters light uniformly in all directions and is the phenomenon we most normally associate with shading. Diffuse reflection also often impart a strong body color to the light since it is caused by selective absorption and re-emission of light inside the object's material.

The shading equation for diffuse reflection can be written as

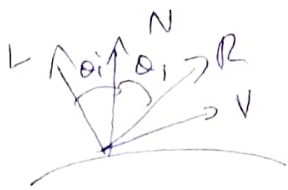
$$L_d(\hat{v}_r; \lambda) = \sum_i L_i(\lambda) f_d(\lambda) \cos^+ \theta_i =$$

$$\sum_i L_i(\lambda) f_d(\lambda) [\hat{v}_i \cdot \hat{n}]^+$$

where $[\hat{v}_i \cdot \hat{n}]^+ = \max(0, \hat{v}_i \cdot \hat{n})$.

Specular Reflection

For a perfect mirror, light is reflected about N



$$I_e = \begin{cases} I_i & \text{if } V = R \\ 0 & \text{otherwise} \end{cases}$$

The second major component of BRDF is specular reflection, which depends strongly on the direction of the outgoing light. Consider light reflecting off a mirrored surface. Incident light rays are reflected in a direction that is rotated by 180° around the surface normal $\hat{n}$.

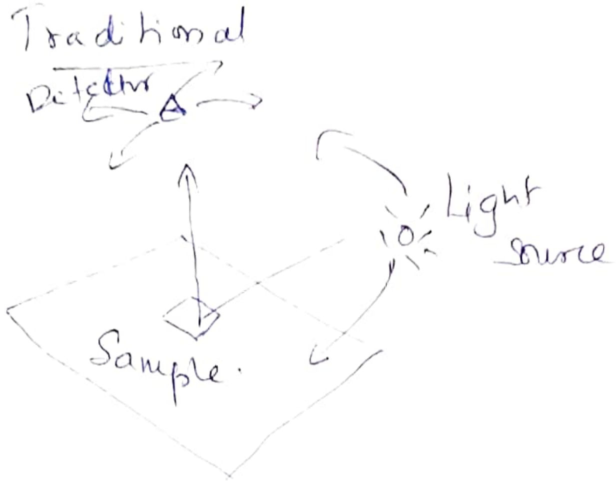
BRDF models

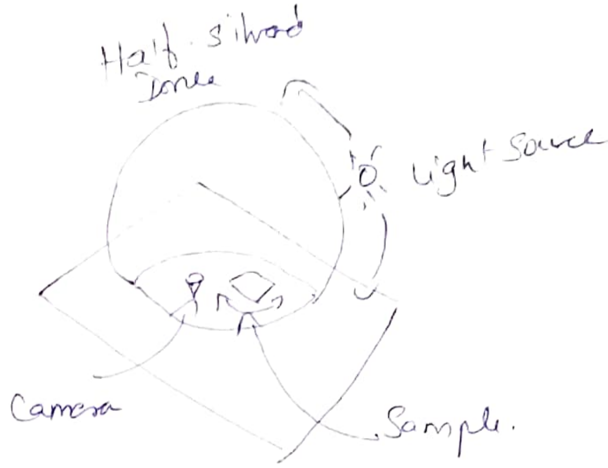
Phenomenological

- Phong
- Ward
- Lafortune
- Ashikhmin

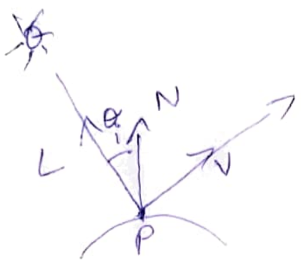
Physical

- Cook-Torrance
- Dichromatic





Diffuse Reflection



$R_e = k_d N \cdot L R_i$

$I = k_d N \cdot L$

Image intensity of P

Shape from shading

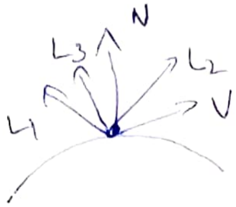
Suppose $k_d = 1$

$I = k_d N \cdot L$

$= N \cdot L$

$= \cos \theta_i$

· Photometric Stereo



$$I_1 = K_d\, N \cdot L_1$$
$$I_2 = K_d\, N \cdot L_2$$
$$I_3 = K_d\, N \cdot L_3$$

Can Write this as a matrix equation:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = K_d \begin{bmatrix} L_1^T \\ L_2^T \\ L_3^T \end{bmatrix} N$$

Solving the equations

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}}_{\substack{I \\ 3\times 1}} = \underbrace{\begin{bmatrix} L_1^T \\ L_2^T \\ L_3^T \end{bmatrix}}_{\substack{L \\ 3\times 3}} \underbrace{K_d N}_{\substack{G \\ 3\times 1}}$$

$$G = L^{-1} I$$

$$K_d = \| G \|$$

$$N = \frac{1}{K_d} G$$

Sampling and aliasing

During sampling process, a continuous-time signal is converted to discrete-time signals by taking samples to continuous-time signal at discrete time intervals.

$$x(nTs) = x(t)$$

T - Sampling Interval

$x(t)$ - Analog input signal

Sampling Theorem

* Sampling theorem gives the criteria for minimum number of samples that should be taken.

* Sampling criteria :- "Sampling frequency must be twice of highest frequency"

$$f_s = 2W$$

f_s = Sampling frequency

W = higher frequency content

Proof of sampling theorem

* There are two parts

-> representation of $x(t)$ in its samples

-> reconstruction of $x(t)$

ALIASING

→ While providing sampling theorem we considered

$f_s = 2W$

→ Consider the case that $f_s < 2W$

Effect of Aliasing

1. Distortion
2. The data is lost and it cannot be recovered.

To avoid Aliasing

1. Sampling rate must be $f_s >= 2W$
2. Strictly ~~bandwidth~~ limit the signal to 'W'

Point Operations.

Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis.

It has two broad categories

- Spatial domain methods
- Frequency domain methods.

→ Spatial domain methods are operate directly on the pixels.

-> Point processing operation deals with pixel intensity values individually.

-> Enhanced at any point in an image depends only on the gray level at that point techniques are referred as point processing.

-> Most spatial domain enhancement operations can be reduced to the form of.

$$g(x,y) = T[f(x,y)]$$

T is referred to as a gray level transformation function or a point processing operations.

$$S = T(r)$$

S -> processed image pixel value

r -> original image pixel value

-> Mask is a small matrix useful for blurring, sharpening, edge detection

-> Contrast stretching expands the range of intensity levels in an image.

-> Extreme contrast stretching yields Thresholding

Pixel Transforms

A general image processing operator is a function that takes one or more input images and produces an output image.

$$g(x) = h(f(x)) \text{ or } g(x) = h(f_0(x), \ldots f_n(x)).$$

For discrete (sampled) images, the domain consists of finite number of pixel locations, $x = (i, j)$, and we can write $g(i, j) = h(f(i, j))$.

Color Transformation

Use to transform colors to colors.

$$g(x, y) = T[f(x, y)]$$

$f(x, y)$ = input color image.

$g(x, y)$ = output color image.

T = operation on f over a spatial neighborhood

when only data at one pixel is used in the transformation, we can express the transformation as:

$$s_i = T_i(r_1, r_2, \ldots, r_n) \quad i = 1, 2, \ldots n$$

where r_i = color component of $f(x, y)$ RGB, $n = 3$

s_i = Color component of $g(x, y)$

Formula for RGB:

$$S_R(x,y) = k r_R(x,y)$$

$$S_G(x,y) = k r_G(x,y)$$

$$S_B(x,y) = k r_B(x,y)$$

Formula for HSI

$$S_I(x,y) = k r_I(x,y)$$

Formula for CMY:

$$S_C(x,y) = k r_C(x,y) + (1-k)$$

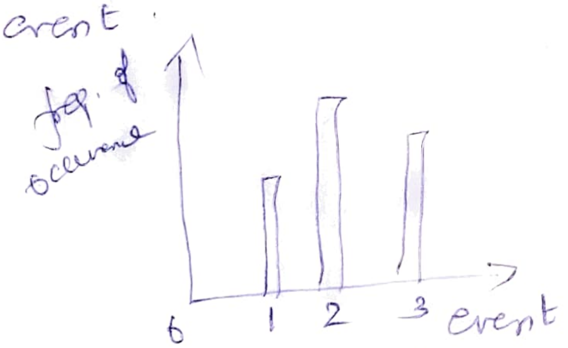
$$S_M(x,y) = k r_M(x,y) + (1-k)$$

$$S_Y(x,y) = k r_Y(x,y) + (1-k)$$

Histogram Processing

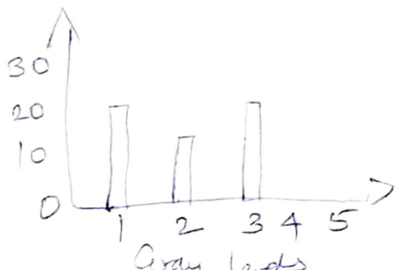
Histogram:

It is a plot of frequency of occurrence of an event.



→ X axis has gray levels & Y-axis has No. of pixels in each gray levels.

Gray level	No of pixels
0	40
1	20
2	10
3	
4	15



Filtering

-> Filtering is a technique used for modifying or enhancing an image like highlight certain features or remove other features.

-> Image filtering include smoothing, sharpening and edge enhancement

-> Term "Convolution" means applying filter to an image.

-> It may be applied in either
- Spatial domain
- frequency domain

Linear Spatial filtering

Linear filter of an image f of size MxN with a filter mask size of mxn given by the expression:

$$g(i,j) = \sum_{k,l} f(i+k, j+l)\, h(k,l).$$

The entries in the weight kernel or mask $h(k,l)$ are often called the filter coefficients.

-> The process of linear filtering is same as convolution.

-> When interest lies on the response, R, of an mxn mask at any point (x,y) and not on the mechanism of implementing mask convolution.