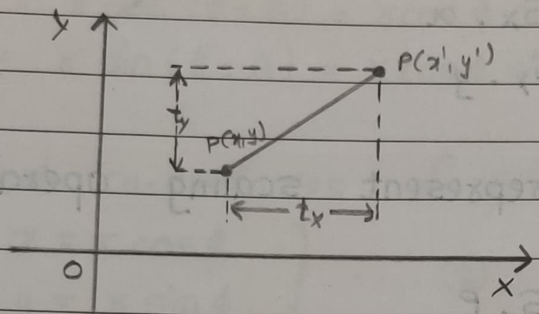
UNIT – III

2D, 3D Transformations & Projections

★ Translation –

→ • Translation is defined as shifting of an object along a straight path



- The vector $T = [t_x, t_y]$ specifying the amount of shift in both directions is called translation vector

- Translation operation is formulated as,

$$x' = x + t_x$$
$$y' = y + t_y$$

- Equivalent matrix representation of this is,

$$P' = T + P$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

★ Scaling -

- This operation can be carried out for polygons by multiplying co-ordinate axes (x, y) of each vertex by scaling factors S_x and S_y

$$\therefore x' = S_x \cdot x$$
$$y' = S_y \cdot y$$

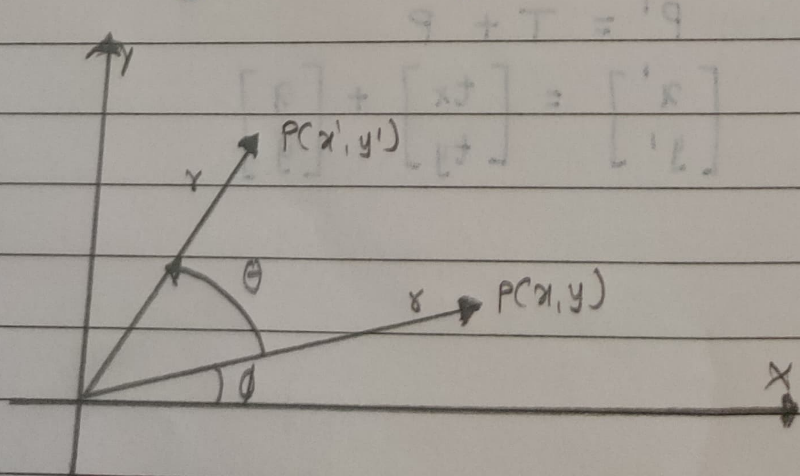
- We can represent scaling operation in matrix form as,

$$P' = S \cdot P$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Scaling may alter the shape of an object

★ Rotation -

- It is described by repositioning all points of object along circular path



- Consider rotation of object about origin
- Here, r is constant distance of point from origin, angle ϕ is original angular position of point from horizontal and θ is rotation angle.

- Using standard trigonometric equations,

$$\left.\begin{aligned} x' &= r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta \\ y' &= r\sin(\phi+\theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \end{aligned}\right\} \cdots ①$$

- Original co-ordinates of point in polar form are,

$$\left.\begin{aligned} x &= r\cos\phi \\ y &= r\sin\phi \end{aligned}\right\} \cdots ②$$

- Substituting eqn ② into ①, we get

$$\left.\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}\right\} \cdots ③$$

- Above eqn can be represented in matrix form as,

$$[x' \;\; y'] = [x \;\; y]\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Q.1 Perform scaling on a triangle (1, 1), (8, 1) and (1, 9) with scaling factor of 2 in both x and y - directions. Find final co-ordinates

→ Given,
triangle A(1, 1), B(8, 1), C(1, 9) with parameters
$S_x = S_y = 2$

Transformed co-ordinates are given as,

$$P' = S . P$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 8 & 1 \\ 1 & 1 & 9 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 16 & 2 \\ 2 & 2 & 18 \\ 1 & 1 & 1 \end{bmatrix}$$

Q.] Find transformation of triangle A(1,0), B(0,1), C(1,1) by translating one unit in x & y directions and then rotating 45° about origin

→ $P' = R_{(\theta=45^\circ)} \cdot T \cdot P$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.71 & -0.71 & 0 \\ 2.13 & 2.13 & 2.84 \\ 1 & 1 & 1 \end{bmatrix}$$

* Homogeneous Co-ordinates -

- To combine sequence of transformations, we have to eliminate matrix addition associated with translation terms in M_2
- To achieve this, we have to represent matrix m as 3×3 instead of 2×2 introducing an additional dummy co-ordinate w.
- This co-ordinate system is called Homogeneous co-ordinate system.
- Homogeneous co-ordinate is represented by a triplet (x_w, y_w, w), where

$$x = \frac{x_w}{w}, \quad y = \frac{y_w}{w}$$

- For 2-D transformations, w can be any non-zero value, but we usually have $w = 1$

- Homogeneous co-ordinate for translation:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

- Homogeneous co-ordinate for rotation:

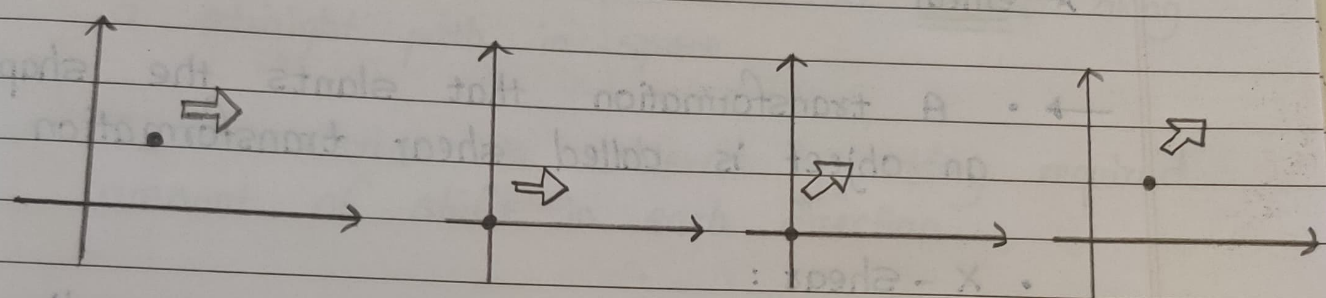
$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- For scaling, $S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

★ Rotation about an Arbitary Point –

→ • General pivot point rotation can be performed in following steps :

1) Translate pivot point to origin
2) Rotate the object
3) Translate back the pivot point to its original place.



Object to be rotated | Translate ref. point to origin | Rotate object about origin | Translate back refer. point.

• If m is final transformation matrix then,

$$m = T^{-1}.R.T = \begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & -x_r\cos\theta + y_r\sin\theta \\ \sin\theta & \cos\theta & -x_r\sin\theta - y_r\cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_r - x_r\cos\theta + y_r\sin\theta \\ \sin\theta & \cos\theta & y_r - x_r\sin\theta - y_r\cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r\sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

★ **Shear** –

→ • A transformation that slants the shape of an object is called shear transformation

• X - shear :

- The x shear preserves y - co-ordinates, but changes x values which causes vertical lines to tilt right / left
- Transformation matrix for x shear is :

$$X\_sh = \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$x' = x + Sh_x \cdot y$ & $y' = y$

• Y - shear:

- The y shear preserves x - co-ordinates, but changes y values causing horizontal lines to move up / down
- Transformation matrix for y shear is :

$$Y\_sh = \begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Date
Page

$\therefore x' = x$ and $y' = y + Sh_y \cdot x$

★ 3D Translation -

→ • The 3D translation is a process of moving an object from one location to another along a straight path in space

• The translation is achieved by adding required amount of shift in each direction.

• Matrix representation is :

$$P' = T.P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

★ 3D Scaling -

→ • Matrix representation in homogeneous co-ordinate system -

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

★ 3D rotation about Principal Axis –

- About X – Axis :
  - Matrix representation :

$$P' = R_x(\theta) . P$$

where
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- About Y – Axis :
  - Matrix representation :

$$P' = R_y(\theta) . P$$

where
$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Page

* Projections -

→ • Projection is the process of transforming an object representation from n-dimensional space to less than n-dimensional space.

• Parallel projection is achieved by passing parallel rays from object vertices and projecting the object on view plane.

• Perspective projection produces realistic views but does not preserve relative proportions.

* Oblique Parallel projections -

→ • If projectors are parallel to each other but are not perpendicular to view plane, it is called oblique parallel projection.

• α is the angle between oblique projector and plane

• If $\alpha = 45°$ (angle between projector and view plane) projection is called Cavalier projection

• If $\alpha = 63.4°$, the type of projection is known as Cabinet Projection.

★ Two-point Perspective Projection :-

→ • Two-point perspective projection occurs when view plane is parallel to one of principal axes or if view plane intersects exactly two principal axes

• X-direction perspective projection is given by,

$$P' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & q & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= [x \ y \ z \ (px + qy + 1)]$$

$$x' = \frac{x}{px + qy + 1}, \quad y' = \frac{y}{px + qy + 1}, \quad z' = \frac{z}{px + qy + 1}$$

• In similar way, we can derive value of (x', y', z') for other 2 cases

UNIT - IV

Light, Colour, Shading and Hidden Surfaces

★ Properties of Light -

- When light is incident on a object, some frequency is absorbed and some is reflected.

- Dominant frequency decides the color of object. Hence, it is also called Hue or colour

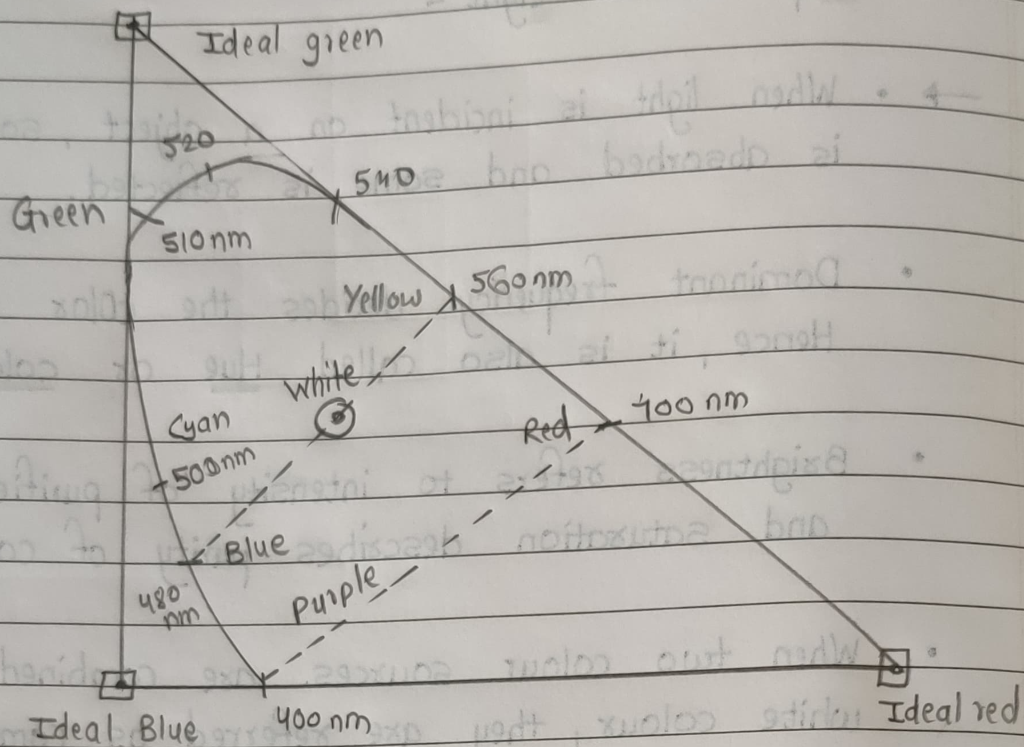
- Brightness refers to intensity of purified light and saturation describes purity of colour

- When two colour sources are combined to produce white colour, they are referred as complementary colours

- Usually, colour model use combination of three colours to produce wide range of colours, called colour gamut for that model

- Basic colours used to produce color gamut in particular model are called primary colours.

* CIE Chromaticity Diagram –



- CIE XYZ color model was first mathematically defined color model
- Any color C can be defined in this model as :

  $C = xX + yY + zZ$,

  where x, y, z represent vector in 3D color space and X, Y, Z represent amount of standard primary colors of model

- With above expression, we can define chromaticity values by normalizing against luminance (X+Y+Z). They can be given as,

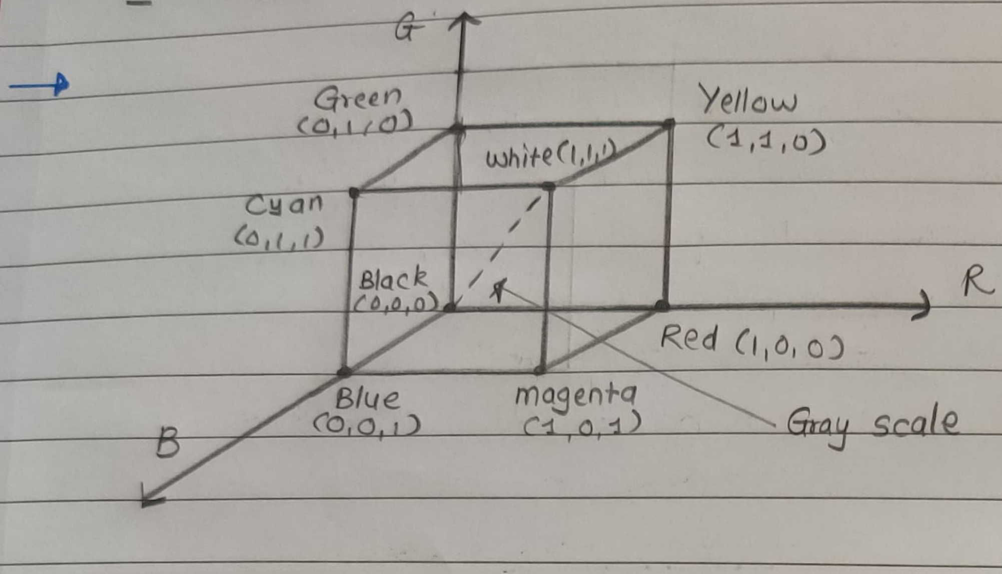
$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}$$

where $x + y + z = 1$

- The interior and boundary of tongue-shaped region represent all visible chromaticity values
- The points on the boundary are pure colours and a standard white light defined by light source C, is marked by centre dot

★ RGB Color model

→



- In this model, individual red, green and blue are added together to get resultant colour.

- Vertex of cube on axes represent primary colours and remaining vertices represent the complementary colour for each primary colour

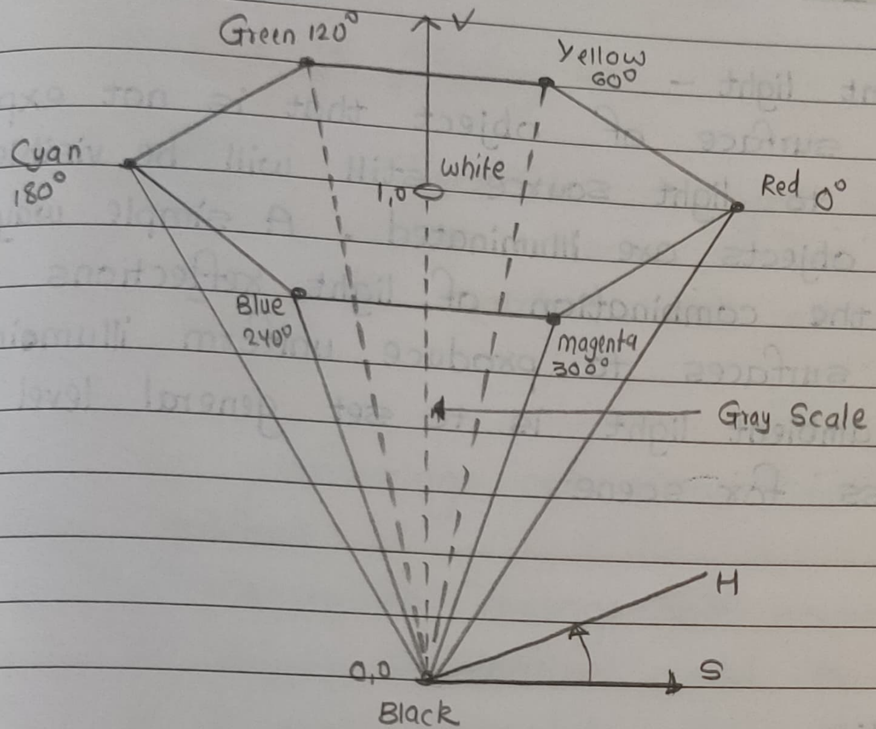
- Main diagonal of cube, represent the gray levels. End of origin of diagonal represent black and other end represent white

- Intensities of primary colours are added to get resultant colour.

  Thus, resultant colour C_λ is expressed as

  $C_\lambda = RR + GG + BB$

★ HSV Colour Model -



- This model uses 3 colour parameters : Hue (H), Saturation (S), Value (V)

- Hue (H) distinguishes among colours such as red, green, purple & yellow
- Saturation (S) refers to how far colour is from a gray of equal intensity
- Value (V) indicates level of brightness

- HSV is close to how human perceives the colour.

- **Definition –**

Ambient light –

A surface of object that is not exposed directly to light source still will be visible, if nearby objects are illuminated. A simple way of model the combination of light reflections from various surfaces to produce uniform illumoination called ambient light is to set general level of brightness for scene

- **Definition –**

Diffuse Illumination –

- An object illumination is as important as its surface properties in computing its intensity
- The object may be illuminated by light which does not come from any particular source but which comes from all directions.
- When such illumination is uniform from all directions, it is called diffuse illumination

- **Definition –**

Specular Reflection –

When we illuminate a shiny surface with bright light, we observe highlight or bright spot on shiny surface. This phenomenon of reflection of incident light is called specular reflection

★ Shading Algorithms -

★ Gouraud Shading -

→ • Polygon surface is displayed by linearly interpolating intensity values across surface. Here, intensity values are matched with values of adjacent polygons

- By performing calculations, we can display polygon surface :
  1) Determine average unit normal vector
  2) Apply illumination model to each polygon vertex
  3) Linearly interpolate vertex intensities over surface

- We can obtain normal vector at each polygon vertex by averaging surface normals of all polygons
  Eg. Three surface normals N_1, N_2, N_3 sharing vertex v

$$\therefore N_v = \frac{N_1 + N_2 + N_3}{|N_1 + N_2 + N_3|}$$

- Next step is to find vertex intensities. It can be determined by applying illumination model to each vertex

- Finally, each polygon is shaded by linear interpolating of vertex intensities along each edge.

★ Phong Shading –

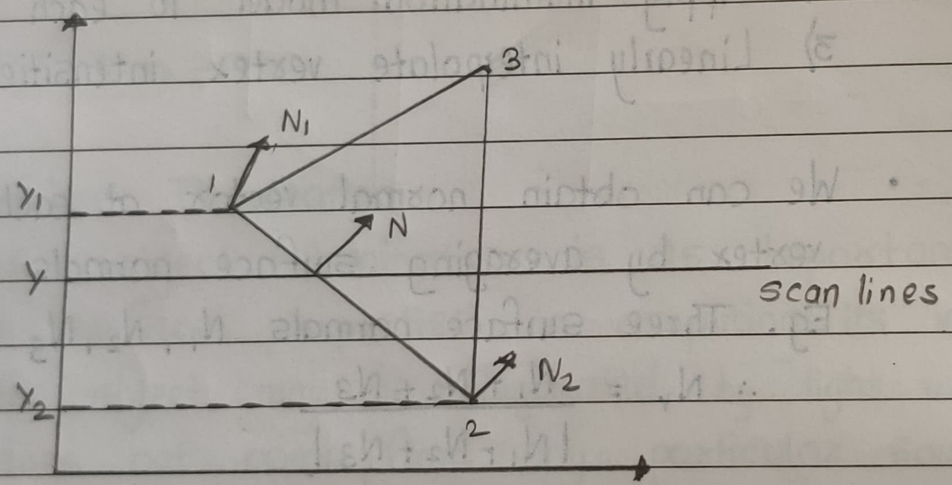
→ • Also known as normal vector interpolation shading

• This algorithm interpolates normal vector rather than intensity and apply illumination model at interpolated normal vectors.

• It works as follows:

1) Compute averaged unit normal vector
2) Linearly interpolate vertex normal vector
3) Apply illumination model at pixel along scan line

•



• Firstly, we find the average unit normal vector

• In second step, vertex normals are linearly interpolated over surface.
Normal at intersection of scan line y is:

$$N = \frac{y - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y}{y_1 - y_2} N_2$$

- Lastly, illumination model is applied using this normal.

★ Difference between Phong and Gouraud Shading –

Phong Shading	Gouraud Shading
1) Most sophisticated model	1) Less sophisticated model
2) Published in 1973	2) Published in 1971
3) Intensity is calculated at interpolating normal	3) Intensity is computed by interpolating colour
4) Requires more computation and produce smooth intensity	4) Requires less computation and produce realistic rendering
5) Light equation is used at each pixel	5) Light equation is used at each vertex
6) Produces smooth and shiny surfaces	6) Produces smooth surfaces.

★ Back Face Detection & Removal –

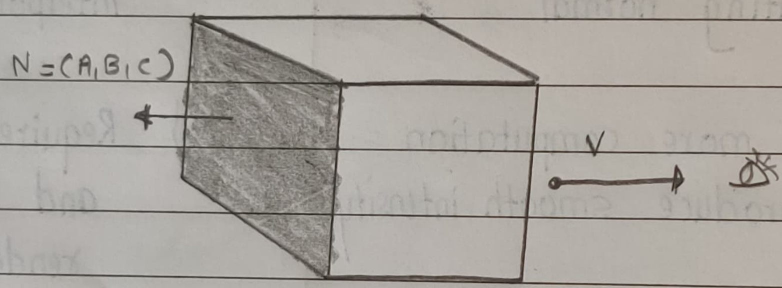
→ • If a polygon is visible, light surface should face towards us and dark surface should face away from us.

• Direction of light face can be identified by examining result,

$N.V > 0$

where, N = Normal vector

V = vector in viewing direction from eye



• If dot product is positive (vectors in same direction), we can say that polygon faces towards the viewer, otherwise it faces away and should be removed

• If viewing direction is along z-axis, then $V = (0, 0, V_z)$ & $N = (A, B, C)$

$\therefore N.V = (A, B, C) \cdot (0, 0, V_z) = V_z \cdot C$

• Hence, if z component is positive, then polygon faces towards us else it faces away

★ Depth Buffer (Z-buffer) Algorithm -

→ • Z-Buffer Algorithm -

1) Initialize z-buffer and frame buffer, so that for all buffer pos. z-buffer $(x,y) = 0$ & frame-buffer $(x,y) = I_{background}$

2) During scan conversion, for each position on polygon surface, compare depth values to previous values to determine visibility.
Calculate z-value for each (x,y) position,
If $z >$ z-buffer (x,y) then set
z-buffer $(x,y) = z$, frame buffer $(x,y) = I_{surface}(x,y)$

3) Stop.

- To calculate z-values, plane equation $Ax + By + Cz + D = 0$ is used.

$$\therefore \quad z = \frac{-Ax - By - D}{C}$$

★ Painter's Algorithm –

→ • 1) Sort all polygons in order of decreasing depth

2) Determine all polygons Q (preceding P) in polygon list whose z - extent overlap P

3) Perform test 2 for all Q :
   a) If every Q passes , scan convert polygon P
   b) If test fails for some Q , swap P & Q in list.
   If Q is already swapped , use plane containing polygon P to divide Q in 2 polygons Q_1 & Q_2
   Replace Q with Q_1 & Q_2

▲ Advantages –
Simple & Easy transparency

◂ Disadvantages –
Have to sort first , need to split polygons.

★ Warnock's Algorithm -

1) Initialize area to be whole screen

2) Create list of polygons by sorting them with z-values of vertices. Don't include disjoint polygons

3) Find relationship of each polygon

4) Perform visibility decision test:

a) If all polygons are disjoint from area, fill area with background colour

b) If there is only one contained polygon then first fill entire area with background colour, then fill part of contained polygon with colour of polygon

c) If there is single surrounding polygon, then fill area with colour of surrounding polygon

d) If surrounding polygon is closer to viewpoint than all other polygons, fill area with colour of surrounding polygon

e) If area is pixel (x, y), then compute z-coordinate of all polygons in list. Pixel is then set to colour of polygon which is closer to viewpoint

5) If none tests are true, then subdivide the area and go to step 2

UNIT-V

Curves & Fractals

★ Interpolation (Lagrange Method) -

→ • In curve generation, process of determining intermediate points between known sample points is achieved using interpolation

• The blending functions give control of the curve to each of the sample points for different values of 'u'.

• Assume that first sample point (x_1, y_1, z_1) has complete control when $u = -1$, second when $u = 0$, third $u = 1$ and so on
i.e When $u = -1 \rightarrow B_1(u) = 1$ and 0 for $u = 0, 1, 2 \ldots (n-2)$

• To get $B_1(u) = 1$ at $u = -1$ and 0 for $u = 0, 1, 2, \ldots n-2$, expression is,

$$B_1(u) = \frac{u(u-1)(u-2) \ldots [u-(n-2)]}{(-1)(-2) \ldots (1-n)}$$

where denominator is constant

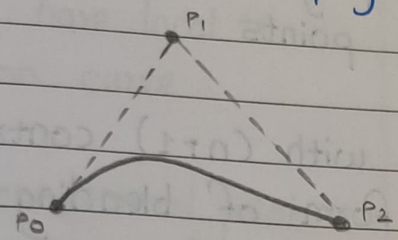
• Generalized formula for blending function is,

$$B_i(u) = \frac{(u+1)(u)(u-1) \ldots [u-(i-3)][u-(i-1)] \ldots [u-i-2]}{(i-1)(i-2)(i-3) \ldots (1)(-1) \ldots (i-n)}$$

• Approximation of curve using above expression is called Langrange interpolation

* Bezier Curve

→ • In general, a Bezier curve section can be fitted to any number of control points
• However, as number of control points increases, degree of Bezier polynomial also increases

• 

Bezier curve with 3 control points

• With n+1 control points, parametric equation of Bezier curve approximating points p_0 to p_n,

$$P(t) = \sum_{i=0}^{n} p_i \cdot BEZ_{i,n}(t), \quad 0 \leq t \leq 1$$

• Blending function of Bezier curves are directly derived from Bernstein polynomials

$$BEZ_{i,n}(t) = C(n,i)\,t^i(1-t)^{n-i}$$

• ~~For~~ * ~~direction~~, Properties of Bezier curve –
- Basis functions are real
- Degree of curve is one less than no. of control points
- Always interpolates first and last control points and approximates remaining two
- Bezier curve always satisfies convex hull property
- Bezier curve can fit any number of control points

* B-Spline Curve

- B-Spline curves are most widely used for approximating the shape of curve

- The degree of polynomial is independent of no. of control points

- B-Spline curve with (n+1) control points is expressed in form of blending function as,

$$P(t) = \sum_{i=0}^{n} P_K \cdot B_{i,d}(t), \quad t_{min} \le t \le t_{max}$$

and $2 \le d \le n+1$

- Properties –
  - B-Spline curves do not interpolate any control points
  - The curve has n+1 blending functions, if it is described with n+1 control points
  - B-Spline allows local control
  - Change in one control point influence shape of curve sections
  - B-Spline curve satisfy property of convex hull

Date
Page

★ Difference between Bezier and B-spline curve

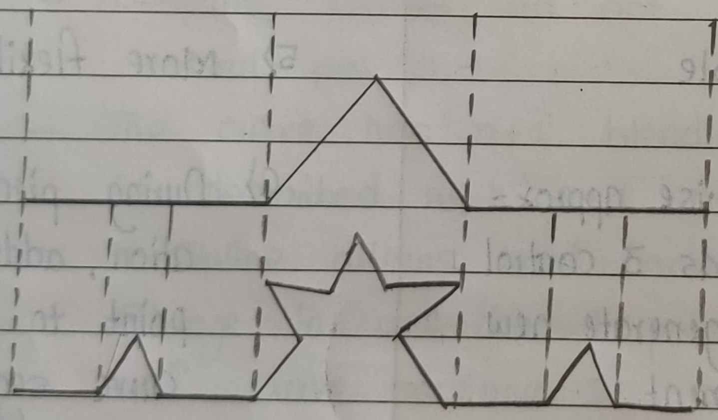
Bezier Curve	B-Spline Curve
1) Passes from first and last control point	1) Does not interpolate any control point
2) Does not have local control on curve	2) It has local control
3) Changing one control point affects position of all points	3) Changing one control point affects only few points
4) Degree is determined by no. of control points	4) Degree is independent of no. of control points
5) Less flexible	5) More flexible
6) During piecewise approximation, adds 3 control points to generate new curve segment	6) During piecewise approximation, adds only 1 control point to generate new curve segment.

- **Definition** -

Fractals -
Rough, jagged and random surfaces are called as fractals

★ Koch Curve -

→ • Koch Curve can be drawn by dividing line into 4 equal segment with scaling factor 1/3 and middle two segments are adjusted that they form adjacent sides of an equilateral triangle.



- Each repetition increases length of curve by factor 4/3
- Length of curve is infinite

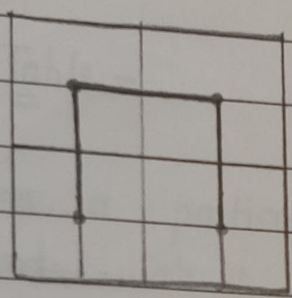
- Point sets, curves & surfaces which give fractal dimension greater than topological dimension are called Fractals

Date
Page

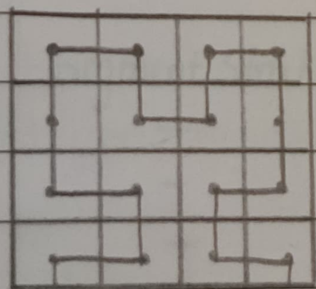
★ Hilbert Curve

- Generation of Hilbert curve is simple. It visits centre of every square grid having size $2^n \times 2^n$

- In first approximation, square is divided into 4 quadrants and curve joins the centre of all quadrants



- In second approximation, each quadrant is further divided into 2x2 grids, ending in 16 squares. Curve joins centre of all quadrants



- In third approximation, each of 16 squares is further subdivided into 2x2 grids, ending in 64 squares. Curve joins centre of all quadrants

- There is no limit on depth and hence there is no limit on curve length

Segment, Animation & Gaming

▪ Definition –

Segment –
Part of the display file is called a segment.
Typically, each segment corresponds to one concept in scene to be displayed

★ Segment Table –

- To access a particular segment and information associated with it, we must have a unique name assigned to each segment
- The structure used to organise all information related to segment is segment table

Segment Name	Segment Start	Segment size	Visibility	Scale x	Scale y
1	–	–	ON	–	–
2	–	–	OFF	–	–
3	–	–		–	–

* Algorithm to create segment –

Step 1 : If any segment is open, give error "segment still open" and go to step 8

Step 2 : Read name of new segment

Step 3 : If name of segment is not valid, then show error " segment name not valid" and go to step 8

Step 4 : If name of segment is already present in segment table, show error " segment name already exist" and go to step 8

Step 5 : Initialize next free space in display file as start of new segment

Step 6 : Set size of new segment to zero and set all attributes to default

Step 7 : Inform that new segment is open

Step 8 : Stop

* Algorithm to delete segment -

→ Step 1 : Get name of segment to be deleted

Step 2 : If name is not valid then show error "Segment name not valid" and go to step 7

Step 3 : If segment is open, then show error "Cannot delete open segment" and go to step 7

Step 4 : If size of segment is less than 0, no processing is needed, go to step 7

Step 5 : All segments following segment to be deleted are shifted up by one position

Step 6 : Recover free space by resetting index of free space

Step 7 : Stop

- **Definition** -

Animation -
Computer animation is the art of creating moving images via the use of computers. It is a subfield of computer graphics and animation

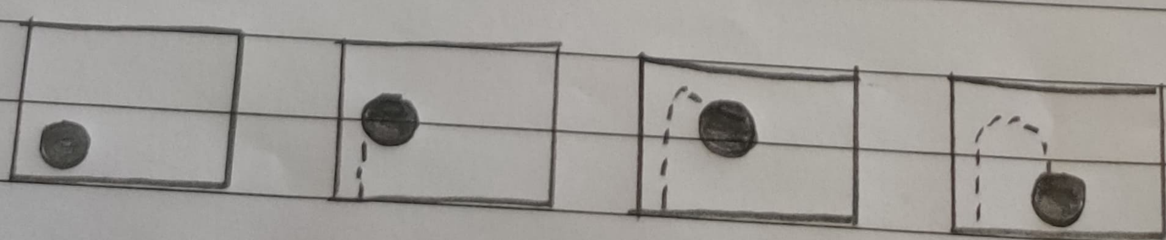
★ **Design of animation sequences**

- Animation sequence is designed by following steps :
  - Storyboard Layout
  - Object definition
  - Path specification
  - Keyframe specification
  - Generation of in-between frames

★ **Keyframes** -

- Keyframe animation does interpolation to generate intermediate frames from given pair of frames
- More keyframes generate smooth animation

- Eg,



★ Morphing -

- Morphing is a way of smoothly transformating one shape to another shape.
- Animator has to provide first & last images, called source and destination resp.

- Applications -
  - It is used to produce special effect in motion pictures and animations
  - It is used as transition technique between one scene and another.

★ NVIDIA Workstation –

→ • In gaming, we need a processor to perform many graphic objects simultaneously and in real time.

• NVIDIA workstation uses NVIDIA GPU which provides high floating point execution bandwidth, hardware multithreading and so on

• Features of NVIDIA gaming platform –

1) Highest level performance and smoothest experience
2) Enables developers to add graphics effects
3) Dedicated ray tracing hardware enables fast real-time ray tracing
4) Provides variable rate shading & faster refresh rates.