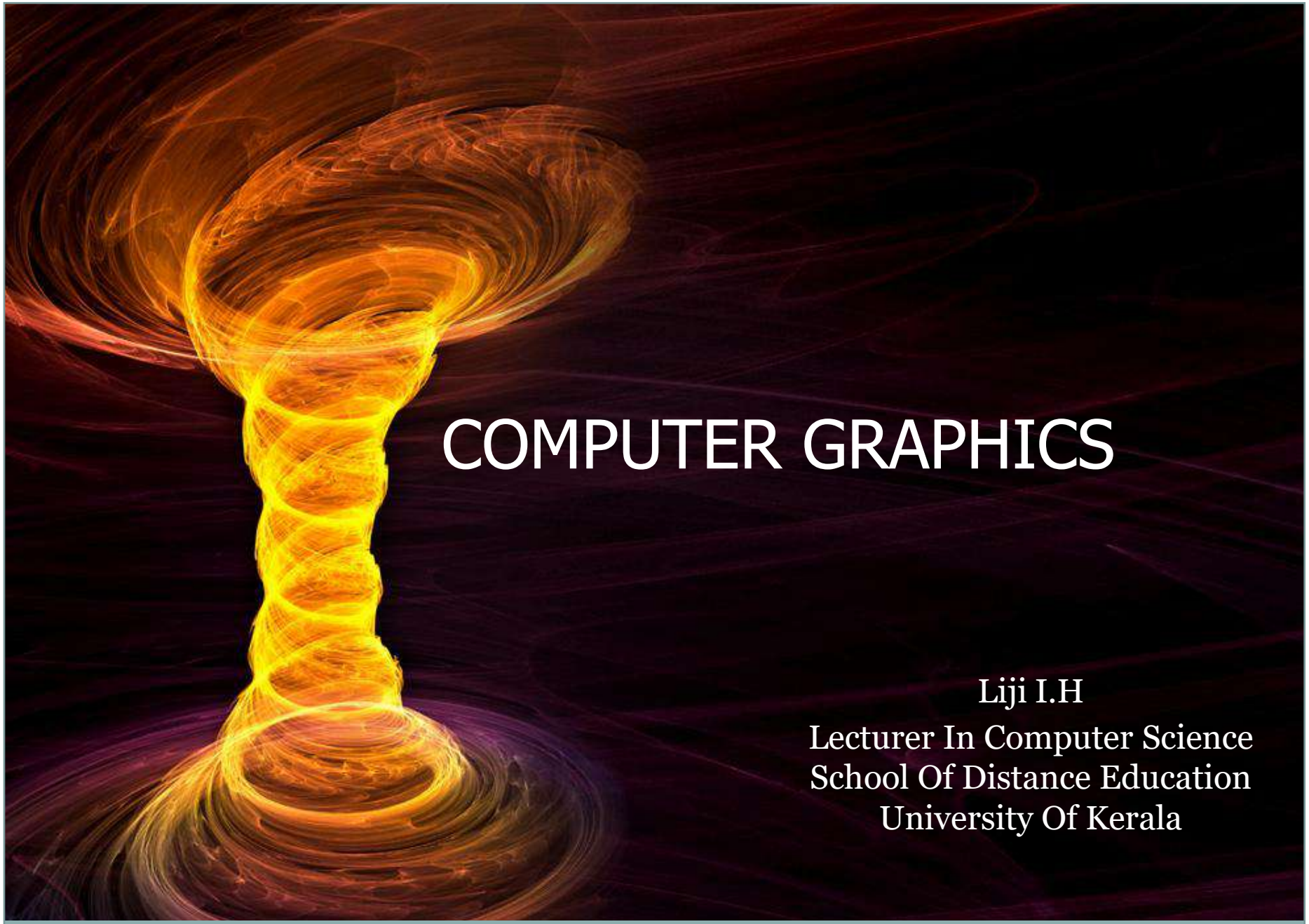


COMPUTER GRAPHICS

Liji I.H
Lecturer In Computer Science
School Of Distance Education
University Of Kerala

Course overview

- Introduction to Computer Graphics
- Video Display Device
- Random Scan & Raster Scan
- Color CRT Monitors
- Line drawing Algorithms
- Circle Drawing Algorithm
- Polygon Filling Algorithms
- 2D Transformations
- Homogenous Coordinates
- Composite Transformations
- Window to View Port Transformations
- Clipping
- 3D Transformations
- Projections
- Hidden Surface Removal Methods
- Color Spaces & Illumination Models
- Polygon Rendering Methods
- Animation Sequences

What is Computer Graphics?

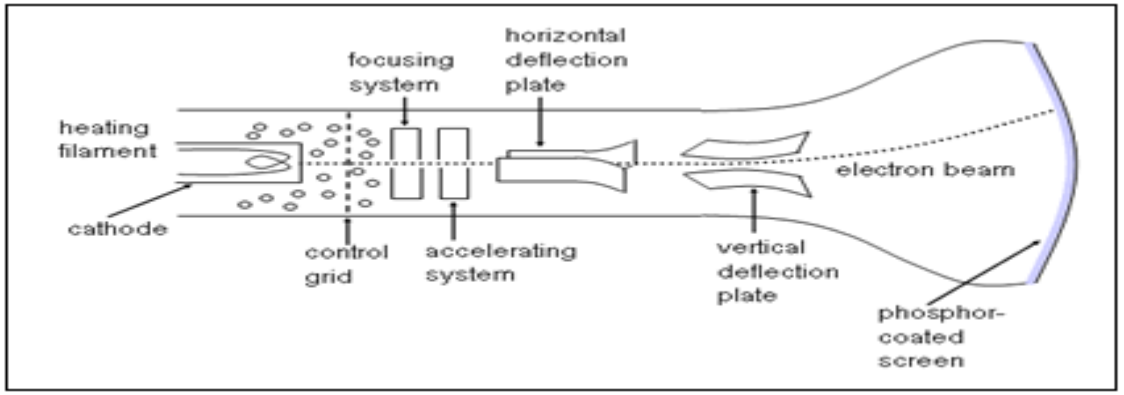
Computer graphics is a field of computer science, which deals with creation, representation and management of images on the computer screen.

Applications

- ✓ graphical user interface
- ✓ Computer-Aided Design
- ✓ Image Processing
- ✓ Entertainment
- ✓ Computer Art and commerce
- ✓ Simulation and modeling
- ✓ Cartography
- ✓ Scientific and business Visualization

Video Display Devices

Primary output device in a graphics system is a video monitor. The operation of most video monitors is based on the standard Cathode Ray Tube (CRT) .



Working Principle

- A beam of electrons (cathode rays), emitted by an electron gun, passes through focusing and deflection system that direct the beam toward specified positions on the phosphor-coated screen. The phosphor then emits a small spot of light at each position contacted by the electron beam.
- This type of display is called Refresh CRT.

Measurement of Performance

Persistence

Persistence is defined as the time it takes the emitted light from the screen to decay one-tenth of its original intensity.

pixel

A pixel is a smallest addressable screen element. On the monitor of a computer, a pixel is usually a square. Every pixel has a color and all the pixel together forms the picture.

Resolution

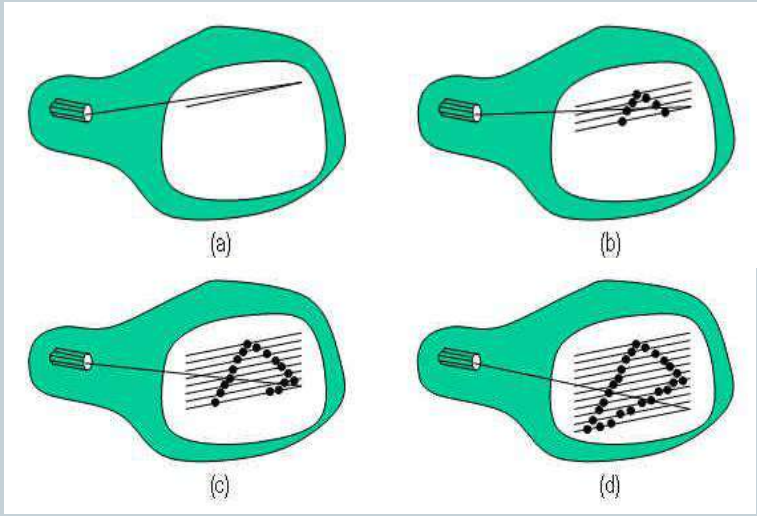
Resolution refers to the sharpness and clarity of an image. Resolution indicates the maximum number of points that can be displayed without overlap on a CRT.

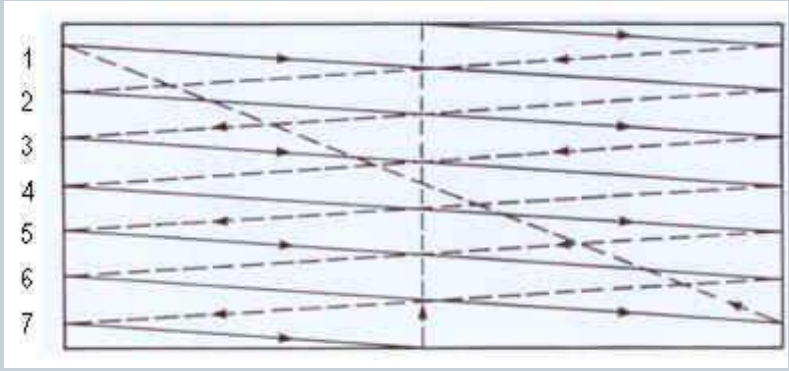
Aspect Ratio

Aspect ratio of an image or a screen is a proportional relationship between its width and height. Or This number gives the ratio of horizontal points to vertical points of an image or screen.

Raster Scan Displays.

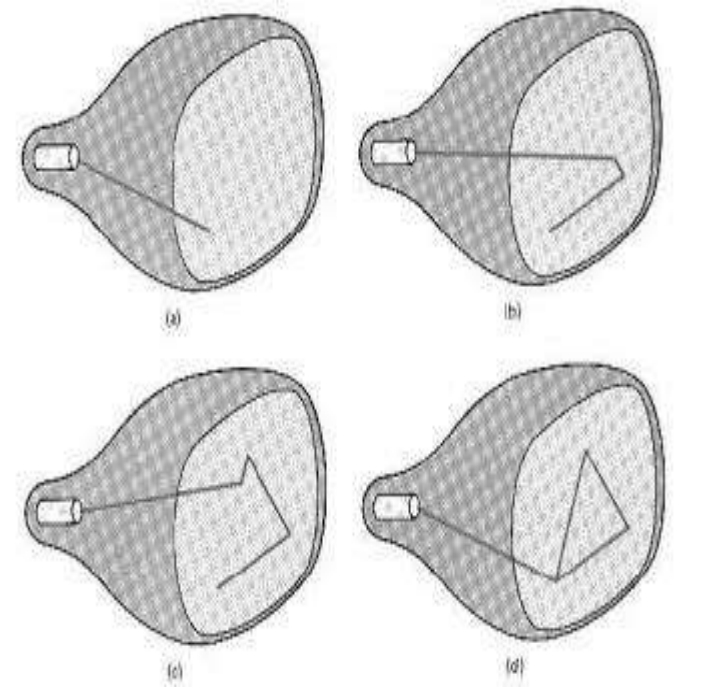
- In this electron beam is swept across the screen one row at a time from top to bottom.
- As beams moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots.
- Picture definition is stored in a memory area called the **frame buffer**.
- It well suited for the realistic display of scenes.
- The return of beams to the left of the screen, after refreshing each scan line, is called the **horizontal retrace** .
- At the end of each frame, the electron beam returns the top left corner of the screen to begin the next frame is called vertical retrace.





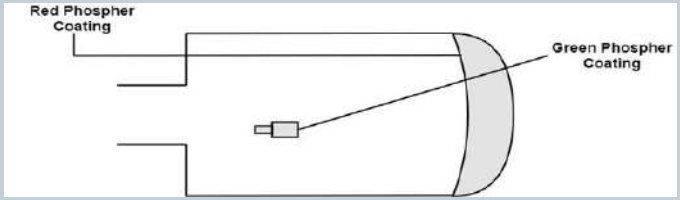
Random Scan Display

- Here electron beam is directed to only those parts of the screen where a picture is to be drawn.
- The picture definition is stored as a set of line-drawing commands in a refresh display file or a refresh buffer .
- Higher resolution than raster systems and can produce smooth line drawings.
- Also called **vector displays or stroke writing displays.**



Color CRT Monitors

Beam Penetration Method



- Random scan monitors use the beam penetration method for displaying color picture.
- Inside of CRT screen is coated two layers of phosphor namely red and green.
- It can produce maximum of 4 colors.
- Less Expensive
- Quality of picture is not much as good as another method.

Shadow Mask Method

- Raster scan systems use shadow mask methods to produce a much more range of colors.
- In this, CRT has three phosphor color dots, one emits a red , second emits green and third emits a blue light.
- produce realistic images using millions of colors and shadows scenes.
- Expensive
- Poor Resolution

Digital Differential Analyzer (DDA) Algorithm

- DDA is a scan conversion line algorithm.
- Given Starting and ending coordinates of a line (X_0, Y_0) and (X_n, Y_n).
- This algorithm attempt to generate points between starting and ending coordinates.

Steps

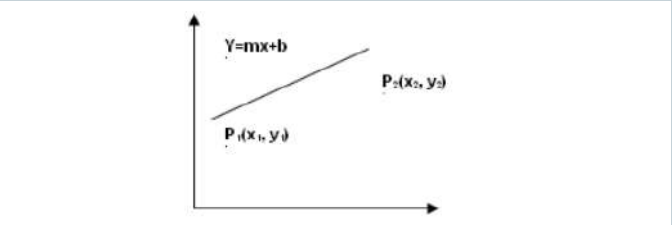
1. Calculate D_x, D_y and M from the given input.

$D_x = X_n - X_0$, $D_y = Y_n - Y_0$ $M = D_y / D_x$

2. If (absolute(D_x) > absolute(D_y)) then

steps= absolute(D_x);

Else steps=absolute(D_y);



3. If current point is (X_k, Y_k) then next point is

(X_{k+1}, Y_{k+1}) .

There are three cases.

- ✓ Case 1: If M<1 ,then X_{k+1} =roundoff$(1+X_k)$

Y_{k+1} = roundoff$(M+Y_k)$.

- ✓ Case 2:If M>1,then X_{k+1} =roundoff$(1/M+X_k)$

Y_{k+1} = round off $(1+Y_k)$.

- ✓ Case 3:If M=1,then,X_{k+1} =roundoff $(1 + X_k)$

Y_{k+1} = round off $(1+Y_k)$.

- Repeating the above step until the end points is reached

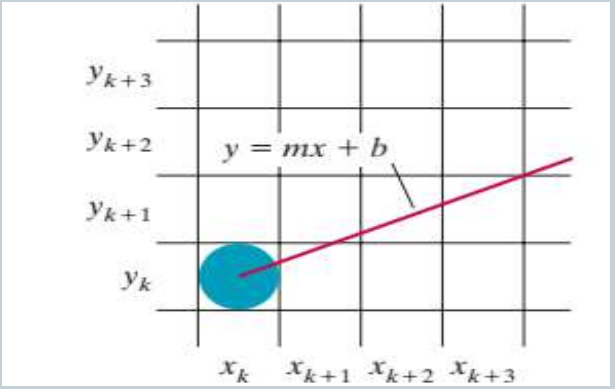
Bresenham's Line Algorithm

This algorithm is very efficient since it scan convert lines using only incremental integer calculations, so can be adapted to display circles and other curves.

- Input the two line end points and store the left end points in (X_0, Y_0) .
- Set the color for frame buffer position (X_0, Y_0) that is plot the first point.
- Calculate the constants D_x ,D_y , and starting value for Decision parameter $P_0 = 2D_y - D_x$.
- At each X_k along the line starting at k= 0 Perform the following task,

  If $P_k < 0$,the next point to plot is $(X_k + 1, Y_k)$ and $P_{k+1} = P_k + 2D_y$.

  Otherwise next point to plot is $(X_k + 1, Y_k + 1)$ $P_{k+1} = P_k + 2D_y - 2D_x$.
- Perform the above step until the end point is reached or D_x -1 times.



Advantages

- It is easy to implement and fast.
- It uses fixed points only and more accurate than DDA algorithm.

Disadvantages

- The resulted line is not smooth.
- This algorithm is for basic line drawing and cannot handle diminishing jaggies.

Midpoint Circle Algorithm

- The midpoint circle algorithm is an incremental algorithm for drawing circles.
- Here we use eight-way symmetry .
- So only calculate the points for the top right eighth of a circle, and then use symmetry to get the rest of the points.
- The equation of circle centered at origin and radius r is given by
- $f_{circle}(x,y)= x^2 +y^2-r^2$

Procedure

Step 1:Input radius r ,and circle center (x_c,y_c) then set coordinates for the first point on the circumference of a circle centered on the origin as,

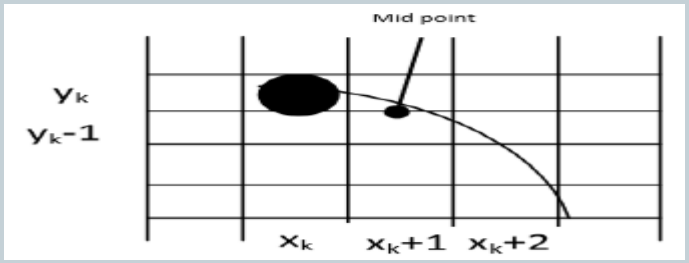
$$(x_0,y_0)=(0,r)$$

Step 2:Calculate initial value of decision parameter as

$$P_0 = 5/4 - r \ .$$

Step 3: At each X_k along the line starting at k= 0 Perform the following task,

If $P_k <0$,the next point to along the circle is (x_{k+1} ,y_k) and $P_{k+1} = P_k + 2x_{k+1} +1$.



Otherwise$(x_k +1,y_k -1)$ and $P_{k+1} = P_k + 2x_{k+1} +1-2y1$

where $2x_{k+1}=2x_k+2$ and $2y_{k+1}=2y_k -2$

Step 4:Determine symmetry points in other seven octants.

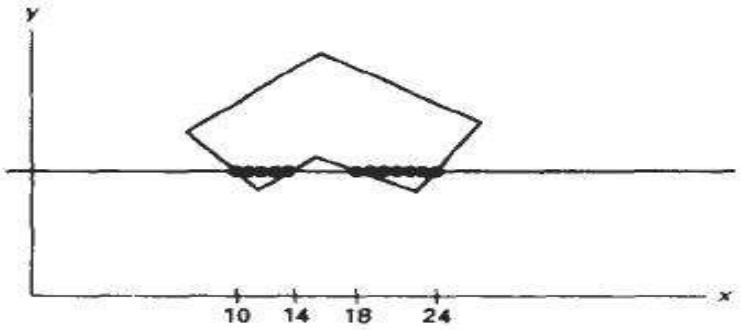
Step 5:Move each calculated pixel position(x,y) onto the next point along the circle is the circular path centered at(x_c,y_c) and plot the coordinate values.

$$x=x+ x_c, \quad y= y+ y_c$$

Step 5:Repeat step 3 through 5 until x>=y.

Polygon Filling Algorithms

- A polygon can be represented as a group of connected edges ,forming a closed figure.
- Polygon filling algorithms are used for coloring the area of the polygon.
- ✓ **In Scan-Line Polygon Fill Algorithm** For each scan line crossing a polygon, this algorithm locates the intersection points of the scan line with the polygon edges.
- ✓ These intersection points are then sorted from left to right, and the corresponding frame buffer positions between each intersection pair are set to the specified fill color.

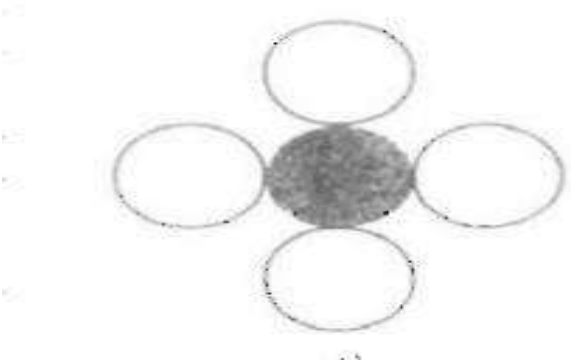


Polygon Filling Algorithms

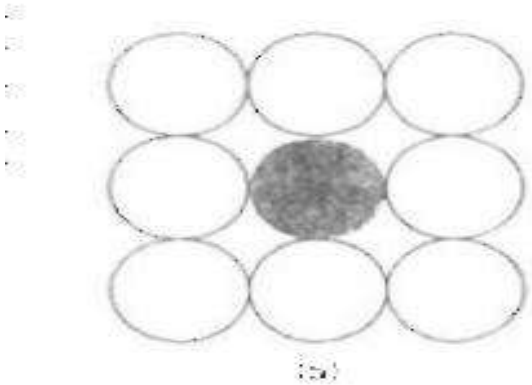
Boundary Fill Algorithm

- The algorithms which are defining the boundary of the polygon are called boundary fill algorithm.
- In boundary filling method ,start from an interior point(seed) and paint interior outward towards the boundary.
- If the boundary is specified in a single color , the the fill algorithm proceeds outward pixel by pixel until the boundary color encountered.

4 Connected



8 Connected



2D Transformations

- Transformation is a process of modifying and repositioning the existing graphics.
- When a transformation takes place on a 2D plane, it is called 2D transformation.
- Basic Transformations are Translation ,Rotation, Scaling.
- **Translation** is applied to an object by re-positioning it along a straight line path from one co-ordinate location to another.

Translation

Here translate a two-dimensional point(x,y) by adding translation distances, t_x, t_y then getting a new position (x', y') as:

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \qquad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\therefore P' = P + T$$

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

2D Transformations

- In rotation, we rotate the object at particular angle θ (theta) from its origin along a circular path in XY plane. You will get a new point P′ (X′, Y′).

Using standard trigonometric , P(X, Y) can be represented as

$x = r\cos\phi$

$y = r\sin\phi$

$x' = r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta$

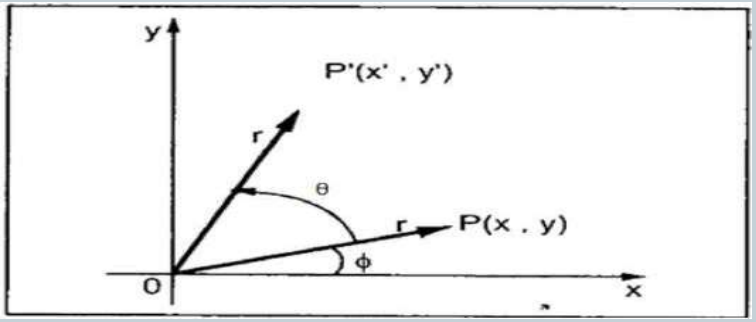
$y' = r\sin(\phi+\theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta$

Substituting value of x,y in x′,y′ we get

$x' = x\cos\theta - y\sin\theta$

$y' = x\sin\theta + y\cos\theta$

Rotation



Representing the above equation in matrix form,

$$[X'\ Y'] = [X\ Y] \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

OR

$$P' = P \cdot R, \quad \text{Where } R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2D Transformations

- A scaling transformation alters the size of the object.
- Scaling is carried out by multiplying the co-ordinate values (x, y) of each vertex by scaling factor s_x and s_y to produce transformed co-ordinates (x', y').
- i.e. $x' = x.s_x$

  $y' = y.s_y$
- If both scaling factors have same value then the scaling is known as **uniform scaling**.
- Otherwise it is called differential Scaling.

Scaling

- The matrix equation for scaling is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

OR

$$P' = S.P$$

Homogenous Coordinates

- We can combine the multiplicative and translational terms for 2D geometric transformations into a single matrix representation by expanding 2x2 matrix to 3x3.
- In this way, we can represent the point by 3 numbers instead of 2 numbers, which is called Homogenous Coordinate system.

$$(x, y) \rightarrow (x_h, y_h, h)$$

- In this system, we can represent all the transformation equations in matrix multiplication.

Translation

$P' = T(t_x, t_y).P$

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

Rotation

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

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling $P' = S(S_x, S_y).P$

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

Composite Transformations

- It is sequence of geometric transformations.

Example :Animation

- **Translations**

  If two successive translation vectors (t_{x1}, t_{y1}) and (t_{x2}, t_{y2}) are applied to a coordinate position P, the final transformed location P' is calculated as

$$P' = T(t_{x2}, t_{y2}).\{T(t_{x1}, t_{y1}).P\}$$

$$P' = \{T(t_{x2}, t_{y2}).T(t_{x1}, t_{y1}).P\}$$

$$\begin{bmatrix} 1 & 0 & t_{x2} \\ 0 & 1 & t_{y2} \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & t_{x1} \\ 0 & 1 & t_{y_1} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{x1} + t_{x2} \\ 0 & 1 & t_{y1} + t_{y2} \\ 0 & 0 & 1 \end{bmatrix}$$

$T(t_{x2}, t_{y2}).T(t_{x1}, t_{y1}) =$

$T(t_{x1} + t_{x2}, t_{y1} + t_{y2})$

which demonstrates that two successive translations are additive.

Composite Transformations

Rotation

- Two successive rotations applied to point **p** produce the transformed position

$$P' = R(\theta_2).\{R(\theta_1).P\}$$
$$= \{R(\theta_2).R(\theta_1)\}.P$$

- By multiplying the two rotation matrices, we can verify that two successive rotations are additive:

$$R(\theta_2).R(\theta_1) = R(\theta_1 + \theta_2)$$

- So that the final rotated coordinates can be calculated with the composite rotation matrix as

$$P' = R(\theta_1 + \theta_2).P$$

Scaling

- Concatenating transformation matrices for two successive scaling operations produces the following composite scaling matrix:

$$\begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & sy_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x1}.s_{x2} & 0 & 0 \\ 0 & s_{y1}.s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$s(s_{x2}, s_{y2}).s(s_{x1}, s_{y1}) = s(s_{x1}.s_{x2}, s_{y1}.s_{y2})$$

- Indicates that successive scaling operations are multiplicative.

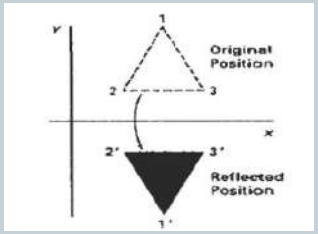
Other Transformations

Reflection

- Reflection is a transformation that produces a mirror image of an object.
- In 2D-transformation, reflection is generated relative to an axis of reflection.
- Relative axis of reflection, is same as 180^o rotation about the reflection axis

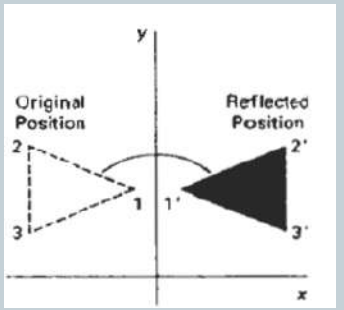
Reflection about X-axis

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



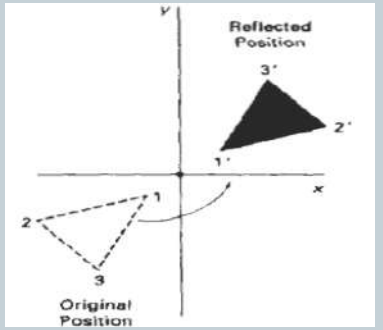
Reflection about Y-axis

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



Reflection about origin

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



Other Transformations

Shearing

- A transformation that distorts the shape of an object is called shearing.
- **X-direction Shear** : shearing relative to x-axis is ,

  which transforms

  $x' = x + sh_x$ and y' = y

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & s_{hx} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- **Y-direction shear:**

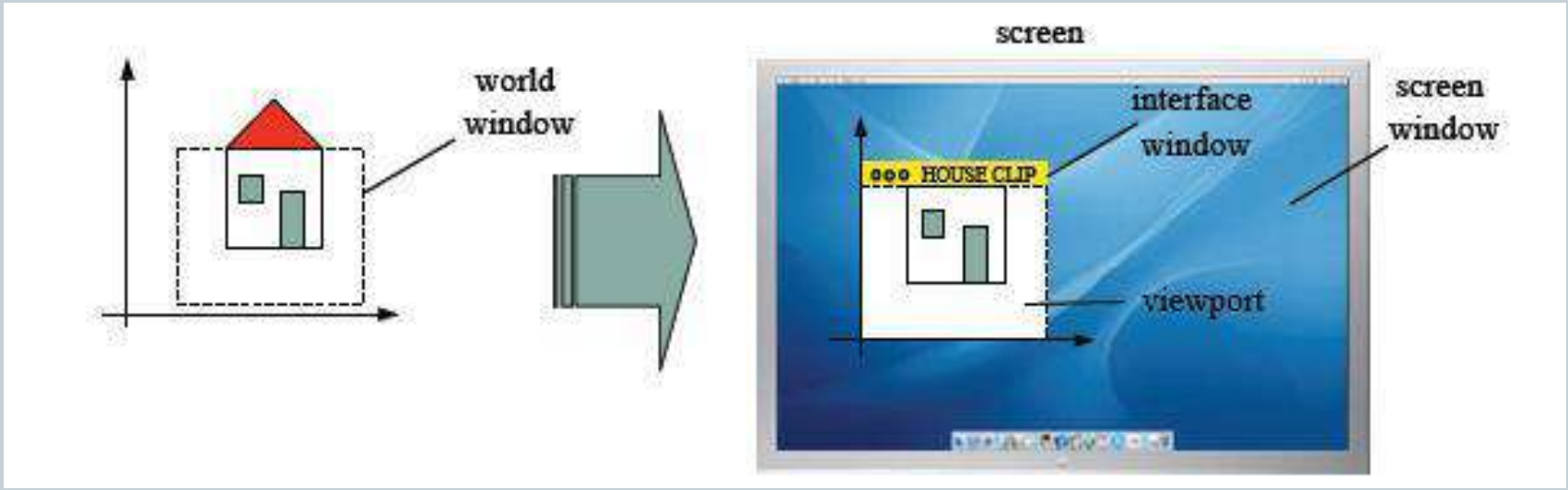
  Any-direction shear relative to y-axis which transforms

  x‘= x and y'= $x.sh_y+y$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ s_{hy} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Window to Viewport Transformation

- **Window**: A world-coordinate area selected for display.
- **Viewport**: An area on a display device to which a window is mapped.
- The window defines ***what*** is to be viewed
- the viewport defines ***where*** it is to be displayed



Window to Viewport Transformation

- A point at position (x_w, y_w) in the window is mapped into position (x_v, y_v) in the associated viewport .

$xv = xv_{min} + (xw - xw_{min})S_x$

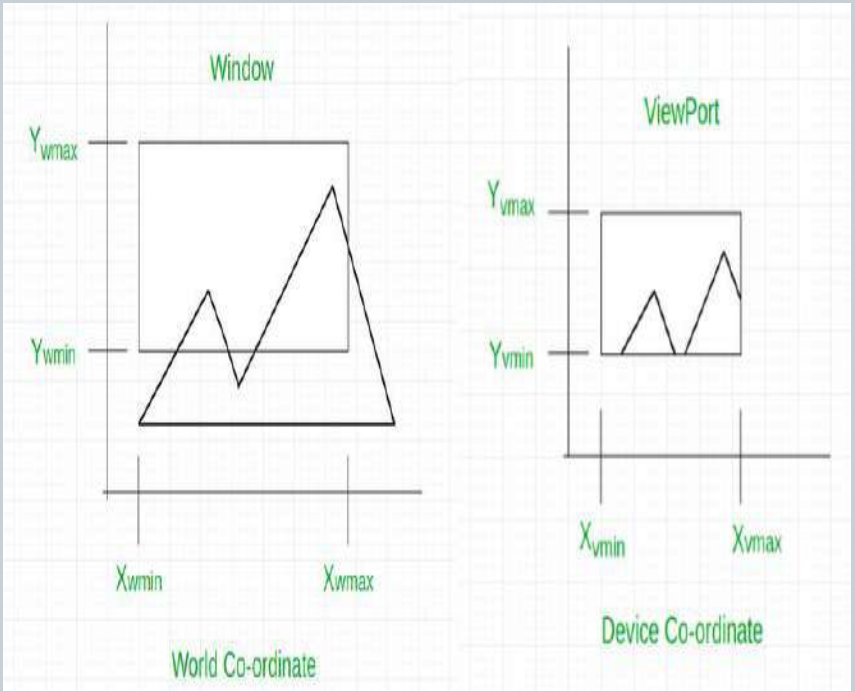
$yv = yv_{min} + (yw - yw_{min})S_y$

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

$$S_x = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$S_y = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$



Where S_x, S_y, are scaling factors

Clipping

- Any procedure that identifies those portions of a picture that are either inside or outside of a specified region of space is called clipping.

Point Clipping

Clip window is a rectangle in standard position, we save a point

P =(x,y) for display if the following inequalities are satisfied:

- The edges of the clip window **(xw_{min}, xw_{max}, yw_{min}, yw_{max})** can be either the world-coordinate window boundaries or viewport boundaries.
- If any one of these four inequalities is not satisfied, the point is clipped .

$xw_{min} \leq x \leq xw_{max}$ $\quad$ $yw_{min} \leq y \leq yw_{max}$

Line Clipping

- First, test a given line segment to determine whether it lies completely inside or completely outside the clipping window.
- If line is not completely inside or completely outside, perform intersection calculations with clipping boundaries.
- Here use inside-outside test .

Line Clipping

Cohen-Sutherland Line Clipping

- To clip a line, we need to consider only its endpoints.
- If both endpoints of a line lie inside the window, the entire line lies inside the window. No clipping is required.

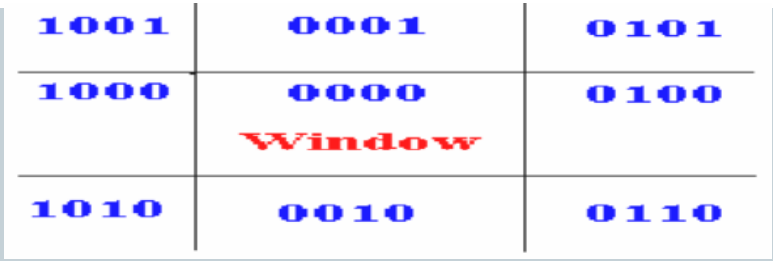
Inside-Outside Window Codes

- Each of the nine regions associated with the window is assigned a 4-bit code to identify the region

1.If both endpoints of line have a region code **0000** then accept this line.

2.Else, perform the logical **AND** operation for both region codes.

If the result is not **0000,** then reject the line.

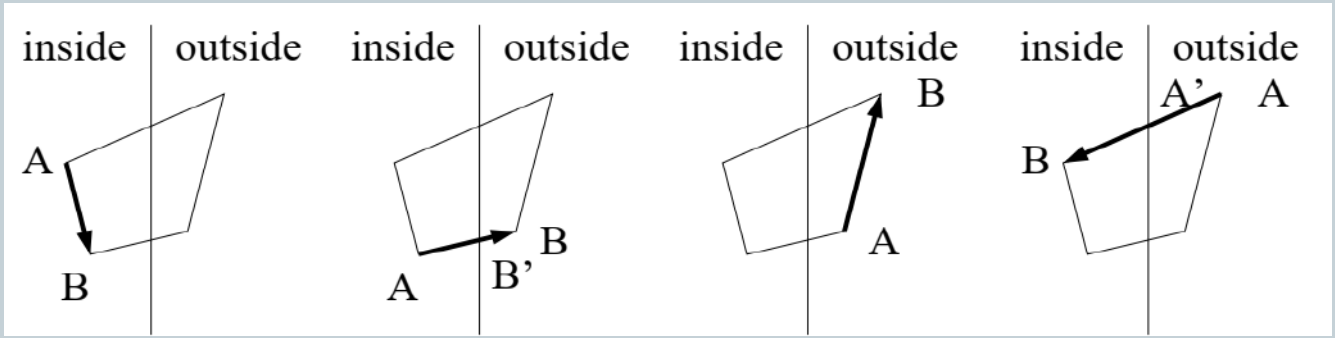


Else you need clipping.

- ✓ Choose an endpoint of the line that is outside the window.
- ✓ Find the intersection point at the window boundary based on region code .
- ✓ Replace endpoint with the intersection point and update the region code.
- ✓ Repeat step 1 until we find a clipped line either trivially accepted or trivially

Sutherland - Hodgeman Polygon Clipping

- The Sutherland - Hodgeman algorithm performs a clipping of a polygon against each window edge in turn.
- There are four possible cases when processing vertices sequence around the perimeter of a polygon
- Assuming vertex A has already been processed,
  Case 1 — vertex B is added to the output list
  Case 2 — vertex B’ is added to the output (edge AB is clipped to AB’)
  Case 3 — no vertex added (segment AB clipped out)
  Case 4 — vertices A’ and B are added to the output (edge AB is clipped to A’B)



3D Object Representation

- A two dimensional(2D) object is an object that has two dimensions such as length and width only and no thickness or height.
- A three dimensional object is an object with three dimensions :a length, width and height or depth.
- Representation schemes for solid objects are:

✓ Boundary Representation (B-reps)
✓ Space Partitioning representation:

Boundary Representation (B-reps):
It describes a three dimensional object as a set of surfaces that separate the object interior from the environment.
Eg: polygon facets and spline patches.

Space Partitioning representation:
It describes the interior properties, by partitioning the spatial region containing an object into a set of small, non overlapping, contiguous solids(usually cubes). Eg: Octree Representation.

3D Transformations

Translation

- Translate a 3D point by adding translation distances, t_x, t_y and t_z, to the original coordinate position (x,y,z):

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

  $z' = z + t_z$

Transformation matrix is,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling (w.r.t the Origin)

- Here scaling factors S_x, S_y and S_z, which are multiplied to the original vertex coordinate positions (x,y,z):

  $x' = x * S_x \qquad y' = y * S_y$

  $z' = z * S_z$

Transformation matrix is,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Transformations

Rotation

- A 3D rotation can be specified around any line in space.

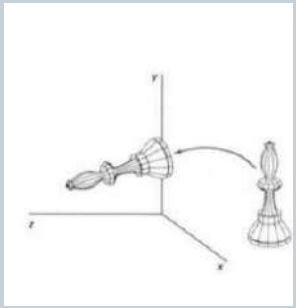
X axis Rotation

The equation for X-axis rotation

$x' = x$

$y' = y\cos\theta - z\sin\theta$

$z' = y\sin\theta + z\cos\theta$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Y axis rotation

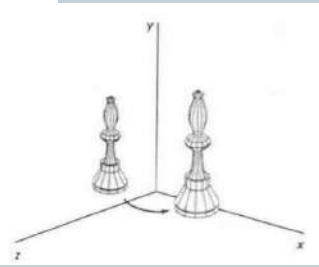
The equation for Y-axis rotaion

$x' = x\cos\theta + z\sin\theta$

$y' = y$

$z' = z\cos\theta - x\sin\theta$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



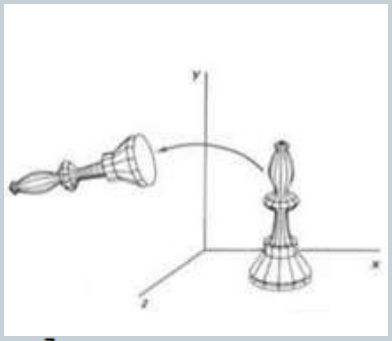
Z axis Rotation

The equation for z-axis rotaion

$x' = x\cos\theta - y\sin\theta$

$y' = x\sin\theta + y\cos\theta$

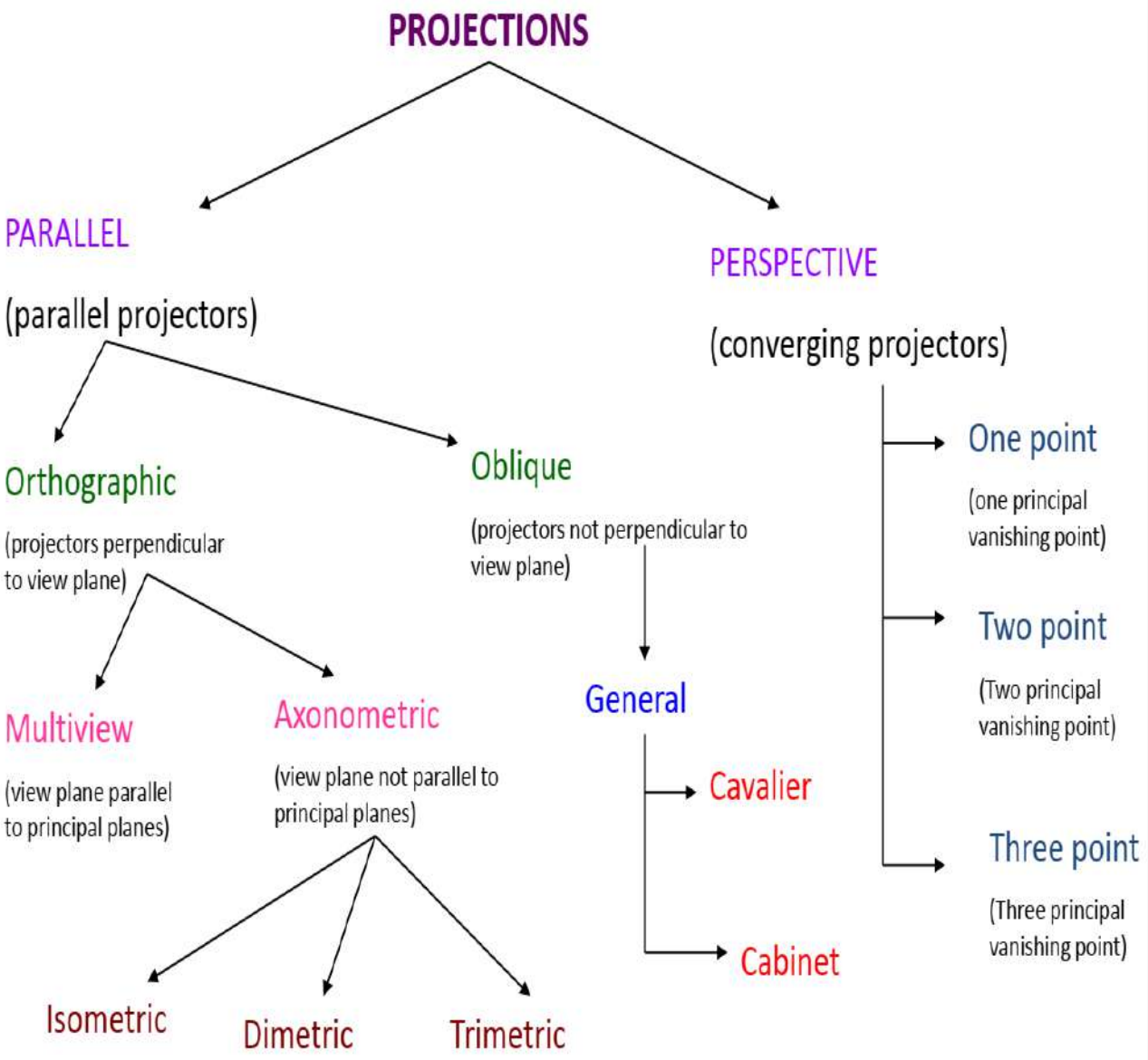
$z' = z$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Projections

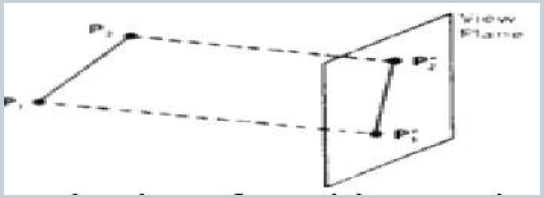
Projection means transformation of a three dimensional area into two dimensional area.



Projections

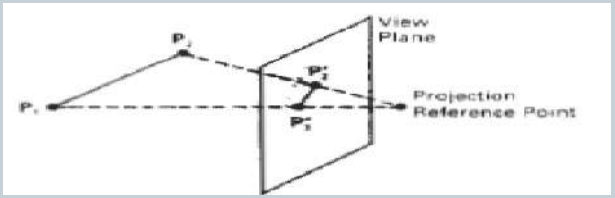
Parallel Projection

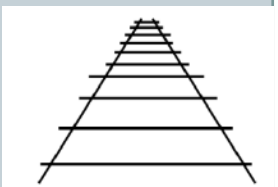
- In parallel projection, co-ordinates positions are transformed to the view plane along parallel lines.
- Not give us a realistic representation of the appearances of 3D objects ,but Preserves relative proportions of objects.



Perspective Projection

- In perspective projection, objects positions are transformed to the view plane along lines that converge to a point called projection reference point (centre of projection).
- Produces realistic views but does not preserve relative proportions.
- Under perspective projections, any set of parallel lines that are not parallel to the Projection plane will converge to a point called **vanishing point.**





Parallel Projections

- In parallel projection, When the projection is perpendicular to view plane it is called orthographic projection.
- In orthographic projection when projection lines making an angle 90^0 with any principal axis ,it is called multi view orthographic projection.
- Otherwise, It is called is called axonometric orthographic projection.
- In axonometric projection, if the projection lines makes equal angles with the three principal axes it is isometric .

- With any two principal axes it is diametric
- Not make any equal angles with principal axes it is trimetric projection.
- When the projection lines are not perpendicular to the view plane, it is oblique projection.
- If the projection lines make an angle of 45^0 view plane, $\tan \alpha = 1$,the views obtained are called cavalier
- When the projection angle chosen so that it is approximately $63.4^{0,}$ $\tan \alpha = 2$,the resulting view is a cabinet projection.

Hidden Surface Removal Methods

- Also called Visible Surface Detection Method
- Its major concern for identifying those parts of a scene that are visible from a chosen viewing position.

These two approaches are:

- ✓ **In object-space method:** compares objects and parts of objects to each other within the scene definition determine which surfaces, should label as visible.
- ✓ **In Image-Space methods**: Visibility is decided, point by point at each pixel position on the projection plane.

  Eg.: Z buffer Method, Scan line Method

Hidden Surface Removal Methods

Z Buffer Method

- It compares surface depths at each pixel position on the projection plane.
- It is called z-buffer method since object depth is usually measured from the view plane along the z-axis of a viewing system.
- The method is usually applied to scenes containing only polygon surfaces.
- In Z-buffer method, two buffers area are required.
- **Depth buffer (z-buffer)**: stores the depth value for each (x, y) position as surfaces are processed.
- **Refresh buffer (Image buffer)**: stores the intensity value for each position.

Scan line Method

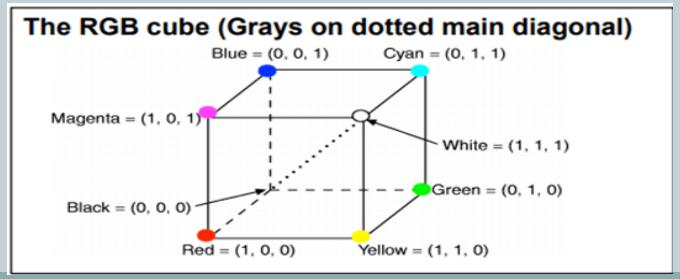
- As each scan line is processed, all polygon surfaces intersecting that line are examined to determine which are visible.
- Across each scan line, depth calculations are made for each overlapping surface to determine which is nearest to the view plane.
- When the visible surface has been determined, the intensity value for that position is entered into the image buffer.
- A set of tables are-set up for the various surfaces- Edge table and a Polygon table.

Color Spaces

- Color spaces are the mathematical representation of a set of colors.

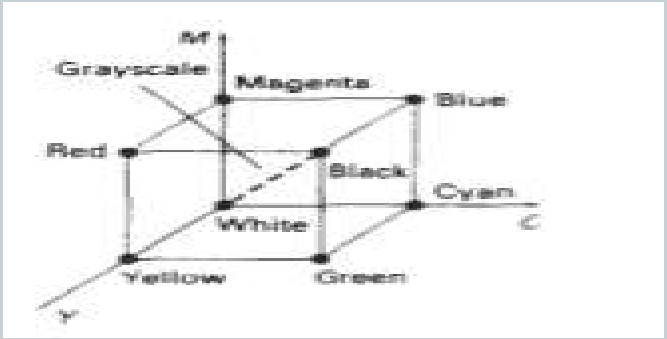
RGB Color Space

- This theory of vision is the basis for displaying color output on a video monitor using three primary colors red ,green and blue referred to as RGB model.
- Represented by the unit cube defined on R,G,B axes .



CMY Color Model

- Defined by the primary colors Cyan, Magenta and Yellow(CMY).
- This is useful for produce a color pattern for hardcopy devices.
- A unit cube representation of the CMY model is shown as,

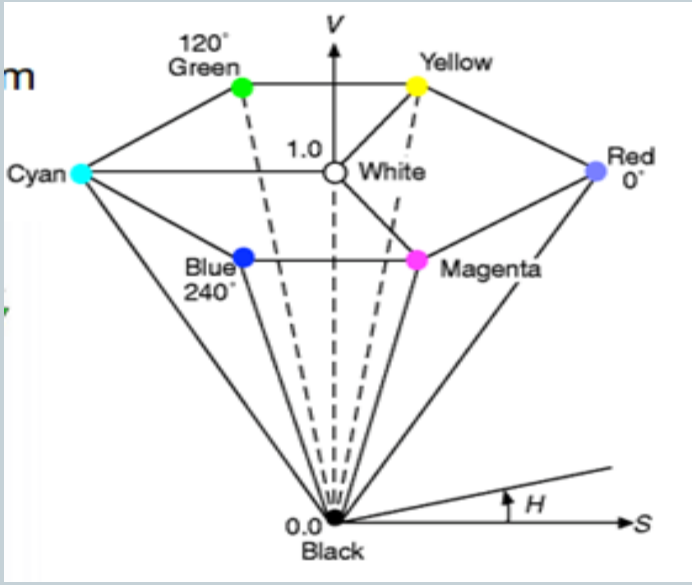


Color Spaces

HSV Color Model

- HSV stands for Hue, Saturation, and Value (brightness).
- To give color specifications ,a user select color and the amount of white and black that is to be added to obtain different shades, tints and tones.
- It is a hexcone subset of the cylindrical coordinate system.

- The vertical axes in this model is called Lightness(L).
- Saturation parameter S specifies the relative purity of a color.



Illumination Model

- The intensity of light seen on each surface depends up on the type of light sources and the surface characteristics of the object.
- If incident light energy is reflected or transmitted, then the object is visible.

Light Sources

- ✓ **Light Reflecting Sources:** These are the sources which reflected the illumination .
- ✓ **Light Emitting Sources**

  Eg:Sun,Bulb etc

- **Point Source :**The simplest model for light emitting is Point source. Sources that are sufficiently far from the scene can be accurately modeled as point source. For example Sun.
- **Distributed Light Source**: This occurs when we have a large nearby source. In this case, the illumination effects cannot be approximated realistically with a point source. For example –tube light.

Basic Illumination Models & Rendering

- An **illumination model** is used to calculate the intensity of the light that is reflected at a given point on a surface.
- A **rendering method** uses intensity calculations from the illumination model to determine the light intensity at all pixels in the image.
- Rendering is the process a computer uses to create an image from a data file.

Ambient Light

- The amount of ambient light incident on each object is a constant for all surfaces and over all directions.
- We can set the level for the ambient light in a scene with parameter I_a and each surface is then illuminated with this constant value.
- The resulting reflected light is a constant for each surface independent of the viewing direction and the spatial orientation of the surface.
- The intensity of reflected light for each surface depends on the optical properties of the surface

Basic Illumination Models & Rendering

Diffuse Reflection

- It is constant over each surface in a scene, independent of the viewing direction.

Specular Reflection

- If we look at an illuminated shiny surface ,we see a highlight or bright spot at certain viewing directions. This is called specular reflection.
- It is the result of total refection of the incident light in a concentrated region around the specular reflection angle.

Intensity Attenuation

- If two parallel surfaces with same optical parameters overlap they would be indistinguishable from each other. The two surfaces would be displayed as one surface.
- Problems by using inverse linear or quadratic functions to attenuate intensities.

Polygon Rendering Methods

Phong Shading

- A more accurate method for rendering a polygon surface is to interpolate normal vectors, and then apply the illumination model to each surface point.
- Also called normal vector interpolation shading.
- It greatly reduces the Mach-band effect.

Steps:

- ✓ Determine the average unit normal vector at each polygon vertex.
- ✓ Linearly interpolate the vertex normals over the surface of the polygon.
- ✓ Apply an illumination model along each scan line to calculate projected pixel intensities for the surface points.

Polygon Rendering Methods

Constant-Intensity Shading (Flat Shading)

- In this single intensity is calculated for each polygon.
- All points over the surface of the polygon are then displayed with the same intensity value.
- Useful for quickly displaying the general appearance of a curved surface.

Gouraud Shading

- Renders a polygon surface by linearly interpolating intensity values across the surface.
- Intensity values for each polygon are matched with the values of adjacent polygons along the common edges.
- Thus eliminating the intensity discontinuities that can occur in flat shading.

Steps

- ✓ Determine the average unit normal vector at each polygon vertex.
- ✓ Apply an illumination model to each vertex to calculate the vertex intensity.
- ✓ Linearly interpolate the vertex intensities over the surface of the polygon

Design of Animation Sequences

- **Storyboard layout** :
- ✓ The storyboard is an outline of the action.
- **Object definitions:**
- ✓ Given for each participant in the action.
- ✓ Objects can be defined in terms of basic shapes
- **Key-frame specifications**:
- ✓ A detailed drawing of the scene at a certain time in the animation sequence.
- **Generation of in-between frames:**
- ✓ The intermediate frames between the key frames.
- ✓ The number of in-betweens needed is determined by the media to be used to display the animation.

Morphing

Transformation of object shapes from one form to another is called morphing.

Tweening

The process of generating intermediate frames between two images to give the appearance that the first image evolves smoothly into the second image.

Image warping

Image warping is the process of digitally manipulating an image such that any shapes portrayed in the image have been significantly distorted.

Zooming

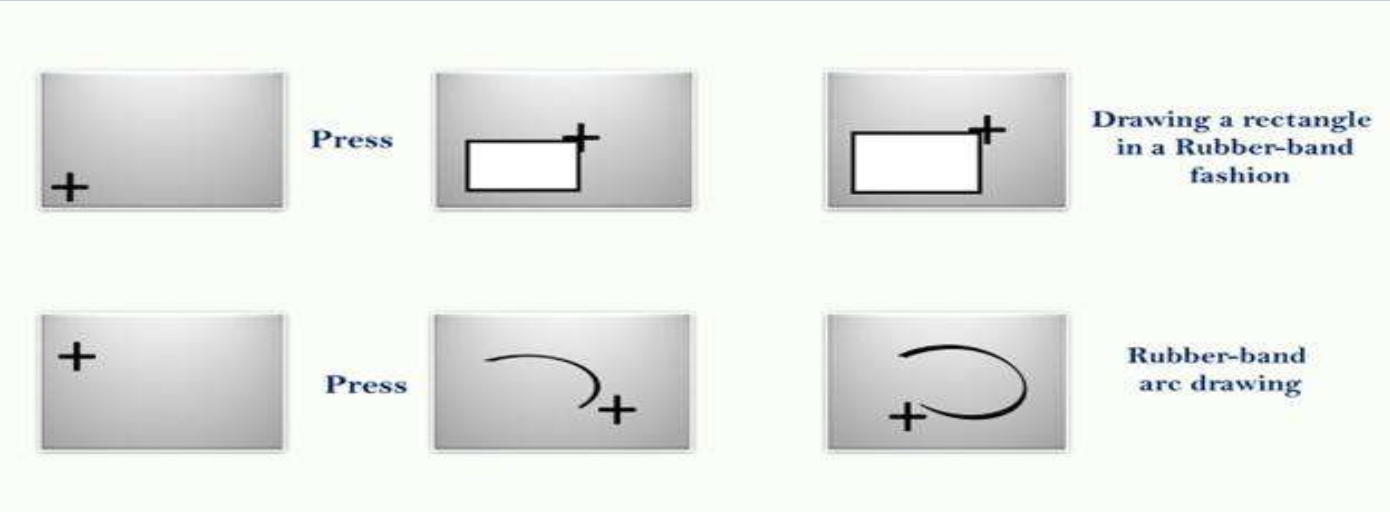
The transformation effectively scales down or blows up a pixel map or a portion of it with the instructions from the user.

Panning

- ✓ Acts as a qualifier to the zooming transformation.
- ✓ Moves the scaled up portion of the image to the center of the screen and depending on the scale factor, fill up the entire screen.

Rubber Band Techniques

Popular technique of drawing geometric primitives such as line, polylines, rectangle, circle and ellipse on the computer screen.



Thank You.....