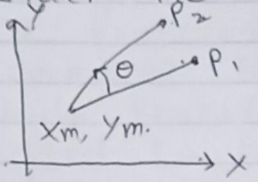
Q1 With neat labelled diagram explain Composite Transformations?

Ans: ① If the ~~figur~~ reference point of rotation is other than origin, then we need to follow a series of basic transformations

② Such transformation is also called as composite transformation

③ Consider the following figure:-



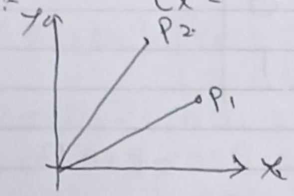
④ If we want to rotate point P_1 with respect to reference point $X_m Y_m$ we should perform the following steps:

ⓐ Translate the reference point to origin:

Translation vector is given as:-

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -X_m & -Y_m & 1 \end{bmatrix}$$

here:- $t_x = -X_m$ & $t_y = -Y_m$

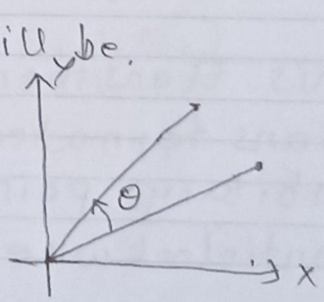


ⓑ Apply ~~reference~~ rotation by given angle θ.

The angle of rotation is anticlockwise in direction.

So our rotation matrix will be.

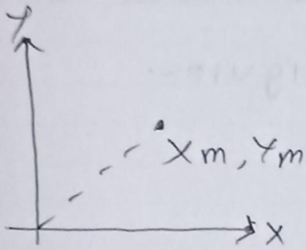
$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Now translate reference point back to its actual position.

∴ So translation matrix T_2 will become:-

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ X_m & Y_m & 1 \end{bmatrix}$$



Now, let us form a combined matrix.

$$\therefore M = T_1 * R * T_2$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -X_m & -Y_m & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ X_m & Y_m & 1 \end{bmatrix}$$

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

This transformation matrix is the overall transformation matrix for rotation about arbitary point (X_m, Y_m) by an angle θ in anticlockwise direction.

Q2 What are Homogenous Co-ordinates? Prove that two successive rotations are additive.

(A) Ans: ① Translation of point by the change of co-ordinate cannot be combined with other transformation by using simple matrix application. Such a combination is essential if we wish to rotate an image about a point other than origin by translation, rotation and agains translation.

② To combine these three transformations into a single transformation, homogenous co-ordinates are used. In homogenous co-ordinate system, two dimensional co-ordinate positions (x, y) are represented by triple co-ordinates i.e (x, y, z).

(B) Proof: $R_1(\theta_1) * R_2(\theta_2) = R(\theta_1 + \theta_2)$

Let us rotate a point (x, y) in anti-clockwise direction by the angle θ_1 followed by angle θ_2.

Let R_1 = Transformation Matrix rotated by angle θ_1

Let R_2 = Transformation Matrix rotated by angle θ_2.

Now,

$$R_1(\theta_1) * R_2(\theta_2) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2 & -\cos\theta_1\sin\theta_2 - \sin\theta_1\cos\theta_2 \\ \sin\theta_1\cos\theta_2 + \cos\theta_1\sin\theta_2 & -\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix}$$

$$= R[\theta_1 + \theta_2]$$

$\therefore R_1(\theta_1) * R_2(\theta_2) = R[\theta_1 + \theta_2]$

$\therefore$ Successive Rotation are additive

Q3 With neat diagram explain Sutherland Hodgman Polygon clipping algorithm.

Ans: It is performed by processing the boundary of polygon against each window corner edge. First of all entire polygon is clipped against one edge, then resulting polygon is considered, then the polygon is considered, then the polygon is considered against the second edge, so on for all four edges. The algorithm should be performed in clockwise order.

- There are four possible cases for any given of given polygon against current clipping edge:

a) Both vertices are inside: Only the second vertex is added to the output list.

b) First vertex is outside while second one is inside: Both the point of intersection of the edge with the clip boundary and the second vertex are added to the output list.

c) First vertex is inside while second one is outside: Only the point of intersection of the edge with the clip boundary is added to the output list.

d) Both vertices are ~~output~~ outside: No vertices are added to the output list.

★ Diagram.

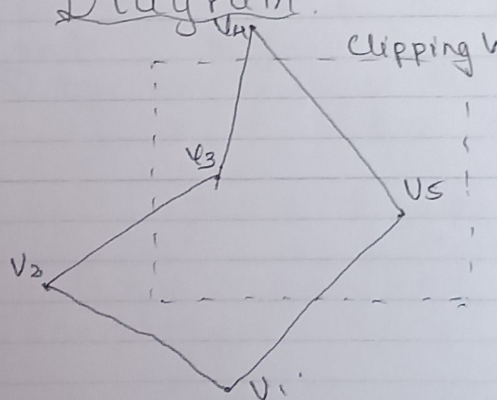
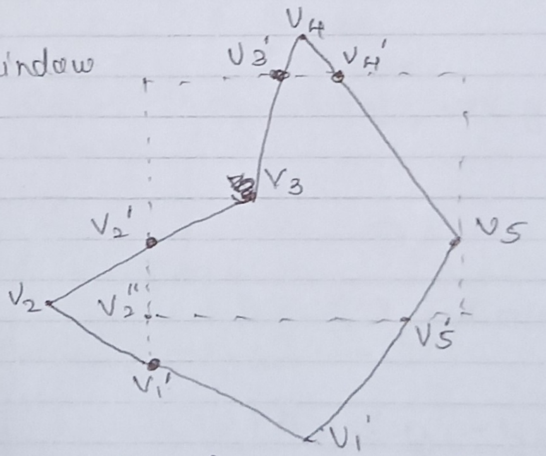


Fig. Before Clipping.



After Boundary Clipping

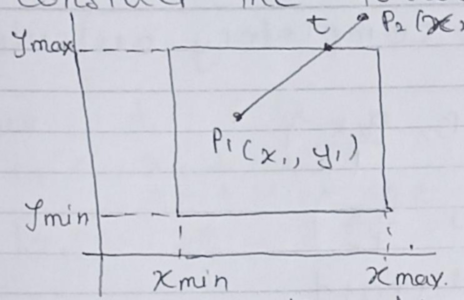
- Sutherland-Hodgeman Polygon Clipping Algorithm

1) Read co-ordinates of all vertices of the polygon.
2) Read co-ordinates of the clipping window.
3) Consider the left edge of the window.
4) Compare the vertices of each edge of the polygons individually with the clipping plane.
5) Save the resulting intersections and vertices in the new list of vertices according to four possible relationships between the edge and the clipping boundary discussed earlier.
6) Repeat the steps 4 and 5 for remaining edges of the clipping window. Each time the resultant list of vertices is successively passed to process the next edge of the clipping window.
7) Stop.

Q4 With neat diagram explain Liang Barsky line clipping algorithm. Apply the algorithm to line (30, 60) and (60, 25) against the window (Xmin, Ymin = 10, 10) and (Xmax, Ymax = 50, 50).

Ans ① Liang - Barsky algorithm uses the paramatric equation of line.

② Consider the following figure:-



Let's consider the line with endpoints $P_1(x_1, y_1)$ & $P_2(x_2, y_2)$

Parametric representation of line is given as:-

$x = x_1 + t \cdot \Delta x$

$y = y_1 + t \cdot \Delta y$

wher 't' is a parameter that controls the line points i.e $0 \leq t \leq 1$.

③ Point is inside the clipping region only if the following inequalities are true:-

$x_{min} \leq x \leq x_{max}$

i.e $x_{min} \leq x_1 + t \cdot \Delta x \leq x_{max}$

and $y_{min} \leq y_1 + t \cdot \Delta y \leq y_{max}$

④ By re-writing the above inequalities, we get.

$x_1 + t\Delta x \leq x_{max}$

$\therefore \boxed{t\Delta x \leq x_{max} - x_1}$ — ①.

and $x_1 + t\Delta x \geq x_{min}$

$\therefore \boxed{t\Delta x \geq x_{min} - x_1}$ $\therefore \boxed{-t\Delta x \leq x_1 - x_{min}}$ — ②

Similarly:- $\boxed{t\Delta y \leq y_{max} - y_1}$ — ③

and $\boxed{t\Delta y \geq y_{min} - y_1}$ $\therefore \boxed{-t\Delta y \leq y_1 - y_{min}}$ — ④

⑤ Comparing the above equation with $tp_k \le q_k$ where $k = 1, 2, 3, 4$, we get,

$\boxed{P_1 = \Delta x, \; P_2 = -\Delta x, \; P_3 = \Delta y, \; P_4 = -\Delta y}$

$\boxed{q_1 = x_{max} - x_1 \quad q_2 = x_1 - x_{min} \quad q_3 = y_{max} - y_1 \quad q_4 = y_1 - y_{min}}$

⑥ To get the final visible segment, we do the following calculations:

- If $P_k = 0$, then line is parallel to clipping region
- If $q_k < 0$, then line is completely outside the window
- If $P_k < 0$, $t_1 = \max\left(0, \frac{q_k}{P_k}\right)$
- If $P_k > 0$, $t_2 = \min\left(1, \frac{q_k}{P_k}\right)$

Now, if $t_1 > t_2 \longrightarrow$ discard.
if $t_1 < t_2$ then find,
$x = x_1 + t\Delta x$.
$y = y_1 + t\Delta y$

Problem: $A = (30, 60)$ & $B = (60, 25)$

Now, $\Delta x = x_2 - x_1 = 60 - 30 = 30$
$\Delta y = y_2 - y_1 = 25 - 60 = -35$

Now, $x_{min} = 10$ and $x_{max} = 50$
$y_{min} = 10$ $y_{max} = 50$.

Now finding P_k & q_k for $K = 1, 2, 3, 4$.

$P_1 = \Delta x = 30$ $\quad q_1 = x_{max} - x_1 = 50 - 30 = 20$
$P_2 = -\Delta x = -30$ $\quad q_2 = x_1 - x_{min} = 30 - 10 = 20$
$P_3 = \Delta y = -35$ $\quad q_3 = y_{max} - y_1 = 50 - 60 = -10$
$P_4 = -\Delta y = 35$ $\quad q_4 = y_1 - y_{min} = 60 - 10 = 50$

Now, $P_k < 0$ at $[P_2, P_3]$

$\therefore t_1 = \max\left(0, \frac{q_k}{P_k}\right)$

$= \max\left(0, \frac{q_2}{P_2}, \frac{q_3}{P_3}\right)$

$= \max\left(0, \frac{20}{-30}, \frac{-10}{-35}\right)$

$\therefore \boxed{t_1 = \frac{2}{7} \text{ or } 0.28}$

Now, $P_k > 0$ at $[P_1, P_4]$

$\therefore t_2 = \min\left(1, \frac{q_k}{p_k}\right)$

$= \min\left(1, \frac{q_1}{p_1}, \frac{q_4}{p_4}\right)$

$= \min\left(1, \frac{20}{30}, \frac{50}{35}\right)$

$\boxed{t_2 = 0.67 \text{ or } \frac{2}{3}}$

Now, $t_1 < t_2$

$\therefore x_1' = x_1 + t_1 \Delta x$
$= 30 + 0.28 \times 30$
$\boxed{x_1' = 38.4}$

$x_2' = x_1 + t_2 \Delta x$
$= 30 + \frac{2}{3} \times 30$
$\boxed{x_2' = 50}$

$y_1' = y_1 + t_1 \Delta y$
$= 60 + \frac{10}{35} - 35$
$\boxed{y_1' = 50}$

$y_2' = y_1 + t_2 \Delta y$
$y_2' = 60 + \frac{2}{3} \times (-35)$
$\boxed{y_2' = 36.67}$

$\therefore$ Visible line segment is (38.4, 50) to (50, 36.67)

$\therefore$ Original line segments: A (30, 60), B (60, 25)
Visible line segments: A' (38.4, 50) B' (50, 36.67)

Q5 Describe Properties of BEIZER CURVE.

Ans: ① They generally show the shape of control polygon
② They always pass through the first and last control point
③ They are contained in the convex hull of their defining control points.
④ It generally follows the shape of defining polygon
⑤ They are invarient under an affine transformation
⑥ The convex hull property for a Bezier curve ensure that the polynomial smoothly follows the control points
⑦ No straight line intersects a Bezier curve more times than it intersects its control polygon.
⑧ Beizer curve exibit global control means moving a control point alters the shape of the whole curve.
⑨ The direction of the tangent vector at the end points is same as that of the vector determined by the first and last segments.

Q6 Write a short note on Depth Buffer Algo:-

Ans It is also called Z-Buffer Algorithm.

Depth Buffer algorithm is simplest image space algorithm. Depth Buffer algorithm requires 2 arrays, intensity and depth each of which is indexed by pixel co-ordinates (x, y).

Algorithm:-

1. Calculate the depth z of the polygon at (x, y)
2. If z < depth [x, y], this polygon is closer to the observer than other already recorded for this pixel. In this case, set depth [x, y] to z and intensity [x, y] to a value corresponding to polygon's shading. If instead z > depth (x, y) the polygon already recorded at (x, y) lies closer to the observer than does this new polygon, and no action is taken.
3. After, all polygon have been processed; the intensity array will contain the solution.
4. The depth buffer algorithm illustrates several features common to all hidden surface algorithms
5. First it requires a representation of all opaque surface in scence polygon in this case.
6. These polygons may be faces of polyhedral recorded in the model of scene or may simply represent thin opaque 'sheets' in the scene.
7. The IInd important feature of the algorithm is its use of a ~~co~~ screen co-ordinate system. Before step 2, all polygons in the scene are transformed into a screen co-ordinate system using matrix multiplication.