Chapter 1-4 Resources

Resource Follow :

1. Follow my Note Chapter 1 - 3
2. DDA Algorithm | Line Drawing Algorithms | Gate Vidyalay
    a. (Part-3.3) DDA Algorithm (Numerical -1) | কম্পিউটার ...
3. Bresenham Line Drawing Algorithm | Gate Vidyalay
4. Bresenham Circle Drawing Algorithm | Gate Vidyalay
5. Mid Point Circle Drawing Algorithm | Gate Vidyalay

final

* **circle symmetry property** : circle looks the same when you rotate/reflect it.
    → same look from diff. angle/direction
    → because it has certain points / line that

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix} \qquad \begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

3.7 REGION FILLING

Region filling is the process of "coloring in" a definite image area or region. Regions may be defined at the pixel or geometric level. At the pixel level, we describe a region either in terms of the bounding

When eight-way symmetry is used to obtain a full circle from pixel coordinates generated for the 0° to 45° or the 90° to 45° octant, certain pixels are set or plotted twice. This phenomenon is sometimes referred to as overstrike. Identify where overstrike occurs.

SOLUTION

At locations resulted from the initial coordinates $(r, 0)$ or $(0, r)$ since $(0, r) = (-0, r)$, $(0, -r) = (-0, -r)$, $(r, 0) = (r, -0)$, and $(-r, 0) = (-r, -0)$.

In addition, if the last generated pixel is on the diagonal line at $(\alpha r, \alpha r)$ where α approximates $1/\sqrt{(2.0)} = 0.7071$, then overstrike also occurs at $(\alpha r, \alpha r)$, $(-\alpha r, \alpha r)$, $(\alpha r, -\alpha r)$, and $(-\alpha r, -\alpha r)$.

Overstrike: when 8-way symmetry is used to obtain a full-circle from pixel coordinates generated for the 0° to 45° on the 90° to 45° octant, certain pixels are set on plotted twice. This phenomenon is known as overstrike.

Final

* How to prevent it?

→ We need to ensure that each pixel is plotted only once.

1) Selective quadrant skipping: skip pixels that are equidistant from circle's center and fall on $x = 0$ or $y = 0$.

2) Update decision variable so that each pixel is considered only once.

3) Maintain symmetry across all octants of the circle to ensure consistent plotting.

Resource for chapter 4

- https://www.gatevidyalay.com/2d-transformation-in-computer-graphics-translation-examples/
- https://www.gatevidyalay.com/2d-rotation-in-computer-graphics-definition-examples/
- https://www.gatevidyalay.com/scaling-in-computer-graphics-definition-examples/
- https://www.gatevidyalay.com/2d-reflection-in-computer-graphics-definition-examples/
- https://www.gatevidyalay.com/2d-reflection-in-computer-graphics-definition-examples/

4.6 Find the transformation that scales (with respect to the origin) by (*a*) a units in the X direction, (*b*) b units in the Y direction, and (*c*) simultaneously a units in the X direction and b units in the Y direction.

SOLUTION

(*a*) The scaling transformation applied to a point $P(x, y)$ produces the point (ax, y). We can write this in matrix form as $S_{a,1} \cdot P$, or

$$\begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax \\ y \end{pmatrix}$$

(*b*) As in part (*a*), the required transformation can be written in matrix form as $S_{1,b} \cdot P$. So

$$\begin{pmatrix} 1 & 0 \\ 0 & b \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ by \end{pmatrix}$$

(*c*) Scaling in both directions is described by the transformation $x' = ax$ and $y' = by$. Writing this in matrix form as $S_{a,b} \cdot P$, we have

$$\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax \\ by \end{pmatrix}$$

rotation by 180°

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

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

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$M \cdot M^{-1} = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

4.7 Write the general form of a scaling matrix with respect to a fixed point $P(h, k)$.

SOLUTION

Following the same general procedure as in Probs. 4.3 and 4.4, we write the required transformation with

$\mathbf{v} = h\mathbf{I} + k\mathbf{J}$ as

$$\begin{aligned} S_{a,b,P} &= T_{\mathbf{v}} \cdot S_{a,b} \cdot T_{-\mathbf{v}} \\ &= \begin{pmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -h \\ 0 & 1 & -k \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} a & 0 & -ah + h \\ 0 & b & -bk + k \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

4.9 Describe the transformation M_L which reflects an object about a line L.

SOLUTION

Let line L in Fig. 4-15 have a y intercept $(0, b)$ and an angle of inclination θ° (with respect to the x axis). We reduce the description to known transformations:

1. Translate the intersection point B to the origin.
2. Rotate by $-\theta^\circ$ so that line L aligns with the x axis.
3. Mirror-reflect about the x axis.
4. Rotate back by θ°.
5. Translate B back to $(0, b)$.

In transformation notation, we have

$$M_L = T_{\mathbf{v}} \cdot R_\theta \cdot M_x \cdot R_{-\theta} \cdot T_{-\mathbf{v}}$$

4.12 The matrix $\begin{pmatrix} 1 & a \\ b & 1 \end{pmatrix}$ defines a transformation called a *simultaneous shearing* or *shearing* for short.

The special case when $b = 0$ is called *shearing in the x direction*. When $a = 0$, we have *shearing in the y direction*. Illustrate the effect of these shearing transformations on the square $A(0, 0)$, $B(1, 0)$, $C(1, 1)$, and $D(0, 1)$ when $a = 2$ and $b = 3$.

SOLUTION

Figure 4-16(a) shows the original square, Fig. 4-16(b) shows shearing in the x direction, Fig. 4-16(c) shows shearing in the y direction, and Fig. 4-16(d) shows shearing in both directions.

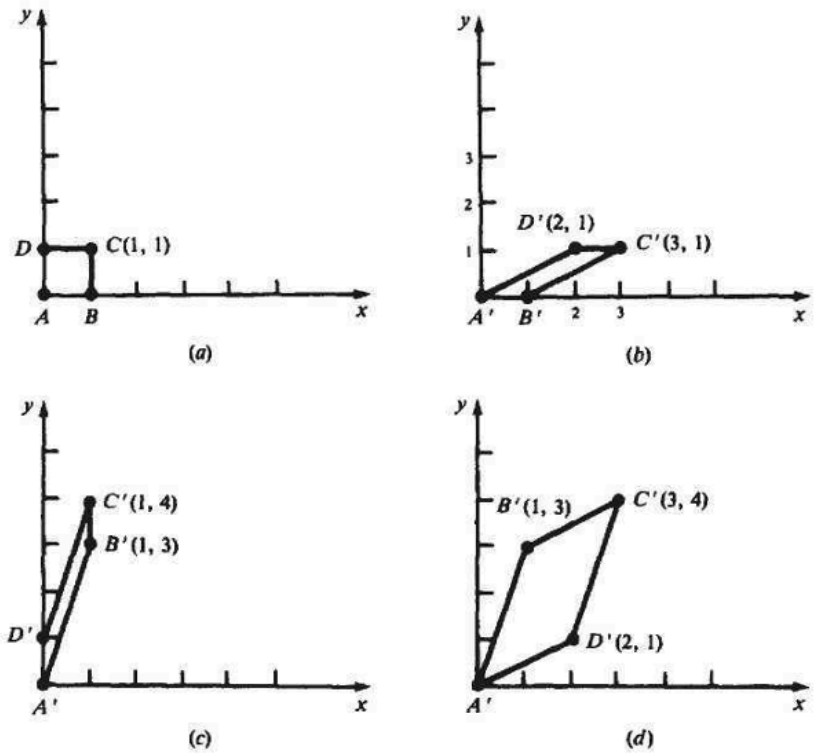


Fig. 4-16

Chapter 5 Theory and Math

Theoretical Question

What are the conditions for the clipping candidate in the Cohen-Sutherland algorithm?

Ans Conditions for the clipping candidate in the Cohen - Sutherland algorithm?

→ ① Both endpoints are inside the clipping window → accepted.

② Both outside → rejected

③ partially inside the window → needs further clipping.

Write the basic differences between Sutherland-Hodgeman and Weiler- Atherton Algorithm.

Sutherland-Hodgman	Weiler-Atherton:
Simple	More complex
Processes one edge at a time	Handles both convex and concave shapes
Best for rectangular clipping regions	Good for irregular clipping regions
Produces a single polygon	Can produce multiple polygons

Name two algorithms that can be used to find out the visible portion of given lines in a certain viewport.

Name 2 algorithms that can be used to find out the visible portion of given lines in a certain viewport:

① Cohen-Sutherland Line Clipping algo: uses a divide-and-conquer strategy to recursively divide the line and check if each segment lies inside the clipping window or needs further clipping.

② Liang-Barsky Line Clipping Algo: uses parametric line equations and inequalities to perform clipping more effeciently by reducing the number of intersection calculations required.

What do you understand by window to viewport mapping?

window to viewpont mapping is the process of transforming a specific part of a langen scene (the window) to fit within a display area (the viewpont). for example, if you've a map showing coordinates from (50, 50) to (150, 150) and you want to display it on a screen area from (0, 0) to (400, 400), you scale and translate the map coordinates to fit this screen area.

Sol: $S_x = \dfrac{V_{xmax} - V_{xmin}}{W_{xmax} - W_{xmin}}$, $S_y = \dfrac{V_{ymax} - V_{ymin}}{W_{ymax} - W_{ymin}}$

Express window-to-viewport mapping in the form of a composite transformation matrix.

① Translate the window to origin:

$$T_1 = \begin{bmatrix} 1 & 0 & -W_{xmin} \\ 0 & 1 & -W_{ymin} \\ 0 & 0 & 1 \end{bmatrix}$$

② Scale the window to fit the viewport.

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

③ Translate the scaled window to viewport.

$$T_2 = \begin{bmatrix} 1 & 0 & V_{xmin} \\ 0 & 1 & V_{ymin} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \boxed{T = T_2 \cdot S \cdot T_1}$$

Composite Transformation matrix

$$= \begin{bmatrix} S_x & 0 & V_{xmin} - S_x \cdot W_{xmin} \\ 0 & S_y & V_{ymin} - S_y \cdot W_{ymin} \\ 0 & 0 & 1 \end{bmatrix}$$

This matrix can be used to map any point in the window to its corresponding point in the viewport.

Q What's region code? What does each bit represent?

→ A region code is a 4-bit binary code used in the Cohen-Sutherland line clipping algorithm to indicate a point's position relative to a rectangular clipping window.

Each bit represents a direction.

TBRL

bit 3 → top	bit 1 → right
bit 2 → bottom	bit 0 → left

This code helps quickly determine whether a line segment is inside, outside or needs clipping.

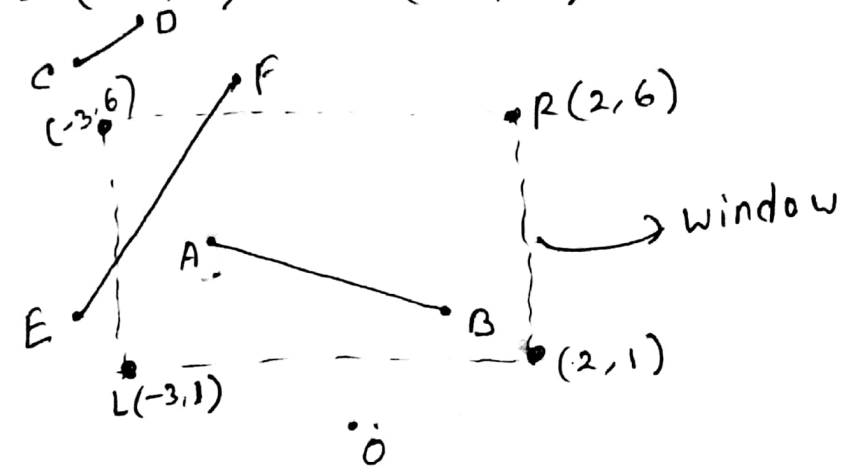
Last Year Ques Solve (math)

Page 1 :

3. (a) Let R be a rectangular window whose lower left-hand corner is at L (−3, 1) and upper right-hand corner is at R (2, 6)

(i) Find the region codes for the endpoints A (−2, 3) → B (1, 2), C (−4, 7) → D (−2, 10) and E (−4, 2) → F (−1, 7)

→



TBRL :

Region codes :

A → 0000 B → 0000

C → 1001 D → 1000

E → 0001 F → 1000

(ii) Find the clipping categories for the line segment in part - (i)

→ A and B both have code 0000

∴ completely inside

$C = 1001$
$D = 1000$
$\overline{\text{AND} = 1000}$

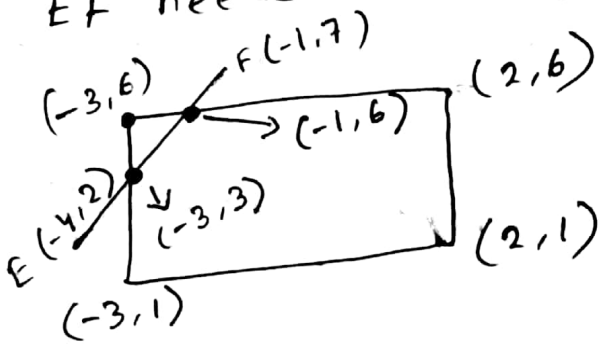
→ completely outside

$E = 0001$
$F = 1000$
$\overline{\text{AND} = 0000}$

→ partially ~~out~~ inside
→ requires clipping

(iii) Use the cohen-sutherland algo to clip the line segments in part-(i)

→ Only EF needs clipping (partially inside)



Clipping against left boundary $x = -3$

for E $(-4, 2)$; $m = \frac{7-2}{-1+4} = \frac{5}{3}$

$\therefore y = y_1 + m(x - x_1)$

$= 2 + \frac{5}{3}(-3+4)$

$= 2 + \frac{5}{3} = \frac{11}{3} \approx 3.67 = 3$

$\therefore$ intersection point $(-3, 3)$

Clipping against top boundary $y = 6$

for $(-1, 7)$; $m = \frac{5}{3}$
$x_1 \; y_1$

$\therefore \; y - y_1 = m(x - x_1)$

$\rightarrow \; x = \frac{y - y_1}{m} + x_1$

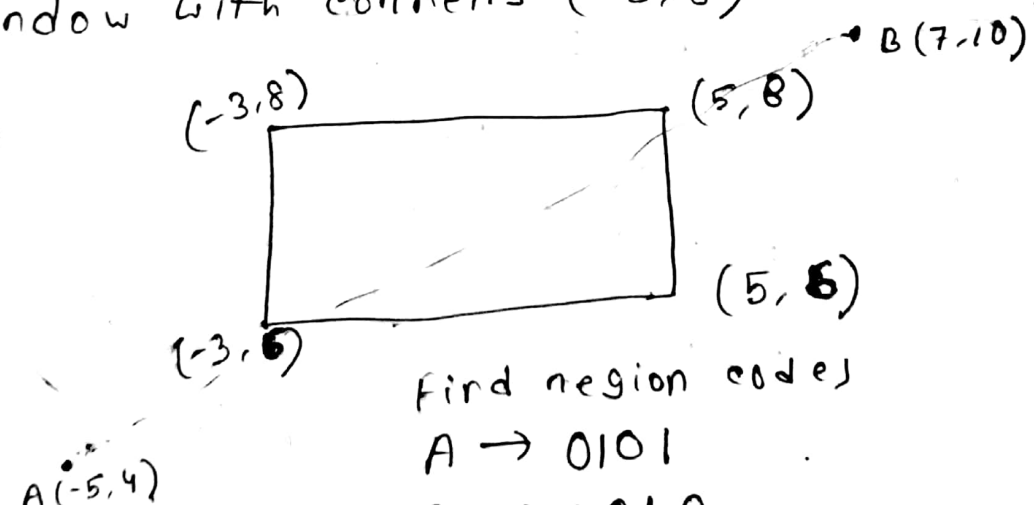
$= \frac{6 - 7}{5/3} + (-1)$

$= -\frac{3}{5} - 1 = \frac{-8}{5} \approx -1$

$\therefore$ intersection point $(-1, 6)$.

Page: 2 5 (a) ~~Fin~~ Apply cohen - sutherland line clipping to clip the line segment with endpoints $(-5, 4)$ and $(7, 10)$ against a window with corners $(-3, 8)$ and $(5, 6)$.

$\rightarrow$



Find region codes

A $\rightarrow$ 0101

B $\rightarrow$ 1010

AND $\rightarrow$ 0000 $\rightarrow$ partially inside

needs clipping; clipping against bottom boundary $y = 6$

for $(-5, 4)$; $m = \frac{10-4}{7+5} = \frac{3}{12} = \frac{1}{4}$

$x = \frac{y - y_1}{m} + x_1$

$= \frac{6-4}{1/4} - 5 = 8 - 5 = 3$

$\therefore$ intersection point $(3, 6)$

clipping against right boundary $x = 5$

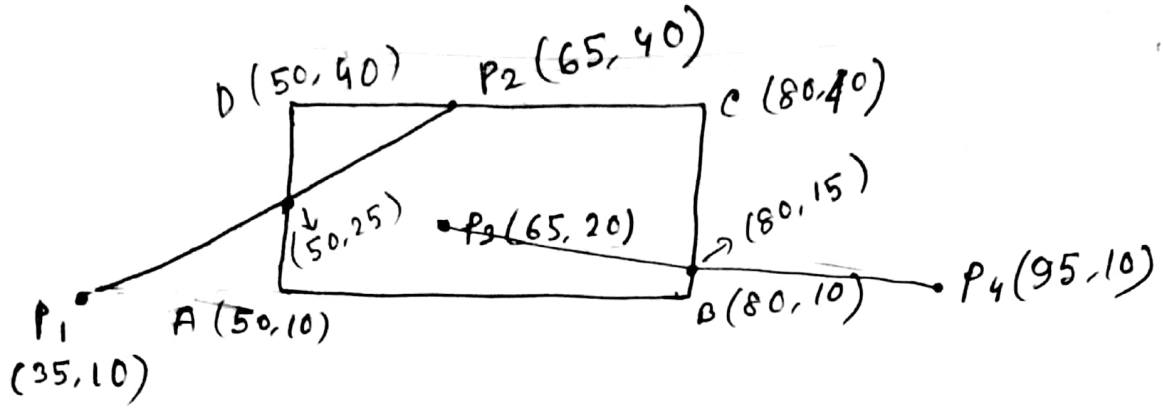
for $(7, 10)$ $m = \frac{1}{4}$

$y = m(x - x_1) + y_1 = \frac{1}{4}(5 - 7) + 10$

$= -1/2 + 10 = 9.5 \approx 9$

$\therefore$ intersection point $(5, 9)$

6. (a) Use the cohen-suthenland algo to clip two lines $P1(35, 10)$, $P2(55, 40)$ and $P3(65, 20)$, $P4(95, 10)$ against a window $A(50, 10)$, $B(80, 10)$, $C(80, 40)$ and $D(50, 40)$. Also find the clipping position.



<u>Region codes:</u> $P_1 : 0001$
$P_2 : 1000$ > AND = 0000 needs clipping (partially inside)

$P_3 : 0000$
$P_4 : 0110$ > AND = 0000

<u>$P_1 \rightarrow P_2$:</u> clipping against left boundary

$x = 50$

$\therefore$ for (35, 10): $m = \frac{40-10}{65-35} = 1$

$y = m(x - x_1) + y_1$

$= 1(50 - 35) + 10 = 15 + 10 = 25$

$\therefore$ intersection point (50, 25)

<u>$P_3 \rightarrow P_4$:</u> clipping against right boundary

$x = 80$;

for (95, 10)

$m = \frac{10-20}{95-65} = \frac{-10}{30} = -\frac{1}{3}$

$y = -\frac{1}{3}(80 - 95) + 10 = \frac{15}{3} + 10 = 15$

$\therefore$ intersection point (80, 15)

Page-4: 2.c (III) Clipping against rectangular windows whose sides are aligned with the x and y axes involves computing intersections with vertical and horizontal lines. Find the intersections of a line segment P_1P_2 with

a) the vertical line $x = a$ and

b) the horizontal line $y = b$

→ ~~Writing the equation of P_1P_2:~~

Let $P_1(x_1, y_1)$, $P_2(x_2, y_2)$

$$\therefore \quad m = \frac{y_2 - y_1}{x_2 - x_1}$$

(I) So, when clipping against $x = a$

for (x_1, y_1): $y = m(x - x_1) + y_1$

$$= \frac{y_2 - y_1}{x_2 - x_1}(a - x_1) + y_1$$

$\therefore$ intersection point $\left(a, \frac{(a - x_1)(y_2 - y_1)}{x_2 - x_1} + y_1\right)$

(II) clipping against, $y = b$

$$x = \frac{y - y_1}{m} + x_1 = \frac{b - y_1}{\frac{y_2 - y_1}{x_2 - x_1}} + x_1$$

$\therefore$ intersection point $\left(\frac{(b - y_1)(x_2 - x_1)}{y_2 - y_1} + x_1, b\right)$

5.b (ii) How can we determine whether a point $P(x, y)$ lies to the left or to the right of a line segment joining the points $A(x_1, y_1)$ and $B(x_2, y_2)$. Explain with necessary diagrams.

→ Steps:

A. (i) $\overrightarrow{AB} = (x_2 - x_1, y_2 - y_1)$

(ii) $\overrightarrow{AP} = (x - x_1, y - y_1)$

B. Cross product:

$$\overrightarrow{AB} \times \overrightarrow{AP} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ x_2 - x_1 & y_2 - y_1 & 0 \\ x - x_1 & y - y_1 & 0 \end{vmatrix}$$

$$= \hat{i}(0) - \hat{j}(0) + \hat{k}\{(y - y_1)(x_2 - x_1) - (y_2 - y_1)(x - x_1)\}$$

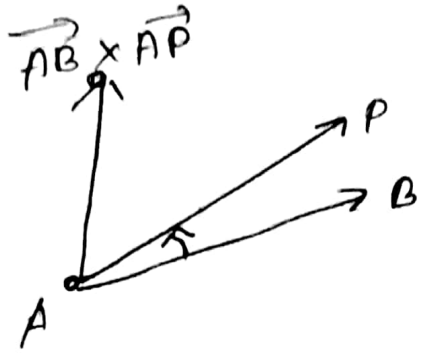
$$= (y - y_1)(x_2 - x_1) - (y_2 - y_1)(x - x_1)$$

C. if cross product is

(+) → P is left

(−) → P is right

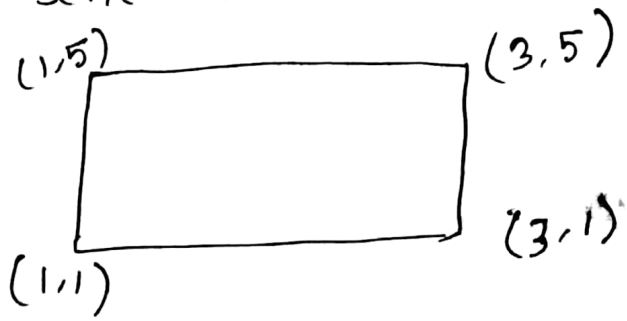
0 → P lies on the AB (same line)



AB to AP
→ Anticlockwise
→ positive
→ P left

4. a (iii) Find the normalization ~~techn~~ transformation that maps a window whose lower left corner is at (1,1) and upper right corner is at (3,5) onto -

a) a viewport that is the entire normalized device screen.



For normalized device screen.

$$(x_{vmin}, y_{vmin}) = (0,0)$$
$$(x_{vmax}, y_{vmax}) = (1,1)$$

$\boxed{\text{Scaling factor}}$ ∴

$$S_x = \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} = \frac{1-0}{3-1} = \frac{1}{2}$$

$$S_y = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}} = \frac{1-0}{5-1} = \frac{1}{4}$$

$\boxed{\text{Translation factor}}$

$$T_x = x_{vmin} - S_x \cdot x_{wmin}$$
$$= 0 - \frac{1}{2} \cdot 1 = -\frac{1}{2}$$

$$T_y = y_{vmin} - S_y \cdot y_{wmin}$$
$$= 0 - \frac{1}{4} \cdot 1 = -\frac{1}{4}$$

$\therefore$ Normalization Transformation:

$$x' = S_x \cdot x + T_x$$
$$= \frac{1}{2} x + \left(-\frac{1}{2}\right)$$

$$y' = S_y \cdot y + T_y$$
$$= \frac{1}{4} \cdot y - \frac{1}{4}$$

b) a viewport that has lower left corner at (0, 0) and upper right corner at (1/2, 1/2).

$$(x_{vmin}, y_{vmin}) = (0, 0)$$
$$(x_{vmax}, y_{vmax}) = \left(\tfrac{1}{2}, \tfrac{1}{2}\right)$$

$$\therefore S_x = \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} = \frac{1/2 - 0}{3 - 1} = \frac{1}{4}$$

$$S_y = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}} = \frac{1/2 - 0}{5 - 1} = \frac{1}{8}$$

$T_x = x_{vmin} - S_x \cdot x_{wmin} = 0 - \frac{1}{4} \cdot 1 = -\frac{1}{4}$

$T_y = y_{vmin} - S_y \cdot y_{wmin} = 0 - \frac{1}{8} \cdot 1 = -\frac{1}{8}$

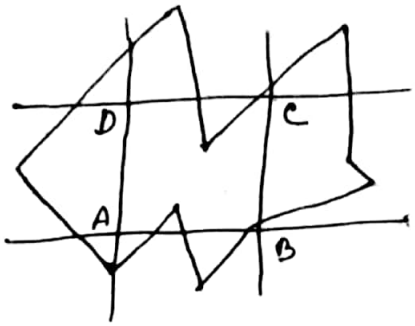
$\therefore$ Normalization Transformation:

$x' = S_x \cdot x + T_x = \frac{1}{4} x - \frac{1}{4}$

$y' = S_y \cdot y + T_y = \frac{1}{8} x - \frac{1}{8}$

Page: 2: 5 (a) same as 4.a.iii.(a)

Page: 5: 4.c (1)



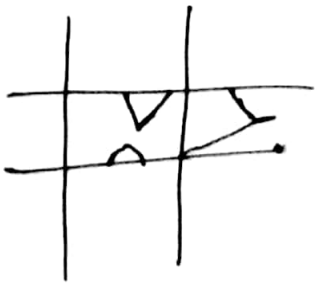
Clip the above polygon using Sutherland-Hodgeman algo. show the fig. at each step.

Ans:

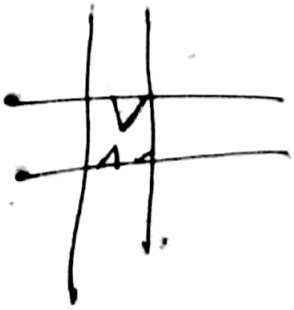
Step 1: clipping against A-B:

Step: 2: A-D

Step-3: D-C



Step: 4: B-C



Extra Theory and Math

Chapter 6

Tilting is defined as a rotation about the x-axis followed by a rotation about the y-axis: (i) find the tilting matrix; (ii) does the order of performing rotation matter? Show mathematically.

6.1) Tilting is defined as a notation about the
p-128 x-axis followed by y-axis.

① find the tilting matrix.

② Does the order of performing notation matter?

→ ⓐ $T = \cancel{R_x(\theta)} \cdot R_y(\phi) \cdot R_x\theta$

(order)

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

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

ⓑ To determine, If order matters, we need to consider the opposite order.

$$T' = R_x\theta \cdot R_y\phi$$

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

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

$\therefore T' \neq T$

Yes does matter

Chapter 8

What do you require for 3D viewing an object? Mathematically how do you specify the view plane?

18 Final

* What do you require for 3D viewing an object? Mathematically how do you specify the view plane?

→ Requirements for 3D Viewing:

1) 3D object: A model defined by vertices, edges and surfaces.

2) Camera setup:

a) Position: where the camera is located.

b) Orientation: The direction the camera is pointing.

c) Field of View (FOV): The extent of the scene visible to the camera.

d) Clipping Planes: Near and far distances where objects are visible.

Specifying the view plane: The view plane is the surface where the 3D scene is projected to form a 2D image. Here's how to specify it:

1) Camera position (Eye): $E = (E_x, E_y, E_z)$

2) Look-at Point $L = (L_x, L_y, L_z)$

3) Up vector : $U = (U_x, U_y, U_z)$

4) Compute the camera Basis vectors:

* View Direction D : $D = L - E$

Normalize D to get the unit vector d

$$d = \frac{D}{|D|} = \left(\frac{D_x}{|D|}, \frac{D_y}{|D|}, \frac{D_z}{|D|}\right)$$

* Right vector R :

$$R = U \times d$$

Normalize R to get the unit vector n

$$n = \frac{R}{|R|}$$

* True up vector U': $U' = d \times n$

unit vector $u' = \frac{U'}{|U'|}$

* View plane center c :

$$c = E + \text{distance} \cdot d$$

distance from camera to the view plane

* View plane dimension:

① Height: $h = 2 \cdot d \cdot \tan\left(\frac{FOV_y}{2}\right)$

② width : $w = h \cdot$ aspect ratio

Transforming to the view plane :

To convert world coordinates to camera coordinates, use the view transformation matrix :

$$M_{view} = \begin{bmatrix} r_x & r_y & r_z & -r \cdot E \\ u_x' & u_y' & u_z' & -u' \cdot E \\ d_x & d_y & d_z & -d \cdot E \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$r, u', d \rightarrow$ camera's basis vector

$E \rightarrow$ camera position

Ideally the view volume is infinite but in reality we prefer to use a finite volume, why?

Ideally view volume is infinite but in reality we prefer to use a finite volume. Why

→ It's computationally manageable, focusing resources on what's visible. The clipping process is efficient and straightforward.

volume?

How do you specify the finite view volume?

How do you specify the finite view volume?

→ you set boundaries that describe how far and wide the camera can see. This include deciding how close and far objects appear, and the shape of what's visible, whether like a box or a pyramid.

Chapter 9

What do you understand by polygon and polylines? Explain their differences.

Polygon vs polylines

Polygon	Polyline
① Always a closed shape	① can be either open on closed
② Ex: Triangle, square	② Ex: zigzag line, a walking path.
③ used to represent bounded areas, formed by a finite no. of line segments connected end-to-end.	③ used to represet paths, routes that don't necessarily form a closed shape.

What are the geometric forms used for modeling of objects?

① Polygons (Triangles, Quads): Used for mesh models.

② Curves and Surfaces: for smooth surfaces.

③ Solid Geometry: basic shapes like cubes, spheres and cylinders.

What is histogram equalization in image processing?

18 Final

What's "Histogram Equalization" in image processing?

→ a technique that adjusts the contrast of an image by redistributing the pixel values in its histogram.

Pros: simplicity and effectiveness in improving the contrast of images.

Cons: Artificial noise into the image. tendency to amplify noise.

What is analog image?

18 Final

What's analog image?

→ a continuous visual signal, not divided into pixels. Typically captured by analog devices like film cameras or CRT televisions, where light is captured smoothly on film, resulting in detailed and seamless images.

How do you model an object in computer graphics?

Q How do you model an object in CG?

→ In CG, objects are typically modeled using geometric primitives like points, lines and polygons. The most common method is using meshes made up of triangles or other polygons that define the object's surface.

What are the geometric forms used for modeling of objects?

Q What are the geometric forms used for modeling of objects?

1. Polygons (Triangles, Quads): Used for mesh models.
2. Curves and Surfaces: for smooth surfaces.
3. Solid Geometry: basic shapes like cubes, spheres and cylinders.

Page: 1

2. (a) Under the standard perspective trans. formation Perk, what's the projected image of a point in the place $z = -d$? what does this anomaly called?

→ We have a 3D point (x, y, z) which we want to project onto a 2D image.

$$\therefore x' = \frac{x \cdot d}{z+d} = \frac{x \cdot d}{-d+d} = \frac{x \cdot d}{0}$$

$$y' = \frac{y \cdot d}{z+d} = \frac{y \cdot d}{-d+d} = \frac{y \cdot d}{0}$$

Here, $z + d$ → distance from the point to the observer

So, the point would be infinitely far away on the image. It doesn't appear on the 2D image as a visible point.

This is called the vanishing plane anomaly.

CS CamScanner

page: 2

(b) Given points $P_1(1,2,0)$, $P_2(3,6,20)$ and $P_3(2,4,6)$ and a viewpoint $C(0,0,-10)$ determine which points obscure the others when viewed from C.

$\rightarrow \overrightarrow{CP_1} = (1-0, 2-0, 0+10) = (1,2,10)$

Distance $= \sqrt{(1-0)^2+(2-0)^2+(0+10)^2} = 10.25$

$\overrightarrow{CP_2} = (3-0, 6-0, 20+10) = (3,6,30)$

Distance $\sqrt{(3-0)^2+(6-0)^2+(20+10)^2} = 30.74$

$\overrightarrow{CP_3} = (2-0, 4-0, 6+10) = (2,4,16)$

Distance $= \sqrt{(2-0)^2+(4-0)^2+(6+10)^2} = 16.6$

As P_1 is closest to C, it'll be obscured by any other points. NO

P_3 is farther than P_1, it could be obscured by P_1, if they are on the same line.

P_2 can be obscured by both P_1 and P_3.

CamScanner

6 (b) find a transformation Av which aligns a given vector V with the vector K along the positive z axis.

→ Given vector $\vec{V}(V_x, V_y, V_z)$

Target vector $\vec{K}(0, 0, 1)$

∴ rotation axis $= \vec{V} \times \vec{K}$

$$= \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ V_x & V_y & V_z \\ 0 & 0 & 1 \end{vmatrix} = \hat{i} \cdot V_y - \hat{j} \cdot V_x + \hat{k} \cdot 0$$

∴ rotation axis $(V_y, -V_x, 0)$

rotation angle:

$$\cos\theta = \frac{\vec{V} \cdot \vec{K}}{|\vec{V}||\vec{K}|} = \frac{0 + 0 + V_z}{\sqrt{V_x^2 + V_y^2 + V_z^2} \cdot \sqrt{0^2 + 0^2 + 1^2}}$$

$$\therefore \theta = \cos^{-1}\left(\frac{V_z}{\sqrt{V_x^2 + V_y^2 + V_z^2}}\right)$$

rotation matrix:

$$R = I + \sin\theta \cdot K + (1 - \cos\theta) K^2$$

Transformation matrix:

$$V_T = R \cdot V$$

6 (c) Under the standard perspective transformation P_{erk}, what is the projected image of the line segment joining $P_1(-1,1,-2d)$ to $P_2(2,2,0)$. Use suitable figures

→ Given point, $P(x,y,z)$ in 3D space
It's projection P' onto the 2D image plane at distance d from the viewer:

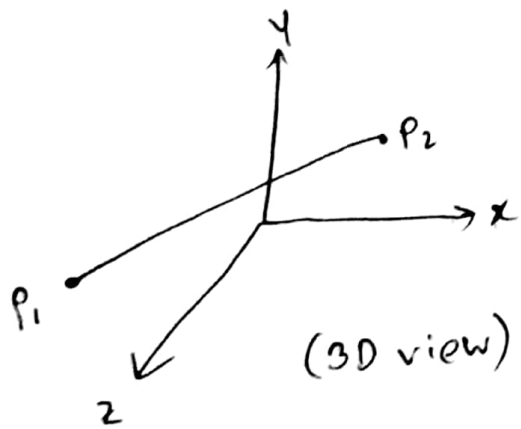
$$P'\left(\frac{x \cdot d}{z+d}, \frac{y \cdot d}{z+d}\right)$$

For $P_1(-1,1,-2d)$:

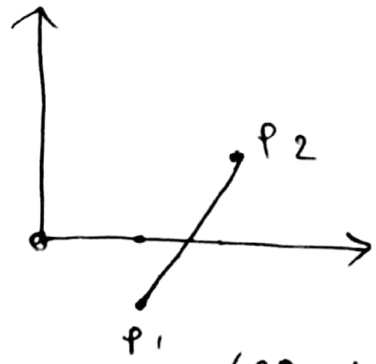
$$P_1'\left(\frac{-d}{-2d+d}, \frac{d}{-2d+d}\right) = (1,-1)$$

For $P_2(2,2,0)$

$$P_2'\left(\frac{2d}{0+d}, \frac{2d}{0+d}\right) = (2,2)$$



(3D view)



(2D view)