Computer Graphics

MD RAKIBUL HAQUE

LECTURER, CSE, RUET.

Object Space vs Image Space

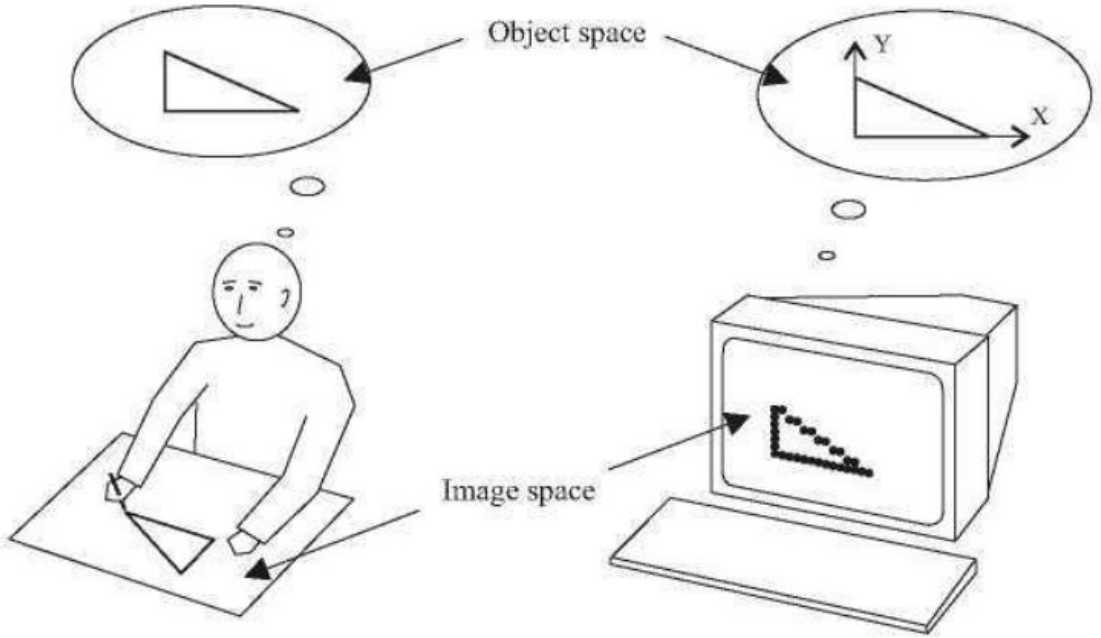


Fig. 1. Drawing A Triangle

Basic Computer Graphics Pipeline

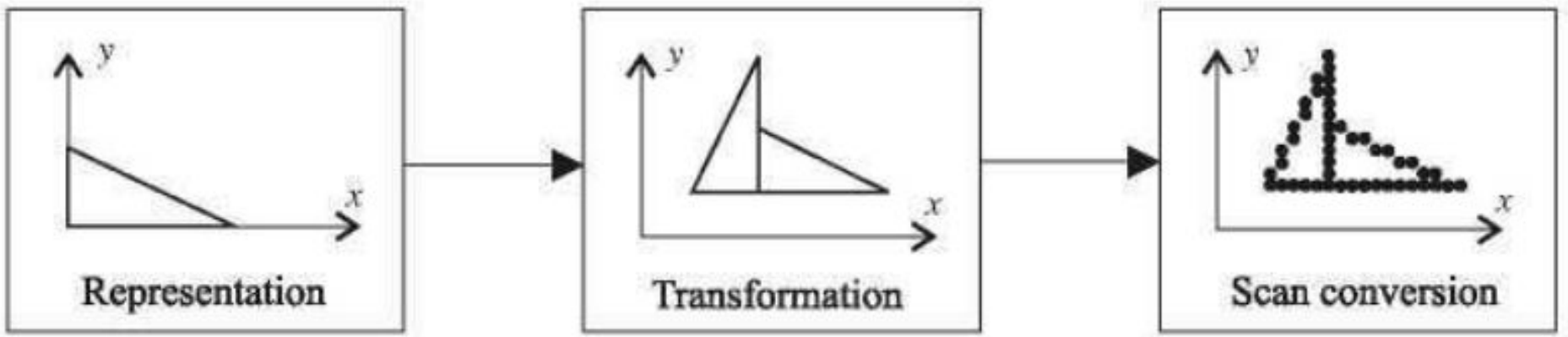


Fig. 2. A Simple Graphics Pipeline

Computer Graphics Vs Image Processing

Fig. 3. Computer Graphics Vs Image Processing

Image Representation

- Digital Image
  - Set of Discreet Pixels arranged in row column fashion to form a rectangular picture area known as a **raster**.

- Resolution
  The number of pixel per unit length.

- Aspect Ratio
  Image Width/ Image Height [Unit Can be Inch or Pixels]

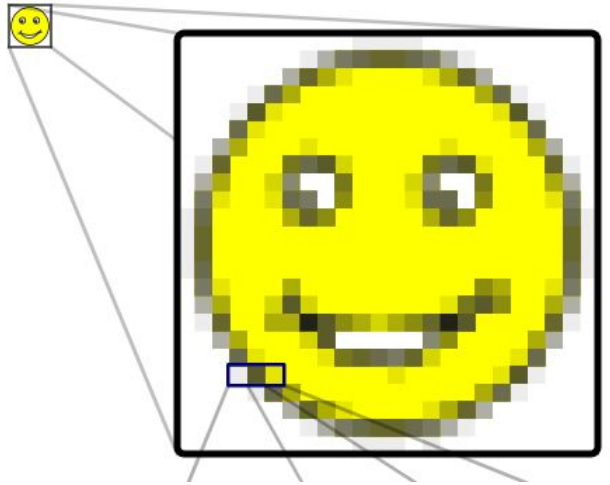


Figure 4. A raster image

Image Coordinate

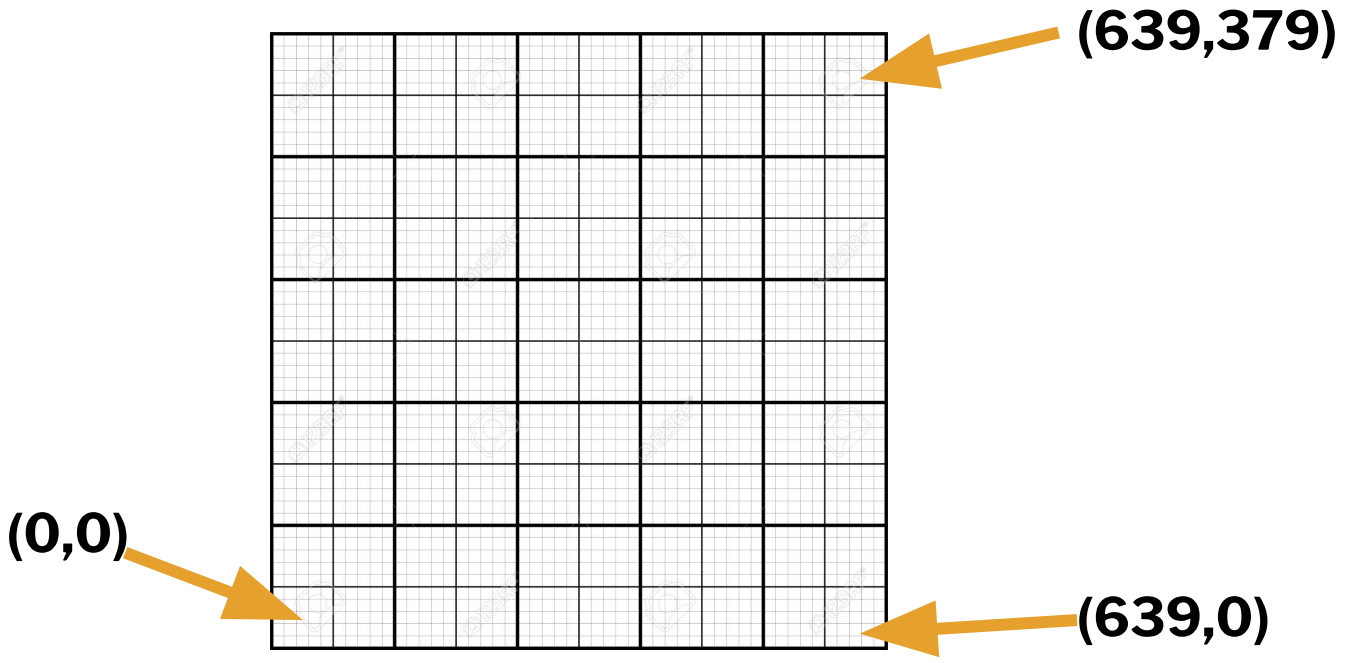
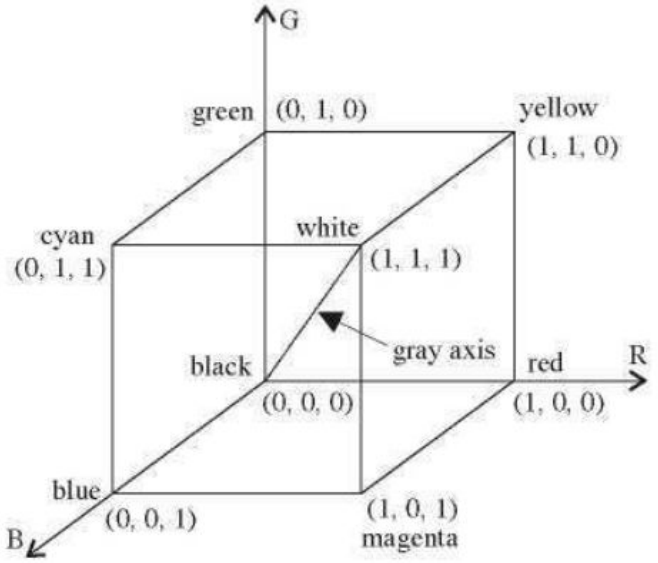


Fig. 4. A 640 x 380 Pixel Image

RGB COLOR SPACE



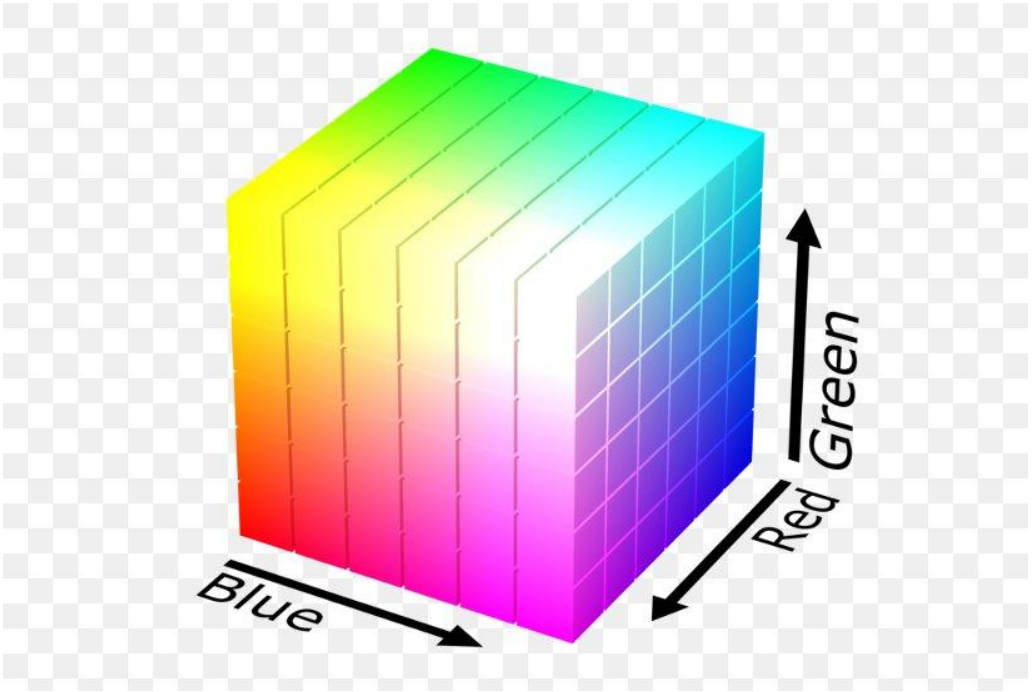
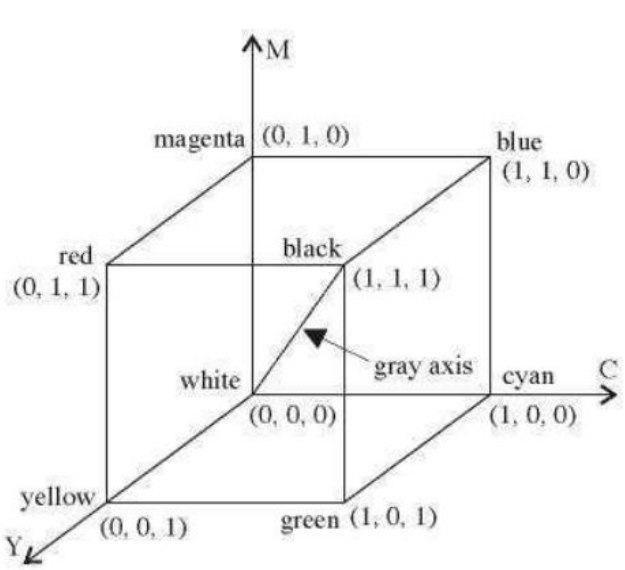


Fig 5. A RGB COLOR SPACE

CYM COLOR MODEL



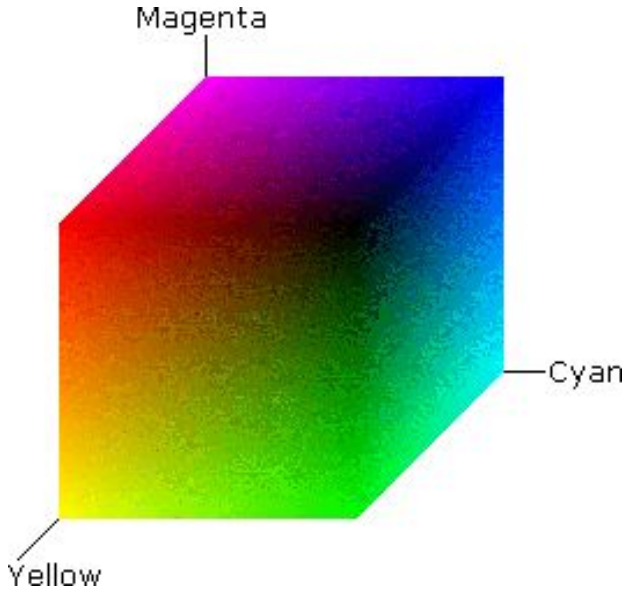


Fig 6. A CYM COLOR Model

RGB ⟺ CYM

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

DIRECT CODING

bit 1: r	bit 2: g	bit 3: b	color name
0	0	0	black
0	0	1	blue
0	1	0	green
0	1	1	cyan
1	0	0	red
1	0	1	magenta
1	1	0	yellow
1	1	1	white

Fig 7. Direct Coding of color using 3 bits

DIRECT CODING

- 24-Bit or 3 Bytes Format or True Color Image

  Red ---- 8 Bit ----- 256 Level
  Green -- 8 Bit ----- 256 Level
  Blue ---- 8 Bit ----- 256 Level

  A pixel will have a 256*256*255 choices or 16.7 Million Choices.

- 1000 x 1000 Pixels will take 3 million bits.

LOOKUP TABLE

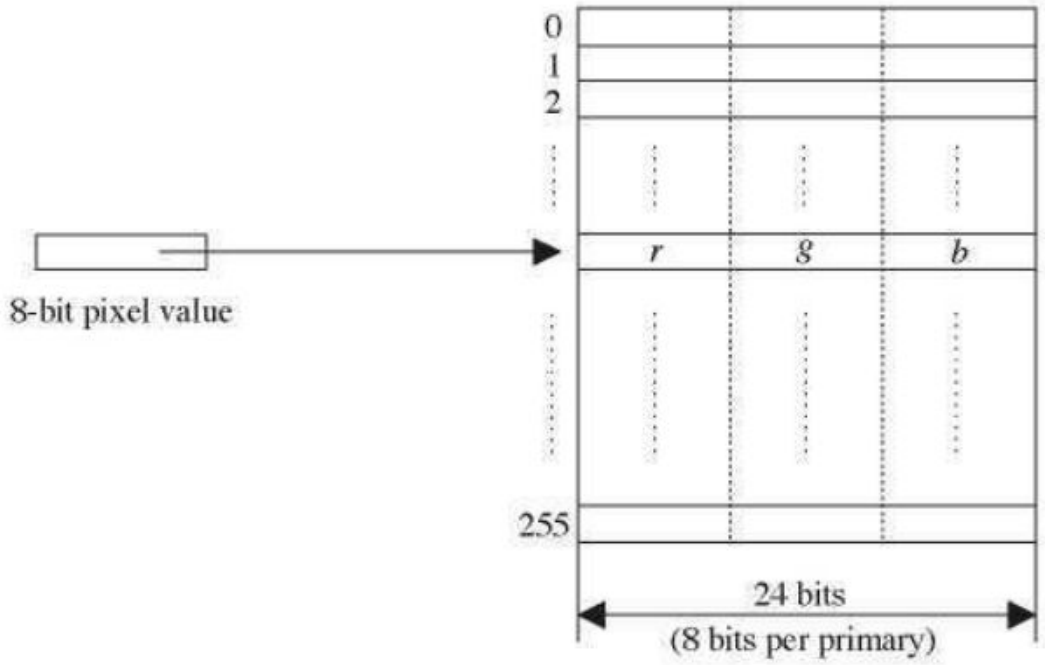


Fig 7. A 24 bit 256-Entry LookUp TABLE

Raster Image Vs Vector Image

CRT MONITOR

Fluorescence
Phosphorescence

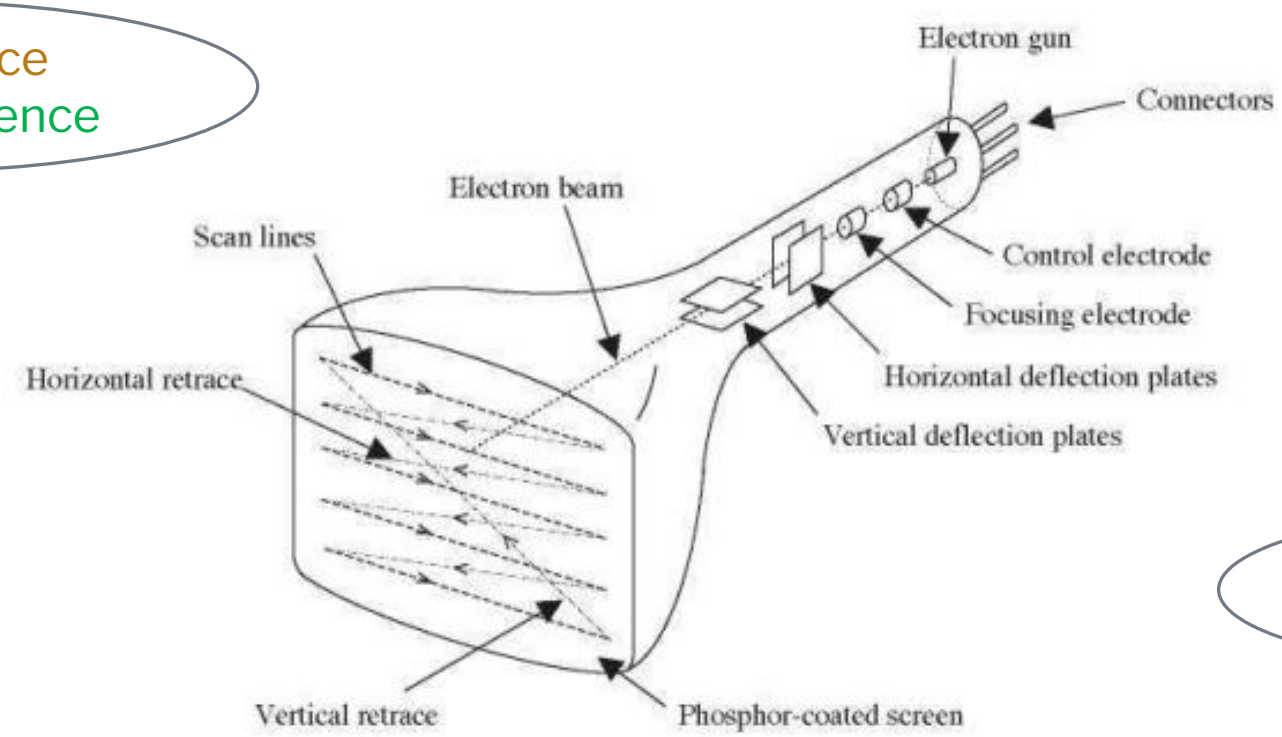


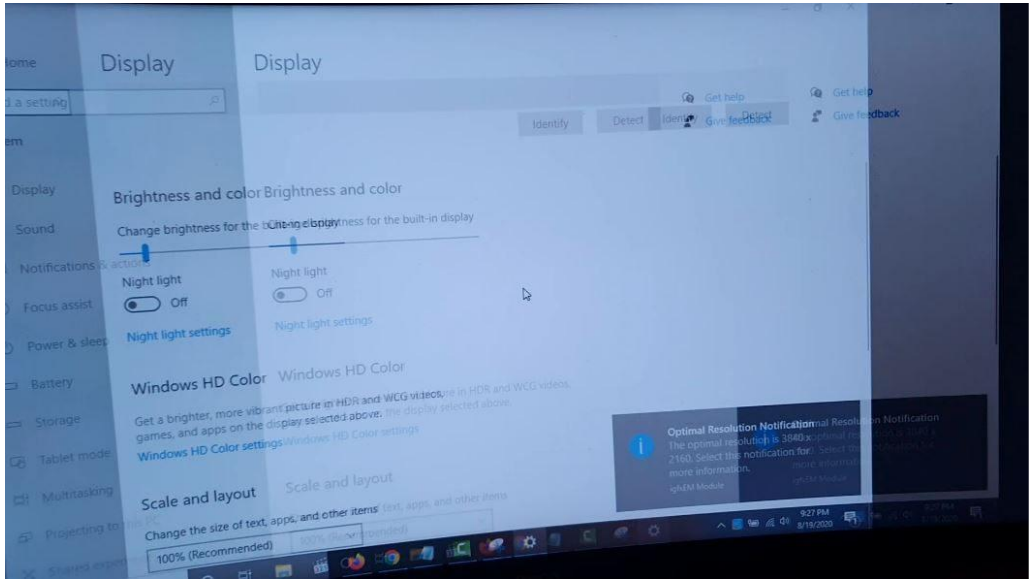
Fig. 8. Monochromatic CRT Tube

Refresh Rate
Interlacing

CRT MONITOR

SCAN LINES

FLICKERING



CRT MONITOR

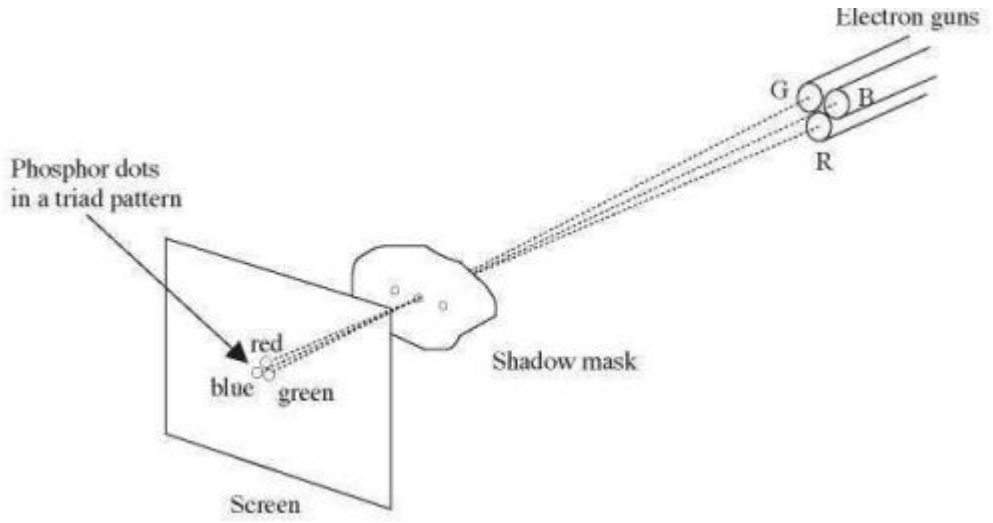


Fig. 8. Color CRT Tube

CRT MONITOR

Maths

- Compute the size of a 640 x 480 image at 240 pixels per inch.
- Compute the resolution of a 2 x 2 inch image that has 512 x 512 pixels.
- If we want to resize a 1024 x 768 image to one that is 640 pixels wide with the same aspect ratio, what would be the height of the resized image?
- Sometimes the pixel at the upper left corner of an image is considered to be at the origin of the pixel coordinate system (a left-handed system). How to convert the coordinates of a pixel at (x, y) in this coordinate system into its coordinates (x',y') in the lower-left-corner-as-origin coordinate system (a right-handed system)?

Maths

- The direct coding method is flexible in that it allows the allocation of a different number of bits to each primary color. If we use 5 bits each for red and blue and 6 bits for green for a total of 16 bits per pixel, how many possible simultaneous colors do we have?
- If we use 2-byte pixel values in a 24-bit lookup table representation, how many bytes does the lookup table occupy?

References

- Chapter 1
- Chapter 2

Geometric Transformation

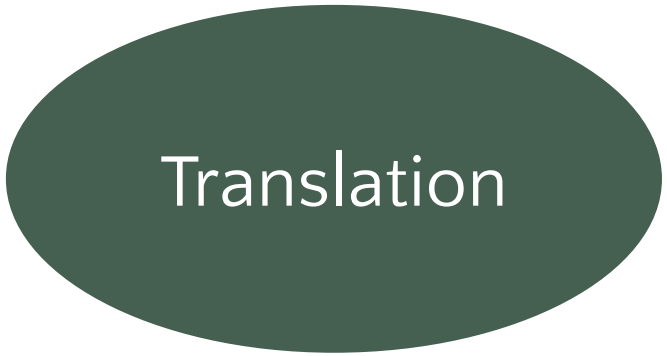
Md Rakibul Haque

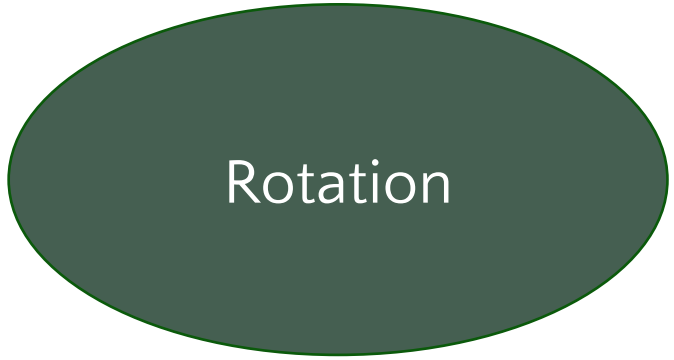
Lecturer, CSE , RUET.

Geometric Transformation vs Coordinate Transformation

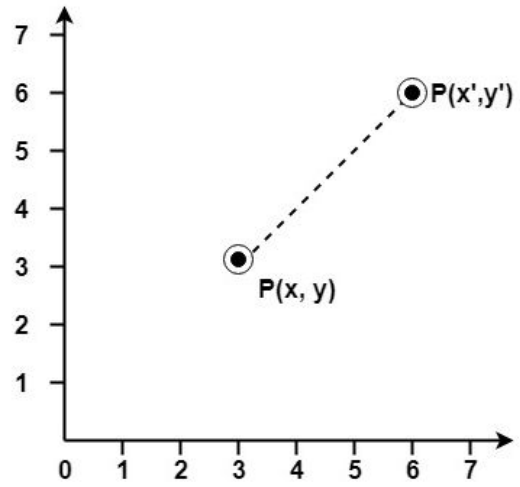
Geometric Transformation	Coordinate Transformation
The objects moves while the co-ordinate systems remain stationary.	The Co-ordinate system is moved while the object remains fixed

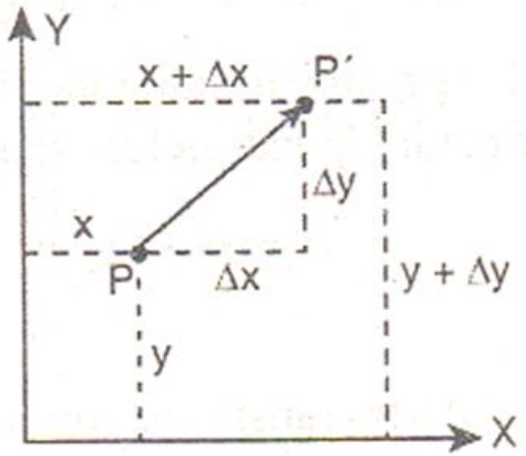
Geometric Transformation





Translation

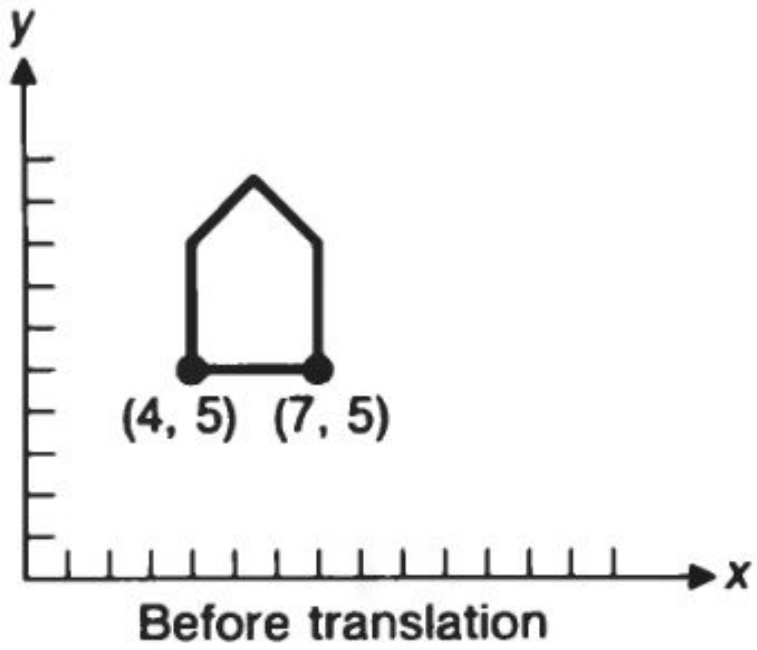


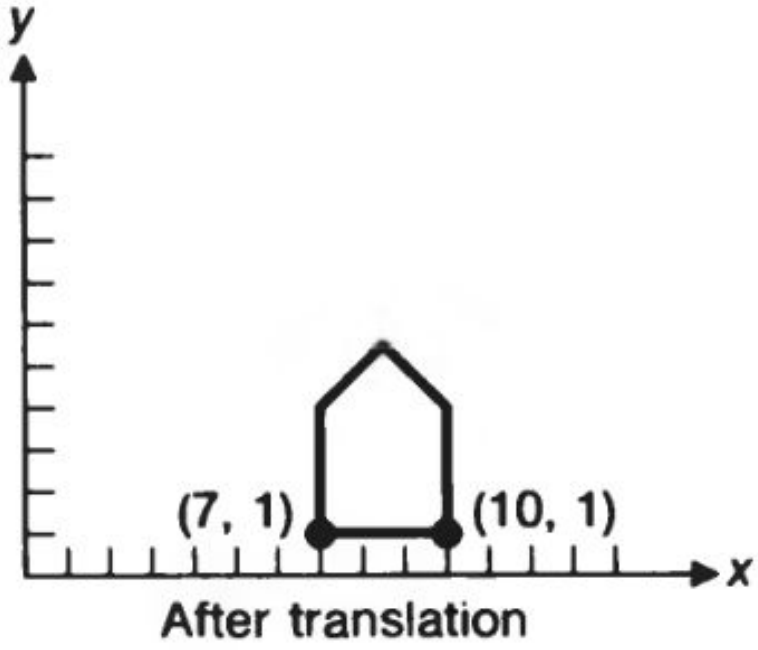


$$P = \begin{bmatrix} x \\ y \end{bmatrix}, P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, T = \begin{bmatrix} d_x \\ d_y \end{bmatrix},$$

$$P' = P + T.$$

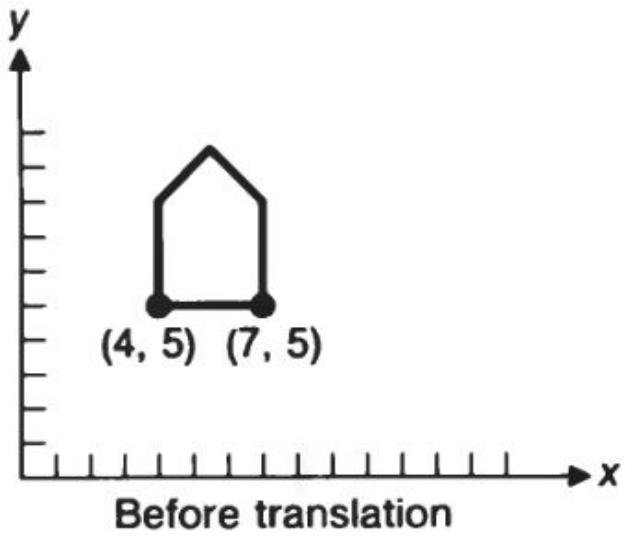
Translation Example

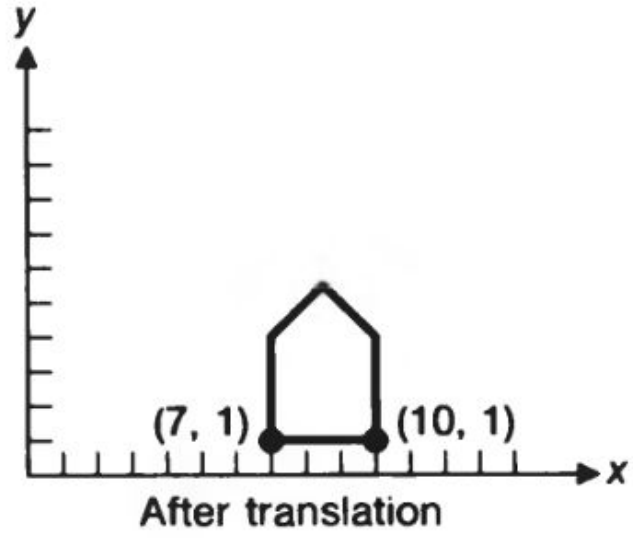




Translating a object by (3,–4)

Translation Properties



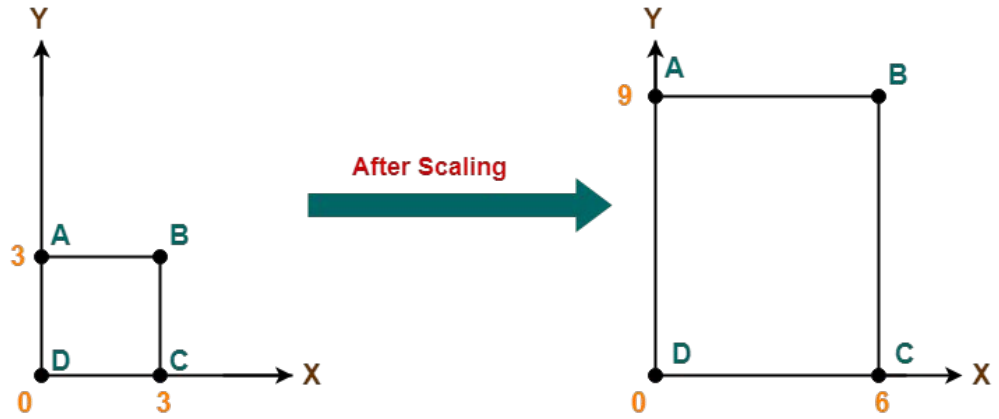


- No shape changes
- Shifting

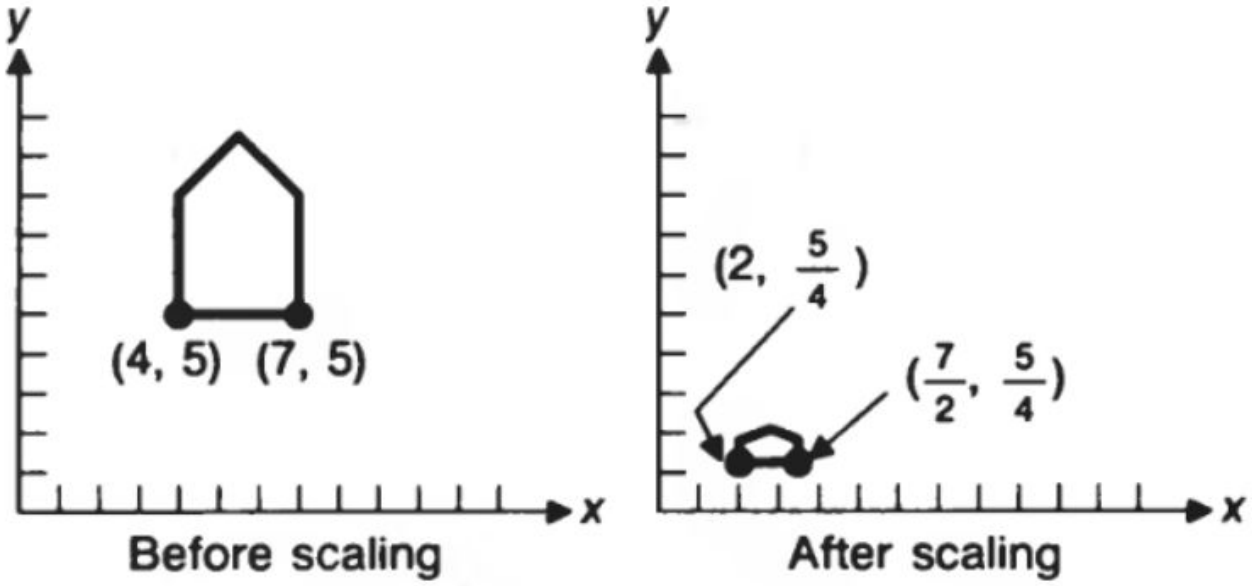
Scaling

$$x' = s_x \cdot x, \qquad y' = s_y \cdot y.$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \text{ or } P' = S \cdot P,$$



Scaling Example



Scaled by ½ in X and ¼ in Y

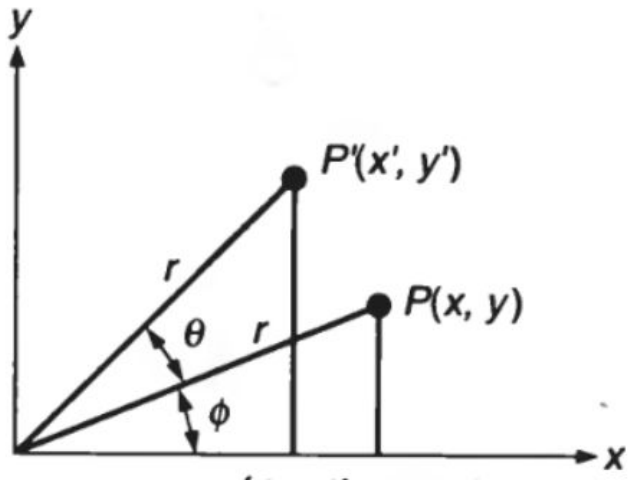
Scaling Types

- Differential Scaling, when $S_x \neq S_y$
- Uniform Scaling, when $S_x = S_y$
  - *If $S_i > 1$, then zooming, where i=x,y*
  - *If $S_i < 1$, then shrinking, where i= x,y*
  - *If $S_i = 1$, then same, where i= x,y*

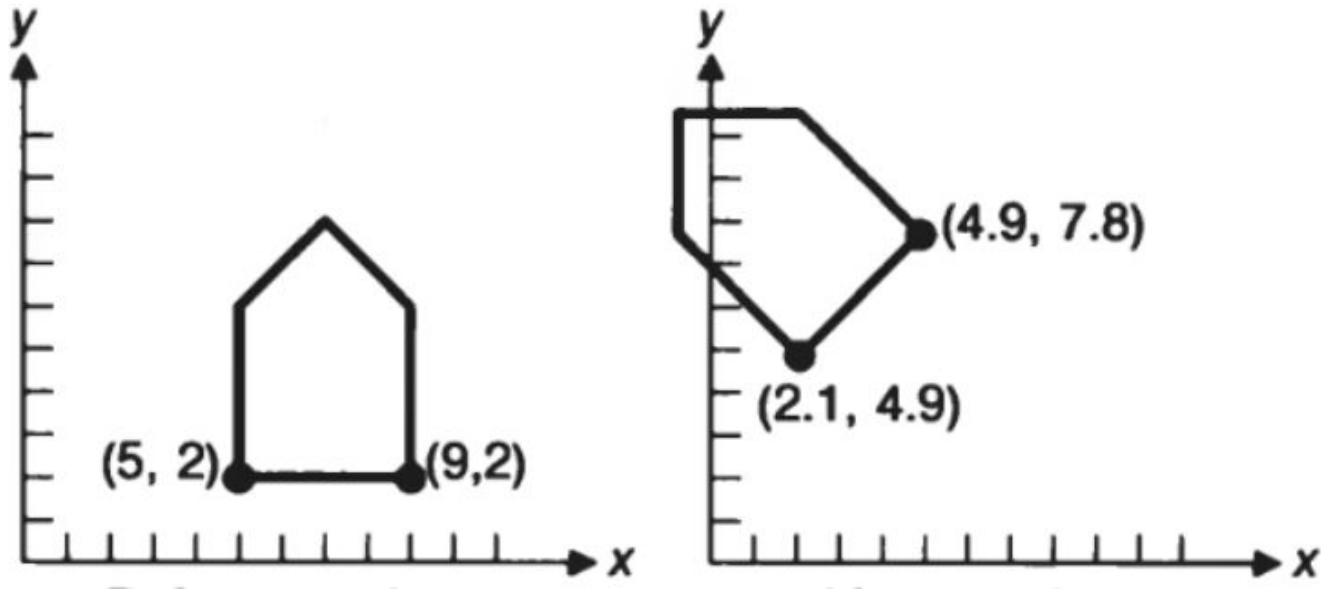
Rotation

$$x' = x \cdot \cos\theta - y \cdot \sin\theta, \quad y' = x \cdot \sin\theta + y \cdot \cos\theta.$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \text{ or } P' = R \cdot P,$$



Rotation Example



Rotation by 45 Degree

Reflection

Special type of Scaling

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

Reflection Matrix

(Reflection Along X Axis)

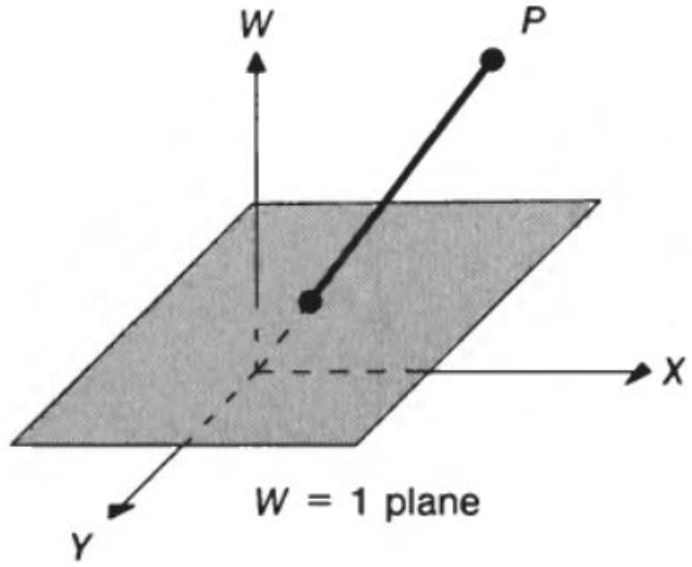
$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

Reflection Matrix

(Reflection Along Y Axis)

Homogeneous Coordinates

$$P' = T + P,$$
$$P' = S \cdot P,$$
$$P' = R \cdot P.$$



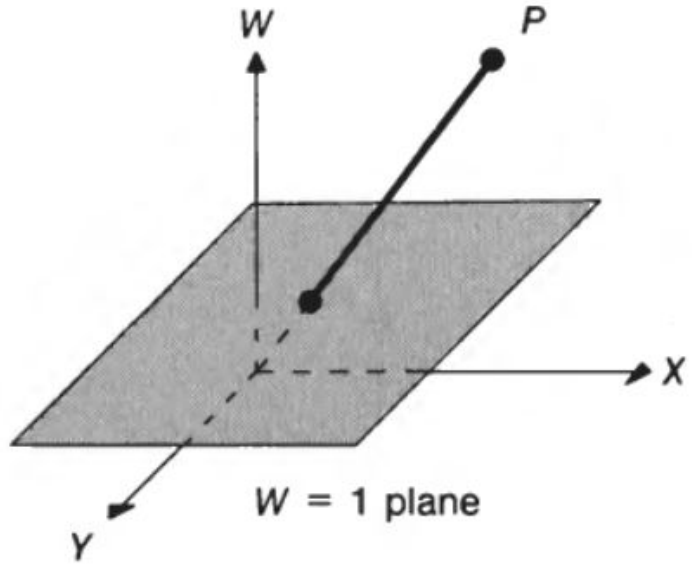
(x, y) ⟶ (x', y', w)

- (2,3,6) and (4,6,12) are the same in homogeneous coordinates.

(x', y', w) ⟶ (x'/w, y'/w, 1) ⟶ (x'/w, y'/w)

Homogeneous Coordinates

$$P' = T + P,$$
$$P' = S \cdot P,$$
$$P' = R \cdot P.$$



(x, y) ⟶ (x, y, w)

- (2,3,6) and (4,6,12) are the same in homogeneous coordinates.

(x, y, w) ⟶ (x/w, y/w, 1) ⟶ (x/w, y/w)

Translation Scaling and Rotation in Homogeneous Coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

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

Double Translation

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

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

Double Translation

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

Double Scaling

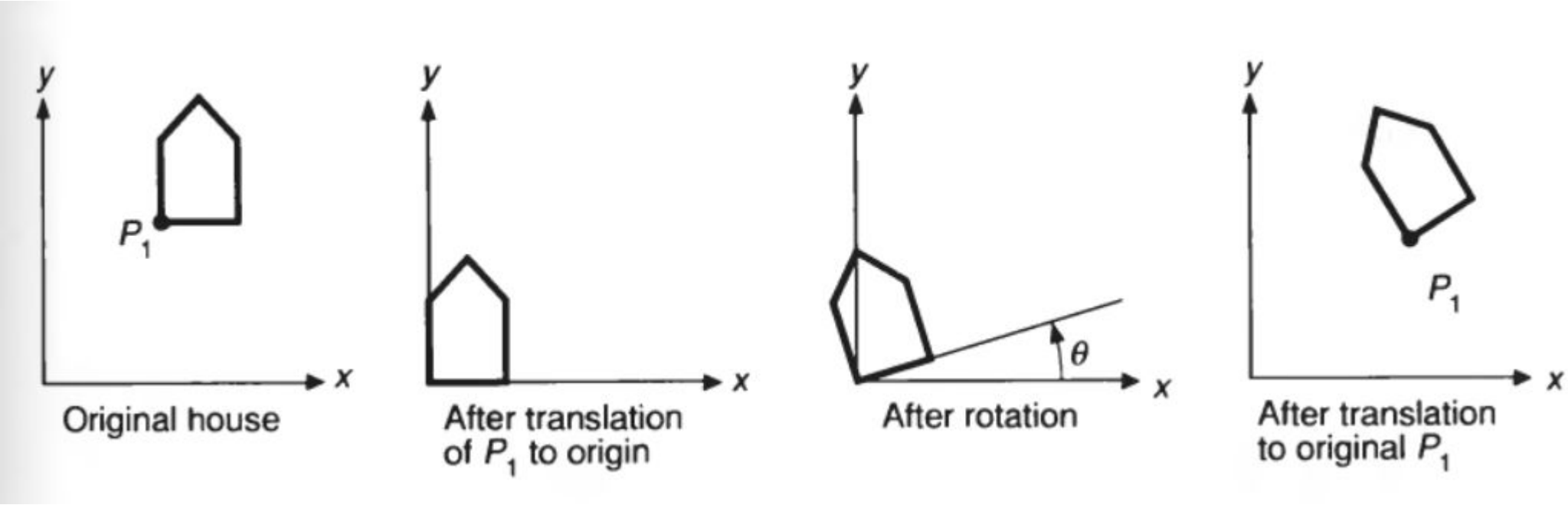
$$P' = S(s_{x1}, s_{y1}) \cdot P,$$
$$P'' = S(s_{x2}, s_{y2}) \cdot P',$$

Double Scaling

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

Double Rotation

Rotate about any arbitrary point



Rotate about any arbitrary point

$$T(x_1, y_1) \cdot R(\theta) \cdot T(-x_1, -y_1) = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

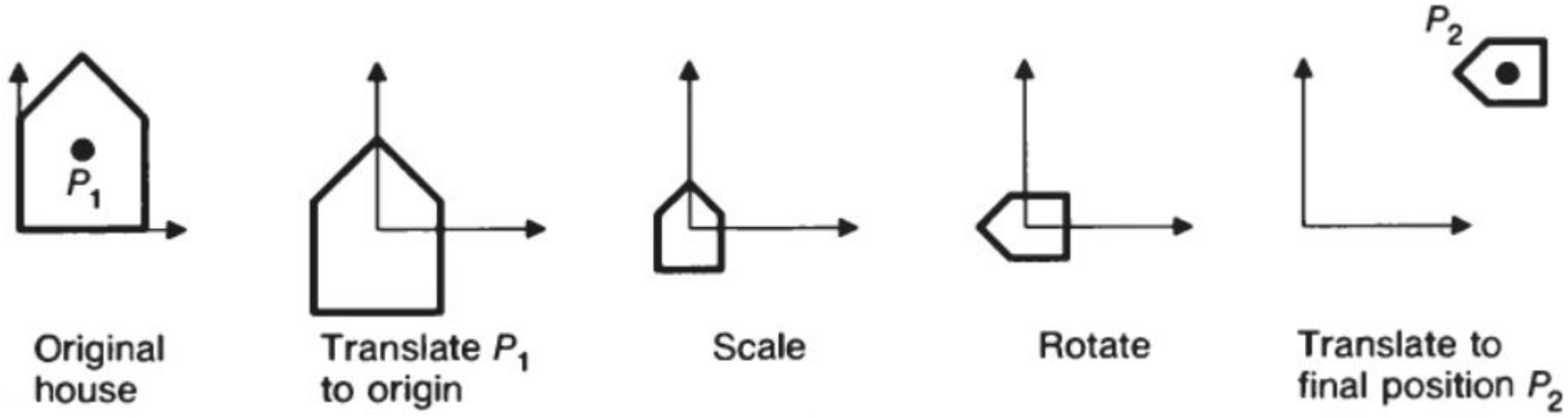
$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_1(1 - \cos\theta) + y_1\sin\theta \\ \sin\theta & \cos\theta & y_1(1 - \cos\theta) - x_1\sin\theta \\ 0 & 0 & 1 \end{bmatrix}. \qquad (5.30)$$

Scale about any arbitrary point

$$T(x_1, y_1) \cdot S(s_x, s_y) \cdot T(-x_1, -y_1) = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} s_x & 0 & x_1(1 - s_x) \\ 0 & s_y & y_1(1 - s_y) \\ 0 & 0 & 1 \end{bmatrix}. \qquad (5.31)$$

Scale and Rotate about any arbitrary point



Order of Operations

M_1	M_2
Translate	Translate
Scale	Scale
Rotate	Rotate
Scale (with $s_x = s_y$)	Rotate

Problems Related to 2D Transformation

- Derive the transformation that rotates an object point θ° about the origin. Write the matrix representation for this rotation.
- Find the matrix that represents rotation of an object by 30° about the origin.
- What are the new coordinates of the point P(2, -4) after the rotation?

- Describe the transformation that rotates an object point, Q(x, y), θ° about a fixed center of rotation P(h, k).
- Write the general form of the matrix for rotation about a point P(h, k).
- Perform a 45° rotation of triangle ,A(0, 0),B(1, 1), C(5, 2)
  - (a) about the origin
  - (b) about P(-l.-l).

Problems Related to 2D Transformation

- Write the general form of a scaling matrix with respect to a fixed point P(h, k).
- Magnify the triangle with vertices A (0, 0), B(1, 1), and C(5,2) to twice its size while keeping C(5,2)fixed.

Problems Related to 2D Transformation

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

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 $(0, b)$.

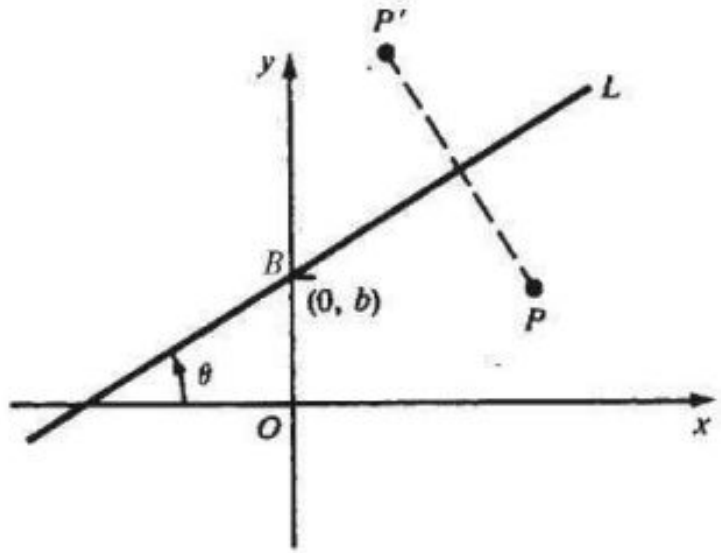
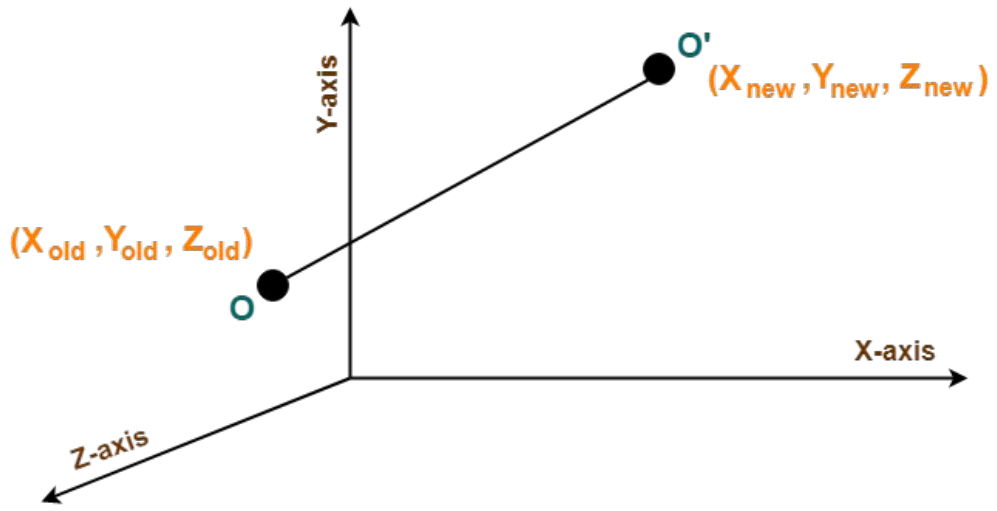


Fig. 4-15

Problems Related to 2D Transformation

- Reflect the diamond-shaped polygon whose vertices are A(-1, 0), B(0, -2), C(1, 0), and D(0, 2) about (a) the horizontal line y = 2, (b) the vertical line x = 2, and (c) the line y = x + 2.

3D Transformation– Translation

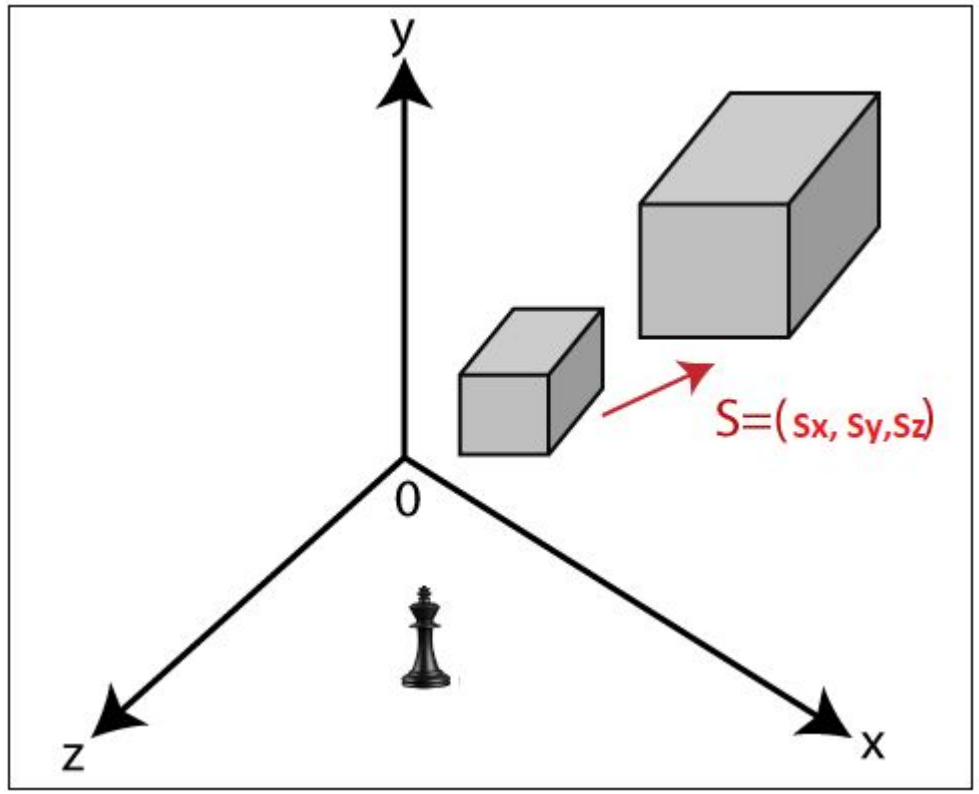


3D Translation in Computer Graphics

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

3D Translation Matrix

3D Transformation– Scaling



$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

3D Scaling Matrix

3D Transformation – Rotation

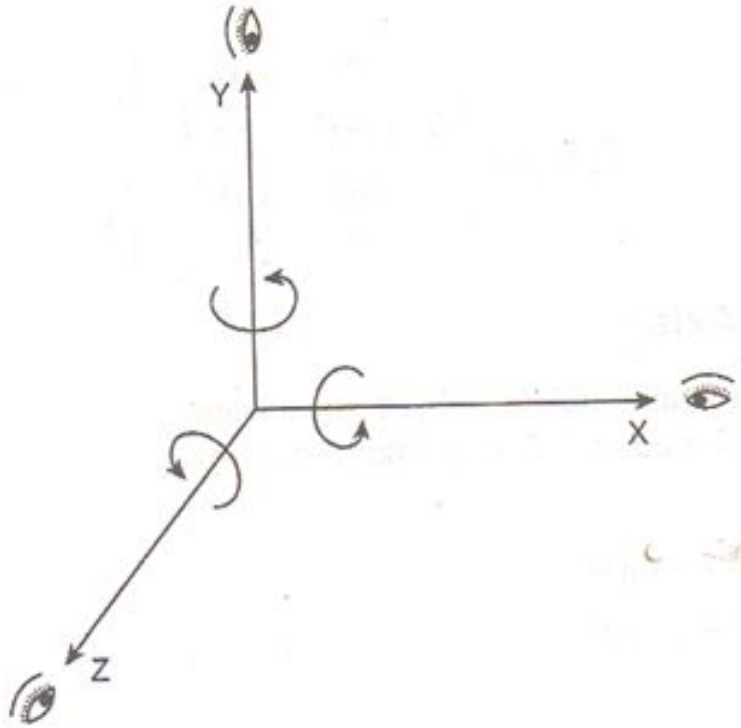


Fig: 3D Rotation

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

3D Rotation Matrix

(For X-Axis Rotation)

3D Transformation – Rotation

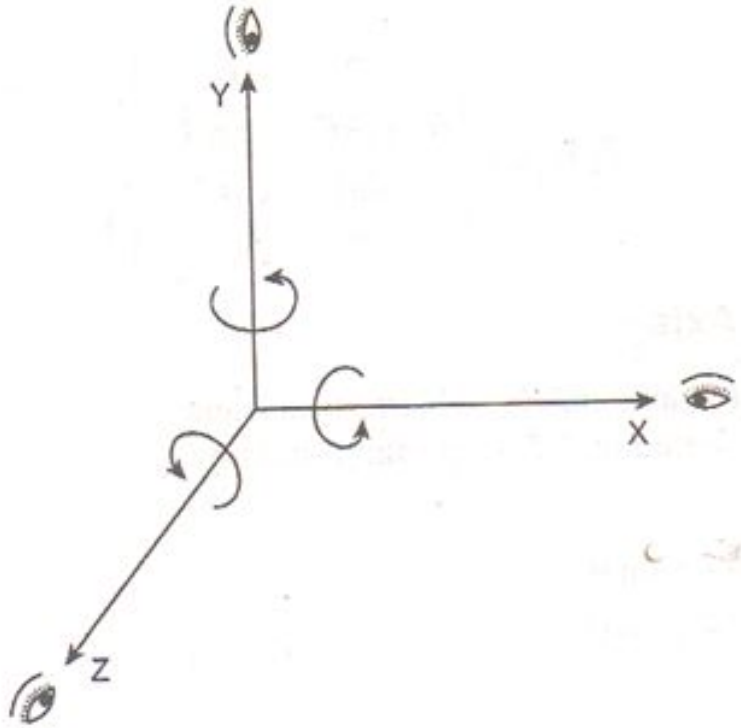


Fig: 3D Rotation

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

3D Rotation Matrix

(For Y-Axis Rotation)

3D Transformation – Rotation

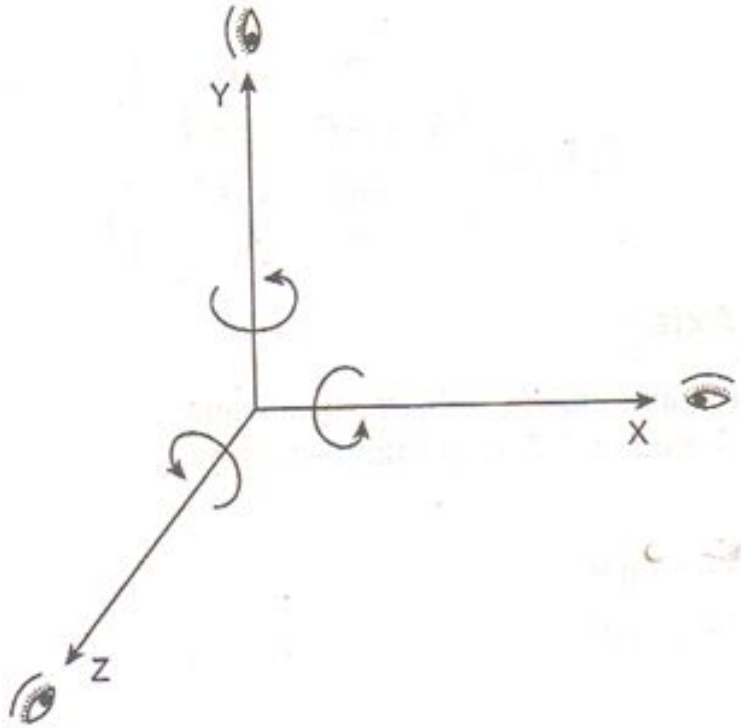


Fig: 3D Rotation

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

3D Rotation Matrix

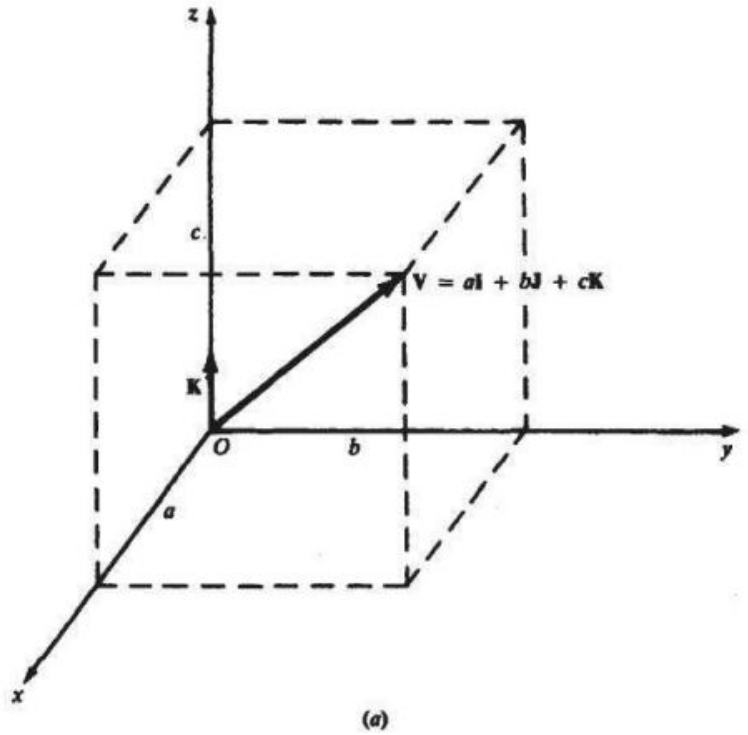
(For Z-Axis Rotation)

3D VISULIZATION

- https://hal.bim.msu.edu/CMEonLine/AxisReferenceSystems/AxisSystems/rotation.html
- https://www.geogebra.org/m/kb4vcwuk?fbclid=IwAR0KeOwxPwgJkc_uMJh0yOfM1tZIvv08NvoJhoqH6LQvfOzMhoFAyDPZJio

Tilting

- Define tilting as a rotation about the x axis followed by a rotation about the y axis: (a) find the tilting matrix; (b) does the order of performing the rotation matter?



(a)

Tilting

(*a*) We can find the required transformation T by composing (concatenating) two rotation matrices:

$$T = R_{\theta_y,\mathbf{J}} \cdot R_{\theta_x,\mathbf{I}}$$

$$= \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta_y & \sin\theta_y \sin\theta_x & \sin\theta_y \cos\theta_x & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Tilting

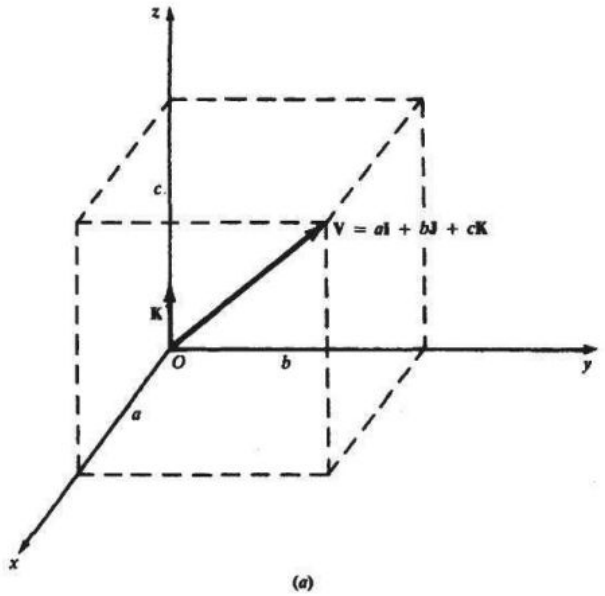
(*b*) We multiply $R_{\theta_x,\mathbf{I}} \cdot R_{\theta_y,\mathbf{J}}$ to obtain the matrix

$$\begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ \sin\theta_x \sin\theta_y & \cos\theta_x & -\sin\theta_x \cos\theta_y & 0 \\ -\cos\theta_x \sin\theta_y & \sin\theta_x & \cos\theta_x \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

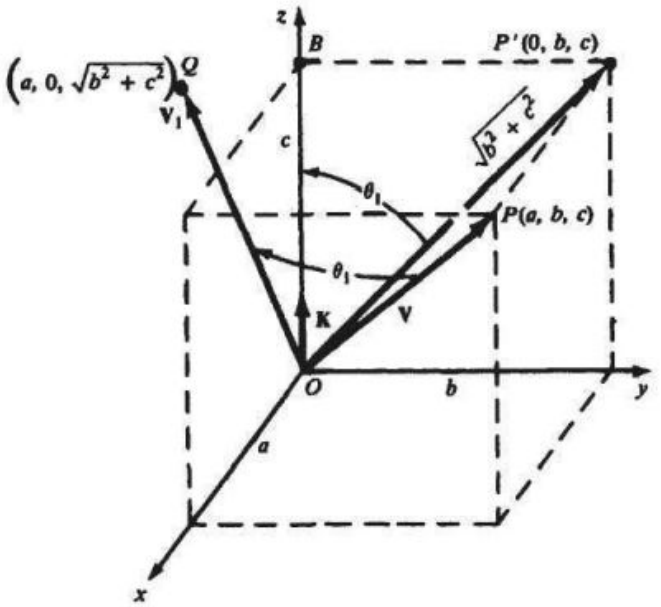
This is not the same matrix as in part *a*; thus the order of rotation matters.

Tilting

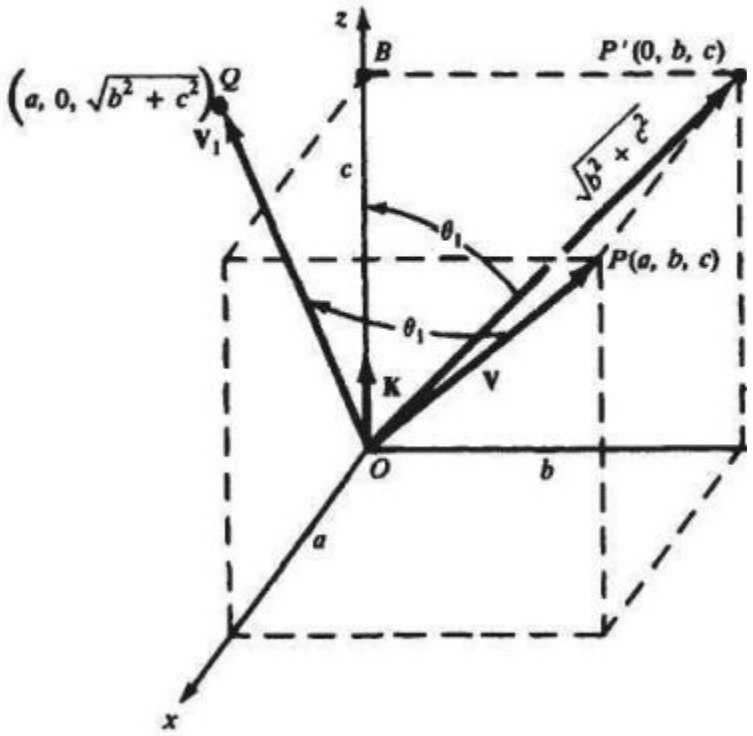
. Find a transformation A_V which aligns a given vector **V** with the vector **K** along the positive z axis.



(a)



Tilting



Implementing step 1 from Fig. 6-4(b), we observe that the required angle of rotation θ_1 can be found by looking at the projection of **V** onto the *yz* plane. (We assume that b and c are not both zero.) From triangle $OP'B$:

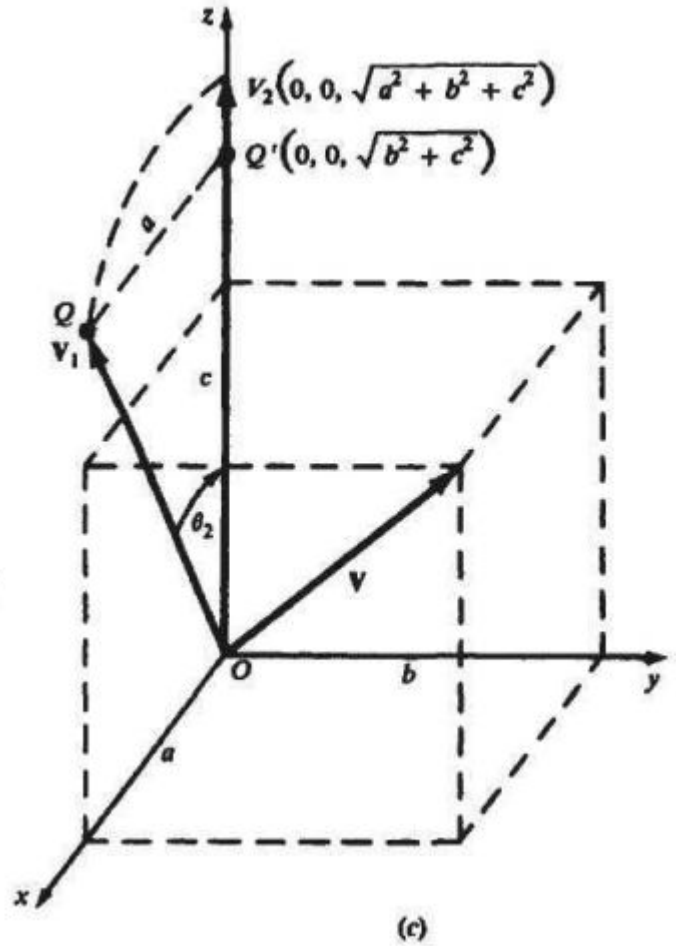
$$\sin\theta_1 = \frac{b}{\sqrt{b^2 + c^2}} \qquad \cos\theta_1 = \frac{c}{\sqrt{b^2 + c^2}}$$

The required rotation is

$$R_{\theta_1,\mathbf{I}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \dfrac{c}{\sqrt{b^2 + c^2}} & -\dfrac{b}{\sqrt{b^2 + c^2}} & 0 \\ 0 & \dfrac{b}{\sqrt{b^2 + c^2}} & \dfrac{c}{\sqrt{b^2 + c^2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Applying this rotation to the vector **V** produces the vector $\mathbf{V}_1$ with the components $(a, 0, \sqrt{b^2 + c^2})$.

Tilting



(c)

Implementing step 2 from Fig. 6-4(c), we see that a rotation of $-\theta_2$ degrees is required, and so from triangle OQQ':

$$\sin(-\theta_2) = -\sin\theta_2 = -\frac{a}{\sqrt{a^2+b^2+c^2}} \quad \text{and} \quad \cos(-\theta_2) = \cos\theta_2 = \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}}$$

Then

$$R_{-\theta_2,\mathbf{J}} = \begin{pmatrix} \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}} & 0 & \frac{-a}{\sqrt{a^2+b^2+c^2}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{a}{\sqrt{a^2+b^2+c^2}} & 0 & \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Tilting

Since $|\mathbf{V}| = \sqrt{a^2 + b^2 + c^2}$, and introducing the notation $\lambda = \sqrt{b^2 + c^2}$, we find

$$A_{\mathbf{V}} = R_{-\theta_2,\mathbf{J}} \cdot R_{\theta_1,\mathbf{I}}$$

$$= \begin{pmatrix} \frac{\lambda}{|\mathbf{V}|} & \frac{-ab}{\lambda|\mathbf{V}|} & \frac{-ac}{\lambda|\mathbf{V}|} & 0 \\ 0 & \frac{c}{\lambda} & \frac{-b}{\lambda} & 0 \\ \frac{a}{|\mathbf{V}|} & \frac{b}{|\mathbf{V}|} & \frac{c}{|\mathbf{V}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Math Problems

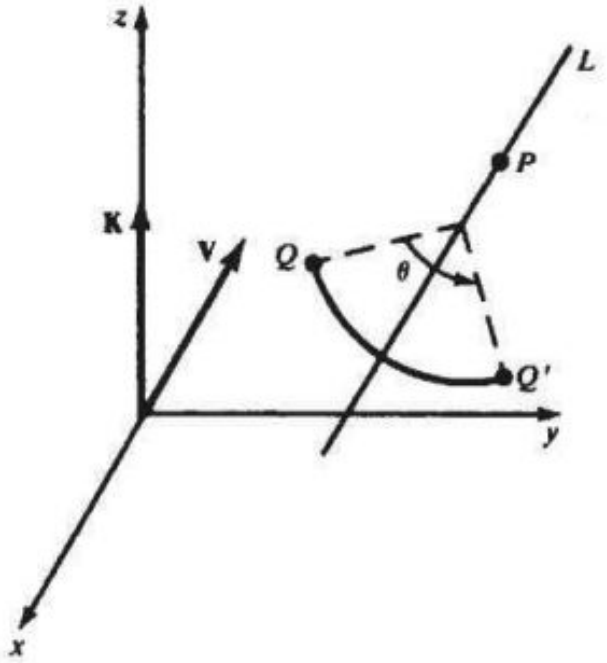
In the same manner we calculate the inverse transformation that aligns the vector **K** with the vector **V**:

$$A_{\mathbf{v}}^{-1} = (R_{-\theta_2,\mathbf{J}} \cdot R_{\theta_1,\mathbf{I}})^{-1} = R_{\theta_1,\mathbf{I}}^{-1} \cdot R_{-\theta_2,\mathbf{J}}^{-1} = R_{-\theta_1,\mathbf{I}} \cdot R_{\theta_2,\mathbf{J}}$$

$$= \begin{pmatrix} \frac{\lambda}{|\mathbf{V}|} & 0 & \frac{a}{|\mathbf{V}|} & 0 \\ \frac{-ab}{\lambda|\mathbf{V}|} & \frac{c}{\lambda} & \frac{b}{|\mathbf{V}|} & 0 \\ \frac{-ac}{\lambda|\mathbf{V}|} & -\frac{b}{\lambda} & \frac{c}{|\mathbf{V}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Math Problems

- Let an axis of rotation L be specified by a direction vector V and a location point P. Find the transformation for a rotation of $\theta°$ about L.

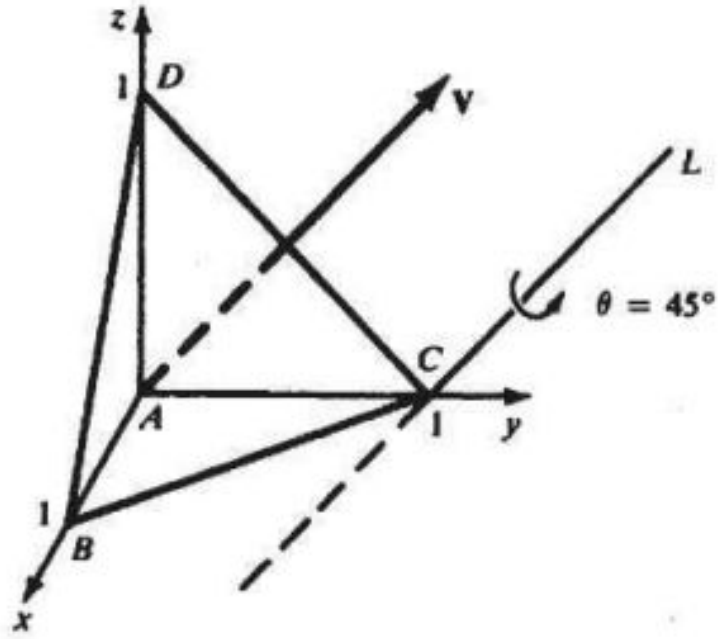


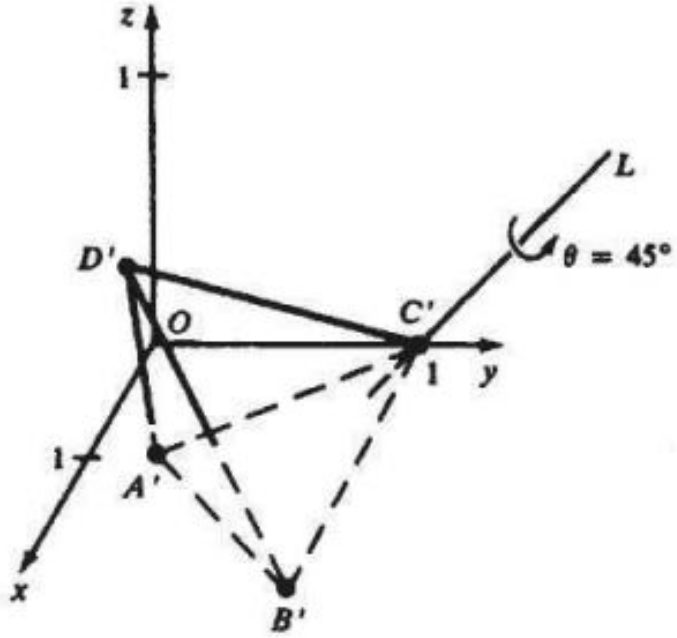
1. Translate P to the origin.
2. Align $\mathbf{V}$ with the vector $\mathbf{K}$.
3. Rotate by $\theta°$ about $\mathbf{K}$.
4. Reverse steps 2 and 1.

$$R_{\theta,L} = T_{-P}^{-1} \cdot A_{\mathbf{V}}^{-1} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{V}} \cdot T_{-P}$$

Math Problems

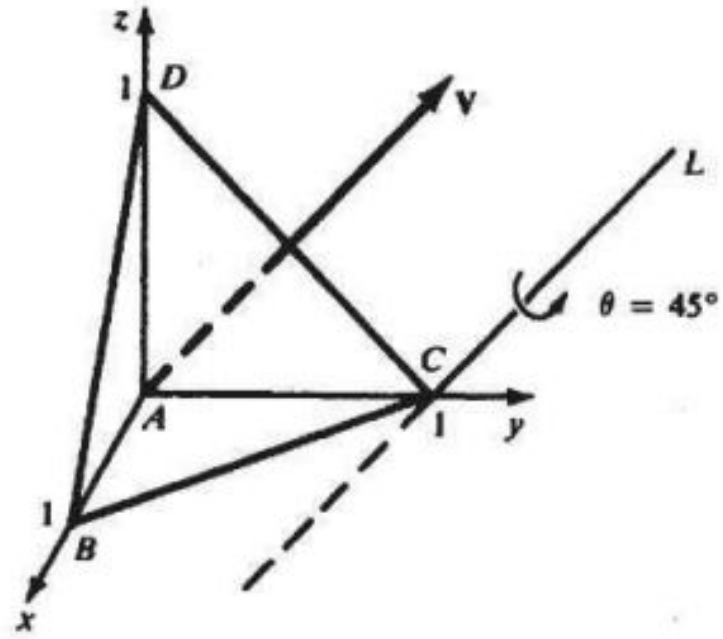
- The pyramid denned by the coordinates ,A(0, 0, 0), B(1, 0, 0), C(0, 1, 0), and D(0, 0, 1) is rotated 45° about the line L that has the direction V = J + K and passing through point C(0, 1, 0) (Fig. 6-6). Find the coordinates of the rotated figure.





Math Problems

- The pyramid denned by the coordinates ,A(0, 0, 0), B(1, 0, 0), C(0, 1, 0), and D(0, 0, 1) is rotated 45° about the line L that has the direction V = J + K and passing through point C(0, 1, 0) (Fig. 6-6). Find the coordinates of the rotated figure.



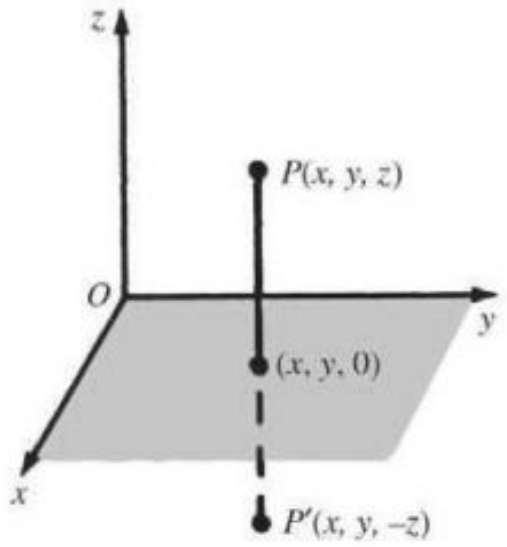
Math Problems

Find a transformation $A_{V,N}$ which aligns a vector V with a vector N.

$$\mathbf{A_{V,N}} = \mathbf{A_N^{-1}} \cdot \mathbf{A_V}$$

Math Problems

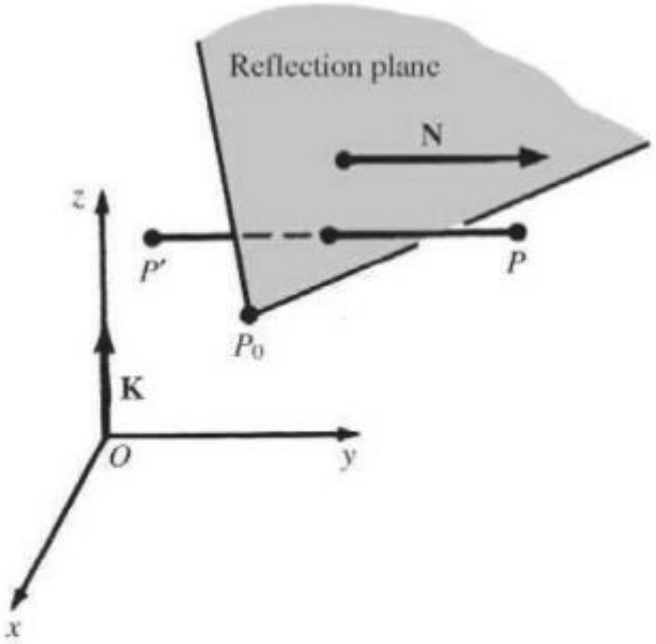
Find the transformation for mirror reflection with respect to the xy plane..



$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Math Problems

Find the transformation for mirror reflection with respect to a given plane

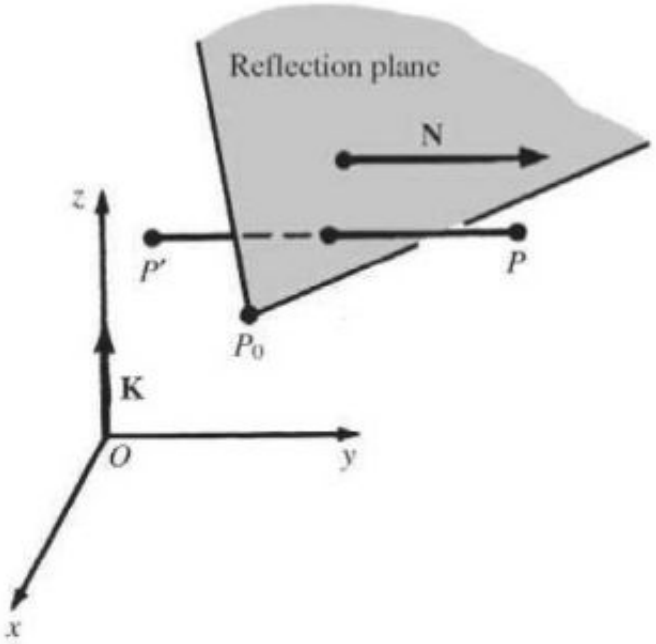


1. Translate P_0 to the origin:
2. Align the normal vector **N** with the vector **K** normal to the xy plane.
3. Perform the mirror reflection in the xy plane (Prob. 6.6).
4. Reverse steps 1 and 2.

$$M_{\mathbf{N},P_0} = T_{\mathbf{V}}^{-1} \cdot A_{\mathbf{N}}^{-1} \cdot M \cdot A_{\mathbf{N}} \cdot T_{\mathbf{V}}$$

Math Problems

Find the matrix for mirror reflection with respect to the plane passing through the origin and having a normal vector whose direction is N = I + J + K



$$M_{\mathbf{N},P_0} = T_{\mathbf{V}}^{-1} \cdot A_{\mathbf{N}}^{-1} \cdot M \cdot A_{\mathbf{N}} \cdot T_{\mathbf{V}}$$

Math Problems

Find a transformation $A_{V,N}$ which aligns a vector V= i+j+k with a vector N=2I-J-K.

References

Chapter 5

- Computer Graphics Principle and Practice (Foley)

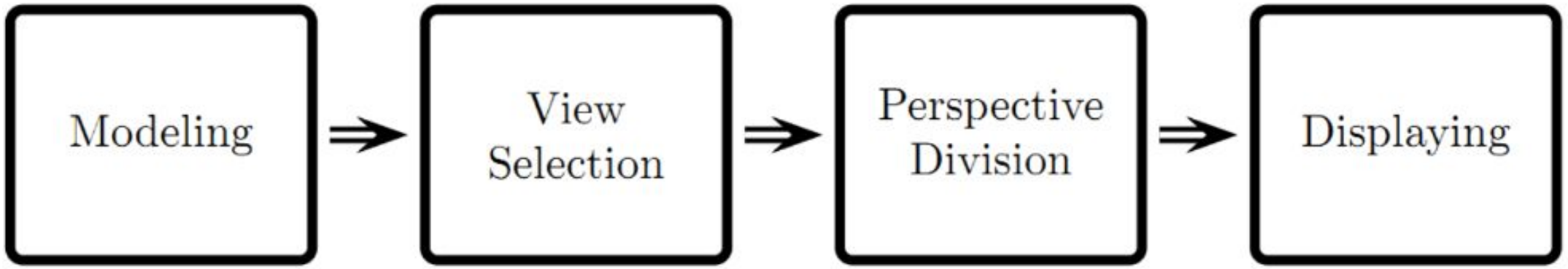
Chapter 4

- Computer Graphics (Schaum')

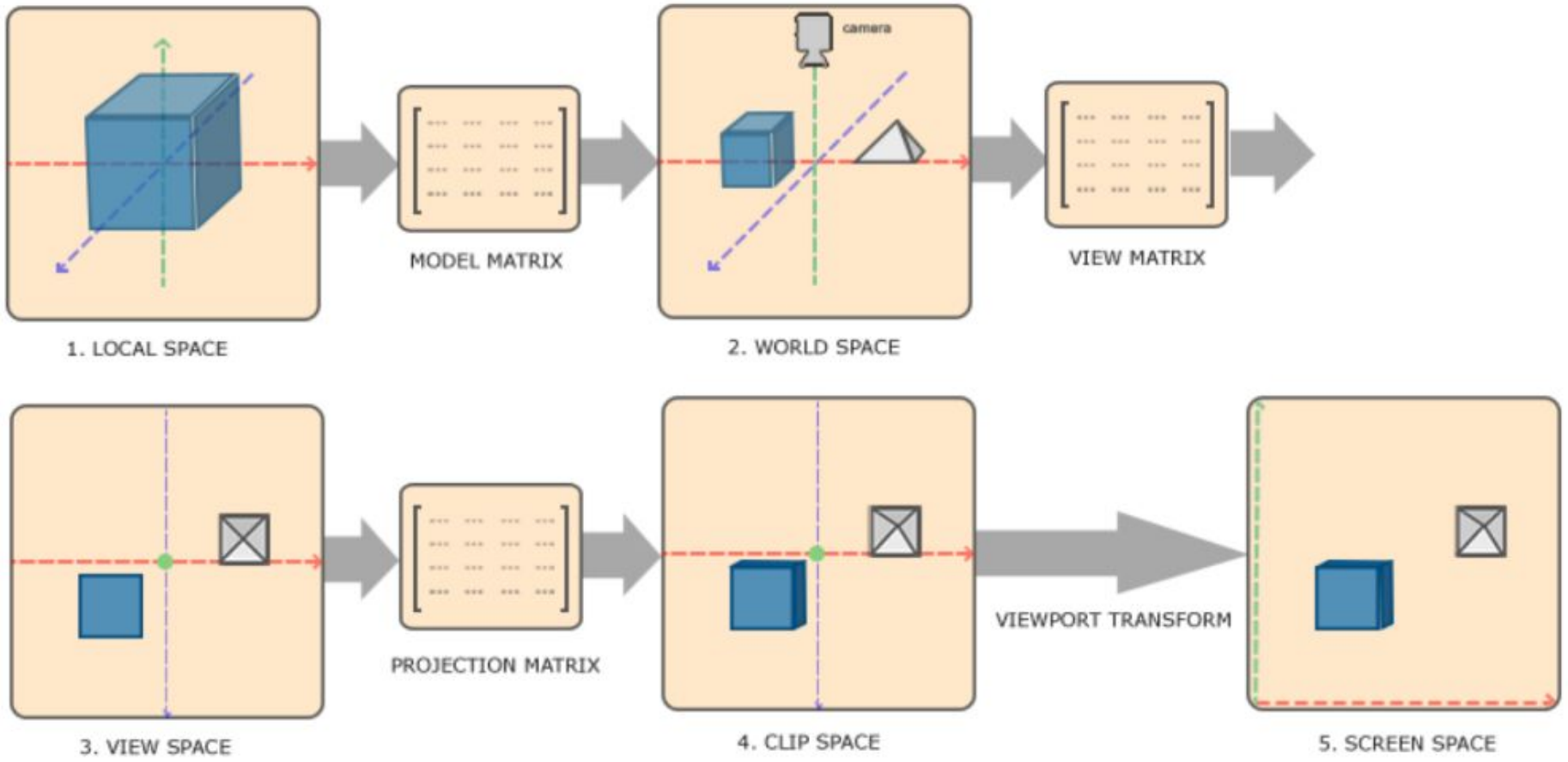
OPENGL & PROJECTION

MD RAKIBUL HAQUE
LECTURER
CSE, RUET.

OPENGL RENDERING PIPELINE



THE GLOBAL PICTURE



OPENGL RENDERING PIPELINE

MODELING

- 3D MODEL OF SHAPE TO BE DISPLAYED IS CREATED
- MODEL VIEW MATRIX IS APPLIED TO LOCAL CO-ORDINATES
- WORLD COORDINATE IS GENERATED FROM LOCAL COORDINATES
- MODEL VIEW MATRIX REPRESENTS AFFINE TRANSFORMATION SUCH AS TRANSLATION, SCALING AND ROTATION

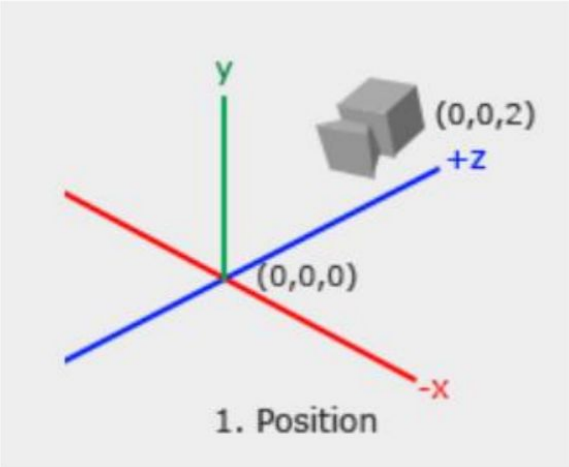
OPENGL RENDERING PIPELINE

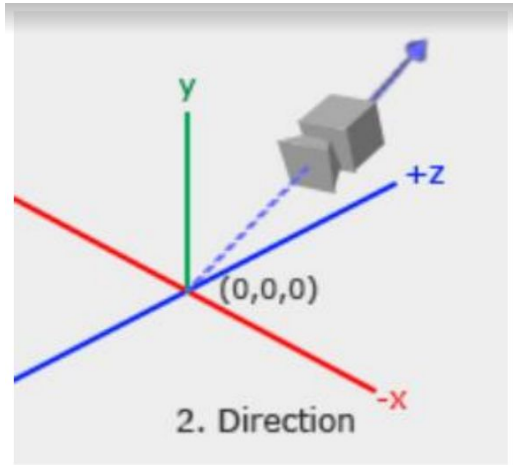
VIEW SELECTION

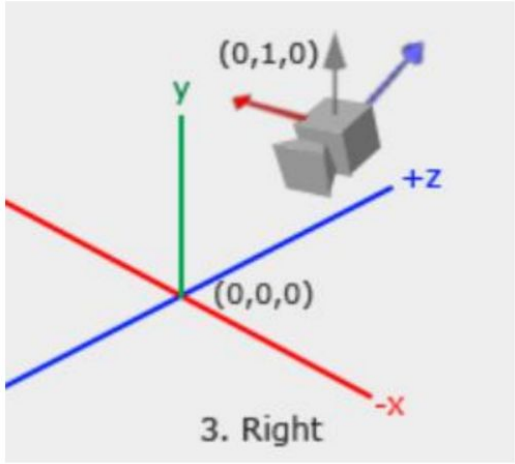
- The view matrix transforms all the world coordinates into view coordinates that are relative to the camera's position and direction.
- To define a camera we need its
  - position in world space,
  - the direction it's looking at,
  - a vector pointing to the right and
  - a vector pointing upwards from the camera.

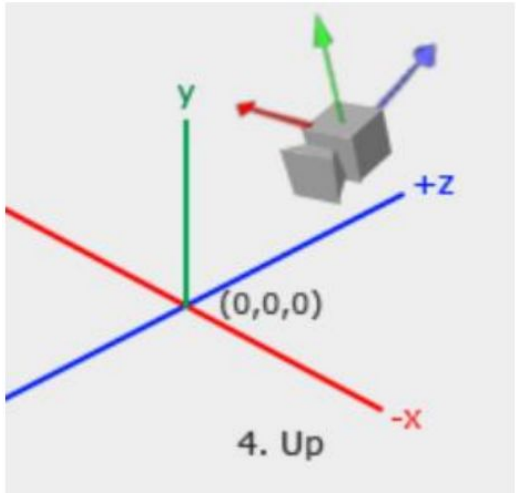
OPENGL RENDERING PIPELINE

VIEW SELECTION





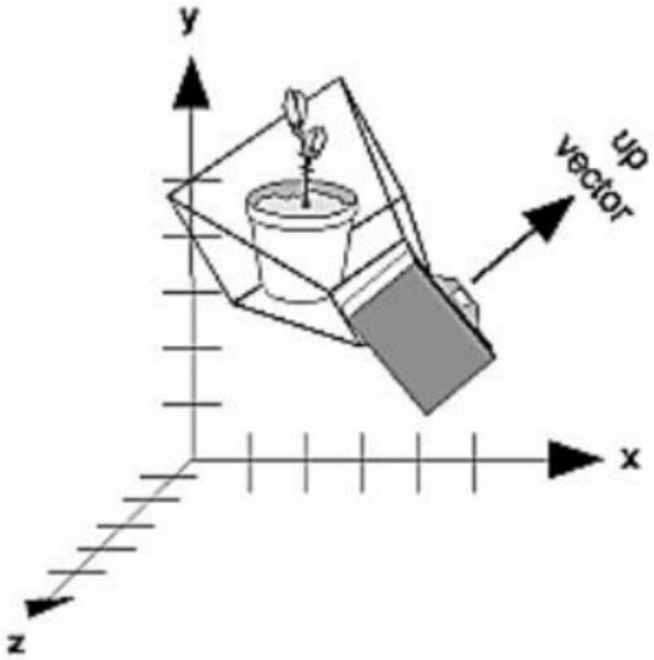




OPENGL RENDERING PIPELINE

VIEW SELECTION

$$LookAt = \begin{bmatrix} R_x & R_y & R_z & 0 \\ U_x & U_y & U_z & 0 \\ D_x & D_y & D_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



gluLookAt(4.0, 2.0, 1.0, 2.0, 4.0, -3.0, 2.0, 2.0, -1.0);

OPENGL RENDERING PIPELINE

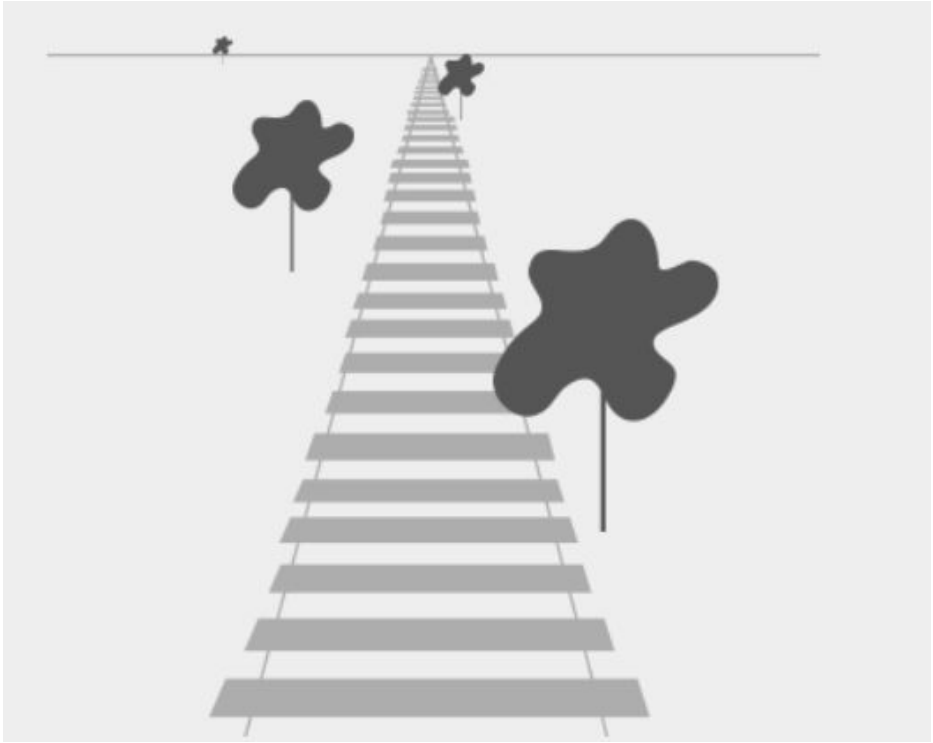
CLIP SPACE

- OpenGL expects the coordinates to be within a specific range and any coordinate that falls outside this range is clipped. Coordinates that are clipped are discarded, so the remaining coordinates will end up as fragments visible on your screen. This is also where clip space gets its name from.

OPENGL RENDERING PIPELINE

PERSPECTIVE DIVISION

- The projection matrix maps a given view co-ordinates to clip space, but also manipulates the w value of each vertex coordinate in such a way that the further away a vertex coordinate is from the viewer, the higher this w component becomes. Once the coordinates are transformed to clip space they are in the range -w to w (anything outside this range is clipped). OpenGL requires that the visible coordinates fall between the range -1.0 and 1.0 as the final vertex shader output, thus once the coordinates are in clip space, perspective division is applied to the clip space coordinates.



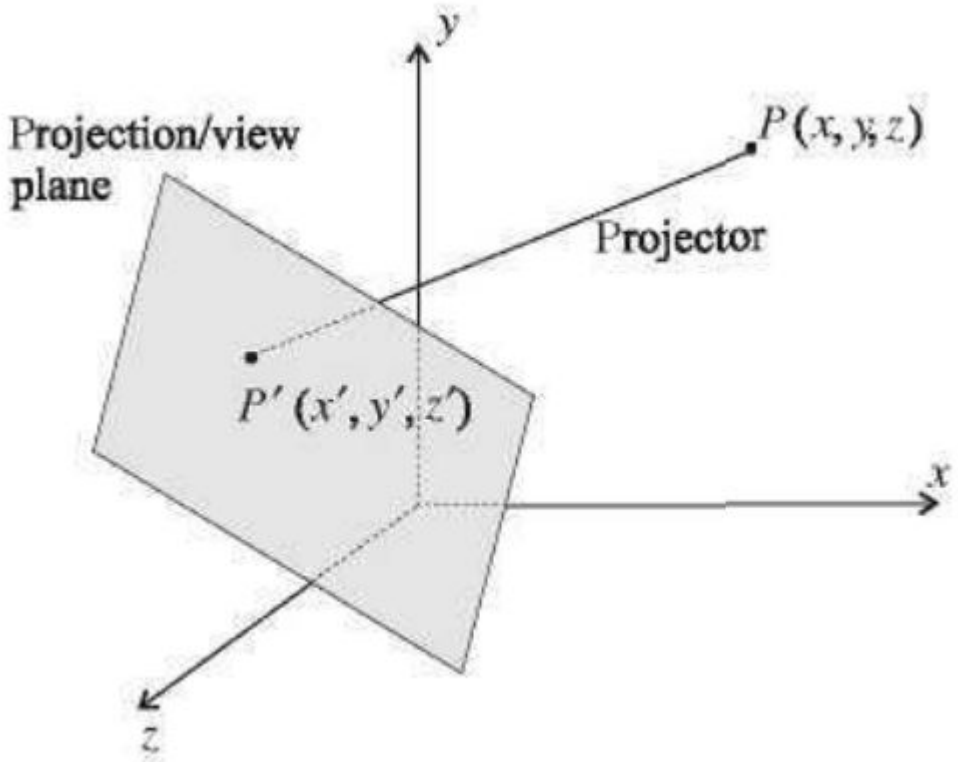
OPENGL RENDERING PIPELINE

DISPLAYING

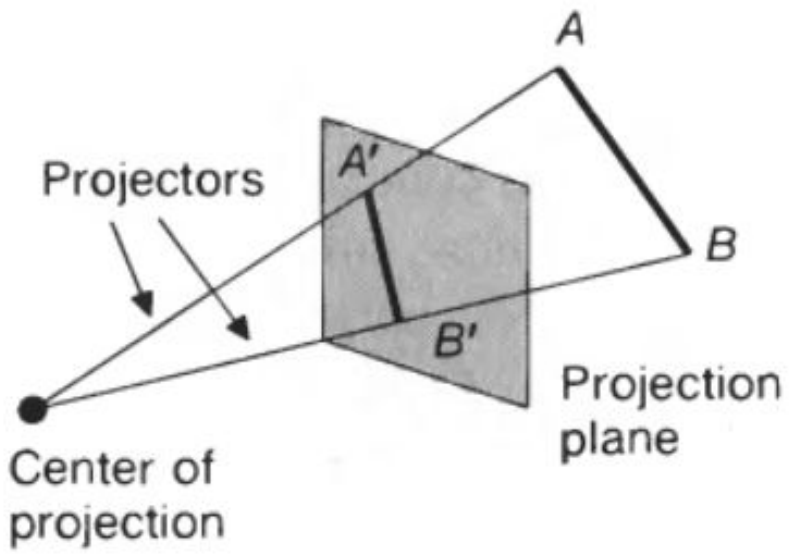
Transform the clip coordinates to screen coordinates in a process we call viewport transform that transforms the coordinates from -1.0 and 1.0 to the coordinate range defined by glViewport. The resulting coordinates are then sent to the rasterizer to turn them into fragments.

PROJECTION

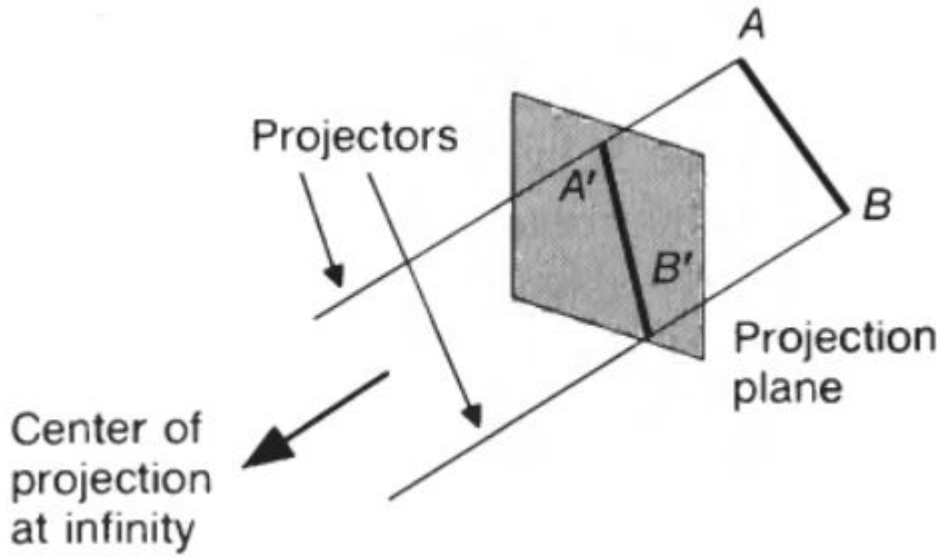
- PROJECTION TRANSFORMS A POINT IN N-DIMENSIONAL CO-ORDINATE SYSTEM IN A CO-ORDINATE SYSTEM OF LESS THAN N.
- PROJECTING OBJECT WITH 3-D COORDINATES TO A 2-D PLANE.
- PLANAR GEOMETRIC PROJECTION.



TYPES OF PROJECTION

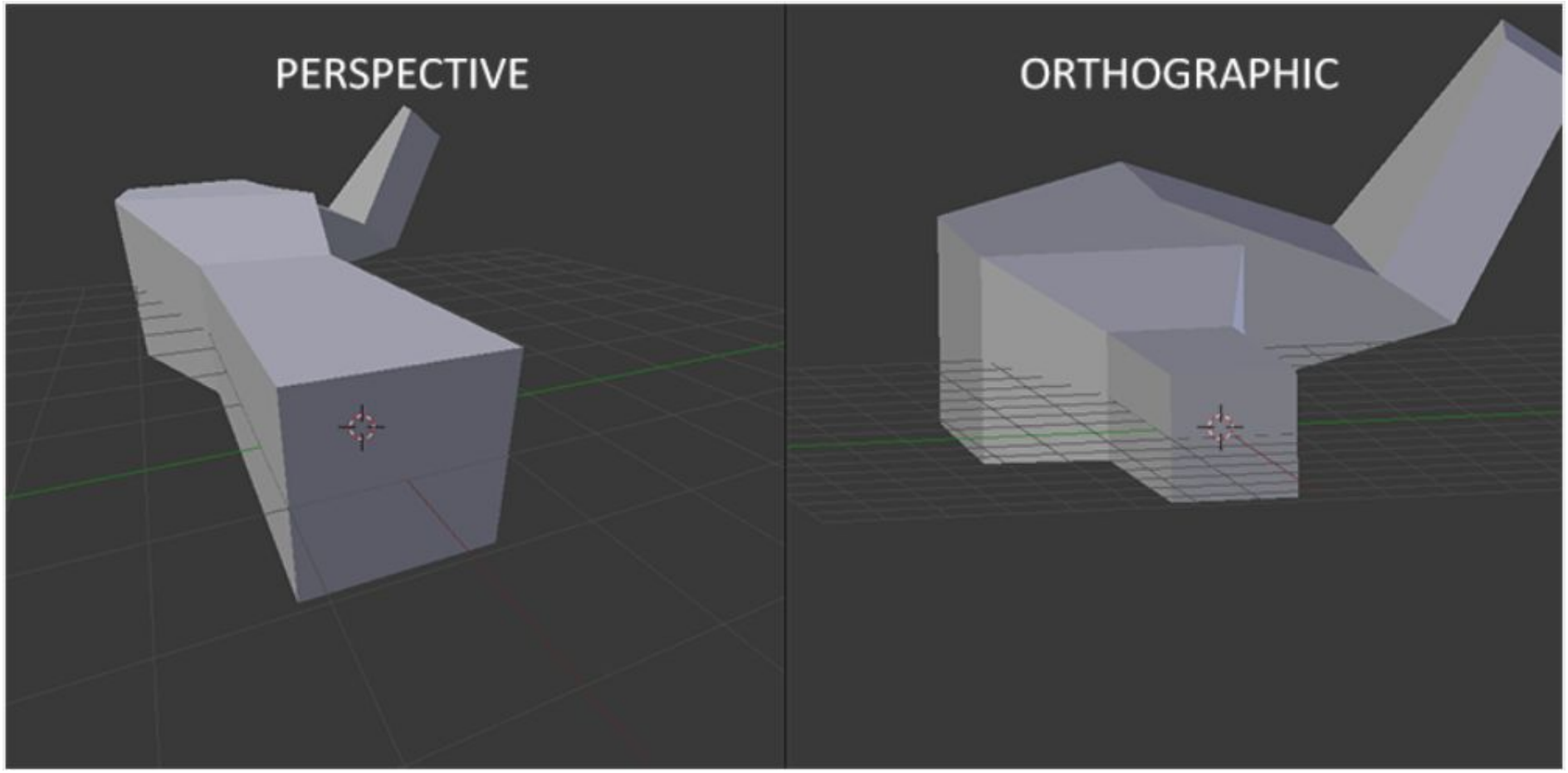


PERSPECTIVE PROJECTION

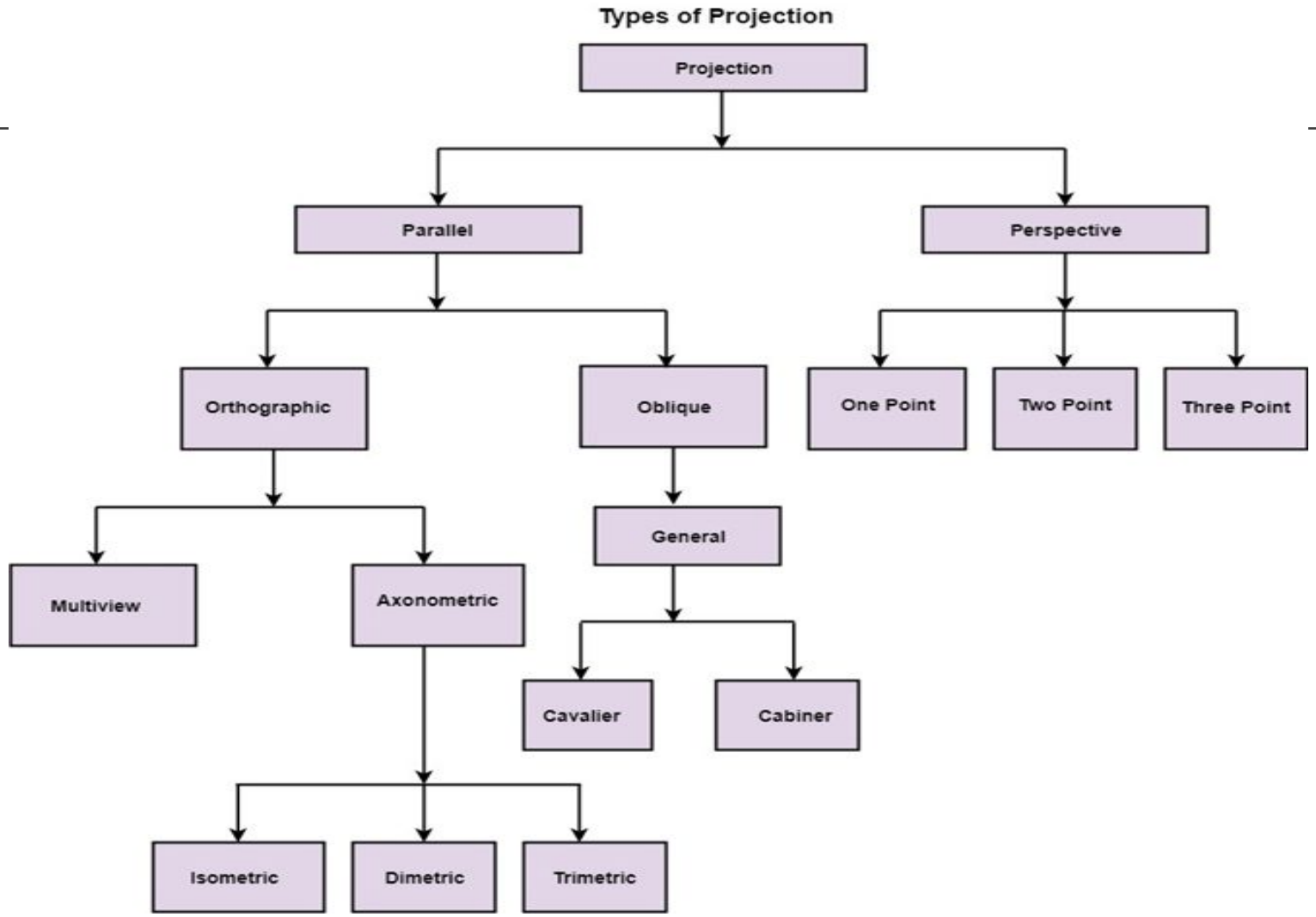


PARALLEL PROJECTION

TYPES OF PROJECTION

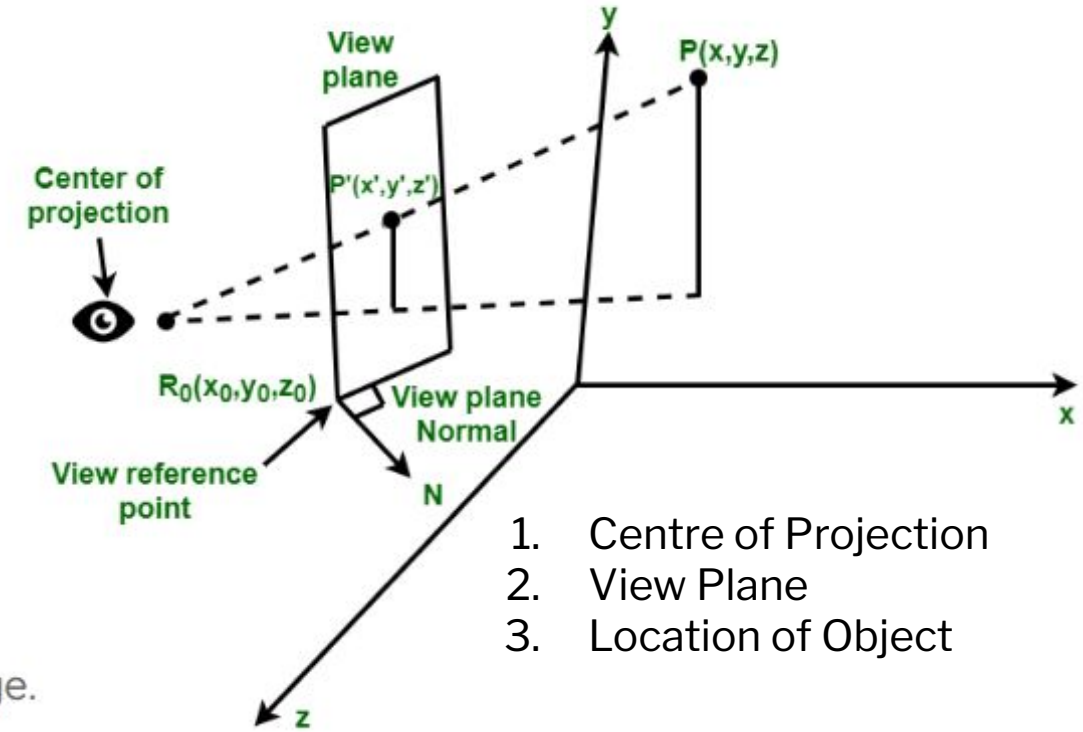


TYPES OF PROJECTION



PERSPECTIVE PROJECTION

In Perspective Projection the center of projection is at finite distance from projection plane. This projection produces realistic views but does not preserve relative proportions of an object dimensions. Projections of distant object are smaller than projections of objects of same size that are closer to projection plane



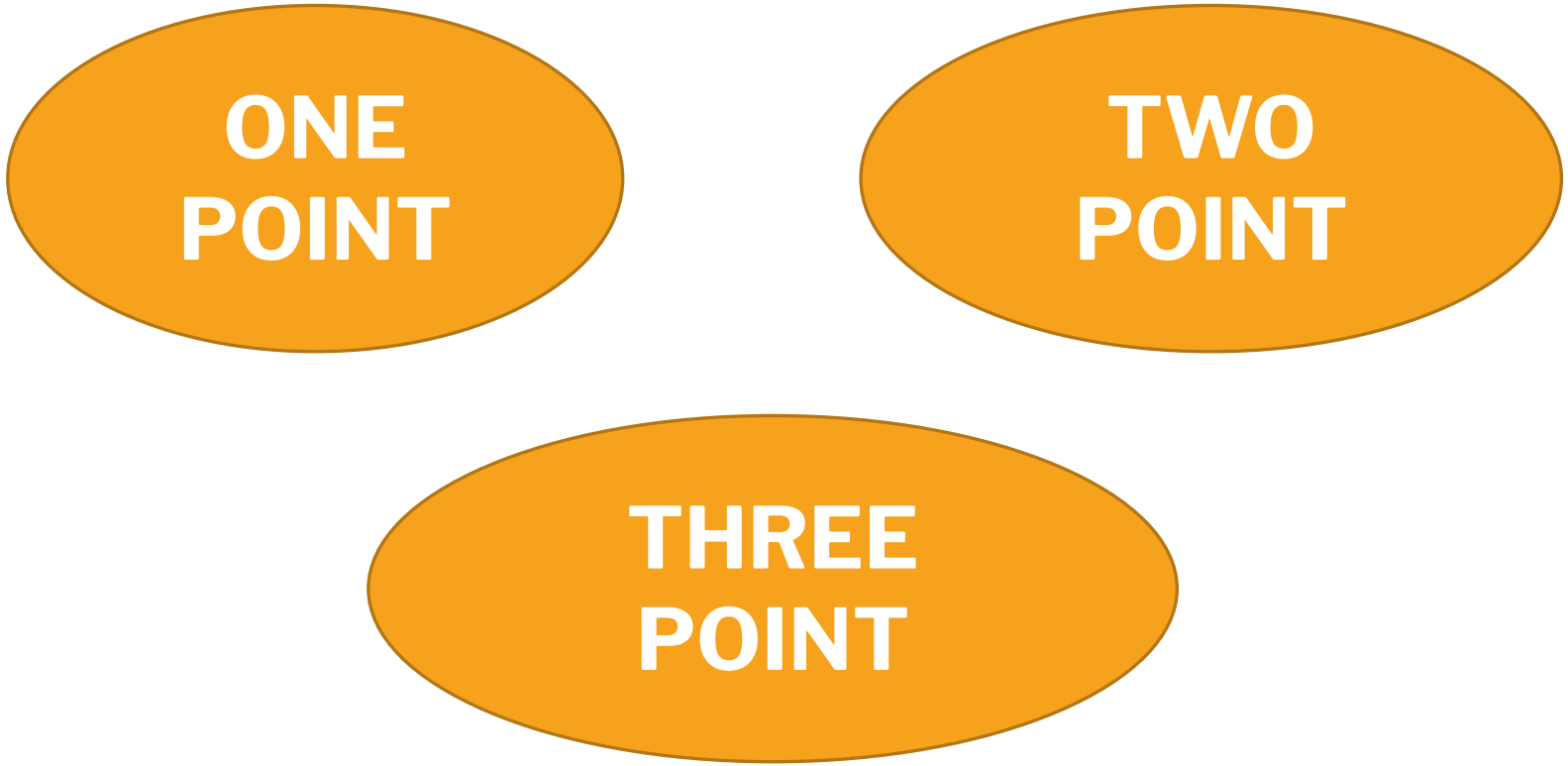
1. Centre of Projection
2. View Plane
3. Location of Object

- It is used in 3D games and graphics editing.
  - It is also used to represent 3D objects.
  - We can also include perspective in the background of an image.
  - We can also insert the shadow effect in an image.

VANISHING POINT

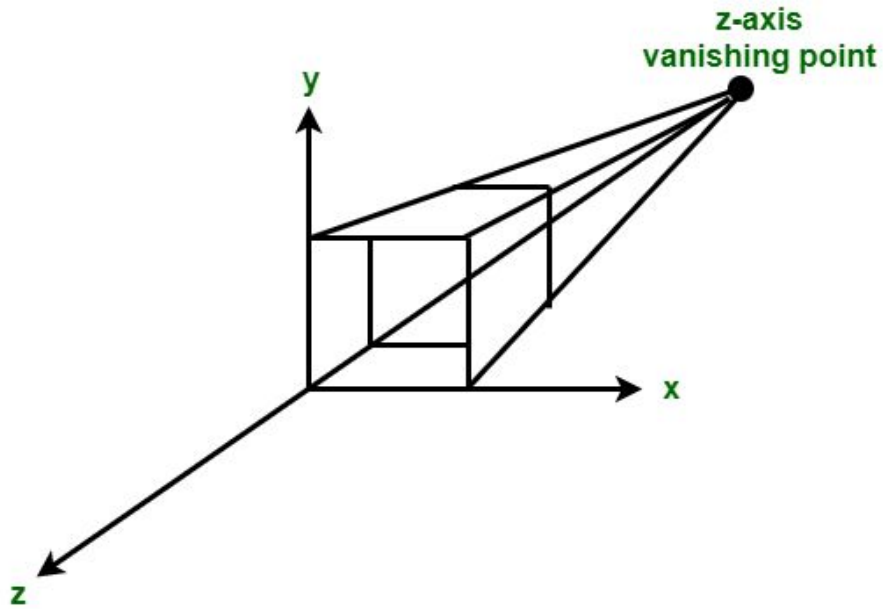
- A vanishing point is a point on the image plane of a perspective drawing where the two-dimensional perspective projections of mutually parallel lines in three-dimensional space appear to converge.
- Perspective Projection is classified based upon vanishing point.

TYPES OF PERSPECTIVE PROJECTION



ONE POINT

- One point perspective projection occurs when any of principal axes intersects with projection plane

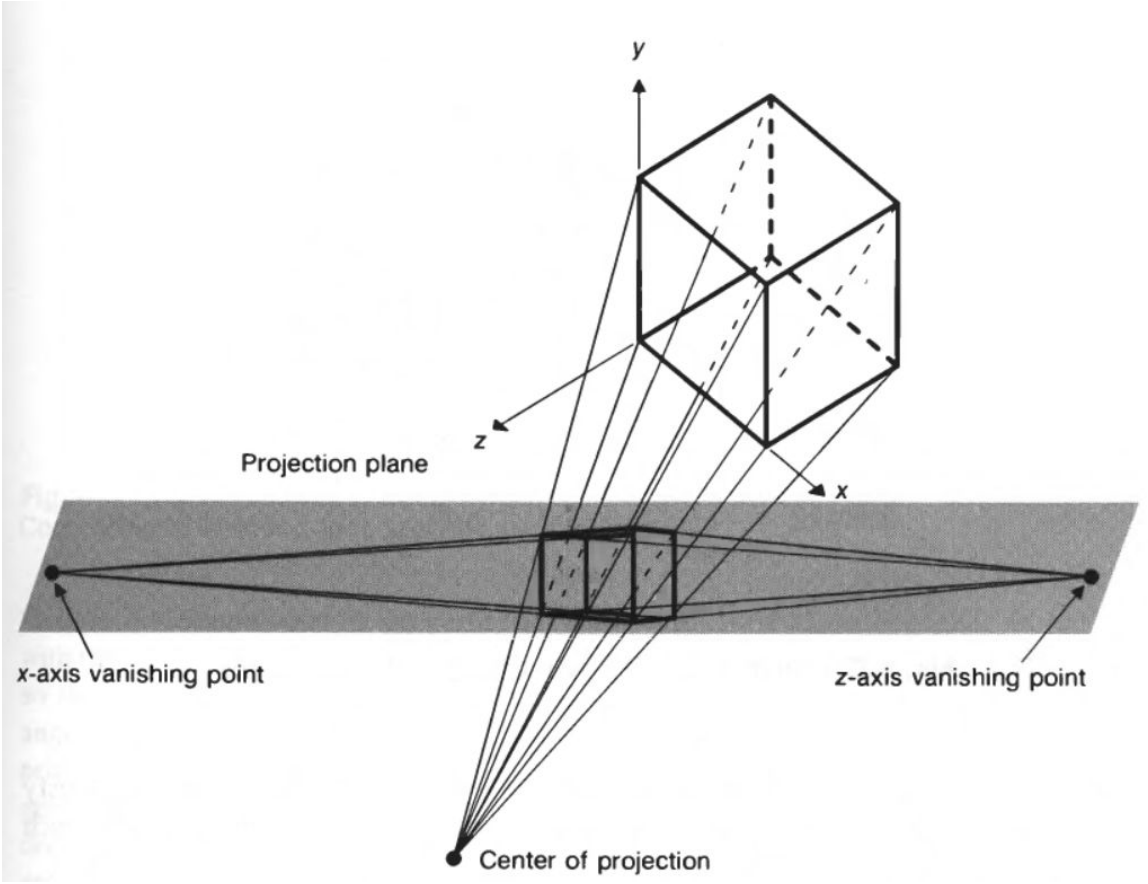


- Use of One Point- The One Point projection is mostly used to draw the images of roads, railway tracks, and buildings.

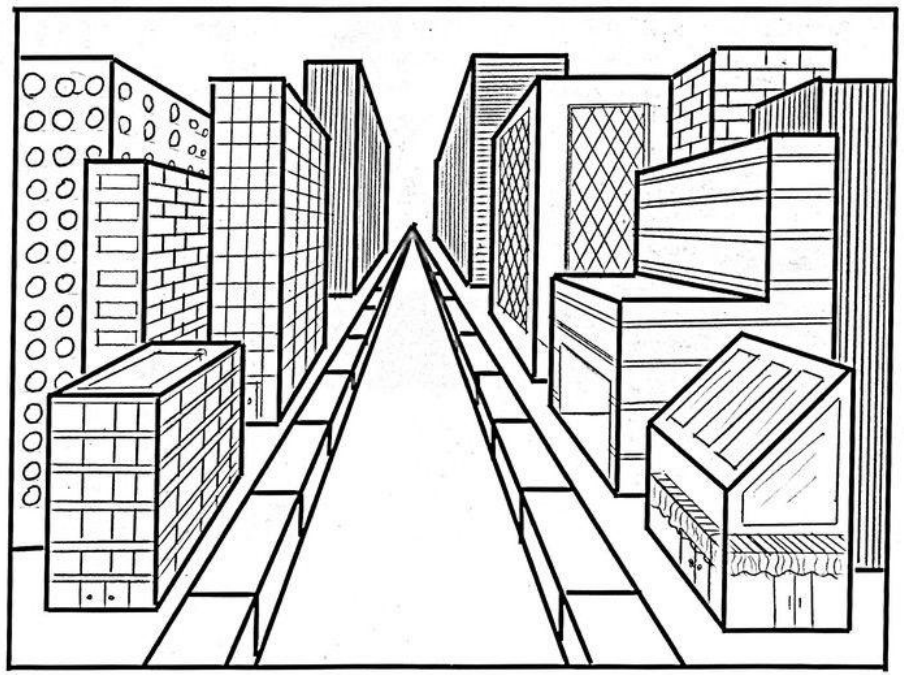
- In the above figure, z axis intersects projection plane whereas x and y axis remain parallel to projection plane

TWO POINT

- Two point perspective projection occurs when projection plane intersects two of principal axis.
- In the figure, x and z axis intersects projection plane whereas y axis remain parallel to projection plane
- The main use of Two Point projection is to draw the two corner roads. It is also used in architectural, industrial , engineering and advertisement drawing.

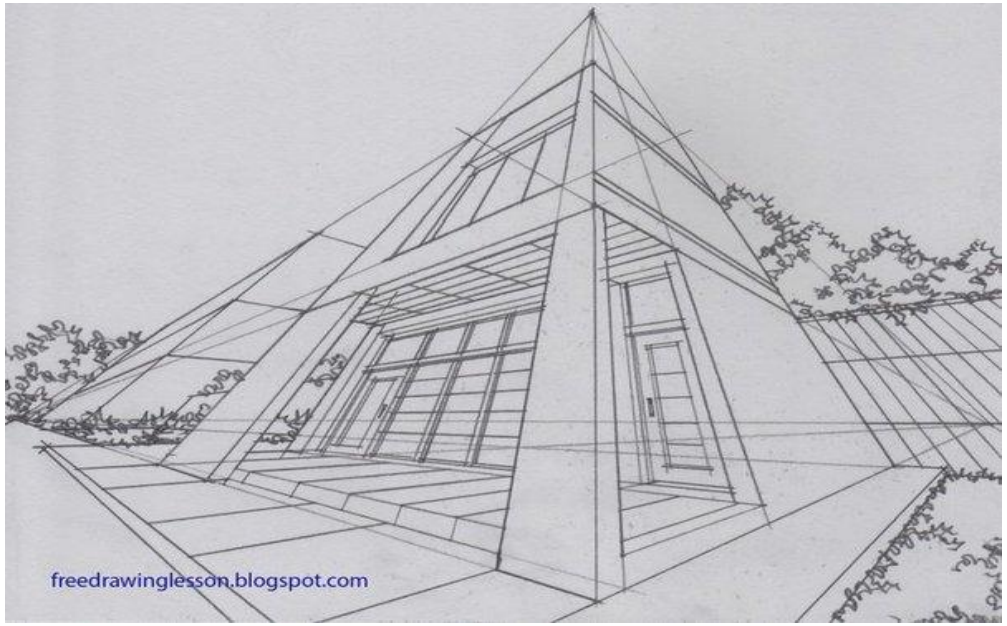


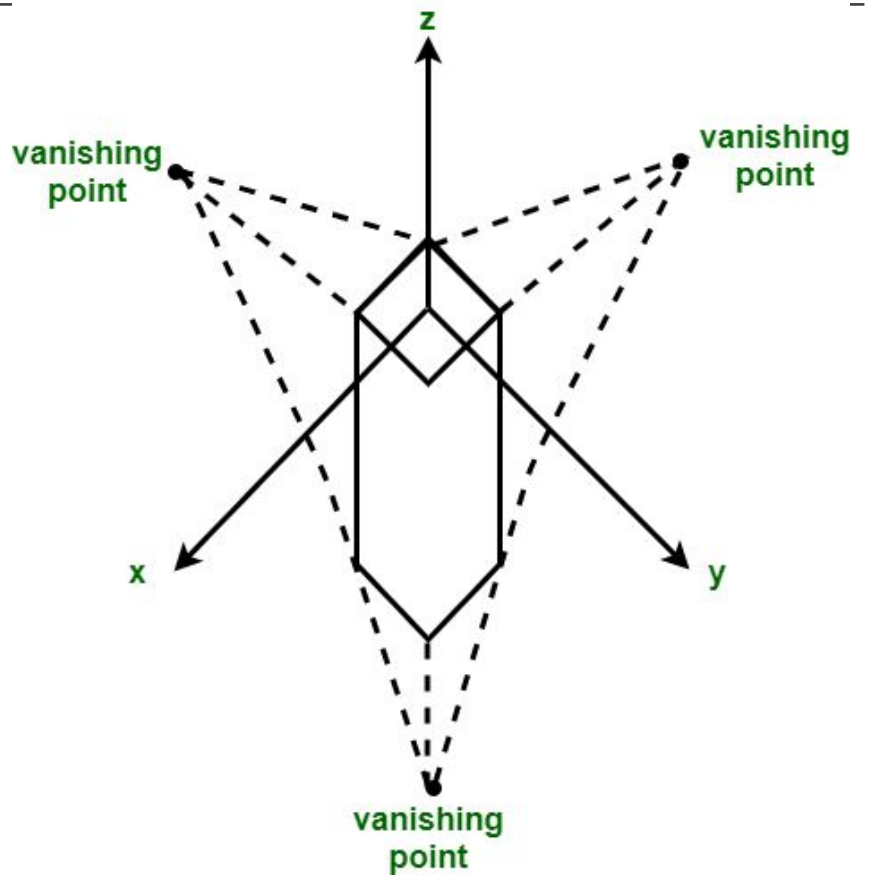
ONE POINT VS TWO POINT



THREE POINT

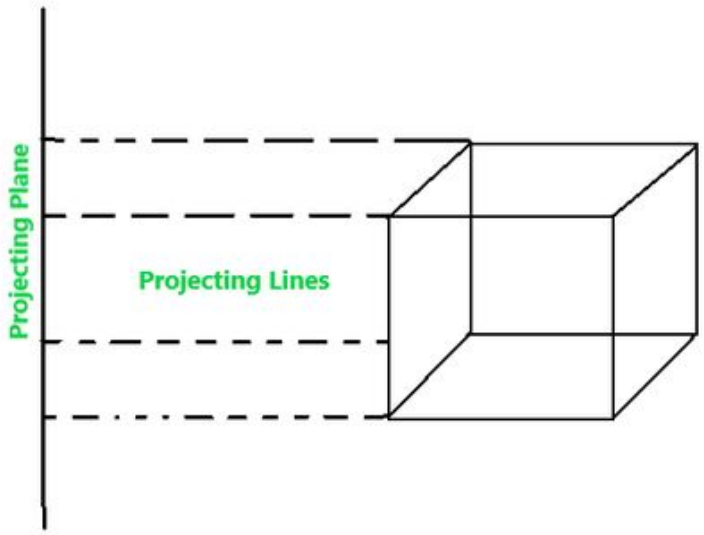
- Three point perspective projection occurs when all three axis intersects with projection plane. There is no any principle axis which is parallel to projection plane.





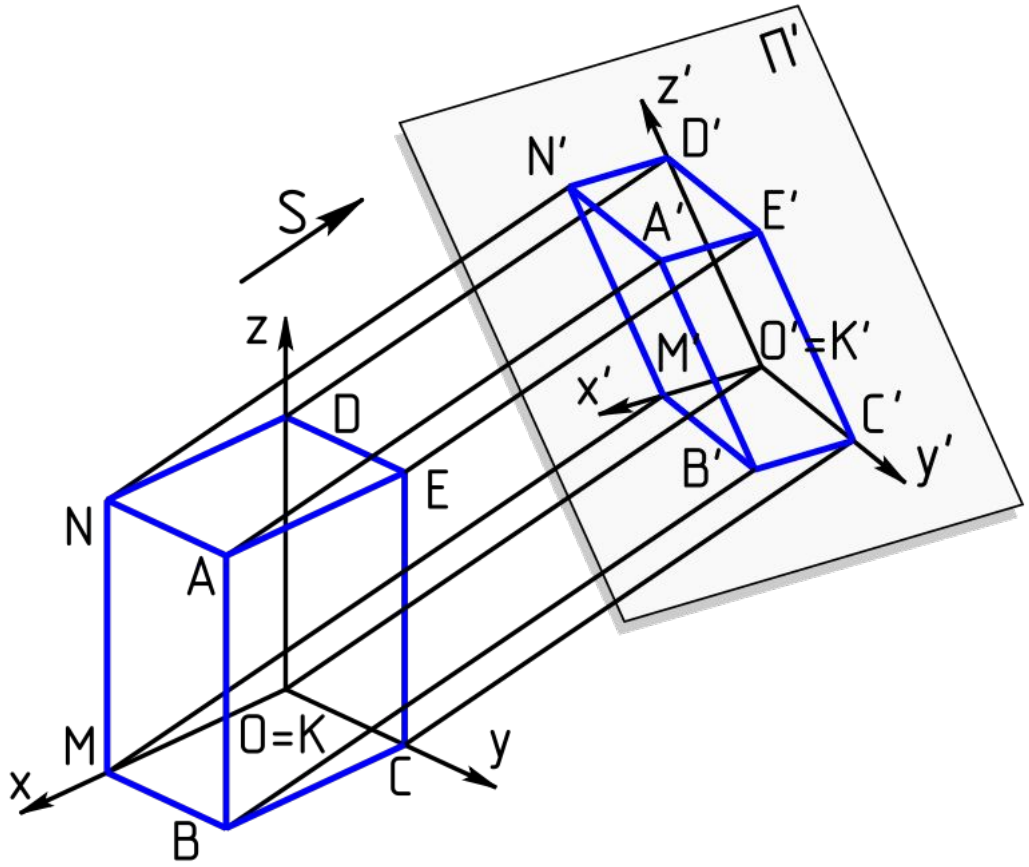
PARALLEL PROJECTION

- Parallel projection is a kind of projection where the projecting lines emerge parallelly from the polygon surface and then incident parallelly on the plane. In parallel projection, the centre of the projection lies at infinity. In parallel projection, the view of the object obtained at the plane is less-realistic as there is no for-shortcoming. and the relative dimension of the object remains preserves.



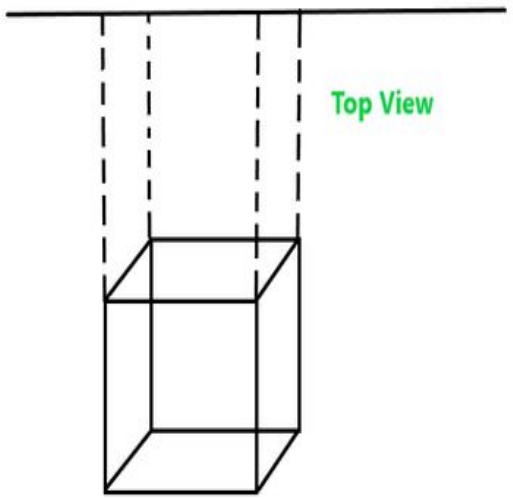
ORTHOGRAPHIC PROJECTION

- It is a kind of parallel projection where the projecting lines emerge parallelly from the object surface and incident perpendicularly at the projecting plane.
- Two Types
  1. Multi-View Projection
  2. Axonometric Projection

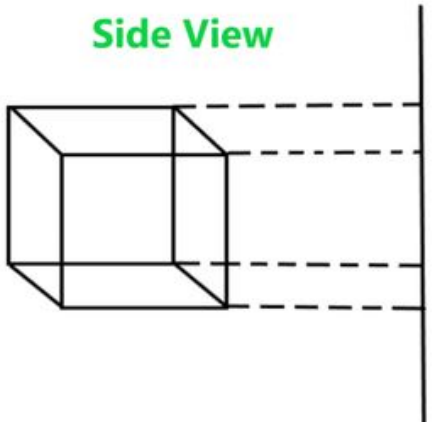


MULTI VIEW PROJECTION

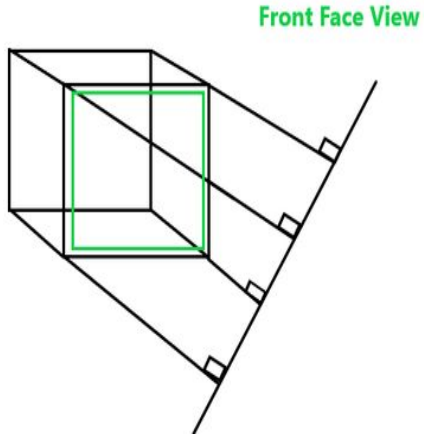
In this projection, the rays that emerge from the top of the polygon surface are observed.



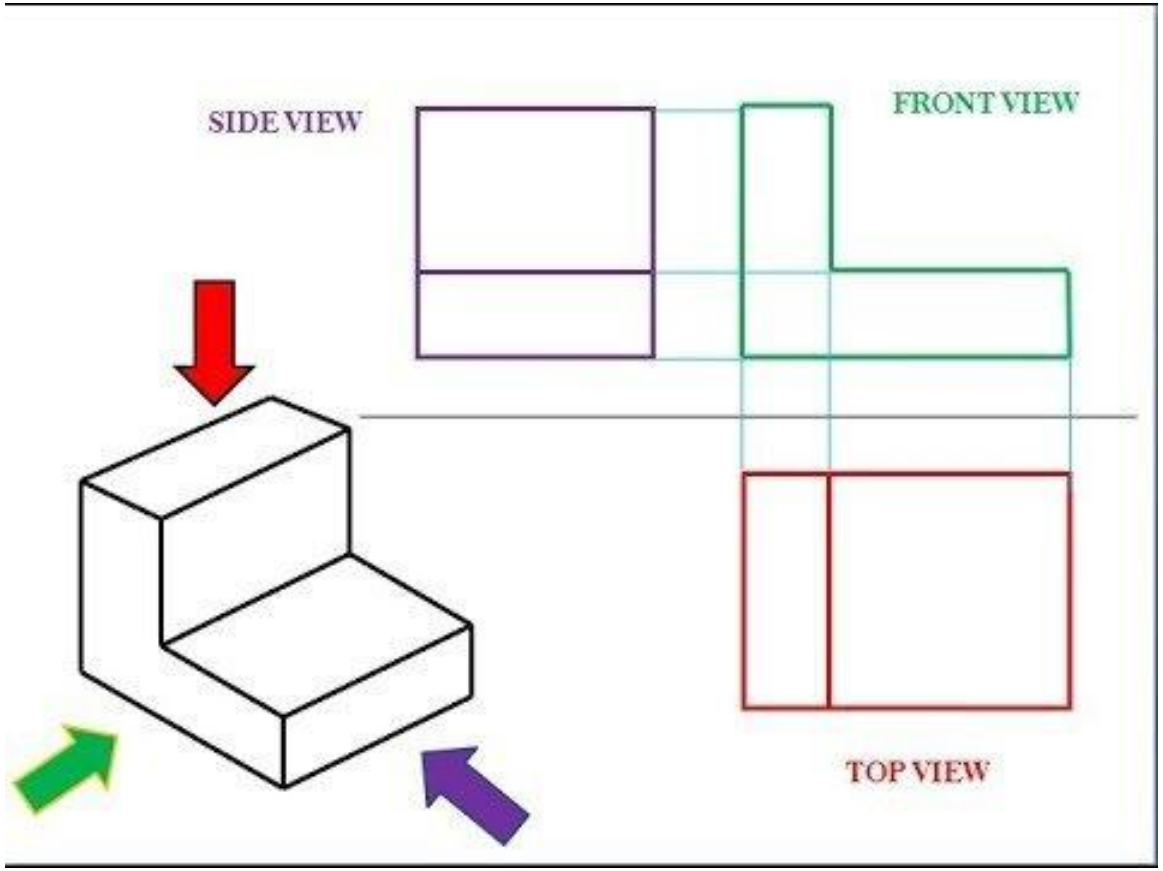
It is another type of projection orthographic projection where the side view of the polygon surface is observed.



In this orthographic projection front face view of the object is observed.

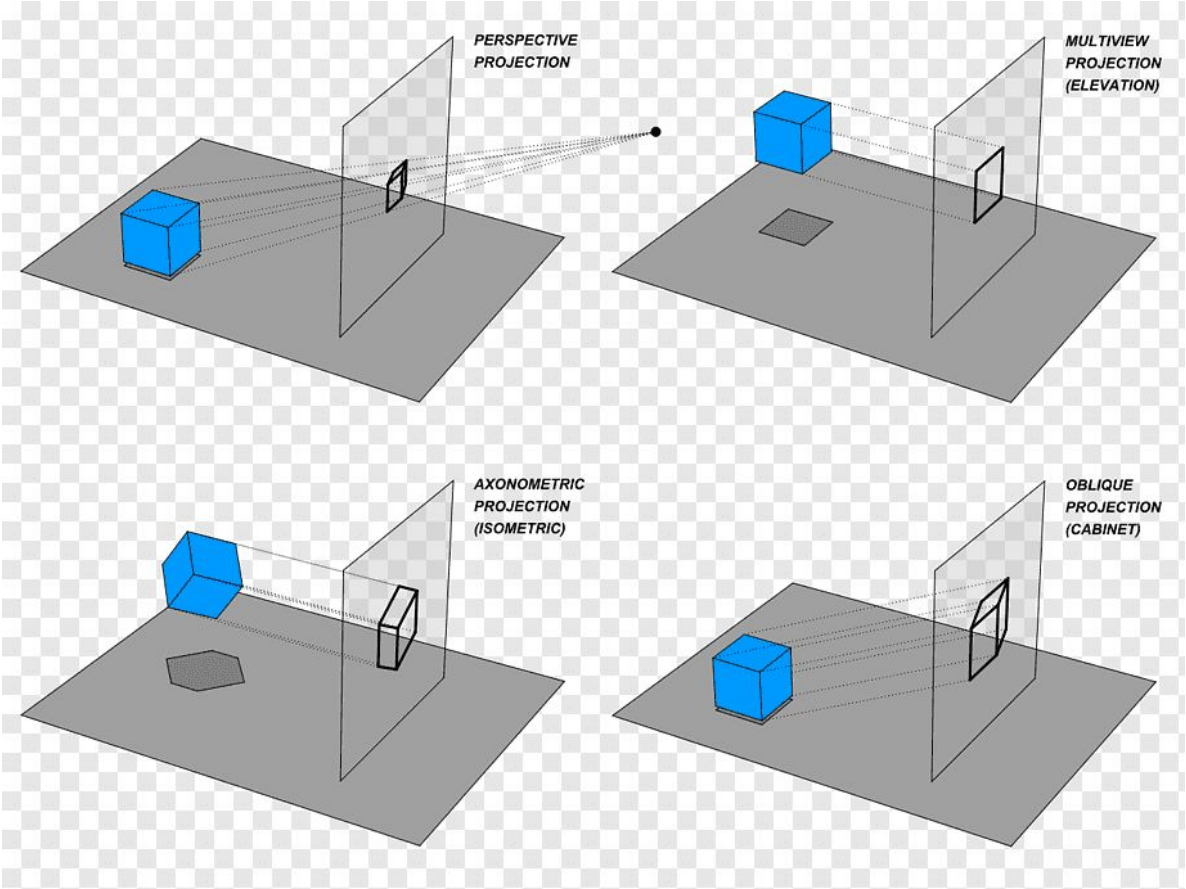


MULTI VIEW PROJECTION



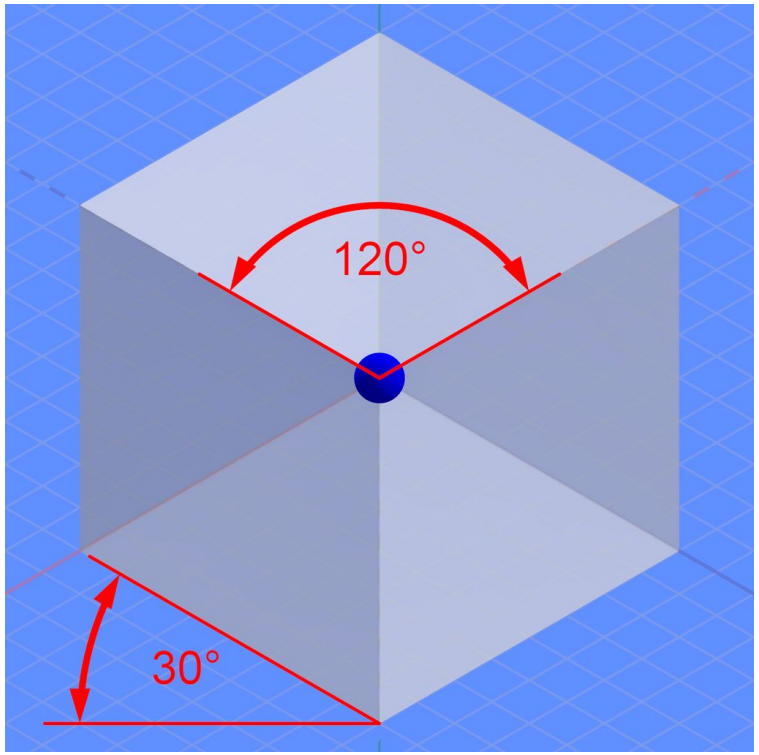
AXONOMETRIC PROJECTION

Axonometric projection is an orthographic projection, where the projection lines are perpendicular to the plane of projection, and axes to show multiple sides. the object is rotated around one or more of its axes to reveal multiple sides

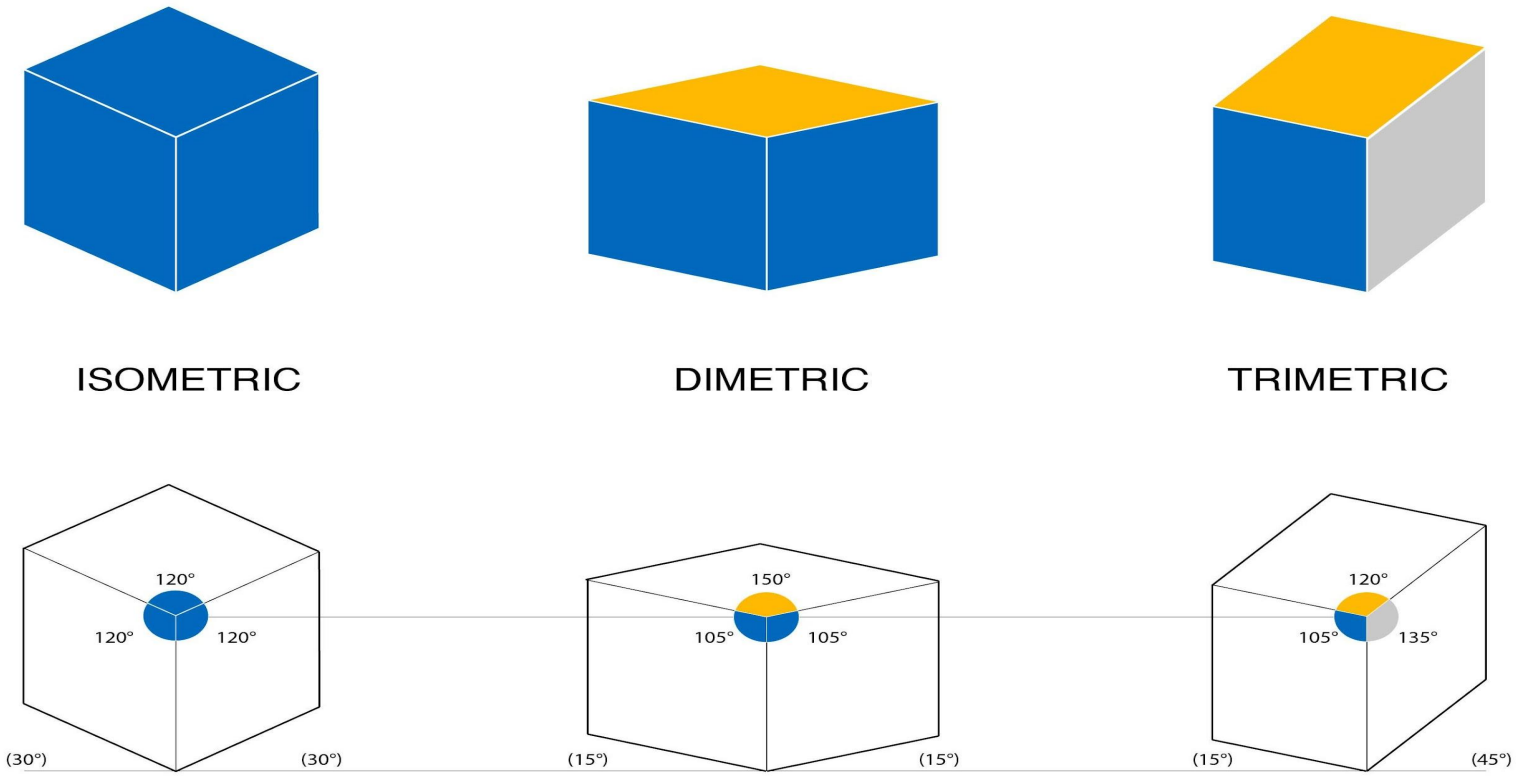


ISOMETRIC PROJECTION

Isometric projection is a method for visually representing three-dimensional objects in two dimensions in technical and engineering drawings. It is an axonometric projection in which the three coordinate axes appear equally foreshortened and the angle between any two of them is 120 degrees.



DIMETRIC & TRIMETRIC PROJECTION

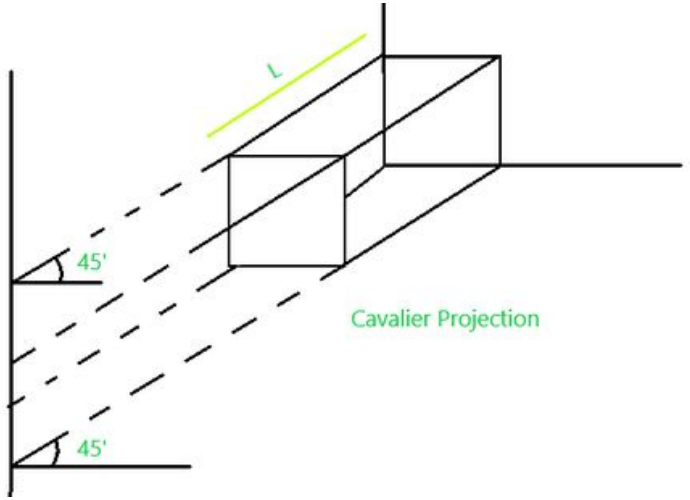


OBLIQUE PROJECTION

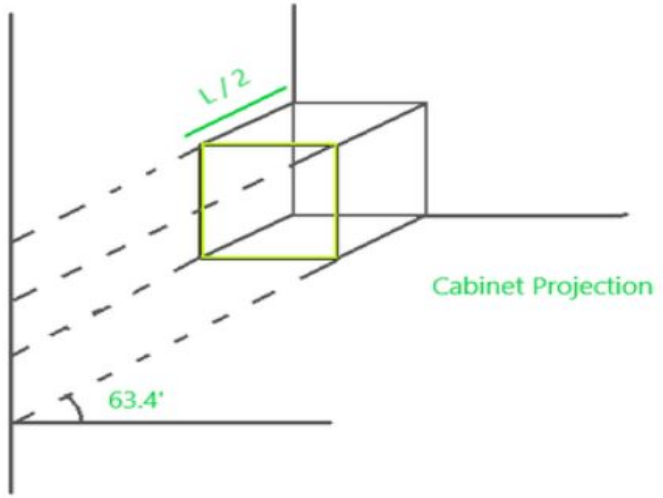
It is a kind of parallel projection where projecting rays emerges parallelly from the surface of the polygon and incident at an angle other than 90 degrees on the plane.

CAVALIER AND CABINET PROJECTION

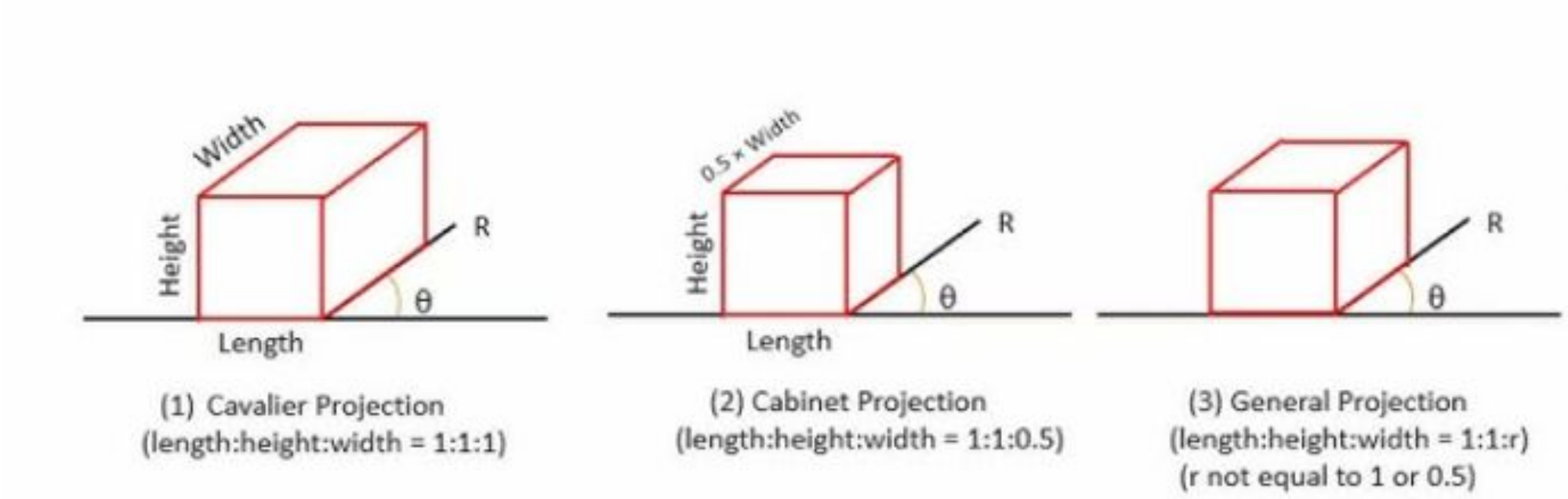
It is a kind of oblique projection where the projecting lines emerge parallelly from the object surface and incident at 45‘ rather than 90′ at the projecting plane. In this projection, the length of the reading axis is larger than the cabinet projection.



It is similar to that cavalier projection but here the length of reading axes just half than the cavalier projection and the incident angle at the projecting plane is 63.4′ rather 45′.



CAVALIER AND CABINET PROJECTION



(1) Cavalier Projection
(length:height:width = 1:1:1)

(2) Cabinet Projection
(length:height:width = 1:1:0.5)

(3) General Projection
(length:height:width = 1:1:r)
(r not equal to 1 or 0.5)

Mathematics of Planar Geometric Projection

Perspective Projection:

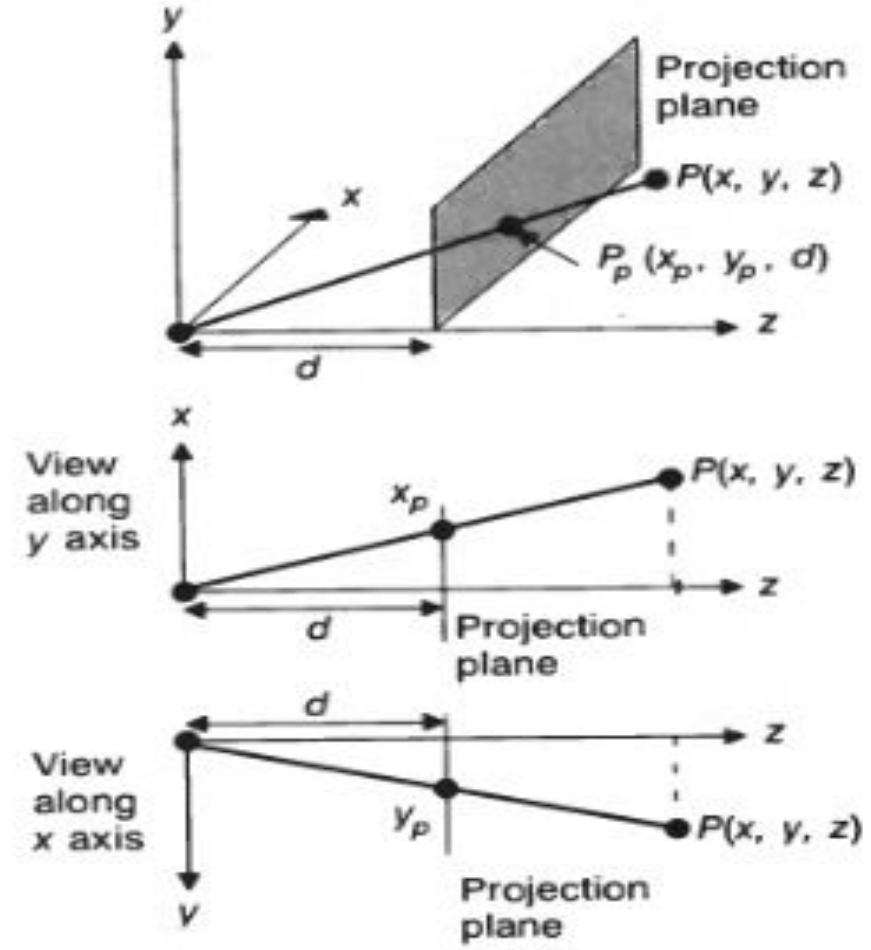
1. Projection plane is normal to the z axis at z=d.
2. COP is at origin.

$$\frac{x_p}{d} = \frac{x}{z}; \qquad \frac{y_p}{d} = \frac{y}{z}.$$

Multiplying each side by d yields

$$x_p = \frac{d \cdot x}{z} = \frac{x}{z/d}, \qquad y_p = \frac{d \cdot y}{z} = \frac{y}{z/d}.$$

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}.$$



Mathematics of Planar Geometric Projection

Perspective Projection:

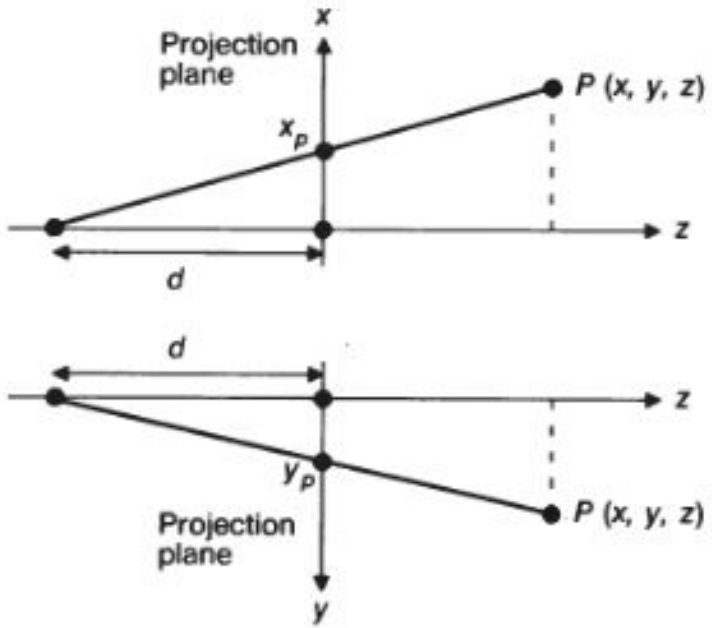
1. Projection plane is at z=0 and COP is at z=-d

$$\frac{x_p}{d} = \frac{x}{z+d}, \qquad \frac{y_p}{d} = \frac{y}{z+d}.$$

Multiplying by d, we get

$$x_p = \frac{d \cdot x}{z+d} = \frac{x}{(z/d)+1}, \qquad y_p = \frac{d \cdot y}{z+d} = \frac{y}{(z/d)+1}.$$

$$M'_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}.$$



Mathematics of Planar Geometric Projection

Orthographic Projection:

1. Projection plane is at z=0 and normal to z axis

$$x_p = x, \quad y_p = y, \quad z_p = 0.$$

$$M_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

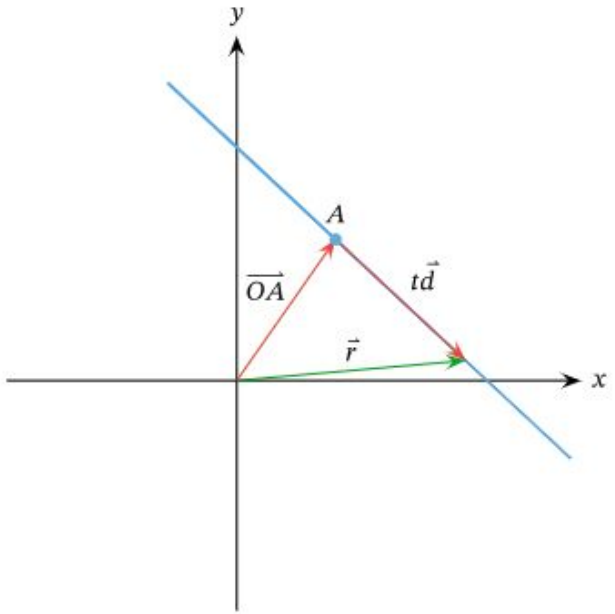
Generalized Planar Geometric Projection

- ✔ M_{per} applies only in the Perspective Projection.
- ✔ M_{ortho} applies only in orthographic projection.

$$\vec{r} = \overrightarrow{OA} + t\vec{d}.$$

line passing through the point $A(x_0, y_0)$

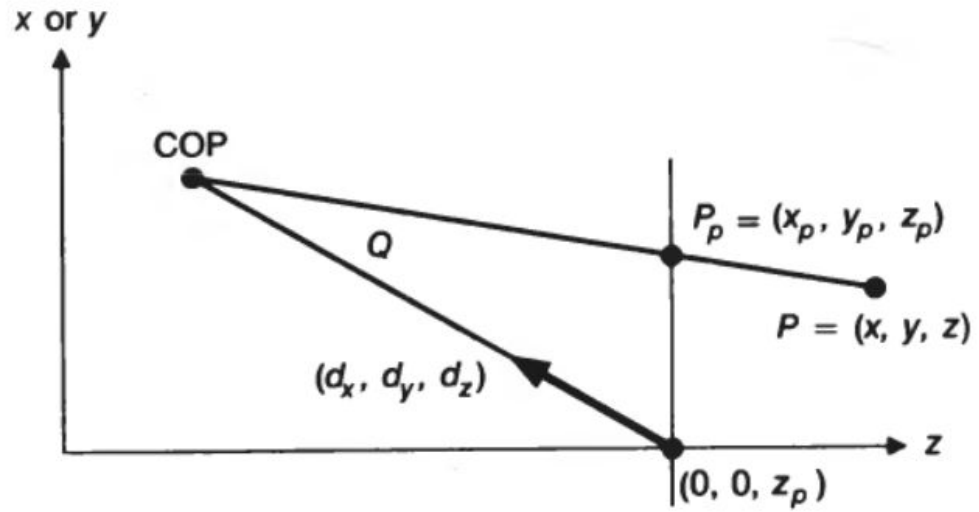
parallel to the direction vector $\vec{d} = (a, b)$.



Generalized Projection Matrix

- ✓ In the figure , the projection of the general point $p = (x, y, z)$ onto the projection plane is $P_p = (x_p, y_p, z_p)$
- ✓ The projection plane is perpendicular to the z axis at a distance Z_p from the origin
- ✓ The center of projection (COP) is a distance Q from the point (0, 0, Z_p,).
- ✓ The direction from (0, 0, Z_p) to COP is given by the normalized direction vector (d_x, d_y, d_z,).
- ✓ P_p is on the line between COP and P, which can be specified parametrically as

$$= COP + t(P - COP) \quad - (1) \qquad where \quad 0 < t < 1$$



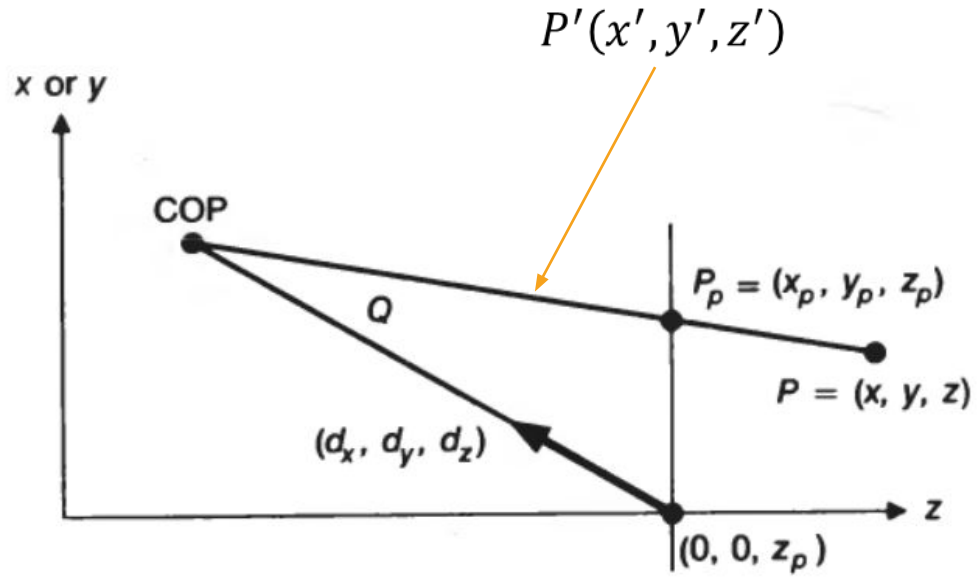
Generalized Projection Matrix

$$COP + t(P - COP) \quad - \ (1)$$

$$COP = (0,0,Z_p) + Q(dx, dy, dz)$$

Rewriting the equation (1) as separate equation for the arbitrary point $P'(x', y', z')$ on the line with

$$x' = Q\,d_x + (x - Q\,d_x)t,$$
$$y' = Q\,d_y + (y - Q\,d_y)t,$$
$$z' = (z_p + Q\,d_z) + (z - (z_p + Q\,d_z))t.$$

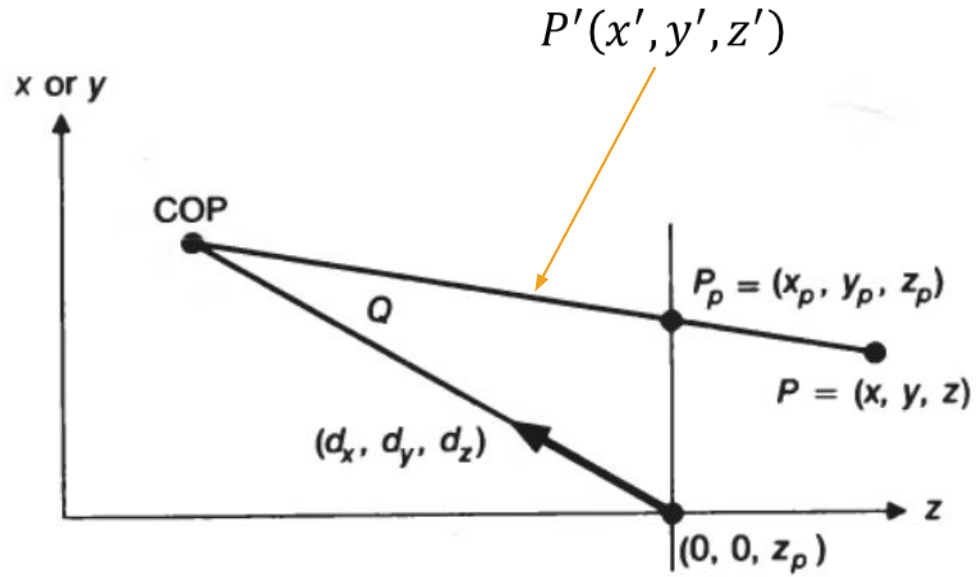


Generalized Projection Matrix

$$x' = Q\,d_x + (x - Q\,d_x)t,$$
$$y' = Q\,d_y + (y - Q\,d_y)t,$$
$$z' = (z_p + Q\,d_z) + (z - (z_p + Q\,d_z))t.$$

we find the projection P_p *of the point* P, at the intersect *of the line between COP and P. by substituing* $z = Z_p$ and solving for t:

$$t = \frac{z_p - (z_p + Q\,d_z)}{z - (z_p + Q\,d_z)}.$$



Generalized Projection Matrix

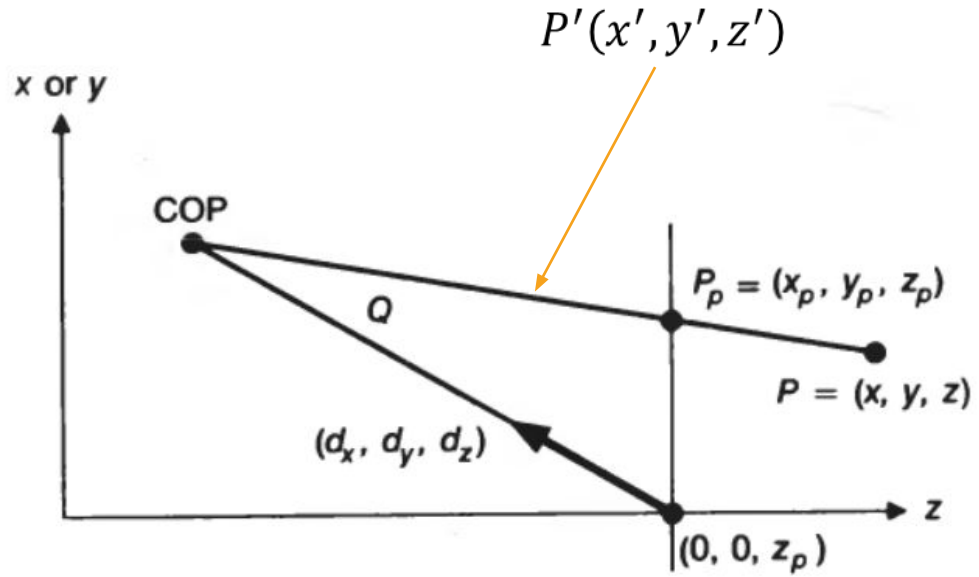
$$x' = Q\,d_x + (x - Q\,d_x)t,$$
$$y' = Q\,d_y + (y - Q\,d_y)t,$$
$$z' = (z_p + Q\,d_z) + (z - (z_p + Q\,d_z))t.$$

Substituting the value for t and setting $x'=x_p$ and $y'=y_p$ yields

$$x_p = \frac{x - z\frac{d_x}{d_z} + z_p\frac{d_x}{d_z}}{\frac{z_p - z}{Q\,d_z} + 1},$$

$$y_p = \frac{y - z\frac{d_y}{d_z} + z_p\frac{d_y}{d_z}}{\frac{z_p - z}{Q\,d_z} + 1}.$$

$$z_p = z_p\frac{\frac{z_p - z}{Q\,d_z} + 1}{\frac{z_p - z}{Q\,d_z} + 1}$$



Generalized Projection Matrix

$$M_{\text{general}} = \begin{bmatrix} 1 & 0 & -\frac{d_x}{d_z} & z_p\frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} & z_p\frac{d_y}{d_z} \\ 0 & 0 & -\frac{z_p}{Q\,d_z} & \frac{z_p^2}{Q\,d_z} + z_p \\ 0 & 0 & -\frac{1}{Q\,d_z} & \frac{z_p}{Q\,d_z} + 1 \end{bmatrix}.$$

	z_p	Q	$[d_x$	d_y	$d_z]$
M_{ort}	0	∞	[0	0	−1]
M_{per}	d	d	[0	0	−1]
M'_{per}	0	d	[0	0	−1]

Projection Math

A unit cube is projected onto XY plane. Find the coordinates of the projected cube

1. if the projected plane is at z=0 and the center of projection is at z=-10