Prof. Supriya. S

Projections.

3D to 2D projection:

- Parallel Projections
  - Orthographic
  - Oblique
- Perspective

Projection of a 3D object is defined by straight projection rays (projectors) emanating from the center of projection (COP) passing through each point of the object and intersecting the projection plane.

Perspective Projections.

Distant of from COP to projection plane is finite. The projections are not parallel & we specify a center of projection (COP).

Center of projection is also called Perspective reference point.

Perspective foreshortening:

The size of the perspective projection of an object is inversely proportional to / varies inversely with the distances of object from the center of projection.

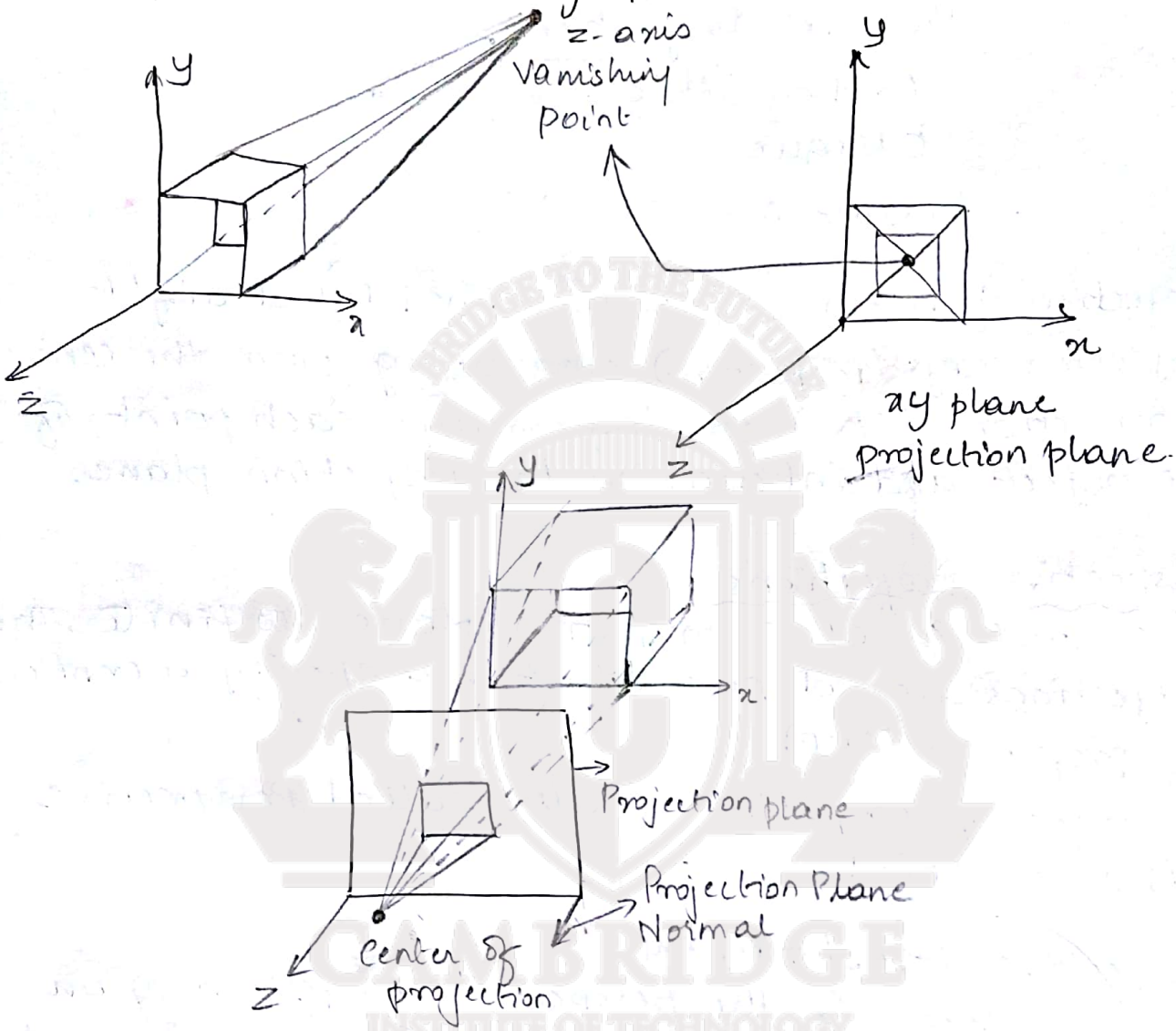
object far → looks smaller
object close → looks larger.

Vanishing point: A railway tracle (rail lines), when you see a railway tracle from a fixed point on the tracle, as we observe the tracle to larger distances they observe to meet (converge) at a point even though we know the tracle lines are rail. Such converging point is known as Vanishing point.

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Def: The perspective projections of any set of wc lines that are not parallel to the projection plane converge to a vanishing point.



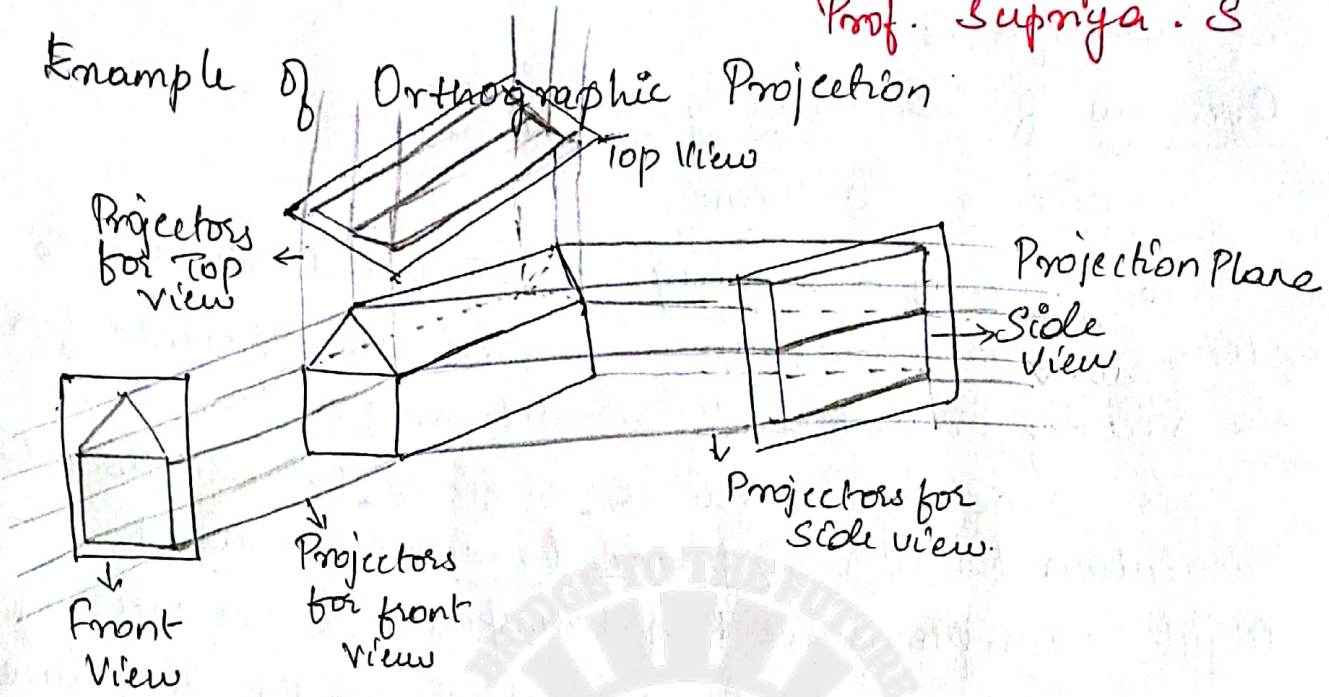
Prespective projection

BRIDGE TO THE FUTURE
CAMBRIDGE
INSTITUTE OF TECHNOLOGY
(SOURCE DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya. S

Example of Orthographic Projection



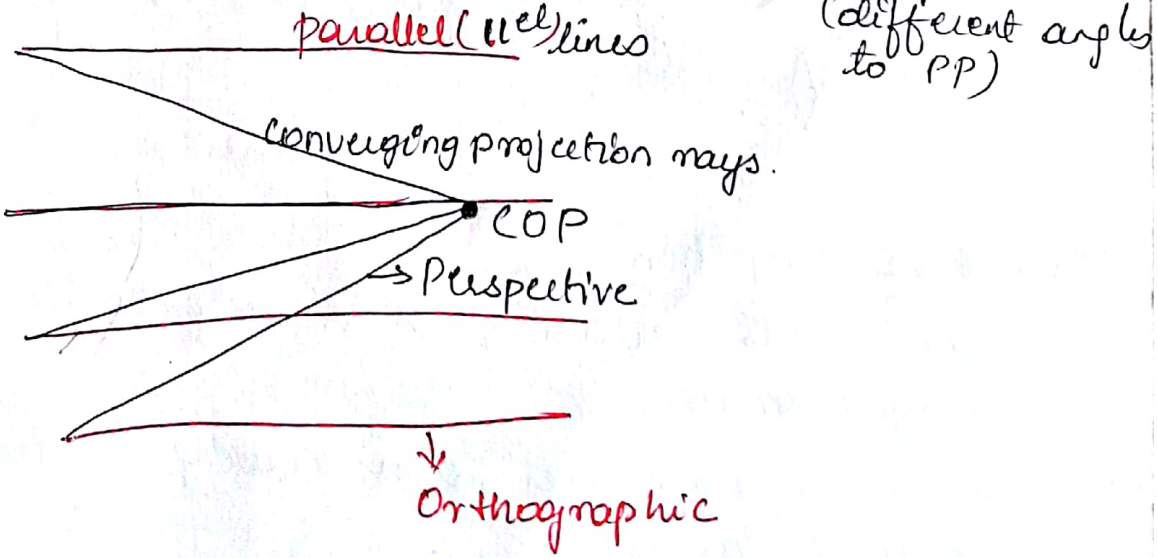
Axonometric Orthographic projection.

Use planes of projection that are not normal to a principal axis. (show multiple face of an object).

Isometric: Projection plane normal makes equal angles with each principle axis. DOP vector : [1 1 1]

Difference between Orthographic & Perspective.

Orthographic is infinite (⊥lar angle to projection plane)
Perspective is finite, COP is at a fixed point.



Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Overview of 3D viewing concepts

Viewing a 3D scene.

To obtain a display of 3D world-coordinate scene, we first set up a co-ordinate reference for the viewing or "camera" parameters.

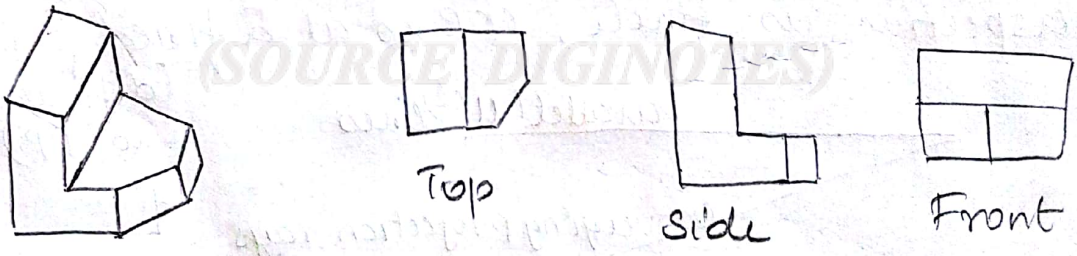
This co-ordinate reference defines the position & orientation for a view plane (projection plane)

Object descriptions are then transfered to the viewing reference co-ordinates & projected on to the view plane.

- View of an object on the output device is like a wireframe (outline), later lighting, surface rendering are applied to obtain realistic shading of visible surface.

Projections.

Parallel projection - project the points on the object surface along parallel lines onto a view plane.



Perspective projection -

causes objects farther from viewing position to be displayed smaller.

causes objects nearer from viewing position to be displayed larger

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Depth cueing

- depth information is important in 3D scene - for a viewing direction, which is front & which is back of each displayed object.
- A simple method to indicate depth in wire frame ~~mode~~ display is to vary the brightness of line segments based on their distances from the viewing position. Line closest to the viewing position are displayed with highest intensity, lines father away are displayed with decreasing intensity.
- Another method of depth cuing is modelling the effect of the atmosphere on the perceived intensity (color) of objects. distant objects - appear dimmer than near objects due to light scattering by dust particles, haze & smoke.

Identifying Visible lines and surfaces.

can clarify depth information/relationships using.

(1) highlight the visible lines or to display them in a different color.

(2) display non-visible lines as dashed lines. or remove the non-visible lines from display. (removes info about shape of back surfaces).

when realistic scene is to be produced → back parts of objects are completely eliminated so that only visible surfaces are displayed.

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Surface Rendering.

Rendering the object surfaces using lighting conditions in scene & assigned surface characteristics.

Surface → transparent or opaque.
↳ smooth or rough

Surface rendering is combined with perspective & visible surface identification to generate a degree of realism.

Lighting conditions → by specifying color, location of light source, & background illumination effects.

Exploded & Cutaway Views.

→ used to show the internal structures & relationships of object parts.

Refer figures from textbook.

Stereoscopic Viewing.

3D views can be obtained by reflecting a raster image from a vibrating, flexible mirror.

Stereoscopic devices present 2 views of scene, one for the left eye & other for right eye. The viewing positions correspond to the eye position of the viewer. It is displayed on alternate refresh cycles of raster monitor.

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

The Three-Dimensional Viewing Pipeline

- Viewing position corresponds to where we would place a camera.
- Choose viewing position according to what we want to display (front, back, side, top, bottom view of the scene)
- Orientation of the camera

Some of the viewing operations for a 3D scene are same as in 2D viewing pipeline as described in the below figure.

* 2D clipping window is used to select a view that is to be mapped to the viewport.
* 2D viewport is used to position a projected view of 3D scene on output device.

In 3D viewing, a 3D clipping window is positioned on a selected view plane & scenes are clipped against an enclosing volume of space defined as set of clipping planes.

The viewing position, view plane, clipping window & clipping planes are all specified within the viewing-co-ordinate reference frame.

(1) Once the scene has been modeled in world co-ordinates, a viewing co-ordinate system is selected & the description of scene is converted to viewing co-ordinates which defines viewing parameters including the position & orientation of the projection plane (view plane) – a camera film plane.

(2) A 2D clipping window, corresponding to a selected region as to camera lens, is defined on the

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

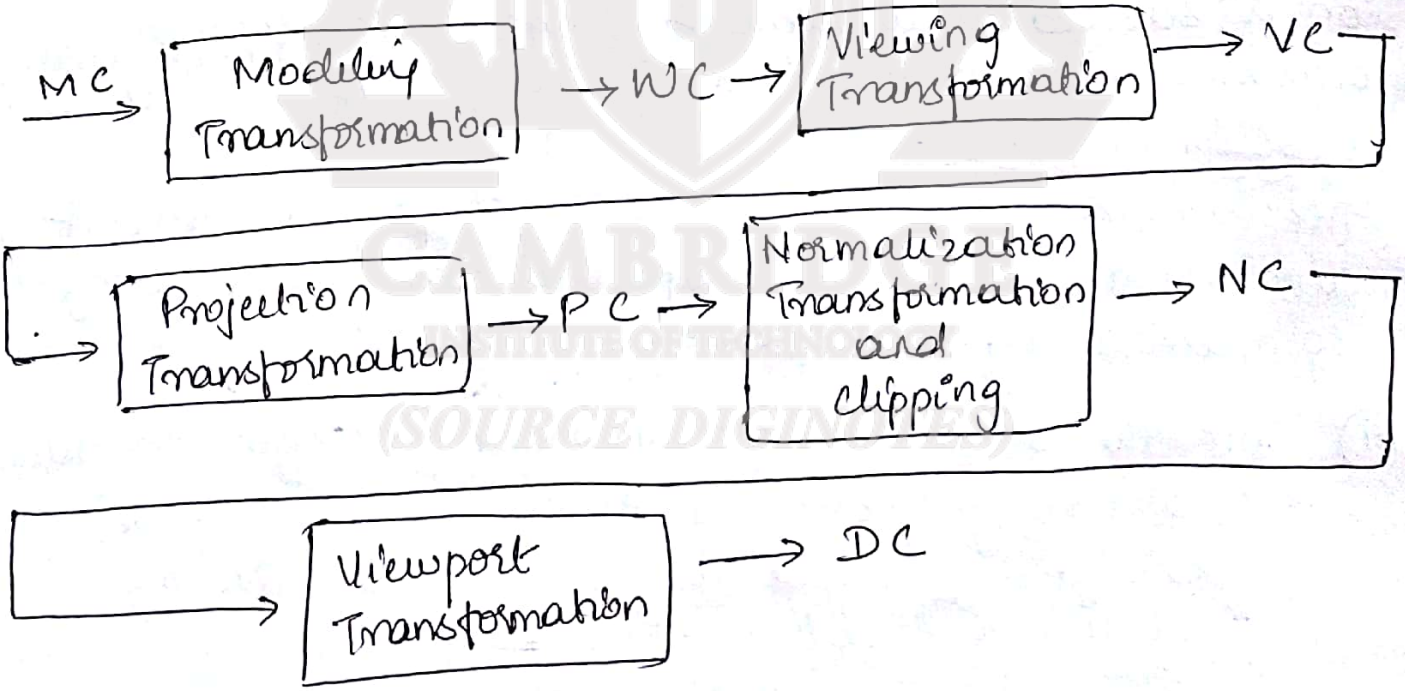
projection plane & 3D clipping region is established This clipping region is called View volume (its shape & size depends on dimensions of clipping window, type of projections ~~operation~~, direction of viewing)

(3) Projection operations are performed to convert viewing co-ordinate discription of scene to co-ordinate positions on projection plane.

(4) Objects are mapped to normalized co-ordinates & Object parts outside the view volume are clipped off.

(5) finally to viewport transformations (other tasks such as identifying visible surfaces, apply surface-rendering)

(6) Final step is to map viewport coordinates to device co-ordinates within a selected display window.



3D Transformation Pipeline

Source: diginotes.in Go paperless. Save the Earth.

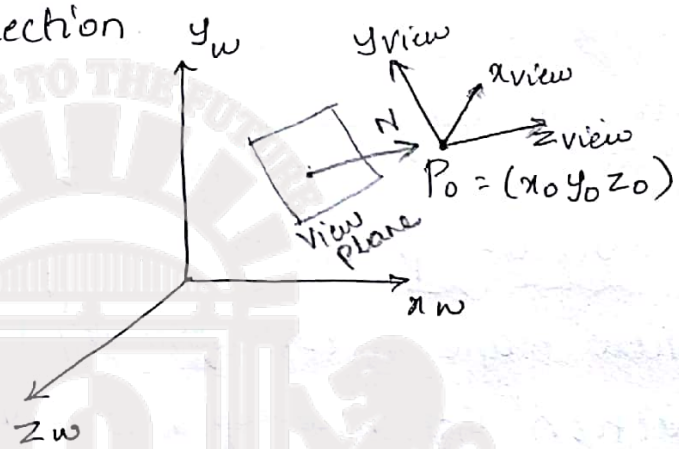
Prof. Supriya S

Three-Dimensional Viewing-coordinate Parameters

Select a world co-ordinate position $P_0 = (x_0 y_0 z_0)$ for a viewing origin, which is called the view point or viewing position (eye position or camera position)

A view-up vector 'V' which defines y_{view} direction.

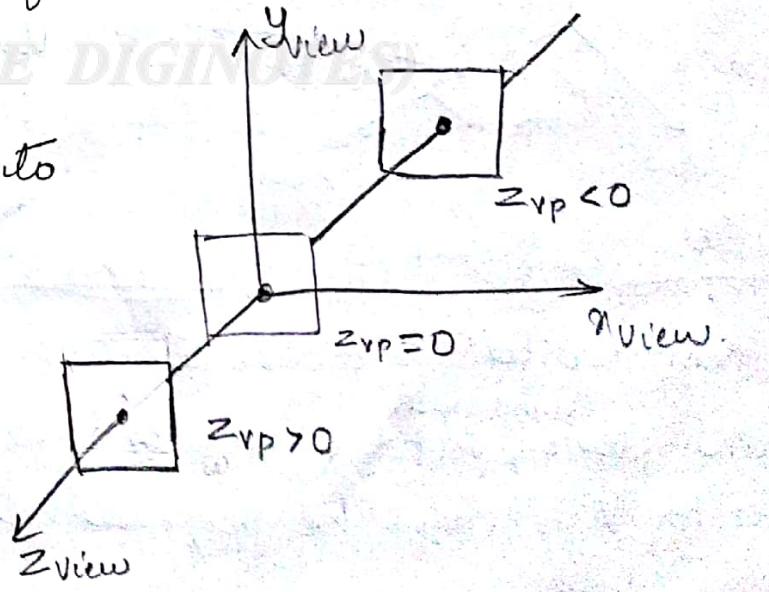
z_{view} → viewing direction



The View-Plane Normal Vector

* Viewing direction is along z_{view} axis, the view plane or projection plane is normally assumed to be perpendicular to z_{view} axis
* Orientation of view plane as well as direction of +ve z_{view} can be defined with a view-plane normal vector N

Viewplane is parallel to x_{view} & y_{view}



Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Vector N can be specified in various ways

① the direction of N is defined as the line from world-coordinate to selected point position.

② or direction from a reference point P_{ref} to viewing origin P_0.

Reference point P_{ref} is referred to as look-at-point with viewing direction opposite to direction of N. See figure 3 in below.

The View-up vector.

view plane vector Normal 'N' is choosen, the direction of view-up vector 'V' can also be set in the positive direction of y-axis (where the 'up' of the camera is)

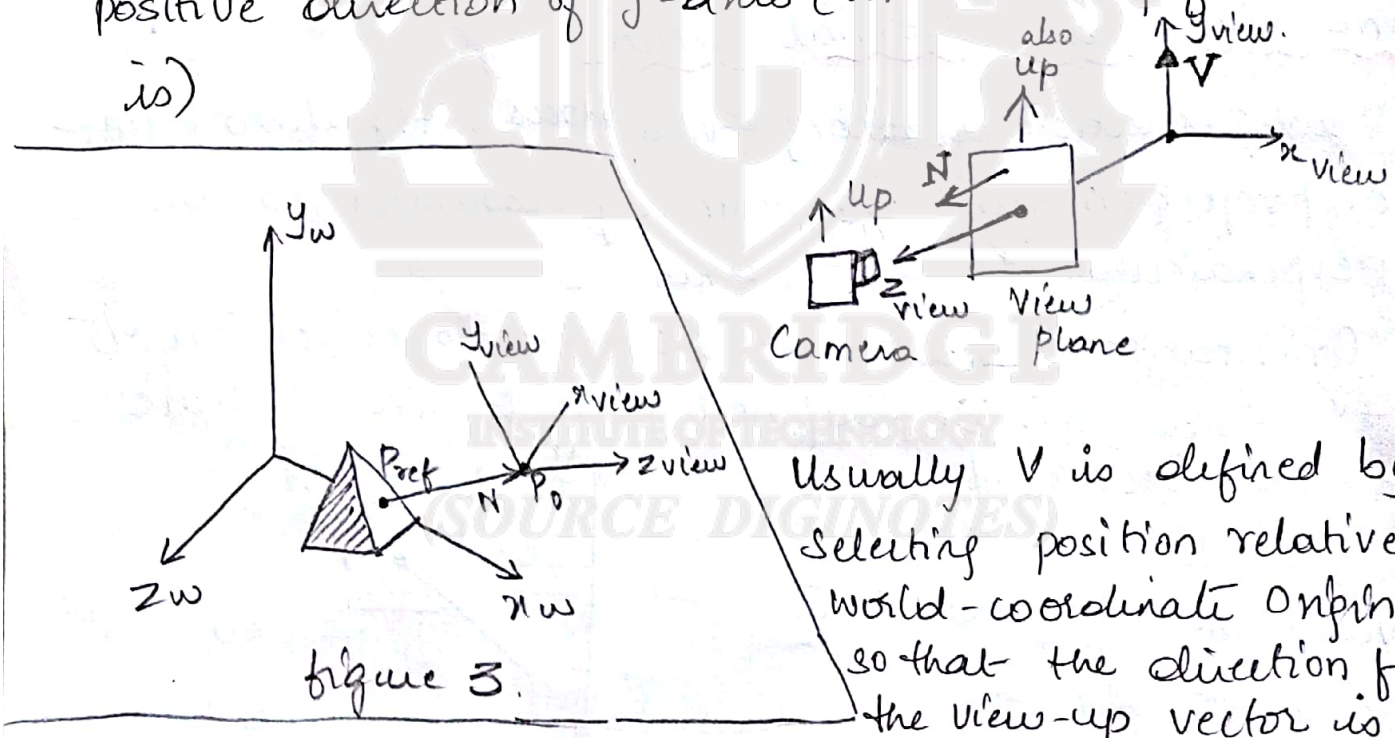


figure 3.

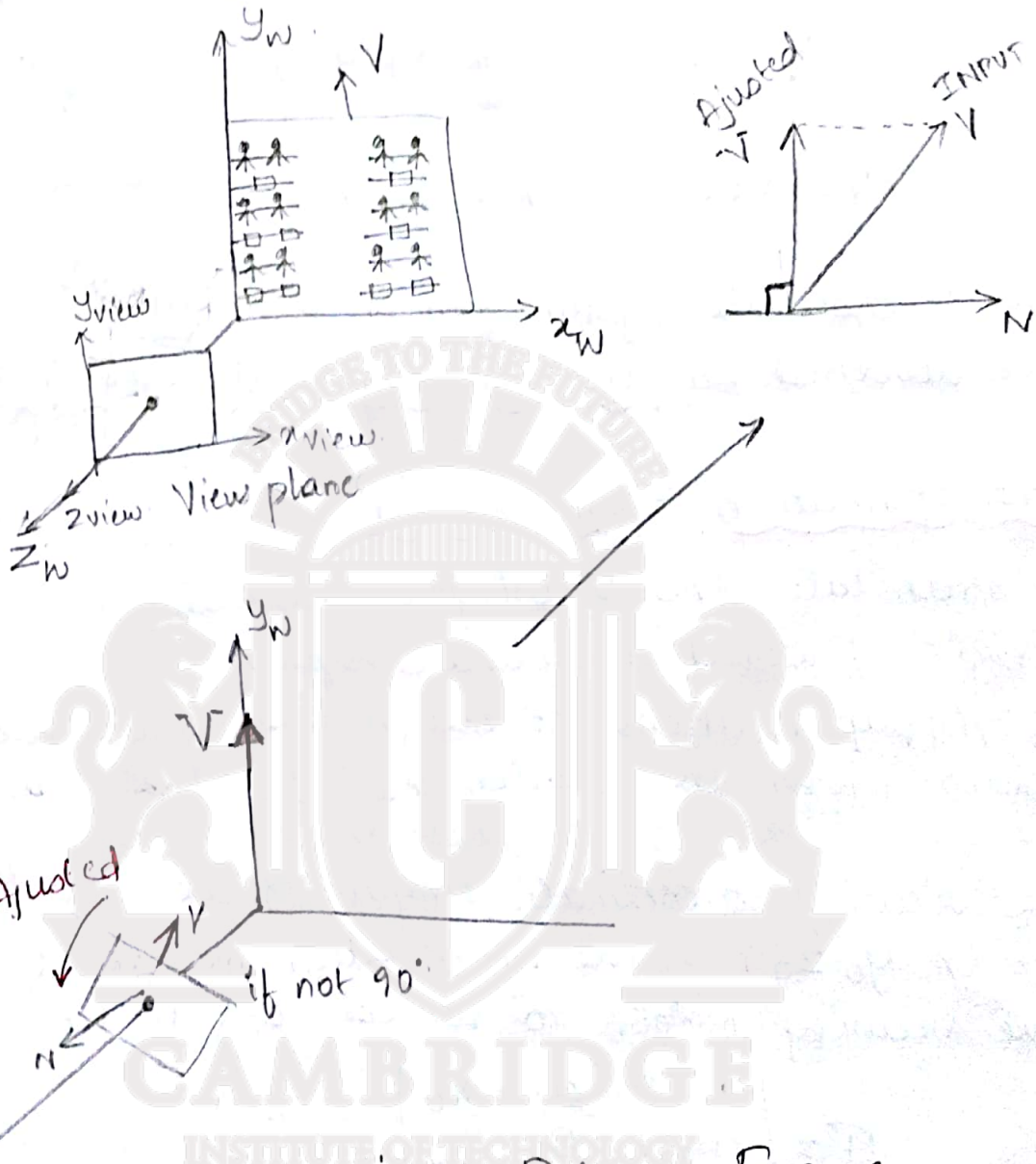
Usually V is defined by selecting position relative to world-coordinate origin. so that the direction for the view-up vector is from world co-ordinate origin to this selected position

View-plane normal vector V defines the direction of z_{view}-axis, vector V should be perpendicular to N.

Source: diginotes.in Go paperless. Save the Earth.

Therefore viewing routines typically adjust the user-defined orientation of vector V.

Ex:



The uvn Viewing-coordinate Reference Frame.

View plane normal vector defines the direction for zview, & view up vector is used to obtain the direction for yview, we need only to determine the direction for xview (U vector)

U vector is perpendicular to both N & V.

- taking the cross product of N and V.

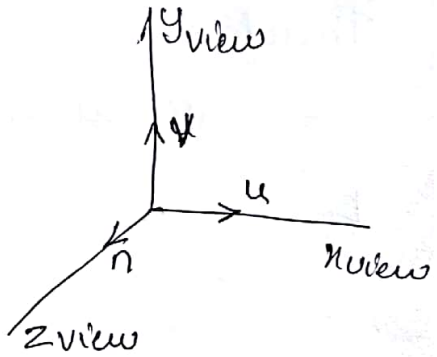
CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE: DIGINOTES)

Prof. Supriya S

$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

$$u = \frac{V \times n}{|V|} = (u_x, u_y, u_z)$$

$$v = n \times u = (v_x, v_y, v_z)$$



The co-ordinate system formed with these unit vectors are described as uvn viewing-coordinate reference frame

Transformation from World to Viewing Coordinates

1. Translate the viewing-coordinate origin to the origin of world co-ordinate system.
2. Apply rotations to align the x_{view}, y_{view} & z_{view} axes with the world x_w, y_w & z_w axes respectively

The viewing co-ordinate origin is at world position $P = (x_0\ y_0\ z_0)$ ∴ The translation matrix translating the viewing origin to world origin is

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For rotation transformation, the unit vectors u, v & n are used to form composite rotation matrix.

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Source: diginotes.in Go paperless. Save the Earth.

$$M_{WC,VC} = R.T$$

$$= \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} u_x & u_y & u_z & -x_0 u_x - y_0 u_y - z_0 u_z \\ v_x & v_y & v_z & -x_0 v_x - y_0 v_y - z_0 v_z \\ n_x & n_y & n_z & -x_0 n_x - n_y y_0 - z_0 n_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} u_x & u_y & u_z & -u \cdot P_0 \\ v_x & v_y & v_z & -v \cdot P_0 \\ n_x & n_y & n_z & -n \cdot P_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$-u \cdot P_0 = -x_0 u_x - y_0 u_y - z_0 u_z$$
$$-v \cdot P_0 = -x_0 v_x - y_0 v_y - z_0 v_z$$
$$-n \cdot P_0 = -x_0 n_x - y_0 y_n - z_0 n_z$$

CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE: DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Projection Transformations.

① Orthogonal Projections

Distance from COP to projection plane is infinite – Parallel projection.

* A transformation of object descriptions to a view plane along lines that are all parallel to the view plane normal vector N is called orthogonal projection
* Projection lines are perpendicular to view plane.
* commonly used to produce the front, side & top views of an object.
* front, side & rear orthogonal projections – called Elevators.
* Top orthogonal projection is called plan view.

Refer figure from text book page no : 357 – 3rd edition

Axonometric & Isometric Orthogonal Projections

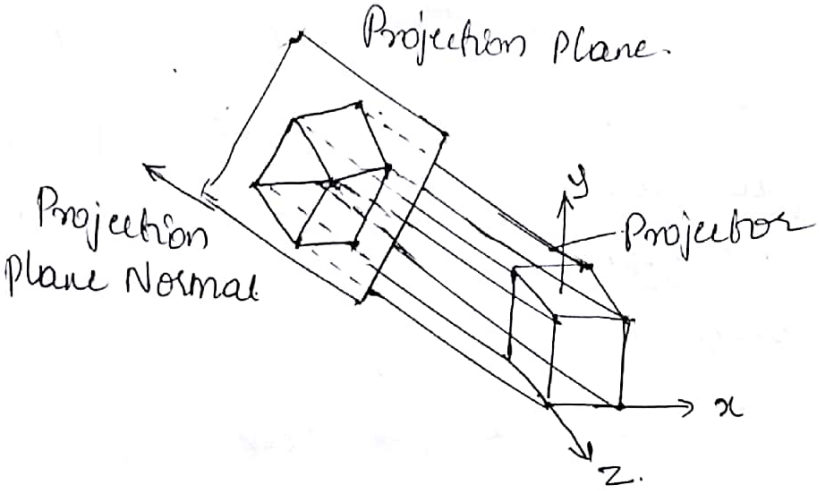
An Orthogonal projections formed that displays more than one face of an object are called Axonometric orthogonal projections.

The most commonly used axonometric projection is the isometric projection.

Axonometric projections use planes of projection that are not normal to a principle axis (they therefore show multiple face of an object)

Isometric projection : projection plane normal makes equal angles with each principle axis

Source: diginotes.in Go paperless. Save the Earth.



Example of Isometric projection.

All 3 principal axes are foreshortened equally in an isometric projection, so that relative ~~projection~~ proportions are maintained.

Clipping Window and Orthogonal - Projection View Volume

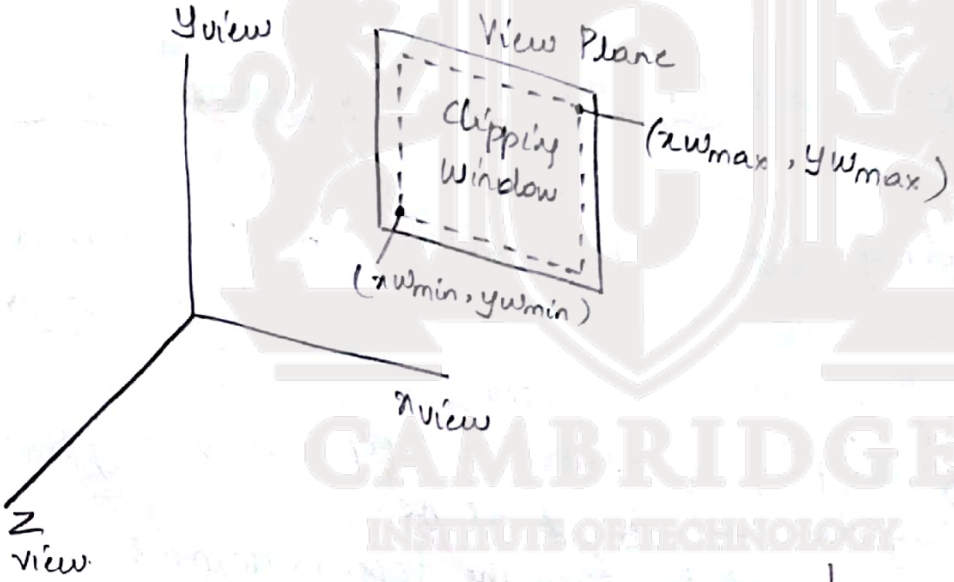
* For 3D viewing, the clipping window is positioned on the view plane with its edges parallel to x_{view} & y_{view} axis.
* The edges of clipping window specify the x & y limits - that display the part of the scene.
* These limits are used to form the top, bottom, & 2 sides of clipping region called orthogonal-projection view volume.
* since The projection lines are perpendicular to the view, these four boundaries are planes that are also perpendicular to the viewplane.
* The extent of Orthogonal view volume is limited in the z_{view} direction by selecting positions for one or two additional boundary planes that are

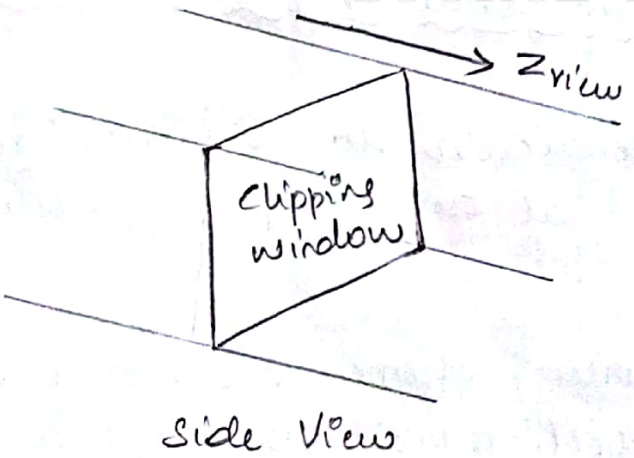
Source: diginotes.in Go paperless. Save the Earth.

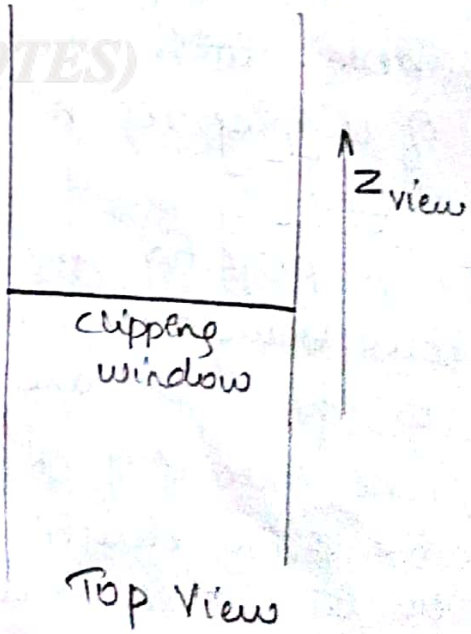
Prof. Supriya S

parallel to view plane. These two planes are called near-far clipping planes or front-back clipping planes.

* The near and far planes allow us to exclude objects that are in front of or behind the part of scene that we want to display.
* $z_{far} < z_{near}$ so that the far plane is farther out along the negative z_{view} axis.
* When a near and far planes are specified, we obtain a finite orthogonal view volume which is rectangular parallelepiped.

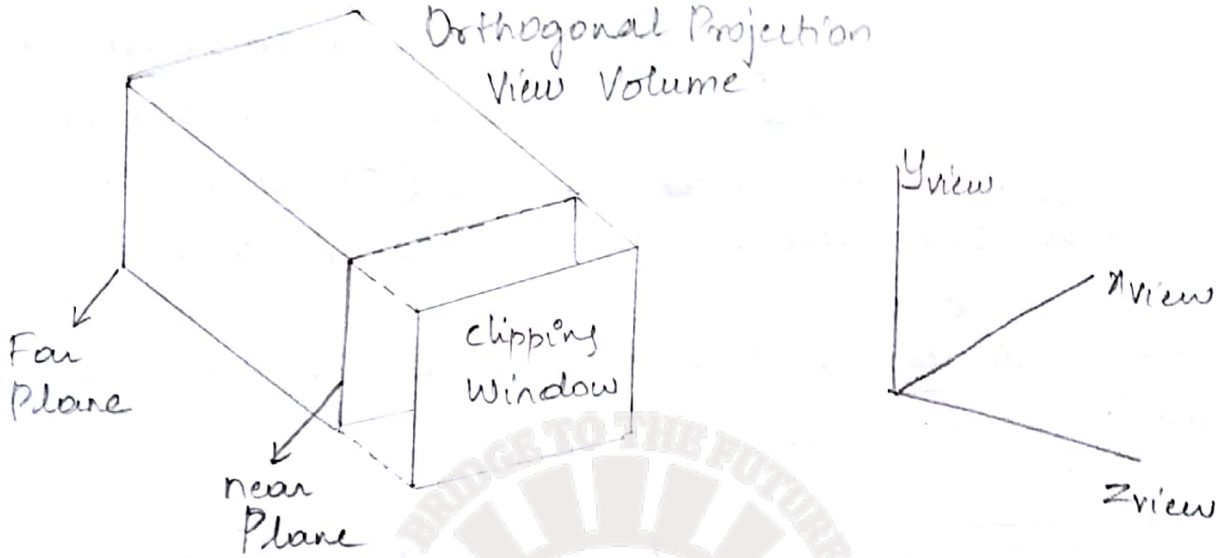






Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S



Normalization Transformation for an Orthogonal Projection

* Using an Orthogonal transfer of co-ordinate position onto the view plane, projected position of any spatial point (x,y,z) is represented simply as (x,y).
* once the limits of view volume is established, the co-ordinate descriptions inside this rectangular parallelepiped are the projection co-ordinates & they are mapped into a normalized view volume.
* x,y,z co-ordinates are normalized in range from 0 to 1. / range from -1 to 1. represented in left-handed system.
* Position (x_{min}, y_{min}, z_{near}) is mapped to normalized position (-1,-1,-1) & position (x_{max}, y_{max}, z_{far}) is mapped to (1,1,1)
* z-co-ordinate positions for the near & far planes are denoted as z_{near} and z_{far}.

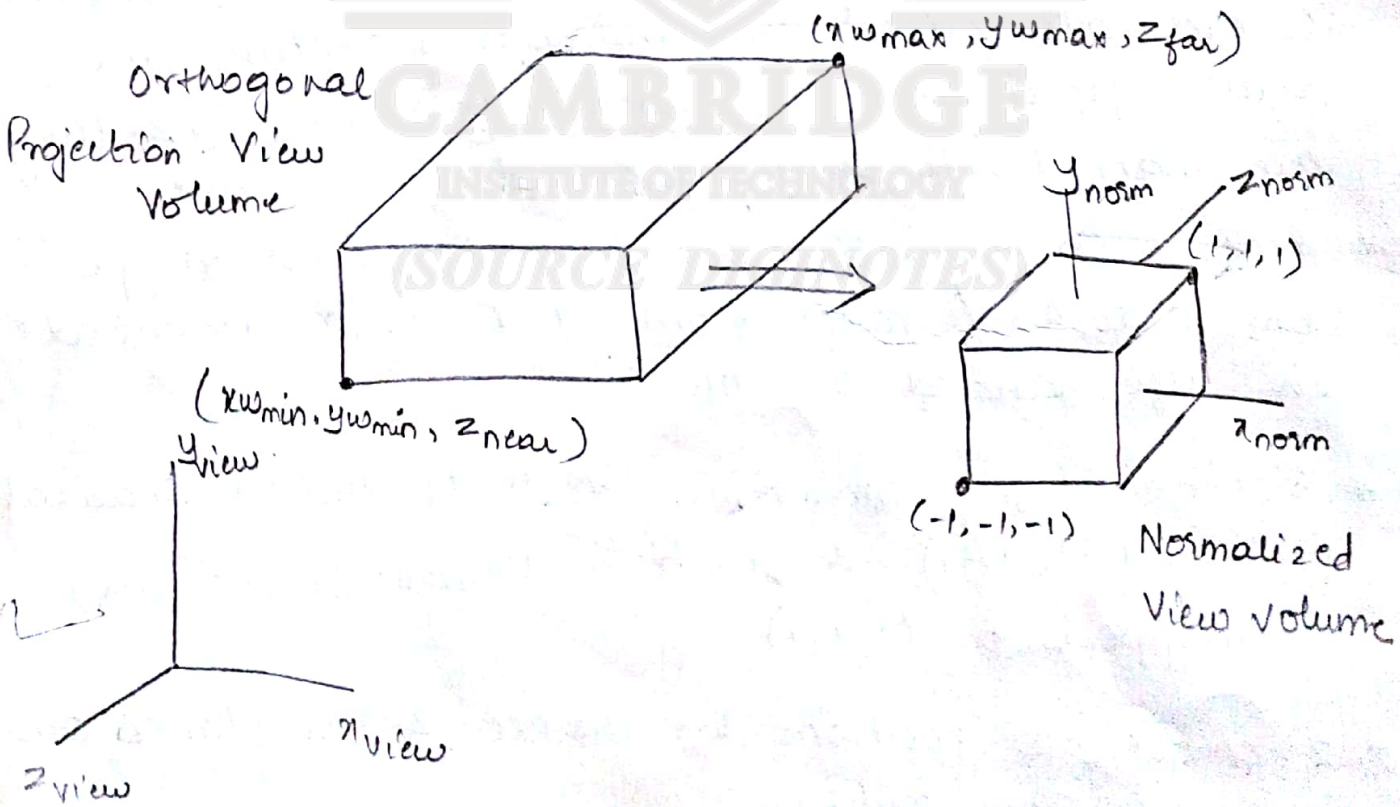
Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

* Check Module 3 for converting the clipping window into normalized symmetric square. In addition, z-co-ordinate values are to be transformed in the range from z_{near} to z_{far} (-1 to 1) using similar calculations as discussed prior.

* The Normalization transformation for the orthogonal view volume is

$$M_{ortho,norm} = \begin{bmatrix} \frac{2}{x_{wmax}-x_{wmin}} & 0 & 0 & \frac{-x_{wmax}+x_{wmin}}{x_{wmax}-x_{wmin}} \\ 0 & \frac{2}{y_{wmax}-y_{wmin}} & 0 & \frac{-y_{wmax}+y_{wmin}}{y_{wmax}-y_{wmin}} \\ 0 & 0 & \frac{-2}{z_{near}-z_{far}} & \frac{z_{near}+z_{far}}{z_{near}-z_{far}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Perspective Projection

- Distance from COP to projection plane is finite. The projectors are not parallel & we specify the center of projection (COP)

COP is also called as Perspective reference point or Projection Reference point.

Perspective foreshortening

The size of the perspective projection of the object varies inversely with the distance of the object from the center of projection.

Vanishing Point.

The perspective projections of any set of parallel lines that are not parallel to the projection plane converge to a vanishing point.

* Perspective projection → projections of distant objects are smaller than the projections of objects of same size that are closer to the view plane.

* The projection path of a spatial position (x, y, z) to a general ~~position~~ projection reference point at $(x_{prp}, y_{prp}, z_{prp})$. The projection line intersects the viewplane at coordinate position (x_p, y_p, z_{vp}) where z_{vp} is some selected position for the view plane along the z_{View}-axis.

$P(x', y', z')$ any point along the projection line

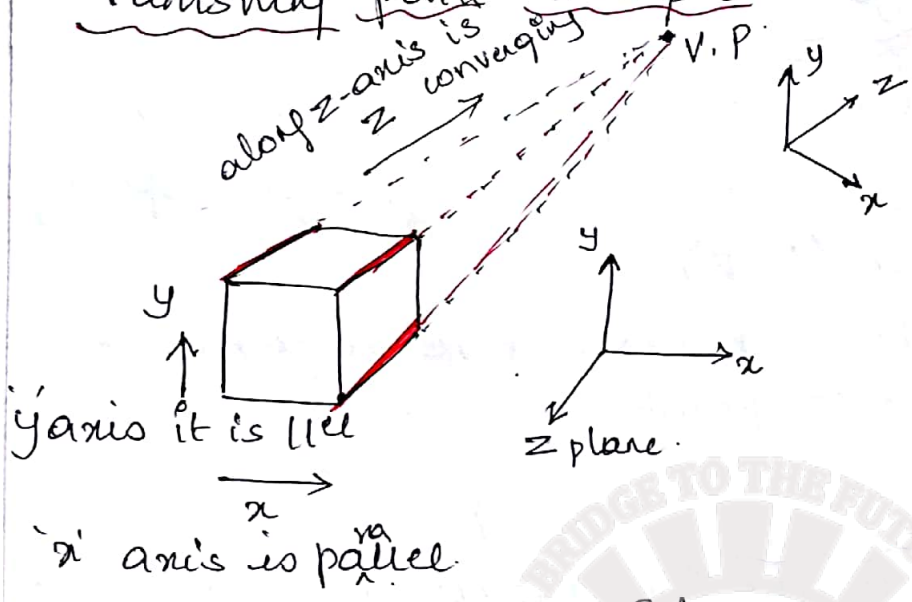
$$\left.\begin{aligned} x' &= x - (x - x_{prp})u \\ y' &= y - (y - y_{prp})u \\ z' &= z - (z - z_{prp})u \end{aligned}\right\}$$ Parametric form

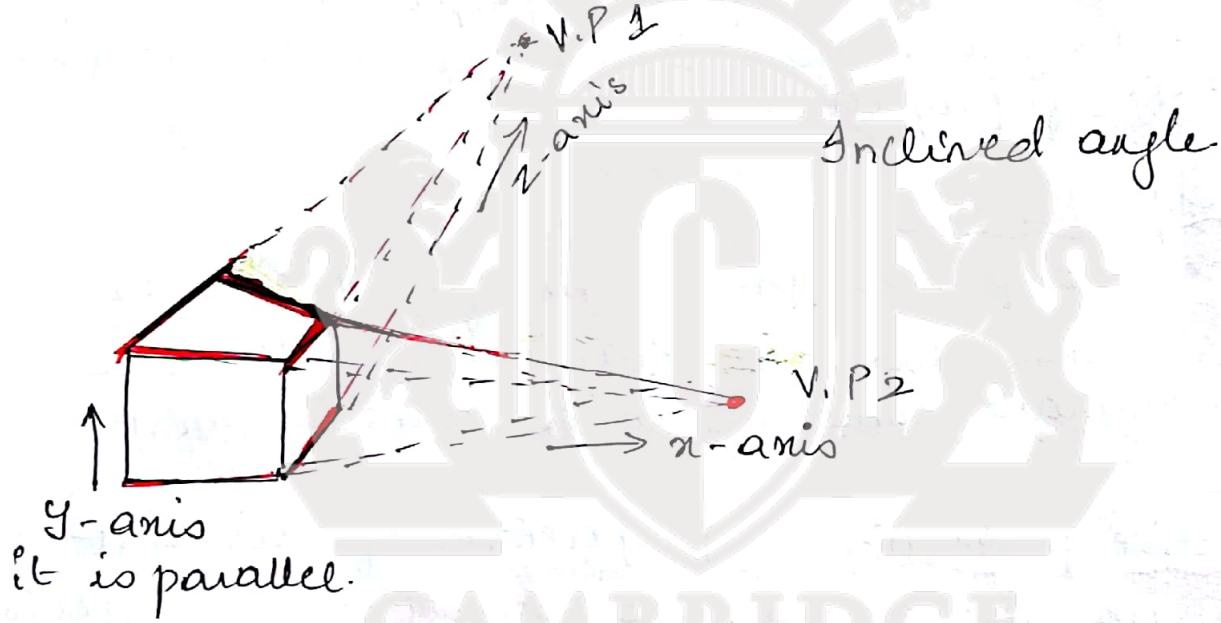
$0 \le u \le 1$ explained in module 5

Source: diginotes.in Go paperless. Save the Earth.

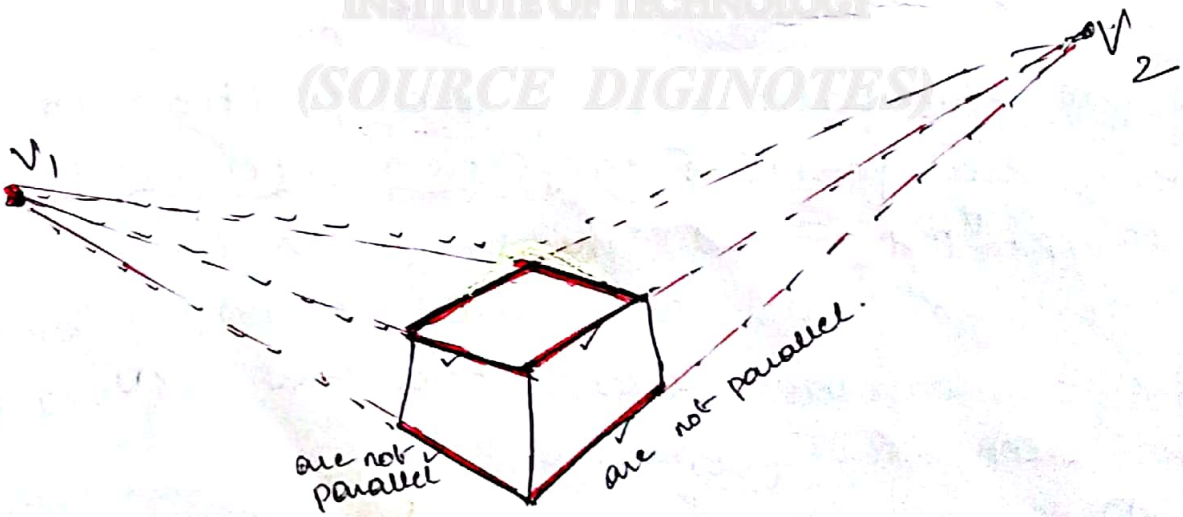
Prof. Supriya S

Vanishing point example.





(OR)



For a set of points that are parallel to one of the principle axis of an object, it is referred to as Principal vanishing point

Source: diginotes.in
Go paperless. Save the Earth.

Cont^n :

* where $x' y' z'$ represents any point along the projection line.

$u = 0$, when we are at point $P = (x, y, z)$

$u = 1$, when we are at the other end. $(x_{prp}, y_{prp}, z_{prp})$

* on view plane $z' = z_{vp}$, solving z' for parameter u at position $z' = z_{vp}$ along the projection line

$$z' = z - (z - z_{prp})u$$

$$z_{vp} = z - (z - z_{prp})u$$

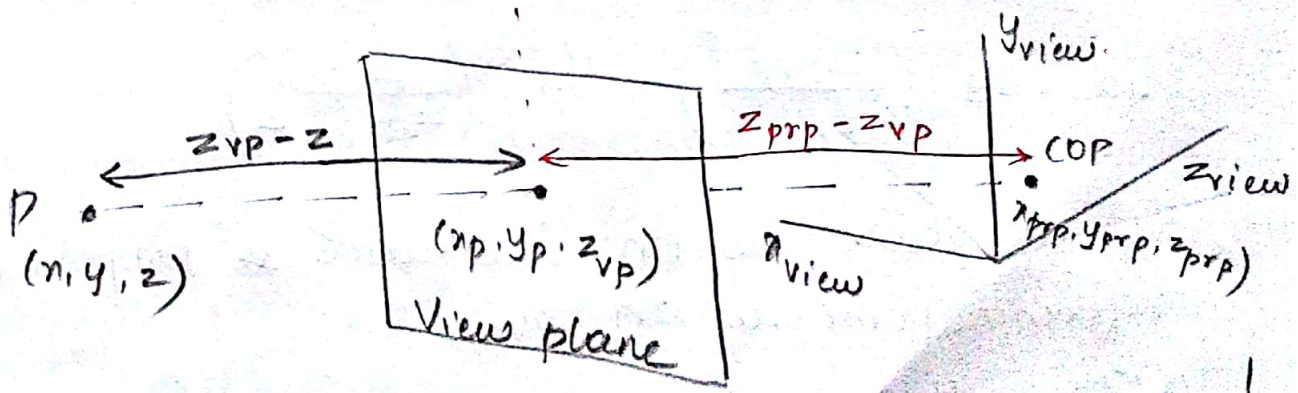
$$(z_{vp} - z) = -(z - z_{prp})u$$

$$u = \frac{z_{vp} - z}{z_{prp} - z}$$

$$\boxed{u = \frac{z_{vp} - z}{z_{prp} - z}}$$

Substituting u in x' & y', we get general perspective transformation eq^n's.

$$\boxed{\begin{aligned} x_p &= x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + x_{prp}\left(\frac{z_{vp} - z}{z_{prp} - z}\right) \\ y_p &= y\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + y_{prp}\left(\frac{z_{vp} - z}{z_{prp} - z}\right) \end{aligned}}$$



$$COP + u(P - COP) \quad 0 \le u \le 1$$

(OR)

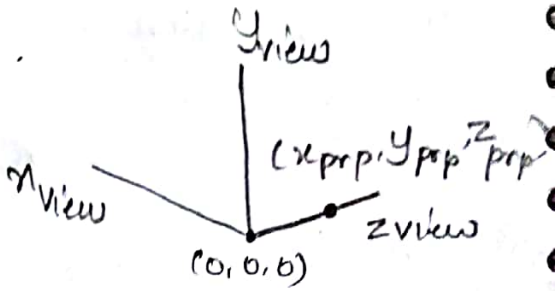
$$P + u(P - COP)$$

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Perspective - Projection Equations : Special cases.

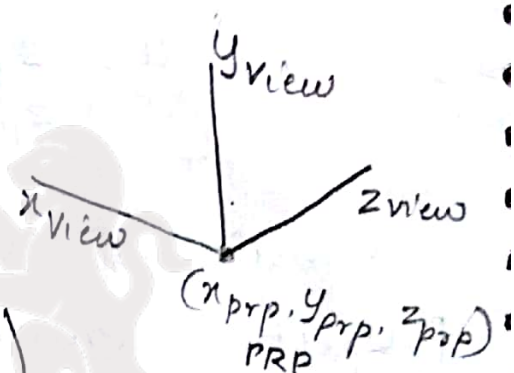
The projection reference point could be limited to positions along z-axis then.



① $x_{prp} = y_{prp} = 0$

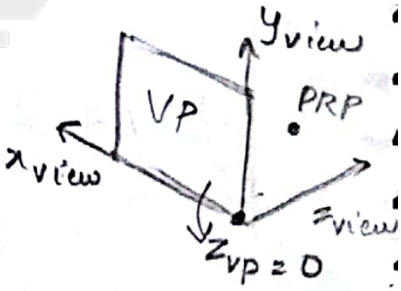
then $x_p = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)$

$y_p = y\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)$



② $(x_{prp}, y_{prp}, z_{prp}) = (0, 0, 0)$

$x_p = x\left(\frac{z_{vp}}{z}\right) \quad y_p = y\left(\frac{z_{vp}}{z}\right)$



③ If view plane is on the uv plane.

$z_{vp} = 0$:

$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right) - x_{prp}\left(\frac{z}{z_{prp} - z}\right)$

$y_p = y\left(\frac{z_{prp}}{z_{prp} - z}\right) - y_{prp}\left(\frac{z}{z_{prp} - z}\right)$

④ if uv plane is on view plane & projection reference point is on z_{view}.

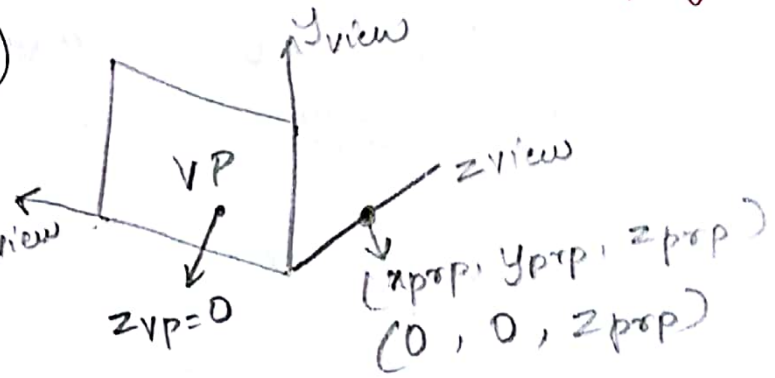
$x_{prp} = y_{prp} = z_{vp} = 0$

Source: diginotes.in Go paperless. Save the Earth.

then

$$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right)$$

$$y_p = y\left(\frac{z_{prp}}{z_{prp} - z}\right)$$

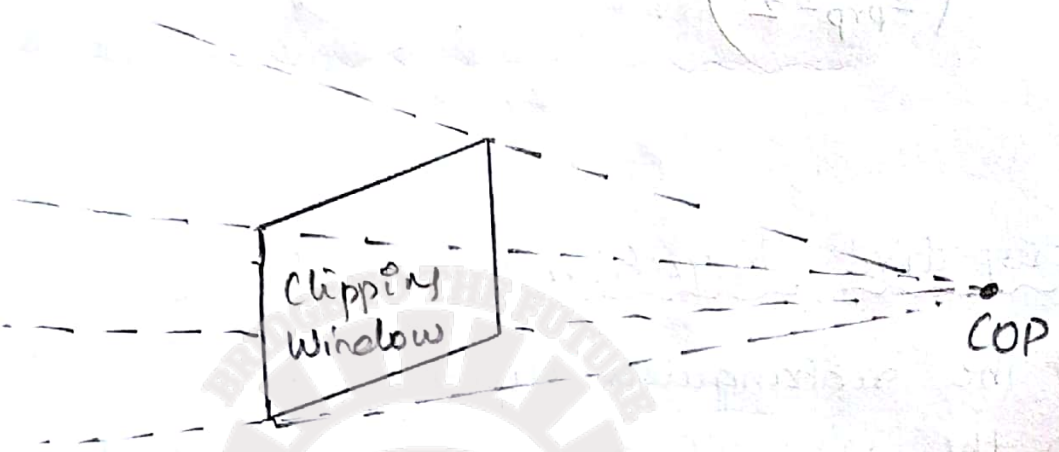


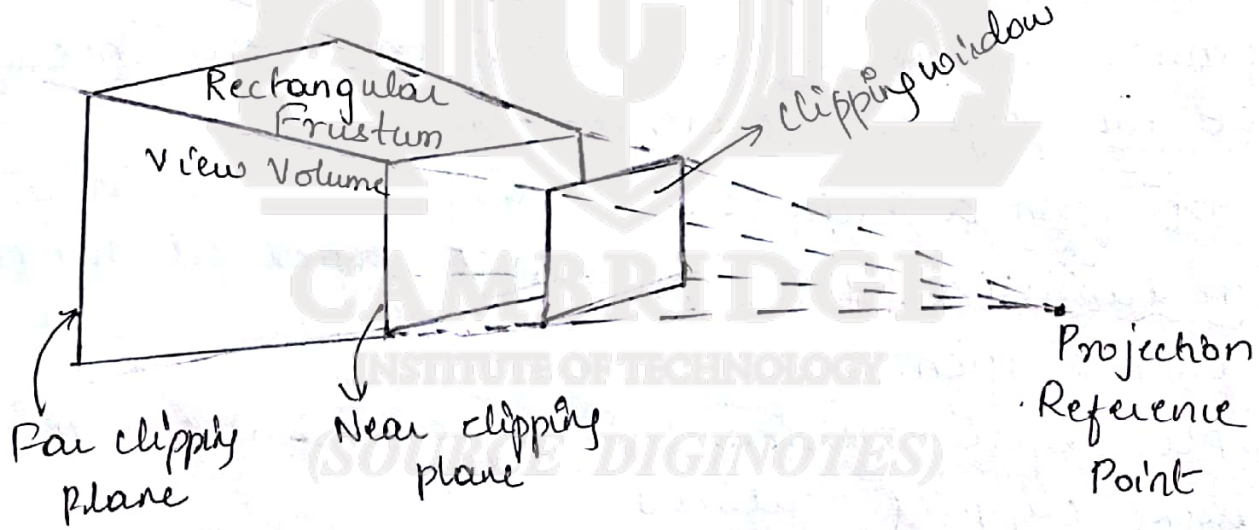
Perspective - Projection View Volume.

* The rectangular clipping window is positioned on the view plane.
* The bounding planes for the view volume are not parallel, because the projection lines are not parallel.
* The bottom, top, & sides of view volume are planes through window edges that all intersect at the projection reference point.
* It form a view volume that is an infinite rectangular pyramid with its apex at the centre of projection.
* All objects outside this pyramid are eliminated using clipping routines.
* The perspective projection view volume is referred as pyramid of vision. (cone of vision of our eyes or camera).
* By adding near and far clipping planes that are perpendicular to Zview axis (& ll to view plane), the other parts of the infinite perspective projection view.

Source: diginotes.in Go paperless. Save the Earth.

volume are chopped off, forming a truncated pyramid or frustum., view volume.





CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE: DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

Prof. Lipsa S

Perspective Projection Transformation matrix

perspective projection equations.

$$x_p = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + x_{prp}\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$y_p = y\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + y_{prp}\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

} consider the denominator $h = z_{prp} - z$

and also consider

$$x_h = x(z_{prp} - z_{vp}) + x_{prp}(z_{vp} - z)$$

$$y_h = y(z_{prp} - z_{vp}) + y_{prp}(z_{vp} - z)$$

Place all the x, y, z corresponding values in the matrix form

Using homogeneous co-ordinate representations, we get

$$x_p = \frac{x_h}{h} \quad \text{and} \quad y_p = \frac{y_h}{h}$$

The perspective-projection transformation of a viewing co-ordinate position is then accomplished in 2 steps

① calculate homogenous co-ordinates using the perspective transformation matrix.

$$P_h = M_{pers} \cdot P$$

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} z_{prp} - z_{vp} & 0 & -x_{prp} & x_{prp} z_{vp} \\ 0 & z_{prp} - z_{vp} & -y_{prp} & y_{prp} z_{vp} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{prp} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(x, y, z, h)

② after the normalized transformation & clipping routines are applied, homogeneous co-ordinates

Source: diginotes.in Go paperless. Save the Earth.

are divided by parameter h to obtain the true transformation - co-ordinate positions.

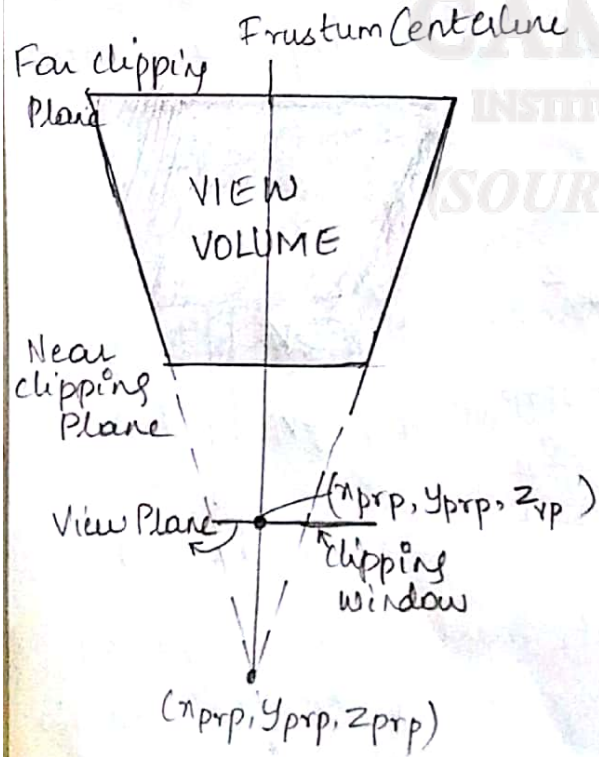
Symmetric Perspective-Projection frustum

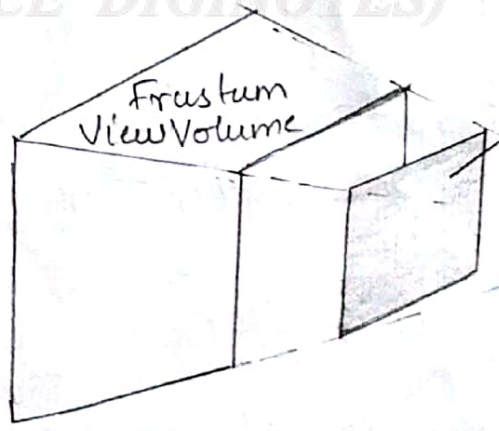
The line from the projection reference point through the center of the clipping window & on through the view volume is the centerline for a perspective projection frustum. If this centerline is perpendicular to view plane – symmetric frustum.

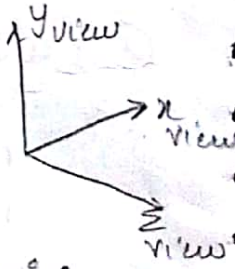
* Frustum centerline intersects the view plane at co-ordinate location $(x_{prp}, y_{prp}, z_{vp})$. The corner positions of clipping window are

$$x_{wmin} = x_{prp} - \frac{width}{2} \qquad x_{wmax} = x_{prp} + \frac{width}{2}$$

$$y_{wmin} = y_{prp} - \frac{height}{2} \qquad y_{wmax} = y_{prp} + \frac{height}{2}$$







CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

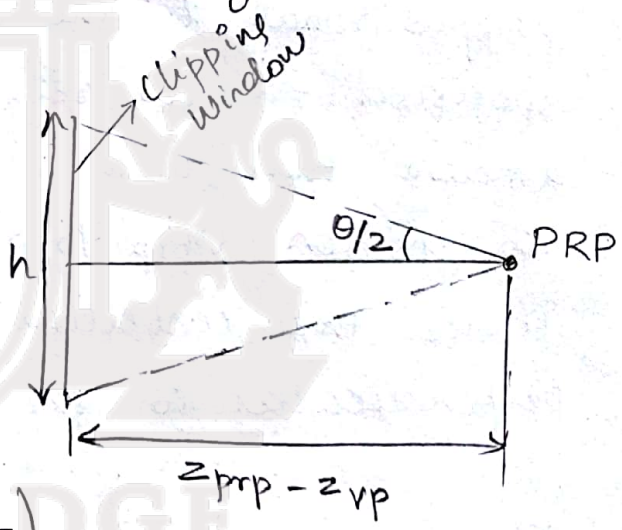
* Another way to specify a symetric perspective projection is to use parametes / properties of camera lens.

A photograph is produced with symetric PP of a scene onto a film plane. Reflected light rays from the objects in scene are collected on film plane from within the "cone of vision" of the camera. This cone of vision is referenced with field-of view angle – measure of size of camera lens.

* field-of view angle → angle b/w the top clipping plane and bottom clipping plane of frustum (determines the height)

aspect ratio = $\frac{\text{width}}{\text{height}}$.

$$\tan\left(\frac{\theta}{2}\right) = \frac{\text{height}/2}{z_{prp} - z_{vp}}$$

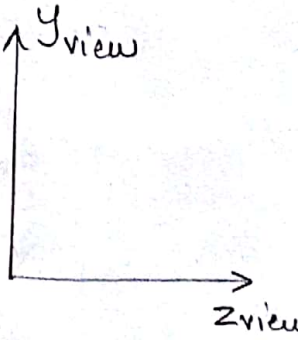


$$\text{height} = 2(z_{prp} - z_{vp}) \tan\left(\frac{\theta}{2}\right)$$

$z_{prp} - z_{vp}$ can be expressed as

$$z_{prp} - z_{vp} = \frac{\text{height}}{2} \cot\left(\frac{\theta}{2}\right)$$

$$= \frac{\text{width} \cdot \cot(\theta/2)}{2 \cdot \text{aspect}}$$



CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

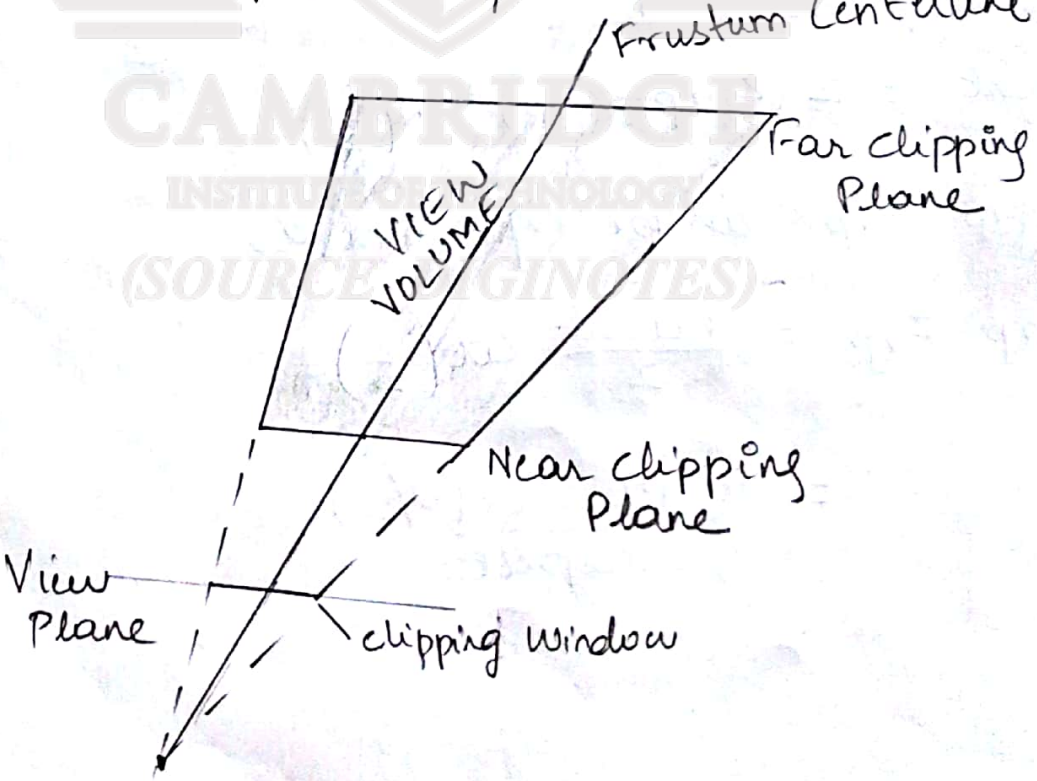
Prof. Supriya S

Oblique Perspective - Projection Frustum

* If the centerline of a perspective-projection view volume is not perpendicular to the view plane
  - oblique Frustum

* An oblique PP view volume can be converted to a symmetric frustum by applying z-axis shearing transformation matrix.
* It shifts all position on any plane that is perpendicular to the z-axis by an amount that is proportional to the distance of the plane from a specified z-axis reference position i.e shift by an amount that will move the center of clipping window to position (x_{prp}, y_{prp}) on the view plane.

Hence the centerline is adjusted such that it is perpendicular to the view plane.



Source: diginotes.in Go paperless. Save the Earth.

Taking PRP as $(x_{prp}, y_{prp}, z_{prp}) = (0,0,0)$, shearing matrix is

$$M_{zshear} = \begin{bmatrix} 1 & 0 & Sh_{zx} & 0 \\ 0 & 1 & Sh_{zy} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

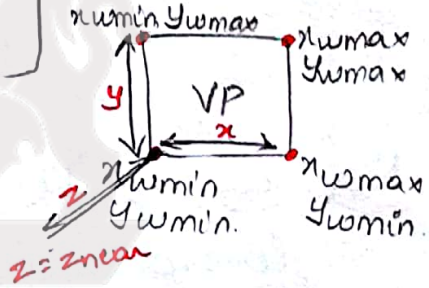
if the view plane is at the position of the near clipping plane then (move it) $z_{vp} = z_{near}$ & $x_p, y_p = (0,0)$

$$\begin{bmatrix} x_p \to 0 \\ y_p \to 0 \\ z_{vp} \to z_{near} \\ 1 \to 1 \end{bmatrix} = M_{zshear} \cdot \begin{bmatrix} \frac{x_{wmin} + x_{wmax}}{2} \\ \frac{y_{wmin} + y_{wmax}}{2} \\ z_{near} \\ 1 \end{bmatrix}$$

$\frac{x_{wmin} + x_{wmax}}{2}$ → x direction of views plane

$\frac{y_{wmin} + y_{wmax}}{2}$ → y direction of views plane

z_{near} → z direction



Matrix Multiply

$$\frac{x_{wmin} + x_{wmax}}{2} + Sh_{zx} \cdot z_{near}$$

$$\boxed{Sh_{zx} = -\frac{(x_{wmin} + x_{wmax})}{2 \cdot z_{near}}}$$

$$\frac{y_{wmin} + y_{wmax}}{2} + Sh_{zy} \cdot z_{near}$$

$$\boxed{Sh_{zy} = -\frac{(y_{wmin} + y_{wmax})}{2 \cdot z_{near}}}$$

Thus the Perspective projection matrix when PRP is at viewing origin $(x_{prp}, y_{prp}, z_{prp}) = (0,0,0)$ and view plane is the near clipping plane $z_{VP} = z_{near}$ then the matrix M_{pres} is simplified as

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

$$M_{pres} = \begin{bmatrix} -z_{near} & 0 & 0 & 0 \\ 0 & -z_{near} & 0 & 0 \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

finally

$$M_{obliquepers} = M_{pres} \cdot M_{zshear}$$

substitute for sh_{zx} & sh_{zy}

$$= \begin{bmatrix} -z_{near} & 0 & \frac{x_{wmin}+x_{wmax}}{2} & 0 \\ 0 & -z_{near} & \frac{y_{wmin}+y_{wmax}}{2} & 0 \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Normalized Perspective - Projection Transformation co-ordinates.

The final step in the perspective-projection transformation process is to map this parallelepiped to a Normalized view volume.

Normalised ation matrix for Perspective projection transformation

$$M_{normpers} = M_{xyscale} \cdot M_{obliquepers}$$

$$= \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -z_{near} & 0 & \frac{x_{wmin}+x_{wmax}}{2} & 0 \\ 0 & -z_{near} & \frac{y_{wmin}+y_{wmax}}{2} & 0 \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Source: diginotes.in Go paperless. Save the Earth.

$$= \begin{bmatrix} -z_{near}S_x & 0 & S_x\frac{x_{wmin}+x_{wmax}}{2} & 0 \\ 0 & -z_{near}S_y & S_y\frac{y_{wmin}+y_{wmax}}{2} & 0 \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

we obtain homogeneous co-ordinates

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = M_{normpers} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

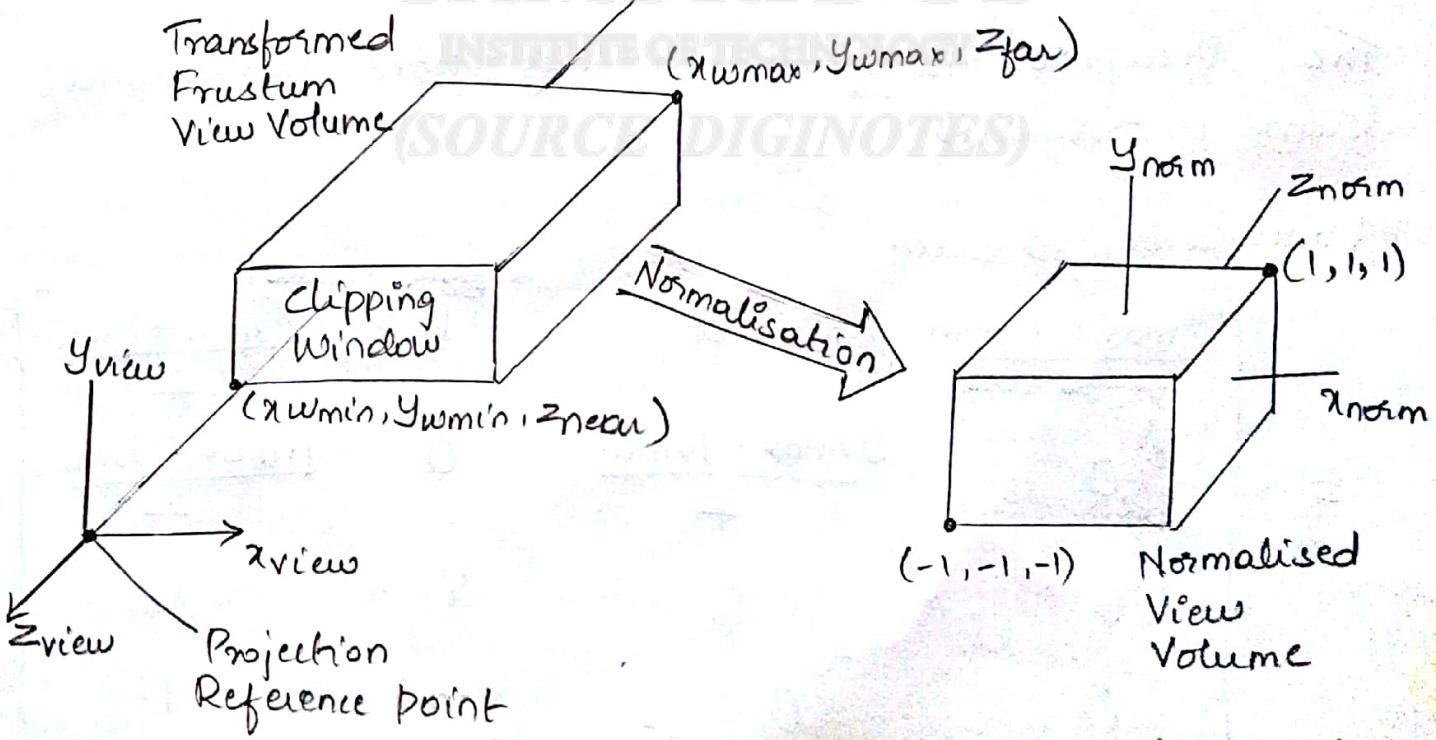
(multiply the above matrix)

and projection coordinates are

$$x_p = \frac{x_h}{h} = \frac{-z_{near}S_x x + [S_x(x_{wmin}+x_{wmax})/2]}{-z}$$

$$y_p = \frac{y_h}{h} = \frac{-z_{near}S_y y + [S_y(y_{wmin}+y_{wmax})/2]}{-z}$$

$$z_p = \frac{z_h}{h} = \frac{S_z z + t_z}{-z}$$



Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Substitute $(x_{wmin}, y_{wmin}, z_{near})$ as $(-1,-1,-1)$ & $x_{wmax}, y_{wmax}, z_{far}$ as $(1,1,1)$.

$$S_x = \frac{2}{x_{wmax} - x_{wmin}} \qquad S_y = \frac{2}{y_{wmax} - y_{wmin}}$$

$$S_z = \frac{z_{near} + z_{far}}{z_{near} - z_{far}} \qquad t_z = \frac{2 z_{near} z_{far}}{z_{near} - z_{far}}$$

Hence

$$M_{normpers} = \begin{bmatrix} \frac{-2 z_{near}}{x_{wmax} - x_{wmin}} & 0 & \frac{x_{wmax} + x_{wmin}}{x_{wmax} - x_{wmin}} & 0 \\ 0 & \frac{-2 z_{near}}{y_{wmax} - y_{wmin}} & \frac{y_{wmax} + y_{wmin}}{y_{wmax} - y_{wmin}} & 0 \\ 0 & 0 & \frac{z_{near} + z_{far}}{z_{near} - z_{far}} & \frac{2 z_{near} z_{far}}{z_{near} - z_{far}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

The Viewport Transformation and three-Dimensional Screen co-ordinates

$M_{normviewvol,\ 3D\ screen}$

$$= \begin{bmatrix} \frac{x_{Vmax} - x_{Vmin}}{2} & 0 & 0 & \frac{x_{Vmax} + x_{Vmin}}{2} \\ 0 & \frac{y_{Vmax} - y_{Vmin}}{2} & 0 & \frac{y_{Vmax} + y_{Vmin}}{2} \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya. S

Open GL 3D viewing functions – Refer from text book.

CAMBRIDGE
INSTITUTE OF TECHNOLOGY
(SOURCE -DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Visible Surface detection [Hidden surface elimination]

Given a set of 3D surfaces to be projected onto a 2D screen, obtain the nearest surface corresponding to any point on the screen.

* Some methods require more memory, some involve more processing time, some apply only to special types of objects
* Identifying the surface in 3D requires calculation of 'z' co-ordinate (depth) value & surface Normal (if curved).

Classification of Visible surface detection Algorithms

① Object space methods (continous)

* compares objects and parts of objects to each other to determine which surfaces should be labeled as visible (use of bounding box, check limits along each direction).
* Order the surface being drawn, such that it provides the correct impression of depth variations and positions.

② Image space methods (discrete)

Visibility is decided point by point at each pixel position on the projection plane. Screen resolution can be a limitation.

× { Hidden surfaces – (a) surface for rendering
or
(b) Line drawing.

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya. S

Visible surface detection Algorithm most use sorting and coherence methods to improve Performance.

* Sorting - is used to facilitate depth comparisions by ordering the individual surfaces in scene according to their distances from the view plane.

* Coherence methods are used to taice the advantage of regularities in the scene.
  - if one object is entirely seperate from another, do not compare (object coherence)
  - face coherence : smooth variations across a face, incramentally modify.
  - Edge coherence : visibility change if a edge cross behind a visible face.
  - Implied edge coherence : line of intersection of a planar face penetrating another, can be obtained from two points on the intersection. (2nd face penetrating one face)
  - Scan line coherence - successive lines have similar spans.
  - Depth coherence - Use difference equation to estimate depths of nearby points on the same surface.
  - Frame coherence - Pictures of 2 successive frames of an animation sequence are quite similar. (small changes in object & viewpoint).

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S.

Back-face detection (object-space Method)

A polygon (in 3D) is a back face if $V.N > 0$

* Concepts of front-back tests
* A point (x, y, z) is behind a polygon surface if

$$Ax + By + Cz + D < 0$$

where A, B, C, D are plane parameters for the polygon

* Consider the direction of normal vector N for a polygon surface. if V_{view} is a vector in the viewing direction from our camera position., the polygon is back face if $\boxed{V_{view} \cdot N > 0}$

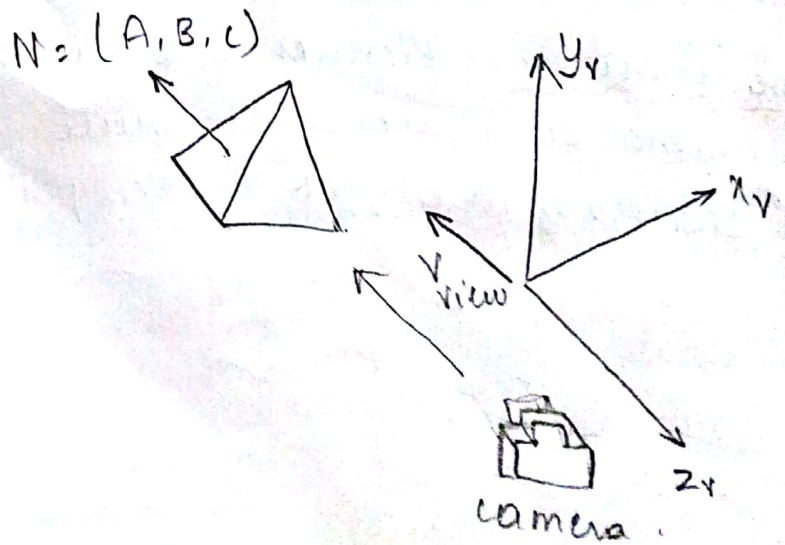
(viewing direction is parallel to viewing V_z axis)

Let $V = (0, 0, V_z)$ and $N = Ax + By + Cz$.

then $V.N = V_z \cdot C$ (z component of normal vector N is to be considered)

if the viewing direction is along the -ve z-axis, a polygon is a back face if the z component, C, of its normal N satisfies $C < 0$.

$\therefore \boxed{C \leq 0}$



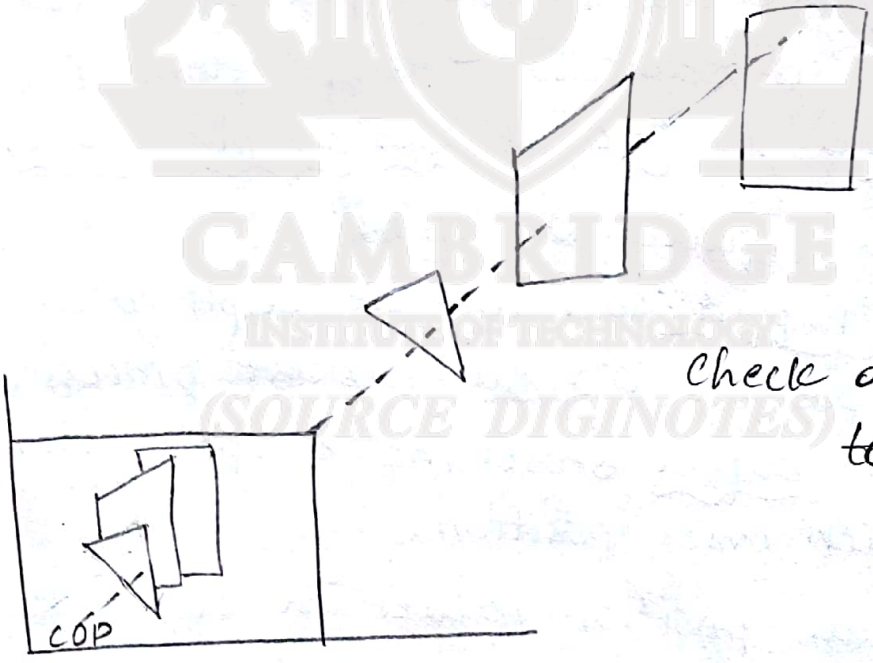
CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

* Back face have normal vectors that point away from the viewing position (viewing direction is along +ve z-axis) are identified by $C \geq 0$.
* For convex polygons, this test identifies all the hidden surfaces in scene since each surface is completely visible or completely hidden.
* For concave polygons, more tests are to carried out to determine whether there are any additional faces that are totally or partially obscured by other faces.

Depth-Buffer Method (image space approach)

also referred as z-buffer method.



Check algorithm from text Book

* It compares surface depth values throughout a scene for each pixel position on the projection Plane.
* Each surface of scene is processed seperately, one

CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

pixel at a time across the surface.

* Refered to as z-buffer since object depth is usually measured along the z-axis of viewing system.
* The figure shows 3 surfaces at varying distances along the orthographic projection line from position (x,y) on a view plane.
* Surfaces can be processed in any order
* As each surface is processed, its depth from the view plane is compared to previously processed surface.
* If a surface is closer than any previously processed surfaces, its surface color is calculated & saved, along with its depth.
* depth ~~tobo~~ buffer algorithm is typically caried out in normalized co-ordinates - range from 0 at near clipping plane to 1.0 at far clipping plane.
* The method requires 2 buffers
  - depth buffer used to store depth values for each (x,y) position as surfaces are processed.
  - frame buffer stores the surface-color values for each pixel position.
* Initially all position values in depth buffer are set to 1.0 (maximum depth), frame buffer is initialised to the background color.
* Each surface listed in polygon tables are processed, one scan line at a time, by calculating the depth

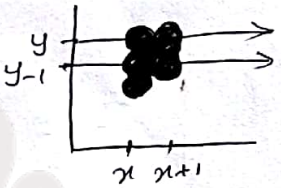
Source: diginotes.in Go paperless. Save the Earth.

values at each (x,y) pixel position.

* This calculated depth is compared to the value previously stored in depth buffer for that pixel position
* If the calculated depth is less than the value stored in the depth buffer, a new depth value is stored.
* Then the surface color at that position is computed & placed in the corresponding pixel location in the frame buffer.

At surface position (x,y) the depth is calculated from the plane equation as

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



If x values along the horizontal lines differ by ±1 & y values ~~along~~ on adjacent scan lines differ by ±1

If the depth of position (x,y) has been determined to be z, then the depth z' of next position (x+1, y) is

$$z' = \frac{-A(x+1) - By - D}{C}$$

$$= \frac{-Ax - By - D}{C} \mp \frac{A}{C}$$

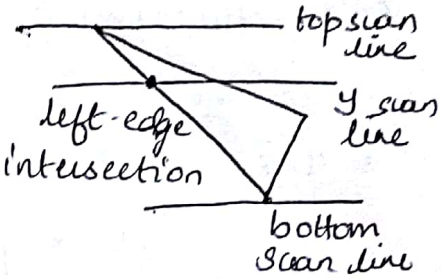
$$\boxed{z' = z \mp \frac{A}{C}}$$

$-A/C$ is constant for each surface.

→ Successive depth values across a scanline are obtained by just adding with single value, $(-A/C)$ addition

Source: diginotes.in Go paperless. Save the Earth.

We can Prof. Supriya S
* To implement the depth buffer algorithm by starting at a top vertex of the polygon, we could then calculate recursively the x-value down the left-edge of the polygon.



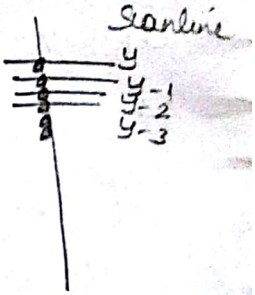
using slope of equation

$$x' = x - \frac{1}{m}, \quad y = y - 1$$

$$\therefore z' = \frac{-A(x - 1/m) - B(y-1) - D}{C}$$

$$= \frac{-Ax - By - D}{C} + \frac{A/m + B}{C}$$

$$\boxed{z' = z + \frac{A/m + B}{C}}$$ → depth values down the edge are obtained



* proceeding down the vertical edge

$$x = x, \quad y = y - 1$$

$$z' = \frac{-Ax - B(y-1) - D}{C}$$

$$= \frac{-Ax - By - D}{C} + \frac{B}{C}$$

$$\boxed{z' = z + \frac{B}{C}}$$

CAMBRIDGE INSTITUTE OF TECHNOLOGY (SOURCE DIGINOTES)

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

OpenGL Visible Surface-detection functions / visibility detection functions.

① OpenGL Polygon Culling functions.

Back face removal is accomplished with

- glEnable (GL_CULL_FACE);
- glCullFace (mode);

mode is assigned to GL_BACK (default)
GL_FRONT
GL_FRONT_AND_BACK

- glDisable (GL_CULL_FACE);

② OpenGL Depth Buffer functions.

- glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

Depth Buffer values can then be initialized with

- glClear (GL_DEPTH_BUFFER_BIT);

There is no need to clear the depth buffer each time we want to display a new frame. In OpenGL depth values are normalized in the range from 0 to 1.0, so that the preceding initialization sets all the depth buffer values to max of 1.0 by default.

Depth buffer visibility detection routines are activated using glEnable (GL_DEPTH_TEST);

glDisable (GL_DEPTH_TEST);

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

Q. Depth-buffer visibility testing can also be set with other initial value for the maximum depth.

glClearDepth (maxDepth);

↓

can be set to any value between 0 & 1

It loads this initialization value into depth buffer.

next glClear (GL_DEPTH_BUFFER_BIT) must be invoked.

* Projection co-ordinates are normalized to range from -1.0 to 1.0

depth values b/w near & far clipping planes are normalised to range from 0.0 to 1.0.

Hence we can adjust these normalization values with

glDepthRange (nearNormDepth, farNormDepth);

↓ 0.0 ↓ 1.0 by default.

* glDepthFunc (testCondition);

↳ GL_LESS (default)
GL_GREATER
GL_EQUAL
GL_NOTEQUAL
GL_LEQUAL
GL_GEQUAL
GL_NEVER.
GL_ALWAYS.

Source: diginotes.in Go paperless. Save the Earth.

Prof. Supriya S

To check if the depth buffer status is read-only or read-write status.

glDepthMask (writeStatus)
↳ GL_TRUE (default value) ↑ both read & write
GL_FALSE (write mode for depth buffer is disabled)

(3) OpenGL ~~Wire~~ Wire-frame surface visibility methods

glPolygonMode (GL_FRONT_AND_BACK, GL_LINE)

(4) OpenGL depth-cueing functions.

To vary the brightness of an object as a function of its distances from the viewing position with.

glEnable (GL_FOG);
glFogi (GL_FOG_MODE, GL_LINEAR);

It applies linear depth fuction values to object colors using $d_{min} = 0.0$ & $d_{max} = 1.0$.

d_{min} & d_{max} can be set to different values.

glFogf (GL_FOG_START, minDepth);
glFogf (GL_FOG_END, maxDepth);

minDepth & maxDepth → assigned floating point values.

Source: diginotes.in Go paperless. Save the Earth.