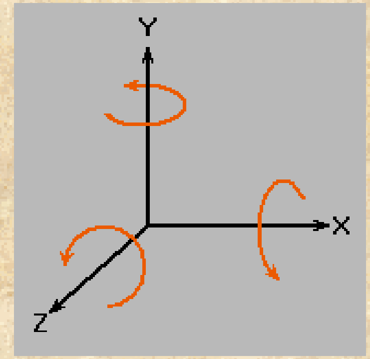
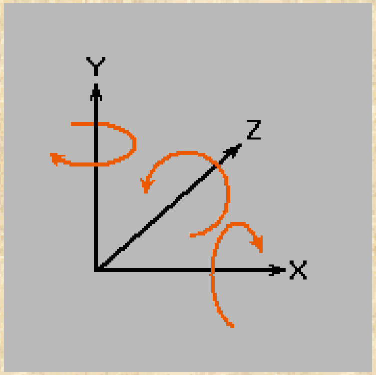
Three-Dimensional Graphics

- Use of a right-handed coordinate system (consistent with math)
- Left-handed suitable to screens.
- To transform from right to left, negate the z values.



Right Handed Space



Left Handed Space

Homogeneous representation of 3D point:

$|x \; y \; z \; 1|^T$

(w=1 for a 3D point)

Transformations will be represented by 4x4 matrices.

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & Sh_x & 0 \\ 0 & 1 & Sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation **Scale** **Shear**

Rotation Matrices along:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

X-axis **Y-axis** **Z-axis**

Why is the sign reversed in one case ?

Rotation About an Arbitrary Axis in Space

Assume we want to perform a rotation about an axis in space passing through the point (x0, y0, z0) with direction cosines (cx, cy, cz) by θ degrees.

1) First of all, translate by: - (x0, y0, z0) = |T|.

2) Next, we rotate the axis into one of the principle axes,
let's pick, Z (|Rx|, |Ry|).

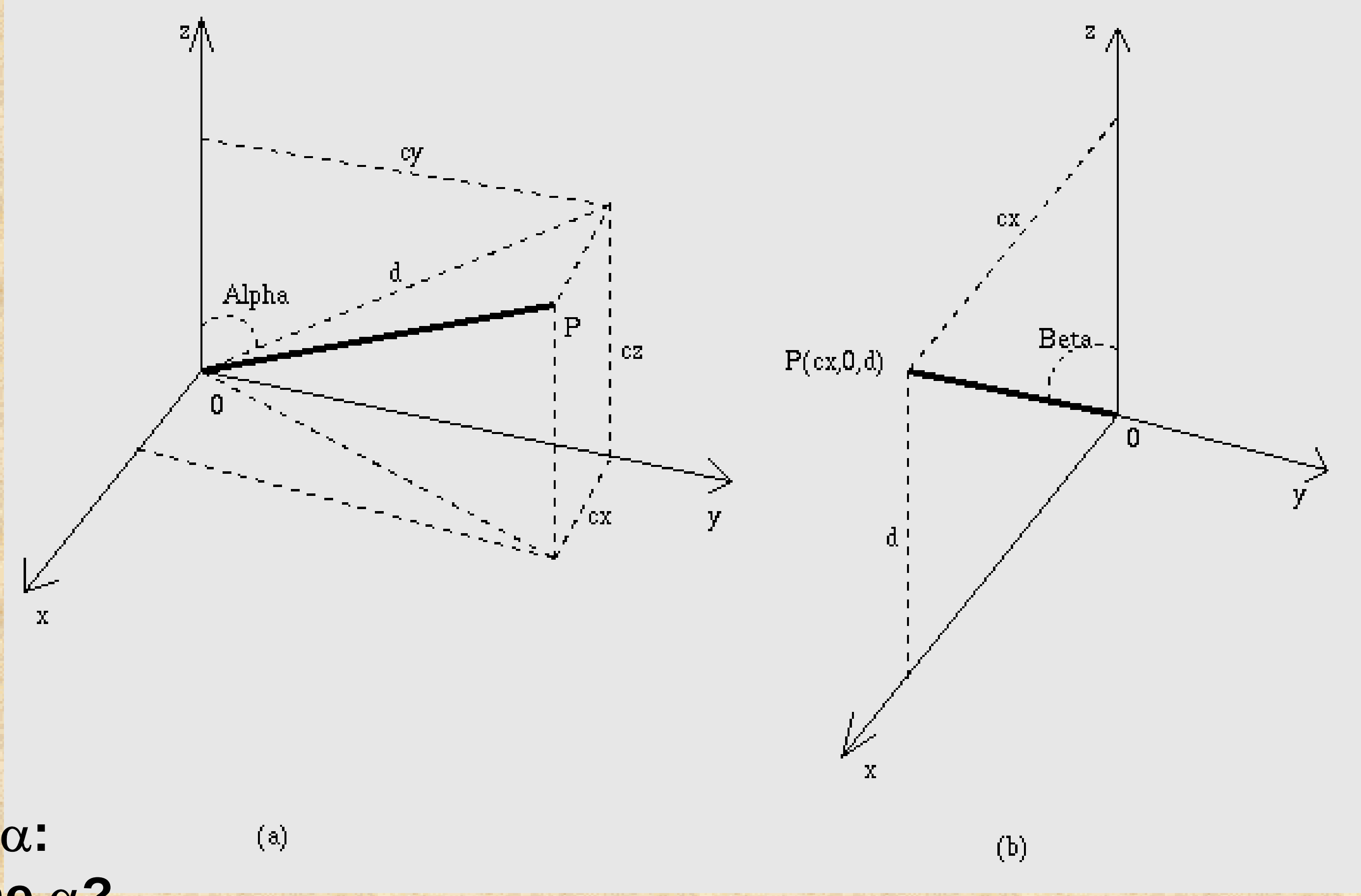
3) We rotate next by θ degrees in Z (|Rz(θ)|).

4) Then we undo the rotations to align the axis.

5) We undo the translation: translate by (x0, y0, z0)

The tricky part is (2) above.

This is going to take 2 rotations,
1 about x (to place the axis in the xz plane) and
1 about y (to place the result coincident with the z axis).



Rotation about x by α:
How do we determine α?

Project the unit vector, along OP, into the yz plane as shown below. **The y and z components are c_y and c_z, the direction cosines of the unit vector along the arbitrary axis. It can be seen from the diagram above, that :**

$$d = \mathrm{sqrt}(c_y^2 + c_z^2),$$

therefore

$$\cos(\alpha) = c_z/d$$

$$\sin(\alpha) = c_y/d$$

Rotation by β about y:

How do we determine β? Similar to above:

Determine the angle β to rotate the result into the Z axis:
The x component is c_x and the z component is d.

$\cos(\beta) = d = d/(\text{length of the unit vector})$
$\sin(\beta) = c_x = c_x/(\text{length of the unit vector})$.

Final Transformation:

$$M = |T|^{-1} |R_x|^{-1} |R_y|^{-1} |R_z| |R_y| |R_x| |T|$$

If you are given 2 points instead, you can calculate the direction cosines as follows:

$$V = | (x1 - x0) \ (y1 - y0) \ (z1 - z0) |^T$$

$c_x = (x1 - x0)/ |V|$
$c_y = (y1 - y0)/ |V|$
$c_z = (z1 - z0)/ |V|$, **where $|V|$ is the length of the vector V.**

Spaces

Object Space
definition of objects. Also called Modeling space.

World Space
where the scene and viewing specification is made

***Eyespace* (Normalized Viewing Space)**
where eye point (COP) is at the origin looking down the Z axis.

3D Image Space
A 3D Perspected space.
Dimensions: -1:1 in x & y, 0:1 in Z.
Where Image space hidden surface algorithms work.

Screen Space (2D)
Coordinates 0:width, 0:height

Projections

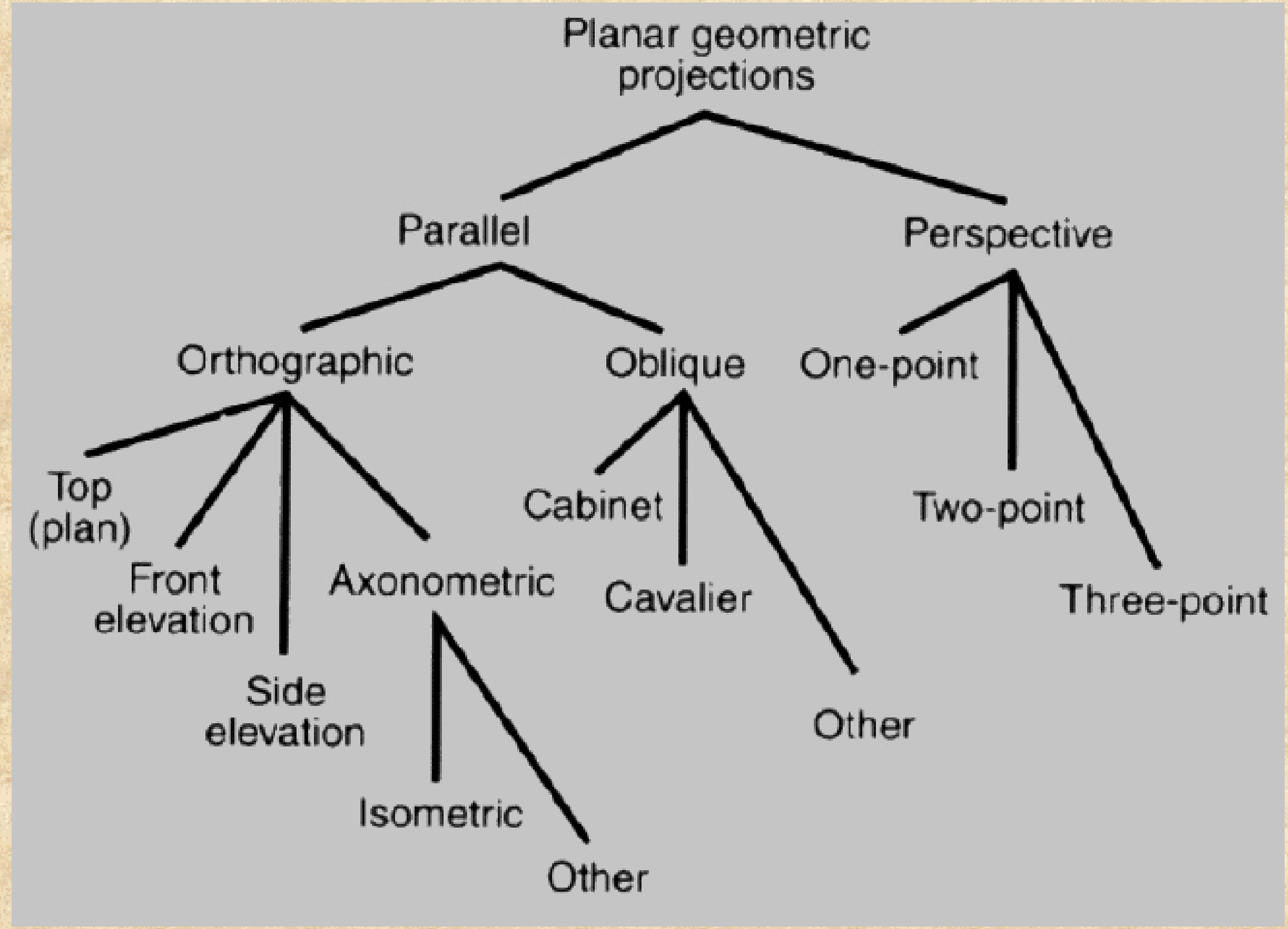
We will look at several planar geometric 3D to 2D projection:

-Parallel Projections
Orthographic
Oblique

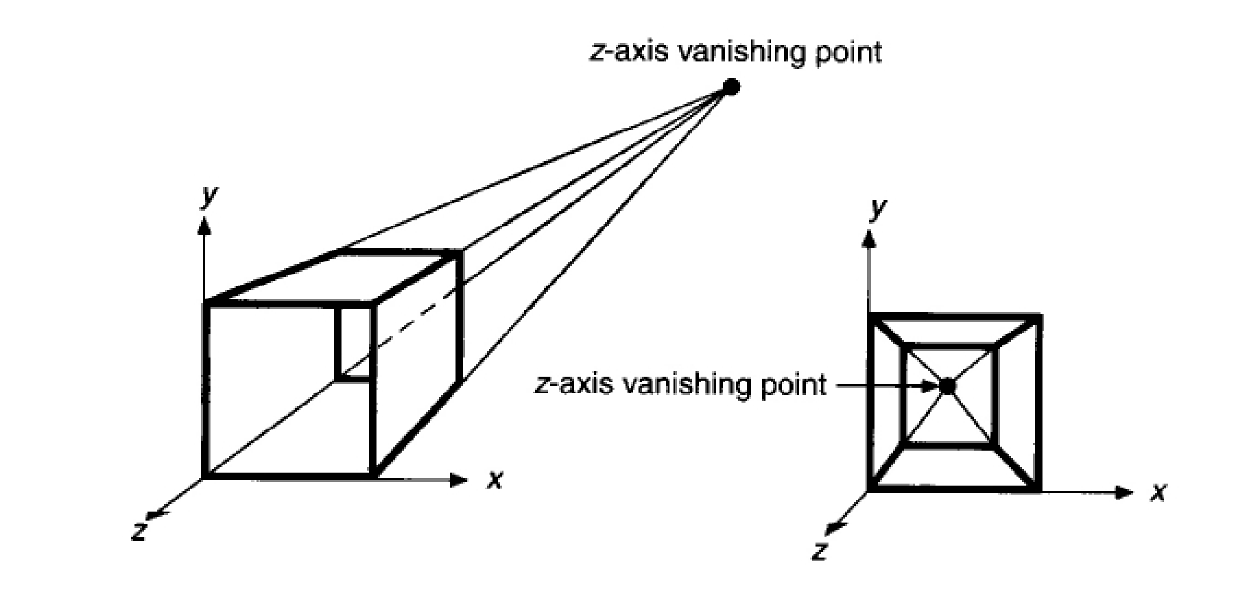
-Perspective

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.

Classification of Geometric Projections



Perspective Projections

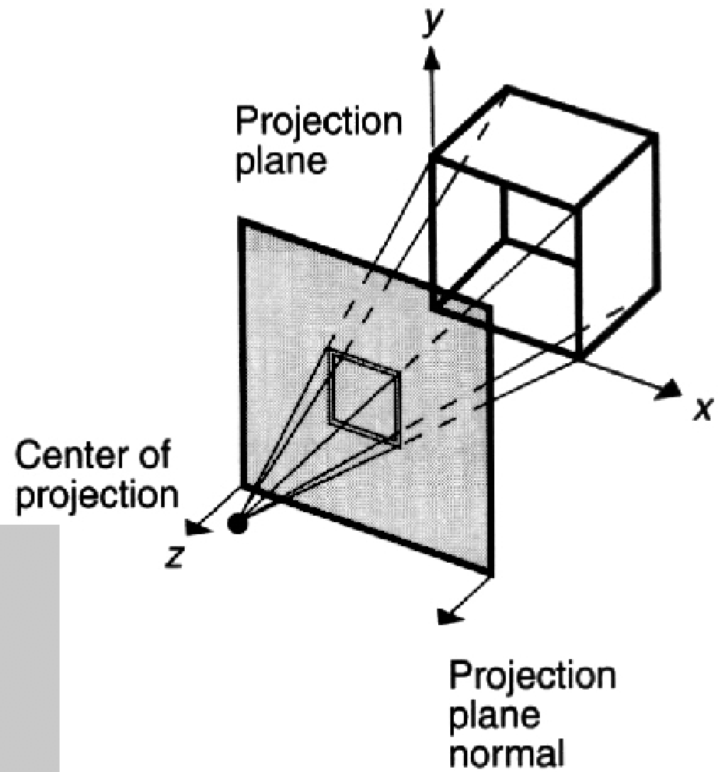


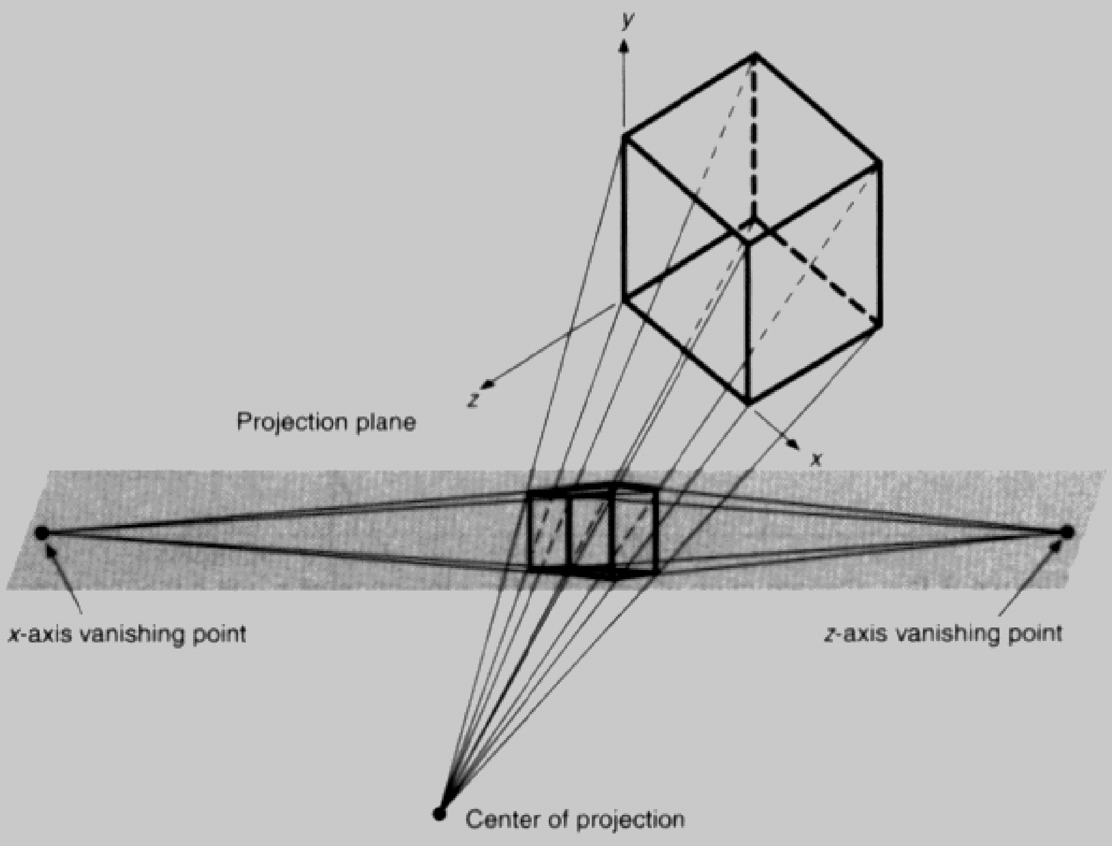
Distance from COP to projection plane is finite. The projectors are not parallel & we specify a center of projection.

Center of Projection is also called the Perspective Reference Point
COP = PRP

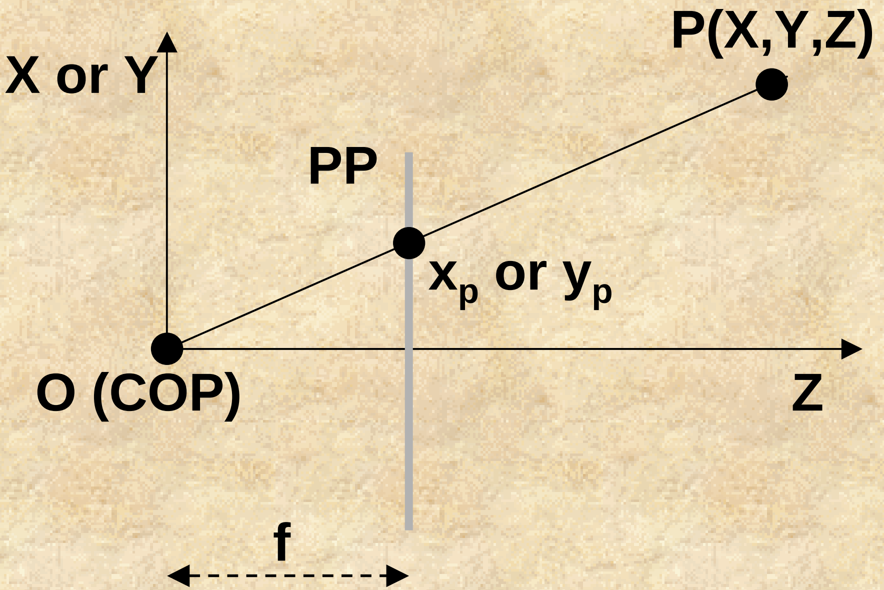
Perspective foreshortening: the size of the perspective projection of the object varies inversely with the distance of the object from the center of projection.

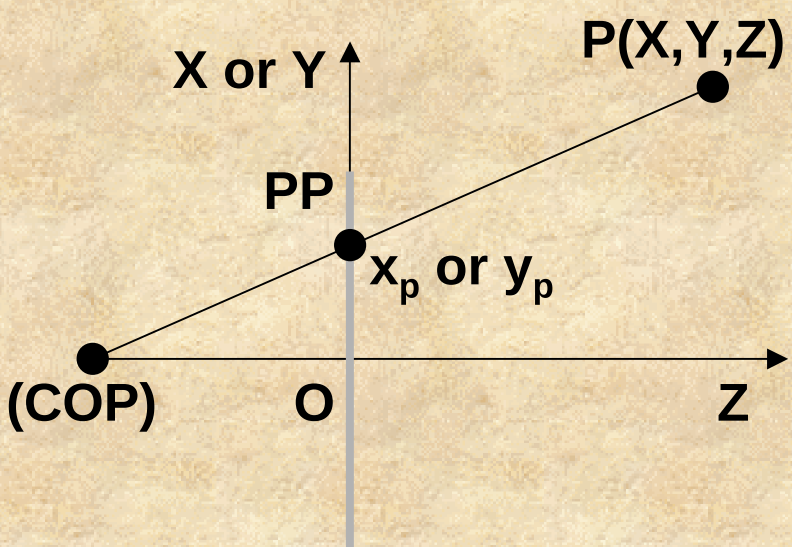
Vanishing Point: The perspective projections of any set of parallel lines that are not parallel to the projection plane converge to a vanishing point.

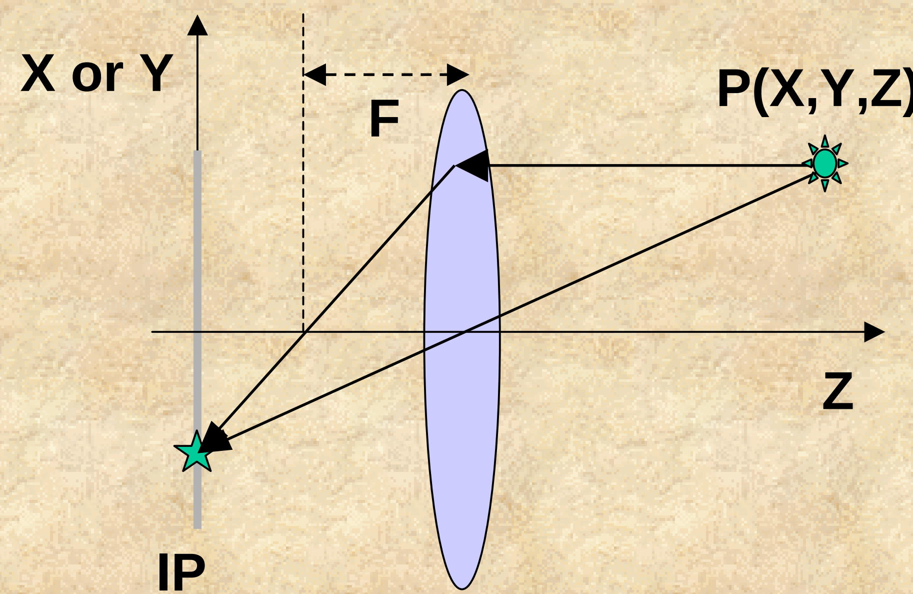

y
Projection plane
x
Center of projection
z
Projection plane normal


y
z
Projection plane
x
x-axis vanishing point
z-axis vanishing point
Center of projection

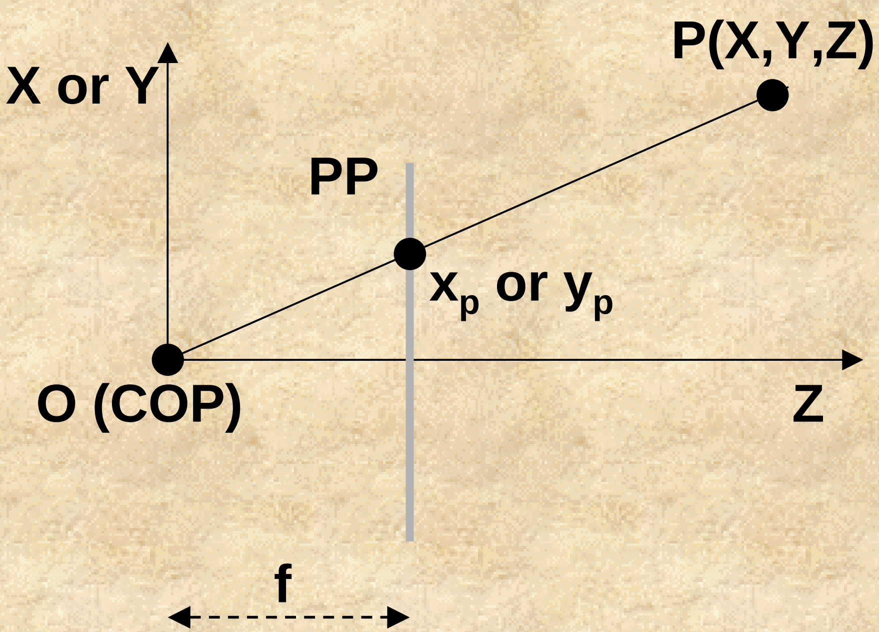
Perspective geometry and Camera Models







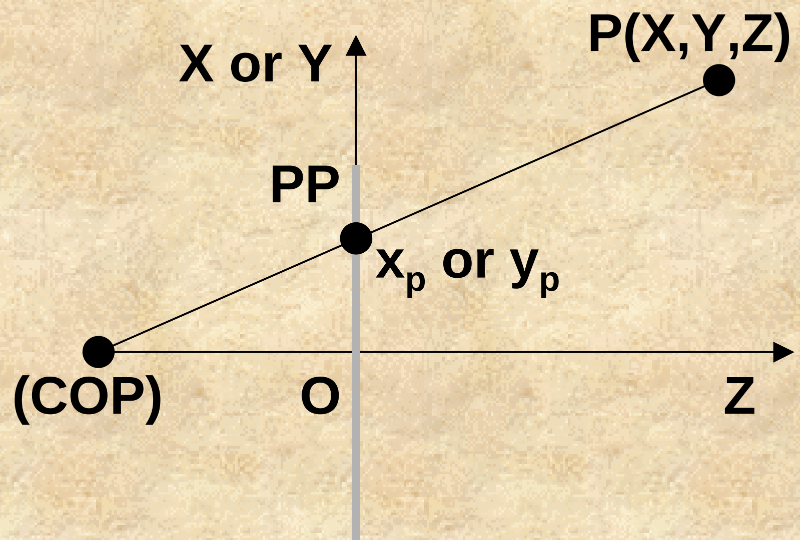
Equations of Perspective geometry



$$x_p / f = X / Z ; \quad y_p / f = Y / Z ;$$

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

$P' = M_{per} . P$; where $P = [X \ Y \ Z \ 1]^T$



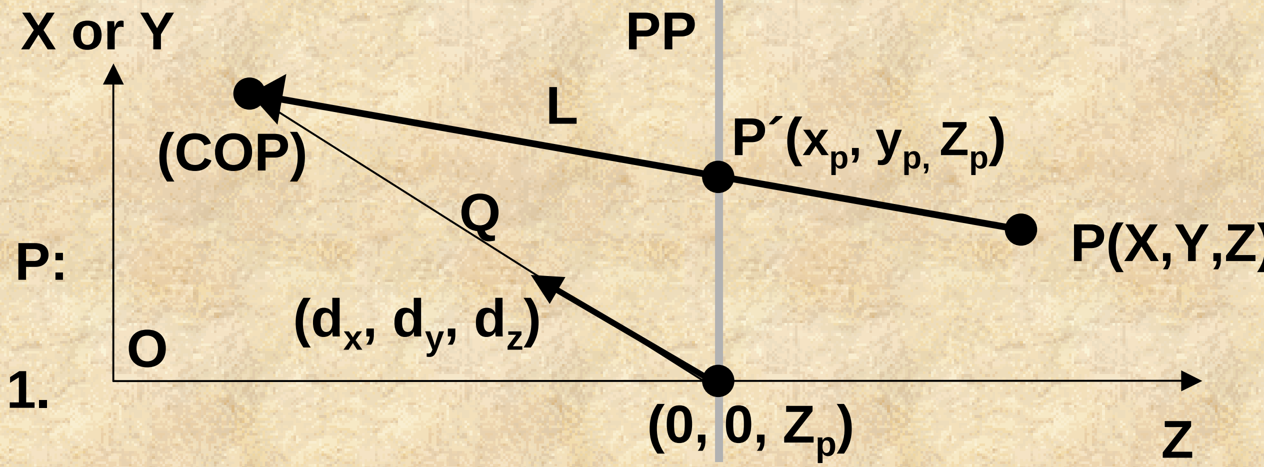
$$\frac{x_p}{f} = \frac{X}{Z+f} ; \quad \frac{y_p}{f} = \frac{Y}{Z+f} ;$$

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

Generalized formulation of perspective projection:

Parametric eqn. of the line L between COP and P:

COP + t(P-COP); $0 \le t \le 1$.



Let the direction vector from $(0, 0, Z_p)$ to COP be (d_x, d_y, d_z), and Q is the distance from (0, 0, Zp) to COP. Then COP = $(0, 0, Z_p) + Q(d_x, d_y, d_z)$.

The coordinates of any point on line L is:

$X' = Qd_x + (X - Qd_x)t;$

$Y' = Qd_y + (Y - Qd_y)t;$

$Z' = (Z_p + Qd_z) + (Z - (Z_p + Qd_z))t;$

Using the condition $Z' = Z_p$, at the intersection of line L and plane PP:

$$t = \frac{-Qd_z}{Z - (Z_p + Qd_z)}$$

Class work:

Now subsitute to obtain, x_p and y_p.

$$x_p = \frac{X - Z\frac{d_x}{d_z} + Z_p\frac{d_x}{d_z}}{\frac{Z_p - Z}{Qd_z} + 1}$$

$$y_p = \frac{Y - Z\frac{d_y}{d_z} + Z_p\frac{d_y}{d_z}}{\frac{Z_p - Z}{Qd_z} + 1}$$

$$M_{gen} = \begin{bmatrix} 1 & 0 & -\frac{d_x}{d_z} & z_p\frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} & z_p\frac{d_y}{d_z} \\ 0 & 0 & -\frac{Z_p}{Qd_z} & \frac{Z_p^2}{Qd_z} + Z_p \\ 0 & 0 & -\frac{1}{Qd_z} & \frac{Z_p}{Qd_z} + 1 \end{bmatrix}$$

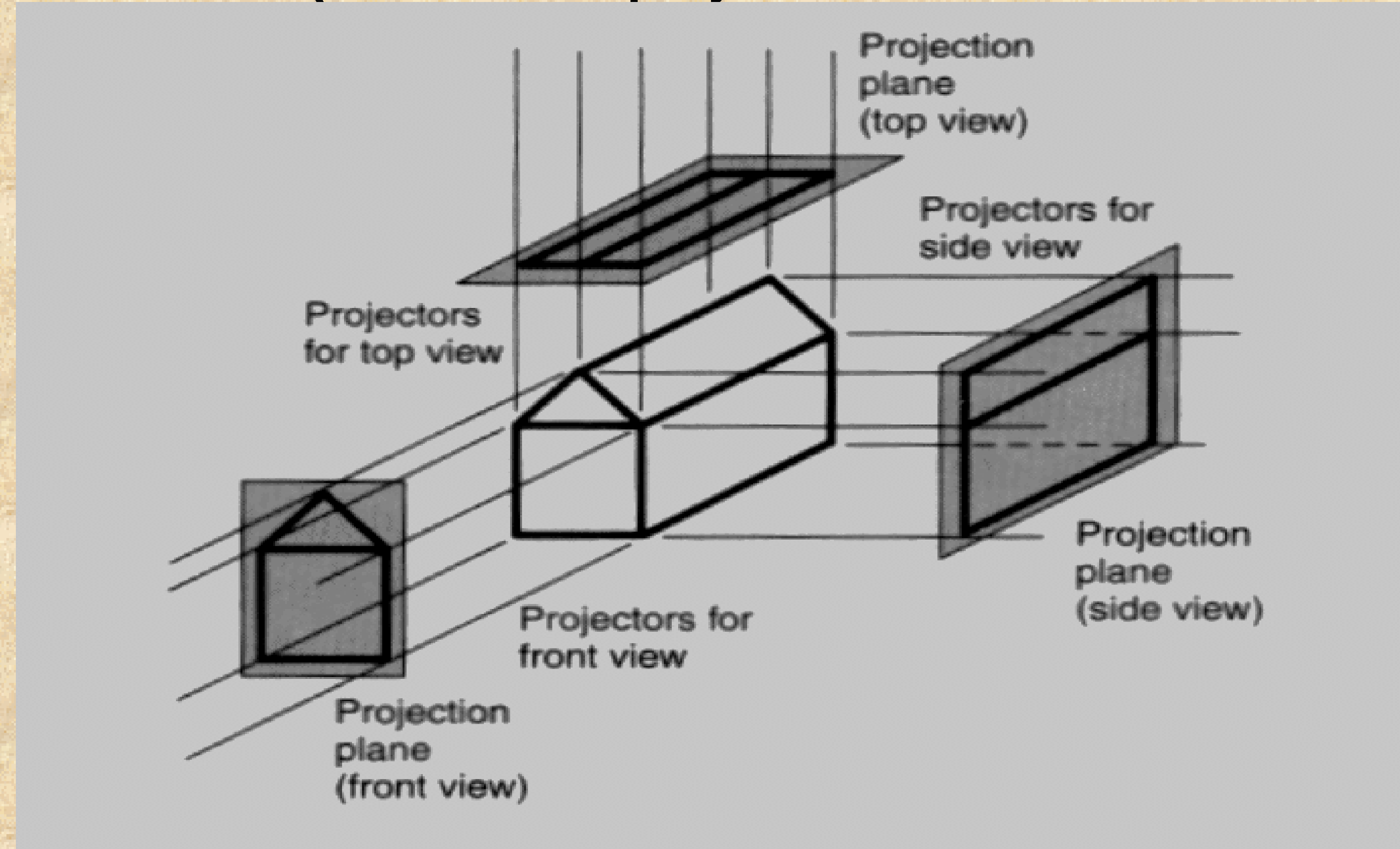
Parallel Projection

Distance from COP to projection plane is infinite.

Therefore, the projectors are parallel lines & we specify a direction of projection (DOP)

Orthographic: the direction of projection and the normal to the projection plane are the same. (direction of projection is normal to the projection plane).

Example of Orthographic Projection:

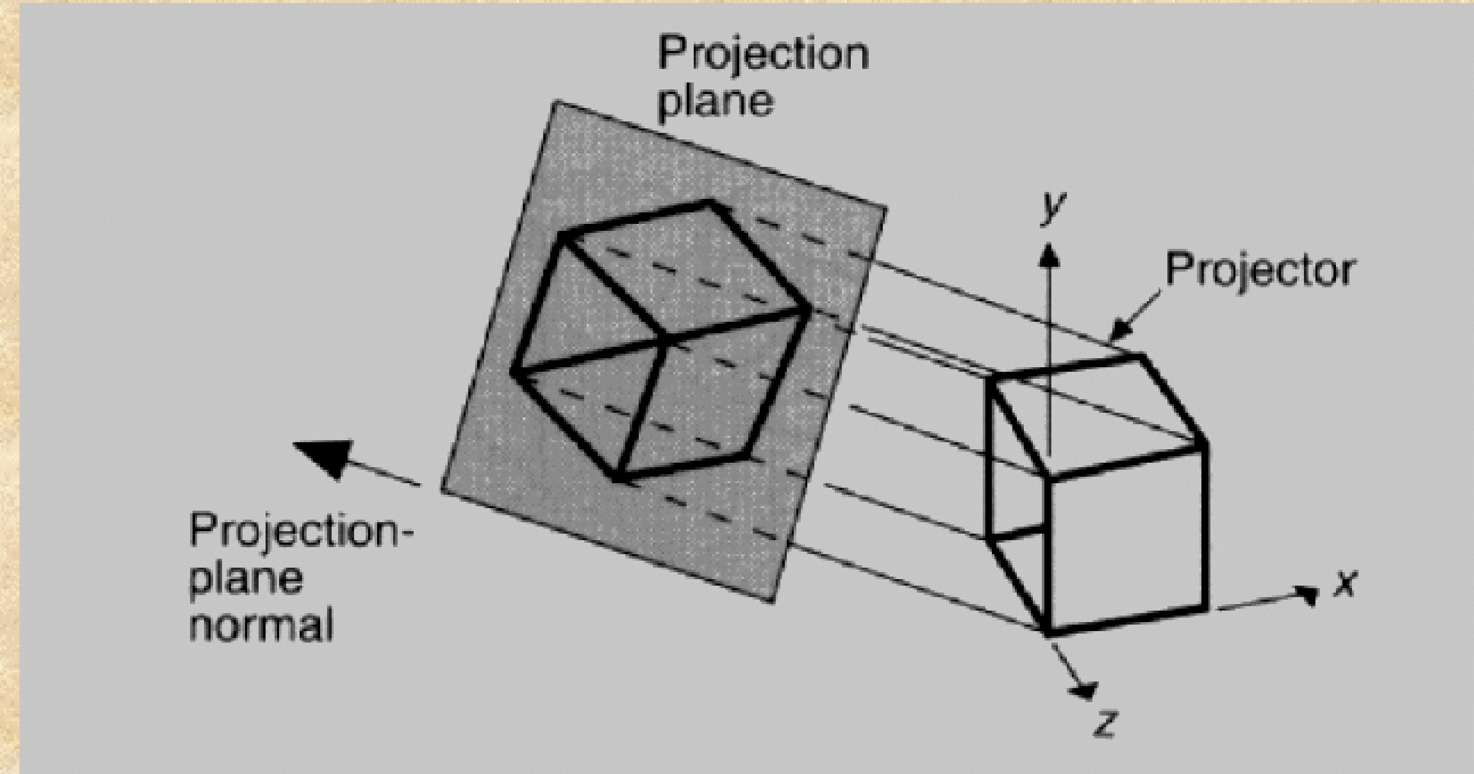


Axometric orthographic projections **use planes of projection that are not normal to a principal axis (they therefore show multiple face of an object.)**

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

All 3 axis are equally foreshortened allowing measurements along the axes to be made with the same scale.

Example Isometric Projection:

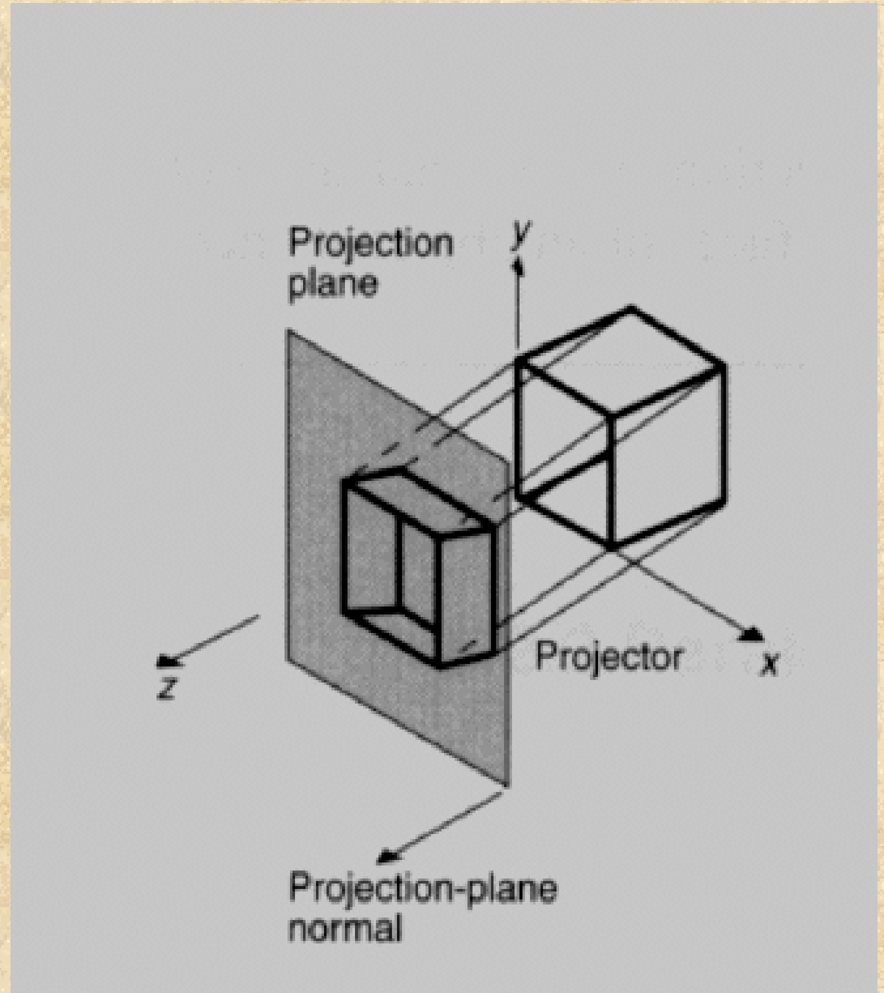


<u>Oblique projections</u> **: projection plane normal and the direction of projection differ.**

Plane of projection is normal to a principle axis

Projectors are not normal to the projection plane

Example Oblique Projection



General oblique projection of a point/line:

Projection Plane: x-y plane; P´ is the projection of P(0, 0, 1) onto x-y plane. `*l*´ is the projection of the z-axis unit vector onto x-y plane and $\alpha$ is the angle the projection makes with the x-axis. When DOP varies, both `*l*´ and α will vary.

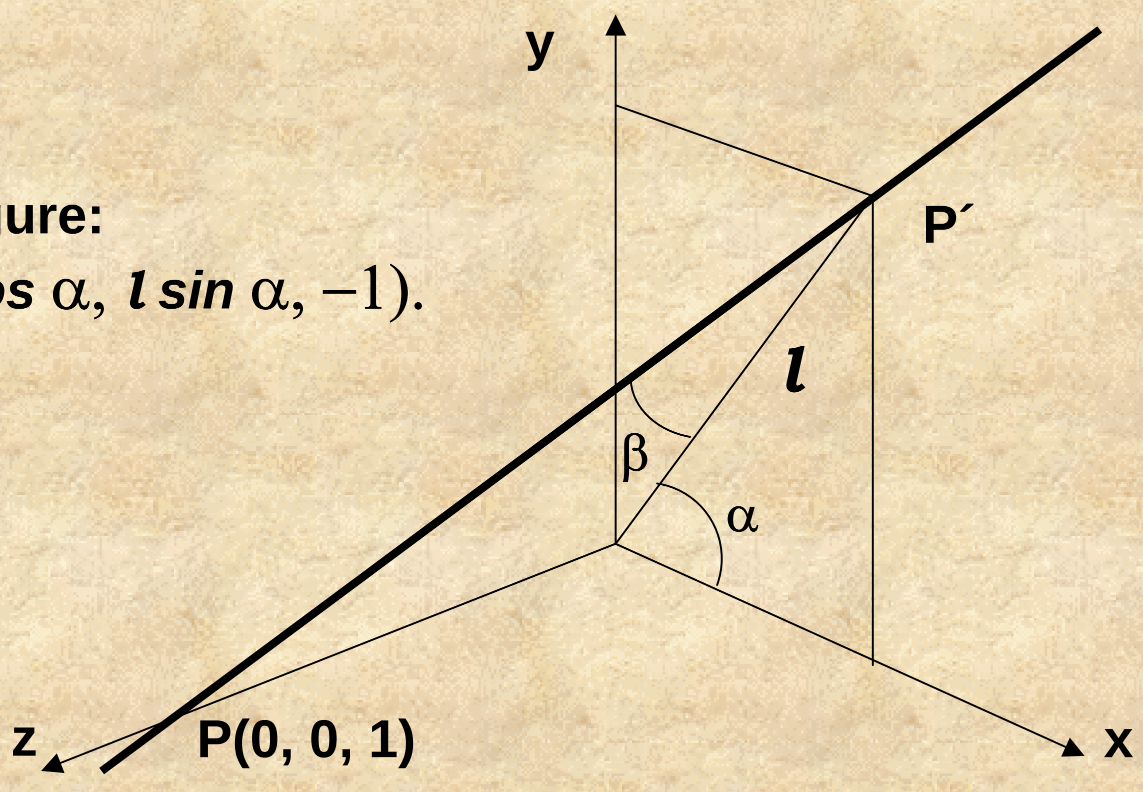
Coordinates of P´:

(*l cos* α, *l sin* α, 0).

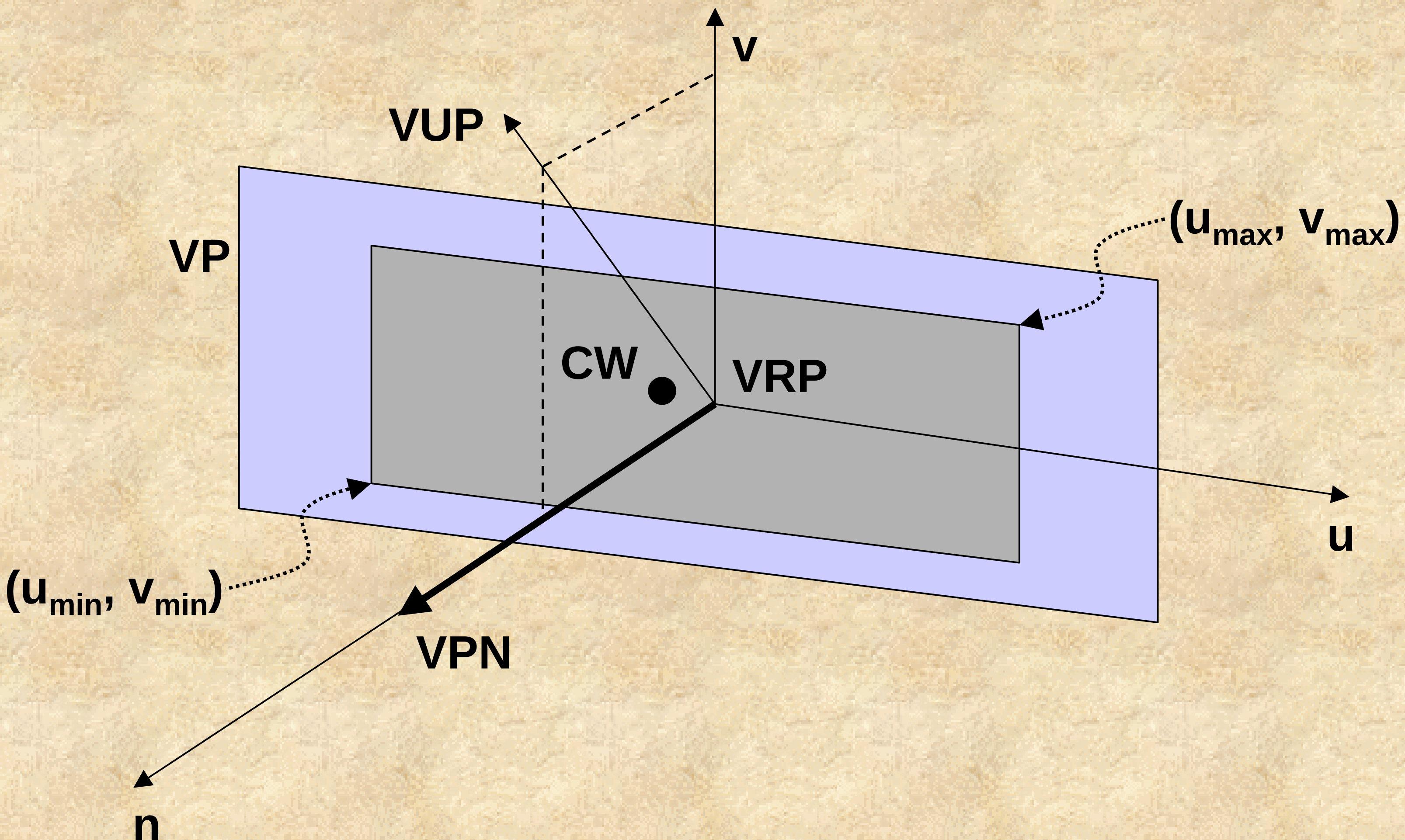
As given in the figure:

DOP is (dx, dy, -1) or (*l cos* α, *l sin* α, -1).

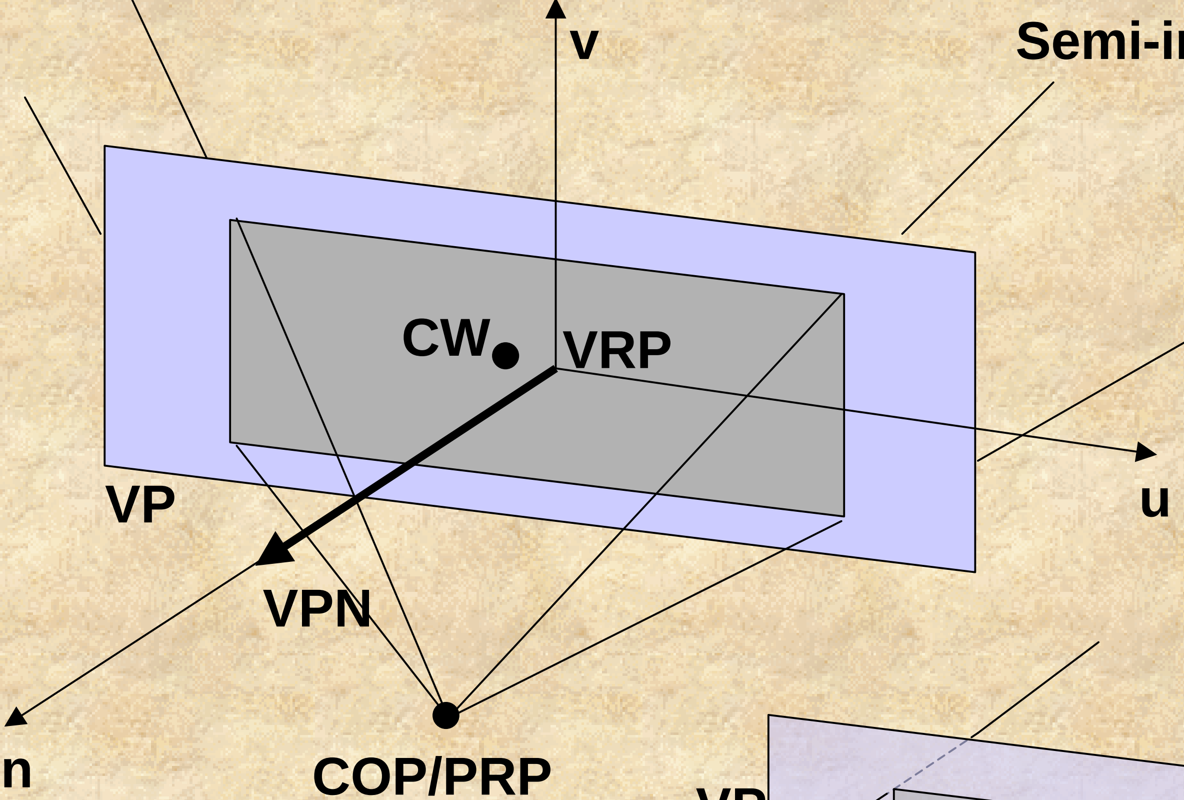
What is β ?

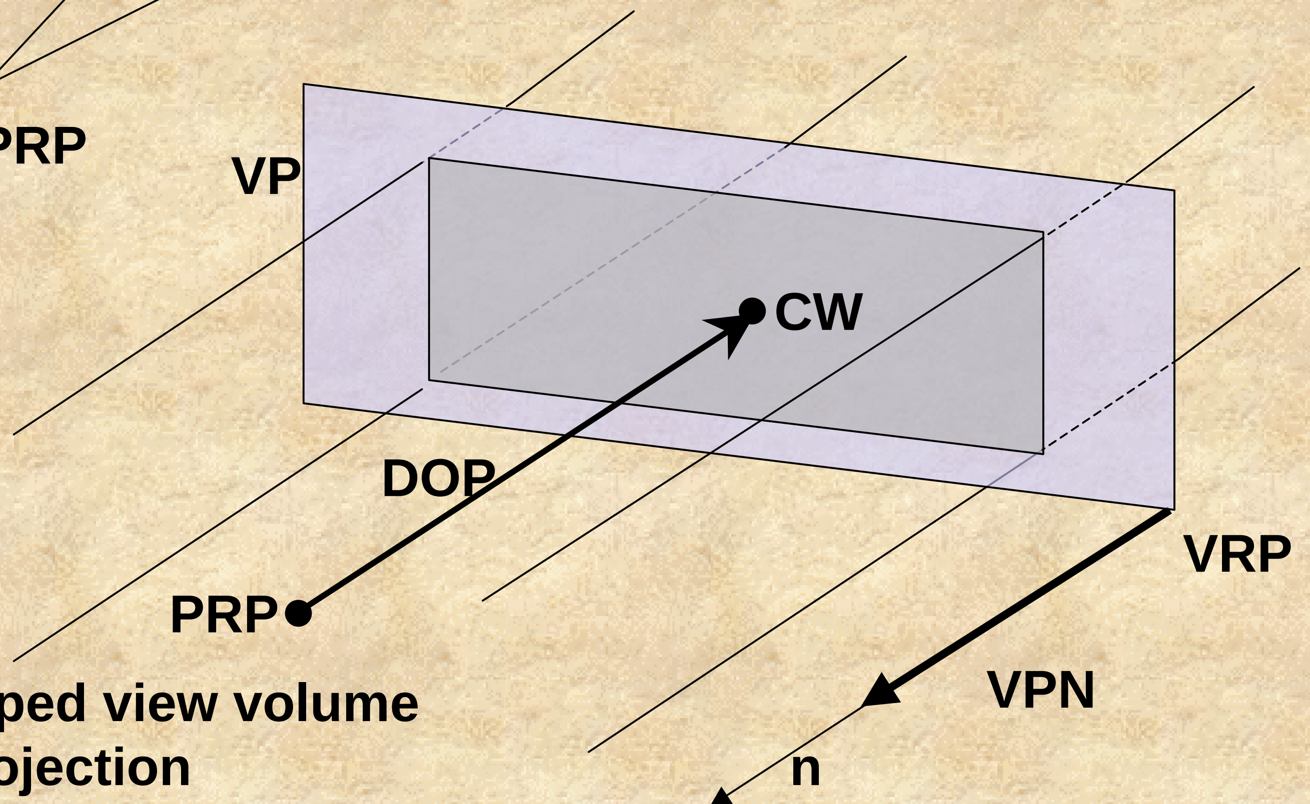


View Specifications:
VP, VRP, VUP, VPN, PRP, DOP, CW, VRC



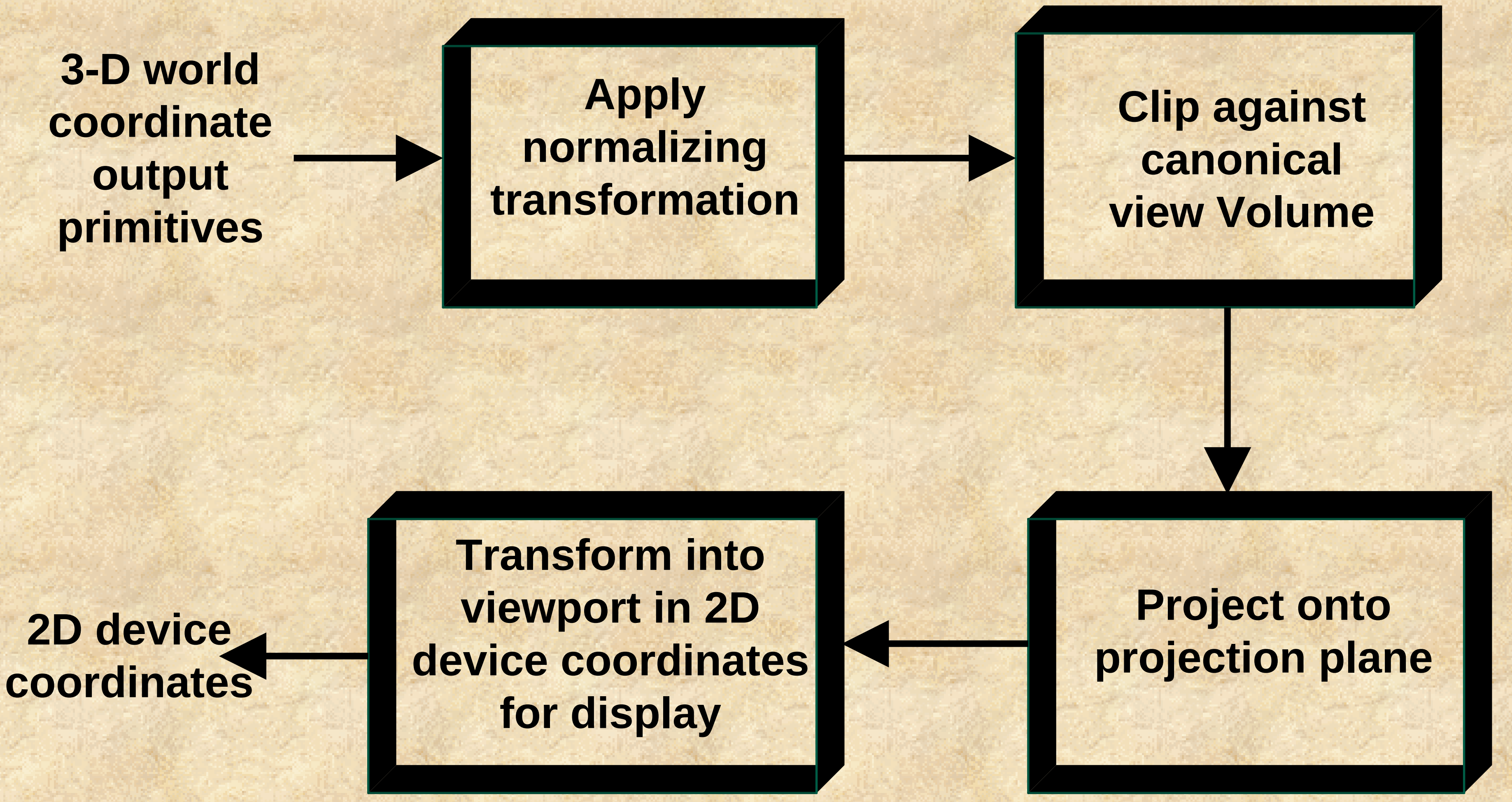
Semi-infinite pyramid view volume for perspective projection





Infinite parallelopiped view volume for perspective projection

Implementation of 3D Viewing

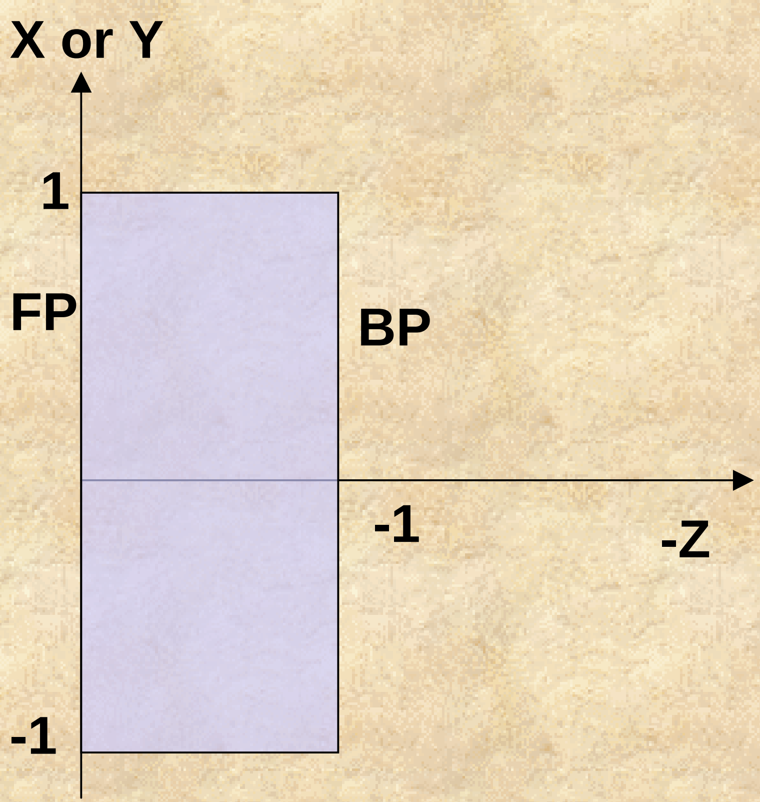


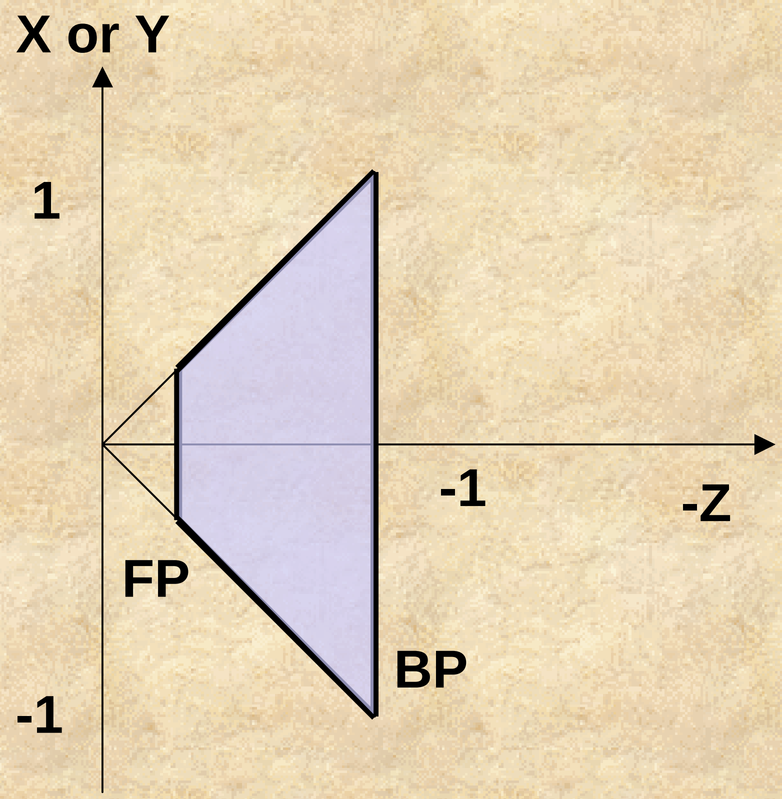
Canonical view volume for *parallel projection* is defined by six planes:

X = -1; X = 1;
Y = -1; Y = 1;
Z = 0; Z = -1.

Canonical view volume for *perspective projection* is defined by six planes:

X = Z; X = -Z;
Y = -Z; Y = Z;
Z = -Z_{min}; Z = -1.





Steps for implementing normalizing transformation matrix for parallel projection

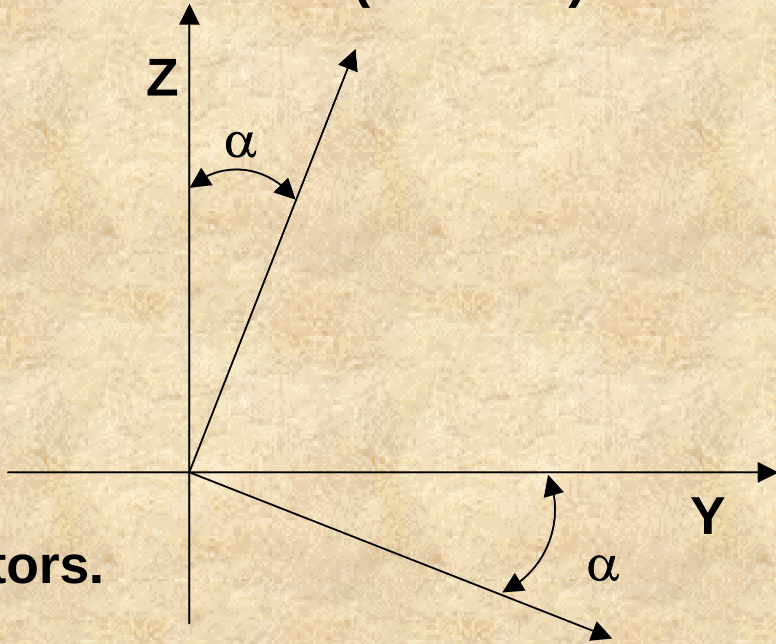
- Translate the VRP to origin
- Rotate VRC such that VPN (n-axis) aligns with Z-axis
  (also, u with X and v with Y)
- Shear (not necessary for pure orthographic) such that DOP is parallel to the Z-axis
- Translate and scale into parallel-projection canonical view volume

Expressions for Step 2 must be derived.

Implement using the concept of combined transformation (rotation).

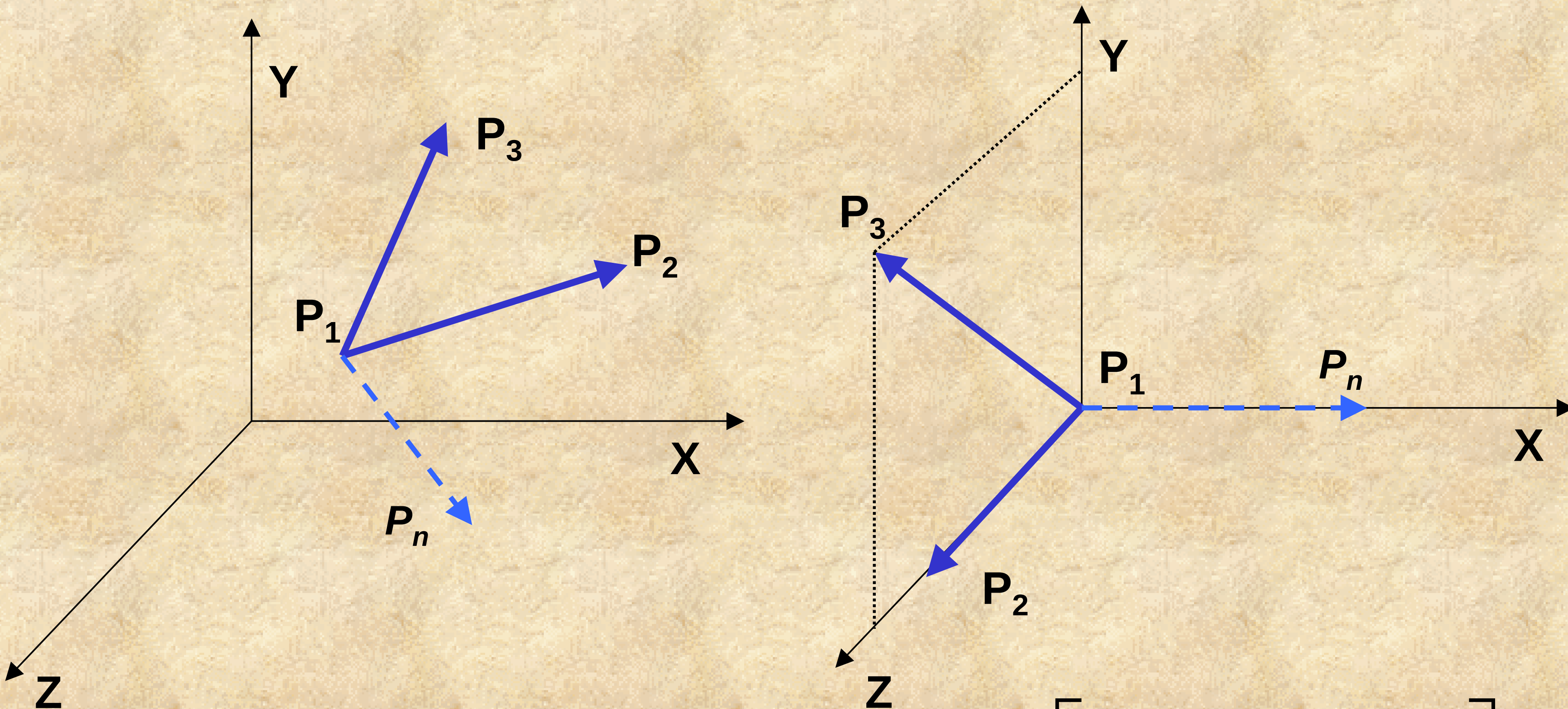
Take R_x =

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- **Rows are unit vectors, when rotated by Rx, will align with the Y and Z axis respectively.**
- **When unit vectors along the principle axes are rotated by Rx, they form the column vectors.**

Consider a general scenario of combined rotations and use the property derived based on the orthogonality of the R matrix.



Let the effective rotation matrix be a combination of three rows as:

$$\begin{bmatrix} r_{1x} & r_{2x} & r_{3x} \\ r_{1y} & r_{2y} & r_{3y} \\ r_{1z} & r_{2z} & r_{3z} \end{bmatrix}$$

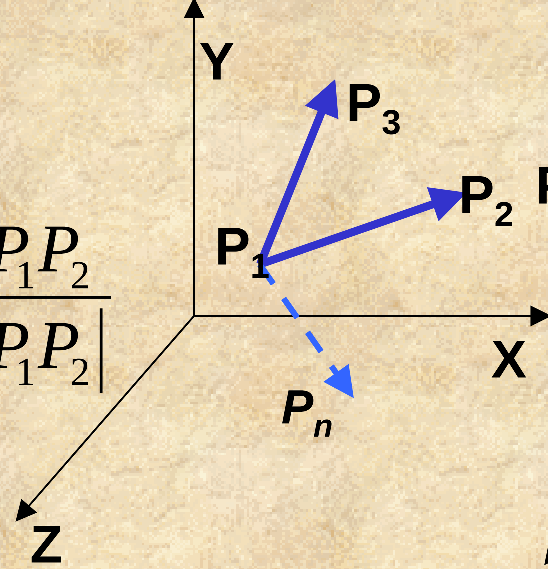
Where,

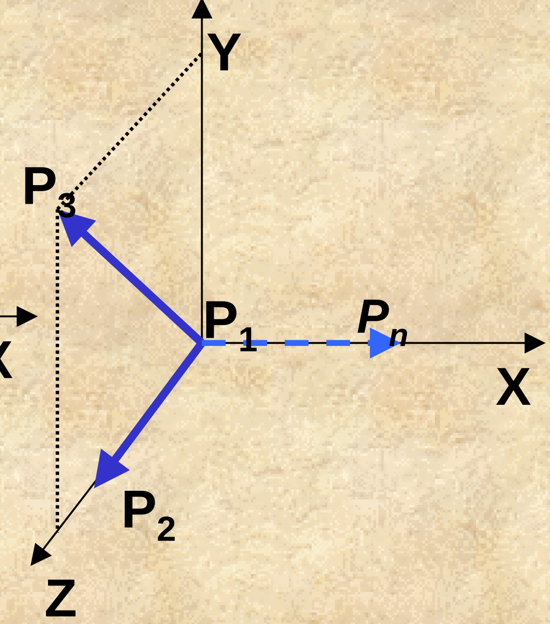
$$R_z = [r_{1z}\ r_{1z}\ r_{1z}]^T = \frac{P_1P_2}{|P_1P_2|}$$

$$R_x = [r_{1x}\ r_{1x}\ r_{1x}]^T = \frac{P_1P_2 \text{ x } P_1P_2}{|P_1P_2 \text{ x } P_1P_2|}$$

and

$$R_y = [r_{1y}\ r_{1y}\ r_{1y}]^T = R_z \text{ x } R_x$$





The rotation matrix of step 2 in normalizing transformations, can be formulated as:

$$R = \begin{bmatrix} r_{1x} & r_{2x} & r_{3x} & 0 \\ r_{1y} & r_{2y} & r_{3y} & 0 \\ r_{1z} & r_{2z} & r_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where,

$$R_z = \frac{VPN}{|VPN|};$$

$$R_x = \frac{VUP \text{ x } R_z}{|VUP \text{ x } R_z|};$$

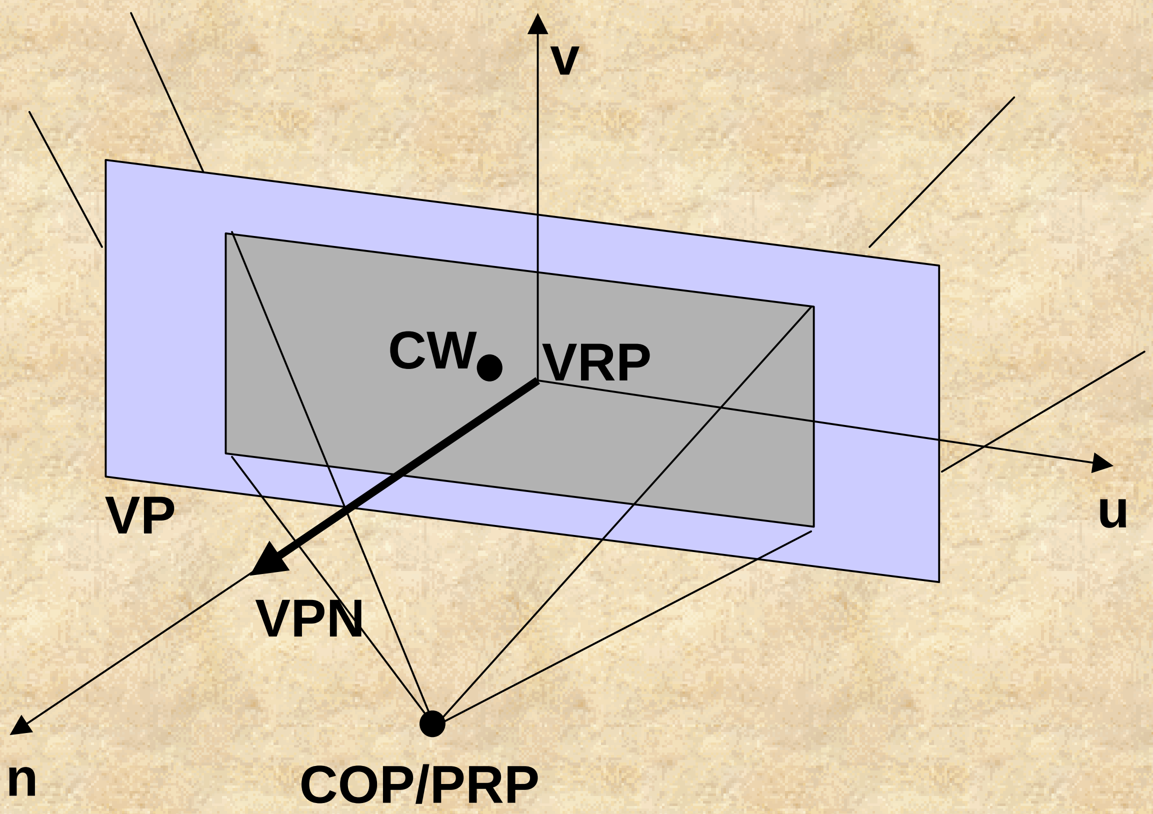
and

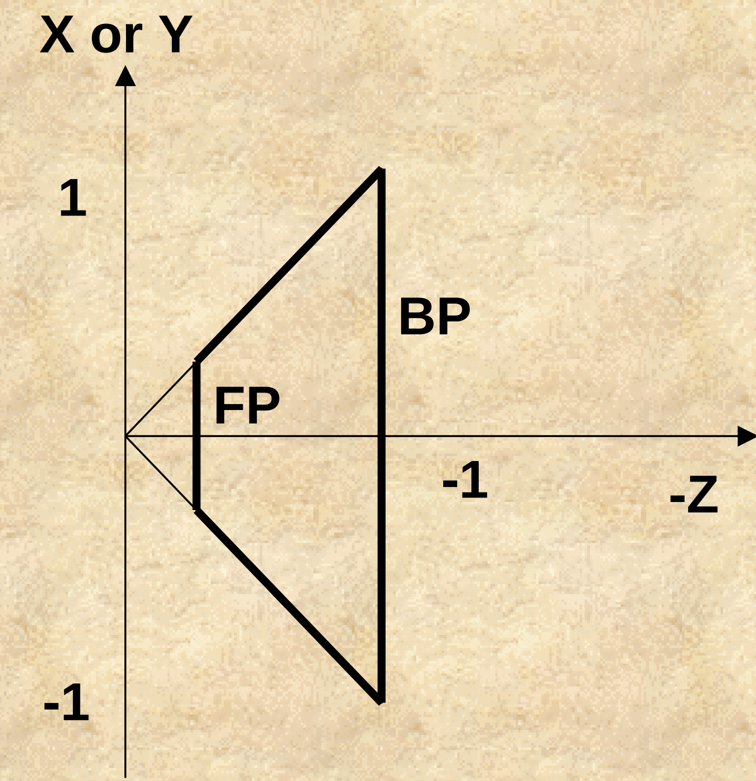
$$R_y = R_z \text{ x } R_x$$

The overall combined transformation matrix
(for WCSVV -> PPCVV), for parallel projection is:

$$N_{par} = S_{par}\ T_{par}\ SH_{par}\ R\,T(-VRP)$$

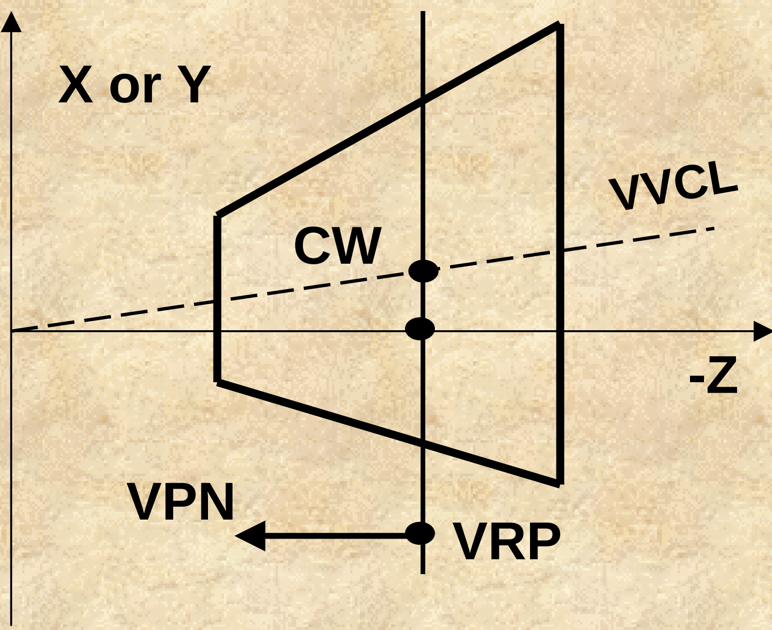
Implementing normalizing transformation matrix for perspective projection





Steps for implementing normalizing transformation matrix for perspective projection

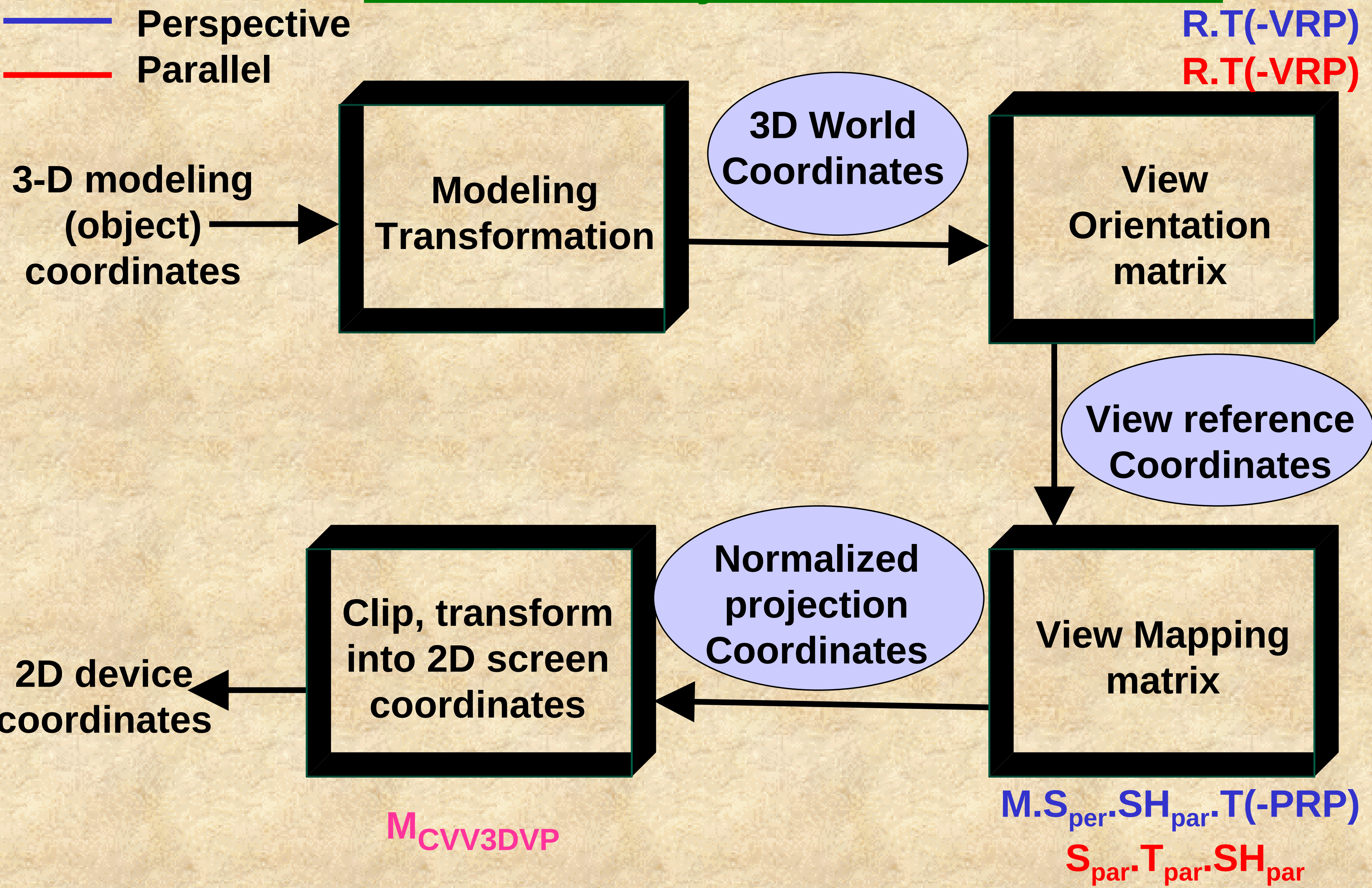
- Translate the VRP to origin
- Rotate VRC such that VPN (n-axis) aligns with Z-axis
  (also, u with X and v with Y)
- Translate such that COP (or PRP) is at the origin
- Shear such that center of line of view volume (VVCL) becomes z-axis
- Scale such that VV becomes the canonical view volume



$$N_{per} =$$

$$S_{per} \; SH_{par} \; T(-PRP) \, R \, T(-VRP)$$

Coordinate Systems and Matrices



where after clipping,
use $M_{CVV3DVP}$ =

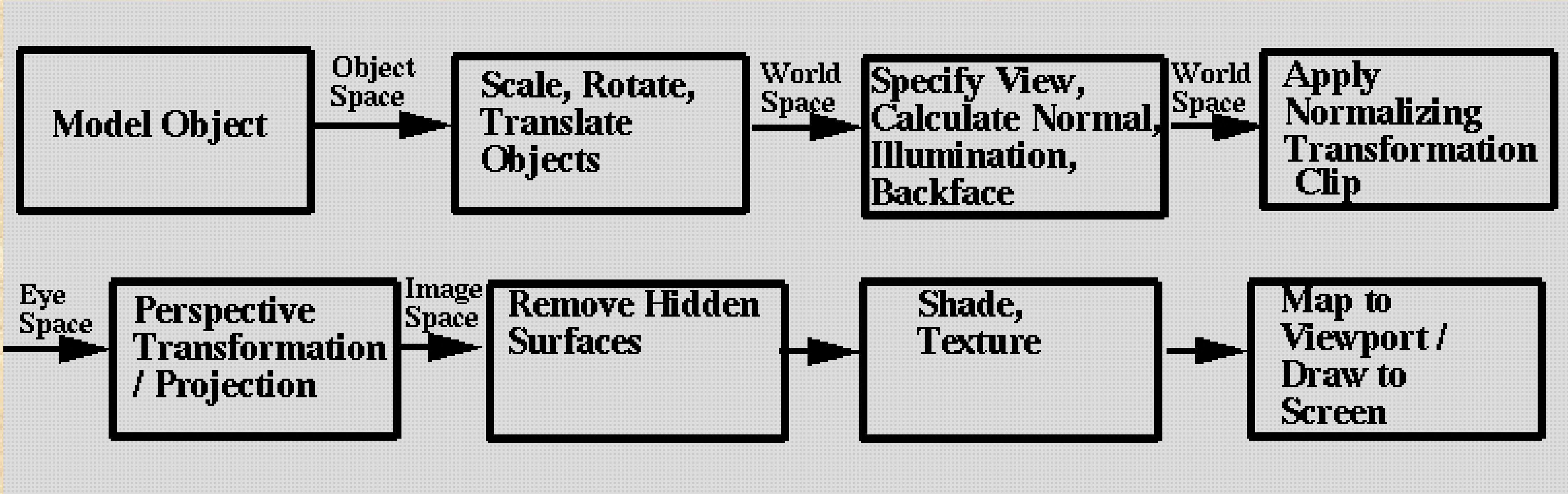
$$T(X_{v\min}, Y_{v\min}, Z_{v\min}) \,.\, S(\frac{X_{v\max} - X_{v\min}}{2}, \frac{Y_{v\max} - Y_{v\min}}{2}, Z_{v\max} - Z_{v\min}) \,.\, T(1,1,1)$$

The 3D Viewing Pipeline

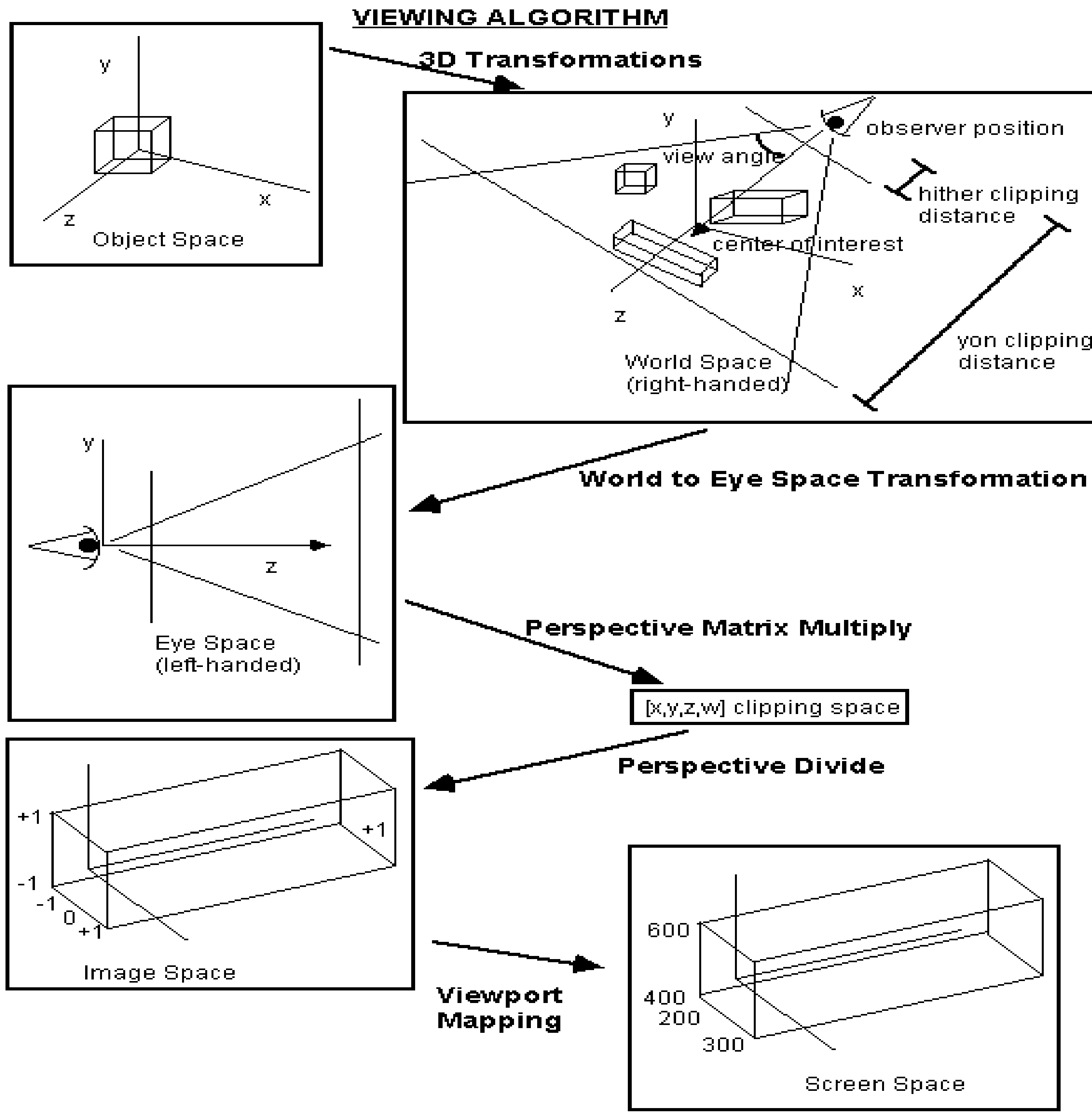
- Objects are modeled in object (modeling) space.
- Transformations are applied to the objects to position them in world space.
- View parameters are specified to define the view volume of the world, a projection plane, and the viewport on the screen.

- Objects are clipped to this View volume.
- The results are projected onto the projection plane (window) and finally mapped into the 3D viewport.

- Hidden objects are then removed.
- The objects are scan converted and shaded if necessary.



The Computer Graphics Pipeline Viewing Process



The Camera Model

We specify our initial camera model by identifying the following parameters.

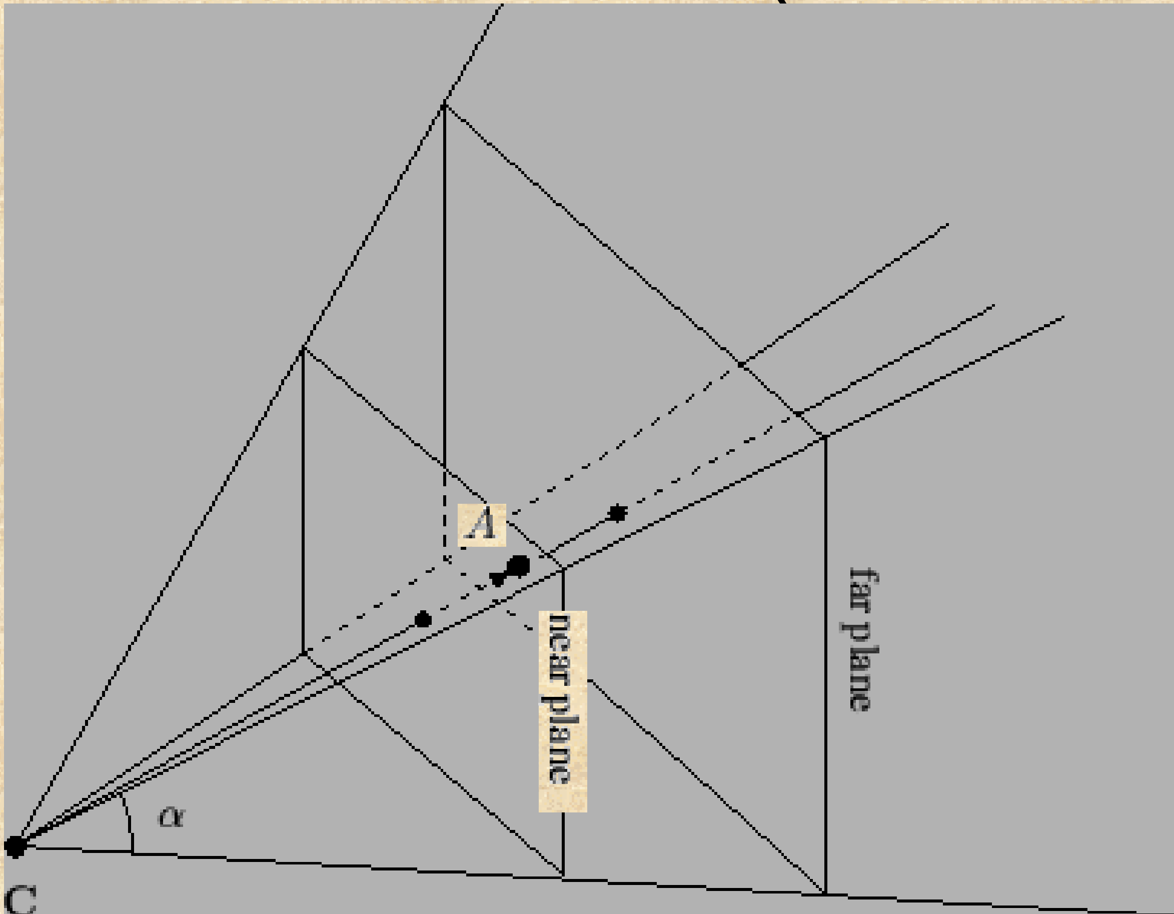
1. A scene, consisting of polygonal elements each represented by their vertices;

2. A point that represents the camera position -- $C = [C_x, C_y, C_z]$,;

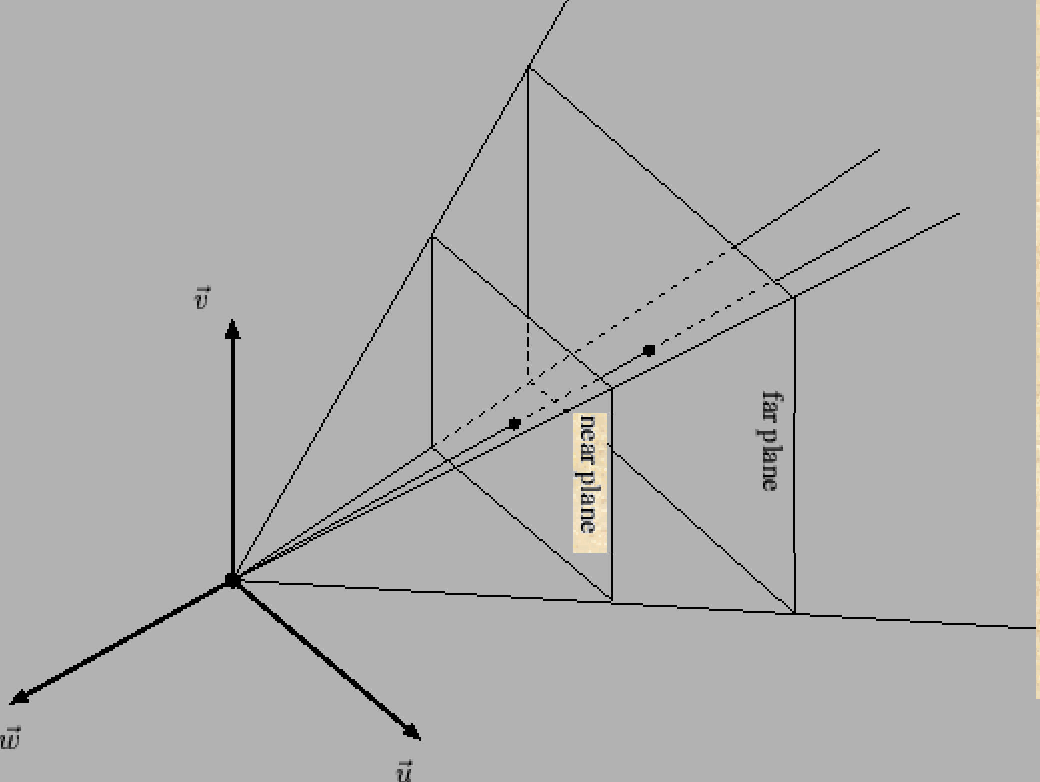
3. A point that represents the ``center-of-attention'' of the camera (i.e. where the camera is looking):
$A = [A_x, A_y, A_z]$;

4. A field-of-view angle, α , representing the angle subtended at the apex of the viewing pyramid.

5. The specification of ``near'' and ``far'' bounding planes. These planes considered perpendicular to the direction-of-view vector at a distance of d_n and d_f from the camera, respectively.

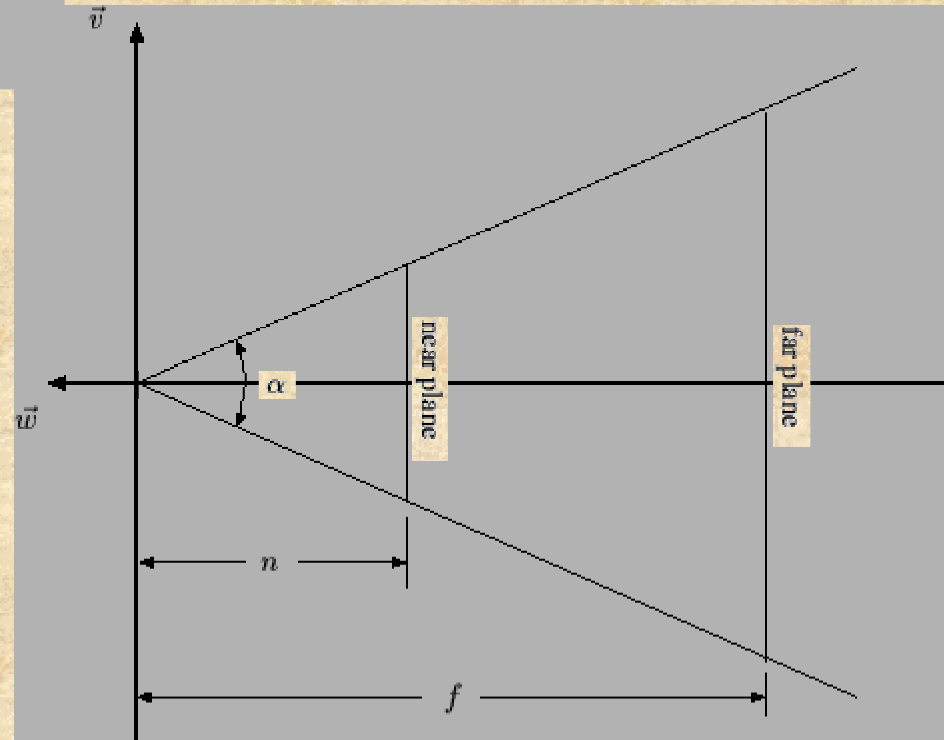


The Viewing Pyramid



3D view of the viewing space

Side view of the viewing space



The image space volume:

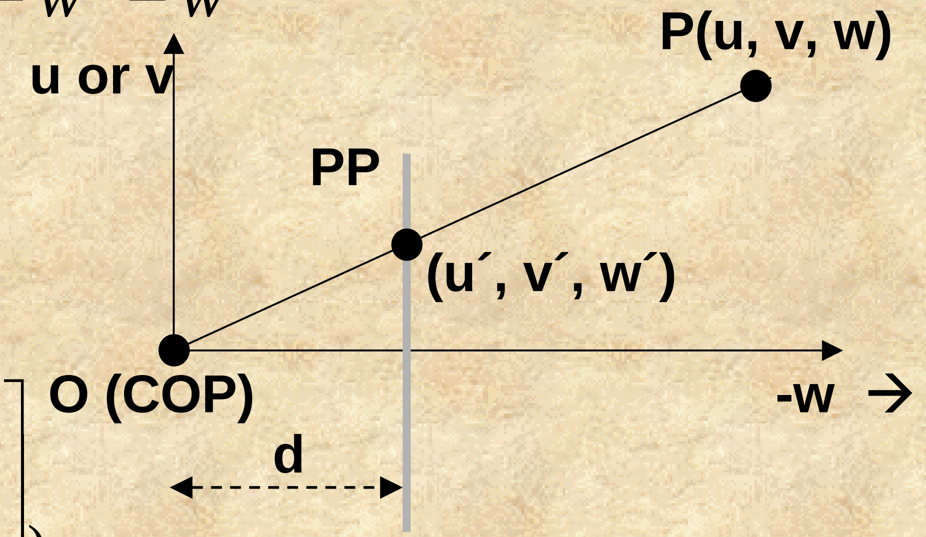
$$-1 \le u, v, w \le 1$$

Derivation of the viewing transformation matrix, in terms of camera parameters:

$$(u, v, w) \rightarrow (\frac{d.u}{-w}, \frac{d.u}{-w}, -d) = (\frac{d.u}{-w}, \frac{d.u}{-w}, \frac{d.w}{-w})$$

Thus,

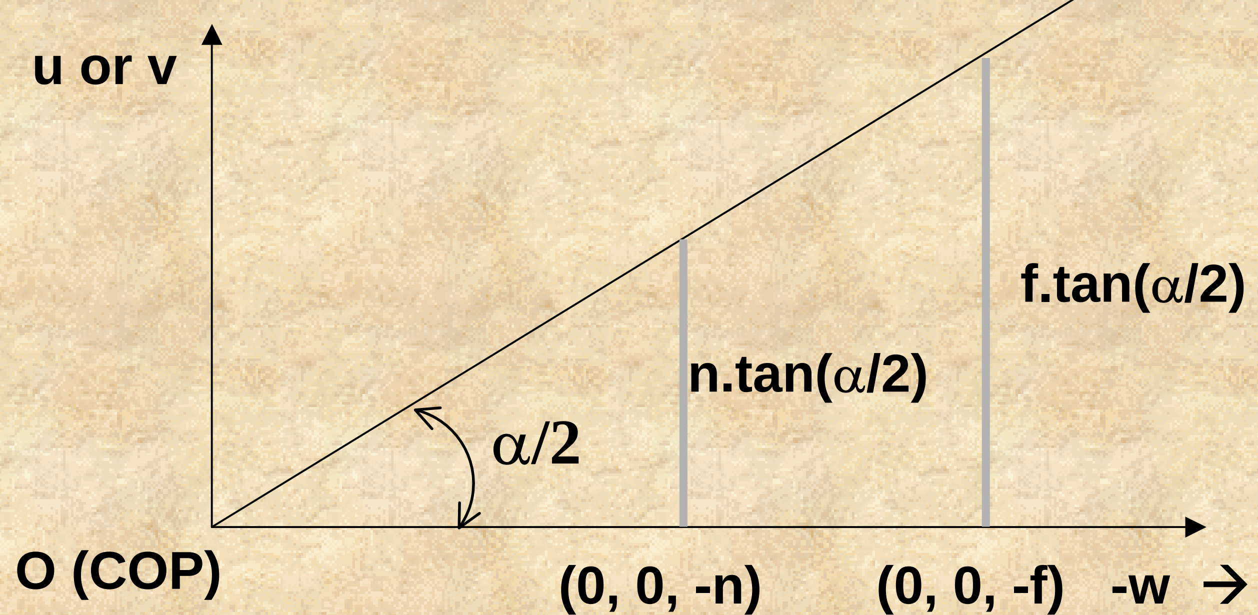
$$(u, v, w, 1) \rightarrow (d.u, d.v, d.w, -w)$$



Express as transformation:

$$P_d = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; (\ or \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{d} \\ 0 & 0 & 0 & 0 \end{bmatrix})$$

$$\begin{bmatrix} u & v & w & 1 \end{bmatrix} \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} d.u & d.v & d.w & -w \end{bmatrix}$$



Transformation of the finite (truncated) viewing pyramid to the cube, $-1 \leq u, v, w \leq 1$. Let us first analyze w-axis only. Use the transformation matrix:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & -1 \\ 0 & 0 & b & 0 \end{bmatrix};$$

such that,
(0, 0, -n)P → (0, 0, 1)
and
(0, 0, -f)P → (0, 0, -1)

Solve for parameters <u>*a and b*</u>, please:

From the constraints of the above two equations:

$$-an + b = n$$

$$and$$

$$-a.f + b = -f$$

The solution:

$$a = \frac{f+n}{f-n};$$

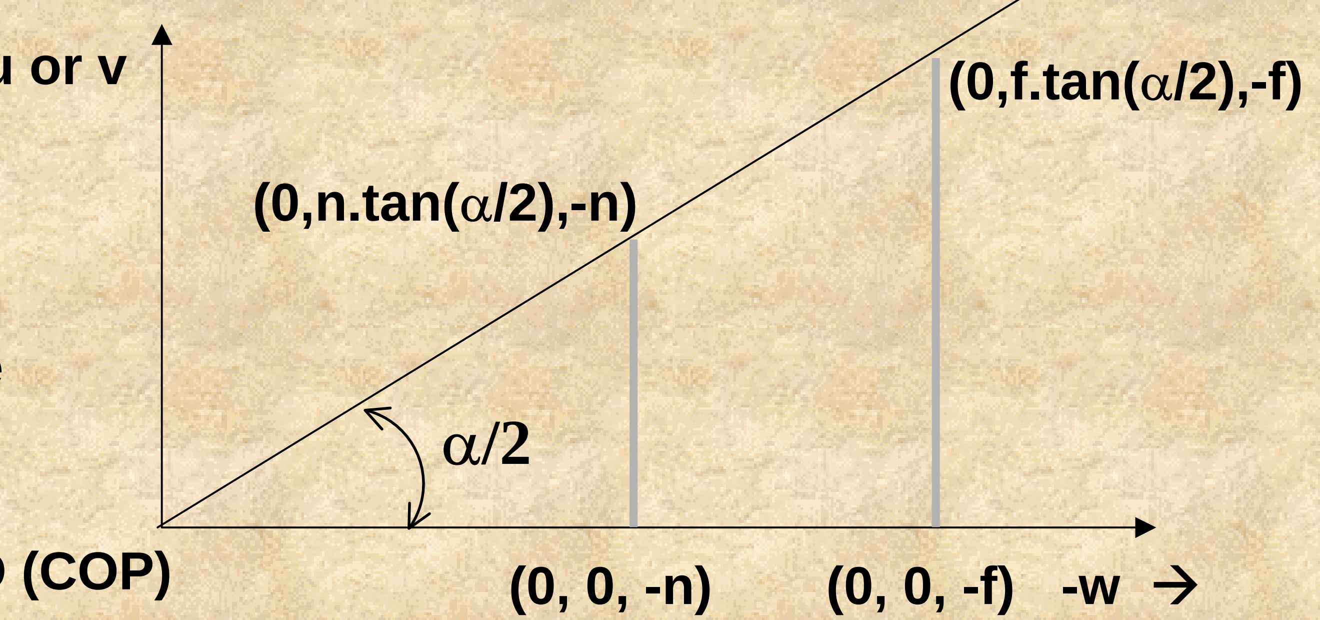
$$b = \frac{2f.n}{f-n}$$

Hence the transformation is:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -1 \\ 0 & 0 & \frac{2f.n}{f-n} & 0 \end{bmatrix}$$

What about u and v-axis transformations in the pyramid ?

Transformations for the two points are as follows:



$$[0 \quad n.\tan(\alpha/2) \quad -n \quad 1]P = \left[0 \quad n.\tan(\alpha/2) \quad -n\frac{f-n}{f+n}+\frac{2nf}{f+n} \quad n\right]$$

$$= [0 \quad n.\tan(\alpha/2) \quad -n \quad n]$$

$$[0 \quad f.\tan(\alpha/2) \quad -f \quad 1]P = \left[0 \quad f.\tan(\alpha/2) \quad -f\frac{f-n}{f+n}+\frac{2nf}{f+n} \quad f\right]$$

$$= [0 \quad f.\tan(\alpha/2) \quad -f \quad f]$$

Desired normalized 3-D coordinates for both the points: [0 1 -1 1]

Thus modify P to be:

$$P = \begin{bmatrix} \cot(\alpha/2) & 0 & 0 & 0 \\ 0 & \cot(\alpha/2) & 0 & 0 \\ 0 & 0 & \dfrac{f+n}{f-n} & -1 \\ 0 & 0 & \dfrac{2f.n}{f-n} & 0 \end{bmatrix}$$

Its inverse has the form:

$$P^{-1} = \begin{bmatrix} \tan(\alpha/2) & 0 & 0 & 0 \\ 0 & \tan(\alpha/2) & 0 & 0 \\ 0 & 0 & 0 & \dfrac{f-n}{2fn} \\ 0 & 0 & -1 & \dfrac{f+n}{2fn} \end{bmatrix}$$

The Viewing Transformation Matrix

$$P_f = P_d . P$$

$$= \begin{bmatrix} d.\cot(\alpha/2) & 0 & 0 & 0 \\ 0 & d.\cot(\alpha/2) & 0 & 0 \\ 0 & 0 & \dfrac{f(d+1)+n(d-1)}{f-n} & -d \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

or

$$= \begin{bmatrix} \cot(\alpha/2) & 0 & 0 & 0 \\ 0 & \cot(\alpha/2) & 0 & 0 \\ 0 & 0 & \dfrac{f+n}{f-n} & \dfrac{1}{d} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

using the regular expression of P_d