

3D Projection



But first...



- Viewports

Return to the graphics pipeline



- 1) Transform points
- 2) Clip
- 3) Window -> viewport

- What does this mean w.r.t. what we've done so far?

Transforms and OpenGL



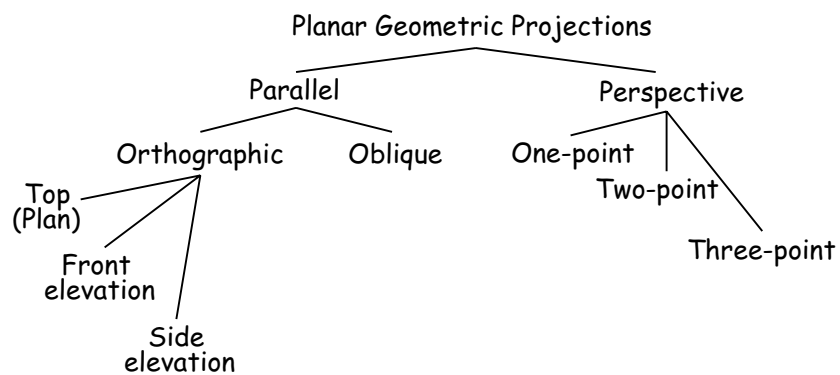
- Transforms go onto matrix "stacks"
 - i.e. `glTranslate`, ...
- All vertices are xformed by the matrices
- Stacks used to store "temporary values"

Why two stacks?

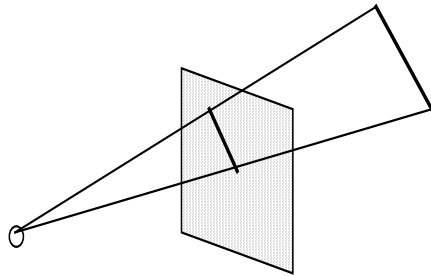
- Want to separate “modeling” transformations from viewing (projection) transforms

Planar projections

Why planar?

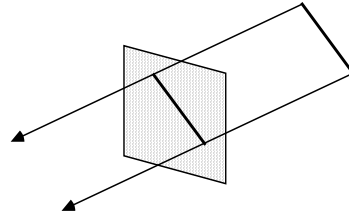


Planar projections



Perspective:

- Distance to CoP is finite
- Forshortening?



Parallel:

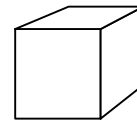
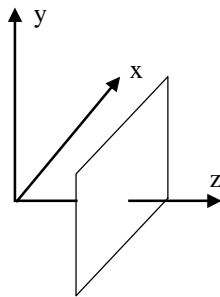
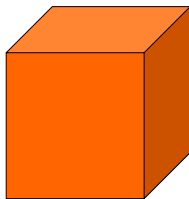
- CoP is at infinity

One-point projection



One-point:

One principle axis cut by projection plane
One axis vanishing point

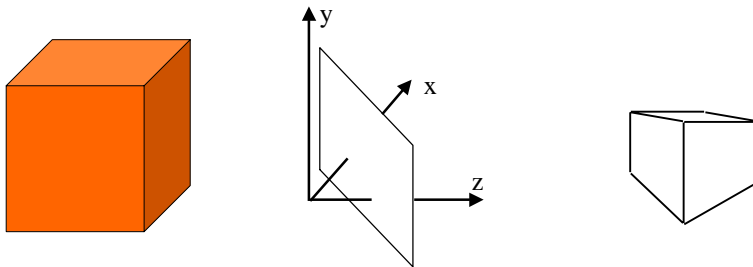


Two-point projection

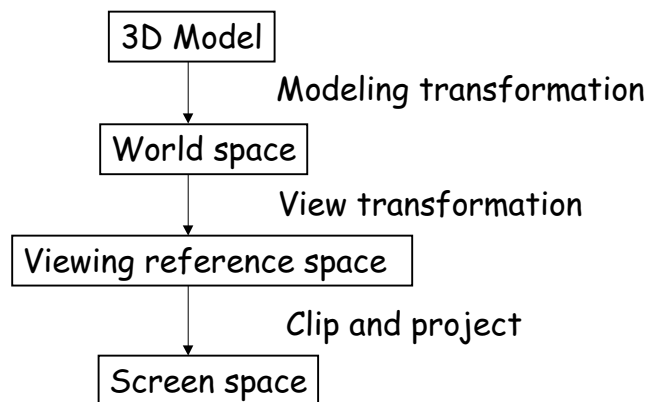
Two-point:

One CoP, but two axis vanishing points

Two principle axes cut by projection plane



3D viewing process



Easy case: just projection



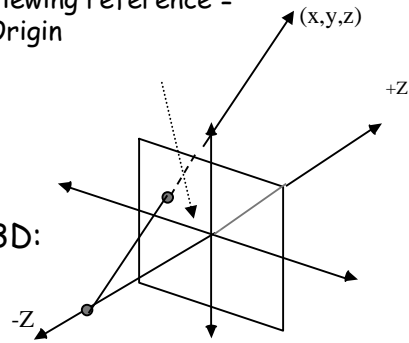
The viewing coordinate is the same as the world

Center of projection =
Viewing reference =
Origin

In homogeneous coordinates,
we can express as:

$$P' = (\text{Projection matrix}) * I_{4 \times 4} * P_{3D}$$

Let $P' = (X, Y, Z, W)$,
divided by W to come back to 3D:
 $(X/W, Y/W, Z/W)$

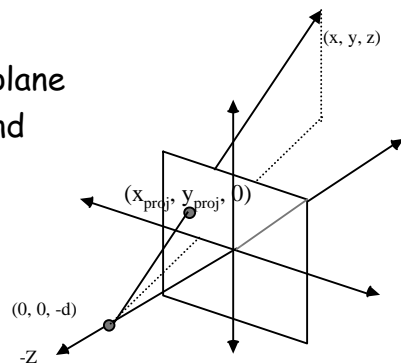


One-point projection



Center of Projection on the
negative z-axis with viewplane
parallel to the x-y plane and
through the origin.

Using similar triangle:



Alternative: moving CoP



Position center of projection at the origin and
the viewplane at $(0,0,d)$

What is the projection matrix?

