Projection Taxonomy

Planar geometric projections

Parallel
- Orthographic
  - Top
  - Side
  - Front
- Axonometric
  - Isometric
  - Other

Perspective
- Oblique
  - One-point (projection plane cuts one axis)
  - Two-point (projection plane cuts two axes)
  - Three-point (projection plane cuts three axes)
Specifying a camera

- **Camera**
  - Tell rendering system where camera is in world coordinates
  - Need to specify focal point and film plane.
  - Convenient to construct a coordinate system for the camera with origin on film plane

- **Clipping volume**
  - We render only a window in the film plane
  - Things beyond any of four sides don’t get rendered
  - Things that are too far away don’t get rendered
  - Things that are too near don’t get rendered.
Specifying a camera

World coordinates

View reference point (VRP)

View-plane normal (VPN)

Project VUP onto film plane to get v

View up vector (VUP)

v

u

n
Specifying a camera

World coordinates

Camera Plane

Projection reference point (PRP)
Specifying a camera

World coordinates

Window given by $(u_{\text{min}}, u_{\text{max}})$, $(v_{\text{min}}, v_{\text{max}})$, denote center by CW

Projection reference point (PRP)
View reference point and view plane normal specify film plane.

Up vector gives an “up” direction in the film plane. Vector v is projection of up vector into film plane ($= n \parallel \text{VUP} \parallel n$).

$u$ is chosen so that $(u, v, n)$ is a right handed coordinate system; i.e. it is possible to rotate $(x->u, y->v, z->n)$ (and we’ll do this shortly).

VRP, VPN, VUP must be in world coords; PRP could be in world coords or in camera coords.
U, V can be used to specify a window in the film plane; only this section of film ends up on the screen.

This window defines four planes; points outside these planes are not rendered.

Hither and yon clipping planes, which are always given in terms of camera coordinates, and always parallel to the film plane, give a volume - known as the view frustum.

Orthographic case: - view frustum is cuboid (i.e. all angles right angles, but edges not necessarily of equal length).
Camera configuration in world coordinates

Transform to standard camera frame

Clip in 3D

Project to 2D using camera model

Render 2D polygons using lighting info

Determine what's in front

Object

Transform object from world coords to camera coords

Clip against view frustum

Project using standard camera model

Lighting information
• Advantages of clipping against view frustrum:
  – Don’t project objects that aren’t drawn
  – cf. clip against hither/yon, project, clip against window in film plane
  – hence slightly less work.

• Advantage of clipping in camera frame (rather than in world frame):
  – Better supports transform to standard view frustrum, where clipping is easiest.

• Advantage of transforming to camera frame:
  – Easiest to compute the effects of the camera in this frame.
• If we clip against the frustum blindly, clipping is hard - this is because planes bounding the frustum have a complex form.

• Thus, to test in/out, must test the sign of \( ax + by + cz + d \) for some \( a, b, c, d \) - much worse than a simple compare.

• Solution: transform view frustrum into a canonical form, where clip planes have easy form - e.g. \( z=x \), \( z=-x \), \( z=y \), \( z=-y \), \( z=-1 \), \( z=d \).
Canonical Frustum

If image plane transforms to \( z = m \) then in new frame, projection is easy:

\[(x, y, z) \rightarrow (m x/z, m y/z)\]
Transform object from world coords to camera coords

Further transform so that frustum is canonical frustum.

Object

Transform object from world coords to camera coords

Clip against view frustrum

Project using standard camera model

Transform object from world coords to camera coords
Step 1. Translate VRP to world origin. Call this $T_1$. $T_1$ maps world points (note opposite transformations for object and coordinate frame).
Step 2. Rotate camera coordinate frame so that u is x, v is y, and n is z. The matrix is ?
Step 2. Rotate camera coordinate frame so that $u$ is $x$, $v$ is $y$, and $n$ is $z$. The matrix is:

$$
\begin{bmatrix}
  u^T & 0 \\
  v^T & 0 \\
  n^T & 0 \\
  0 & 0 \\
  0 & 0 \\
  0 & 1 
\end{bmatrix}
$$

(why?)