3D Graphics Concepts
(Watt ch. 5, Foley et al ch. 6)

- Modeling: For now, objects will be collections of polygons in 3D. Complex shapes will be many small polygons.
- Issues:
  - Which polygons can be seen? (some polygons hide others, and some are outside the relevant volume of space and need to be clipped).
  - Where do they go in the 2D image? (key abstraction is a virtual camera)
  - How bright should they be? (for example, to make it look as if we are looking at a real surface)
Today

Camera information (world coordinates)

Objects (world coords)

Transform to standard camera coordinates

Clip in 3D (against canonical view frustum)

Project to 2D with standard camera model

Render 2D polygons

Determine what is in front*

* different parts have different positions in the pipeline, depending on strategy

Lighting information (was in world coordinates, but now transformed)
Pinhole cameras

- Abstract camera model--box with a small hole in it
- Pinhole cameras work for deriving algorithms--a real camera needs a lens
Distant objects are smaller
Parallel lines meet*

Common to draw film plane
_in front_of the focal point
(virtual image)

*Exceptions?
Vanishing points

• Each set of parallel lines (=direction) meets at a different point
  – The vanishing point for this direction

• Sets of parallel lines on the same plane lead to collinear vanishing points.
  – The line is called the horizon for that plane
  – Standard horizon is the horizon of the ground plane.

• One way to spot fake images
Projections

- Mathematical definition of a projection: $PP = P$
- (Doing it a second time has no effect).
- Generally rank deficient (non-invertable)--exception is $P=I$
- Transformation loosees information (e.g., depth)
- Given a 2D image, there are many 3D worlds that could have led to it.
Projections

Center of projection

Projectors
Parallel Projection

Projectors (parallel)

Center of projection at infinity
Parallel Projection

Parallel lines remain parallel, some 3D measurements can be made using 2D picture

Does not give realistic 3D view because eye is more like perspective projection.

If projection plane is perpendicular to projectors the projection is *orthographic*
Orthographic example (onto $z=0$)

$(x, y, z) \rightarrow ?$
Orthographic example (onto z=0)

\[(x, y, z) \rightarrow (x, y, 0)\]
The camera matrix
The camera matrix

\[
\begin{bmatrix}
x \\ y \\ 0 \\ w \\
\end{bmatrix} = \begin{bmatrix}
x \\ y \\ 0 \\ 1 \\
\end{bmatrix} \begin{bmatrix}
x \\ y \\ z \\ w \\
\end{bmatrix}
\]
Perspective example (onto $z=f$)

$$(x, y, z) \rightarrow ?$$
Perspective example (onto $z=f$)

By similar triangles,

$$(x, y, z) \rightarrow (f \frac{x}{z}, f \frac{y}{z}, f)$$
The equation of projection

• In homogeneous coordinates

$$(x, \ y, \ z, \ 1) \ [f \frac{x}{z}, \ f \frac{y}{z}, \ f, \ 1)$$

• Equivalently

$$(x, \ y, \ z, \ 1) \ [x, \ y, \ z, \ \frac{z}{f})$$

• (Now H.C. are being used to store foreshortening)
The camera matrix

\[
\begin{bmatrix}
x \\
y \\
z & = \\
z \\
f
\end{bmatrix} =
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
The camera matrix
Projection Taxonomy

Planar geometric projections

Parallel

Orthographic
  Top
  Side
  Front

Axonometric
  Isometric

Oblique

One-point
  (projection plane cuts one axis)

Two-point
  (projection plane cuts two axes)

Three-point
  (projection plane cuts three axes)

Perspective

Other
Specifying a camera

- **Camera**
  - Tell rendering system where camera is in world coordinates
  - We want to transform the world into camera coordinates so that projection is easy (i.e., we can use the projection matrix from a few slides back).
  - 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

u

v

x

y

z
Specifying a camera

• Why use VUP?
  – Convenient for the user but there are other ways (OpenGL has several ways to negotiate camera parameters, including very much how we are doing it).
  – A world centric coordinate system is natural for the user. In particular, the user may think in terms of the camera rotation around the axis (n) relative to a natural horizon and/or “up” direction.
  – This will mean that VUP cannot be parallel to n. Often one will fix VUP (e.g. to the Y-axis) but this is too restrictive for some applications.

• Why use a “backwards” pointing n?
  – It is more natural to make the camera direction point the other way, but this makes the camera coordinates left handed. (You will see it done both ways).
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)