3D Graphics Concepts
(H&B ch, 7, 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)
Objects (world coords)

Transform to standard camera coordinates

Clip in 3D (against canonical view frustum)

Project to 2D with standard camera model

Determine what is in front*

Render 2D polygons

Camera information (world coordinates)

Back face culling

Today

* 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 looses information (e.g., depth)
- Given a 2D image, there are many 3D worlds that could have lead 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) -> ?
Orthographic example (onto $z=0$)

$$(x, y, z) \rightarrow (x, y, 0)$$
Can we do this with a linear transformation?

\[
\begin{bmatrix}
x \\
y \\
0 \\
\end{bmatrix} = \begin{bmatrix}
? \\
\end{bmatrix}
\]
Can we do this with a linear transformation?

\[
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
\end{bmatrix}
= 
\begin{bmatrix}
x \\
y \\
z \\
\end{bmatrix}
\]
Camera matrix in homogeneous coordinates

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

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

By similar triangles,

$$(x, y, z) \rightarrow (f \frac{x}{z}, f \frac{y}{z}, f)$$
Can we do this with a linear transformation?

\[
\begin{align*}
  f(x) &= 0 \\
  f(y) &= 0 \\
  f(z) &= 0
\end{align*}
\]
Can we do this with a linear transformation?
Now try homogeneous coordinates

- Of course, in homogeneous coordinates

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

- Equivalently

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

- (Now H.C. are being used to store foreshortening)
Is there a linear transformation?

\[
\begin{align*}
\begin{bmatrix} x \\ y \\ z \end{bmatrix} & = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \\
\begin{bmatrix} x \\ y \\ z \\ f \end{bmatrix} & \quad ? \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
\end{align*}
\]
The camera matrix

\[
\begin{pmatrix}
x \\ y \\ z \\ \frac{f}{z}
\end{pmatrix} =
\begin{pmatrix}
x \\ y \\ z \\ \frac{1}{f}
\end{pmatrix}
\]
Summary so far

- Parallel and perspective are projections (non-invertable)
- Perspective viewing is not a linear transform in \((x,y,z)\)
- Perspective viewing can be encoded as a linear transformation in homogenous coordinates
- Perspective becomes parallel projection as \(f\) becomes infinite
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)

Details Optional
Specifying a camera

- Makes sense to 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
- Taken together, this means that everything outside a truncated pyramid (frustum) is irrelevant (lots more on this soon).
Camera information (world coordinates)

Object (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)

Next topic
Object in world coordinates (after modeling transforms)

Transform object from world coordinates to standard camera coordinates

Clip against canonical view frustum

Project using standard camera model

Next topic
Specifying a camera

World coordinates

View reference point (VRP)

View-up vector (VUP)

View-plane normal (VPN)

Project VUP onto film plane to get v

x

y

z

u

v

n
Specifying a camera

World coordinates

Camera Plane

Projection reference point (PRP)