Introduction to Computer Graphics

5. Viewing in 3D (A)

National Chiao Tung Univ, Taiwan
By: I-Chen Lin, Assistant Professor

Outline

- Classical views
- Computer viewing
- Projection matrices
Classical Viewing

- Viewing requires three basic elements
  - One or more objects
  - A viewer with a projection surface
  - Projectors that go from the object(s) to the projection surface

- Each object is assumed to be constructed from flat *principal faces*
  - Buildings, polyhedra, manufactured objects
Planar Geometric Projections

- Standard projections project onto a plane.

- Projectors are lines that either
  - converge at a center of projection
  - are parallel

- Such projections preserve lines
  - but not necessarily angles

- When do we need non-planar projections?
Classical Projections

Front elevation
Elevation oblique
Plan oblique
Isometric
One-point perspective
Three-point perspective
Perspective vs Parallel

- Classical viewing developed different techniques for drawing each type of projection

- Mathematically parallel viewing is the limit of perspective viewing

- Computer graphics treats all projections the same and implements them with a single pipeline
Taxonomy of Planar Geometric Projections

planar geometric projections

parallel

multiview
orthographic

axonometric

oblique

1 point

2 point

3 point

perspective

isometric
dimetric
trimetric
Perspective Projection

Object

Projector

Projection plane

COP
Parallel Projection
Orthographic Projection

- Projectors are orthogonal to projection surface
**Multiview Orthographic Projection**

- Projection plane parallel to principal faces
- Usually form front, top, side views

Isometric (not multiview orthographic view)

In CAD and architecture, we often display three multiviews plus isometric.

Front, Top, Side views.
Advantages and Disadvantages

- Preserves both distances and angles
  - Shapes preserved
  - Can be used for measurements
    - Building plans
    - Manuals

- Cannot see what object really looks like because many surfaces hidden from view
  - Often we add the isometric
Axonometric Projections

- Allow projection plane to move relative to object

classify by how many angles of a corner of a projected cube are the same

none: trimetric
two: dimetric
three: isometric

θ₁

θ₂

θ₃

Projection plane
Types of Axonometric Projections

Dimetric

Trimetric

Isometric
Advantages and Disadvantages

- Lines are scaled (*foreshortened*) but can find scaling factors

- Lines preserved but angles are not
  - Projection of a circle in a plane not parallel to the projection plane is an ellipse

- Does not look real because far objects are scaled the same as near objects

- Used in CAD applications
Oblique Projection

- Arbitrary relationship between projectors and projection plane
Perspective Projection

- Projectors coverage at center of projection
Vanishing Points

- Parallel lines (not parallel to the projection plan):
  - converge at a single point in the projection (the *vanishing point*)
- Drawing simple perspectives by hand uses these vanishing point(s)
Three-Point Perspective

- No principal face parallel to projection plane
- Three vanishing points for cube
**Two-Point Perspective**

- On principal direction parallel to projection plane
- Two vanishing points for cube
One-Point Perspective

- One principal face parallel to projection plane
- One vanishing point for cube
Advantages and Disadvantages

- Diminution:
  - Objects further from viewer are projected smaller (Looks realistic)

- Nonuniform foreshortening:
  - Equal distances along a line are not projected into equal distances

- Angles preserved only in planes parallel to the projection plane

- More difficult to construct by hand than parallel projections
Computer Viewing
Let's skip the clipping temporarily!
Computer Viewing

- Three aspects of the viewing process implemented in the pipeline:
  - Positioning the camera
    - Setting the model-view matrix
  - Selecting a lens
    - Setting the projection matrix
  - Clipping
    - Setting the view volume
The OpenGL Camera

- In OpenGL, initially the object and camera frames are the same
  - Default model-view matrix is an identity

- The camera is located at origin and points in the negative z direction

- OpenGL also specifies a default view volume that is a cube with sides of length 2 centered at the origin
  - Default projection matrix is an identity
Moving the Camera Frame

- If we want to visualize objects with both positive and negative z values, we can either:
  - Move the camera in the positive z direction
    - Translate the camera frame
  - Move the objects in the negative z direction
    - Translate the world frame

- Both of these views are equivalent and are determined by the model-view matrix
  - Want a translation (`glTranslatef(0.0, 0.0, -d);`) if `d > 0`
Moving Camera back from Origin

frames after translation by $-d$

$\quad d > 0$

default frames

(a)

(b)
Moving the Camera

- We can move the camera to any desired position by a sequence of rotations and translations.

- Example: side view
  - Rotate the camera
  - Move it away from origin
  - Model-view matrix $C = TR$
gluLookAt

- glLookAt(eyex, eyey, eyez, atx, aty, atz, upx, upy, upz)
**Projections and Normalization**

- The default projection in the eye (camera) frame is orthogonal.
- For points within the default view volume:
  - $x_p = x$
  - $y_p = y$
  - $z_p = 0$
**Homogeneous Coordinate Representation**

default orthographic projection

- \( x_p = x \)
- \( y_p = y \)
- \( z_p = 0 \)
- \( w_p = 1 \)

\[
p_p = Mp
\]

\[
M = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

In practice, we can let \( M = I \) and set the \( z \) term to zero later.
Simple Perspective

- Center of projection at the origin
- Projection plane \( z = d, \ d < 0 \)
**Perspective Equations**

Top view

- \( (x_p, y_p, d) \)
- \( (x, z) \)
- \( z = d \)

Side view

- \( (y_p, y, z) \)
- \( (y, z) \)

\[
\begin{align*}
x_p &= \frac{x}{z/d} \\
y_p &= \frac{y}{z/d} \\
z_p &= d
\end{align*}
\]
Homogeneous Coordinate Form

Consider \( \mathbf{q} = \mathbf{Mp} \) where

\[
\mathbf{M} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1/d & 0
\end{bmatrix}
\]

\[
\mathbf{p} = \begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]

\[\Rightarrow\]

\[
\mathbf{q} = \begin{bmatrix}
x \\
y \\
z \\
z/d
\end{bmatrix}
\]
**Perspective Division**

- However $w \neq 1$, so we must divide by $w$ to return from homogeneous coordinates.
- This *perspective division* yields

$$
x_p = \frac{x}{z/d} \quad y_p = \frac{y}{z/d} \quad z_p = d$$

the desired perspective equations.
Viewport Transformation

- From the working coordinate to the coordinate of display device.

By 2D scaling and translation

Next: clipping and normalization!