VIEWING

view volume

eye (camera)

objects outside of view

projection plane

objects in world space
OUTLINE

• Positioning a Camera
• Projections
  • Orthogonal
  • Perspective
COMPUTER VIEWING

There are three aspects of the viewing process, all of which are 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 (\texttt{Translate}(0.0,0.0,-d);)

\(d > 0\)
MOVING CAMERA BACK FROM ORIGIN

default frames

frames after translation by \(-d\)  
\(d > 0\)
A LOOKAT FUNCTION

- Given:
  - the location of the camera/eye (a point)
  - the location of the target to look at (a point)
  - a suitable “up” direction in the world space, usually y axis (a vector)
  - Create the transformation matrix to “move” the camera/world so it reflects this configuration
CAMERA COORDINATE SYSTEM

\[ \text{fwd} = \text{normalize}(\text{eye} - \text{target}) \]
\[ \text{side} = \text{normalize}(-\text{fwd} \times \text{Y}) \]
\[ \text{up} = \text{normalize}(\text{side} \times -\text{fwd}) \]
THE LOOKAT MATRIX

\[
\begin{bmatrix}
\overrightarrow{side}_x & \overrightarrow{side}_y & \overrightarrow{side}_z & -(\overrightarrow{side} \cdot \overrightarrow{eye}) \\
\overrightarrow{up}_x & \overrightarrow{up}_y & \overrightarrow{up}_z & -(\overrightarrow{up} \cdot \overrightarrow{eye}) \\
-\overrightarrow{fwd}_x & -\overrightarrow{fwd}_y & -\overrightarrow{fwd}_z & -(\overrightarrow{-fwd} \cdot \overrightarrow{eye}) \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
private Matrix 3D lookAt(Point3D eye, Point3D target, Vector 3D y) {
    Vector3d eyeV = new Vector3D(eye);
    Vector3D targetV = new Vector3D(target);
    Vector3d fwd = (targetV.minus(eyeV)).normalize();
    Vector3D side = (fwd.cross(y)).normalize();
    Vector3d up = (side.cross(fwd)).normalize();

    ...
A LOOKAT METHOD IN JAVA/JOGL (2/2)

... Matrix3D look = new Matrix3D();
look.setElementAt(0, 0, side.getX());
look.setElementAt(1, 0, up.getX());
look.setElementAt(2, 0, -fwd.getX());
look.setElementAt(3, 0, 0.0f);
look.setElementAt(0, 1, side.getY());
look.setElementAt(1, 1, up.getY());
look.setElementAt(2, 1, -fwd.getY());
look.setElementAt(3, 1, 0.0f);
look.setElementAt(0, 2, side.getZ());
look.setElementAt(1, 2, up.getZ());
look.setElementAt(2, 2, -fwd.getZ());
look.setElementAt(3, 2, 0.0f);
look.setElementAt(0, 3, side.dot(eyeV.mult(-1)));  
look.setElementAt(1, 3, up.dot(eyeV.mult(-1)));  
look.setElementAt(2, 3, (fwd.mult(-1).dot(eyeV.mult(-1)));
look.setElementAt(3, 3, 1.0f);
return(look);  
}
CLASSICAL PROJECTIONS

Front elevation

Elevation oblique

Plan oblique

Isometric

One-point perspective

Three-point perspective
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
- Nonplanar projections are needed for applications such as map construction
MAIN CLASSES OF PLANAR GEOMETRICAL PROJECTIONS

- Perspective: determined by Center of Projection (COP) (in our diagrams, the “eye”)
  - More natural, simulates what our eyes or a camera sees
- Parallel: determined by Direction of Projection (DOP) (projectors are parallel—do not converge to “eye” or COP).
  - Used in engineering and architecture for measurement purposes

In general, a projection is determined by where you place the projection plane relative to principal axes of object (relative angle and position), and what angle the projectors make with the projection plane

![Diagram](image-url)
TAXONOMY OF PLANAR GEOMETRIC PROJECTIONS

- Parallel
  - Orthographic
  - Multiview
- Axonometric
  - Isometric
  - Dimetric
  - Trimetric
- Perspective
  - 1 point
  - 2 point
  - 3 point
  - Oblique
ORTHOGONAL VIEWING

near and far measured from camera
EARLY FORMS OF PROJECTION

• Ancient Egyptian Art:
  – Multiple Viewpoints
  – Parallel Projection

– Tomb of Nefertari, Thebes (19th Dyn, ~1270 BC), Queen led by Isis. Mural

• Note how the depiction of the body implies a front view but the feet and head imply side view (early cubism)
PERSPECTIVE
EARLY ATTEMPTS AT PERSPECTIVE

- In art, an attempt to represent 3D space more realistically
- Earlier works invoke a sense of 3D space but not systematically
  - Lines converge, but no single vanishing point

Giotto
Franciscan Rule Approved
Assisi, Upper Basilica
c.1295-1300
MORE REALISTIC PERSPECTIVE
ORTHOGRAPHIC PROJECTION

Projectors are orthogonal to projection surface
MULTIVIEW ORTHOGRAPHIC (PARALLEL)

- **Used for:**
  - engineering drawings of machines, machine parts
  - working architectural drawings

- **Pros:**
  - accurate measurement possible
  - all views are at same scale

- **Cons:**
  - does not provide “realistic” view or sense of 3D form

- Usually need multiple views to get a three-dimensional feeling for object
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
TYPES OF AXONOMETRIC PROJECTIONS

Dimetric

Trimetric

Isometric
AXONOMETRIC (PARALLEL)

• Same method as multiview orthographic projections, except projection plane not parallel to any of coordinate planes; parallel lines equally foreshortened

• **Isometric**: Angles between all three principal axes equal (120°). Same scale ratio applies along each axis

• **Dimetric**: Angles between two of the principal axes equal; need two scale ratios

• **Trimetric**: Angles different between three principal axes; need three scale ratios

![Diagram of axonometric projections: dimetric, isometric, and orthographic views.](Carlomb Fig.3-8)
ISOMETRIC PROJECTION

- Used for:
  - catalogue illustrations
  - patent office records
  - furniture design
  - structural design
  - 3d Modeling in real time

- Pros:
  - don’t need multiple views
  - illustrates 3D nature of object
  - measurements can be made to scale along principal axes

- Cons:
  - lack of foreshortening creates distorted appearance
  - more useful for rectangular than curved shapes

Construction of an isometric projection: projection plane cuts each principal axis by 45°
A DESK IN PARALLEL

multiview orthographic

cavalier  cabinet

Carlbom Fig. 3-2
LACK OF FORESHORTENING
Video games have been using isometric projection for ages.

- It all started in 1982 with *Q*Bert and *Zaxxon* which were made possible by advances in raster graphics hardware.

Still in use today when you want to see things in distance as well as things close up (e.g. strategy, simulation games).

- *SimCity*
- *StarCraft*
ORTHOGONAL MATRIX

- Two steps
  - Move center to origin
    \[
    T\left(-\frac{(\text{left}+\text{right})}{2}, -\frac{(\text{bottom}+\text{top})}{2}, (\text{near}+\text{far})/2\right)
    \]
  - Scale to have sides of length 2
    \[
    S\left(\frac{2}{(\text{left}-\text{right})}, \frac{2}{(\text{top}-\text{bottom})}, \frac{2}{(\text{near}-\text{far})}\right)
    \]

\[
P = \begin{bmatrix}
    2 & 0 & 0 & \frac{\text{right} - \text{left}}{\text{right} - \text{left}} \\
    0 & 2 & 0 & \frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} \\
    0 & 0 & 2 & \frac{\text{far} + \text{near}}{\text{near} - \text{far}} \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]
ORTHOGONAL PROJECTION MATRIX

\[
\begin{bmatrix}
\frac{2}{R-L} & 0 & 0 & -\frac{R+L}{R-L} \\
0 & \frac{2}{T-B} & 0 & -\frac{T+B}{T-B} \\
0 & 0 & 1 & -\frac{Z_{\text{near}}}{Z_{\text{far}} - Z_{\text{near}}} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
PERSPECTIVE PROJECTION

Projectors converge at center of projection
PERSPECTIVE PROJECTIONS

- Used for:
  - fine art
  - Human visual system…

- Pros:
  - gives a realistic view and feeling for 3D form of object

- Cons:
  - does not preserve shape of object or scale (except where object intersects projection plane)

- Different from a parallel projection because
  - parallel lines not parallel to the projection plane converge
  - size of object is diminished with distance
  - foreshortening is not uniform

- Two understandings: **Vanishing Point and View Point**

If we were viewing this scene using parallel projection, the tracks would not converge
VANISHING POINTS

- Parallel lines (not parallel to the projection plan) on the object converge at a single point in the projection (the *vanishing point*)
- Drawing simple perspectives by hand uses these vanishing point(s)
VANISHING POINTS

- Lines extending from edges converge to common vanishing point(s)
- For right-angled forms whose face normals are perpendicular to the x, y, z coordinate axes, number of vanishing points = number of principal coordinate axes intersected by projection plane

One Point Perspective (z-axis vanishing point)

Two Point Perspective (z, and x-axis vanishing points)

Three Point Perspective (z, x, and y-axis vanishing points)
THREE-POINT PERSPECTIVE

- No principal face parallel to projection plane
- Three vanishing points for cube
TWO-POINT PERSPECTIVE

- One 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

- Objects further from viewer are projected smaller than the same sized objects closer to the viewer (*diminution*)
  - Looks realistic
- Equal distances along a line are not projected into equal distances (*nonuniform foreshortening*)
- Angles preserved only in planes parallel to the projection plane
- More difficult to construct by hand than parallel projections (but not more difficult by computer)
PERSPECTIVE PROJECTION

- Need four parameters:
  - aspect ratio: width/height of near and far clipping planes
  - field of view: vertical angle of the viewing space
  - projection/near clipping plane
  - far clipping plane
- This forms the view volume, or frustum
PERSPECTIVE PROJECTION MATRIX FORMATION

\[ q = \frac{1}{\tan\left(\frac{\text{fieldOfView}}{2}\right)} \]

\[ B = \frac{Z_{\text{near}} + Z_{\text{far}}}{Z_{\text{near}} - Z_{\text{far}}} \]

\[ A = \frac{q}{\text{aspectRatio}} \]

\[ C = \frac{2 \times (Z_{\text{near}} \times Z_{\text{far}})}{Z_{\text{near}} - Z_{\text{far}}} \]
PERSPECTIVE PROJECTION MATRIX

\[
\begin{bmatrix}
A & 0 & 0 & 0 & 0 \\
0 & q & 0 & 0 & 0 \\
0 & 0 & B & C & 0 \\
0 & 0 & 0 & -1 & 0
\end{bmatrix}
\]
NOTES

- We stay in four-dimensional homogeneous coordinates through both the modelview and projection transformations
  - Both these transformations are nonsingular
  - Default to identity matrices (orthogonal view)
- Normalization lets us clip against simple cube regardless of type of projection
- Delay final projection until end
  - Important for hidden-surface removal to retain depth information as long as possible
SUMMARY

• Positioning a Camera
• Projections
  • Orthogonal
  • Perspective