3D Polygon Rendering Pipeline

Thomas Funkhouser
Princeton University
COS 426, Fall 2000

Many applications use rendering of 3D polygons with direct illumination

Ray Casting Revisited

• For each sample ...
  ◦ Construct ray from eye position through view plane
  ◦ Find first surface intersected by ray through pixel
  ◦ Compute color of sample based on surface radiance

3D Polygon Rendering

More efficient algorithms utilize spatial coherence!

3D Polygon Rendering

• What steps are necessary to utilize spatial coherence while drawing these polygons into a 2D image?
This is a pipelined sequence of operations to draw a 3D primitive into a 2D image.

OpenGL executes steps of 3D rendering pipeline for each polygon.

Example: OpenGL

```
glBegin(GL_POLYGON);
glVertex3f(0.0, 0.0, 0.0);
glVertex3f(0.0, 0.0, 0.0);
glVertex3f(0.0, 0.0, 0.0);
glVertex3f(0.0, 0.0, 0.0);
glEnd();
```

Transform into 3D world coordinate system
Illuminate according to lighting and reflectance
3D Rendering Pipeline (for direct illumination)

Transformations

Viewing Transformations

Transformations map from one coordinate system to another

Viewing Transformation
Camera Coordinates

- Canonical coordinate system
  - Convention is right-handed (looking down -z axis)
  - Convenient for projection, clipping, etc.
  - Camera up vector maps to Y axis
  - Camera back vector maps to Z axis (pointing out of page)
  - Camera right vector maps to X axis

Viewing Transformation

- Transformation matrix maps camera basis vectors to canonical vectors in camera coordinate system

Viewing Transformations

- p(x,y,z)
  - 3D Object Coordinates
  - Modeling Transformation
  - 3D World Coordinates
  - Viewing Transformation
  - 3D Camera Coordinates
  - Projection Transformation
  - 2D Screen Coordinates
  - Perspective Transformation
  - 2D Image Coordinates
  - p'(x', y')

Projection

- General definition:
  - Transform points in n-space to m-space (m<n)
- In computer graphics:
  - Map 3D camera coordinates to 2D screen coordinates

Taxonomy of Projections
**Parallel Projection**
- Center of projection is at infinity
- Direction of projection (DOP) same for all points

**Orthographic Projections**
- DOP perpendicular to view plane

**Oblique Projections**
- DOP not perpendicular to view plane

**Parallel Projection View Volume**

**Parallel Projection Matrix**
- General parallel projection transformation:

\[
\begin{bmatrix}
    x'_w \\
    y'_w \\
    z'_w \\
    w'_w
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & L_x \cos \phi & 0 & x \\
    0 & 1 & L_y \sin \phi & 0 & y \\
    0 & 0 & 0 & 1 & z \\
    0 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

**Taxonomy of Projections**
Perspective Projection

- Map points onto “view plane” along “projectors” emanating from “center of projection” (COP)

Center of Projection
View Plane

Angel Figure 5.9

Perspective Projection

- How many vanishing points?

3-Point Perspective
2-Point Perspective
1-Point Perspective

Angel Figure 5.10

Perspective Projection View Volume

Compute 2D coordinates from 3D coordinates with similar triangles

D
(x,y,z)
(0,0,0)

View Plane

H&B Figure 12.30

Perspective Projection

- Compute 2D coordinates from 3D coordinates with similar triangles

(xD/z, yD/z)

View Plane

Perspective Projection Matrix

- 4x4 matrix representation?

\[
\begin{bmatrix}
\frac{x}{z} \\
\frac{y}{z} \\
\frac{z}{w} \end{bmatrix}
= \begin{bmatrix}
x \\
y \\
z \\
1 \end{bmatrix}
\]

\[
\begin{bmatrix}
? & ? & ? & x \\
? & ? & ? & y \\
? & ? & ? & z \\
? & ? & ? & 1 \end{bmatrix}
\]

What are the coordinates of the point resulting from projection of (x,y,z) onto the view plane?
**Perspective Projection Matrix**

- 4x4 matrix representation?

<table>
<thead>
<tr>
<th></th>
<th>x'</th>
<th>y'</th>
<th>z'</th>
<th>w'</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>y</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>z</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>w</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ x' = x \frac{D}{z} \]
\[ y' = y \frac{D}{z} \]
\[ z' = D \]
\[ w' = z / D \]

---

**Perspective vs. Parallel**

- **Perspective projection**
  - Size varies inversely with distance - looks realistic
  - Distance and angles are not (in general) preserved
  - Parallel lines do not (in general) remain parallel

- **Parallel projection**
  - Good for exact measurements
  - Parallel lines remain parallel
  - Angles are not (in general) preserved
  - Less realistic looking

---

**Taxonomy of Projections**

- Classical Projections
  - Map 3D world coordinates to 3D camera coordinates
  - Matrix has camera vectors as rows

- Perspective Transformation
  - Map 3D camera coordinates to 2D screen coordinates
  - Two types of projections: Parallel, Perspective

---

**Summary**

- Camera transformation
  - Map 3D world coordinates to 3D camera coordinates
  - Matrix has camera vectors as rows

- Projection transformation
  - Map 3D camera coordinates to 2D screen coordinates
  - Two types of projections: Parallel, Perspective
Next Time

3D Geometric Primitives

- Transform into 3D world coordinate system
- Illuminate according to lighting and reflectance
- Transform into 3D camera coordinate system
- Transform into 2D camera coordinate system
- Clip primitives outside camera’s view
- Draw pixels (includes texturing, hidden surface, etc.)