Implementation of projections

Luc RENAMBOT
3D Graphics

- Convert a set of polygons in a 3D world into an image on a 2D screen
- After theoretical view
- Implementation
Transformations

P(X,Y,Z) → Modeling Transformation → 3D Object Coordinates

→ Viewing Transformation → 3D World Coordinates

→ Projection Transformation → 3D Camera Coordinates

→ Window-to-Viewport Transformation → 2D Screen Coordinates

→ 2D Image Coordinates

P'(X',Y') → 3D Camera Coordinates

3D World Coordinates

3D Object Coordinates

2D Screen Coordinates

2D Image Coordinates
3D Rendering Pipeline

3D Geometric Primitives

- Modeling Transformation
- Lighting
- Viewing Transformation
- Projection Transformation
- Clipping
- Scan Conversion
- Image

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 (including texturing, hidden surface, etc.)
Orthographic Projection

VPN

Front Clipping Plane

View Plane

Back Clipping Plane
Perspective Projection
Viewing Reference Coordinate system
Projection Reference Point

Projection Reference Point (PRP)
Center of Window (CW)
View Reference Point (VRP)
View-Plane Normal (VPN)
Implementation

- Lots of Matrices
- Orthographic matrix
- Perspective matrix

- 3D World → Normalize to the canonical view volume → Clip against canonical view volume → Project onto projection plane → Translate into viewport
Canonical View Volumes

- Used because easy to clip against and calculate intersections
- Strategies: convert view volumes into “easy” canonical view volumes
- Transformations called Npar and Nper
Parallel Canonical Volume

- Defined by 6 planes
- $X = -1$ and $X = 1$
- $Y = -1$ and $Y = 1$
- $Z = 0$ and $Z = -1$
Perspective Canonical Volume

- Defined by 6 planes
  - $X = Z$ and $X = -Z$
  - $Y = Z$ and $Y = -Z$
  - $Z = Z_{\text{min}}$ and $Z = -1$
Normalizing Transformation

- $N_{\text{per}}$: normalizing transformation for *perspective* projection: it transforms the world-coordinate positions such that the view volume becomes the perspective projection canonical view volume.

- $N_{\text{par}}$: normalizing transformation for *parallel* projection in order to transform world-coordinate positions such that the view volume is transformed into the canonical view volume.
Implementation

- Two methods
- Main difference being whether clipping is performed in world coordinates or homogeneous coordinates
- See p.279 in white book
- The second way is more general
Method 1

- Clipping is performed in world coordinates
  1. Extend 3D coordinates to homogeneous coordinates
  2. Apply Npar or Nper to normalize the homogeneous coordinates
  3. Divide by W to go back to 3D coordinates
  4. Clip in 3D against the appropriate view volume
  5. Extend 3D coordinates to homogeneous coordinates
  6. Perform projection using either Mort or Mper (with d=1)
  7. Translate and Scale into device coordinates
  8. Divide by W to go to 2D coordinates
Method 2

- Clipping is performed in homogeneous coordinates
  1. Extend 3D coordinates to homogeneous coordinates
  2. Apply Npar or Nper' to normalize the homogeneous coordinates
  3. Clip in homogeneous coordinates
  4. Translate and Scale into device coordinates
  5. Divide by W to go to 2D coordinates
Step 1

- **Extend 3D coordinates to homogeneous coordinates**
- This is easy: we just take \((x, y, z)\) for every point and add a \(W=1\) \((x, y, z, 1)\)
- As we did previously, we are going to use homogeneous coordinates to make it easy to compose multiple matrices
Step 2

- Normalizing the homogeneous coordinates

- We normalize the homogeneous coordinates so we can clip against the canonical view volumes

- Manipulate the world so that the parts of the world that are in the existing view volume are in the new canonical view volume

- We want to create $N_{par}$ and $N_{per}$, matrices to perform this normalization
Computing Npar

1. Translate VRP to the origin
2. Rotate VRC so n-axis (VPN) is z-axis, u-axis is x-axis, and v-axis is y-axis
3. Shear so direction of projection is parallel to z-axis (only needed for oblique parallel projections - that is where the direction of projection is not normal to the view plane)
4. Translate and Scale into canonical view volume
Step 2.1

- **Translate** VRP to the origin

\[ T(-\text{VRP}) \]
Step 2.2

- **Rotate** VRC
- \( R_z = \frac{V_{PN}}{||V_{PN}||} \)
  - so \( R_z \) is a unit length vector in the direction of the VPN
- \( R_x = \frac{V_{UP} \times R_z}{||V_{UP} \times R_z||} \)
  - so \( R_x \) is a unit length vector perpendicular to \( R_z \) and \( V_{UP} \)
- \( R_y = R_z \times R_x \)
  - so \( R_y \) is a unit length vector perpendicular to the plane formed by \( R_z \) and \( R_x \)
Rotation Matrix

$$R_x = \begin{bmatrix}
  r_{1x} & r_{2x} & r_{3x} & 0 \\
  r_{1y} & r_{2y} & r_{3y} & 0 \\
  r_{1z} & r_{2z} & r_{3z} & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}$$

- Where $r_{ab}$ is $ab$th element of $R_b$
- VPN now rotated into Z axis, U into X axis and V into Y axis
- PRP now in world coordinates
Step 2.3

- **Shear** so direction of projection is parallel to z-axis (only needed for oblique parallel projections - that is where the direction of projection is not normal to the view plane) makes DOP coincident with the z axis

- Direction of projection

- DOP is now CW - PRP
Step 2.3 (cont)

- **DOP = CW - PRP**

\[
\begin{bmatrix}
DOP_x \\
DOP_y \\
DOP_z \\
0
\end{bmatrix} =
\begin{bmatrix}
(u_{\text{max}} + u_{\text{min}})/2 \\
(v_{\text{max}} + v_{\text{min}})/2 \\
0 \\
1
\end{bmatrix} -
\begin{bmatrix}
PRP_u \\
PRP_v \\
PRP_n \\
1
\end{bmatrix}
\]

- We need DOP as:

\[
\begin{bmatrix}
0 \\
0 \\
DOP'_z \\
1
\end{bmatrix}
\]

- Shear matrix

\[
\begin{bmatrix}
1 & 0 & -DOP_x/DOP_z & 0 \\
0 & 1 & -DOP_y/DOP_z & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Step 2.4

- **Translate and Scale** the sheared volume into canonical view volume

\[
T_{\text{par}} = T \begin{bmatrix}
-(u_{\text{max}} + u_{\text{min}})/2 \\
-(v_{\text{max}} + v_{\text{min}})/2 \\
-F
\end{bmatrix}
\]

\[
S_{\text{par}} = S \begin{bmatrix}
2/(u_{\text{max}} - u_{\text{min}}) \\
2/(v_{\text{max}} - v_{\text{min}}) \\
1/(F - B)
\end{bmatrix}
\]
Computing Npar

- $N_{par} = S_{par} \times T_{par} \times S_{H_{par}} \times R \times T(-VRP)$
- Scaling to fit volume
- Translation into volume
- Shear transformation
- Rotation of axis
- Translation to origin
Computing Nper

1. Translate VRP to the origin
2. Rotate VRC so n-axis (VPN) is z-axis, u-axis is x-axis, and v-axis is y-axis
3. Translate so that the center of projection (PRP) is at the origin
4. Shear so the center line of the view volume is the z-axis
5. Scale into canonical view volume
Step 2.1

- **Translate** VRP to the origin is the same as step 2.1 for Npar
- T(-VRP)
Step 2.2

- **Rotate** VRC so n-axis (VPN) is z-axis, u-axis is x-axis, and v-axis is y-axis is the same as step 2.2 for Npar
Step 2.3

- **Translate** PRP to the origin
- \( T(-\text{PRP}) \)
Step 2.4

- **Shear** so the center line of the view volume is the z-axis
- The same as step 2.3 for Npar
- The PRP is now at the origin but the CW may not be on the Z axis
- If it isn't then we need to shear to put the CW onto the Z axis
Step 2.5

- **Scale** into the canonical view volume
- Up until step 2.3, the VRP was at the origin, afterwards it may not be
- The new location of the VRP is:

\[
VRP' = SHpar \ast T(-PRP) \ast \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\]

- So

\[
Sper = \begin{bmatrix} 2VRP'_z / [(u_{max} - u_{min})(VRP'_z + B)] \\ 2VRP'_z / [(v_{max} - v_{min})(VRP'_z + B)] \\ -1/(VRP'_z + B) \end{bmatrix}
\]
Computing Nper

- So finally, we have

\[ Nper = Sper \times SHpar \times T(-PRP) \times R \times T(-VRP) \]

- Scaling to canonical view
- Shear to center the line of view of the volume
- Translate the center of projection to the origin
- Rotation of VRC
- Translate VRP to origin
Comments

• Npar and Nper matrices depends only on the camera parameters
• If the camera parameters do not change, these matrices do not need to be recomputed
• Conversely if there is constant change in the camera, these matrices will need to be constantly recreated
Comments

• Now, here is where the 2 methods diverge with method one going back to 3D coordinates to clip while method 2 remains in homogeneous coordinates.
• The choice is based on whether $W$ is ensured to be $> 0$
  • If so method 1 can be used, otherwise method 2 must be used.
• With what we have discussed in this class so far, $W$ will be $> 0$, and $W$ should remain 1 through the normalization step
• You get $W < 0$ when you do fancy stuff like b-splines
Step 3

- **Divide** by $W$ to go back to 3D coordinates
- We just take $(x, y, z, W)$ and divide all the terms by $W$ to get $(x/W, y/W, z/W, 1)$
- We ignore the 1 to go back to 3D coordinates
- We probably do not even need to divide by $W$ as it should still be 1
Step 4

- Clip in 3D against the appropriate view volume
- At this point we want to keep everything that is inside the canonical view volume, and clip away everything that is outside the canonical view volume
- Using Cohen-Sutherland algorithm we used in 2D and extend it to 3D, except now there are 6 bits instead of four
Clipping in 3D

For the parallel case the 6 bits are:

- point is above view volume: $y > 1$
- point is below view volume: $y < -1$
- point is right of view volume: $x > 1$
- point is left view volume: $x < -1$
- point is behind view volume: $Z < -1$
- point is in front of view volume: $z > 0$
Clipping in 3D

For the perspective case the 6 bits are:

- point is above view volume: \( y > -z \)
- point is below view volume: \( y < z \)
- point is right of view volume: \( x > -z \)
- point is left view volume: \( x < z \)
- point is behind view volume: \( z < -1 \)
- point is in front of view volume: \( z > z_{\text{min}} \)

Equations in book, page 273
Step 5

- Back to **homogeneous** coordinates again
- This is easy we just take \((x, y, z)\) and add a \(W=1\)

\((x, y, z, 1)\)
Step 6

- Perform Projection
- Parallel projection
- Perspective projection
Parallel Projection

- The projection plane is normal to the z-axis at z=0.
- \( X_p = X \) and \( Y_p = Y \) and \( Z \) is set to 0 to do the projection onto the projection plane.
- Points that are further away in \( Z \) still retain the same \( X \) and \( Y \) values.
- those values do not change with distance
Parallel Projection

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

- Multiplying the Mort matrix and the vector \((X, Y, Z, 1)\) holding a given point, gives the resulting vector \((X, Y, 0, 1)\)
Perspective Projection

- The projected X and Y values do depend on the Z value. Objects that are further away should appear smaller than similar objects that are closer.

\[
M_{\text{ort}} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 & 0
\end{bmatrix}
\]

- Multiplying the Mper matrix and the vector \((X, Y, Z, I)\) holding a given point, gives the resulting vector \((X, Y, Z, -Z)\).
Step 7

• **Translate and Scale** into device coordinates

• All of the points that were in the original view volume are still within the following range:
  
  • $-1 \leq X \leq 1$
  
  • $-1 \leq Y \leq 1$
  
  • $-1 \leq Z \leq 0$
Step 7

- Mapping points into the viewport by moving to device coordinates

- Steps:
  - Translate view volume so its corner (-1, -1, -1) is at the origin
  - Scale to match the size of the 3D viewport (which keeps the corner at the origin)
  - Translate the origin to the lower left hand corner of the viewport
Step 7

- Matrix: view volume to 3D viewport

\[ M_{vv3dv} = T \begin{bmatrix} X_{view_{min}} \\ Y_{view_{min}} \\ Z_{view_{min}} \end{bmatrix} \times S_{vv3dv} \times T \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \]

- With

\[ S_{vv3dv} = S \begin{bmatrix} (X_{view_{max}} - X_{view_{min}})/2 \\ (Y_{view_{max}} - Y_{view_{min}})/2 \\ (Z_{view_{max}} - Z_{view_{min}})/1 \end{bmatrix} \]

- Independent of the camera settings
- Updated only if viewport changes
Step 8

- Divide by $W$ to go from homogeneous to 2D coordinates
- Just take $(x, y, z, W)$ and divide all the terms by $W$ to get $(x/W, y/W, z/W, 1)$ and then we ignore the 1 to go back to 3D coordinates
- Parallel
  - 3D coordinates with $Z=0$ (proj plane)
- Perspective: $W=-Z$
  - So get point $(-X/Z, -Y/Z, -1)$
  - $Z=-1$ Plane
Method 2

- There is still the second method where we clip in homogeneous coordinates to be more general.
- The normalization step (step 2) is slightly different here, as both the parallel and perspective projections need to be normalized into the canonical parallel perspective view volume.
• $Npar$ above does this for the parallel case.
• $Nper'$ for the perspective case is $M \times Nper$.
• $Nper$ is the normalization given in step 2. This is the same normalization that needed to be done in step 8 of method 1 before we could convert to 2D coordinates.
Clipping (step 3)

Now in both the parallel and perspective cases the clipping routine is the same.

Again we have 6 planes to clip against:

- $X = -W$, $X = W$, $Y = -W$, $Y = W$, $Z = -W$, $Z = 0$

$$M = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{1 + Z_{\text{min}}} & \frac{z}{1 + Z_{\text{min}}} & 0 \\
0 & 0 & -1 & 0 & 0
\end{bmatrix}$$
Step 4

- Translate and Scale into device coordinates
  - translate view volume so its corner (-1, -1, -1) is at the origin
  - scale to match the size of the 3D viewport (which keeps the corner at the origin)
  - translate the origin to the lower left hand corner of the viewport
- \( \text{Mvv3dv} = T(X_{\text{viewmin}}, Y_{\text{viewmin}}, Z_{\text{viewmin}}) \times S_{\text{vv3dv}} \times T(1, 1, 1) \)
- \( S_{\text{vv3dv}} = S\left( (X_{\text{viewmax}}-X_{\text{viewmin}})/2, (Y_{\text{viewmax}}-Y_{\text{viewmin}})/2, (Z_{\text{viewmax}}-Z_{\text{viewmin}})/1 \right) \)
Step 5

- Divide by $W$ to go from homogeneous to 2D coordinates
- In the perspective projection case, dividing by $W$ will affect the transformation of the points
Next Time

• Vision and Light
Coming...

- The Midterm end of the month
- I will be supplying the paper, so all you need to bring is a few writing utensils
- The exam will be closed book, closed note, closed neighbour, and open mind. No calculators allowed.
- Please turn off all beepers and portable telephones before coming to class