COS426 Precept8

Rasterizer (Part 1)
Presented by: Linguang Zhang
Rasterizer

• Render a lot of triangles in the image plane
  • Projection – orthogonal (naïve) or perspective
  • Which triangles are in the front? (z buffering)
  • How does the triangle react to the light? (reflection model)
• Meshes are coarse. How to cheat our eyes? (interpolation)
• How does the material affect the color? (texture mapping)
• How to add fine details at low cost? (normal mapping)
GUI & Demo

COS426 Assignment 3B
Rendering: Rasterization
Switch to: Writeup
Student Name <NetID>
In this precept

- perspective projection
- barycentric coordinates
Perspective Projection

objects must be on the negative $z$ axis, otherwise cannot be seen.
Near and Far Planes

n and f are usually positive values. But near plane locates at \(-n\) and far plane locates at \(-f\).
Graphics Projection Transform

- Map x-component of a point to (-1, 1)
- Map y-component of a point to (-1, 1)
- Map z-component of a point from (near, far) to (-1, 1)
- Believe it or not, this matrix does the transformation:

\[
\begin{pmatrix}
  \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\
 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\
 0 & 0 & \frac{f+n}{f-n} & -1 \\
 0 & 0 & \frac{2fn}{f-n} & 0
\end{pmatrix}
\]
Use the Projection Matrix

• What is the fourth dimension?
  • This matrix is in homogeneous form and it should be multiplied with homogeneous coordinates: \((x, y, z, 1)^T\). Then you get \((x', y', z', w)\).
  • transform it back -> \((x'/w, y'/w, z'/w)\)
  • if \(z\) is outside (near, far), don’t do the projection because it can’t be seen.

\[
\begin{pmatrix}
\frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\
0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\
0 & 0 & \frac{f+n}{f-n} & \frac{2fn}{f-n} \\
0 & 0 & -1 & 0
\end{pmatrix}
\]
Changing Camera Pose

• This projection matrix can only be directly used when the camera coordinate is perfectly aligned with the world coordinate. What if the camera is moving?

• We represent the pose of the camera in the world space as: \([R|t]\), also in homogeneous form (4x4 matrix). \([R|t]\) transforms a point represented in the camera coordinate system to the world coordinate system.

• But we want to transform a point in the world coordinate system to the camera coordinate system. So we simply use inv([R|t]).

• Concatenate with the previous projection matrix:

\[
\begin{pmatrix}
\frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\
0 & \frac{2n}{i-b} & \frac{t+b}{t-b} & 0 \\
0 & 0 & \frac{f+n}{f-n} & -1 \\
0 & 0 & \frac{2fn}{f-n} & 0
\end{pmatrix}
\] x inv([R|t]) (given as viewMat in the code)
Barycentric Coordinates

• Any point in the triangle can be represented as a linear combination of the three vertices
  • Q is a linear combination of A2 and A3
  • P is a linear combination of Q and A1
Barycentric Coordinates

• \( P = \alpha A_1 + \beta A_2 + \gamma A_3 \)
• \( \alpha + \beta + \gamma = 1 \)
• if any of \( \alpha, \beta, \gamma < 0 \), \( P \) is not in the triangle.

See this article for detailed computation:
https://fgiesen.wordpress.com/2013/02/06/the-barycentric-conspirac/
Use Barycentric Coordinates

• Weight average of the values on the 3 coordinates
  • Interpolate z coordinate
  • Interpolate color
  • Interpolate normal direction
  • Interpolate texture coordinates
Q&A