Lecture 19: Rendering with a Z-buffer

October 30, 2018
Render this Rectangle

Z-Buffer

Red Plane

Green Plane

Blue Plane

What colors and when.
Using a Z-Buffer

• Record depth at every vertex
• For every pixel in polygon (previous lecture)
  – Interpolate to get depth at specific pixel.
  – Is depth less then currently recorded?
    • Yes: Record in Z-Buffer and paint pixel
    • No: Move along, nothing to do here
• “Paint” is shorthand for compute the surface illumination for that position on the polygon.
About depth: the z-value

• Z-buffering based upon pseudo-depth is key to modern polygon rendering.

• Depth already revealed in SageMath notebook on the Canonical View Volume.

• Here let us briefly dive into the calculation of pseudo-depth using essentially that example.
SageMath Notebook

- Emphasize the z coordinate of transform

\[ P_{cc} = \begin{bmatrix}
-\frac{u_{max}+u_{min}}{u_{max}-u_{min}} & \frac{2 \text{ near}}{u_{max}-u_{min}} & 0 & \frac{u_{max}+u_{min}}{u_{max}-u_{min}} & 0 & 0 & 0 & 0 & 0 \\
\frac{v_{max}+v_{min}}{v_{max}-v_{min}} & 0 & 0 & \frac{v_{max}+v_{min}}{v_{max}-v_{min}} & 0 & 0 & 0 & 0 & 0 \\
\frac{2 \text{ far\near}}{f_{ar\near}} & \frac{2 \text{ near\near}}{f_{ar\near}} & -\frac{2 \text{ far\near}}{f_{ar\near}} & \frac{2 \text{ near\near}}{f_{ar\near}} & 0 & 0 & 0 & 0 & 0 \\
\text{far\near} & \text{far\near} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{far\near} & \text{far\near} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{z}{u_{max}+u_{min}} & \frac{z}{v_{max}+v_{min}} & \frac{z}{2 \text{ near\near}} & \frac{z}{2 \text{ near\near}} & \frac{z}{\text{far\near}} & \frac{z}{\text{far\near}} & \frac{z}{\text{far\near}} & \frac{z}{\text{far\near}} & \frac{z}{\text{far\near}} \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix} \]

and the z term only

\[ p_z = \frac{2 \text{ far\near}}{\text{far\near}} - \frac{(\text{far\near})z}{\text{far\near}} \]

In [68]: if (case != 'sym'): 
    max(lob, loc)
Remember, the house lies between $z$ of 30 and 54 in world coordinates.

Even pushing the far clipping plane 2 orders of magnitude further back from -75 still results in the house occupying most of the pseudo-depth range between 0 and 1.
Back to the Math

- Camera at origin no world cam. rotation

\[
\begin{bmatrix}
-u_{max} + u_{min} - u_{max} - u_{min} \\
-v_{max} + v_{min} - v_{max} - v_{min} \\
2f_{near} - (f_{ar} + n_{ear})z - \frac{2f_{near}}{f_{ar} - n_{ear}}
\end{bmatrix}
\begin{bmatrix}
2n_{ear} \\
0 \\
0
\end{bmatrix}
= 
\begin{bmatrix}
\frac{2n_{ear}}{u_{max} - u_{min}} \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
0 \\
\frac{2n_{ear}}{v_{max} - v_{min}} \\
0
\end{bmatrix}
\begin{bmatrix}
-\frac{u_{max} + u_{min}}{u_{max} - u_{min}} \\
-\frac{v_{max} + v_{min}}{v_{max} - v_{min}} \\
-\frac{(f_{ar} + n_{ear})z - \frac{2f_{near}}{f_{ar} - n_{ear}}}{z}
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}
\]

\[
P_{cc} = 
\begin{bmatrix}
-u_{max} + u_{min} - u_{max} - u_{min} \\
-v_{max} + v_{min} - v_{max} - v_{min} \\
2f_{near} - (f_{ar} + n_{ear})z - \frac{2f_{near}}{f_{ar} - n_{ear}}
\end{bmatrix}
\begin{bmatrix}
2f_{near} \\
0 \\
0
\end{bmatrix}
- \frac{(f_{ar} + n_{ear})z}{z}
\]

and the z term only

\[
p_{z} = 
\begin{bmatrix}
2f_{near} \\
0 \\
0
\end{bmatrix}
- \frac{(f_{ar} + n_{ear})z}{z}
\]

Pseudo-depth
\( pz \) At near and far

- **Equation:**

\[
pz = \frac{2 \cdot \text{far} \cdot \text{near}}{(\text{far} - \text{near}) \cdot z} - \frac{(\text{far} + \text{near})}{(\text{far} - \text{near})}
\]

Let \( z \) equal \( \text{near} \)

\[
pz = \frac{2 \cdot \text{far} \cdot \text{near}}{(\text{far} - \text{near}) \cdot \text{near}} - \frac{(\text{far} + \text{near})}{(\text{far} - \text{near})}
\]

\[
pz = \frac{2 \cdot \text{far} - \text{far} - \text{near}}{(\text{far} - \text{near})}
\]

\[
pz = \frac{\text{far} - \text{near}}{(\text{far} - \text{near})}
\]

\[
pz = 1
\]

Similarly ...

Let \( z \) equal \( \text{far} \)

\[
pz = -1
\]
Plot actual Depth to Pseudo-depth

Far = -75
Plot actual Depth to Pseudo-depth

Far = -750
Plot actual Depth to Pseudo-depth

Far = -7,500
Interpolate Z-value

There are various ways to interpolate in order to arrive at an estimated z-value for an interior point on any given triangle.

Common is to first interpolate up the sides and then to interpolate across.
Z-Buffer Summary

• A Z-buffer is an array of doubles
• Size of the frame buffer / image
• Initialized to -1.0, i.e. far clipping plane
• Now consider a specific triangle
• For each pixel to be filled
  – Interpolate pixels z-value
  – Test if larger then what is in the Z-buffer
  – If yes then “paint” that pixel for that triangle
What if you Want Depth?

• Mapping may be inverted.

\[
pz = \frac{2 \times far \times near}{(far - near) \times z} - \frac{(far + near)}{(far - near)}
\]

\[
z = \frac{2 \times far \times near}{(far - near) \times pz + far + near}
\]

There are worse things then checking your work in a symbolic math package.