Lecture 15: Perspective Projection Pipeline

October 22, 2019
How do Pixels get Colored?

- Step back and consider two alternatives

```
for (each_object) render (object)
```
- Reasoning goes “What pixels are altered by this object?”
- Basis for OpenGL rendering pipeline.
- Highly Parallel on modern hardware.
- Fast, but object interactions extremely hard.

```
for (each_pixel) color (pixel)
```
- Reasoning goes “What object alters this pixels coloring?”
- Basis for Ray Casting and Ray Tracing.
- Highly Parallel
- Cost mainly intersecting 3D objects with light rays cast back into the scene.
Fill Pixels Inside Triangles

HALF-TONE PERSPECTIVE DRAWINGS BY COMPUTER

Chris Wylie, et al
Utah University

Prepared for:
Advanced Research Projects Agency
12 February 1968

Fifty years ago
“Scan Conversion”
A Characterization of Ten Hidden-Surface Algorithms

IVAN E. SUTHERLAND*, ROBERT F. SPROULL**, AND ROBERT A. SCHUMACKER

This paper discusses the hidden-surface problem from the point of view of sorting. The various surfaces of an object to be shown in hidden-surface or hidden-line form must be sorted to find out which ones are visible at various places on the screen. Surfaces may be sorted by lateral position in picture (XY), by depth (Z), or by other criteria. The paper shows that the order of sorting and the types of sorting used form differences among the existing hidden-surface algorithms. To reduce the work of sorting, each algorithm capitalizes on some coherence property of the objects represented: “Scan-line coherence,” the fact that one TV scan line of output is likely to be nearly the same as the previous TV scan line, is one commonly used kind of coherence. “Frame coherence,” the fact that the entire picture does not change very much between successive frames of a motion picture can be very helpful if it is applicable.

Forty six years ago, draw triangles so closest occludes all others.
Thirty six years ago, the foundation of Silicon Graphics and beginning of OpenGL.
Four Major Tasks

• Think of rendering objects in terms of:
  – Modeling
  – Geometry Processing
  – Rasterization
  – Fragment Processing
Moving to Formulation #3

• Review, #1 origin at PRP.
• Review, #2 origin at image center.
• Review, #1 and #2
  – No useful information on the z-axis
• We now have a new goal
• Project into a canonical view volume
• A rectangular volume with bounds:
  – U: -1 to 1, V: -1 to 1, D: 0 to 1
Remember When We Started

• What happens if you multiply a point in homogeneous coordinates by a scalar?
• Nothing!

\[
\begin{bmatrix}
  x \\
  y \\
  z \\
  w \\
\end{bmatrix}
= 
\begin{bmatrix}
  sx \\
  sy \\
  sz \\
  sw \\
\end{bmatrix}
=
 s 
\begin{bmatrix}
  x \\
  y \\
  z \\
  w \\
\end{bmatrix}
\]
Scalar Multiplication Continued

- Multiply a homogeneous matrix by a scalar?
- Again, nothing changes.

\[
\begin{bmatrix}
ax + by + cz + dw \\
ex + fy + gz + hw \\
ix + jy + kz + lw \\
mx + ny + oz + pw
\end{bmatrix}
= \begin{bmatrix}
a & b & c & d \\
e & f & g & h \\
i & j & k & l \\
m & n & o & p
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
w
\end{bmatrix}
\]

\[
\begin{bmatrix}
ax + by + cz + dw \\
ex + fy + gz + hw \\
ix + jy + kz + lw \\
mx + ny + oz + pw
\end{bmatrix}
= \begin{bmatrix}
s(ax + by + cz + dw) \\
s(ex + fy + gz + hw) \\
s(ix + jy + kz + lw) \\
s(mx + ny + oz + pw)
\end{bmatrix}
= s \cdot 
\begin{bmatrix}
a & b & c & d \\
e & f & g & h \\
i & j & k & l \\
m & n & o & p
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
w
\end{bmatrix}
\]
Form #1 & Textbook Derivation

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1/d & 0
\end{bmatrix}
= 
\begin{bmatrix}
d & 0 & 0 & 0 \\
0 & d & 0 & 0 \\
0 & 0 & d & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

7.3 Perspective Projection

The mechanism of projective transformations makes it simple to implement the division by \( z \) required to implement perspective. In the 2D example shown in Figure 7.8, we can implement the perspective projection with a matrix transformation as follows:

\[
\begin{bmatrix}
y_s \\
1
\end{bmatrix}
\sim
\begin{bmatrix}
d & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
y \\
z
\end{bmatrix}
\]

This transforms the 2D homogeneous vector \([y; z; 1]^T\) to the 1D homogeneous vector \([dy; z; 1]^T\).
Now introduce clipping planes

- Text introduces $n$, the near clipping plane.
- Also introduces $f$, the far clipping plane.
- Sets what first called $d$ to $n$.
- The handling of $z$ now carries information?

\[
\begin{bmatrix}
    n & 0 & 0 & 0 \\
    0 & n & 0 & 0 \\
    0 & 0 & n+f & -fn \\
    0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
    nx \\
    ny \\
    zn + zf - fn \\
    z
\end{bmatrix} =
\begin{bmatrix}
    n & 0 & 0 & 0 \\
    0 & n & 0 & 0 \\
    0 & 0 & n+f & -fn \\
    0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z \\
    1
\end{bmatrix}
\]
Visualize View Volume (View 1)
Visualize View Volume (View 2)
Consider Some Key Points
The 1,1 corner of the near clipping plane maps to 1,1 on image plane. The 1,1 corner on the far clipping plane comes toward the optical axis and becomes \( n/f \) on the image plane.
Visualize Frustum Change
3D Projective Viewing with House

Ross Beveridge, October 23, 2018

It is helpful to be able to review the linear algebra and the 3D geometry of viewing using a tool such as Sage. The example setup here originated as a series of Maple Worksheets I developed in CS 410 nearly 20 years ago. Here the example is updated and this example carries two key ideas. You should see the mathematics of the four major components of the Perspective Projection Pipeline geometry in both symbolic form and also in specific numerical examples. Second, the object model, a house, can be viewed using a 3D viewer both before and after being transformed in a camera's canonical view volume.

Keep in mind the canonical view volume is a cube bounded between -1 and 1 along the camera's X, Y and Z axes. The X and Y axes are the horizontal and vertical axes of the image plane. The Z axes carries pseudo-depth information. The 'pseudo' is appended to the front of 'depth' to highlight that while larger values mean further from the camera, a non-linear warping is introduced. This can be seen in the examples below.

This one Notebook includes both symbolic forms of the key components in the transformation along with different concrete examples. There are a series of different examples supported in this worksheet. These are defined and selected between below. First, some basic Sage graphics defaults are customized and then the house model is defined and viewed.

\[
M_{WC} = M_O M_P M_R M_T
\]

Understand each of these!
Canonical View
Volume

Image Plane
Looking Down

\[
M_{WC} = \begin{pmatrix}
-\frac{2}{13} \sqrt{17} & \sqrt{\frac{13}{17}} & 0 \\
\frac{8}{39} \sqrt{\frac{13}{17}} & -\frac{2}{3} \sqrt{\frac{13}{17}} & \frac{4}{39} \sqrt{17} \\
-\frac{22}{153} \sqrt{17} & -\frac{22}{153} \sqrt{17} & -\frac{11}{51} \sqrt{17} \\
\frac{2}{17} \sqrt{17} & \frac{2}{17} \sqrt{17} & \frac{3}{17} \sqrt{17}
\end{pmatrix}
\begin{pmatrix}
\frac{40}{13} \sqrt{17} & \sqrt{\frac{13}{17}} \\
\frac{120}{13} \sqrt{\frac{13}{17}} & -\frac{260}{17} \sqrt{17}
\end{pmatrix}
\]