Lecture 18: Recursive Ray Tracing

October 24, 2017
First, Old Business

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

• In lecture Thursday some questions arose about the non-linear nature of depth calculations.

• Here let us briefly dive into the calculation of pseudo-depth with an example.
First the Symptom

Near = -25
Far  = -75

Near = -25
Far  = -750

Near = -25
Far  = -7500

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

\[
\begin{bmatrix}
\frac{2\text{near}}{\text{umax} - \text{umin}} & 0 & \frac{\text{umax} + \text{umin}}{\text{umax} - \text{umin}} & 0 \\
0 & \frac{2\text{near}}{\text{vmax} - \text{vmin}} & 0 & \frac{\text{vmax} + \text{vmin}}{\text{vmax} - \text{vmin}} \\
\frac{2\text{far near}}{\text{far near}} & 0 & \frac{\text{far} + \text{near}}{\text{far near}} & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
P_{cc} = \begin{bmatrix}
\frac{\text{umax} + \text{umin}}{\text{umax} - \text{umin}} \\
\frac{\text{vmax} + \text{vmin}}{\text{vmax} - \text{vmin}} \\
\frac{2\text{far near}}{\text{far near}} + \frac{(\text{far} + \text{near})z}{\text{far near}} \\
1
\end{bmatrix}
\]

and the z term only

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

Pseudo-depth
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
How about Reflections?

- Note reflections
- Granite tabletop
- Visible on base
- Also on handle

This is a featured picture on the English language Wikipedia (Featured pictures) and is considered one of the finest images. (October 2012)
Rationale for Interreflection

• Not all the light striking a surface comes directly from a light source.
• Some reflects from one surface onto another.
• We ignore diffuse reflected light:
  – Because its small, and we can get away with it
  – Because it is very expensive to compute
• Specular reflection much more sensitive
  – Just consider reflections in previous image.
In the case of mirrors or shiny objects, this can have a major impact.
Rays of Reflection

- To add interreflections, we need the light hitting the surface from the reflected viewing ray.

- Add to ambient, diffuse and specular
Computing $V_R$

- $R_V = 2(V \cdot N)N - V$
  - Just like $R_L$, but $V$ replaces $L$
- To be more detailed…
  - $N_V = (V \cdot N)N$
  - $T_V = N_V - V = (V \cdot N)N - V$
  - $R_V = V + 2T_V = 2(V \cdot N)N - V$
Recursive Ray Tracing

• Generalize ray trace – light from a ray
  – Ray leaves surface from point of intersection
  – In the direction of $R_v$

• And how does it compute illumination?

• Exactly as we did it before, when …
  – Starting at the pixel
  – In the direction of $V$

• So ray tracing is recursive!
Now in SageMath

- Complete implementation in SageMath
- Scene with three sphere and inter-reflection
- Can modify recursion depth

Illumination with Reflection for Spheres

Ross Beveridge, October 24, 2017

In this Notebook is a full implementation of ambient, diffuse and specular reflection - in color - for a scene consisting multiple light sources and multiple spheres. Also in this Notebook is an example of recursive ray tracing. In other words, one object may be seen to reflect on the surface of another. The first bit of code that follows is book keeping to help setup libraries and display defaultst. Read on below to get more about the substance of this ray tracer.
Scene
Rendering