Lecture 13:
Ray Sphere Intersection

October 9, 2018
Sphere Intersection

- Sphere centered at \( P_c \) with radius \( r \).

\[
|P - P_c|^2 = r^2
\]

\[
|L + sU - P_c|^2 - r^2 = 0
\]

Substitute

\( T = P_c - L \)

Yielding a quadratic equation

\[
(sU - T) \cdot (sU - T) - r^2 = 0
\]
Brute Force (II)

- Expand to see what is happening.

\[
\begin{pmatrix}
  u_x & t_x \\
  u_y & t_y \\
  u_z & t_z
\end{pmatrix}
\cdot
\begin{pmatrix}
  u_x & t_x \\
  u_y & t_y \\
  u_z & t_z
\end{pmatrix}
- r^2 = 0
\]

\[
W \cdot W - r^2 = 0 \quad W = \begin{pmatrix}
  su_x - t_x \\
  su_y - t_y \\
  su_z - t_z
\end{pmatrix} \quad T = \begin{pmatrix}
  x_c - x_0 \\
  y_c - y_0 \\
  z_c - z_0
\end{pmatrix}
\]

\[
(su_x - t_x)^2 + (su_y - t_y)^2 + (su_z - t_z)^2 - r^2 = 0
\]
Sphere Intersection (III)

- Multiply then expand and collect terms.

\[(u_x^2 + u_y^2 + u_z^2) s^2 + (-2t_x u_x - 2t_y u_y - 2t_z u_z) s + (t_x^2 + t_y^2 + t_z^2) - r^2 = 0\]

- \(U\) is length 1, so \((u_x^2 + u_y^2 + u_z^2) = 1\)

- So the equation may be written as:

\[s^2 - 2(U \cdot T) s + T \cdot T - r^2 = 0\]
Reduces to Quadratic

Note vector dot products.

\[ as^2 + bs + c = 0 \]

where

\[ a = 1 \]
\[ b = -2(U \cdot T) \]
\[ c = T^2 - r^2 \]

Therefore:

\[
\begin{align*}
  s &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\
  s &= \frac{2(U \cdot T) \pm \sqrt{4(U \cdot T)^2 - 4(T^2 - r^2)}}{2} \\
  s &= (U \cdot T) \pm \sqrt{(U \cdot T)^2 - T^2 + r^2}
\end{align*}
\]
Actual Intersection Points

• Compute the two $s$ values for the two intersections:

$$s_1 = (U \cdot T) + \sqrt{(U \cdot T)^2 - T^2 + r^2}$$
$$s_2 = (U \cdot T) - \sqrt{(U \cdot T)^2 - T^2 + r^2}$$

8 multiplies/squares (caching results; $r^2$ previously stored)
9 additions/subtractions
1 square root

• Compute the actual positions along the ray for the smallest positive $s$:

$$s^* = \min(s_1, s_2)$$
$$P^* = L + s^* U$$

3 multiplies
3 additions
1 min
First Example

Sphere center: \( C = (5, 5, 5) \)

Ray start: \( L = (0, 0, 0) \)

Ray direction: \( U = \left( \frac{1}{3} \sqrt{3}, \frac{1}{3} \sqrt{3}, \frac{1}{3} \sqrt{3} \right) \)

Base to Center: \( T = (5, 5, 5) \)
Second Example

Sphere center: \( C = (5, 5, 5) \)
Ray start: \( L = (0, 0, 0) \)
Ray direction: \( U = \left( \frac{1}{3}, \frac{2}{3}, \frac{2}{3} \right) \)
But Wait

• The proceeding follows naturally from parametric approach to intersection.
• But is it smart? Is there a better way?
• Yes.
• Let’s see it,
Faster Method

\[ v = (C - L) \cdot U \]

\[ v^2 + b^2 = c^2 \]

\[ d^2 + b^2 = r^2 \]

\[ d^2 = (r^2 - (c^2 - v^2)) \]

\[ d = \sqrt{r^2 - (c^2 - v^2)} \]

If \( d^2 \) less than zero, no intersection. Otherwise, \( Q = L + (v - d)U \)
Faster Method - How Fast?

- Recall how \( ray \) is defined …
  \[
  R(s) = L + sU \\
  v = (C - L) \cdot U
  \]

- Need to compute \( v \) …
  - 3 multiplies, 2 additions.

- \( r^2 \), and possibly \( c^2 \), are already computed.
  - \( c^2 \) computed once for ray from focal point
  - Otherwise length of array \( (C - L) \)

- Test if \( r^2 \) is greater than \( (c^2 - v^2) \)
  - 1 multiply, one conditional.

- Only if intersection, \( Q = E + (v - d)U \).
  - 1 subtract, 3 multiplies, 3 additions.

- Another reference on this approach:
  - [http://www.groovyvis.com/other/raytracing/basic.html](http://www.groovyvis.com/other/raytracing/basic.html)
Sphere center:  \( C = (5, 5, 5) \)

Ray start:  \( L = (0, 0, 0) \)

Ray direction:  \( U = \left( \frac{1}{26} \sqrt{26}, \frac{3}{26} \sqrt{26}, \frac{2}{13} \sqrt{26} \right) \)

Base to Center:  \( T = (5, 5, 5) \)

\[ r^2 - b^2 = -\frac{58}{13} \]
Example 2

Sphere center: \( C = (5, 5, 5) \)

Ray start: \( L = (1, 0, 1) \)

Ray direction: \( U = \left( \frac{1}{6} \sqrt{6}, \frac{1}{3} \sqrt{6}, \frac{1}{6} \sqrt{6} \right) \)

Base to Center: \( T = (4, 5, 4) \)
Option: Rays from Focal Point

- Our rays originate from pixel in world coordinates $L$.
- With the unit vector pointing from focal point $E$ to pixel $L$.
- Alternatively, let rays originate from focal point $E$.

$$R(s) = L + sU$$

$$U = \frac{L - E}{\|L - E\|}$$

$$R(s) = E + sU$$

How might this help?
Intermediate values remain constant across all pixels when always using the focal point $E$ as the base of the ray.
Why Spheres in a Triangle World?

• How do spheres help with big polygonal models??

• Define a sphere around your model.
  – Intersecting the sphere is necessary but not sufficient for intersecting polygons in the model.
  – May help greatly to reduce work.

• What factors contribute to savings?