Ray Tracing

Outline
- Introduction
- Concept
- Ray Object Intersections
  - Sphere
  - Plane
  - Polygon
  - Box
  - Quadrics

Ray Tracing..?
- Method for producing visual images constructed in 3D computer graphics environments, with more photorealism than ray casting or scanline rendering techniques
- Works by tracing a path from an imaginary eye through each pixel in a virtual screen, accumulating the contribution of each light in the scene
- As each ray is cast from the eye, it is tested for intersection versus objects within the scene

Ray Casting Vs Ray Tracing..?
- Idea behind ray casting is to shoot rays from the eye, one per pixel, and find the closest object blocking the path of that ray
- Rays are not traced further from the first intersection

Advantages
- Realistic simulation of lighting
- Effects such as reflections and shadows are a natural result
- Simple to implement
- Impressive results

Disadvantages
- Performance
- Requires implementation of close approximation to the physical equations of light to be photorealistic
Algorithm/Concept..?

The Big Picture..!!

- A primary ray is shot through each pixel and tested for intersection against all objects in the scene.
  - S: Shadow Rays
  - T: Refracted/Transmitted Rays
  - R: Reflected Rays
- Obvious efficiency bottleneck
  - Each secondary ray is tested against all the objects in the scene.

Ray Object Intersections

- A ray is defined as
  \[ R_0 = [x_0, y_0, z_0] \] - origin of ray
  \[ R_d = [x_d, y_d, z_d] \] - direction of ray

Then, a set of points on the ray are defined as
\[ R(t) = R_0 + R_d * t \] where \( t > 0.0 \)

If \( R_d \) is normalized, then \( t \) equals the distance of the ray from origin in World Coordinates.

Ray Sphere Intersection

- Sphere is defined by its center \((S_c = [x_c, y_c, z_c])\) and radius \( S_r \)

Thus, the equation of Sphere
\[ (x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2 = S_r^2 \]

Substitute equation of ray and solve quadratic eqn for \( t \),
  - If discriminant <0 \( \Rightarrow \) No intersection
  - If discriminant \( \geq 0 \) \( \Rightarrow \) Intersection. Smaller \( t \) gives the nearest intersection point

Ray Plane Intersection

- Plane is defined by \( Ax + By + Cz + D = 0 \)
  Substitute the equation for ray and solve for \( t \),
  \[ t = -(Ax_0 + By_0 + Cz_0 + D) / (Ax_d + By_d + Cz_d) \]
  \[ = -(P_n \cdot R_0 + D) / (P_n \cdot R_d) \]
  where \( P_n \) is normal unit vector

If \( P_n \cdot R_0 = 0 \) \( \Rightarrow \) Ray is parallel to the plane
If \( P_n \cdot R_0 > 0 \) \( \Rightarrow \) Normal of plane points away from the ray
If \( t < 0 \) \( \Rightarrow \) Ray intersects plane behind origin
If \( t > 0 \) \( \Rightarrow \) Substitute \( t \) in equation for intersection point

Ray Polygon Intersection

- Check whether the ray intersects the polygon plane
- Calculate the point of intersection
- Check whether the point is within the polygon. \textbf{How?}
  - Cross products
  - Shoot a ray from the point and count edges crossed, if odd then inside
  - ....
Ray Box Intersection

- Transform box vertices \((V_i)\) to align with the coordinate axes
- Solve for intersection with \(x_{\text{min}}\) and \(x_{\text{max}}\) planes; similarly solve for \(Y\) and \(Z\).
- Check for common interval or intersection
  
  If, \(\max (t_{\text{min}}) \leq \min (t_{\text{max}})\)
  
  Then, \(t_0 = \max (t_{\text{min}})\)
  \(t_1 = \min (t_{\text{max}})\)

Ray Quadric Intersection

- Class of quadrics (surfaces that can be defined by a quadratic equation) include cylinders, cones, ellipsoids, paraboloids, etc

- The general quadric surface equation is
  \[ Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0 \]

- Substitute the equation of ray, we get the form,
  \[ A_t t^2 + B_t t + C_t = 0 \]

- Solve for \(t\)