Ray Tracing

Announcements: Final Projects

• *Everyone* should post one or more ideas for a final project on the discussion forum (it was your “assignment” over Spring Break)
• Connect with potential teammates (teams of 2 strongly recommended)
• Start reading background papers
• Proposal & summary of background research will be due in a couple weeks

Last Time?

• Rigid Body
• Finite Element Method
  – Stress/Strain
• Fracture
• Deformation
  – Level of Detail
  – Haptics

Today

• Ray Casting
  – Ray-Plane Intersection
  – Ray-Sphere Intersection
  – Point in Polygon
• Ray Tracing
• Recursive Ray Tracing
• Distribution Ray Tracing

Durer’s Ray Casting Machine

• Albrecht Durer, 16th century
Ray Casting

For every pixel
Construct a ray from the eye
For every object in the scene
Find intersection with the ray
Keep if closest
Shade depending on light and normal vector

Finding the intersection and normal is the central part of ray casting.

A Note on Local Shading

• Surface/Scene Characteristics:
  – surface normal
  – direction to light
  – viewpoint
• Material Properties
  – Diffuse (matte)
  – Specular (shiny)
  – ...
• More later!

Ray Representation?

• Two vectors:
  – Origin
  – Direction (normalized is better)
• Parametric line (explicit representation)
  – \( P(t) = \text{origin} + t \cdot \text{direction} \)

3D Plane Representation?

• Plane defined by
  – \( P_o = (x, y, z) \)
  – \( n = (A, B, C) \)
• Implicit plane equation
  – \( H(P) = Ax + By + Cz + D = 0 \)
  – \( n \cdot P + D = 0 \)
• Point-Plane distance?
  – If \( n \) is normalized, distance to plane, \( d = H(P) \)
  – \( d \) is the signed distance!

Explicit vs. Implicit?

• Ray equation is explicit \( P(t) = R_o + t \cdot R_d \)
  – Parametric
  – Generates points
  – Harder to verify that a point is on the ray
• Plane equation is implicit \( H(P) = n \cdot P + D = 0 \)
  – Solution of an equation
  – Does not generate points
  – Verifies that a point is on the plane

Ray-Plane Intersection

• Intersection means both are satisfied
• So, insert explicit equation of ray into implicit equation of plane & solve for \( t \)
  
  \[
  P(t) = R_o + t \cdot R_d \\
  H(P) = n \cdot P + D = 0 \\
  n \cdot (R_o + t \cdot R_d) + D = 0 \\
  t = (-D + n \cdot R_o) / n \cdot R_d 
  \]
Additional Housekeeping

- Verify that intersection is closer than previous
  \( P(t) < t_{\text{current}} \)
- Verify that it is not out of range (behind eye)
  \( P(t) > t_{\text{min}} \)

Ray-Triangle Intersection

- Use barycentric coordinates:
  \[ P(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c \]
  with \( \alpha + \beta + \gamma = 1 \)
  - If \( 0 < \alpha < 1 \) & \( 0 < \beta < 1 \) & \( 0 < \gamma < 1 \)
  then the point is inside the triangle!

Normal

- For shading
  - diffuse: dot product between light and normal
- Normal is constant

How Do We Compute \( \alpha, \beta, \gamma \)?

- Ratio of opposite sub-triangle area to total area
  - \( \alpha = \frac{A_a}{A} \) \hspace{0.2cm} \( \beta = \frac{A_b}{A} \) \hspace{0.2cm} \( \gamma = \frac{A_c}{A} \)
- Use signed areas for points outside the triangle

Using Cramer’s Rule…

- Used to solve for one variable at a time in system of equations
  \[
  \begin{vmatrix}
  a_x - R_{ax} & a_x - c_x & R_{dx} \\
  a_y - R_{ay} & a_y - c_y & R_{dy} \\
  a_z - R_{az} & a_z - c_z & R_{dz}
  \end{vmatrix}
  \frac{A}{|A|} = \frac{a_x - b_x}{|A|}
  \]
  \[
  \begin{vmatrix}
  a_x - b_x & a_x - R_{ax} & R_{dx} \\
  a_y - b_y & a_y - R_{ay} & R_{dy} \\
  a_z - b_z & a_z - R_{az} & R_{dz}
  \end{vmatrix}
  \frac{A}{|A|} = \frac{a_y - c_y}{|A|}
  \]
  \[
  \begin{vmatrix}
  a_x - c_x & a_x - R_{ax} & R_{dx} \\
  a_y - c_y & a_y - R_{ay} & R_{dy} \\
  a_z - c_z & a_z - R_{az} & R_{dz}
  \end{vmatrix}
  \frac{A}{|A|} = \frac{a_z - R_{az}}{|A|}
  \]

Sphere Representation?

- Implicit sphere equation
  \[ H(P) = P \cdot P - r^2 = 0 \]
Ray-Sphere Intersection

• Insert explicit equation of ray into implicit equation of sphere & solve for t

\[ P(t) = R_o + t R_d \]
\[ H(P) = P \cdot P - r^2 = 0 \]
\[ (R_o + t R_d) \cdot (R_o + t R_d) - r^2 = 0 \]
\[ R_d \cdot R_d t^2 + 2 R_d \cdot R_o t + R_o \cdot R_o - r^2 = 0 \]

Ray-Sphere Intersection

• Quadratic: \( at^2 + bt + c = 0 \)
  \(- a = 1 \) (remember, \( ||R_d|| = 1 \))
  \(- b = 2 R_d \cdot R_o \)
  \(- c = R_o \cdot R_o - r^2 \)

• with discriminant \( d = \sqrt{b^2 - 4ac} \)

• and solutions \( t_{\pm} = \frac{-b \pm d}{2a} \)

• What does it mean if there are no solutions, 1 solution, or 2 solutions?

Questions?

depth normals local shading

Today

• Ray Casting
• Ray Tracing
  – Shadows
  – Reflection
  – Refraction
• Recursive Ray Tracing
• Distribution Ray Tracing

How Can We Add Shadows?

Find the point to be shaded
For every light,
  Construct ray from point to light
For every object
  find intersection of ray with object
If no objects between point and light
  Add contribution from light

Mirror Reflection

• Cast ray symmetric with respect to the normal
• Multiply by reflection coefficient (color)
Reflection

- Reflection angle = view angle
- \( R = V - 2 (V \cdot N) N \)

Transparency

- Cast ray in refracted direction
- Multiply by transparency coefficient (color)

Qualitative Refraction

From “Color and Light in Nature” by Lynch and Livingston

Refraction

\[ \frac{\sin \theta_I}{\sin \theta_T} = \eta_I \eta_T \]

Snell-Descartes Law:

\[ \eta_I \sin \theta_I = \eta_T \sin \theta_T \]

\[ \sin \theta_T = \eta_I \sin \theta_I \]

\[ \eta_T = \frac{\eta_I}{\sin \theta_T} \]

- Total internal reflection when the square root is imaginary
- Don’t forget to normalize!

Refractive & the Sidedness of Objects

- Make sure you know whether you’re entering or leaving the transmissive material:

- What about intersecting transparent objects?

Total Internal Reflection

From “Color and Light in Nature” by Lynch and Livingston

5
Questions?

Readings for Today: *read both*


Today

- Ray Casting
- Ray Tracing
- Recursive Ray Tracing
- Distribution Ray Tracing

Ray Tracing

- Does it ever end?

The Ray Tree

Ray Debugging

- Visualize the ray tree for single image pixel
Today
• Ray Casting
• Ray Tracing
• Recursive Ray Tracing
• Distribution Ray Tracing
  – Soft shadows
  – Antialiasing (getting rid of jaggies)
  – Glossy reflection
  – Motion blur
  – Depth of field (focus)

Shadows
• one shadow ray per intersection per point light source

Shadows & Light Sources

Antialiasing – Supersampling
• multiple rays per pixel

Reflection
• one reflection ray per intersection
**Glossy Reflection**
- multiple reflection rays

**Motion Blur**
- Sample objects temporally

**Depth of Field**
- multiple rays per pixel

**Ray Tracing Algorithm Analysis**
- Ray casting
- Lots of primitives
- Recursive
- Distributed Ray Tracing Effects
  - Soft shadows
  - Anti-aliasing
  - Glossy reflection
  - Motion blur
  - Depth of field

\[
\text{cost} = \text{height} \times \text{width} \times \text{num primitives} \times \text{intersection cost} \times \text{size of recursive ray tree} \times \text{num shadow rays} \times \text{num supersamples} \times \text{num glossy rays} \times \text{num temporal samples} \times \text{num focal samples} \times \ldots
\]

*can we reduce this?*

*these can serve double duty*

**Reading for Friday 3/19:**
- Goral, Torrance, Greenberg & Battaile “Modeling the Interaction of Light Between Diffuse Surfaces”, SIGGRAPH ’84

Post a comment or question on the LMS discussion by 10am on Friday 2/29