Ray Tracing

Lots of Logistics

- Happy 2003!
- CS Has Moved
  - New Office:
    - 70-3527
    - Same phone and e-mail

Announcement

- CIA Session
  - Xerox Auditorium (Gleason Building 09)
  - January 14, 2003
  - 1pm – 1:45pm

Announcement

- Computer Animation course
  - Looks like it will be a "go".
  - Focus on Algorithms and Techniques
    - Not ***using*** Maya, but algorithms to ***build*** Maya
  - Format
    - Much like this course (assignments, project, grad readings)
  - Topics
    - Keyframing / Interpolation
    - Kinematics
    - Articulated Figure Motion
    - Procedural Animation
    - Animation and Audio (maybe)

Logistics

- Projects
  - 24 projects
  - Listing of projects now on Web
  - Presentations
    - Last 4 classes (Feb 18, 20, 25, 27)
    - 20 minutes / presentation
    - Sign up
      - E-mail me w/1st, 2nd, 3rd choice of times
      - First come, first served

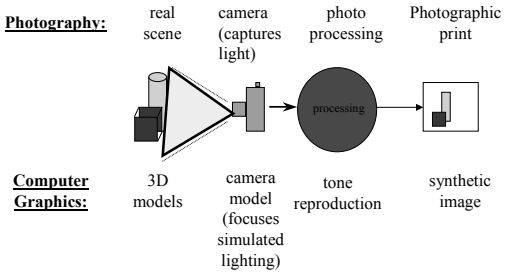
Logistics

- Paper summaries on Ray Tracing
  - Any takers?

Before we begin…

- Any questions?

Computer Graphics as Virtual Photography



Today's Class

- Ray Tracing (Recursive Ray Tracing)
  - Part I
    - Basics
    - Ray Geometry
    - Practical Considerations
  - Part II
    - So you want to write a ray tracer?
    - Assignment #1

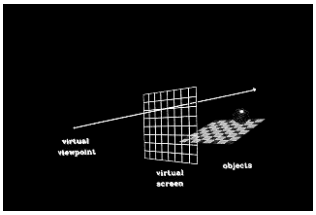
Ray Tracing -- Basics

- Note:
  - Images and applets from SIGGRAPH
  - http://www.siggraph.org/education/materials/HyperGraph/raytrace/rtrace0.htm

Ray Tracing - Basics

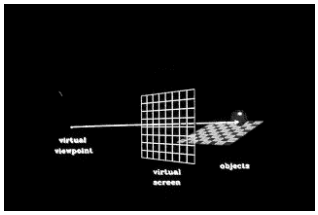
- Light rays are traced from the eye, through a viewing plane, into scene to see what it hits.
- The pixel is then set to the color values returned by the ray.
- This color is a result of the object hit by the ray.

Ray Tracing - Basics



Sometimes you don't hit an object

Ray Tracing - Basics



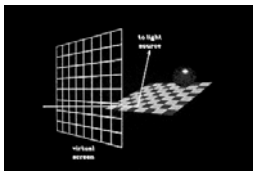
Sometimes you do

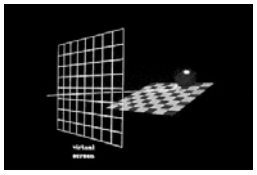
Ray Tracing - Basics

- If you do hit an object, additional rays are spawned and sent into world to determine color at intersection point
  - Shadow ray
  - Reflected ray
  - Transmitted ray

Ray Tracing - Basics

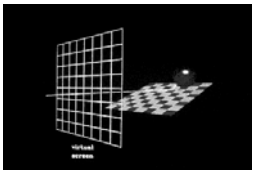
- Shadow ray
  - Ray spawned toward each light source to see if point is in shadow.

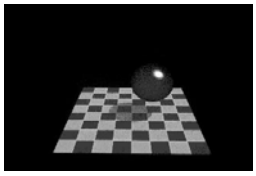




Ray Tracing - Basics

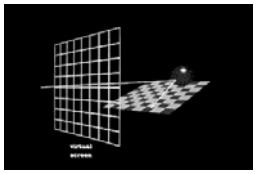
- Shadow ray

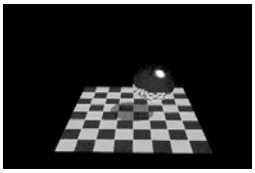




Ray Tracing

- Reflective Ray



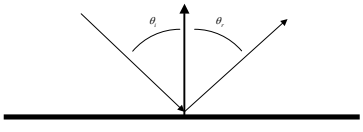


Ray Tracing Basics

Reflection

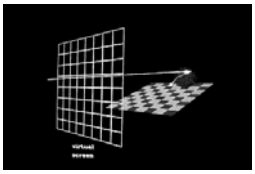
Angle of incidence = Angle of reflectance

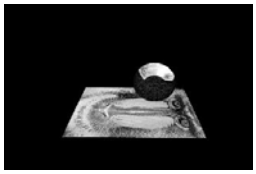
Perfect mirror surface



Ray Tracing

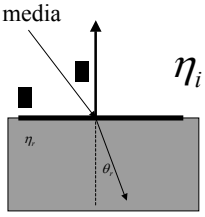
- Transmitted ray





Ray Tracing Basics

- Refraction / Transmission
  - bending of light as it travels through different media



$$\eta_i \sin\theta_i = \eta_r \sin\theta_r$$

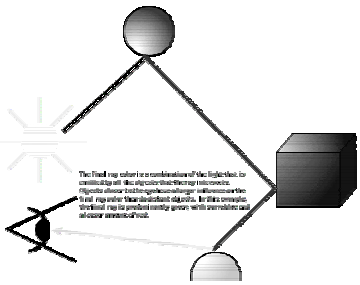
Ray Tracing - Basics

- Illumination at each intersection point

$$I = I_{\text{local}} + k_r I_{\text{reflected}} + k_t I_{\text{transmitted}}$$

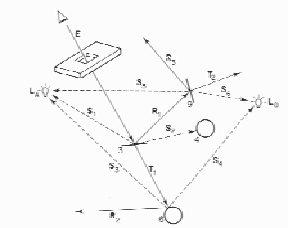
Local Illumination
Determined using your favorite illumination model
Do local only if shadow ray reaches a light source.
k_r & k_t may be wavelength dependent

Ray Tracing - Basics



Ray Tracing - Basics

- Note that formula is recursive



$$I = I_{\text{local}} + k_r I_{\text{reflected}} + k_t I_{\text{transmitted}}$$

Ray Tracing -Basics

- Ray Tracing Tree

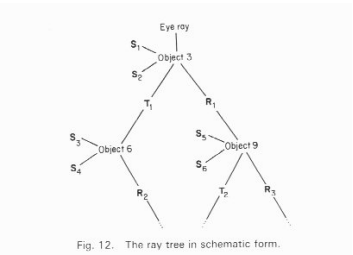


Fig. 12. The ray tree in schematic form.

Ray Tracing -- Basic

- Ray Tracing and the Rendering Equation

$$I = \underbrace{g\varepsilon}_{\text{direct}} + \underbrace{g\varepsilon(Rg)}_{\text{1st scattering}} + \underbrace{g\varepsilon(Rg)^2}_{\text{2nd scattering}} + \underbrace{g\varepsilon(Rg)^3}_{\text{3rd scattering}} + \cdots$$

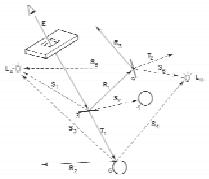
$$I = I_{\text{local}} + k_r I_{\text{reflected}} + k_t I_{\text{transmitted}}$$

Ray tracing actually gives you more since transmitted light is considered!

Ray Tracing - Basics

- Ray Tracing and the Rendering Equation

$$I = \underbrace{g\varepsilon}_{\text{direct}} + \underbrace{g\varepsilon(Rg)}_{\text{1st scattering}} + \underbrace{g\varepsilon(Rg)^2}_{\text{2nd scattering}} + \underbrace{g\varepsilon(Rg)^3}_{\text{3rd scattering}} + \cdots$$



Ray Tracing - Basics

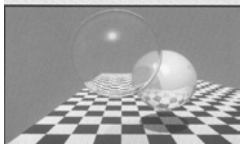
- Ray tracing applet

Ray Tracing - Basics

- Ray Tracing incorporates into a single framework:
  - Hidden surface removal
  - Shadow computation
  - Reflection of light
  - Refraction of light
  - Global Specular Interaction
- Extremely elegant and compact

Ray Tracing Basics

- Basic Ray Tracing -- Example



Whitted

Ray Tracing Basics

- Any questions on the basic idea behind ray tracing?

Ray Tracing - Ray Geometry

- Most of the computation in ray tracing is determining ray object-intersection
- When a ray intersects an object, we need to know:
  - Point of intersection
  - Normal of surface at point of intersection

Ray Tracing – Ray Geometry

- Use mathematical description of a ray and objects to determine intersection.
- Parametric representation of a ray:
  - Origin $X_o = (x_o, y_o, z_o)$
  - Direction $X_d = (x_d, y_d, z_d)$
  - Ray $(t) = X_o + t\, X_d$
- If X_d is normalized, then t will be the distance from origin.

Ray Tracing – Ray / Sphere

- The Sphere
  - A sphere can be defined by:
    - Center (x_c, y_c, z_c)
    - Radius r
  - Equation of a point (x_s, y_s, z_s) on a sphere:

$$(x_s - x_c)^2 + (y_s - y_c)^2 + (z_s - z_c)^2 = r^2$$

Ray Tracing - Ray / Sphere

- Ray - Sphere Intersection
  - Substituting ray equation for (x_s, y_s, z_s)
  - We get:

$$At^2 + Bt + C = 0$$

  - where

$$A = x_d^{\,2} + y_d^{\,2} + z_d^{\,2}$$
$$B = 2(x_d(x_o - x_c) + y_d(y_o - y_c) + z_d(z_o - z_c))$$
$$C = (x_o - x_c)^2 + (y_o - y_c)^2 + (z_o - z_c)^2 - r^2$$

Ray Tracing - Ray / Sphere

- Ray - Sphere intersection

$$t = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

Ray Tracing – Ray / Sphere

- Note that if X_d is normalized
  - A = $x_d^{\,2} + y_d^{\,2} + z_d^{\,2}$ = 1

$$t = \frac{-B \pm \sqrt{B^2 - 4C}}{2}$$

Ray Tracing – Ray / Sphere

- If $B^2 - 4C$ is:
  - < 0 – no real root, no intersection
  - $= 0$ – one root, ray intersects at sphere's surface
  - > 0 – two roots, ray goes through sphere. Use closest value.

Ray Tracing – Ray / Sphere

- Once we found a t_1 for the point of intersection, the actual point is:
  - $(x_i, y_i, z_i) = (x_0 + x_d * t_i , y_0 + y_d * t_i, z_0 + z_d * t_i)$
- The normal at the point of intersection is:
  - $(x_n, y_n, z_n) = ((x_i - x_c)/r, (y_i - y_c)/r, (z_i - z_c)/r)$
- Questions?

Ray Tracing – Ray / Plane

- The plane (boss)
  - A plane can be defined by:
    - A normal vector
  - And has the equation

$$Ax + By + Cz + D = 0$$

    - where $P_n = (A, B, C)$ gives the normal and if normalized ($A^2 + B^2 + C^2 = 1$), D will give the distance to any point on the plane from the origin.

Ray Tracing - Ray / Plane

- Ray - Plane Intersection
  - For plane with equation:

$$Ax + By + Cz + D = 0$$

  - Plug in equation for ray and we get

$$t = \frac{-(Ax_o + By_o + Cz_o + D)}{Ax_d + By_d + Cz_d} = -(P_n \bullet X_0 + D)/(P_n \bullet X_d)$$

Ray Tracing - Ray / Plane

- If $(P_n \bullet X_d)$ is
  - 0 – then ray is parallel to plane, no intersection
- If t is
  - < 0 – then the ray intersects behind the origin…ignore!
  - > 0 – calculate the point of intesection

Ray Tracing – Ray / Plane

- Once we found a t_1 for the point of intersection, the actual point is:
  - $(x_i, y_i, z_i) = (x_0 + x_d * t_i , y_0 + y_d * t_i, z_0 + z_d * t_i)$
- And we already have the normal at the point of intersection is:
  - $P_n = (A, B, C)$
- Questions?

Ray Tracing – Ray / Polygon

- Ray-Polygon Intersection
  - Find the plane in which the polygon sits
  - Find the point of intersection between the ray and the plane
  - If point of intersection is found, see if it lies within the boundaries of the polygon.

Ray Tracing - Ray / Polygon

- Find the plane in which the polygon sits
  - A plane can be defined by:
    - A normal vector
  - And has the equation

$$Ax + By + Cz + D = 0$$

    - where $P_n = (A, B, C)$ gives the normal and if normalized ($A^2 + B^2 + C^2 = 1$), D will give the distance to any point on the plane from the origin.

Ray Tracing - Ray / Polygon

- Find the point of intersection between the ray and the plane
  - Done previously

Ray Tracing - Ray / Polygon

- see if point of intersection lies within the boundaries of the polygon.
  - One algorithm:
    - Draw line from $\boldsymbol{P}_i$ to each polygon vertex
    - Measure angles between lines
    - If sum of angles between lines is 360°, polygon contains $\boldsymbol{P}_i$

Ray Tracing - Practical Considerations

- Problems with Ray Tracing
  - Object - Ray Intersection
  - Ray traced images are point sampled
    - "Too sharp"
    - Sharp shadows
    - Sharp Reflection/Refraction
    - Aliasing

- Time for a break!