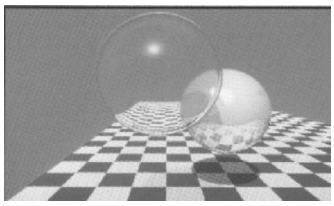
So You Want to Write a Ray Tracer

Checkpoint 3 – Basic Shading

Ray Tracing Assignment

- Goal is to reproduce the following



Whitted, 1980

Ray Tracing Assignment

- Seven checkpoints
  1. Setting the Scene
  2. Camera Modeling
  3. Basic Shading
  4. Procedural Shading
  5. Recursive Ray Tracing – Reflection
  6. Recursive Ray Tracing – Transmission
  7. Tone Reproduction

Ray Tracing Assignment

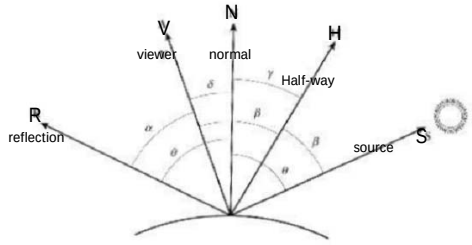
- Seven checkpoints
  1. Setting the Scene
  2. Camera Modeling
  3. Basic Shading
  4. Procedural Shading
  5. Recursive Ray Tracing – Reflection
  6. Recursive Ray Tracing – Transmission
  7. Tone Reproduction

Basic Shading

- Add Phong Illumination to your ray tracer.
  – On intersection
    - Rather than return color of object hit
    - Calculate color at intersection point using Phong Illumination model

Illumination Models

- Geometry



Illumination Models

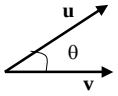
- Geometry
  - N - normal vector
  - S - direction of incoming light
  - R - direction of perfect mirror reflection
  - H - halfway between light direction and viewing direction.
  - V - viewing direction.

Phong Model

- Phong Model
  - introduces ***specular*** (mirror-like) reflections
  - Viewer direction becomes more important
  - three components
    - ambient - background light (k_a)
    - diffuse - Lambertian reflection (k_d)
    - specular – mirror-like reflection(k_s)

Illumination Models

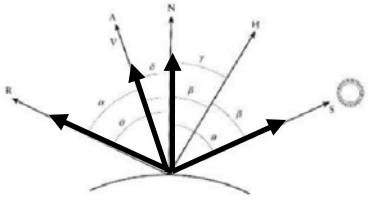
- Recall from Linear Algebra



$$u \bullet v = \|u\|\|v\|\cos\theta$$

Just one reason to normalize!

Phong Model



$$L(V) = \underbrace{k_a L_a}_{\text{ambient}} + \underbrace{k_d \sum_i L_i (S_i \bullet N)}_{\text{diffuse}} + \underbrace{k_s \sum_i L_i (R_i \bullet V)^{k_e}}_{\text{specular}}$$

Note: L_n are radiance terms, include both light and material info

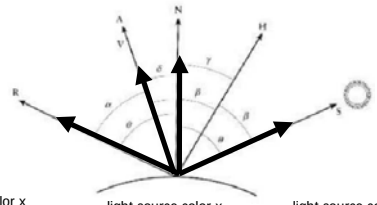
Parameters to add

- To your world:
  - Ambient light – background light (r,g,b)
- To the light source
  - Color (r,g,b) gives intensity and chroma.

Parameters to add

- To each object
  - Phong parameters
    - ambient - background light (k_a)
    - diffuse - Lambertian reflection (k_d)
    - specular – mirror-like reflection(k_s)
    - exponent – controls size of specular highlight (k_e)
  - Object "color"
    - ambient / diffuse color – basic color of object
    - Specular color – color of specular highlight (white usually)

Phong Model



Ambient light color x ambient object color

light source color x diffuse object color

light source color x specular object color

$$L(V) = k_a L_a + k_d \sum_i L_i (S_i \bullet N) + k_s \sum_i L_i (R_i \bullet V)^{k_e}$$

ambient

diffuse

specular

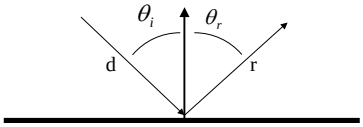
Only apply if you can see the light from point of intersection!

Vectors you will need

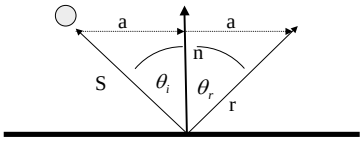
- Point of intersection
  - Get from intersection calculation
- N - normal vector
  - Get from intersection calculation
- S - direction of incoming light
  - Light position – point of intersection
  - Shadow ray: Need to know if we can see the light
- R - direction of perfect mirror reflection
  - On next slides
- V - viewing direction.
  - Camera position – point of intersection
- NORMALIZE ALL VECTORS

Reflection

Reflection
Angle of incidence of ray = Angle of reflectance
Perfect mirror surface



Calculating Reflection



$$r = S + 2a = S - 2\frac{S \bullet n}{\|n\|^2} n$$

where ***S*** is the ray from the intersection point to the light source, ***r*** is the reflected ray, and θ_r equals θ_i

For derivation, see http://astronomy.swin.edu.au/~pbourke/geometry/reflected.

Applying Phong

- If there is an intersection
  - Calculate ambient component
  - Get the point of intersection (P)
  - Spawn a shadow ray from P to the light source
  - If the ray reaches the light before any other object
    - Obtain N, V, S, and calculate R
    - Calculate specular and diffuse components
    - Add to ambient componet
  - Return resultant color.

Basic Shading

- Due date:
  - Must be posted to Web site by Midnight April 7th .
  - Recall:
    - 10% penalty per day
  - Having trouble?
    - Let me know EARLY.
- Questions?