Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

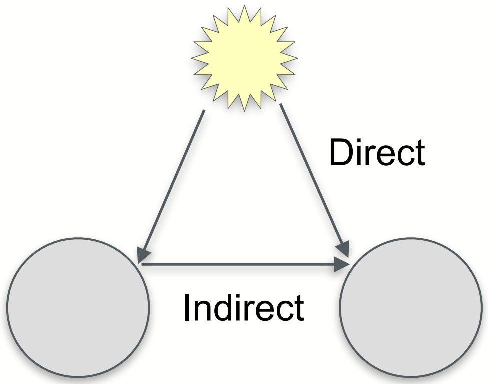
Brian Russin

Johns Hopkins
Whiting School of Engineering

Module 2D
Reflection Models

Local vs. Global Reflection Models

- Local reflection models determine the interaction between a point on the object surface and a light source
  - Only considers direct path illumination
  - Used along with interpolated shading algorithms in common graphics systems
  - Ray-casting – solves for the local reflection at the nearest intersected object
- Global reflection models also consider light reaching the point through indirect paths
  - Reflections from other surfaces
  - Transmission through semi-transparent surfaces
  - Recursive ray-tracing and radiosity methods



Phong Reflection Model – Local Reflection

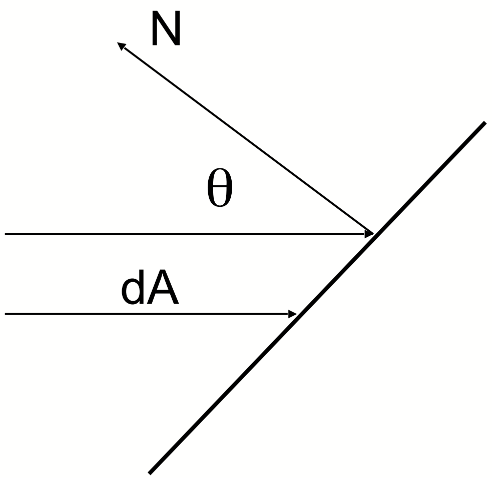
- Ray-casting and ray-tracing require a solution to the local reflection model at the intersection of a ray and the object
- **Phong reflection model**
  - Developed by Phong Bui-Tuong in 1975
  - Model is a linear combination of three components
    - Ambient, diffuse, and specular
  - Has become the standard reflection model used in computer graphics
    - Reflection model for non-perfect reflectors
      - Phong's primary contribution was modeling the specular component
  - Approximation to photo-realism
    - Graphics systems often approximate or simplify the physics of light and surface interaction
      - Mostly to simplify computation
- Other models (e.g., Cook and Torrance) are available
  - More accurate model at the cost of higher computation expense

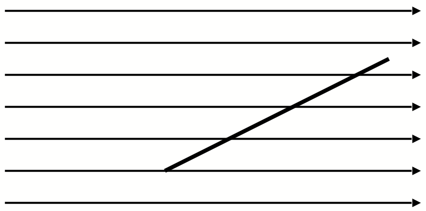
Phong Reflection Model: Ambient Component

- **Ambient** light component is used to simulate global illumination
  - Simulates a diffuse, non-directional source of light
    - Product of multiple reflections from surfaces in the scene
- Ambient component is normally modeled as a constant
  - $I_g = I_a k_a$
    - I_g is global illumination
    - I_a is the ambient illumination (constant)
    - k_a is the material's ambient reflection coefficient (0-1.0)
- Surfaces facing away from light sources become black without the ambient component
  - No light reaches the surface via a direct path
    - Surface may be visible from the view reference point
    - In reality such surfaces would be illuminated from reflections from other objects in the scene

Diffuse Reflection

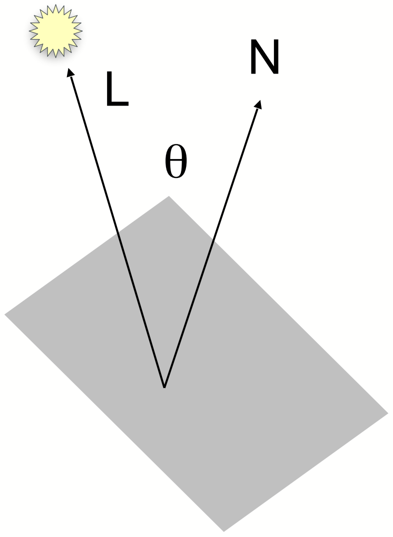
- Diffuse reflection component is constant over each planar surface in a scene regardless of viewing angle
  - Ideal diffuse component described by the BRDF
  - Diffuse reflection is primary reflection in dull surfaces
    - They reflect light uniformly in all directions
- Also called Lambertian reflection
  - Reflection governed by Lambert's cosine law
    - Radiant energy from any small surface area dA in any direction relative to the surface normal is proportional to cos(θ)
  - Reflected intensity depends only on the angle between the light source and the surface normal
- Intensity of reflection depends on the surface's orientation relative to the light source
  - Surface oriented perpendicular to the light direction appears brighter than one tilted away from that direction
  - Projected area of a surface patch perpendicular to the light direction is proportional to cos





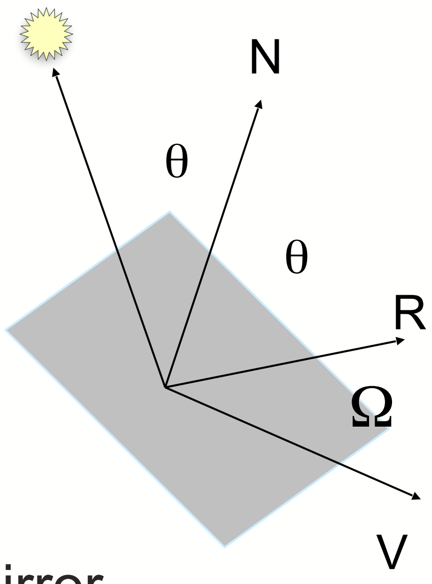
Phong Reflection Model: Diffuse Component

- Diffuse reflection intensity defined as:
  - I_i is the intensity of the light source
  - θ is the angle between the surface normal (at the point being considered) and a line from the point to the light source
  - k_d is the material's diffuse reflection coefficient (0-1)
    - Can be wavelength dependent
- Angle limited to 0 to 90 degrees
  - Surfaces are assumed to be self-occluding
    - Light cast from behind the surface does not illuminate it
- Vector operations for efficiency!
  - If N and L are unit vectors defined as illustrated:
    - $I_d = I_i k_d (L \cdot N)$
  - Diffuse component is constant over a planar surface if θ does not vary
    - Directional light source



Specular Component

- Specular component is a function of the angle between the viewing direction and the mirror direction R
  - $I_s = I_i k_s \cos^n \Omega$
  - Or: $I_s = I_i k_s (R \cdot V)^n$
  - n is an index that simulates surface roughness
  - k_s is the material's specular reflection coefficient (0-1)
- n generates a reflection 'lobe'
  - Specular reflection spread around the mirror axis
  - Large n - very thin lobe
    - Reflection decreases rapidly off R
    - Simulates a glossy surface
    - Perfect mirror - n is infinity and light is restricted to the mirror direction R
  - Small n - wider lobe
    - Simulates a rougher, less glossy surface



Specular Highlights

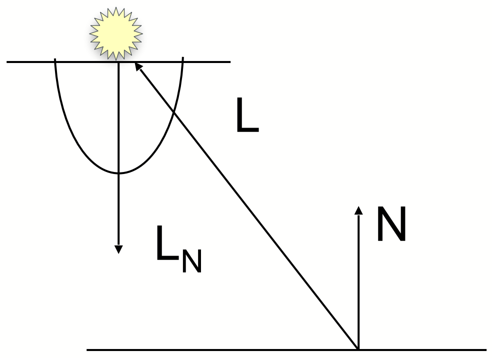
- Specular reflection component effectively produces a **highlight**
  - Highlight is a reflection of the light source spread over an area of the surface
  - Size of the highlight will depend on n
    - **Large n** produces a **smaller** highlight
    - **Small n** produces a **larger** highlight
- Color of specularly reflected light generally different than that of diffuse
  - Look at the glare off a wooden surface illuminated by white light
    - Specular reflection is mostly white in this case
  - In simple models - the color of the specular reflection is the color of the light source
    - e.g., red surface illuminated with white light
      - Diffuse reflection is red
      - Specular highlight will be white
    - Metals: more complex specular interaction

Point vs. Directional Light Sources

- Point light sources emit light equally in all directions
  - Vector L from point to the light varies across a surface
  - Can be attenuated
  - Can be a spotlight
- Directional light is a point light source at infinity
  - All L vectors constant across all surfaces
  - No attenuation
  - If assume infinite viewer (constant V) then H is constant across all surfaces
    - Reflection computation only depends on normal vector N

Spotlights

- Spotlight adds a predominant direction to a point light source
  - Uses an intensity distribution centered around a light direction
  - Similar to applying Phong specular reflection intensity to the light source
- Light source is assumed to have a predominate direction L_N
  - Source intensity is given by: $I_s = I_i(L_N \cdot L)^s$
  - Restricts the light emitted by a source to a cone
    - With a drop-off from the primary direction
    - Can also have a cutoff angle



Local Illumination Equation

- For now we will calculate the local component with a variation of the Phong model
  - Similar to OpenGL except no “light source”ambient
- OpenGL illumination equation (summed over j light sources):

$$I_{\text{local}} = E_m + I_a k_a + \sum_j f_{\text{att}j} S_j \left[I_{dj} k_d \left(L \cdot N \right) + I_{sj} k_s \left(H \cdot N \right)^n \right]$$

  - E_m is the material emission
  - I_a is the global light intensity
  - S_j is the spotlight effect from the jth light source
  - f_{att} is the attenuation factor
  - I_{aj} is the ambient intensity of the jth light source (vice constant ambient term)
  - I_{dj} is the diffuse intensity of the jth light source
  - I_{sj} is the specular intensity of the jth light source
  - k_a, k_d, k_s are the material ambient, diffuse, and specular reflection coefficients
  - L is vector from the intersection location to the light source
  - N is the normal to the surface at the intersection location
  - H is the halfway vector: H = (L + V) / | L + V |
- Solved for R, G, and B

Global Light Terms

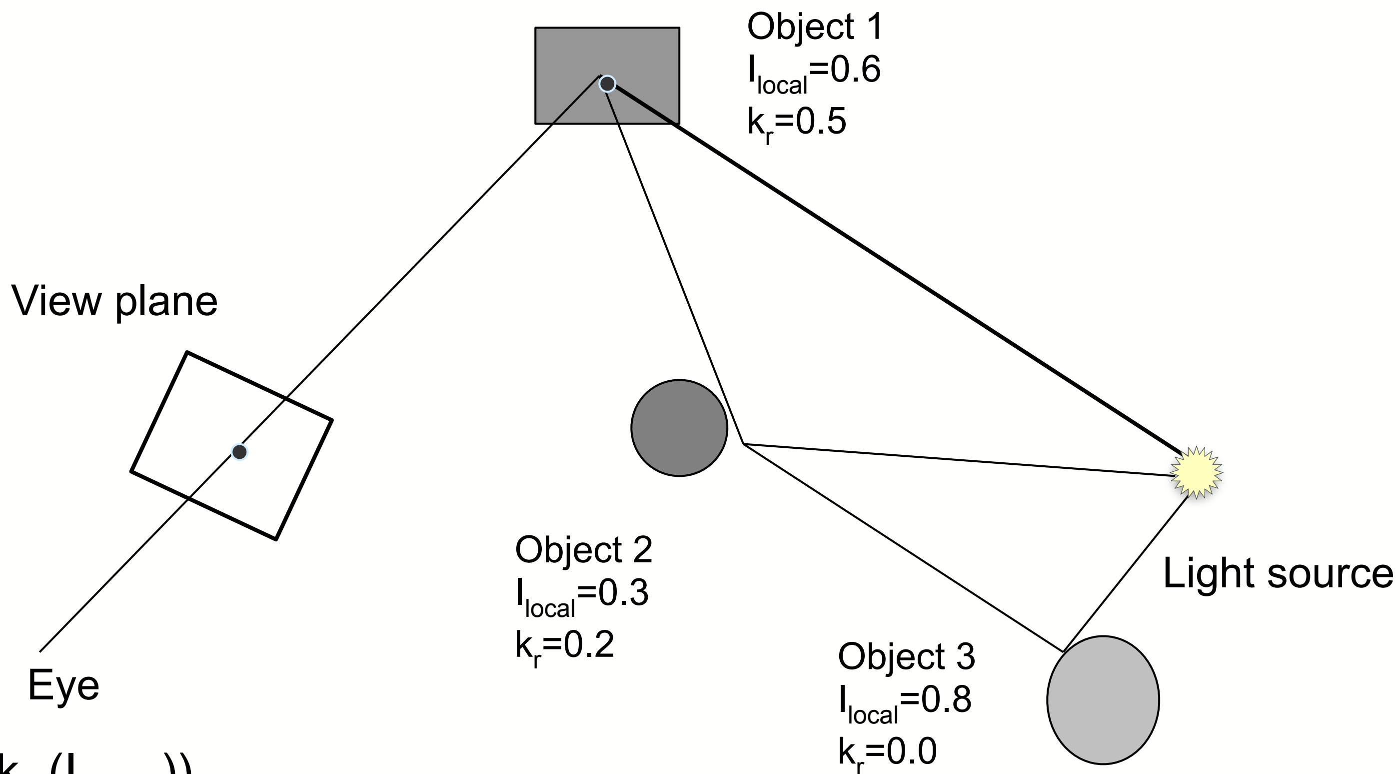
- Ray tracing adds two global terms to the local illumination model
  - Contribution from a reflected ray incident on the surface
  - Also a refracted or transmitted ray (if the object is transparent)
  - Both ray's color contributions are returned from their ray tracing procedure
- $I = I_{local} + k_{rg} I_{reflected} + k_{tg} I_{transmitted}$
  - Expresses the summation of 3 values at a point in the recursive process
    - Similar expressions evaluated at all depths of the ray tracing tree
  - Could write as: $I(P) = I_{local}(P) + k_{rg} I(P_r) + k_{tg} I(P_t)$
    - P is the intersection point under consideration
    - P_r is the first hit of the reflected ray from P
    - P_t is the first hit from the transmitted ray from P

Illumination Model (cont.)

- Final expression is:
- $$I = I_{\mathrm{local}} + I_{global} = E_m + I_a k_a + \sum_j f_{\mathrm{att}j} S_j \left[I_{dj} k_d \left(L \cdot N \right) + I_{sj} k_s \left(H \cdot N \right)^n \right] + k_{rg} I_r + k_{tg} I_t$$
  - As before there are R,G,B components
  - May include distance attenuation coefficients with k_{rg} and k_s
    - Not shown here

Attenuation

- See how illumination intensity is attenuated at each recursion



$$I=I_{local1}+k_{r1}*(I_{local2} + k_{r2}(I_{local3}))$$

$$I=0.6+0.5(0.3+0.2(0.8))=0.83$$

See how the local illumination intensity for object 3 is attenuated by 2 factors: k_{r1} and k_{r2}

Floating Point Imprecision and Self Shadowing

- Early editions of Real-Time Rendering discuss use of an error tolerance in intersection tests
  - Page 711-712 in Edition 2, less description in Edition 3 and 4
  - Due to numerical imprecision and round off
- Imprecision can cause issues in ray-tracing
- **Self shadowing** - ray originating on the surface of an object may cause an intersection with the surface itself
  - Effect is to cause blotches and irregularities in some surfaces
- Some solutions
  - Check if t is within some tolerance
    - e.g., t < EPSILON implies the ray is on the surface
  - Move the intersection point inside or outside the object by a small amount
    - “nudge” each ray origin slightly along the ray direction
  - Convex objects – do not test intersection if ray prior intersection is with that object
    - e.g., spheres, planar surfaces

Basic Ray Tracing Deficiencies

- Despite claims of realism the strength of ray tracing is its generality
  - Ability to integrate major phenomena that contribute to light/object interaction
- Produces “wrong” images
  - Inconsistent specular reflections depending on whether direct or indirect illumination is involved
  - Only possible to trace specular reflection and transmission since rays are infinitely thin
    - Excludes major light transport mechanisms such as interaction of diffuse surfaces
    - Modeling diffuse interaction would require a large number of rays to be spawned at each intersection - quickly becoming computationally impossible
    - This produces the signature ray-traced appearance
- Basic ray tracer we have discussed is inefficient due to excessive number of ray/object intersection calculations

Some Web Resources

- Since text does not have much information on how to build a ray tracer you may want to look at some web sources
  - http://www.flipcode.com/archives/Raytracing_Topics_Techniques-Part_1_Introduction.shtml
  - https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/ray-tracing-practical-example
  - You can find more of course!
- NVidia Optix
  - http://www.nvidia.com/object/optix.html
  - “Interactive” ray-tracing using CUDA computing architecture