Path Tracing

The Lighting Equation

Start with Phong since that's what you probably know, but beware:
Phong lighting is deficient. We need a physically based light equation.

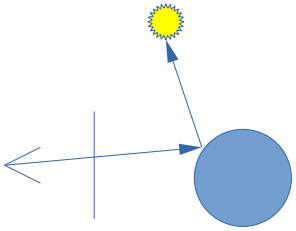
Phong: $I_o = \overbrace{I_a K_d}^{ambient} + \overbrace{I_i (N \cdot L) K_d}^{diffuse} + \overbrace{I_i K_s (R \cdot V)^{\alpha}}^{specular}$

For Ray Tracing, we make many modifications:

- eliminate the (grossly approximate) ambient term,
- separate the **light, geometry**, and **surface** quantities
  $I_o = I_i \cdot (N \cdot L) \cdot (K_d + K_s ???)$
- change notation
  - ω_i and ω_o instead of L and V
  - $\cos\theta_i$ instead of $N \cdot L$
  - $L_o(\omega_o)$ and $L_i(\omega_i)$ instead of I_o and I_i
- add an emission term $L_e(\omega_o)$, and
- eventually fix Phong's flaws with a BRDF $f(\omega_i, \omega_o)$.

Resulting in:

$$L_o(\omega_o) = L_e(\omega_o) + L_i(\omega_i) \cdot \cos\theta_i \cdot f(\omega_i, \omega_o)$$

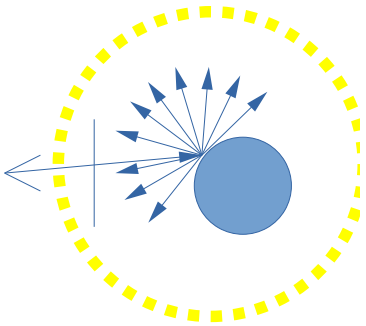


The Rendering Equation

This acknowledges that light comes in from all directions, and each input direction needs to apply the lighting equation

So, at a point x on a surface with normal N:

$$L_o(\omega_o) = L_e(\omega_o) + \int_{\Omega} L_i(\omega_i) \cos\theta_i \, f(\omega_i, \omega_o) \, d\omega_i$$

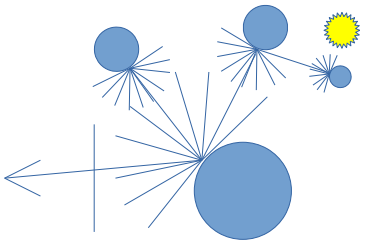


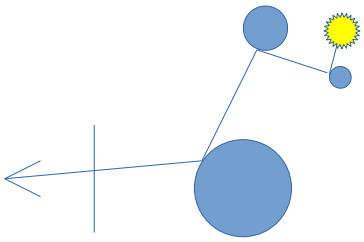
We need a way to (numerically) evaluate integrals. (Monte-Carlo)

Path Tracing Algorithm via Monte-Carlo alg:

For each pixel:
Track a running average of many instances of "light along a path"

Evaluate "light along a path":
generate ray (with a known probability p)
for many bounces:
trace ray into scene → record intersection(object, point, normal, …)
if object is a light:
return <object's emission> * <running product of BRDF's>
eval BRDF*cos() /p → into running product
generate new random ray (with probability p)





Features captured by Path Tracing

Penumbras (soft shadows): Integrate over visible portion of light
Gloss (blurred reflections): Integrate over a range of reflection directions
Translucency: Integrate over range of transmission directions.
Depth of field: Integrate over area of lens
Indirect illumination: Integrate over full diffuse range
Anti-aliasing: Integrate over a pixel's range of directions
Motion blur: Integrate over a length of time.
Sub-surface scattering and volumetric scattering
Prism/rainbow

Features not captured

- Polarization
- Interference
- Fluorescence
- Non-Linear and quantum effects
- Relativistic effects
- Gravitational lensing and atmospheric ray-bending effects.