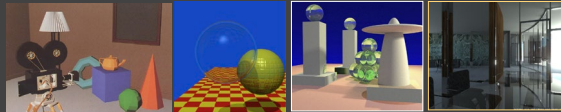


Computer Graphics II: Rendering

CSE 168 [Spr 20], Lecture 6: Direct Lighting Details
Ravi Ramamoorthi

<http://viscomp.ucsd.edu/classes/cse168/sp20>



To Do

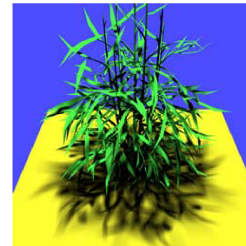
- Homework 2 (Direct Lighting) due Apr 24
- Assignment is on edX edge
- START EARLY
- This lecture goes through details of direct lighting, visibility, Monte Carlo for the assignment
- Ask re any questions

Area Lights

- Physically correct (physically no perfect point light)
- Can talk in terms of radiance at surface
- Enable soft shadows, partial visibility/occlusion
- Shoot multiple rays to light source, average
- Need proper rendering equation for accurate result
 - Only direct lighting here, so reflection equation suffices

Example: Soft Shadows

$$E(x) = \int_{H^2} L_i(x, \omega) \cos \theta d\omega$$

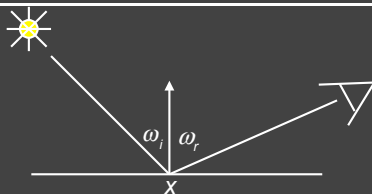


Challenges

- Visibility and blockers
- Varying light distribution
- Complex source geometry

Source: Agrawala, Ramamoorthi, Heirich, Moll, 2000

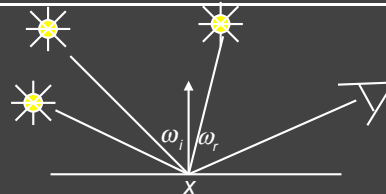
Reflection Equation



$$L_r(x, \omega_r) = L_e(x, \omega_r) + L_i(x, \omega_i) f(x, \omega_i, \omega_r) (\omega_i \cdot n)$$

Reflected Light (Output Image)	Emission	Incident Light (from light source)	BRDF	Cosine of Incident angle
-----------------------------------	----------	--	------	-----------------------------

Reflection Equation

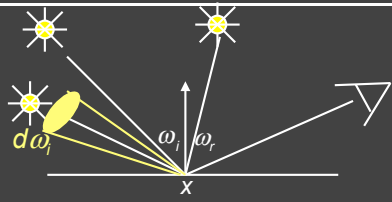


Sum over all light sources

$$L_r(x, \omega_r) = L_e(x, \omega_r) + \sum L_i(x, \omega_i) f(x, \omega_i, \omega_r) (\omega_i \cdot n)$$

Reflected Light (Output Image)	Emission	Incident Light (from light source)	BRDF	Cosine of Incident angle
-----------------------------------	----------	--	------	-----------------------------

Reflection Equation

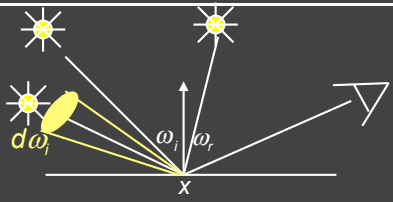


Replace sum with integral

$$L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_i(x, \omega_i) f(x, \omega_i, \omega_r) \cos \theta_i d\omega_i$$

Reflected Light (Output Image)	Emission	Incident Light (from light source)	BRDF	Cosine of Incident angle
-----------------------------------	----------	--	------	-----------------------------

Reflection Equation



As written in assignment (for one pixel)

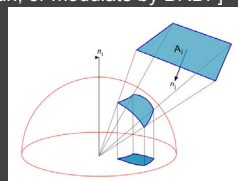
$$L_r(\omega_o) = L_e(\omega_o) + \int_{\Omega} L_i(\omega_i) f(\omega_i, \omega_o) \max(n \cdot \omega_i, 0) d\omega_i$$

Reflected Light (Outgoing Dir. Drop spat. Coord.)	Emission	Incident Light (from area light)	BRDF	Cosine of Incident angle (max not explicit hw)
--	----------	--	------	--

Uniform Polygonal Area Source

- Integrate projection of polygon P onto sphere
- Assume uniform radiance incident from light L_i
- Assume no emission (and no occlusion for now)
- Nusselt's analog (wikipedia image) project to (unit hemi)sphere, then circle (unit disk at equator) for cosine term [but only valid for Lambertian, or modulate by BRDF]

$$L_o(\omega_o) = \int_{\Omega_P} L_i f(\omega_i, \omega_o) (n \cdot \omega_i) d\omega_i$$

$$= \frac{k_d}{\pi} L_i \int_{\Omega_P} (n \cdot \omega_i) d\omega_i \quad [\text{Lambertian}]$$


Analytic Irradiance

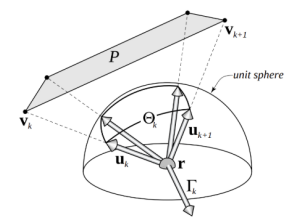
- Analytic irradiance for a polygon [Baum89, Arvo94-95]
- Useful to get precise answer, check Monte Carlo
- Define "irradiance vector" [Arvo 94]

$$\Phi(r) = \frac{1}{2} \sum_{i=1}^n \Theta_i(r) \Gamma_i(r)$$

where $\Theta_1, \Theta_2, \dots, \Theta_n$ are the angles subtended by the n edges of the polygon as seen from r

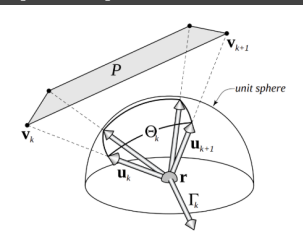
$\Theta_i = \angle \mathbf{v}_i \mathbf{r} \mathbf{v}_{i+1}$ where $\mathbf{v}_k$ are vertices of polygon, in cyclic order ($n+1 = 1$)

$\Gamma_1, \Gamma_2, \dots, \Gamma_n$ are the unit normals of the polygonal cone with cross section P and apex at r



Analytic Irradiance

- Analytic irradiance for a polygon [Baum89, Arvo94-95]
- Useful to get precise answer, check Monte Carlo
- Define "irradiance vector" [Arvo 94]

$$\Phi(r) = \frac{1}{2} \sum_{i=1}^n \Theta_i(r) \Gamma_i(r)$$


$$\Theta_i(r) = \cos^{-1} \left(\frac{(\mathbf{v}_i - \mathbf{r}) \cdot (\mathbf{v}_{i+1} - \mathbf{r})}{\|\mathbf{v}_i - \mathbf{r}\| \|\mathbf{v}_{i+1} - \mathbf{r}\|} \right)$$

$$\Gamma_i(r) = \frac{(\mathbf{v}_i - \mathbf{r}) \times (\mathbf{v}_{i+1} - \mathbf{r})}{\|\mathbf{v}_i - \mathbf{r}\| \times \|\mathbf{v}_{i+1} - \mathbf{r}\|}$$

Analytic Irradiance

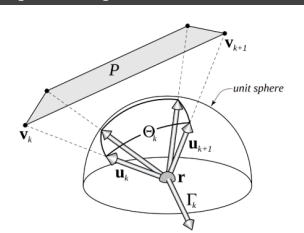
- Analytic irradiance for a polygon [Baum89, Arvo94-95]
- Useful to get precise answer, check Monte Carlo
- Define "irradiance vector" [Arvo 94]

$$\Phi(r) = \frac{1}{2} \sum_{i=1}^n \Theta_i(r) \Gamma_i(r)$$

$$E = L_i (\Phi(r) \cdot \mathbf{n}(r)) \quad [\text{Irradiance}]$$

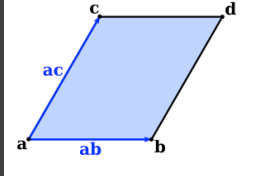
$$L_o(\omega_o) = f E = \frac{k_d}{\pi} E = \frac{k_d}{\pi} L_i (\Phi(r) \cdot \mathbf{n}(r))$$

$$\Theta_i(r) = \cos^{-1} \left(\frac{(\mathbf{v}_i - \mathbf{r}) \cdot (\mathbf{v}_{i+1} - \mathbf{r})}{\|\mathbf{v}_i - \mathbf{r}\| \|\mathbf{v}_{i+1} - \mathbf{r}\|} \right)$$

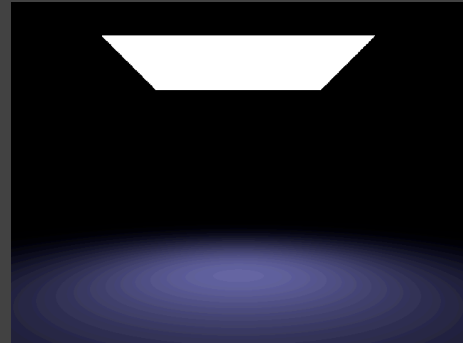
$$\Gamma_i(r) = \frac{(\mathbf{v}_i - \mathbf{r}) \times (\mathbf{v}_{i+1} - \mathbf{r})}{\|\mathbf{v}_i - \mathbf{r}\| \times \|\mathbf{v}_{i+1} - \mathbf{r}\|}$$


Implementing Analytic Irradiance

- New parallelogram light quadLight <a> <ab> <ac> <radiance>
- quadLight 0 0 0 1 0 0 0 1 1 1 1
- Primary rays like ray tracer
- Use analytic formula for shading
- Add contribs. from all lights
- No emission, visibility (yet)



Example (No Noise)



General Direct Lighting

- Include (partial) visibility by tracing multiple rays

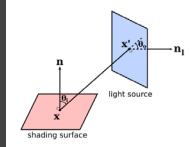
$$L_d(\omega_o) = L_i \int_{\omega_p} f(\omega_i, \omega_o) (n \cdot \omega_i) V(\omega_i) d\omega_i$$

- Include non-Lambertian reflectance (Phong)

$$f(\omega_i, \omega_o) = \frac{k_d}{\pi} + k_s \frac{(s+2)}{2\pi} \max(\mathbf{r} \cdot \omega_i, 0)^s$$

- Integrate over area light, change area measure

$$d\omega_i = \frac{\mathbf{n}_i \cdot \omega_i}{R^2} dA$$



General Direct Lighting

- Include (partial) visibility by tracing multiple rays

$$L_d(\omega_o) = L_i \int_{\omega_p} f(\omega_i, \omega_o) (n \cdot \omega_i) V(\omega_i) d\omega_i$$

- Include non-Lambertian reflectance (Phong)

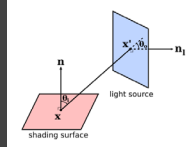
$$f(\omega_i, \omega_o) = \frac{k_d}{\pi} + k_s \frac{(s+2)}{2\pi} \max(\mathbf{r} \cdot \omega_i, 0)^s$$

- Integrate over area light, change area measure

$$d\omega_i = \frac{\mathbf{n}_i \cdot \omega_i}{R^2} dA$$

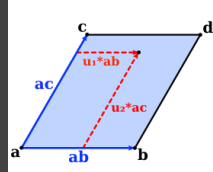
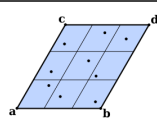
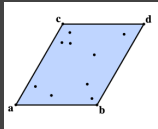
$$L_d(\mathbf{x}, \omega_o) = L_i \int_P f(\mathbf{x}, \omega_i, \omega_o) G(\mathbf{x}, \mathbf{x}') V(\mathbf{x}, \mathbf{x}') dA(\mathbf{x}')$$

$$G(\mathbf{x}, \mathbf{x}') = \frac{(\mathbf{n} \cdot \omega_i)(\mathbf{n}_i \cdot \omega_i)}{R^2} = \frac{\cos \theta_i \cos \theta_o}{|\mathbf{x} - \mathbf{x}'|^2}$$



Monte Carlo Solution

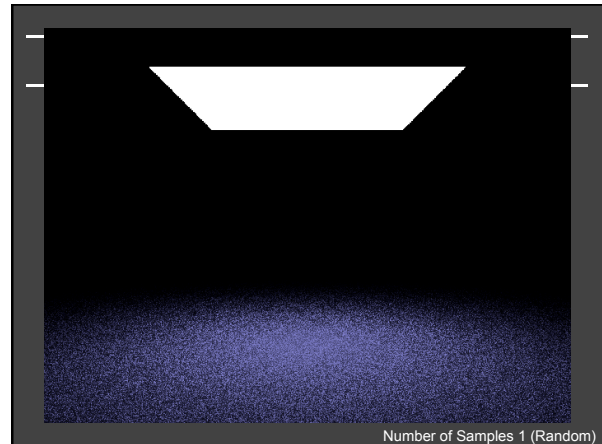
- Sample area light (two random numbers give $\mathbf{x}'_k$)
- Random or stratified (compare)

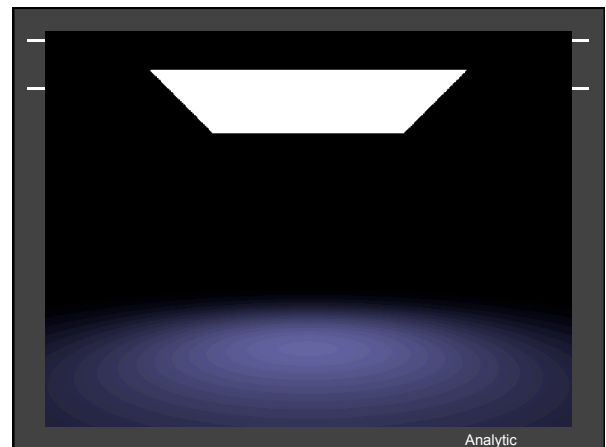
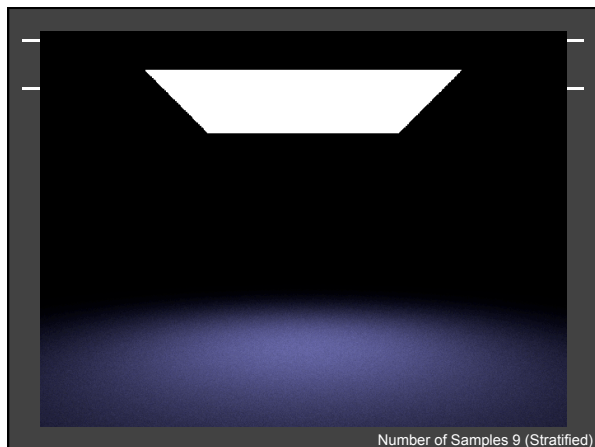
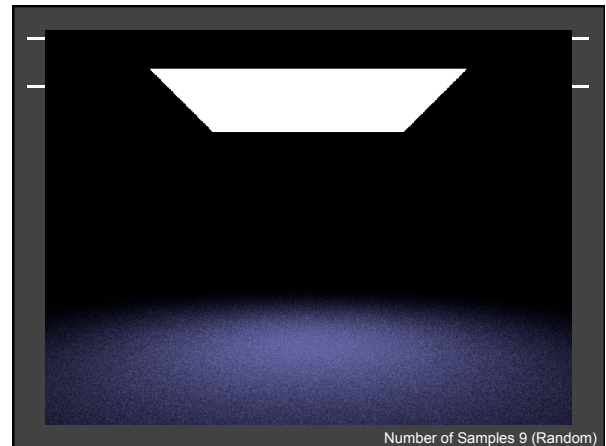
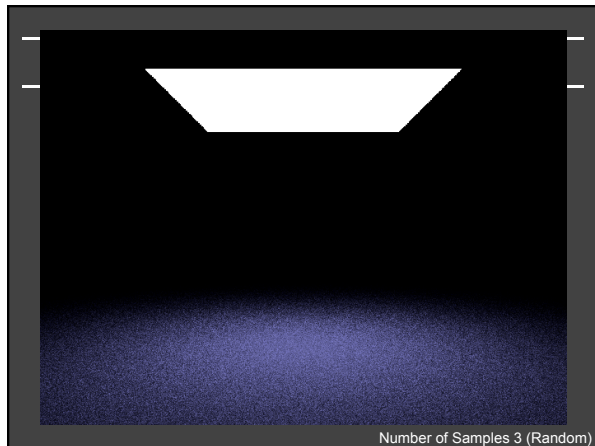


- Monte Carlo evaluation of direct lighting (N samples)

$$L_d(\mathbf{x}, \omega_o) = L_i \int_P f(\mathbf{x}, \omega_i, \omega_o) G(\mathbf{x}, \mathbf{x}') V(\mathbf{x}, \mathbf{x}') dA(\mathbf{x}')$$

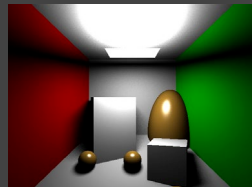
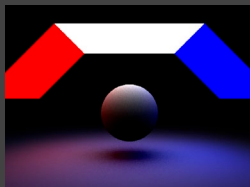
$$L_d(\mathbf{x}, \omega_o) \approx L_i \frac{A}{N} \sum_{k=1}^N f(\mathbf{x}, \omega_i(k), \omega_o) G(\mathbf{x}, \mathbf{x}'_k) V(\mathbf{x}, \mathbf{x}'_k)$$





More Complex Scenes

- For assignment, render with 5x5 stratified
- Direct lighting only, turn off recursive raytracing etc



Submission (bias, variance)

- Monte Carlo introduces noise
 - So auto-feedback can't just pixel difference
- Bias is on average how far from (analytic?) result
 - Monte Carlo rendering is an unbiased (bias 0) technique
 - High bias indicates an error in the program
 - Analytic should be exact (no bias or variance)
- Variance is squared pixel error ($RMS = stdev$)
 - All Monte Carlo results will have some variance
 - May differ in variance from our solution
 - But our bounds are reasonable (too far off = issues?)
- Focus on being correct, not just pretty
 - Can submit to autofeedback as many times as you want
 - Scores only suggestive, will grade manually

Submission (bias, variance)

- Monte Carlo introduces noise
 - So auto-feedback can't just pixel difference
- Bias is on average how far from result (close to 0)
 - Monte Carlo rendering is an unbiased technique
 - High bias indicates an error in the program
- Variance is squared pixel error (RMS = stdev)
 - May differ in variance from our solution
 - But our bounds are reasonable (too far off = issues?)

Student (hover to show example solution)	Noise-free Reference	Grade	Difference image (hover to show example diff)
		Test case PASSED bias: 0.0 ∈ [-1.0, 1.0] variance: 0.0 ∈ [-1.0, 1.0]	
		Test case PASSED bias: 0.0 ∈ [-1.0, 1.0] variance: 28.9 ∈ [25.0, 35.0]	
		Test case PASSED bias: 0.0 ∈ [-1.0, 1.0] variance: 3.8 ∈ [2.0, 5.0]	
		Test case PASSED bias: 0.0 ∈ [-1.0, 1.0] variance: 2.8 ∈ [2.0, 5.0]	

Extra: Reducing Variance

- Increase bias slightly by evaluating everything except visibility (at center [of stratum], or analytic)

$$L_d(\mathbf{x}, \omega_o) = L_i \frac{A}{N} \sum_{k=1}^N f(\mathbf{x}, \omega_i(k), \omega_o) G(\mathbf{x}, \mathbf{x}_k') V(\mathbf{x}, \mathbf{x}_k')$$

$$L_d(\mathbf{x}, \omega_o) = L_i \frac{A}{N} \sum_{k=1}^N f(\mathbf{x}, \omega_i(\mathbf{c}_k), \omega_o) G(\mathbf{x}, \mathbf{c}_k) V(\mathbf{x}, \mathbf{x}_k')$$

$\mathbf{c}_k$ is center of stratum, no noise (not Monte Carlo)

- Noise now only from visibility, none otherwise
 - No noise if no occlusions/shadows
 - Exact if use analytic formula (to evaluate unshadowed irradiance from each strata) instead of stratum center

Extra: Point Lights

- Point lights not physical (but widely used)
- How to include in Monte Carlo Rendering?
- Units aren't even same (can't use radiance)
 - Power per solid angle, not power per unit area per solid angle
- Think of as delta functions (area goes to 0, total radiance integrated over light $I = L_i A$ remains same)
 - No cosine term for light, inverse square falloff, no integral

$$L_d(\omega_o) = L_i \int_{\Omega_p} f(\omega_i, \omega_o) (n \cdot \omega_i) V(\omega_i) d\omega_i$$

$$L_d(\omega_o) = \frac{I}{R^2} f(\omega_i, \omega_o) (n \cdot \omega_i) V(\omega_i)$$

Extra: Directional Lights

- First, assume small angular extent (even sun subtends an angle of 0.5 degrees) $L_d(\omega_o) = L_i \int_{\Omega_p} f(\omega_i, \omega_o) (n \cdot \omega_i) V(\omega_i) d\omega_i$

$$L_d(\omega_o) \approx L_i \frac{\Delta\omega}{N} \sum_{k=1}^N f(\omega_i(k), \omega_o) (n \cdot \omega_i) V(\omega_i(k))$$

- If pure directional light (delta function), then fix product $E = L_i \Delta\omega$ (same units as irradiance)

$$L_d(\omega_o) = E f(\omega_i, \omega_o) (n \cdot \omega_i) V(\omega_i)$$

Combining Point, Directional, Area

- Small angle, area expressed in radiance L_i
 - Use physical units $\text{Wm}^{-2} \text{sr}^{-2}$
- Point is in intensity Wsr^{-2}
 - This is not in same units (e.g. 0...1) as area lights
 - Need to normalize by multiplying by an area unit
 - Point light same intensity as area light: multiply by area
- Directional is in irradiance units Wm^{-2}
 - This is not in same units (e.g. 0...1) as area/point lights
 - Need to normalize by multiplying by solid angle
 - Directional light same intensity as area light: multiply by average solid angle subtended by area light
 - Directional same intensity as point: divide by average squared distance to point light, mult by 2 for avg cosine