Topic 13: Radiometry

- The big picture
- Measuring light coming from a light source
- Measuring light falling onto a patch: Irradiance
- Measuring light leaving a patch: Radiance
- The Light Transport Cycle
- The Bidirectional Reflectance Distribution Function

The Basic “Light Transport” Path

1. Radiance $L(p, d_0)$ emitted from infinitesimal patch at $p$ in direction $d_0$.
2. Irradiance $H(p)$ at an infinitesimal patch due to light received along one or more (or a continuum of) directions.
3. Light source has radiant intensity $I(\omega)$ along a continuum of directions.
Light Transport Between Patches

Now that we have defined radiance, we can think of every surface point as a light source!

The General Light Transport Cycle

Now that we have defined radiance, we can think of every surface point as a light source!
One Step Along Path: Directional Integration

\[ H(\mathbf{p}) = \int \left( \text{radiance along } -\mathbf{d}_i \right) \delta(\mathbf{p} - \mathbf{d}_i) \, d^2\mathbf{d}_i \]

Now that we have defined radiance, we can think of every surface point as a light source!
Topic 13:
Radiometry

- The big picture
- Measuring light coming from a light source
- Measuring light falling onto a patch: Irradiance
- Measuring light leaving a patch: Radiance
- The Light Transport Cycle
- The Bidirectional Reflectance Distribution Function

General Light Transport Cycle: Closing the Loop

1. Radiance emitted at at q is received as incident flux at p
2. Irradiance H(p) at an infinitesimal patch due to light received along one or more (or a continuum of) directions

Q: How is incident light "transformed" to outgoing light?
Definition: The BRDF of a Point

\[ \rho_p(\vec{d}_i, \vec{d}_o) = \frac{\text{radiance in direction } \vec{d}_o}{\text{irradiance due to flux arriving from an infinitesimal solid angle around } \vec{d}_i} \]

radiance \( L(\vec{p}, \vec{d}_o) \)
emitted from infinitesimal patch at \( \vec{p} \) in direction \( \vec{d}_o \)

irradiance \( H(\vec{p}) \)
at an infinitesimal patch due to light received along one or more (or a continuum of) directions

Intuition: The BRDF tells us how bright \( \vec{p} \) will appear if viewed along \( \vec{d}_o \) when it receives light from a small cone of directions along \( \vec{d}_i \)
**Definition: The BRDF of a Point**

\[ \rho_p(d_i, d_o) = \text{radiance in direction } d_o \]
\[ \text{due to flux arriving from an infinitesimal solid angle around } d_i \]

Simpler: Suppose we only have a point light source

Intuition: The BRDF tells us how bright \( p \) will appear if viewed along \( d_o \) and the source is along \( d_i \)

**Radiance Due to a Point Light Source**

\[ \rho_p(d_i, d_o) = \text{radiance in direction } d_o \]
\[ \text{due to flux arriving from an infinitesimal solid angle around } d_i \]

**Example #1:** Source is at \( e \)

- Radiant intensity is \( I(p-e) \)
- \( \rho_p \) at \( p \) is \( \rho_p \)
- \( Q \): What is the radiance along \( d_o \)?

**Answer:**

\[ L(p, d_o) = \rho(d_i, d_o) H(p) = \rho(d_i, d_o) \frac{I(p-e)}{\|p-e\|^2} \cos \theta \]
Radiance Due to an Extended Source

\[ p(\hat{d}_i, \hat{d}_o) = \frac{\text{radiance}}{\text{irradiance}} \] around \( \hat{d}_i \)

\[ \text{irradiance around } \hat{d}_i = L(\hat{q}, -\hat{d}_i) \cdot \cos\theta \cdot d(\hat{d}_i) \]

Example #2: Extended source with radiance \( L(\hat{q}, \hat{d}_i) \)

BRDF at \( \hat{p} \) is \( p_{\hat{p}} \)

Q: What is the radiance along \( \hat{d}_o \)?

Ans: \[ L(\hat{p}, \hat{d}_o) = \int p(\hat{d}_i, \hat{d}_o) \cdot (\text{irradiance around } \hat{d}_i) \cdot d(\hat{d}_i) \]

Using spherical coords (\( \phi, \theta \)) for \( \hat{d}_i \):

Ans: \[ L(\hat{p}, \hat{d}_o) = \int \int p(\hat{d}_i, \hat{d}_o) \cdot (\text{irradiance around } \hat{d}_i) \sin\theta d\theta d\phi \]

L(\( \hat{q}, -\hat{d}_i \)) (\( \hat{v}, \hat{d}_i \))
The BRDF of a Diffuse Point

\[ \rho_P(d_i, d_o) = \text{radiance} \quad \text{in direction } d_o \]
\[ \text{due to flux arriving from an infinitesimal solid angle around} \]

Example #3: What is the BRDF of a diffuse surface point?

For diffuse points:

1. brightness independent of \( d_o \)
2. brightness depends only on total incident flux (i.e. irradiance) not illumination dir

\[ \Rightarrow \rho(d_i, d_o) = \text{constant} \]
\[ \text{what is it equal to?} \]
The BRDF of a Diffuse Point

BRDF: \( \rho_p(\hat{d}_i, \hat{d}_o) = \frac{\text{radiance in direction } \hat{d}_o}{\text{irradiance due to flux arriving from an infinitesimal solid angle around } \hat{d}_i} \)

Example #3: What is the BRDF of a diffuse surface point?

For diffuse points: conservation of energy

- total light coming in = total light going out
- (irradiance) = (radiant exitance)

can show that \( \rho = \frac{1}{\pi} \) (see Leonid's slides)

Radiance of a Diffuse Point Due to Extended Src

\( \rho_p(\hat{d}_i, \hat{d}_o) = \frac{\text{radiance around } \hat{d}_i}{\text{irradiance along } \hat{d}_o} = \frac{L(\hat{q}, -\hat{d}_i)}{L(\hat{q}, \hat{d}_o) \cdot \cos \theta} \cdot d(\hat{d}_i) \)

Example #4: Extended source with radiance \( L(\hat{q}, \hat{d}_i) \)
\( \overline{P} \) is a diffuse point

Q: What is the radiance along \( \hat{d}_o \)?

Ans: \( L(\overline{P}, \hat{d}_o) = \int_{\hat{d}_i} \rho_p(\hat{d}_i, \hat{d}_o) \cdot (\text{irradiance around } \hat{d}_i) \cdot d(\hat{d}_i) \)
Radiance of a Diffuse Point Due to Extended Source

\[ p(d_i, d_o) = \frac{\text{radiance around } d_i}{\text{irradiance around } d_i} \]

Irradiance around \( d_i \) is:
\[ L(\vec{q}, \vec{d_i}) \cdot \cos \theta \cdot d(d_i) \]

Example #4: Extended source with radiance \( L(\vec{q}, \vec{d_i}) \)
\( P \) is a diffuse point

Q: What is the radiance along \( d_o \)?

Ans:
\[ L(\vec{P}, \vec{d_o}) = \frac{1}{\pi} \int L(\vec{q}, \vec{d_i}) (\hat{n} \cdot \hat{d_i}) \, d(d_i) \]

Radiance of Diffuse Point due to Point Light Source

Example #5: Point light source at distance \( r \)
\( P \) is a diffuse point

Q: What is the radiance along \( d_o \)?

Ans:
\[ L(\vec{P}, \vec{d_o}) = \frac{1}{\pi r^2} \int L(\vec{q}, \vec{d_i}) (\hat{n} \cdot \hat{d_i}) \, d(d_i) \]
Radiance of Diffuse Point due to Point Light

Example #5: Point light source at distance \( r \)
\( \mathbf{P} \) is a diffuse point

**Q**: What is the radiance along \( \mathbf{d_0} \)?

**Ans**: 
\[
L(\mathbf{P}, \mathbf{d_0}) = \frac{1}{\pi} \cdot \frac{I(\mathbf{P}-\mathbf{e})}{\|\mathbf{P}-\mathbf{e}\|^2} \cdot (\mathbf{m} \cdot \mathbf{d_0}) = \frac{1}{\pi} I(\mathbf{P}-\mathbf{e})(\mathbf{m} \cdot \mathbf{d_0})
\]
can be ignored if light very far away

“Radiometrically-Correct” Ray Tracing

**Basic loop**:
for each pixel \( \mathbf{q} \)
1. cast ray \( r \) through \( \mathbf{q} \)
2. find 1st intersection of \( \mathbf{q} \) with scene (i.e., point \( \mathbf{P} \))
3. estimate amount of light reaching \( \mathbf{P} \)

Implemented by
- spawning a large set of rays at each step
- directional integration to compute \( I(\mathbf{P}) \)
Distribution Ray Tracing

- In **Whitted Ray Tracing** we computed lighting very crudely
  - Phong + specular global lighting

- In **Distributed Ray Tracing** we want to compute the lighting as accurately as possible
  - Use the formalism of Radiometry
  - **Compute irradiance at each pixel** (by integrating all the incoming light)
  - Since integrals are cannot be done analytically, we will employ **numeric approximations**

Benefits of Distribution Ray Tracing

- Better global diffuse lighting
  - Color bleeding
  - Bouncing highlights
- Extended light sources
- Anti-aliasing
- Motion blur
- Depth of field
- Subsurface scattering
Recall that radiance (shading) at a surface point is given by

\[ L(\vec{p}, \vec{d}_e) = \int_\Omega \rho(\vec{d}_e, \vec{d}_i) L(\vec{p}, -\vec{d}_i) (\vec{n} \cdot \vec{d}_i) d\omega \]

If we parameterize directions in spherical coordinates and assume small differential solid angle, we get

\[ L(\vec{p}, \vec{d}_e) = \int_{\phi, \theta} \rho(\vec{d}_e, (\phi, \theta)) L(\vec{p}, -\vec{d}_i(\phi, \theta)) (\vec{n} \cdot \vec{d}_i(\phi, \theta)) \sin \theta d\theta d\phi \]

Integral is over all incoming direction (hemisphere)
Irradiance at a Pixel

To compute the color of the pixel, we need to compute total light energy (flux) passing through the pixel (rectangle) (i.e. we need to compute the total irradiance at a pixel)

\[ \Phi_{i,j} = \int_{\alpha_{\text{min}} \rightarrow \alpha_{\text{max}}}^{\int_{\beta_{\text{min}} \rightarrow \beta_{\text{max}}} H(\alpha, \beta) \, d\alpha \, d\beta} \]

Integrals is over the extent of the pixel

Numerical Integration (1D Case)

- **Remember**: integral is an area under the curve
- We can approximate any integral numerically as follows

\[ \int_{x_i}^{x_{i+1}} f(x) \, dx \approx \sum_{i=1}^{N} \frac{d_i}{2} f(x_i) \]

\[ \int_{0}^{D} f(x) \, dx \rightarrow \sum_{i=1}^{N} d_i f(x_i) \]
Numerical Integration (1D Case)

- **Remember**: integral is an area under the curve
- We can approximate any integral numerically as follows

\[
\int_{0}^{D} f(x) \, dx = \sum_{i=1}^{N} \frac{D}{N} f(x_i)
\]

Problem: what if we are really unlucky and our signal has the same structure as sampling?

\[
\int_{0}^{D} f(x) \, dx = \sum_{i=1}^{N} \frac{D}{N} f(x_i)
\]
Monte Carlo Integration

- **Idea:** randomize points $x_i$ to avoid structured noise (e.g. due to periodic texture)

- Draw $N$ random samples $x_i$ independently from uniform distribution $Q(x) = U[0,D]$ (i.e. $Q(x) = 1/D$ is the uniform probability density function)

- Then approximation to the integral becomes

  \[
  \frac{1}{N} \sum_{i=1}^{N} w_i f(x_i) = \int f(x) \, dx , \quad \text{for } w_i = \frac{1}{Q(x_i)} \left( \frac{1}{D} \right)
  \]

- **We can also use other $Q$'s for efficiency !!!** (a.k.a. importance sampling)

---

Monte Carlo Integration

- **Idea:** randomize points $x_i$ to avoid structured noise (e.g. due to periodic texture)

- Draw $N$ random samples $x_i$ independently from uniform distribution $Q(x) = U[0,D]$ (i.e. $Q(x) = 1/D$ is the uniform probability density function)

- Then approximation to the integral becomes

  \[
  \frac{1}{N} \sum_{i=1}^{N} w_i f(x_i) = \int f(x) \, dx , \quad \text{for } w_i = \frac{1}{Q(x_i)} \left( \frac{1}{D} \right)
  \]

- **We can also use other $Q$'s for efficiency !!!** (a.k.a. importance sampling)
**Stratified Sampling**

- **Idea:** combination of uniform sampling plus random jitter
- Break domain into $T$ intervals of widths $d_t$ and $N_t$ samples in interval $t$

Integral approximated using the following:

$$\sum_{t=1}^{T} \frac{1}{N_t} \sum_{j=1}^{N_t} d_t f(x_{t,j})$$

- If intervals are uniform $d_t = D/T$ and there are same number of samples in each interval $N_t = N/T$ then this approximation reduces to:

$$\sum_{t=1}^{T} \sum_{j=1}^{N_t} \frac{D}{N} f(x_{t,j})$$

- **The interval size and the # of samples can vary !!!**

Integral approximated using the following:

$$\sum_{t=1}^{T} \frac{1}{N_t} \sum_{j=1}^{N_t} d_t f(x_{t,i})$$
Back to Distribution Ray Tracing

- Based on one of the approximate integration approaches we need to compute
  - Let’s try uniform sampling

\[
L(\vec{p}, \vec{d}_c) = \int_{\phi \in [0,2\pi]} \int_{\theta \in [0,\pi/2]} \rho(\vec{d}_c, \vec{d}_l(\phi, \theta)) L(\vec{p}, -\vec{d}_l(\phi, \theta)) |\vec{n} \cdot \vec{d}_l(\phi, \theta)| \sin \theta \, d\theta \, d\phi
\]

\[
= \sum_{m=1}^{M} \sum_{n=1}^{N} \rho(\vec{d}_c, \vec{d}_l(\phi_m, \theta_n)) L(\vec{p}, -\vec{d}_l(\phi_m, \theta_n)) |\vec{n} \cdot \vec{d}_l(\phi_m, \theta_n)| \sin \theta \Delta \theta \Delta \phi
\]

where

\[
\theta_m = \left( n - \frac{1}{2} \right) \Delta \theta
\]

\[
\phi_m = \left( m - \frac{1}{2} \right) \Delta \phi
\]

midpoint of the interval (sample point)  

Interval width

Importance Sampling in Distribution Ray Tracing

- Problem: Uniform sampling is too expensive (e.g. 100 samples/hemisphere with depth of ray recursion of 4 => 100^4 = 10^8 samples per pixel … with 10^5 pixels => 10^15 samples)

- Solution: Sample more densely (using importance sampling) where we know that effects will be most significant
  - Direction toward point or extended light source are significant
  - Specular and off-axis specular are significant
  - Texture/lightness gradients are significant
  - Sample less with greater depth of recursion
Importance Sampling

- **Idea:** randomize points $x_i$ to avoid structured noise (e.g. due to periodic texture)

\[
\frac{1}{N} \sum w_i f(x_i) \approx \int f(x) \, dx \quad \text{for} \quad w_i = \frac{1}{Q(x_i)}
\]

Benefits of Distribution Ray Tracing

- Better global diffuse lighting
  - Color bleeding
  - Bouncing highlights
- Extended light sources
- Anti-aliasing
- Motion blur
- Depth of field
- Subsurface scattering
Shadows in Ray Tracing

- Recall, we shoot a ray towards a light source and see if it is intercepted.

Anti-aliasing in Distribution Ray Tracer

- Lets shoot multiple rays from the same point and attenuate the color based on how many rays are intercepted.

Images from the slides by Durand and Cutler
Anti-aliasing by Deterministic Integration

- **Idea:** Use multiple rays for every pixel

- **Algorithm**
  - Subdivide pixel \((i,j)\) into squares
  - Cast ray through square centers
  - Average the obtained light

- Susceptible to structured noise, repeating textures

---

Anti-aliasing by Monte Carlo Integration

- **Idea:** Use multiple rays for every pixel

- **Algorithm**
  - Randomly sample point inside the pixel \((i,j)\)
  - Cast ray through point
  - Average the obtained light

- **Does not** suffer from structured noise, repeating textures
How many rays do you need?

1 ray/light  
10 ray/light  
20 ray/light  
50 ray/light  


Soft Shadows with Distribution Ray Tracing

- Let's shoot multiple rays from the same point and attenuate the color based on how many rays are intercepted.

\[ \hat{c}_k = -\hat{d}_{i,j} \]

Images from the slides by Durand and Cutler
Antialiasing – Supersampling

Point Light

Images from the slides by Durand and Cutler

Specular Reflections

Recall, we had to shoot a ray in a perfect specular reflection direction (with respect to the camera) and get the radiance at the resulting hit point.
**Specular Reflections with DRT**

- Same, but shoot multiple rays

\[ \vec{c}_k = -\hat{d}_{i,j} \]

\[ \vec{r}_k = 2(\hat{s}_k \cdot \vec{n}_k)\vec{n}_k - \vec{s}_k \]

- Spread is dictated by BRDF

**Depth of Field**

- So far with our Ray Tracers we only considered **pinhole camera model** (no lens)
  - or alternatively, lens, but tiny aperture

**Image Plane**

- Lens

- optical axis
Depth of Field

- So far with our Ray Tracers we only considered pinhole camera model (no lens)
  - or alternatively, lens, but tiny aperture
- What happens if we put a lens into our “camera”
  - or increase the aperture
- Remember the thin lens equation?

\[
\frac{1}{f} = \frac{1}{z_0} + \frac{1}{z_1}
\]
Changing the focal-length in DRT

**increasing focal length**

---

changing the aperture in DRT

**decreasing aperture**

---

220x400 pixels
144 samples per pixel
~4.5 minutes to render
Depth of Field
Depth of Field
Depth of Field

We ignored the fact that it takes time to form the image
- We ignored this for radiometry

During that time the shutter is open and light is collected
- We need to integrate *temporally*, not only spatially

\[ \int \int \int H(\alpha, \beta, t) \, d\alpha \, d\beta \, dt \]
Motion Blur

Cook, Porter & Carpenter

Motion Blur

Long Exposure Photography
Motion Blur (long exposures)

Golden Gate Bridge
30 sec. exposure @ f4

Bodie State Park
30 min. exposure @ f4

Motion Blur (short exposures)

Doc Edgerton, 1936
Sub-surface Scattering

Bidirectional Surface Scattering Reflectance Distribution Function
Bidirectional Surface Scattering
Reflectance Distribution Function

Semi-Transparencies

Image form http://www.graphics.cornell.edu/online/tutorial/raytrace/
Texture-mapping and Bump-mapping in Ray Tracer

Image form http://www.graphics.cornell.edu/online/tutorial/raytrace/

Caustics

- Hard to do in Distribution Ray Tracing
  - Why?
Caustics

- Hard to do in Distribution Ray Tracing
  - Why?

  Hard to come up with a good importance function for sampling,
  Hence, VERY VERY slow

Caustics

- Often done using bi-directional ray tracing (a.k.a. photon mapping)
  - Shoot light rays from light sources
  - Accumulate the amount of light (radiance) at each surface
  - Shoot rays through image plane pixels to “look-up” the radiance (and integrate irradiance over the area of the pixel)
Photon Mapping

- Simulates **individual photons**
  - In DRT we were simulating radiance (flux)
- Photons are emitted from light sources
- Photons bounce off of specular surfaces
- Photons are **deposited on diffuse surfaces**
  - Held in a 3-D spatial data structure
  - Surfaces need not be parameterized
- Photons **collected by ray tracing** from eye

Photons

- A **photon** is a particle of light that carries flux, which is encoded as follows
  - magnitude (in Watts) and color of the flux it carries, stored as an RGB triple
  - location of the photon (on a diffuse surface)
  - the incident direction (used to compute irradiance)

- **Example** (point light source, photons emitted uniformly)
  - Power of source (in Watts) distributed evenly among photons
  - Flux of each photon equal to source power divided by total # of photons
  - 60W light bulb would sending 100 photons, will result in 0.6 W per photon
How does this actually work?

Special data structures are required to do fast look-up (KD-trees)

Photon Mapping Results

Radiance estimate using 50 photons
Radiance estimate using 500 photons