Soft Shadows & Global Illumination

Course Topics for the Rest of the Term

Ray-tracing & Global Illumination
- This week

Parametric Curves/Surfaces
- March 30/April 1
- Taught by Robert Bridson - I will be at a conference

Overview of current research
- April 3/6 (lo Ihrke - I am still at conference)

April 8 – Final Q&A (I will be back for that)

Area Light Sources

Area lights produce soft shadows:
- In 2D:

Point lights:
- Only one light direction:
  \[ L_{\text{shadow}} = \rho \cdot V \cdot l_{\text{light}} \]
- \( V \) is visibility of light (0 or 1)
- \( \rho \) is lighting model (e.g. diffuse or Phong)

Course News

Assignment 3 (project)
- Due April 1
- Demos in labs April 2-7

Reading
- Chapter 10 (ray tracing), except 10.8-10.10
- Chapter 14 (global illumination)
Are Light Sources

**Area Lights:**
- Infinitely many light rays
- Need to integrate over all of them.
  \[ I_{\text{reflected}} = \int_{\text{light directions}} \rho(\omega) \cdot V(\omega) \cdot I_{\text{light}}(\omega) \cdot d\omega \]
- Lighting model, visibility and light intensity can now be different for every ray!

Integrating over Light Source

**Rewrite the integration**
- Instead of integrating over directions
  \[ I_{\text{reflected}} = \int_{\text{light directions}} \rho(\omega) \cdot V(\omega) \cdot I_{\text{light}}(\omega) \cdot d\omega \]
  we can integrate over points on the light source
  \[ I_{\text{reflected}}(q) = \int \frac{\rho(p - q) \cdot V(p - q) \cdot I_{\text{light}}(p) \cdot ds \cdot dt}{|p - q|^2} \]
  where \( q \) point on reflecting surface, \( p = F(s, t) \) is a point on the area light
  - We are integrating over \( p \)
  - Denominator: quadratic falloff

Integration

**Problem:**
- Except for the simplest of scenes, either integral is **not solvable analytically**!
- This is mostly due to the visibility term, which could be arbitrarily complex depending on the scene

**So:**
- Use numerical integration
- Effectively: approximate the light with a whole number of point lights

Numerical Integration

**Regular grid of point lights**
- Problem: will see 4 hard shadows rather than soft shadow
- Need LOTS of points to avoid this problem

Monte Carlo Integration

**Better:**
- **Randomly** choose the points
- Use different points on light for computing the lighting in different points on reflecting surface
- This produces random noise
- Visually preferable to structured artifacts
Monte Carlo Integration

**Formally:**
- Approximate integral with finite sum
  \[ I_{\text{approx}}(q) = \int_{p} p(p - q) \cdot V(p - q) \cdot I_{\text{light}}(p) \cdot ds \cdot dt \]
  \[ = \frac{1}{N} \sum_{i=1}^{N} p(p_i - q) \cdot V(p_i - q) \cdot I_{\text{light}}(p_i) \]

where
- The \( p \) are randomly chosen on the light source
- With equal probability!
- \( N \) is the total area of the light

Note:
- This approach of approximating lighting integrals with sums over randomly chosen points is much more flexible than this!
- In particular, it can be used for global illumination
  - Light bouncing off multiple surfaces before hitting the eye

Sampling

**Sample directions vs. sample light source**
- Most directions do not correspond to points on the light source
  - Thus, variance will be higher than sampling light directly

Global Illumination

**So far:**
- Have considered only light directly coming from the light sources
  - As well as mirror reflections, refraction

**In reality:**
- Light bouncing off diffuse and/or glossy surfaces also illuminates other surfaces
  - This is called global illumination

Direct Illumination

Global Illumination
Rendering Equation

Equation guiding global illumination:
\[
L_i(x,\omega_i) = L_e(x,\omega_i) + \int \rho(x,\omega_i,\omega_o) L_e(x,\omega_o) d\omega_o
\]

Where
- \( \rho \) is the reflectance from \( \omega_o \) to \( \omega_i \) at point \( x \)
- \( L_i \) is the outgoing (i.e. reflected) \textit{radiance} at point \( x \) in direction \( \omega_i \)
  - Radiance is a specific physical quantity describing the amount of light along a ray
  - Radiance is constant along a ray
- \( L_e \) is the emitted radiance (=0 unless point \( x \) is on a light source)
- \( R \) is the "ray-tracing function". It describes what point it will hit next.

Note:
- The rendering equation is an \textit{integral equation}
- This equation cannot be solved directly
  - Ray-tracing function is complicated!
  - Similar to the problem we had computing illumination from area light sources!

Ray Casting

- Cast a ray from the eye through each pixel
- The following few slides are from Fred Durand (MIT)

Ray Tracing

- Cast a ray from the eye through each pixel
- Trace secondary rays (light, reflection, refraction)

Monte Carlo Ray Tracing

- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
  - Accumulate radiance contribution

Monte Carlo Ray Tracing

- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
- Recurse
Monte Carlo
- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
- Recurse

Monte Carlo Path Tracing

**In practice:**
- Do not branch at every intersection point
- This would have exponential complexity in the ray depth!
- Instead:
- Shoot some number of primary rays through the pixel (10s-1000s, depending on scene!)
- For each pixel and each intersection point, make a single random decision in which direction to go next

How to Sample?

**Simple sampling strategy:**
- At every point, choose between all possible reflection directions with equal probability
- This will produce very high variance/noise if the materials are specular or glossy
- Lots of rays are required to reduce noise!

**Better strategy: Importance sampling**
- Focus rays in areas where most of the reflected light contribution will be found
- For example: if the surface is a mirror, then only light from the mirror direction will contribute!
- Glossy materials: prefer rays near the mirror

Monte Carlo Path Tracing
- Systematically sample primary light

Monte Carlo Path Tracing
- Trace only one secondary ray per recursion
- But send many primary rays per pixel
- (performs antialiasing as well)
How to Sample?

**Sampling strategies are still an active research area!**
- Recent years have seen drastic advances in performance
- Lots of excellent sampling strategies have been developed in statistics and machine learning
  - Many are useful for graphics

Objective:
- Compute light transport in scenes using stochastic ray tracing
  - Monte Carlo, Sequential Monte Carlo
  - Metropolis

[Barthe, Gharbi, Heidrich '05]
[Gharbi, Heidrich '06]
[Gharbi, Doucet, Heidrich '08]

How to Sample?

- E.g. importance sampling (left) vs. Sequential Monte Carlo (right)

More on Global Illumination

**This was a (very) quick overview**
- More details in CPSC 514 (Computer Graphics: Rendering)
- Next offered in January 2010

**Coming Up**

**Monday/Wednesday:**
- Curves & surfaces (Robert Bridson)

**Friday:**
- Overview of current research topics (Ivo Ihrke)

**Monday (April 8):**
- Research demos (Ivo & my PhD students)