Lecture 11: Irradiance Caching

CS 6620, Spring 2009

Kavita Bala

Computer Science

Cornell University

Pure Path Tracing

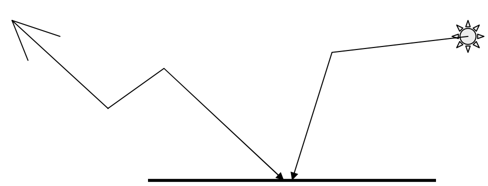
- Advantages
  - No need for meshing
  - General surfaces – requires ray intersections
  - *Unbiased* estimates
- Disadvantages
  - Too noisy/slow
  - Noise is objectionable
  - Treats every pixel independently, and every point on every surface independently
    - Starts from scratch – does not exploit coherence

Biased Methods

- Biased methods: store information (caching)
  - Better type of noise: blurring
- Techniques
  - Greg Ward's Radiance
  - Photon Mapping
  - Radiosity (even)
  - Assumption: common case is diffuse

Path Re-Use

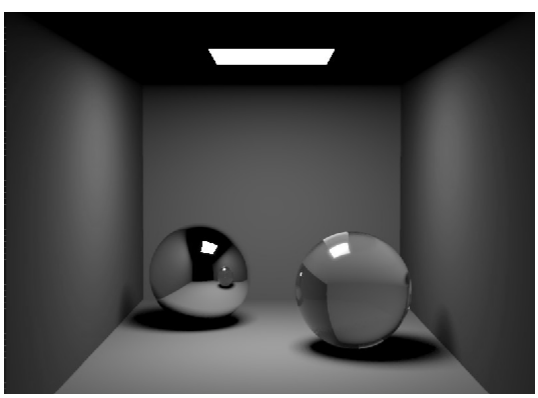
- What is coherence?
  - Nearby values are similar to what we want



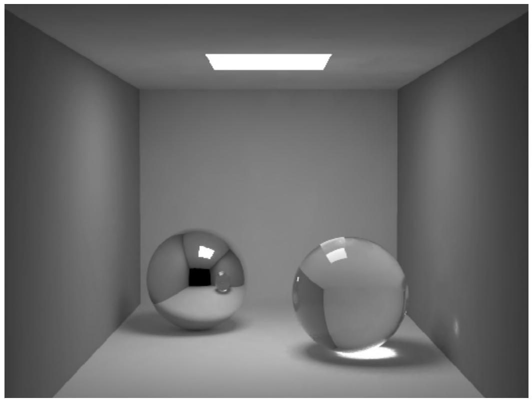
Irradiance Caching

- Introduced by Greg Ward 1988
- Implemented in RADIANCE
  – Public-domain software
- Exploits smoothness of irradiance
  – Cache and interpolate irradiance estimates
  – Also has error, but the right kind of error

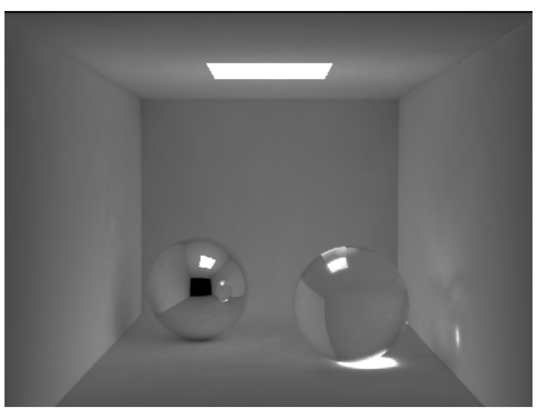
Direct Illumination



GI



Indirect Irradiance



© Kavita Bala, Computer Science, Cornell University

Smoothness

- Diffuse reflectance varies slowly over a surface
  - Incoming direction varies slowly
  - Reuse paths: “Cache” directionally invariant irradiance when possible
- Irradiance tends to vary slowly over a surface
  - e.g., directional light gives constant irradiance over a planar surface
  - point light varies slowly with the cosine of the angle

Basic Idea

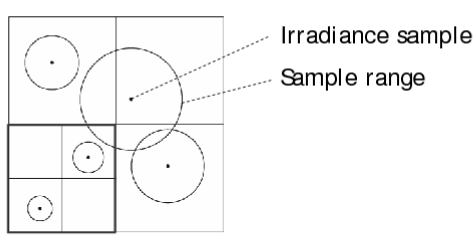
- When we need irradiance at a new point
  - Compute irradiance estimate using cached samples
- The sub-problems are:
  - How do you get the samples in the first place?
  - How to you store/cache them?
  - How do we compute the irradiance estimate?
  - When do we use this cache of irradiance values?

Basic Idea

- Store information about previous paths
  – Compute using Monte Carlo
- Maintain a spatial data structure storing irradiance arriving at points
- If you hit a diffuse surface
  – Compute direct illumination
  – Try to estimate irradiance at that point
    - If the estimate is good, use it
    - If not, use path tracing to estimate the irradiance and store it

Where to store it?

- Some sort of spatial data structure
  – Support insertion of new samples, and search for nearby samples, indexed on position
  – RADIANCE system uses Octrees

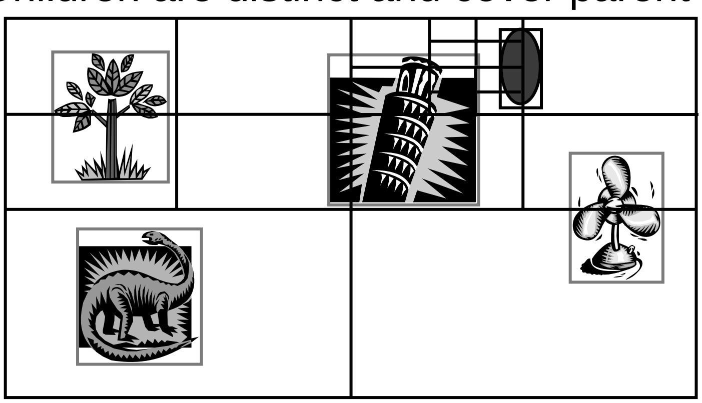


Storing Samples

- Question of precisely what to store:
  - Irradiance (Ei)
  - The location (xi)
  - The normal at that location (ni)
  - The distance that we went to find nearby surfaces (Ri)

Spatial Hierarchy

- Hierarchical spatial subdivision
  - Divides up space
- Children are distinct and cover parent



Where do samples come from?

- Irradiance computed using MC

$$E = \int_{\substack{Solid\\Angle}} L(x \leftarrow \Psi) \cdot \cos\theta \cdot d\omega_\Psi$$

$$E = \int \int L(x \leftarrow \Psi) \cdot \cos\theta \cdot \sin\theta \cdot d\theta d\phi$$

$$E = \frac{\pi}{TP} \sum_t \sum_p L_i(\theta_t, \phi_p)$$

- Path tracing to compute samples
  - Stratify outgoing directions according to cosine

Cosine distribution

$$f = \frac{1}{\pi} \int_0^{2\pi} \int_0^1 \cos\theta \sin\theta d\theta d\phi$$

$$p(\theta, \phi) = \frac{\cos\theta \sin\theta}{\pi}$$

$$CDF(\theta, \phi) = \int_0^\theta \int_0^\phi \frac{\cos\theta \sin\theta}{\pi} d\theta d\phi = (1 - \cos^2\theta) \frac{\phi}{2\pi}$$

$$F(\theta) = 1 - \cos^2\theta$$

$$F(\phi) = \frac{\phi}{2\pi}$$

$$\phi_i = 2\pi u_1 \qquad \theta_i = \cos^{-1}\sqrt{u_2}$$

What information to store?

$$\theta_t = \sin^{-1}\left(\sqrt{\frac{t-u_2}{T}}\right)$$

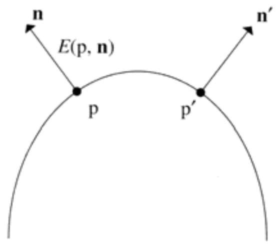
$$\phi_p = 2\pi\frac{p-u_1}{P}$$

- Compute Li using Monte Carlo

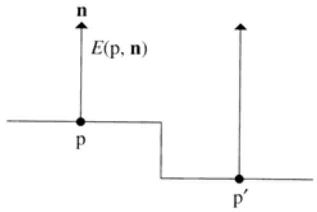
Quality of Existing Data

- The quality of an estimate is based on geometric considerations
- Examine:
  - Surface curvature between the required point and the existing data
    - diffuse illumination changes with curvature
  - Brightness and the distance to other surfaces
    - Influences how fast incoming illumination can change
- If existing values are used, weigh contribution by quality

Geometric Factors



Points and normals should be close



Distance to nearby surfaces should be large – corners lead to fast changes in irradiance

Validity of Sample

- Assign a region where a sample can be reused
- Larger region where irradiance varies smoothly
- Smaller region when it varies a lot

Distance to surfaces

- Harmonic mean heuristic
- Range of sample is given by a radius Ri

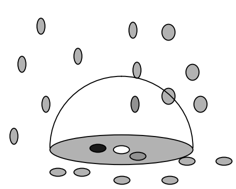
$$\frac{1}{R_i} = \frac{1}{N}\sum \frac{1}{d_j}$$

- Where dj is distance to closest surface
- Why not just average?

Smoothness Measure

- Find nearby samples
  - Query octree for samples that overlap p
  - Check ε at x, x_i is a nearby sample

$$\varepsilon_i(x,\vec{n}) = \frac{\| x_i - x \|}{R_i} + \sqrt{1 - \vec{n} \bullet \vec{n}_i}$$

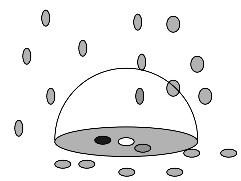


  - Weight samples inversely proportional to ε_i

Smoothness Measure

- Find nearby samples

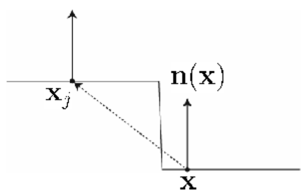
$$E(x,\vec{n}) = \frac{\sum_{i,w_i>1/a,isValid(i)} w_i(x,\vec{n})E_i(x_i)}{\sum_{i,w_i>1/a,isValid(i)} w_i(x,\vec{n})}$$




```
W = 0
for( all irradiance samples j in octree cell
      overlapping with x ) {
  compute weight w_j
  if( <sample is valid> ) {
    W += w_j; wE += w_j*E[j]
  }
}

if( W > 0 ) {
  return wE/W
} else {
  return( compute new irradiance sample )
}
```

© Kavita Bala, Computer Science, Cornell University

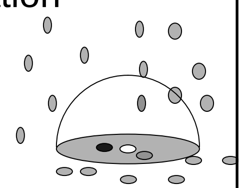
```
<sample is valid> =

dist( x - x[j] ) < r[j] // within range
&& w_j > 1/a            // sufficient weight
&& dot( x[j] - x, n(x) ) < 0
                        // x[j] is behind x
```



Sample at $\mathbf{x}_j$ is invalid

How to compute the irradiance estimate?

- When new sample requested
  - Query octree for samples near location
  - Check ε at x, x_i is a nearby sample

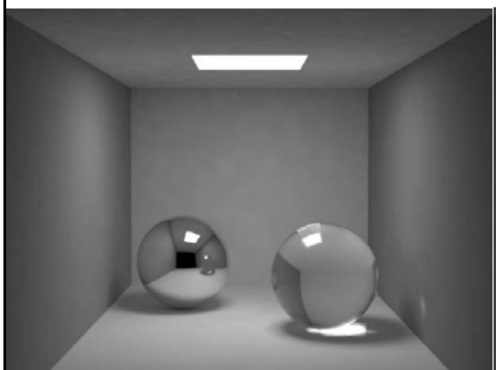


$$\varepsilon_i(x,\vec{n}) = \frac{\| x_i - x \|}{R_i} + \sqrt{1 - \vec{n} \bullet \vec{n}_i}$$

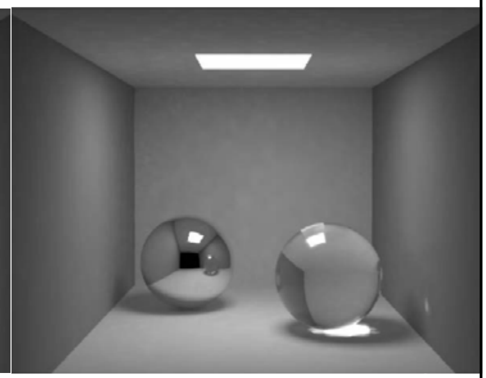
  - Weight samples inversely proportional to ε_i

$$E(x,\vec{n}) = \frac{\sum_{i, w_i > 1/a} w_i(x,\vec{n}) E_i(x_i)}{\sum_{i, w_i > 1/a} w_i(x,\vec{n})}$$

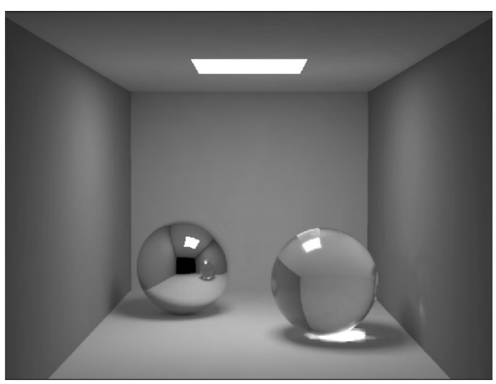
  - Otherwise, compute new sample



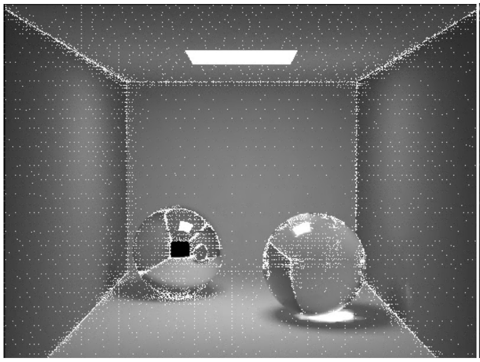
1000 sample rays, w > 10

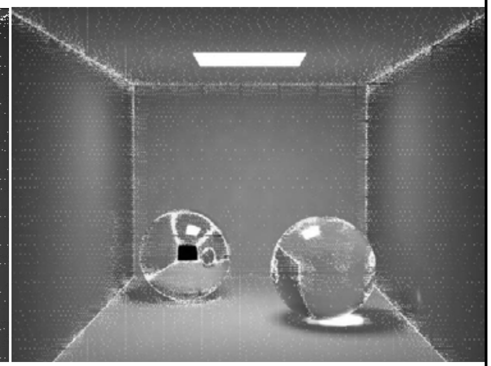


1000 sample rays, w > 20



5000 sample rays, w > 10





1000 sample rays, w > 10

1000 sample rays, w > 20

[Humphreys, Pharr]

Radiance Examples

Radiance: Example

Summary algorithm

- Assume: cache diffuse irradiance
- Start with a basic path tracing algorithm, but …
  - Store irradiance in octree
  - If hit diffuse surface, build an estimate of irradiance at that point
  - Always compute direct lighting explicitly
  - If estimate is good, use it
  - If not, use path tracing to estimate the irradiance and store it

Implementation Notes

- In a full system, how to handle diffuse-only?
  - Error goes up as reflectance varies more from diffuse
- Specular bounces are traced through to a diffuse surface
  - Be careful not to double count
- Must account for transmission and reflection
  - Separate irradiance estimates for both sides of a surface