Computer Graphics

CSE 167 [Win 24], Lecture 19: High Quality Rendering

Ravi Ramamoorthi

http://viscomp.ucsd.edu/classes/cse167/wi24

1

Summary

- Good luck on HW 4, written assignment
- Please consider CSE 168 (Rendering) in spring

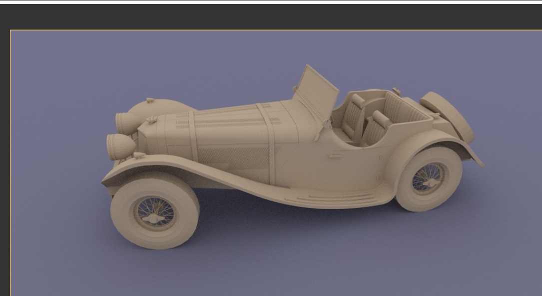
2

Monte Carlo Path Tracing

- General solution to rendering and global illumination
- Suitable for a variety of general scenes
- Based on Monte Carlo methods
- Enumerate all paths of light transport
- Long history, traces back to rendering eqn Kajiya 86
- (More advanced topic: Slides from CSE 168/274)
- Increasingly, basis for production rendering
- Path tracing today real-time in hardware (for example, using NVIDIA's Optix, Turing, Ada RTX)

3

Monte Carlo Path Tracing

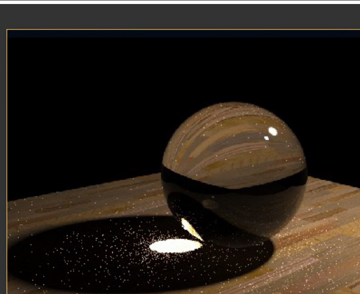


Big diffuse light source, 20 minutes

Jensen

4

Monte Carlo Path Tracing



1000 paths/pixel

Jensen

5

Monte Carlo Path Tracing

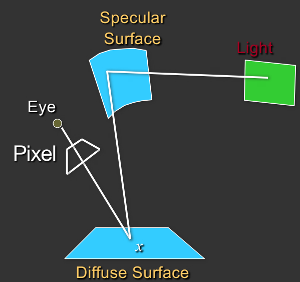
Advantages
- Any type of geometry (procedural, curved, ...)
- Any type of BRDF or reflectance (specular, glossy, diffuse, ...)
- Samples all types of paths (L(SD)*E)
- Accuracy controlled at pixel level
- Low memory consumption
- Unbiased - error appears as noise in final image

Disadvantages (standard Monte Carlo problems)
- Slow convergence (square root of number of samples)
- Noise in final image

6

Monte Carlo Path Tracing

Integrate radiance
for each pixel
by sampling paths
randomly



$$L_o(x,\vec{w}) = L_e(x,\vec{w}) + \int_\Omega f_r(x,\vec{w}',\vec{w}) L_i(x,\vec{w}')(\vec{w}' \bullet \vec{n}) d\vec{w}$$

7

Simplest Monte Carlo Path Tracer

For each pixel, cast n samples and average
- Choose a ray with *p*=camera, *d*=(θ,ϕ) within pixel
- Pixel color += (1/n) * TracePath(*p*, *d*)

TracePath(*p*, *d*) returns (r,g,b) [and calls itself recursively]:
- Trace ray (*p*, *d*) to find nearest intersection *p'*
- Select with probability (say) 50%:
  - Emitted:
    return 2 * (Le_{red}, Le_{green}, Le_{blue}) // 2 = 1/(50%)
  - Reflected:
    generate ray in random direction *d'*
    return 2 * f_r(*d* ➔ *d'*) * (*n*•*d'*) * TracePath(*p'*, *d'*)

8

Simplest Monte Carlo Path Tracer

For each pixel, cast n samples and average over paths
- Choose a ray with *p*=camera, *d*=(θ,ϕ) within pixel
- Pixel color += (1/n) * TracePath(*p*, *d*)

TracePath(*p*, *d*) returns (r,g,b) [and calls itself recursively]:
- Trace ray (*p*, *d*) to find nearest intersection *p'*
- Select with probability (say) 50%:
  - Emitted:
    return 2 * (Le_{red}, Le_{green}, Le_{blue}) // 2 = 1/(50%)
  - Reflected:
    generate ray in random direction *d'*
    return 2 * f_r(*d* ➔ *d'*) * (*n*•*d'*) * TracePath(*p'*, *d'*)

9

Simplest Monte Carlo Path Tracer

For each pixel, cast n samples and average
- Choose a ray with *p*=camera, *d*=(θ,ϕ) within pixel
- Pixel color += (1/n) * TracePath(*p*, *d*)

TracePath(*p*, *d*) returns (r,g,b) [and calls itself recursively]:
- Trace ray (*p*, *d*) to find nearest intersection *p'*
- Select with probability (say) 50%:
  - Emitted:
    return 2 * (Le_{red}, Le_{green}, Le_{blue}) // 2 = 1/(50%)
  - Reflected:
    generate ray in random direction *d'*
    return 2 * f_r(*d* ➔ *d'*) * (*n*•*d'*) * TracePath(*p'*, *d'*)

Weight = 1/probability
Remember: unbiased requires having f(x) / p(x)

10

Simplest Monte Carlo Path Tracer

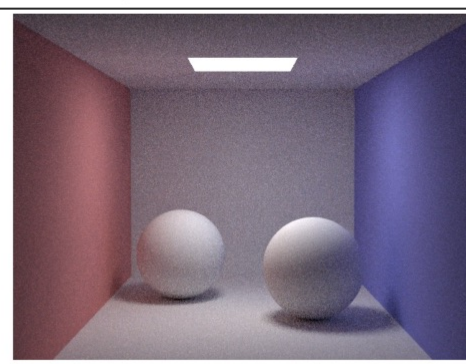
For each pixel, cast n samples and average
- Choose a ray with *p*=camera, *d*=(θ,ϕ) within pixel
- Pixel color += (1/n) * TracePath(*p*, *d*)

TracePath(*p*, *d*) returns (r,g,b) [and calls itself recursively]:
- Trace ray (*p*, *d*) to find nearest intersection *p'*
- Select with probability (say) 50%:
  - Emitted:
    return 2 * (Le_{red}, Le_{green}, Le_{blue}) // 2 = 1/(50%)
  - Reflected:
    generate ray in random direction *d'*
    return 2 * f_r(*d* ➔ *d'*) * (*n*•*d'*) * TracePath(*p'*, *d'*)

Path terminated when Emission evaluated

11

Path Tracing



CS348B Lecture 14 | 10 paths / pixel | Pat Hanrahan, Spring 2009

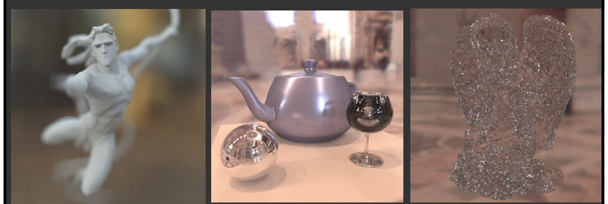
12

Arnold Renderer (M. Fajardo)

- Works well diffuse surfaces, hemispherical light

13

From UCB CS 294 a few years ago



Daniel Ritchie and Lita Cho

14

Importance Sampling

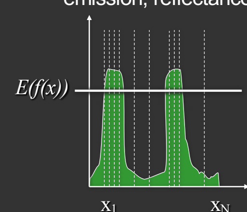
- Pick paths based on energy or expected contribution
  - More samples for high-energy paths
  - Don't pick low-energy paths
- At "macro" level, use to select between reflected vs emitted, or in casting more rays toward light sources
- At "micro" level, importance sample the BRDF to pick ray directions
- Tons of papers in 90s on tricks to reduce variance in Monte Carlo rendering
- Importance sampling now standard in production. I consulted on Pixar's system for movies from 2012+

15

Importance Sampling

Can pick paths however we want, but contribution weighted by 1/probability

- Already seen this division of 1/prob in weights to emission, reflectance



$$\int_\Omega f(x)dx = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

$$Y_i = \frac{f(x_i)}{p(x_i)}$$

16

Importance sample Emit vs Reflect

TracePath(*p*, *d*) returns (r,g,b) [and calls itself recursively]:

- Trace ray (*p*, *d*) to find nearest intersection *p'*
- If Le = (0,0,0) then p_{emit}= 0 else p_{emit}= 0.9 (say)
- If random() < p_{emit} then:
  - Emitted:
    return (1/ p_{emit}) * (Le_{red}, Le_{green}, Le_{blue})
  - Else Reflected:
    generate ray in random direction *d'*
    return (1/(1- p_{emit})) * f_r(*d* → *d'*) * (*n*•*d'*) * TracePath(*p'*, *d'*)

17

More variance reduction

- Discussed "macro" importance sampling
  - Emitted vs reflected
- How about "micro" importance sampling
  - *Shoot rays towards light sources in scene*
  - Distribute rays according to BRDF

18

Path Tracing: Include Direct Lighting

```
Step 1. Choose a camera ray r given the
   (x,y,u,v,t) sample
   weight = 1;
   L = 0
Step 2. Find ray-surface intersection
Step 3.
   L += weight * Lr(light sources)
   weight *= reflectance(r)
   Choose new ray r' ~ BRDF pdf(r)
       Go to Step 2.
```

CS348B Lecture 14 Pat Hanrahan, Spring 2009

19

Monte Carlo Extensions

Unbiased
- Bidirectional path tracing
- Metropolis light transport

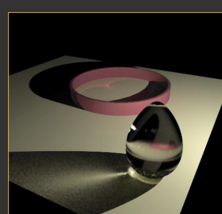
Biased, but consistent
- Noise filtering
- Adaptive sampling
- Irradiance caching

20

Monte Carlo Extensions

Unbiased
- Bidirectional path tracing
- Metropolis light transport

Biased, but consistent
- Noise filtering
- Adaptive sampling
- Irradiance caching



RenderPark

21

Monte Carlo Extensions

Unbiased
- Bidirectional path tracing
- Metropolis light transport

Biased, but consistent
- Noise filtering
- Adaptive sampling
- Irradiance caching

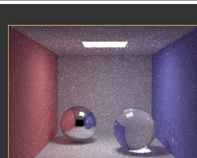
Heinrich

22

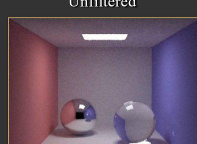
Monte Carlo Extensions

Unbiased
- Bidirectional path tracing
- Metropolis light transport

Biased, but consistent
- Noise filtering
- Adaptive sampling
- Irradiance caching



Unfiltered



Filtered

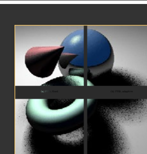
Jensen

23

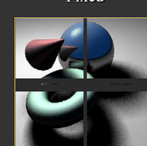
Monte Carlo Extensions

Unbiased
- Bidirectional path tracing
- Metropolis light transport

Biased, but consistent
- Noise filtering
- Adaptive sampling
- Irradiance caching



Fixed



Adaptive

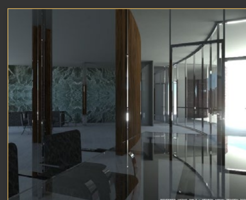
Ohbuchi

24

Monte Carlo Extensions

Unbiased
- Bidirectional path tracing
- Metropolis light transport

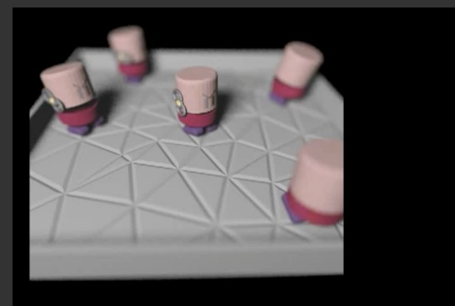
Biased, but consistent
- Noise filtering
- Adaptive sampling
- Irradiance caching



Jenser

25

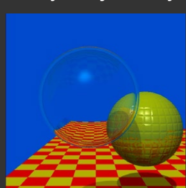
Monte Carlo Denoising



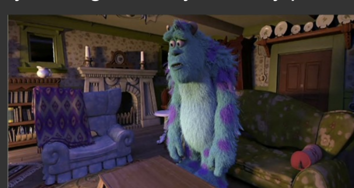
26

Impact: Real-Time

- Extend AAF, FSF, MAAF: Predict Filter based on Deep Learning (sample and AI-based denoising)
- NVIDIA software (OptiX 2017), hardware (RTX 2018)
- 40-year journey: ray tracing curiosity to every pixel



Whitted 79 (74 min 512x512)



NVIDIA RTX 2018, OptiX; Pixar real-time previewer

27

Today: Real-Time Denoising at 1spp

NVIDIA Tiger Demo 2021

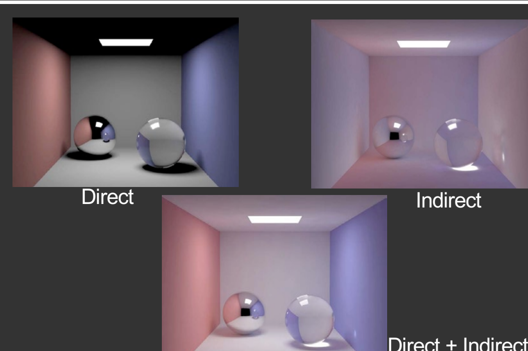
28

Summary

- Monte Carlo methods robust and simple (at least until nitty gritty details) for global illumination
- Must handle many variance reduction methods in practice
- Importance sampling, Bidirectional path tracing, Russian roulette etc.
- Rich field with many papers, systems researched over last 30 years
- Today, hardware for real-time ray, path tracing
- Promising physically-based GPU approach

29

Smoothness of Indirect Lighting



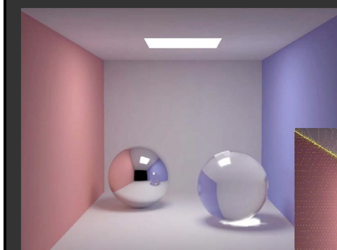
30

Irradiance Caching

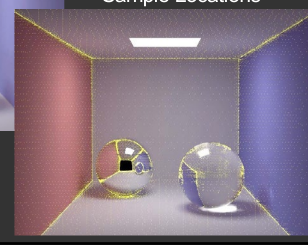
- Empirically, (diffuse) interreflections low frequency
- Therefore, should be able to sample sparsely
- Irradiance caching samples irradiance at few points on surfaces, and then interpolates
- Ward, Rubinstein, Clear. SIGGRAPH 88, *A ray tracing solution for diffuse interreflection*

31

Irradiance Caching Example



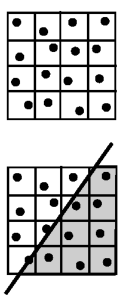
Final Image



Sample Locations

32

Stratified Sampling



Stratified sampling like jittered sampling

Allocate samples per region

$$N = \sum_{i=1}^{m} N_i \quad F_N = \frac{1}{N}\sum_{i=1}^{m} N_i F_i$$

New variance

$$V[F_N] = \frac{1}{N^2}\sum_{i=1}^{m} N_i V[F_i]$$

Thus, if the variance in regions is less than the overall variance, there will be a reduction in resulting variance

For example: An edge through a pixel

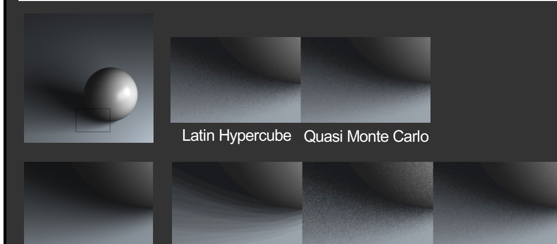
$$V[F_N] = \frac{1}{N^2}\sum_{i=1}^{\sqrt{N}} V[F_j] = \frac{V[F_E]}{N^{1.5}}$$

CS348B Lecture 9 Pat Hanrahan, Spring 2002

D. Mitchell 95, Consequences of stratified sampling in graphics

33

Comparison of simple patterns



16 samples for area light, 4 samples per pixel, total 64 samples

If interested, see my recent paper "A Theory of Monte Carlo Visibility Sampling"

Figures courtesy Tianyu Liu

34

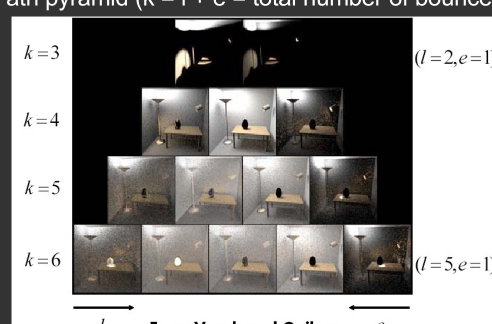
Path Tracing: From Lights

- Step 1. Choose a light ray
- Step 2. Find ray-surface intersection
- Step 3. Reflect or transmit
  u = Uniform()
  if u < reflectance(x)
    Choose new direction d ~ BRDF(O|I)
  goto Step 2
- else if u < reflectance(x)+transmittance(x)
    Choose new direction d ~ BTDF(O|I)
  goto Step 2
- else // absorption=1–reflectance-transmittance
  terminate on surface; deposit energy

35

Bidirectional Path Tracing

Path pyramid (k = l + e = total number of bounces)



36

Comparison

Bidirectional path tracing **Path tracing**

From Veach and Guibas

37

Why Photon Map?

- Some visual effects like caustics hard with standard path tracing from eye
- May usually miss light source altogether
- Instead, store "photons" from light in kd-tree
- Look-up into this as needed
- Combines tracing from light source, and eye
- Similar to bidirectional path tracing, but compute photon map only once for all eye rays
- *Global Illumination using Photon Maps H. Jensen. Rendering Techniques (EGSR 1996), pp 21-30.* (Also book: *Realistic Image Synthesis using Photon Mapping*)

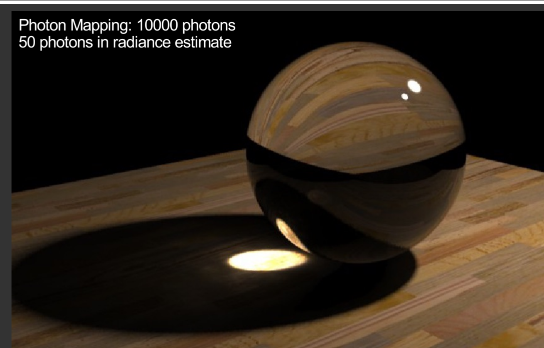
38

Caustics

Slides courtesy Henrik Wann Jensen

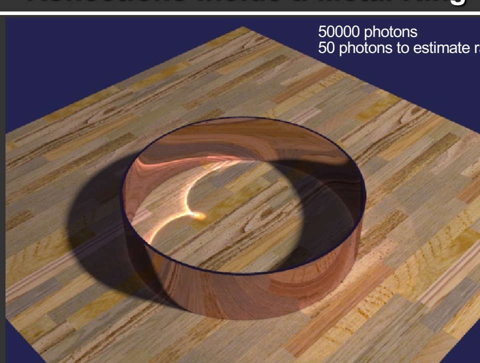
39

Caustics



40

Reflections Inside a Metal Ring



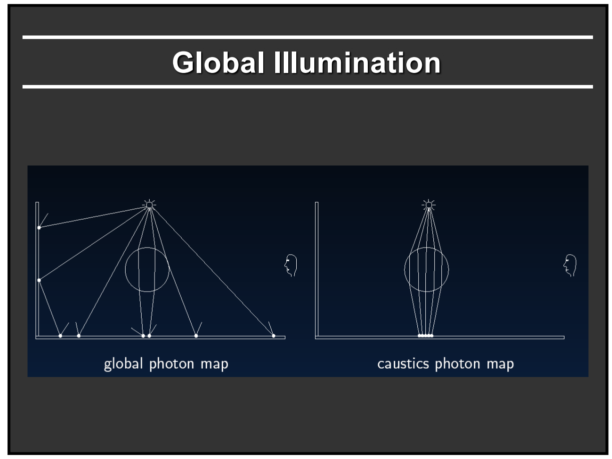
41

Caustics on Glossy Surfaces

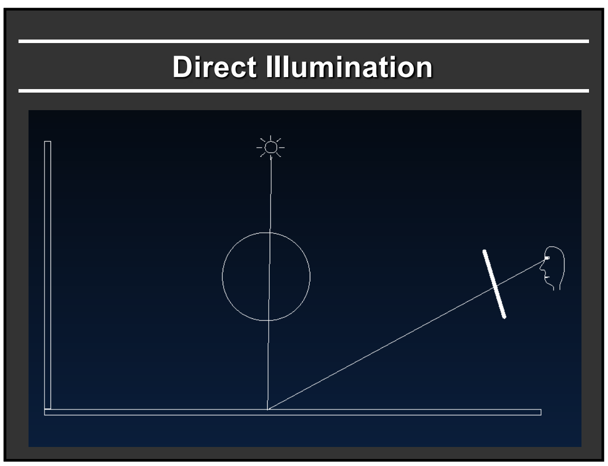
340000 photons, 100 photons in radiance estimate

42

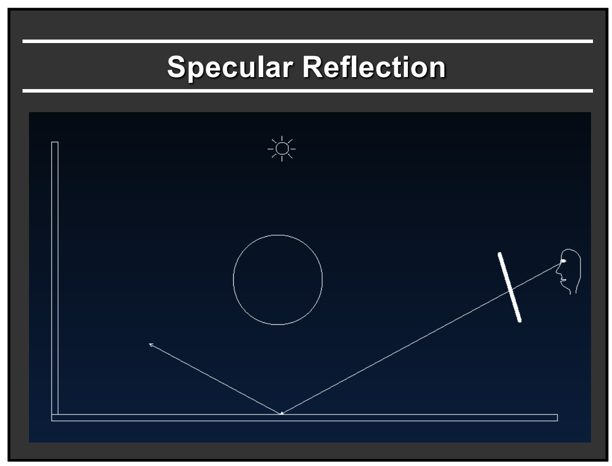
43



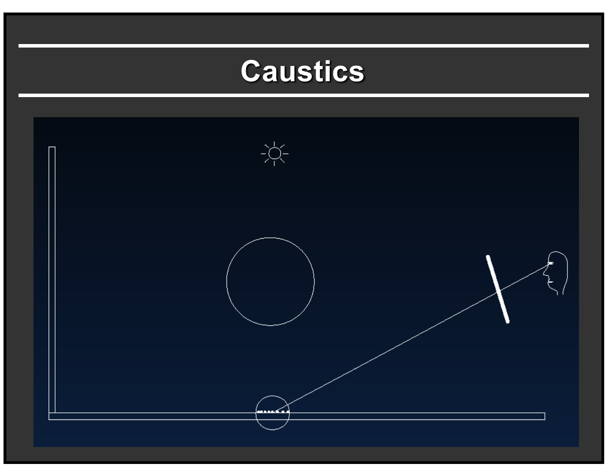
44



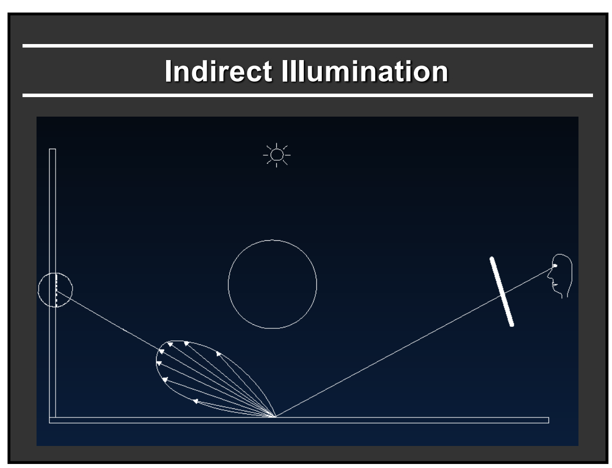
45



46

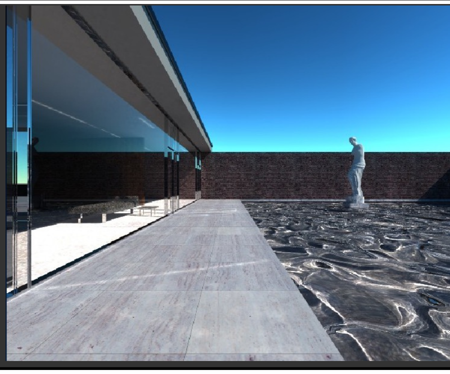


47



48

Mies House: Swimming Pool



49