Monte Carlo Rendering

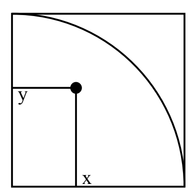
Last Time?

Today

- Monte-Carlo Integration
- Probabilities and Variance
- Analysis of Monte-Carlo Integration
- Monte-Carlo in Graphics
- Stratified Sampling
- Importance Sampling
- Advanced Monte-Carlo Rendering

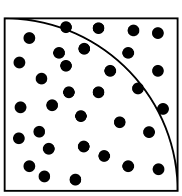
Monte-Carlo computation of π

- Take a square
- Take a random point (x,y) in the square
- Test if it is inside the ¼ disc ($x^2+y^2 < 1$)
- The probability is π /4



Monte-Carlo computation of π

- The probability is π /4
- Count the inside ratio n = # inside / total # trials
- $\pi \approx n * 4$
- The error depends on the number or trials



Convergence & Error

- Let's compute 0.5 by flipping a coin:
  - 1 flip: 0 or 1
    → average error = 0.5
  - 2 flips: 0, 0.5, 0.5 or 1
    → average error = 0. 25
  - 4 flips: 0 (*1),0.25 (*4), 0.5 (*6), 0.75(*4), 1(*1)
    → average error = 0.1875
- Does not converge very fast
- Doubling the number of samples does not double accuracy

Questions?

Today

- Monte-Carlo Integration
- Probabilities and Variance
- Analysis of Monte-Carlo Integration
- Monte-Carlo in Graphics
- Stratified Sampling
- Importance Sampling
- Advanced Monte-Carlo Rendering

Review of Probability (discrete)

- Random variable can take discrete values x_i
- Probability p_i for each x_i

  $0 < p_i < 1$, $\Sigma p_i = 1$
- Expected value $E(x) = \sum_{i=1}^{n} p_i x_i$
- Expected value of function of random variable
  - $f(x_i)$ is also a random variable

$$E[f(x)] = \sum_{i=1}^{n} p_i f(x_i)$$

Variance & Standard Deviation

- Variance σ^2: deviation from expected value
- Expected value of square difference

$$\sigma^2 = E[(x - E[x])^2] = \sum_i (x_i - E[x])^2 p_i$$

- Also

$$\sigma^2 = E[x^2] - (E[x])^2$$

- Standard deviation σ:
  square root of variance (notion of error, RMS)

Monte Carlo Integration

- Consider N random samples over domain with probability *p(x)*
- Define estimator:

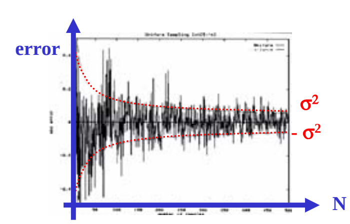
$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

- Probability *p* allows us to sample the domain more intelligently

Example

$$I = \int_0^1 5x^4 dx$$

- We know it should be 1.0
- In practice with uniform samples:



Monte Carlo Analysis

- We want to compute the average of a function
- We can pick a random value of this function and hope to fall in the middle
  - chances are slim
  - but on average we'll be right
- For N tries,
  - the expectation stays the same
  - but the variance decreases by N
  - the standard deviation (error) decreases by $\sqrt{N}$
    i.e. quadrupling the number of sampled points will halve the error, regardless of the number of dimensions

Advantages of MC Integration

- Few restrictions on the integrand
  - Doesn't need to be continuous, smooth, ...
  - Only need to be able to evaluate at a point
- Extends to high-dimensional problems
  - Same convergence
- Conceptually straightforward
- Efficient for solving at just a few points

Disadvantages of MC

- Noisy
- Slow convergence
- Good implementation is hard
  - Debugging code
  - Debugging math
  - Choosing appropriate techniques
- Punctual technique, no notion of smoothness of function

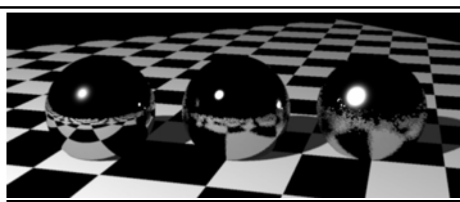
Monte Carlo Recap

- Turn integral into finite sum
- Use random samples
- Convergence $\frac{1}{\sqrt{n}}$
- Independent of dimension
- Very flexible

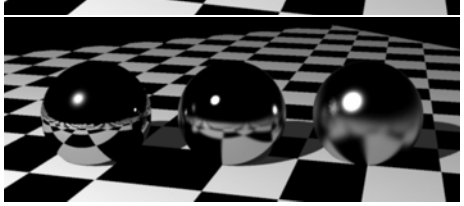
- Tweak sampling/probabilities for optimal result
- A lot of integration and probability theory to get things right

Questions?

1 glossy sample per pixel



256 glossy samples per pixel



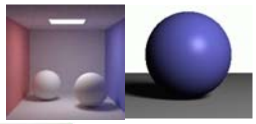
Today

- Monte-Carlo Integration
- Probabilities and Variance
- Analysis of Monte-Carlo Integration
- Monte-Carlo in Graphics
- Stratified Sampling
- Importance Sampling
- Advanced Monte-Carlo Rendering

What can we integrate?

- Pixel: antialiasing
- Light sources: Soft shadows
- Lens: Depth of field
- Time: Motion blur
- BRDF: glossy reflection
- Hemisphere: indirect lighting

$$\int\int\int\int\int L(x,y,t,u,v)\,dx\,dy\,dt\,du\,dv$$

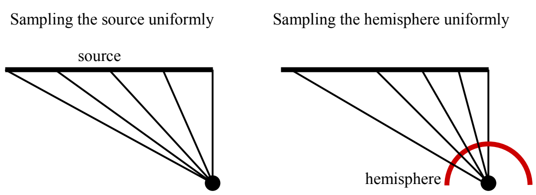


Domains of integration

- Pixel, lens (Euclidean 2D domain)
- Time (1D)
- Hemisphere
  - Work needed to ensure uniform probability
- Light source
  - Same thing: make sure that the probabilities and the measures are right.

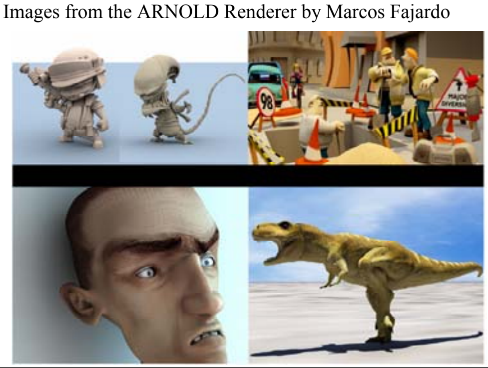
Example: Light source

- Integrate over surface or over angle
- Be careful to get probabilities and integration measure right!



Questions?

- Images from the ARNOLD Renderer by Marcos Fajardo



Today

- Monte-Carlo Integration
- Probabilities and Variance
- Analysis of Monte-Carlo Integration
- Monte-Carlo in Graphics
- Stratified Sampling
- Importance Sampling
- Advanced Monte-Carlo Rendering

Important issues in MC rendering

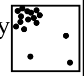
Reduce variance!

- Choose a smart probability distribution
- Choose smart sampling patterns

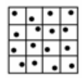
And of course, cheat to make it faster without being noticed

Stratified sampling

- With uniform sampling, we can get unlucky
  - E.g. all samples in a corner



- To prevent it, subdivide domain Ω into non-overlapping regions Ω_i
  - Each region is called a stratum



- Take one random samples per Ω_i

Example

- Borrowed from Henrik Wann Jensen

$f(x) = e^{\sin(3x^2)}$

N	I
1	2.75039
10	1.9893
100	1.79139
1000	1.75146
10000	1.77313
100000	1.77862

Unstratified $O(1/\sqrt{N})$

$f(x) = e^{\sin(3x^2)}$

N	I
1	2.70457
10	1.72858
100	1.77925
1000	1.77606
10000	1.77610
100000	1.77610

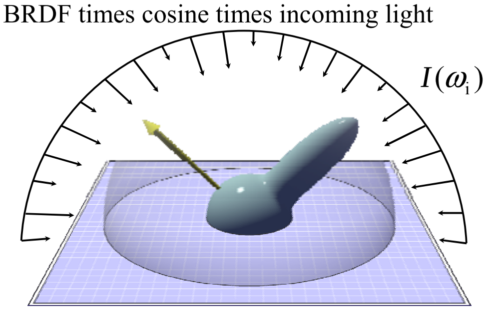
Stratified $O(1/N)$

Stratified Sampling Recap

- Cheap and effective
- Typical example: jittering for antialiasing
  - Signal processing perspective: better than uniform because less aliasing (spatial patterns)
  - Monte-Carlo perspective: better than random because lower variance (error for a given pixel)

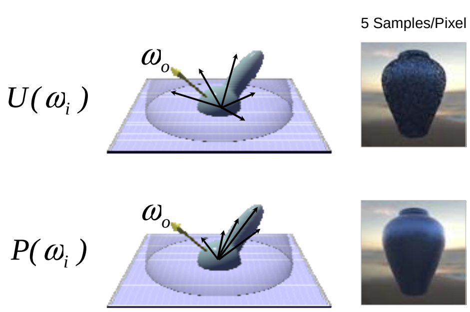
Glossy Rendering

- Integrate over hemisphere
- BRDF times cosine times incoming light



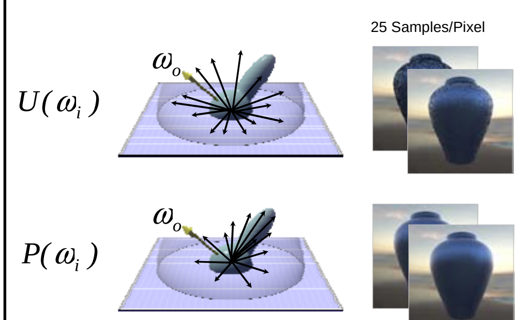
Slide from Jason Lawrence

Sampling a BRDF



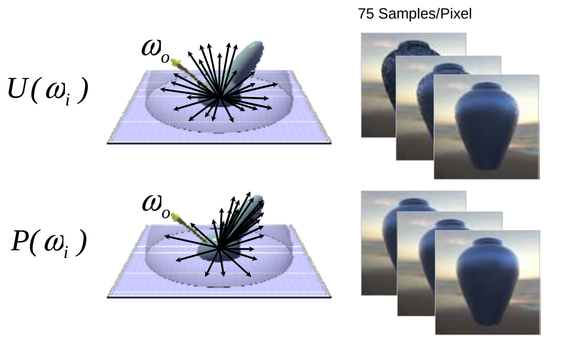
Slide from Jason Lawrence

Sampling a BRDF



Slide from Jason Lawrence

Sampling a BRDF

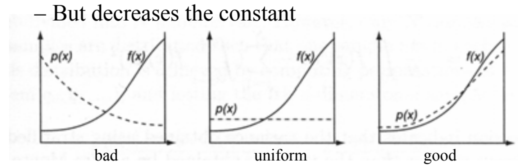


Slide from Jason Lawrence

Importance sampling

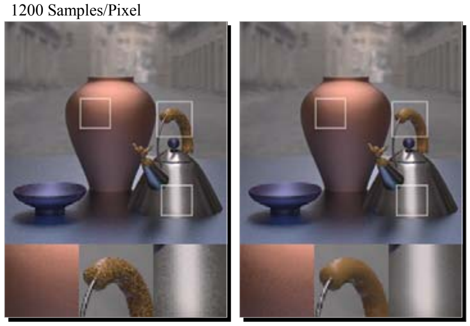
$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

- Choose p wisely to reduce variance
  - p that resembles f
  - Does not change convergence rate (still sqrt)
  - But decreases the constant



Results

1200 Samples/Pixel



Traditional importance function

Better importance by Lawrence et al.

Questions?

- Images by Veach and Guibas

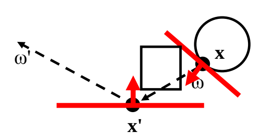
Naïve sampling strategy

Optimal sampling strategy

Today

- Monte-Carlo Integration
- Probabilities and Variance
- Analysis of Monte-Carlo Integration
- Monte-Carlo in Graphics
- Stratified Sampling
- Importance Sampling
- Advanced Monte-Carlo Rendering

The Rendering Equation

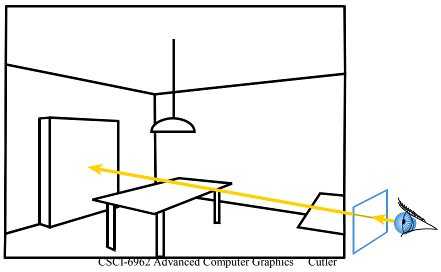


L(x',ω') = E(x',ω') + ∫ρ$_{x'}$(ω,ω')L(x,ω)G(x,x')V(x,x') dA

emission — BRDF — Incoming light — Geometric term — visibility

Ray Casting

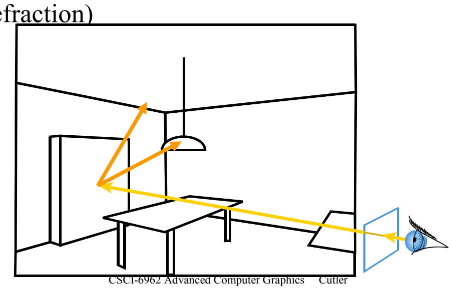
- Cast a ray from the eye through each pixel



CSCI-6962 Advanced Computer Graphics Cutler

Ray Tracing

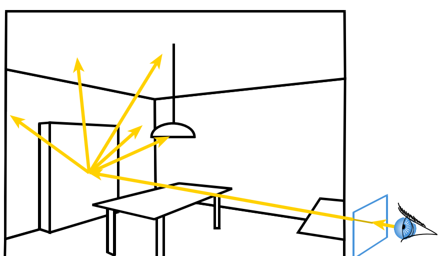
- Cast a ray from the eye through each pixel
- Trace secondary rays (light, reflection, refraction)



CSCI-6962 Advanced Computer Graphics Cutler

Monte-Carlo Ray Tracing

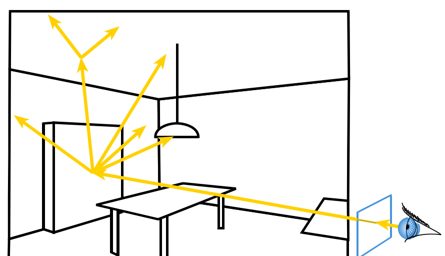
- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
  – Accumulate radiance contribution



CSCI-6962 Advanced Computer Graphics Cutler

Monte-Carlo Ray Tracing

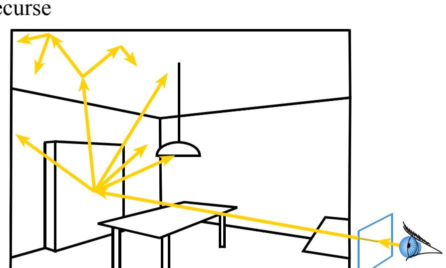
- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
- Recurse



CSCI-6962 Advanced Computer Graphics Cutler

Monte-Carlo

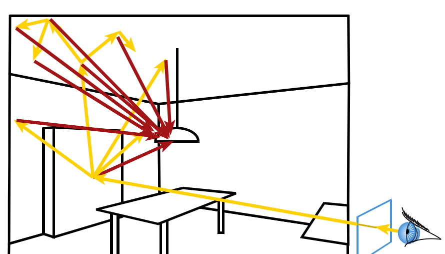
- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
- Recurse



CSCI-6962 Advanced Computer Graphics Cutler

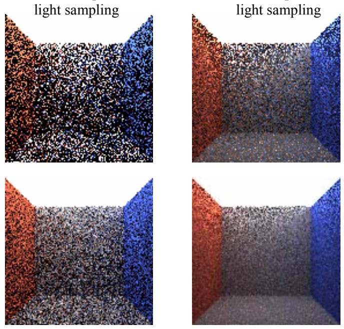
Monte-Carlo

- Systematically sample primary light



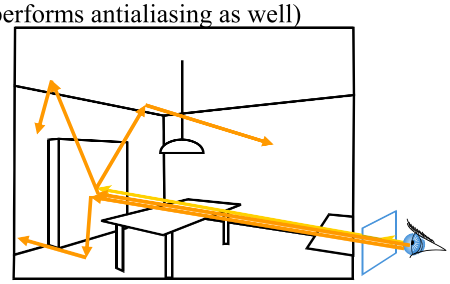
CSCI-6962 Advanced Computer Graphics Cutler

Importance of sampling the light

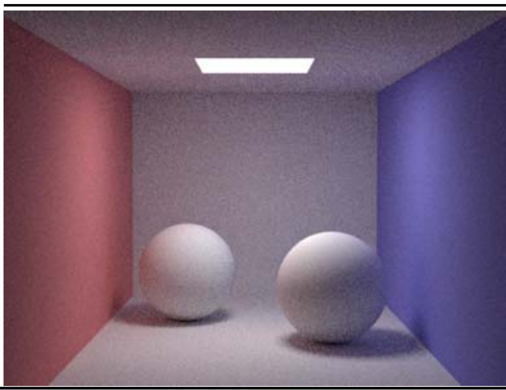


Monte Carlo Path Tracing

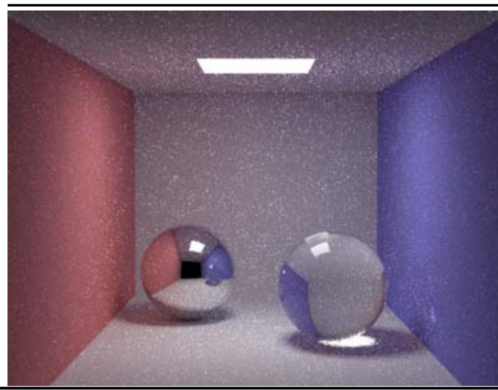
- Trace only one secondary ray per recursion
- But send many primary rays per pixel
- (performs antialiasing as well)



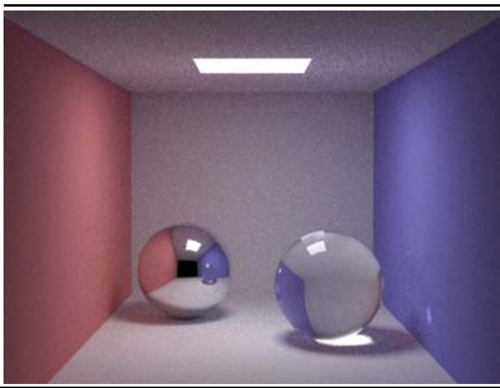
Results: 10 paths/pixel



Results: 10 paths/pixel, glossy



Results: 100 paths/pixel, glossy



Questions?