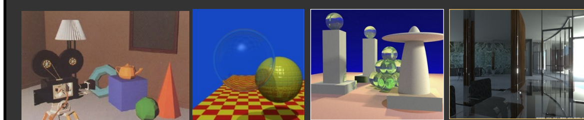
Computer Graphics II: Rendering

CSE 168 [Spr 26], Lecture 12: High Quality Rendering
Ravi Ramamoorthi

http://viscomp.ucsd.edu/classes/cse168/sp26



1

To Do

- Homework 4 (importance sampling) due May 18
- These lectures cover more advanced topics
  - May be relevant for your final project
  - Or curiosity in terms of frontiers of modern rendering

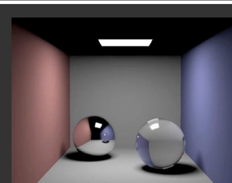
2

Motivation

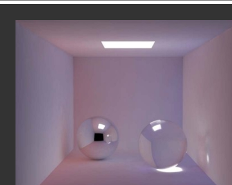
- Rendering Equation since 86, Path Tracer in HW 3
- So, is Monte Carlo rendering solved?
- *Can it be made more efficient (90s until today)?*
  - Multiple Importance Sampling (Homework 4)
  - *Irradiance Caching takes advantage of coherence*
  - Correct sampling: Stratified, Multiple Importance, Bidirectional Path Tracing, Metropolis, VCM/UPS, …
  - Photon Mapping
  - Modern adaptive sampling, cut-based integration
  - Advanced topics (next time)
  - Denoising (next time)

3

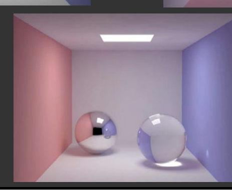
Smoothness of Indirect Lighting



Direct



Indirect



Direct + Indirect

4

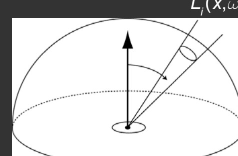
Irradiance Caching

- 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*

5

Irradiance Calculation

$L_i(x,\omega)$



$$E(x) = \int L_i(x,\omega)\cos\theta \, d\omega$$

$$E(x) = \frac{\sum_i w(x_i) E_i(x_i)}{\sum_i w(x_i)} \qquad w(x) = \frac{1}{\varepsilon(x)}$$

$$\epsilon(x) \le \left| \frac{\partial E}{\partial x}(x - x_0) + \frac{\partial E}{\partial \theta}(\theta - \theta_0) \right|$$

position rotation

$$\le E_0 \left(\frac{4}{\pi} \frac{||x - x_0||}{x_{avg}} + \sqrt{2 - 2\vec{N}(x) \cdot \vec{N}(x_0)} \right)$$

Derivation in Ward paper

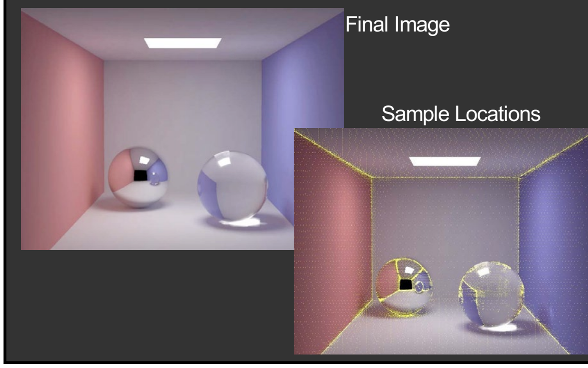
6

Algorithm Outline

- Find all samples with w(x) > q
- if (samples found)
  - interpolate
- else
  - compute new irradiance
- N.B. Subsample the image first and then fill in

7

Irradiance Caching Example



8

Motivation

- Rendering Equation since 86, Path Tracer in HW 3
- So, is Monte Carlo rendering solved?
- *Can it be made more efficient (90s until today)?*
  - Multiple Importance Sampling (Homework 4)
  - Irradiance Caching takes advantage of coherence
  - *Correct sampling: Stratified, Multiple Importance, Bidirectional Path Tracing, Metropolis, VCM/UPS, …*
  - Photon Mapping
  - Modern adaptive sampling, cut-based integration
  - Advanced topics (next time)
  - Denoising (next time)
- High level: refs on slides, ask if need to track down

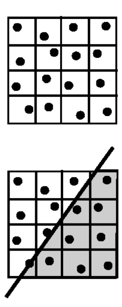
9

Better Sampling

- Smarter ways to Monte Carlo sample
- Long history: Stratified, Importance, Bi-Directional, Multiple Importance, Metropolis
- Good reference is Veach thesis
- We only briefly discuss a couple of strategies

10

Stratified Sampling



Stratified sampling like jittered sampling

Allocate samples per region

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

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

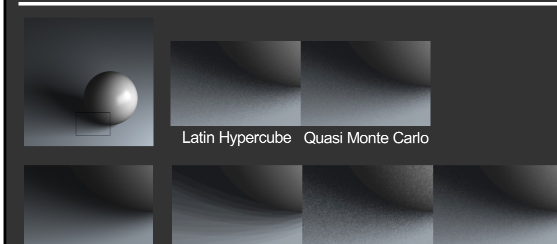
$$V[F_N] = \frac{1}{N^2}\sum_{i=1}^{\sqrt{N}} V[F_j] = \frac{V[F_E]}{N^{1.5}}$$

CS348B Lecture 9 Pat Hanrahan, Spring 2002

D. Mitchell 95, Consequences of stratified sampling in graphics

11

Comparison of simple patterns



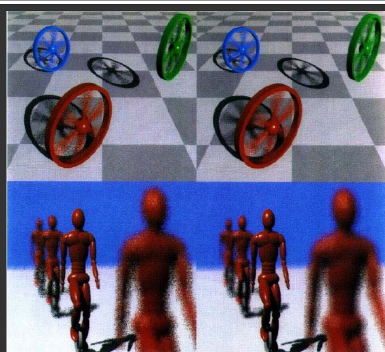
16 samples for area light, 4 samples per pixel, total 64 samples

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

Figures courtesy Tianyu Liu

12

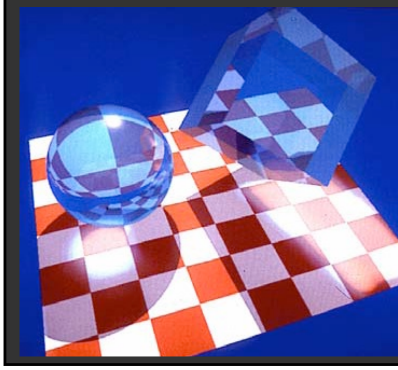
Spectrally Optimal Sampling



Mitchell 91

13

Light Ray Tracing



Backwards Ray Tracing
[Arvo 86]

14

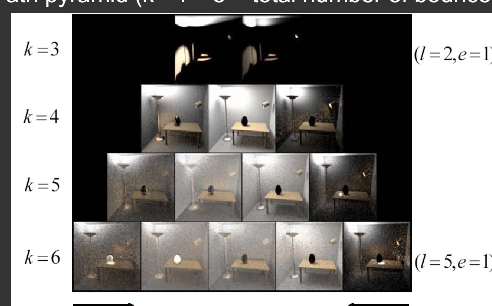
Path Tracing: From Lights

- 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

15

Bidirectional Path Tracing

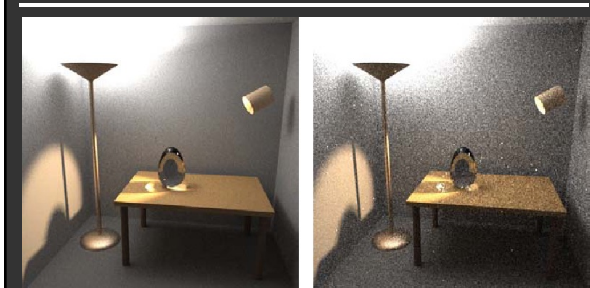
Path pyramid (k = l + e = total number of bounces)



From Veach and Guibas

16

Comparison



Bidirectional path tracing **Path tracing**

From Veach and Guibas

17

Motivation

- Rendering Equation since 86, Path Tracer in HW 3
- So, is Monte Carlo rendering solved?
- *Can it be made more efficient (90s until today)?*
  - Multiple Importance Sampling (Homework 4)
  - Irradiance Caching takes advantage of coherence
  - Correct sampling: Stratified, Multiple Importance, Bidirectional Path Tracing, Metropolis, VCM/UPS, …
  - *Photon Mapping*
  - Modern adaptive sampling, cut-based integration
  - Advanced topics (next time)
  - Denoising (next time)
- High level: refs on slides, ask if need to track down

18

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*)

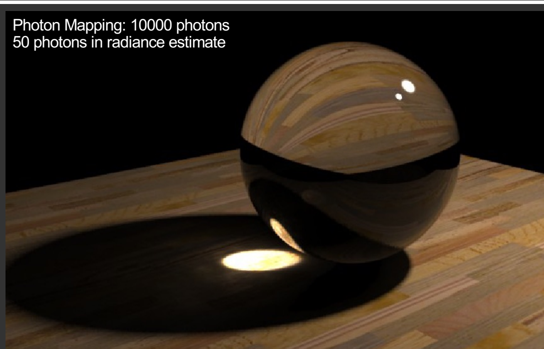
19

Caustics

Slides courtesy Henrik Wann Jensen

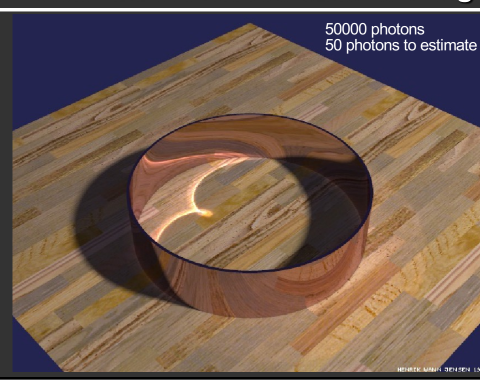
20

Caustics



21

Reflections Inside a Metal Ring



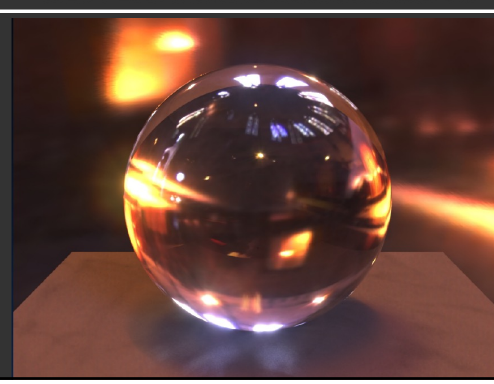
22

Caustics on Glossy Surfaces

340000 photons, 100 photons in radiance estimate

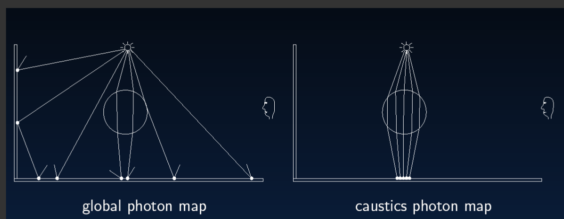
23

HDR Environment Illumination



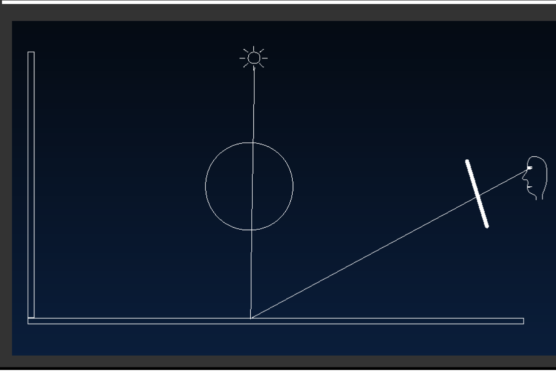
24

Global Illumination



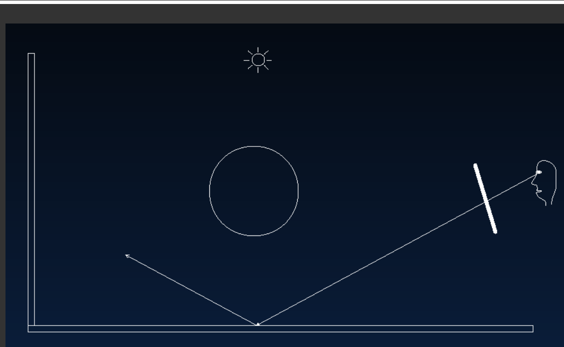
25

Direct Illumination



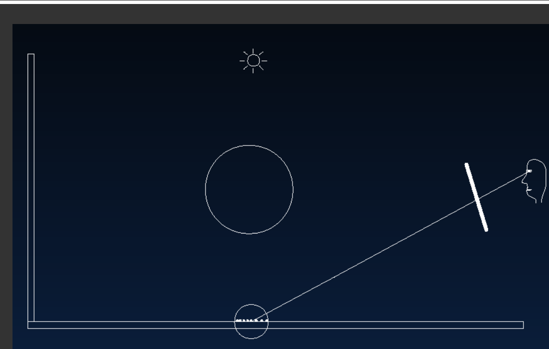
26

Specular Reflection



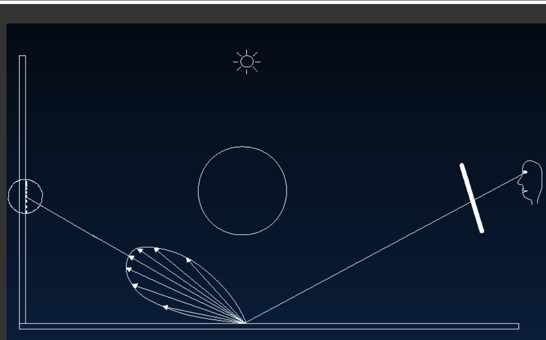
27

Caustics



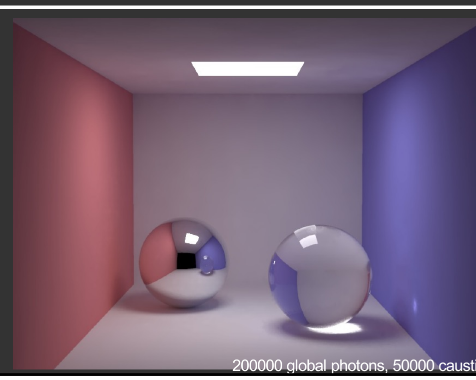
28

Indirect Illumination



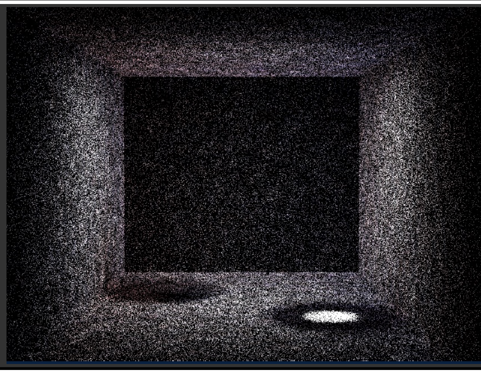
29

Cornell Box



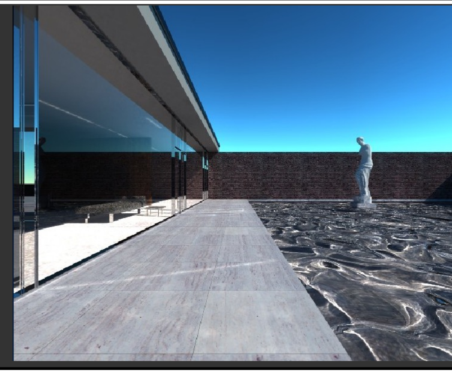
30

Box: Global Photons



31

Mies House: Swimming Pool



32

Motivation

- Rendering Equation since 86, Path Tracer in HW 3
- So, is Monte Carlo rendering solved?
- *Can it be made more efficient (90s until today)?*
  - Multiple Importance Sampling (Homework 4)
  - Irradiance Caching takes advantage of coherence
  - Correct sampling: Stratified, Multiple Importance, Bidirectional Path Tracing, Metropolis, VCM/UPS, …
  - Photon Mapping
  - *Modern adaptive sampling, cut-based integration*
  - Advanced topics (next time)
  - Denoising (next time)
- High level: refs on slides, ask if need to track down

33

Lightcuts

- Efficient, accurate complex illumination

Environment map lighting & indirect
Time 111s

Textured area lights & indirect
Time 98s

(640x480, Anti-aliased, Glossy materials)

From Walter et al. SIGGRAPH 05

34

Complex Lighting

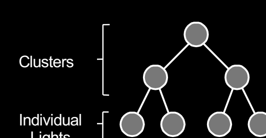
- Simulate complex illumination using point lights
  - Area lights
  - HDR environment maps
  - Sun & sky light
  - Indirect illumination
- Unifies illumination
  - Enables tradeoffs between components

Area lights + Sun/sky + Indirect

35

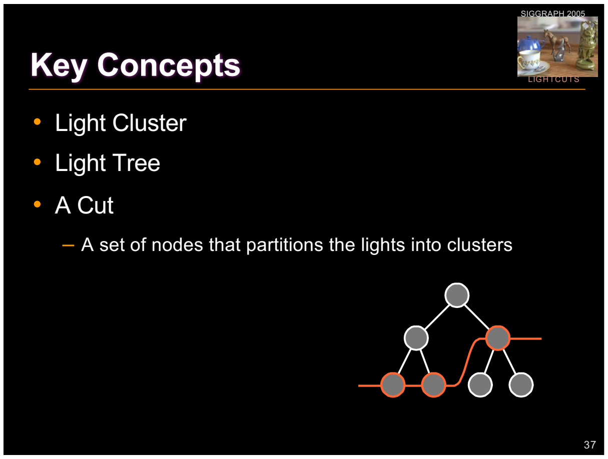
Key Concepts

- Light Cluster
- Light Tree
  - Binary tree of lights and clusters

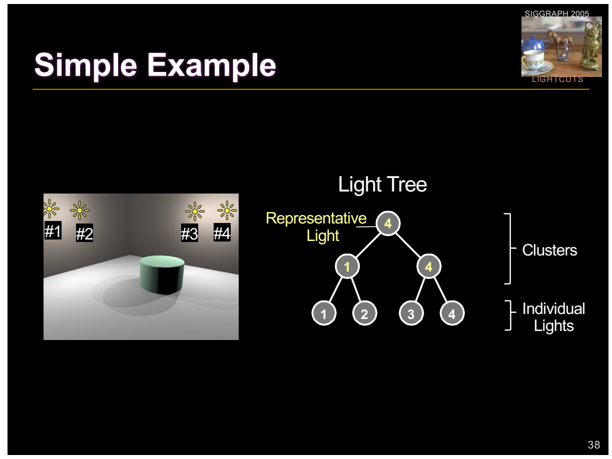


36

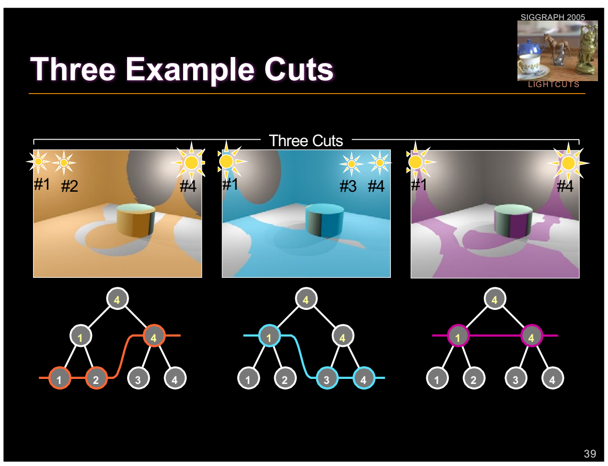
36



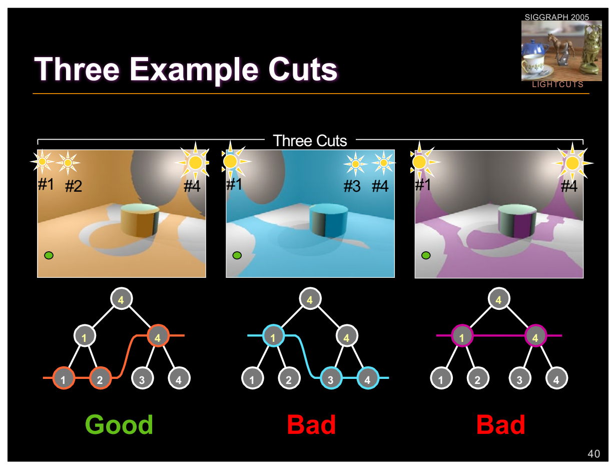
37



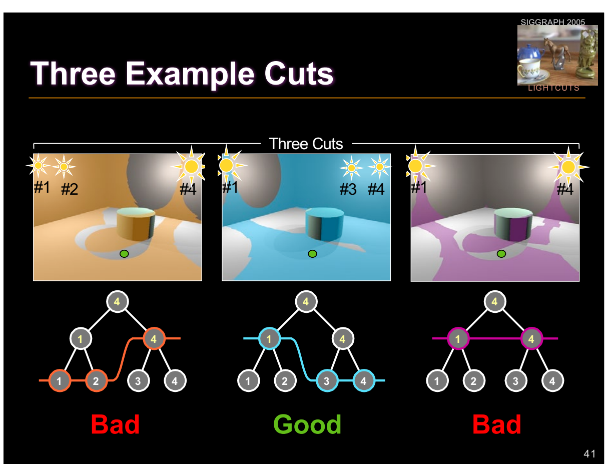
38



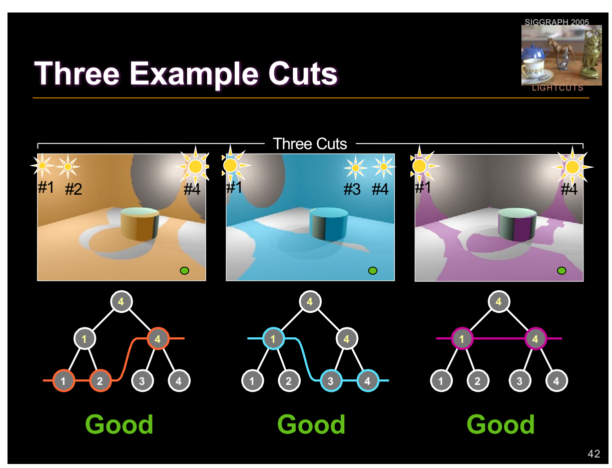
39



40



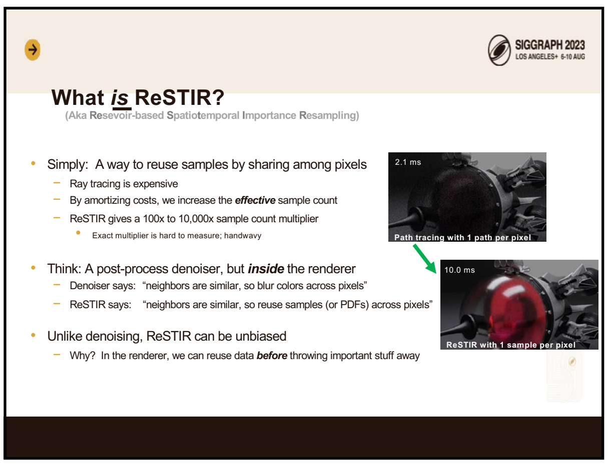
41



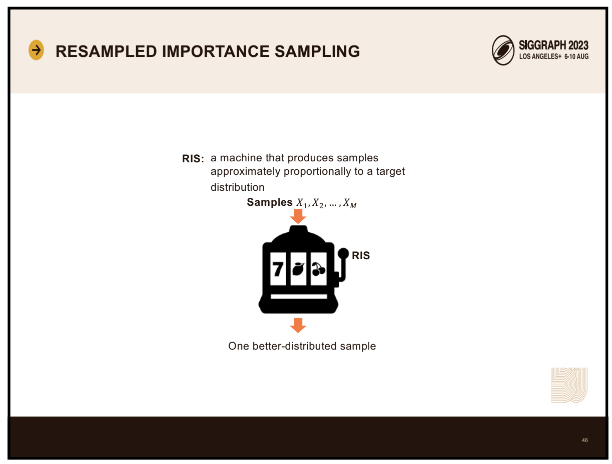
42

43

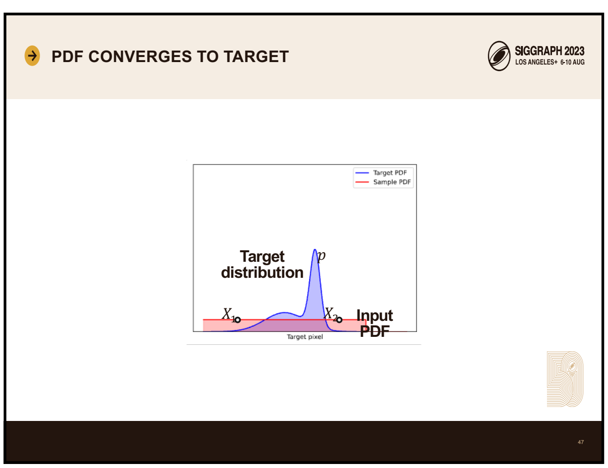
44



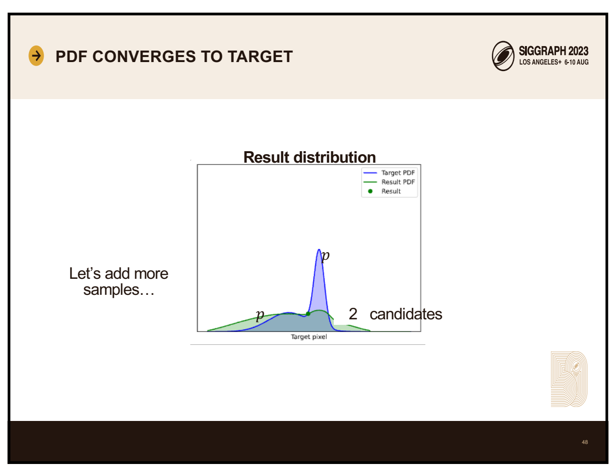
45



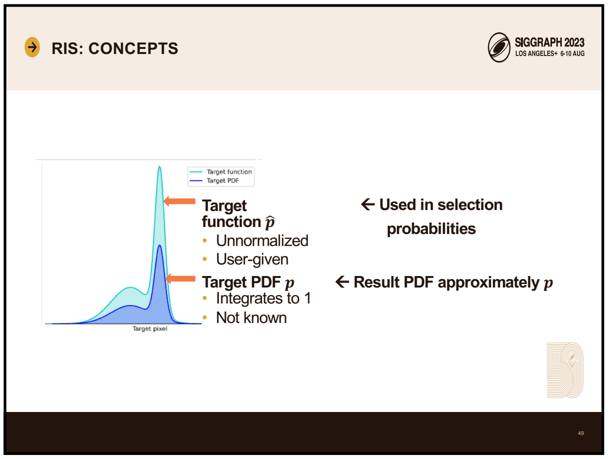
46



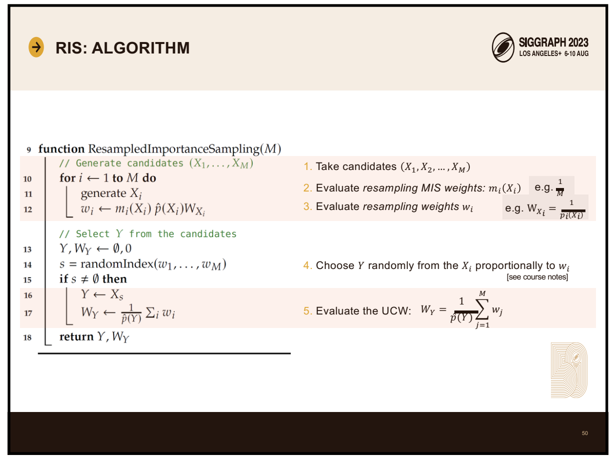
47



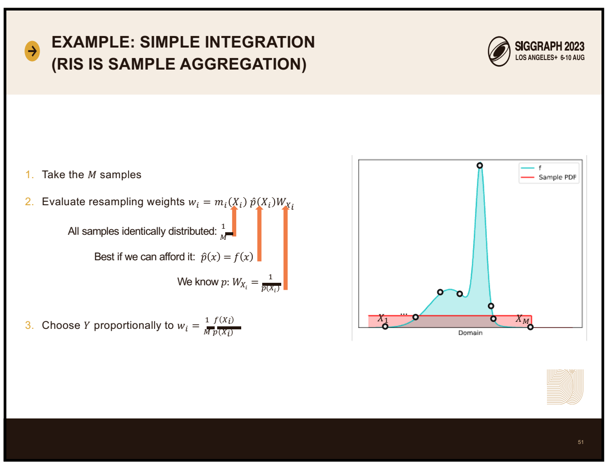
48



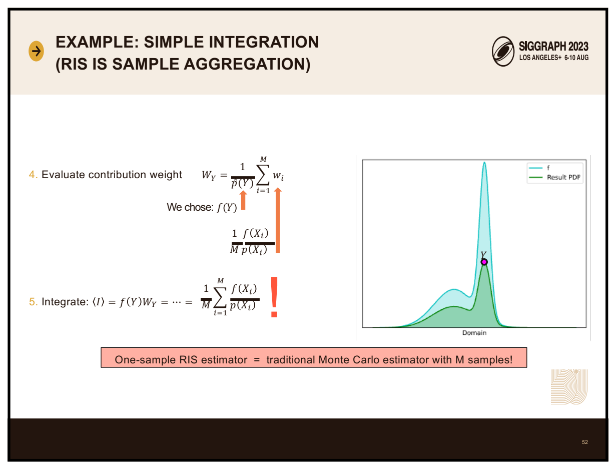
49



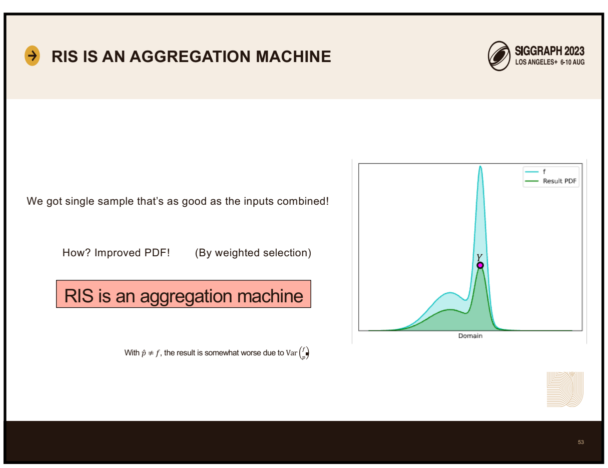
50



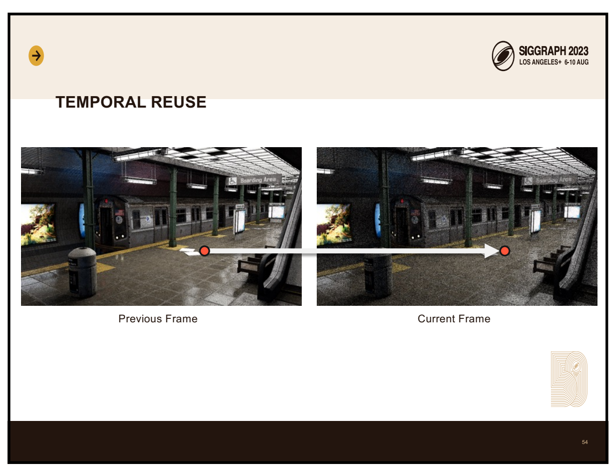
51



52

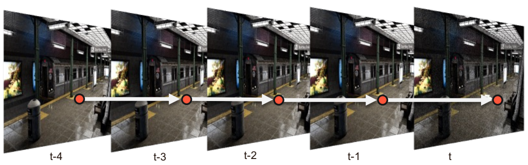


53



54

TEMPORAL REUSE



55

TEMPORAL REUSE

```
void reuseTemporally()
    Reservoir r

    Sample cur = pixelSample[q]
    w = m_time(cur.x) · p̂(cur.x) · cur.W
    r.addSample(cur, w)

    Sample prev = lastFramePixelSample[q]
    w = m_time(prev.x) · p̂(prev.x) · prev.W
    r.addSample(prev, w)

    y = r.sampleOut
    W = 1/p̂(y) · r.w_sum

    pixelSample[q] = Sample {y, W}
```

Render passes:

generateSamples()
reuseTemporally()
reuseSpatially()
reuseSpatially()
...
shadePixel()

56