Choosing the Right Algorithm & Guiding

PHILIPP SLUSALLEK & PASCAL GRITTMANN

Topics for Today

What does an implementation of a high-performance renderer look like?

Review of algorithms – which to choose for production rendering?

Path guiding – how path tracers can handle complex illumination

Brief look at our current research goal: How to adapt VCM to input scenes?

Choosing the Right Rendering Algorithm

REVIEW AND COMPARISON OF THE ALGORITHMS DISCUSSED SO FAR

Path Tracing – Kajiya 1986

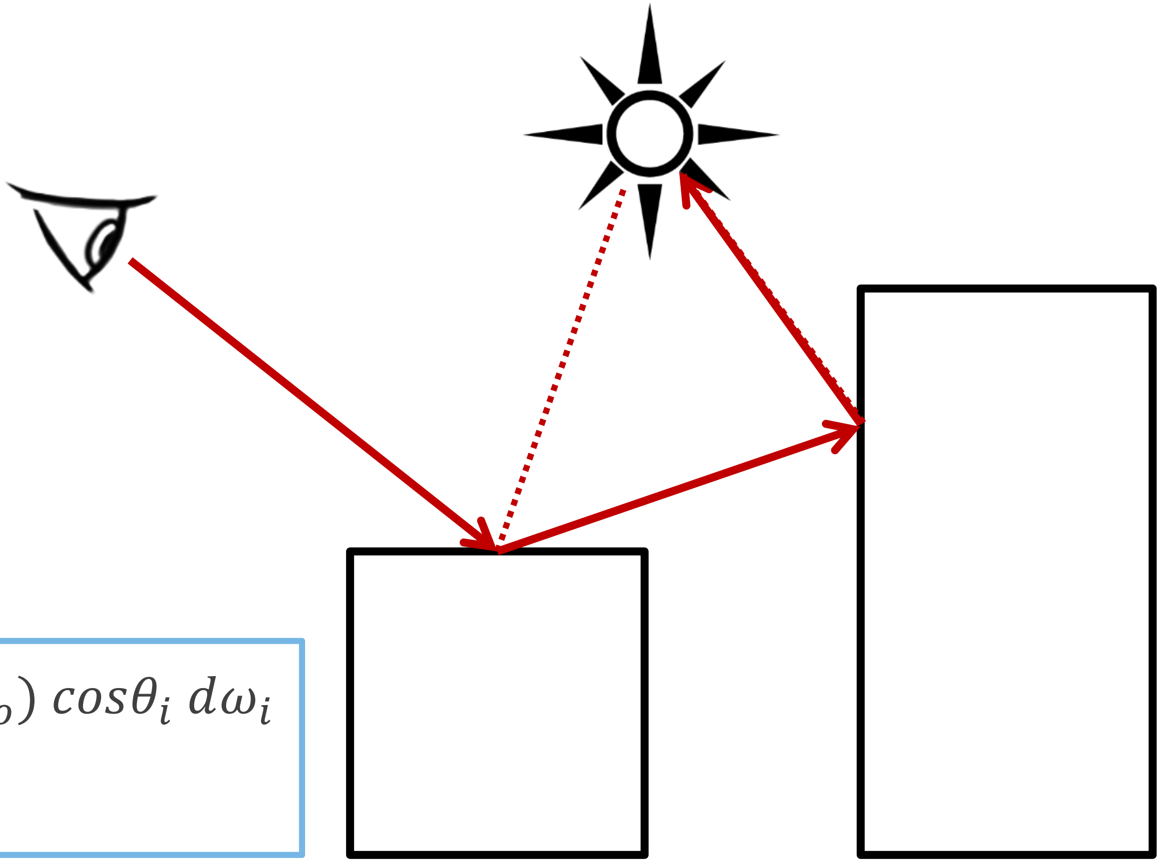
Straightforward Monte Carlo integration of the Rendering Equation

Shoot rays from the eye into the scene

Sample incoming direction at hit points

Shoot secondary ray

Optimization: next event estimation



$$L_o(x,\omega_0) = L_e(x,\omega_0) + \int_{H(x)} L_i(x,\omega_i)\, f_r(\omega_i, x, \omega_o)\, cos\theta_i\, d\omega_i$$

"Light Transport Equation"

Path Tracing – properties

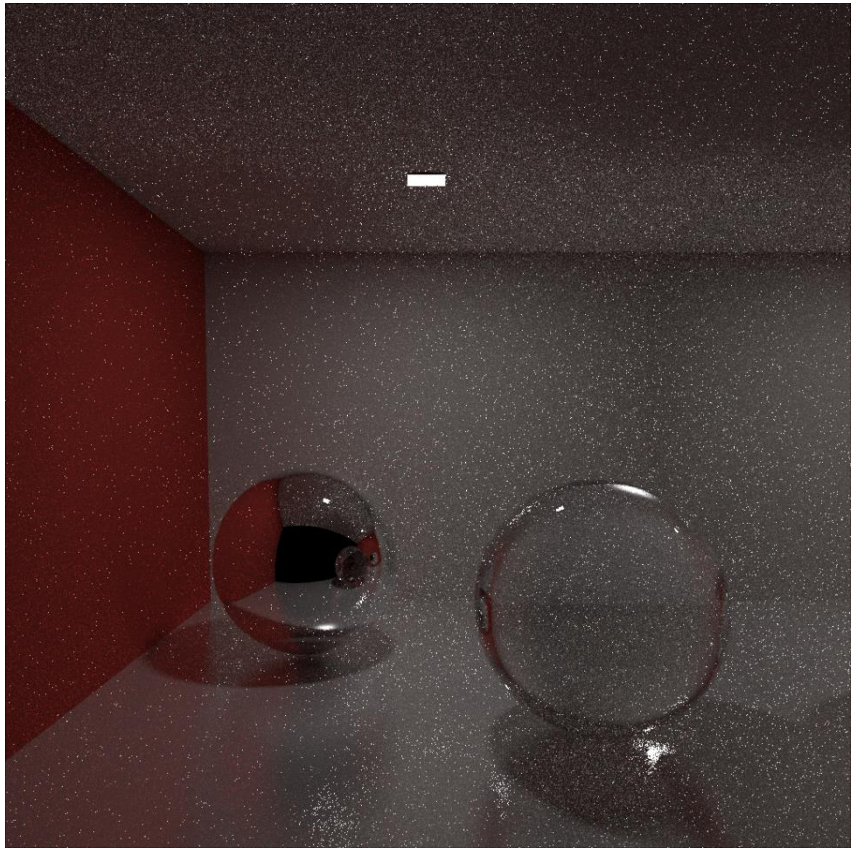
+ Simple to implement

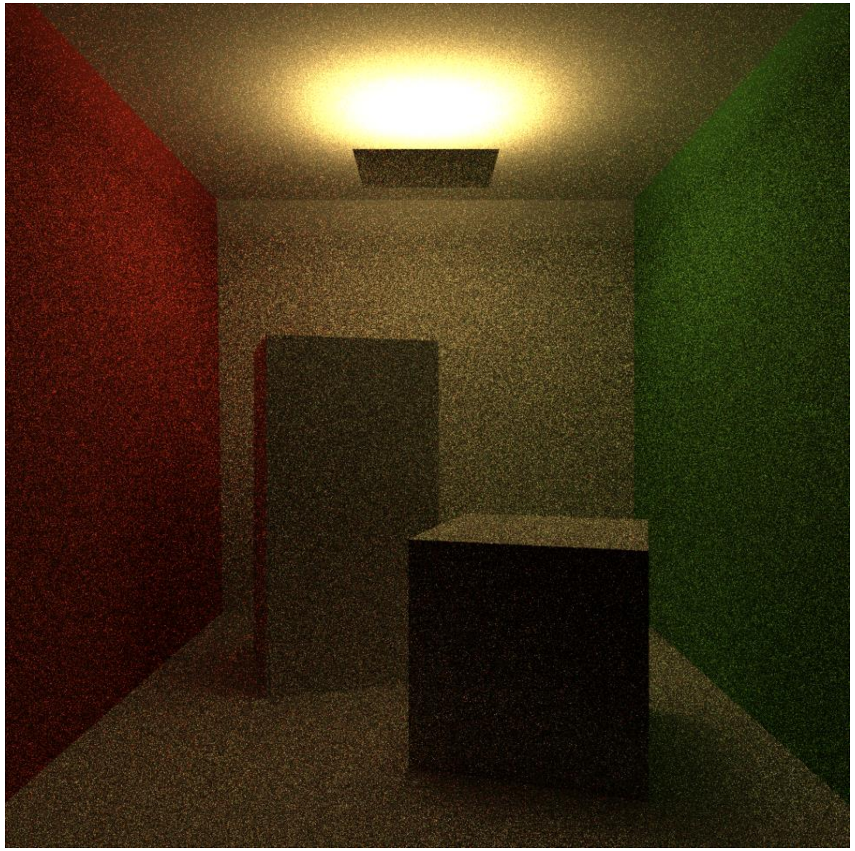
+ Efficient at direct illumination

– Inefficient at caustics

– Inefficient at indirect illumination

Path Tracing – difficult cases





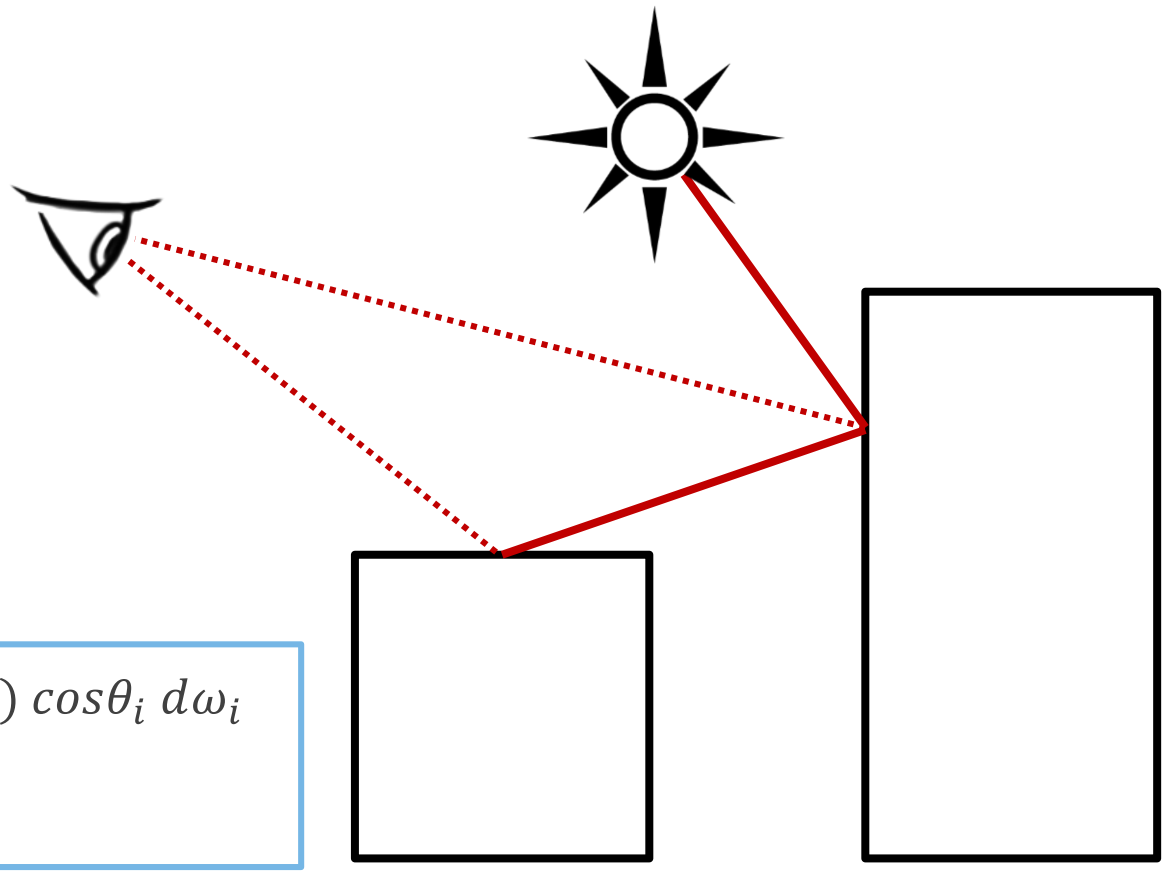
Light Tracing – Dutré et al 1993

Reverse of Path Tracing

Path Tracing: Shoot importance into the scene, gather radiance

Light Tracing: Shoot particles into the scene, gather importance

Next Event Estimation: Only way to get contribution for camera with no actual surface

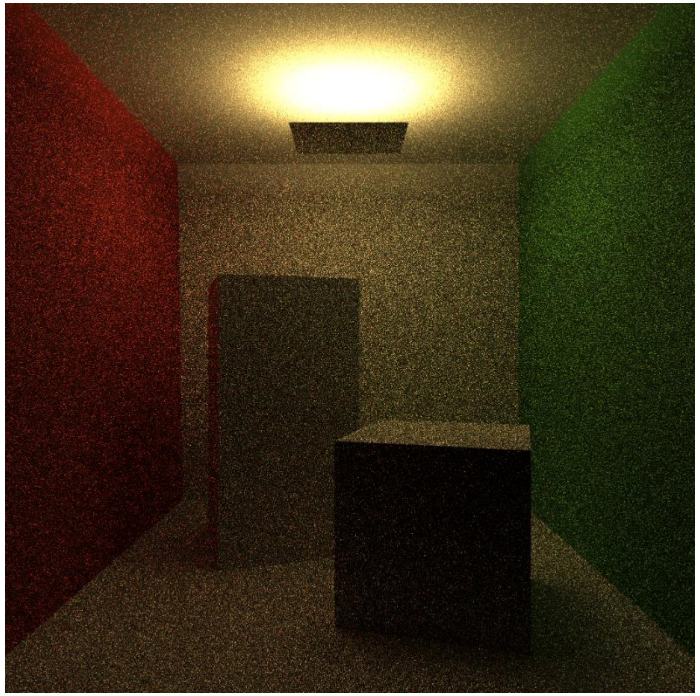


$$W_o(x,\omega_0) = W_e(x,\omega_0) + \int_{H(x)} W_i(x,\omega_i)\, f_r(\omega_i, x, \omega_o)\, cos\theta_i\, d\omega_i$$

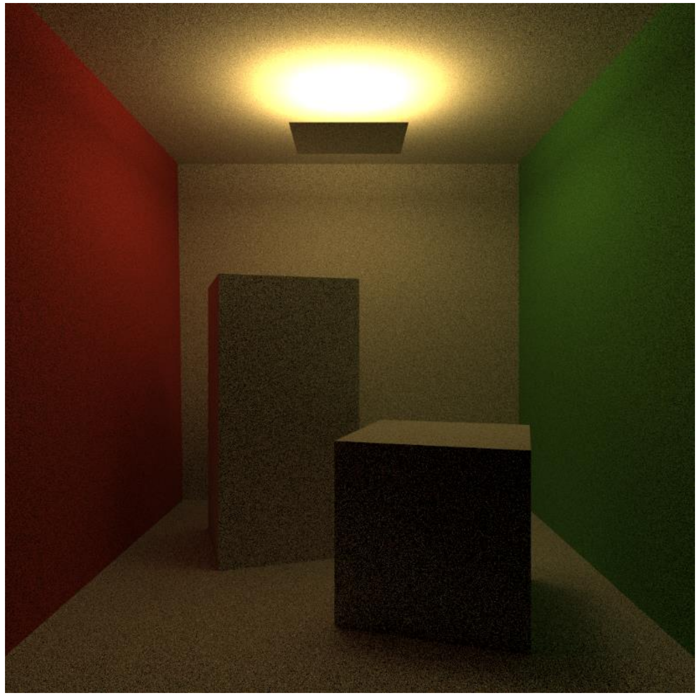
"Importance Transport Equation"

Light Tracing is efficient at rendering indirect illumination

PATH TRACING

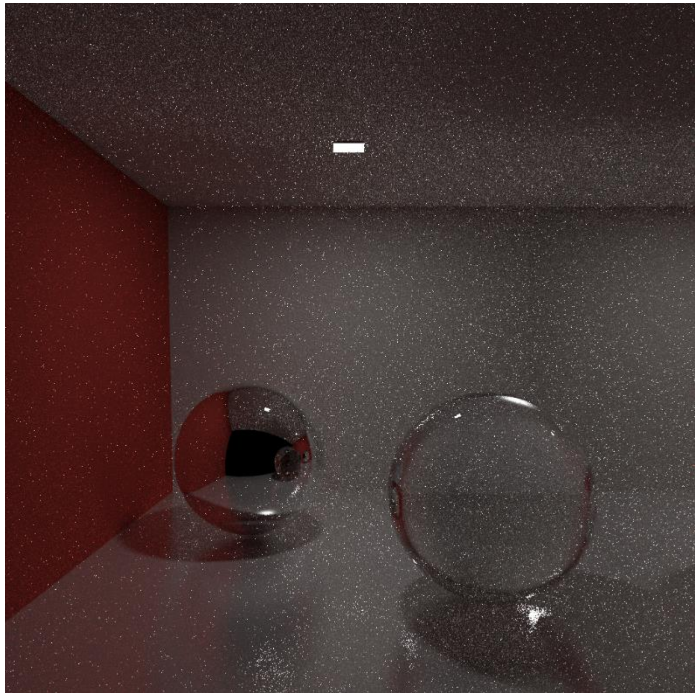


LIGHT TRACING

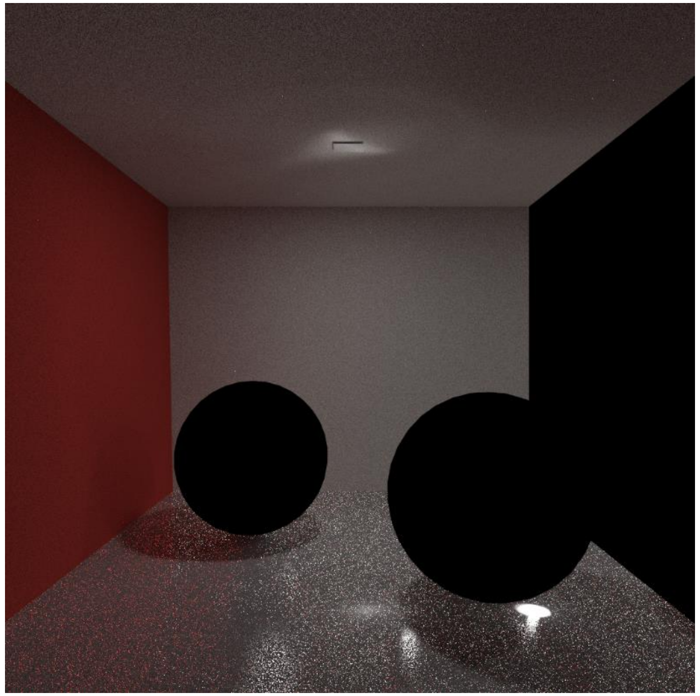


Light Tracing is efficient at rendering caustics

PATH TRACING



LIGHT TRACING



Light Tracing – properties

+ (reasonably) good at indirect illumination

+ efficient at rendering directly visible caustics

- In non-trivial scenes: Wastes **A LOT** of samples in regions of the scene that do not contribute at all! (The solutions to this are either Metropolis or Guiding).

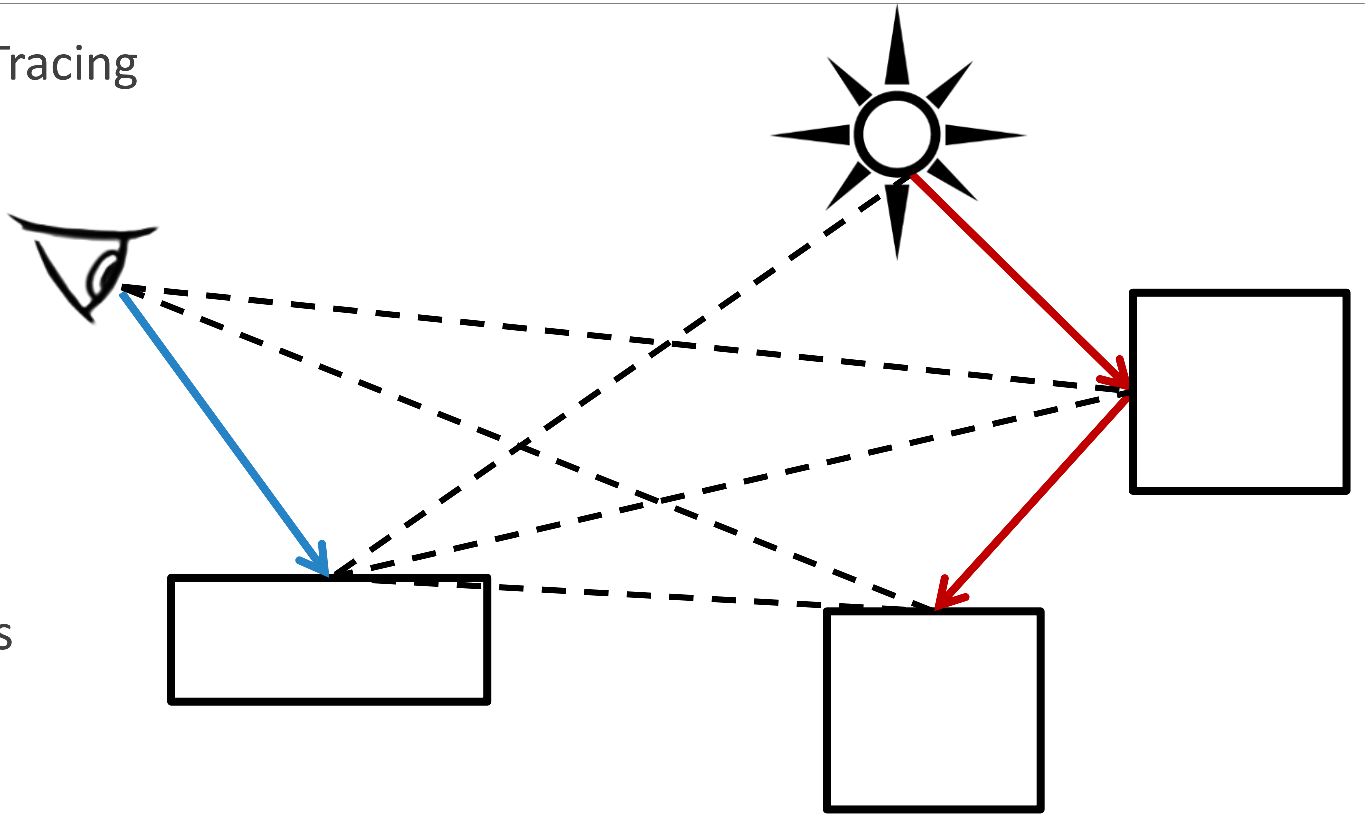
Bidirectional Path Tracing (BPT):
Veach & Guibas ´94, Lafortune & Willems ´93

Combines Light Tracing and Path Tracing

Connect eye and light paths

Combine contributions with MIS:

1. Direct Illumination
2. Randomly hitting light
3. Connecting to the eye
4. Connecting eye and light paths



Multiple Importance Sampling (MIS) – Veach and Guibas 1995

Combine samples from different techniques

Use weighting to select sample with lower variance

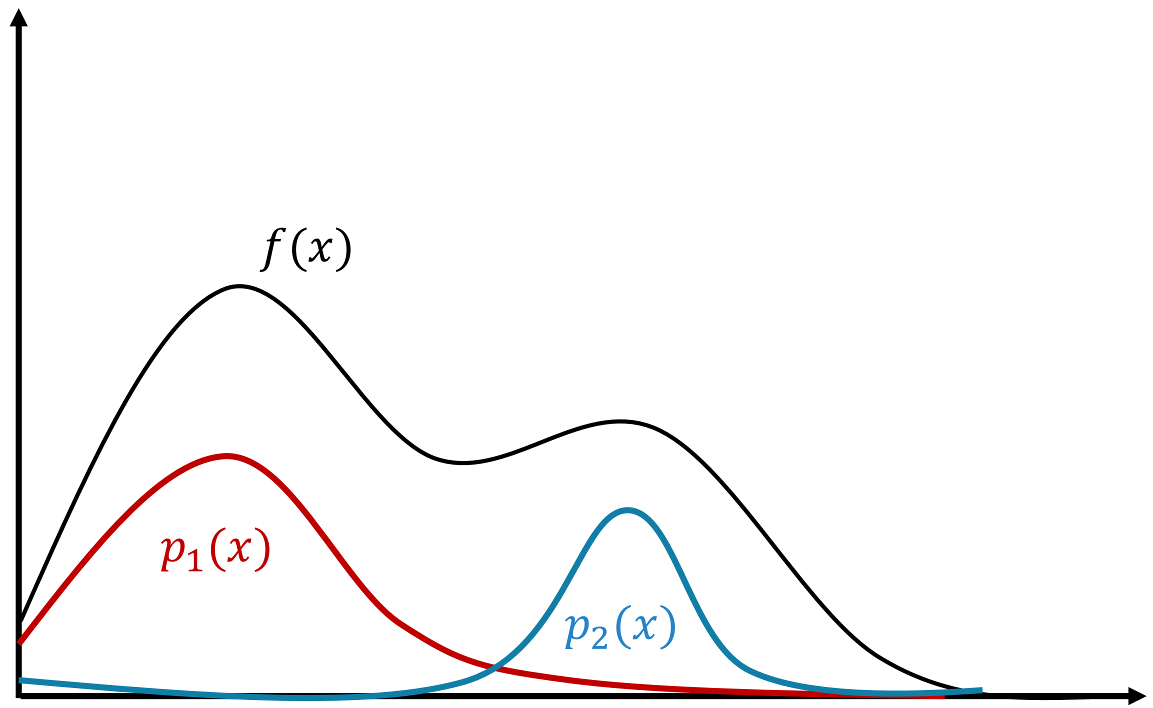
$$\sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i\left(X_{i,j}\right) \frac{f\left(X_{i,j}\right)}{p_i(X_{i,j})}$$

Power heuristic:

$$w_i = \frac{(n_i p_i)^\beta}{\sum_k (n_k p_k)^\beta} = \frac{1}{1+\sum_{k\neq i} \frac{(n_k p_k)^\beta}{(n_i p_i)^\beta}}$$

Balance heuristic:

$$w_i = \frac{1}{1+\sum_{k\neq i} \frac{n_k p_k}{n_i p_i}}$$



Balance Heuristic Weights

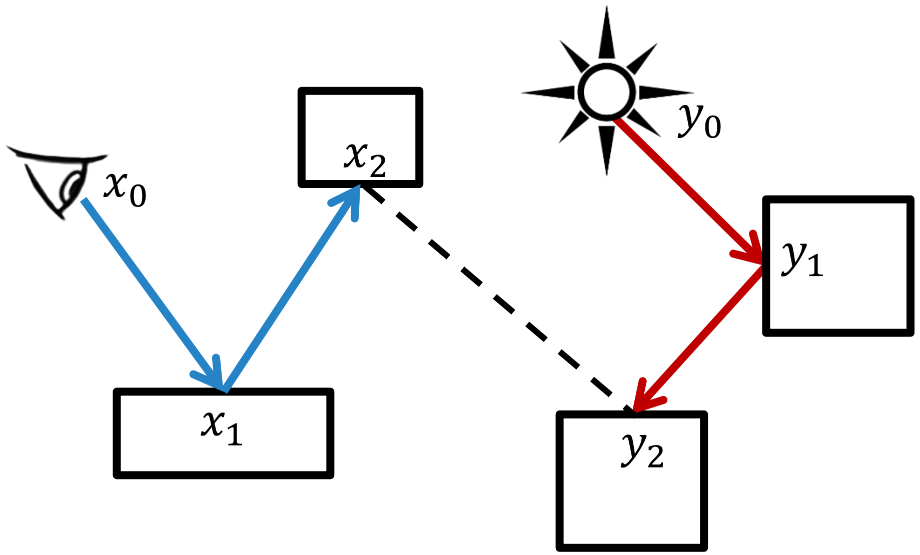
Without loss of generality, we focus on balance heuristic weights.

Assume we sampled a path like the one on the right

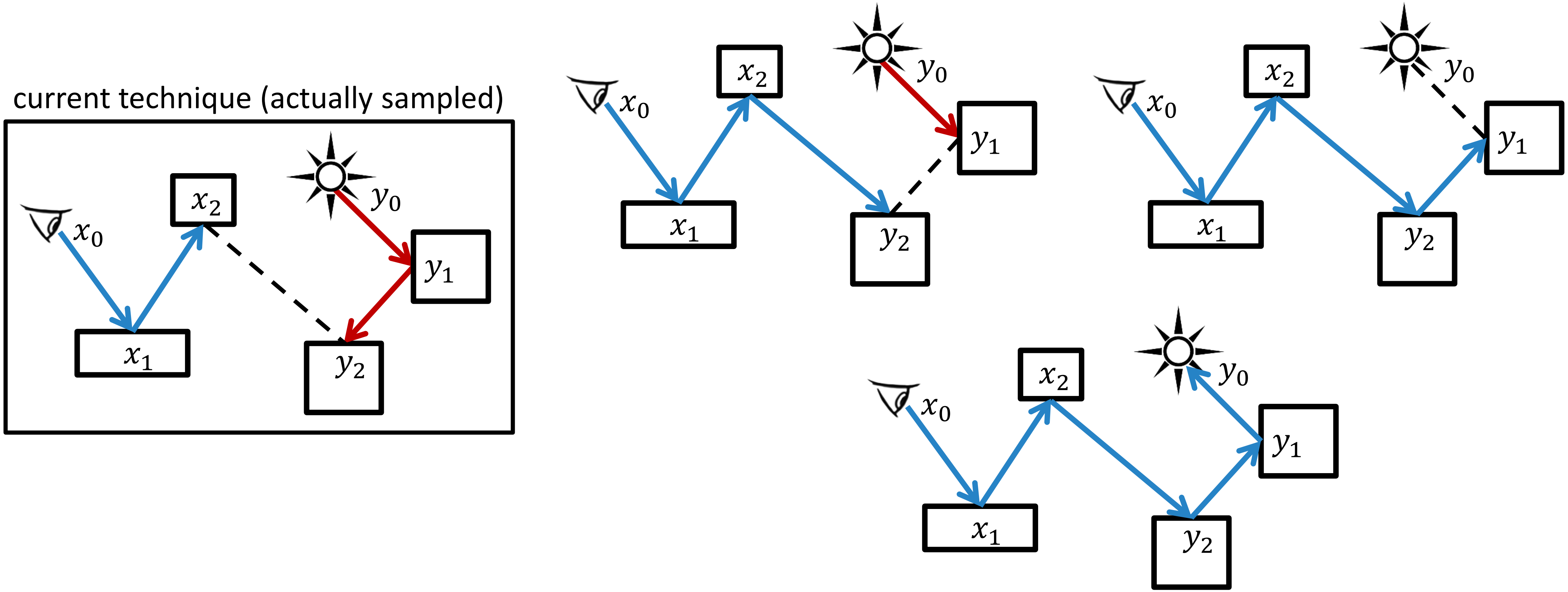
What other techniques can sample this path?

What are the pdfs for these techniques?

What is the weight of our current sample?



Techniques on the light sub-path



Light sub-path PDFs – current technique

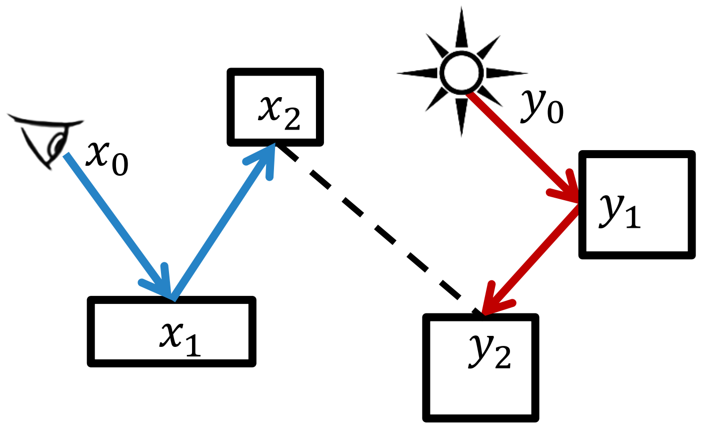
For simplicity, and without loss of generality, we ignore the number of samples from each technique

The PDF of the current technique is the product of:

$p_{cam}(x_0 \to x_1)\, p_{brdf}(x_0 \to x_1 \to x_2)$

$p_{light}(y_0 \to y_1)\, p_{brdf}(y_0 \to y_1 \to y_2)$

$p_{conn}(x_2, y_2)$

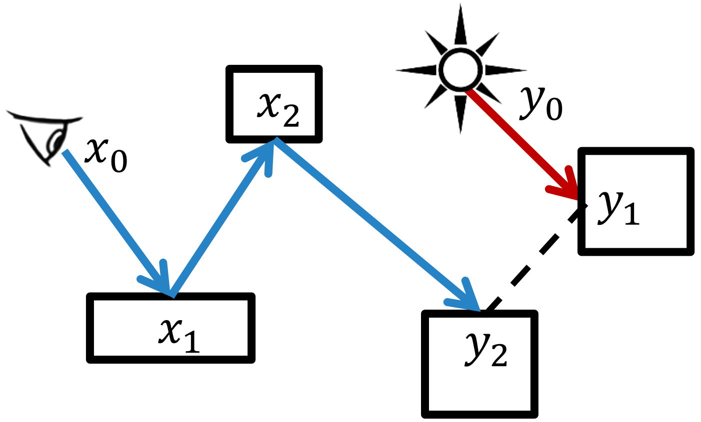


Light sub-path PDFs – connection instead of the last bounce

$p_{cam}(x_0 \to x_1)\, p_{brdf}(x_0 \to x_1 \to x_2)\, p_{brdf}(x_1 \to x_2 \to y_2)$

$p_{light}(y_0 \to y_1)$

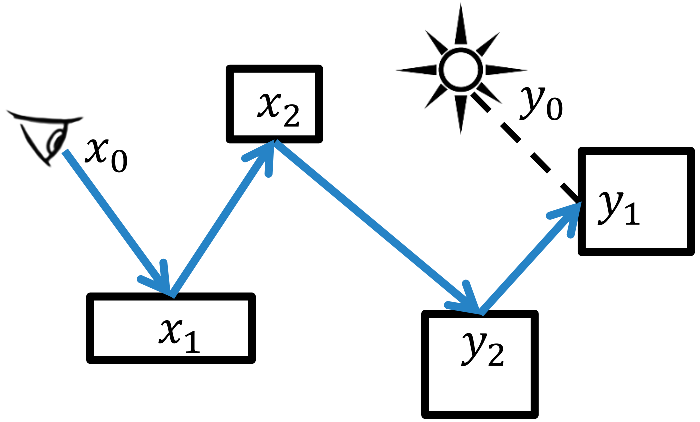
$p_{conn}(y_2, y_1)$



Light sub-path PDFs – Next Event Estimation

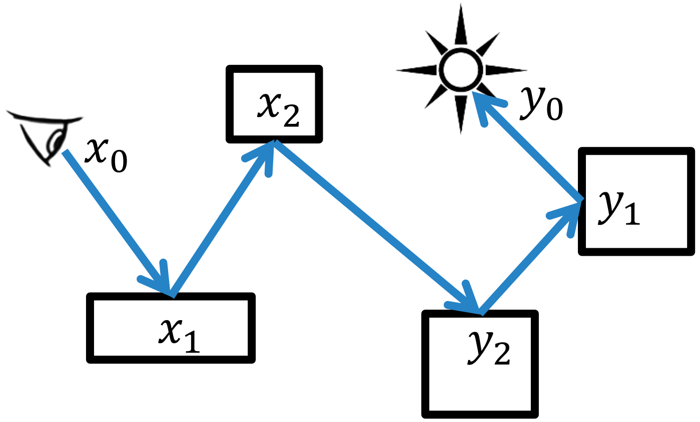
$p_{cam}(x_0 \to x_1)\, p_{brdf}(x_0 \to x_1 \to x_2)\, p_{brdf}(x_1 \to x_2 \to y_2)\, p_{brdf}(x_2 \to y_2 \to y_1)$

$p_{conn}(y_0, y_1)$



Light sub-path PDFs – Unidirectional Path Tracing

$p_{cam}(x_0 \to x_1)\, p_{brdf}(x_0 \to x_1 \to x_2)\, p_{brdf}(x_1 \to x_2 \to y_2)\, p_{brdf}(x_2 \to y_2 \to y_1)\, p_{brdf}(y_2 \to y_1 \to y_0)$



Combined MIS Denominator

$$1$$

$$+ \frac{p_{cam}(x_0 \to x_1)\, \cancel{p_{brdf}(x_0 \to x_1 \to x_2)}\, p_{brdf}(x_1 \to x_2 \to y_2)\, p_{light}(y_0 \to y_1)\, p_{conn}(y_2, y_1)}{\cancel{p_{cam}(x_0 \to x_1)}\, p_{brdf}(x_0 \to x_1 \to x_2)\, p_{light}(y_0 \to y_1)\, p_{brdf}(y_0 \to y_1 \to y_2)\, p_{conn}(x_2, y_2)}$$

$$+ \frac{p_{cam}(x_0 \to x_1)\, \cancel{p_{brdf}(x_0 \to x_1 \to x_2)}\, p_{brdf}(x_1 \to x_2 \to y_2)\, p_{brdf}(x_2 \to y_2 \to y_1)\, p_{conn}(y_0, y_1)}{\cancel{p_{cam}(x_0 \to x_1)}\, p_{brdf}(x_0 \to x_1 \to x_2)\, p_{light}(y_0 \to y_1)\, p_{brdf}(y_0 \to y_1 \to y_2)\, p_{conn}(x_2, y_2)}$$

$$+ \frac{p_{cam}(x_0 \to x_1)\, \cancel{p_{brdf}(x_0 \to x_1 \to x_2)}\, p_{brdf}(x_1 \to x_2 \to y_2)\, p_{brdf}(x_2 \to y_2 \to y_1)\, p_{brdf}(y_2 \to y_1 \to y_0)}{\cancel{p_{cam}(x_0 \to x_1)}\, p_{brdf}(x_0 \to x_1 \to x_2)\, p_{light}(y_0 \to y_1)\, p_{brdf}(y_0 \to y_1 \to y_2)\, p_{conn}(x_2, y_2)}$$

$+$ weight of techniques happening on the camera sub-path

Nothing happening on the camera sub-path matters for the techniques on the light sub-path!

(Except for the last vertex)

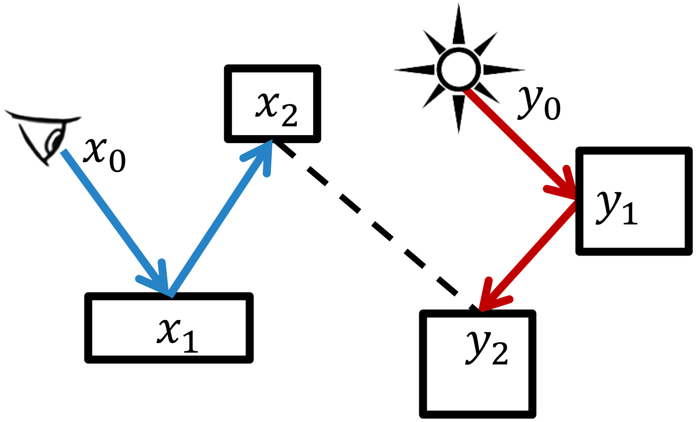
Combined MIS Denominator

$1 +$ camera techniques $+$

$$+\frac{p_{brdf}(x_1 \to x_2 \to y_2)}{p_{brdf}(y_0 \to y_1 \to y_2)}$$ connection instead last bounce

$$+\frac{p_{brdf}(x_1 \to x_2 \to y_2)\, p_{brdf}(x_2 \to y_2 \to y_1)}{p_{light}(y_0 \to y_1)\, p_{brdf}(y_0 \to y_1 \to y_2)}$$ next event

$$+\frac{p_{brdf}(x_1 \to x_2 \to y_2)\, p_{brdf}(x_2 \to y_2 \to y_1)\, p_{brdf}(y_2 \to y_1 \to y_0)}{p_{light}(y_0 \to y_1)\, p_{brdf}(y_0 \to y_1 \to y_2)}$$ hitting the light



p_{conn} is (usually) one

Incremental Update Scheme

Weights for techniques in one sub-path (here the light)

Stored in two float values: current partial sum, inverse pdf of last bounce

Initially:

$$p_{last} = \frac{1}{p_{light}} \qquad \mathrm{w}_{partial} = 0$$

Whenever sampling a ray from y_j to y_{j+1}:

$$w_{partial} = w_{partial} \frac{p_{brdf}(y_j \rightarrow y_{j-1})}{p_{brdf}(y_j \rightarrow y_{j+1})}$$

updates previous techniques

$$+ \frac{p_{last}}{p_{brdf}(y_j \rightarrow y_{j+1})}$$

connection instead of this bounce

$$p_{last} = \frac{1}{p_{brdf}(y_j \rightarrow y_{j+1})}$$

Pseudocode

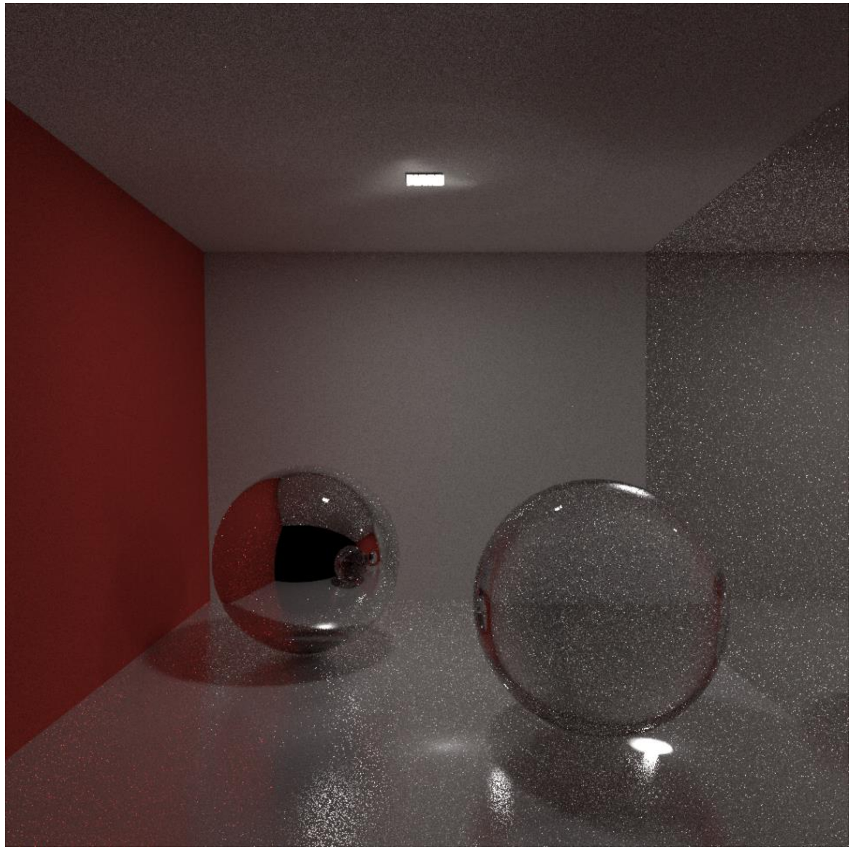
All pdfs have to be w.r.t area measure!

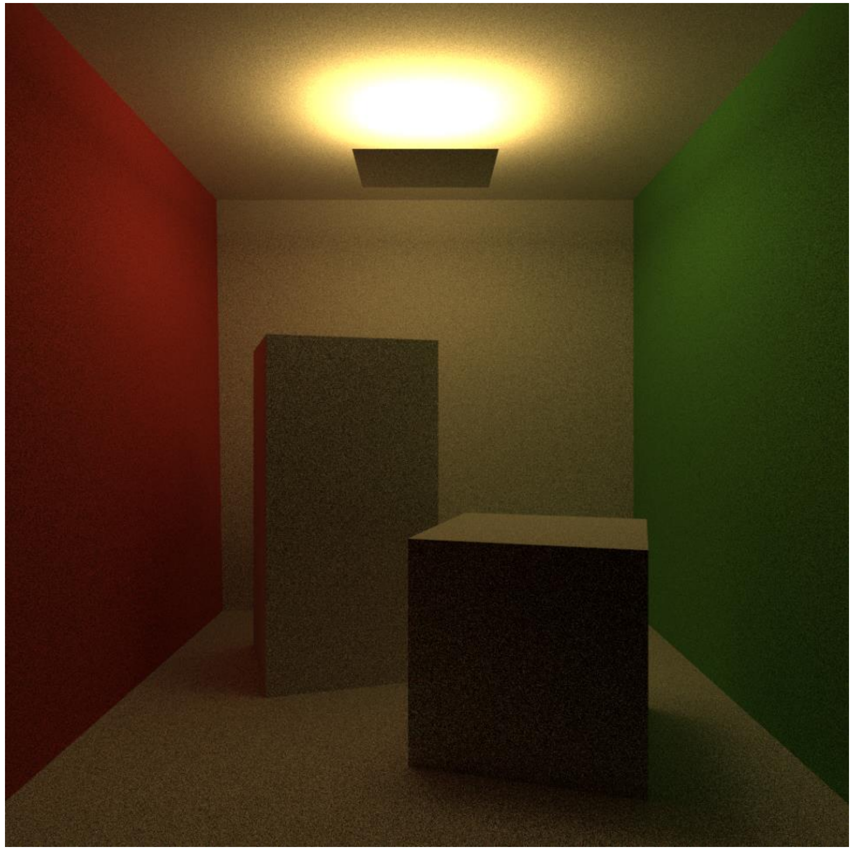
```
// At every bounce:
partial *= bsdf_pdf_reverse;     // For prev. connections: sampling in other direction
partial += 1.0f / last_bsdf_pdf  // weight for connecting instead of the last bounce
partial *= cos_out / bsdf_pdf;   // pdf to continue in current direction
                                 // (solid angle -> area)

// At every hit: finish converting solid angle -> area
last_bsdf_pdf *= cos_in / dist_sqr;
partial /= cos_in
```

To compute the weight: add partials from each sub-path, modify to account for current technique

There are some special cases (Next Event Estimation, hitting light source,…)

BPT combines the benefits from Path Tracing and Light Tracing

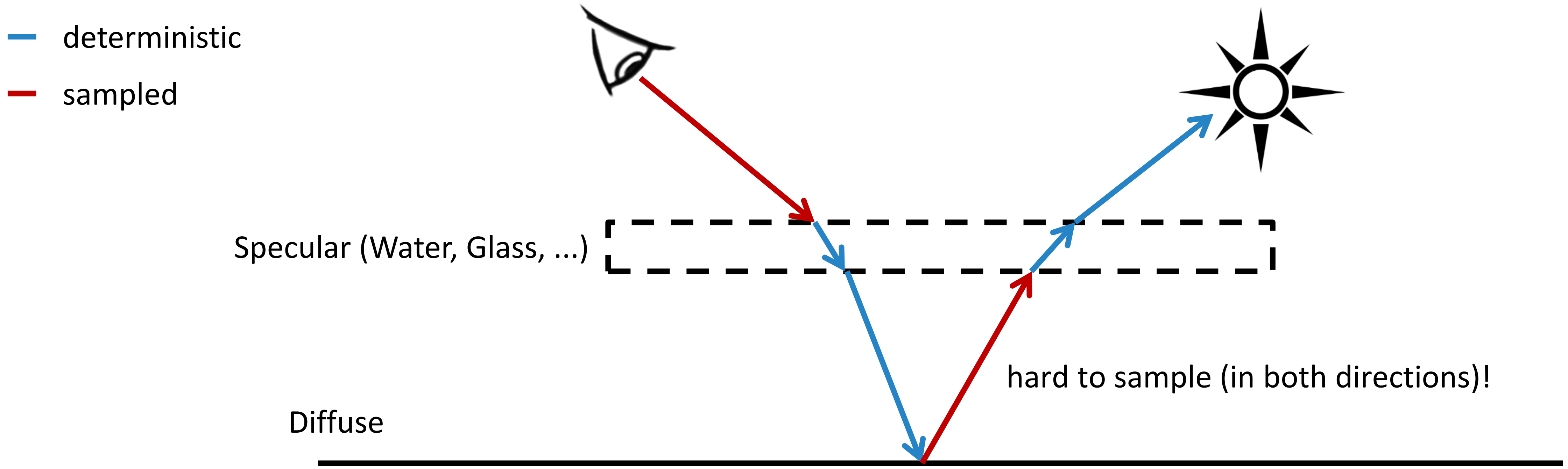




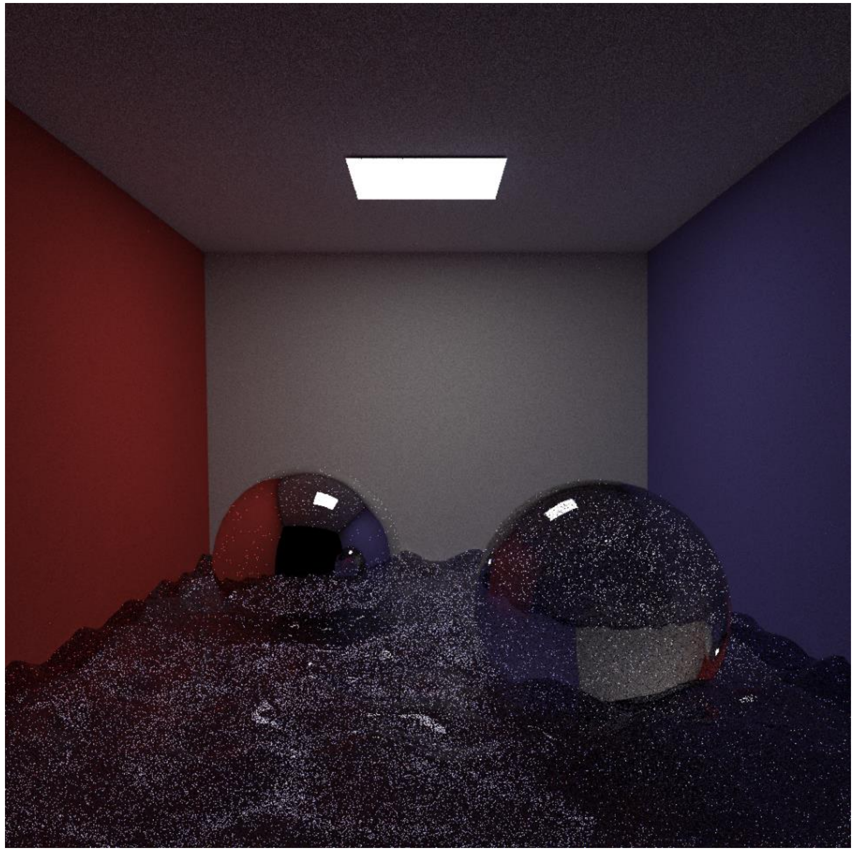
Difficult specular-diffuse-specular (SDS) paths

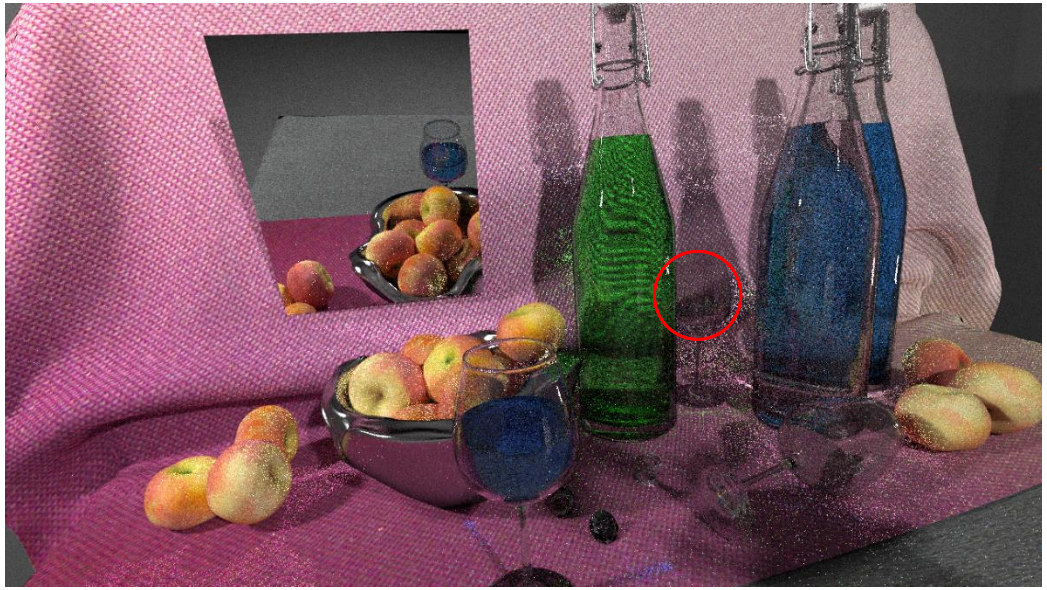
Cannot be captured by light tracing with pinhole camera

Path tracing can handle them (poorly) if only area lights are used



Difficult specular-diffuse-specular (SDS) paths





BPT – properties

+ Better per-sample convergence rate than either PT or LT on its own

- Also suffers from the issue of light paths in regions that are not relevant

- Only technique for SDS paths: PT

- More time for one pixel sample than only PT

Performs worse than PT for SDS paths!

Photon Mapping

Trace particle path from the light

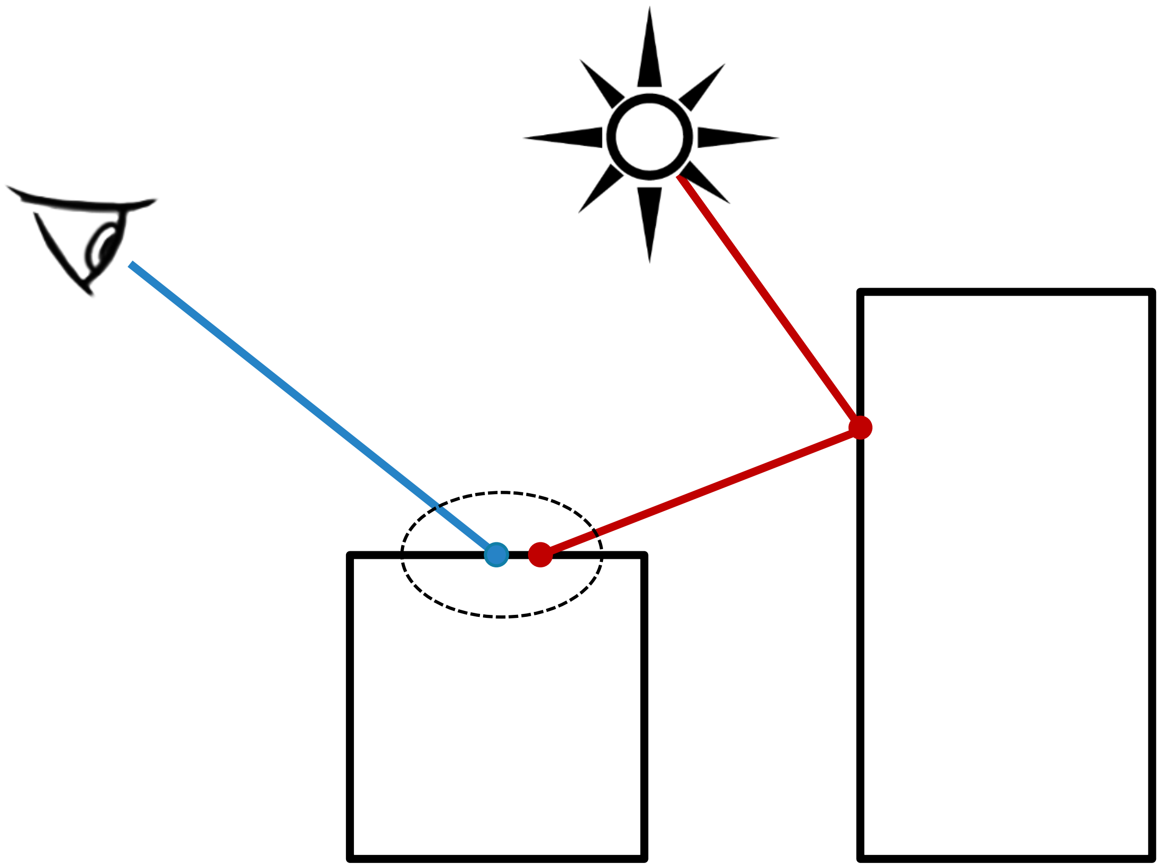
Store vertices

Density Estimation Find vertices within a range, merge them (biased)

Especially good at handling SDS paths

Progressive Photon Mapping [Hachisuka et al 2008] reduces radius with every iteration

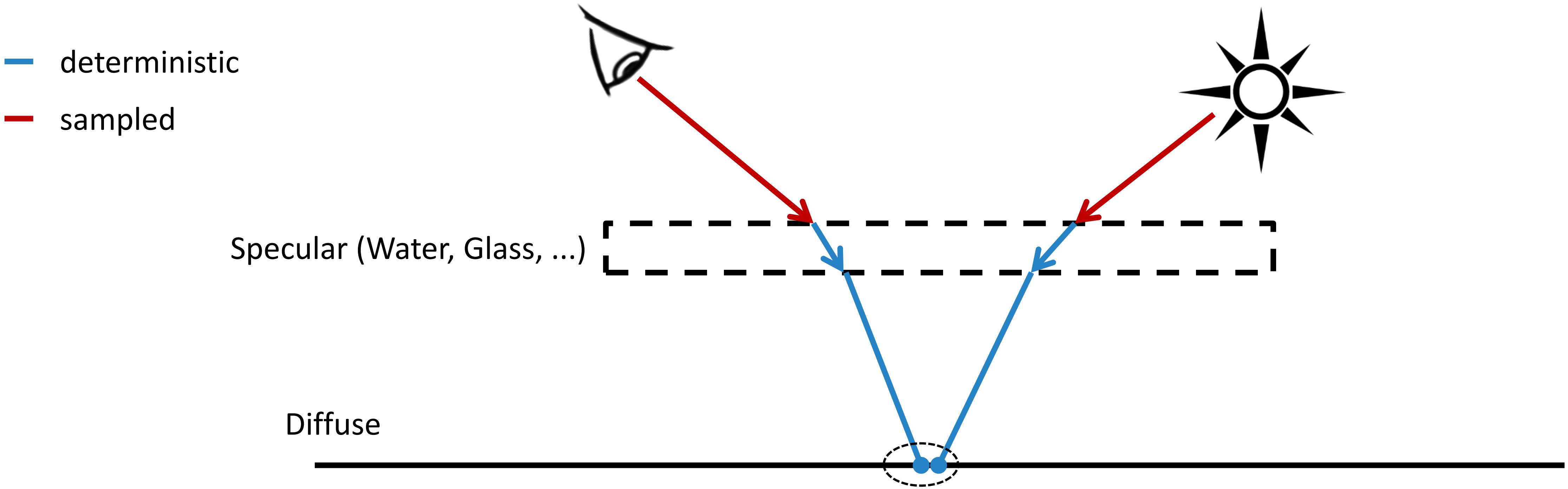
➔ makes algorithm consistent



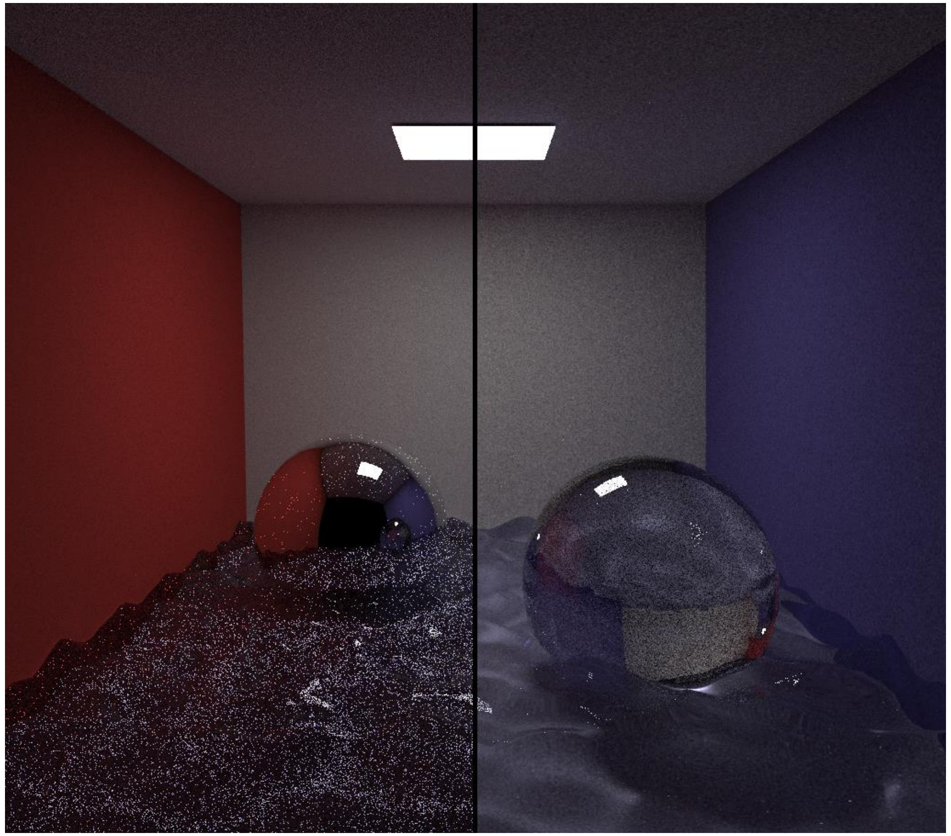
Captures SDS paths – through efficient path reuse!

Prob. that photon falls into circle ≈ prob. to hit the light with path tracing

But: every photon can be used by a lot of camera paths!

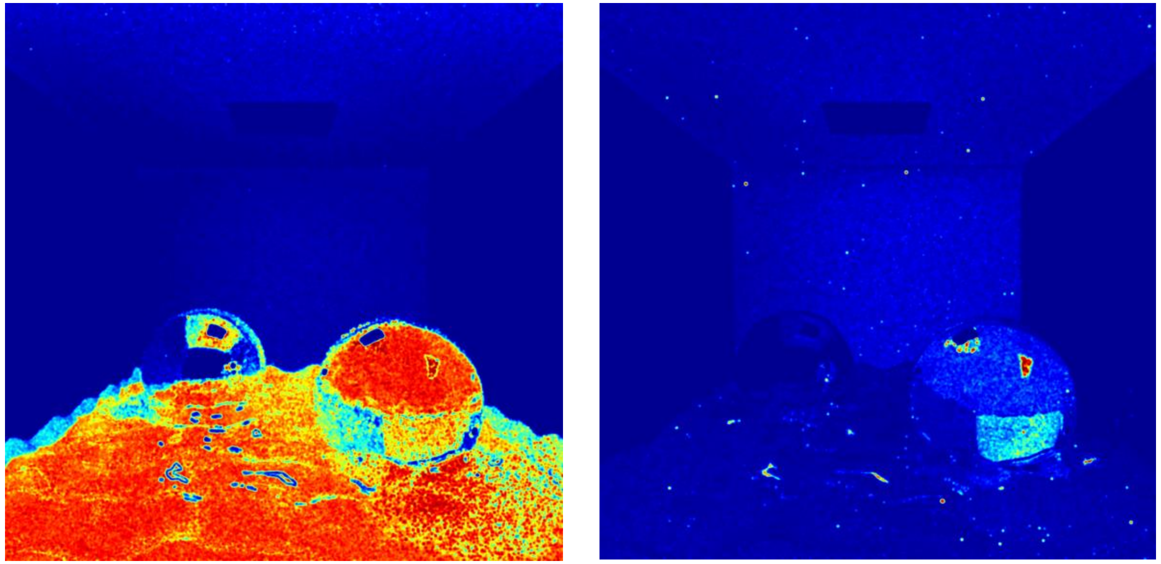


Progressive PM captures SDS paths, but high variance on diffuse surfaces



BPT PPM

Structural Similarity (SSIM)



BPT PPM

Photon Mapping – properties

+ Very efficient for SDS paths

- Performs poorly on glossy surfaces

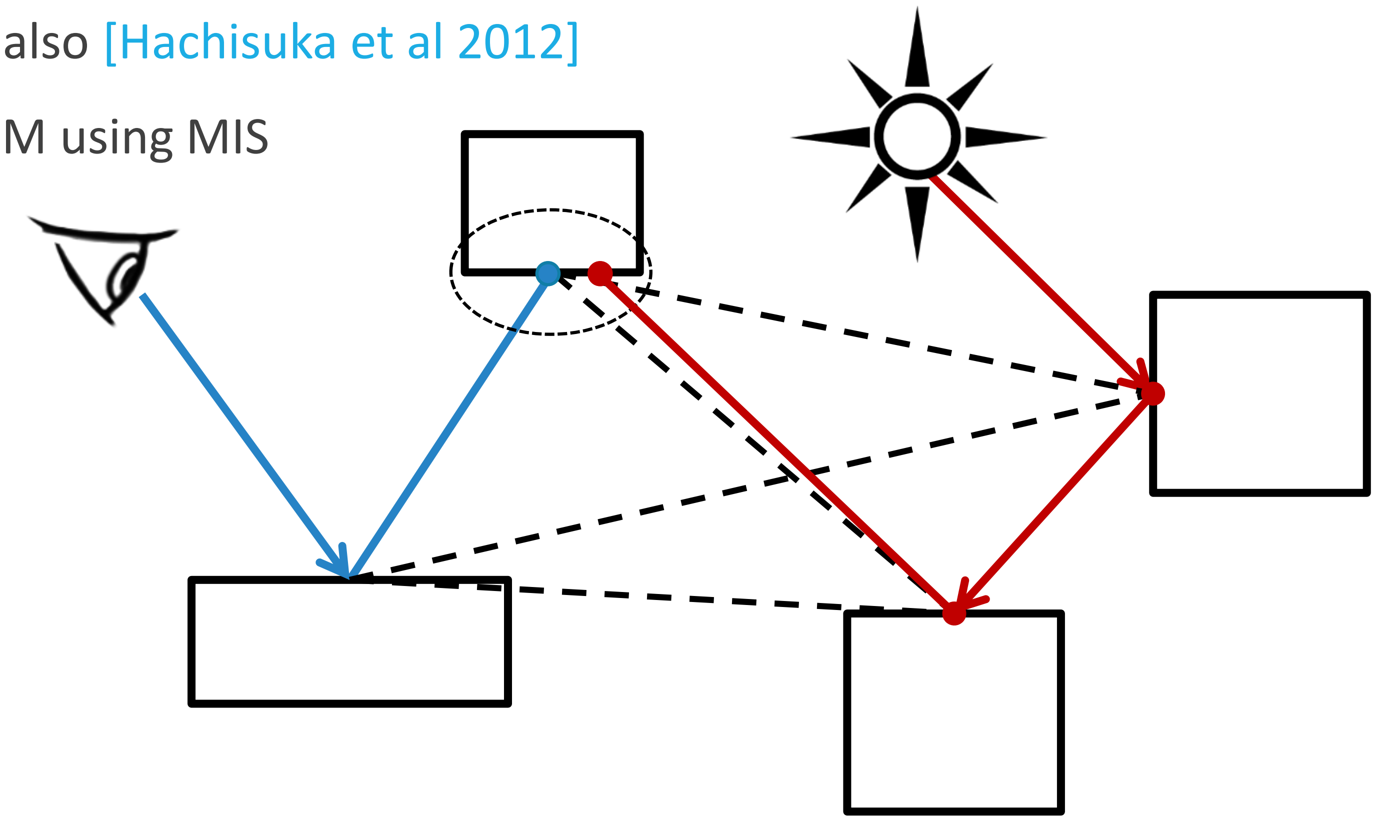
- Biased (but consistent, if progressive)

- Photon density might be too low if many photons end up in non-visible regions!

Vertex Connection and Merging (VCM)

[Georgiev et al 2012] also [Hachisuka et al 2012]

Combine BPT with PPM using MIS



MIS weights for VCM

Idea: Merging can be considered as using Russian Roulette and connect to the previous vertex.

With a given merge radius r we can compute the Russian Roulette PDF:

$p_{acc} = \pi r^2$

When computing the MIS weights, we have to also consider merging at every vertex

In our partial update scheme, this amounts to adding p_{acc} to the partial sum at every bounce

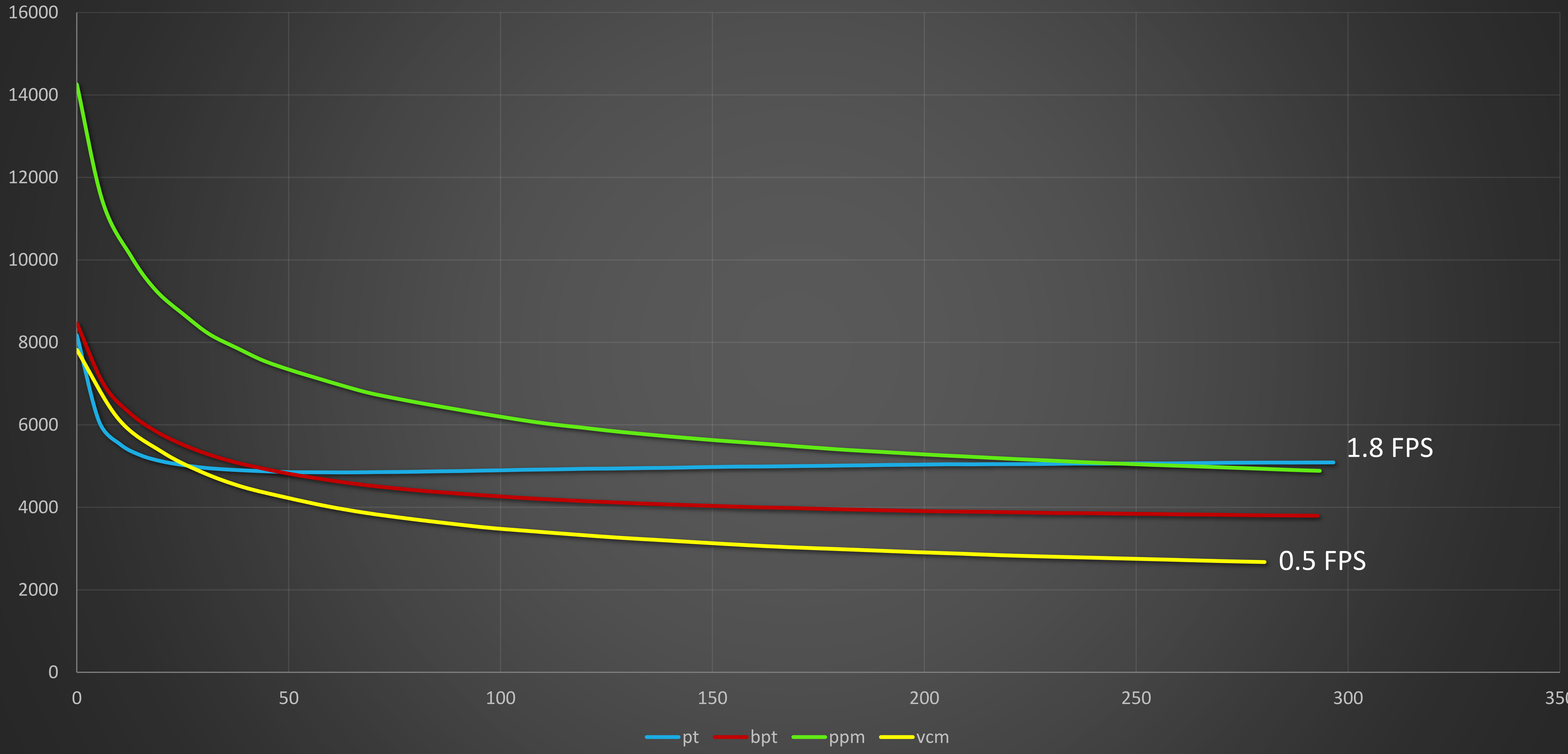
Results after 30 seconds

PT: no / noisy caustics

BPT: only directly visible caustics

VCM: all caustics,
but still visible bias
from the photon map

RMSE over time in the Still Life scene



Which Algorithm to Use in Practice?

VCM is robust – handles most types of scenes well

BUT

Memory Requirements

Requires storing all light path vertices (millions)

each at least 64 bytes -> do not fit in the cache anymore!

Many light vertices are not even useful for the scene!

Complexity

Combines multiple sampling techniques, much more complicated than a path tracer

➔ Difficult to implement efficiently, esp. in parallel

What did Common Production Renderers Choose?

Many pure path tracers,

- e.g. Cycles, Arnold, Disney Hyperion…

Some offer VCM (or variations thereof) in addition to path tracing,

- Meant to be used only for scenes with caustics
- e.g. Pixar RenderMan

Some combine path tracing and light tracing,

- Allows directly visible caustics at little extra cost
- e.g. NVIDIA IRay