Combining path integral estimators and particle density estimators in light transport simulation

Jaroslav Křivánek

Charles University in Prague

Computer Graphics Charles University

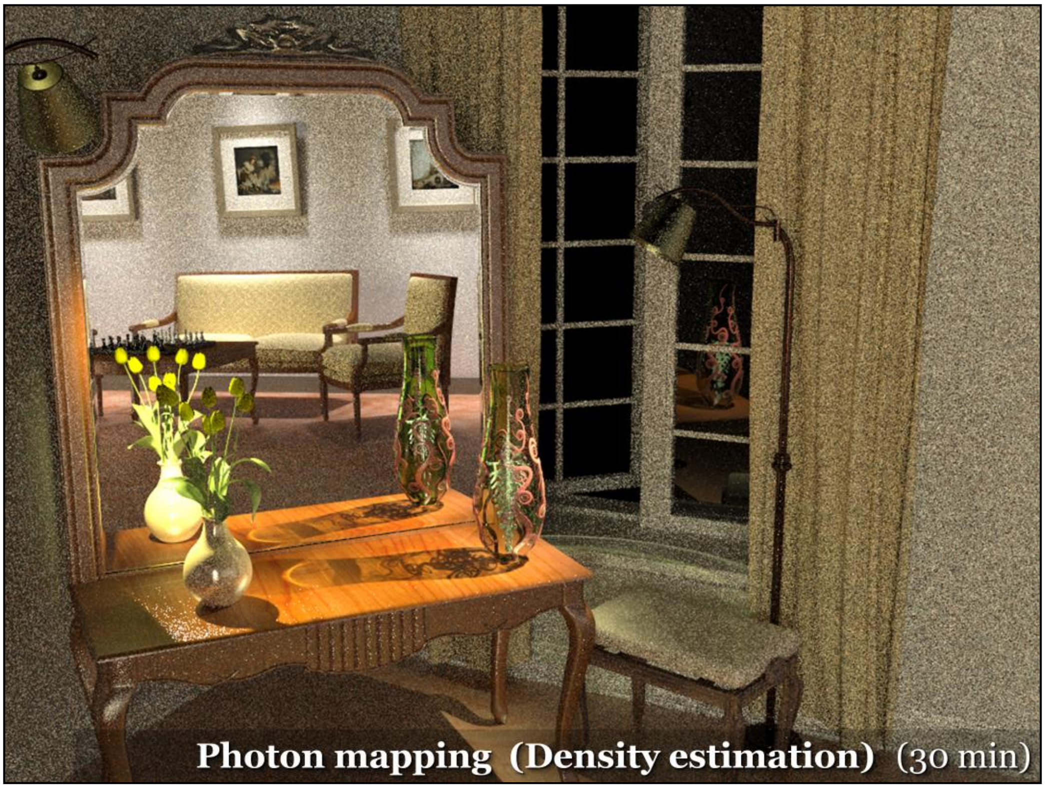
Contributors

- **I. Georgiev**, **J. Křivánek**, **T. Davidovič**, **P. Slusallek**
  SIGGRAPH Asia 2012.

- **T. Hachisuka**, **J. Pantaleoni**, and **H. W. Jensen**
  SIGGRAPH Asia 2012.

- **J. Křivánek**, **I. Georgiev**, **T. Hachisuka**, **P. Vévoda**, **M. Šik**, **D. Nowrouzezahrai**, **W. Jarosz**
  SIGGRAPH 2014.

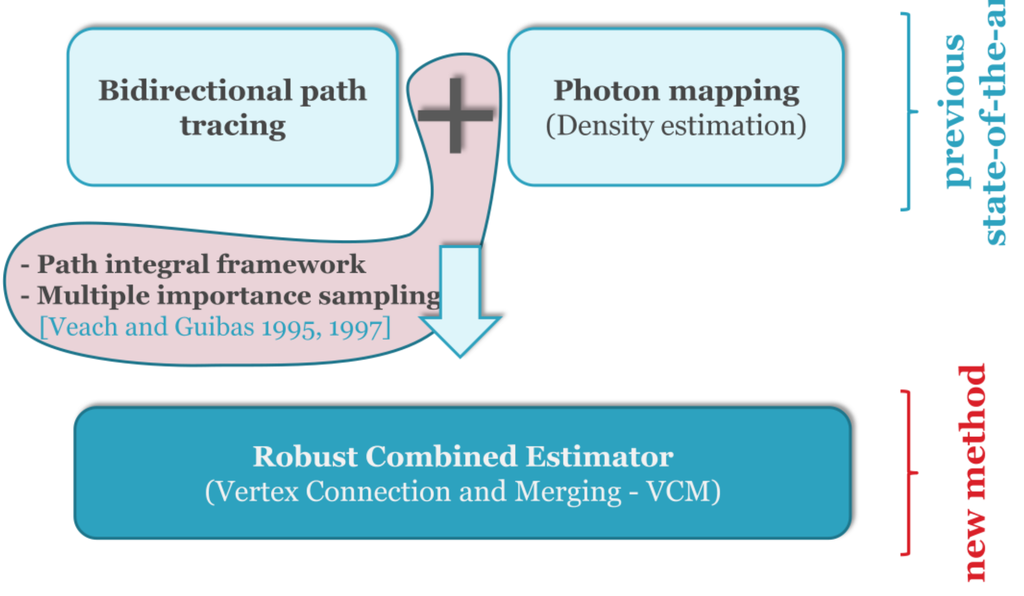
Bidirectional path tracing is one of the most versatile light transport simulation algorithms available today. It can robustly handle a wide range of illumination and scene configurations, but is notoriously inefficient for specular-diffuse-specular light interactions, which occur e.g. when a caustic is seen through a reflection/refraction.



On the other hand, photon mapping (PM) is well known for its efficient handling of caustics. Recently, Hachisuka and Jensen [2009] showed a progressive variant of PM that converges to the correct result with a fixed memory footprint. Their stochastic progressive photon mapping (PPM) algorithm captures the reflected caustics in our scene quite well. However, it has hard time handling the strong distant indirect illumination coming from the part of the scene behind the camera.

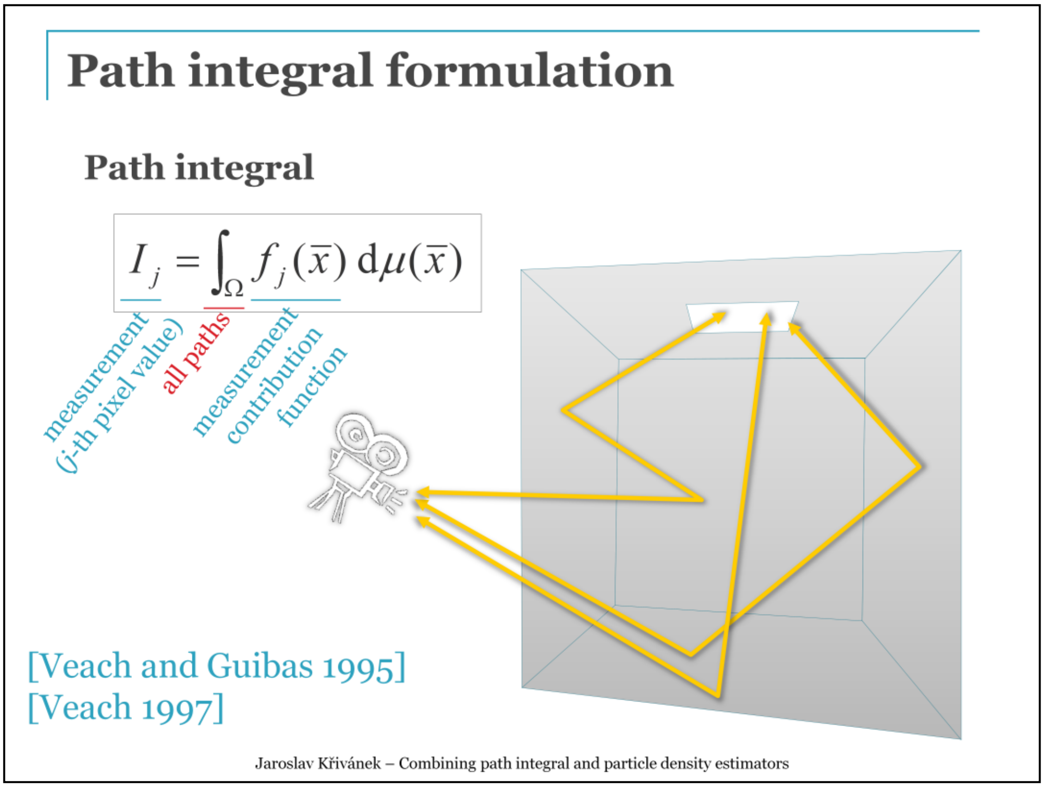
By using multiple importance sampling to combine estimators from bidirectional path tracing and photon mapping, the algorithm I will talk about today automatically finds a good mixture of techniques for each individual light transport path, and produces a clean image in the same amount of time.

Approach



Bidirectional path tracing
Photon mapping
(Density estimation)
previous state-of-the-art
- Path integral framework
- Multiple importance sampling
[Veach and Guibas 1995, 1997]
Robust Combined Estimator
(Vertex Connection and Merging - VCM)
new method

PATH INTEGRAL ESTIMATORS & BIDIRECTIONAL PATH TRACING

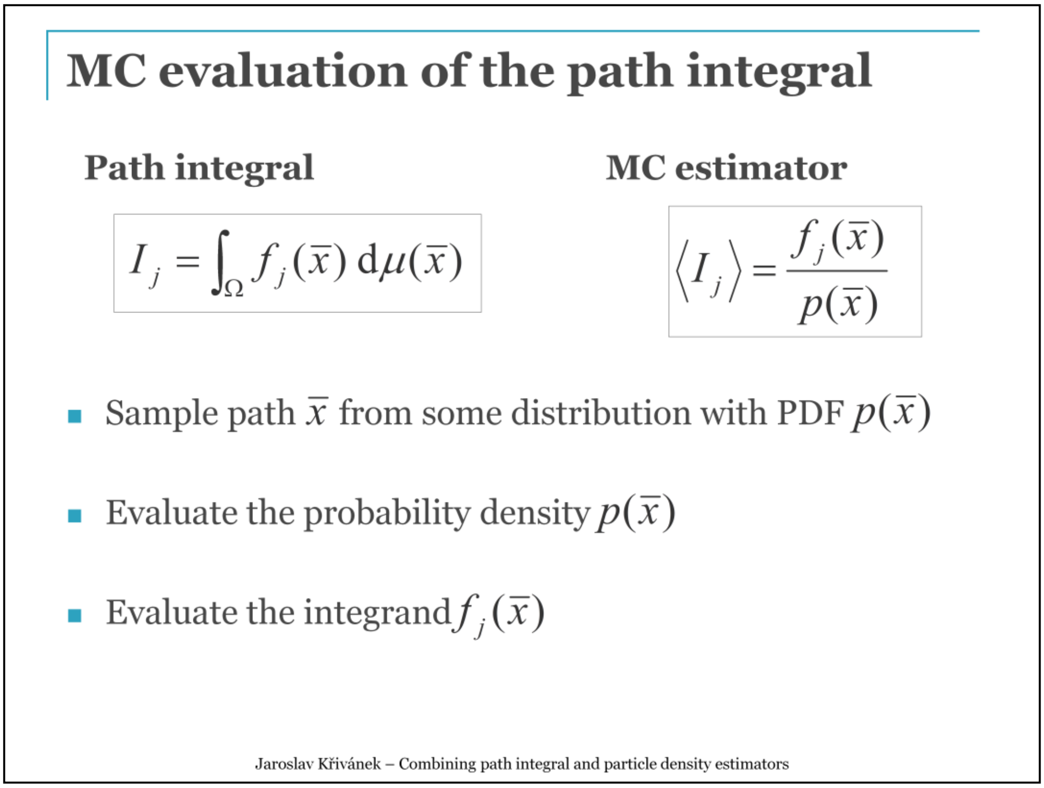


• The path integral formulation of light transport formalizes the idea that the response of the detector - camera in our case - is due to all the light particles – travelling over all possible paths – that hit the detector.

• The detector response is written as an integral over all light transport paths of all lengths in the scene.

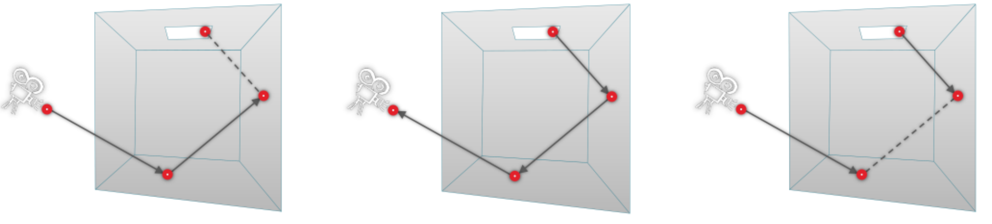
• The integrand of this integral is the so called “measurement contribution function”.

• The path integral formulation gives the pixel value as simple integral, which allows to argue about efficiency of different algorithms in terms of the probability density of sampling some types of paths, and even more importantly, it allows to combine different path sampling techniques through Multiple Importance Sampling.



- Thanks to the formal simplicity of the path integral formulation, applying Monte Carlo integration is really a more-or-less mechanical process.
- For each pixel, we need to repeatedly evaluate the estimator shown at the top right of the slide and average the estimates.
- This involves the following three steps:
  - First, we need to draw (or sample, or generate – all are synonyms) a random light transport path *x* in the scene (connecting a light source to the camera).
  - Then we need to evaluate the probability density of this path, and the contribution function.
  - Finally, we simply evaluate the formula at the top of the slide.
- Evaluating the path contribution function is simple – we have an analytic formula for this that takes a path and returns a number – the path contribution.

Many possible sampling techniques



no single technique works well by itself

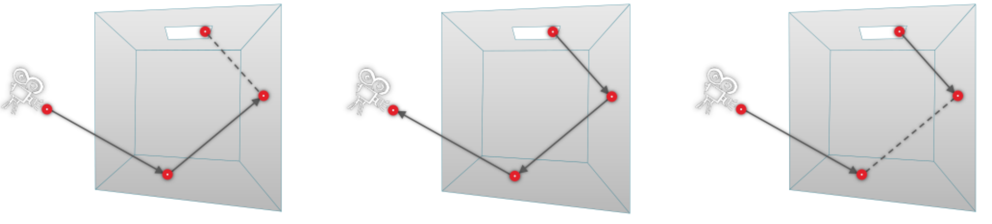
Bidirectional path tracing

- Use **all** of the above sampling techiques
- Combine using **Multiple Importance Sampling**

• The main idea of bidirectional path tracing is to use all of the sampling techniques above and combine them using multiple importance sampling.

Multiple Importance Sampling

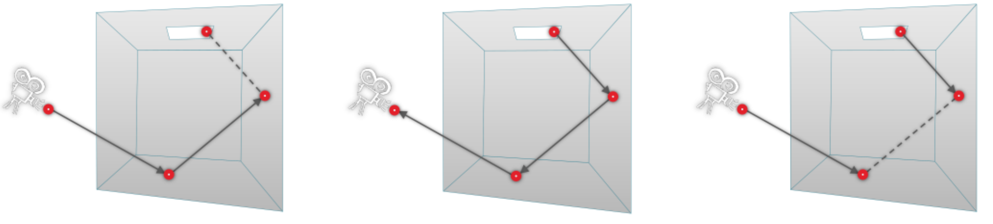
$p_1(\overline{x})$ $p_2(\overline{x})$ $p_3(\overline{x})$



$$w_i(\overline{x}) \propto p_i(\overline{x})$$

Multiple Importance Sampling

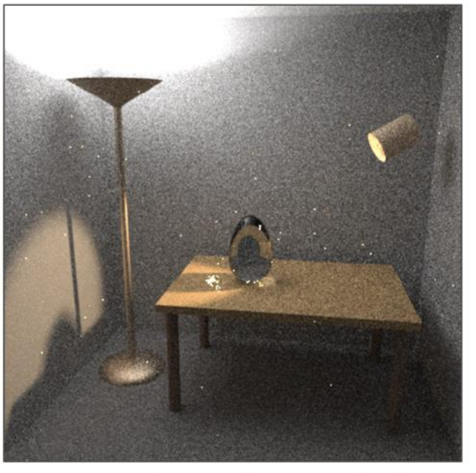
$p_1(\bar{x})$ $p_2(\bar{x})$ $p_3(\bar{x})$



$$w_i(\bar{x}) = \frac{p_i(\bar{x})}{p_1(\bar{x}) + p_2(\bar{x}) + p_3(\bar{x})}$$

Bidirectional path tracing

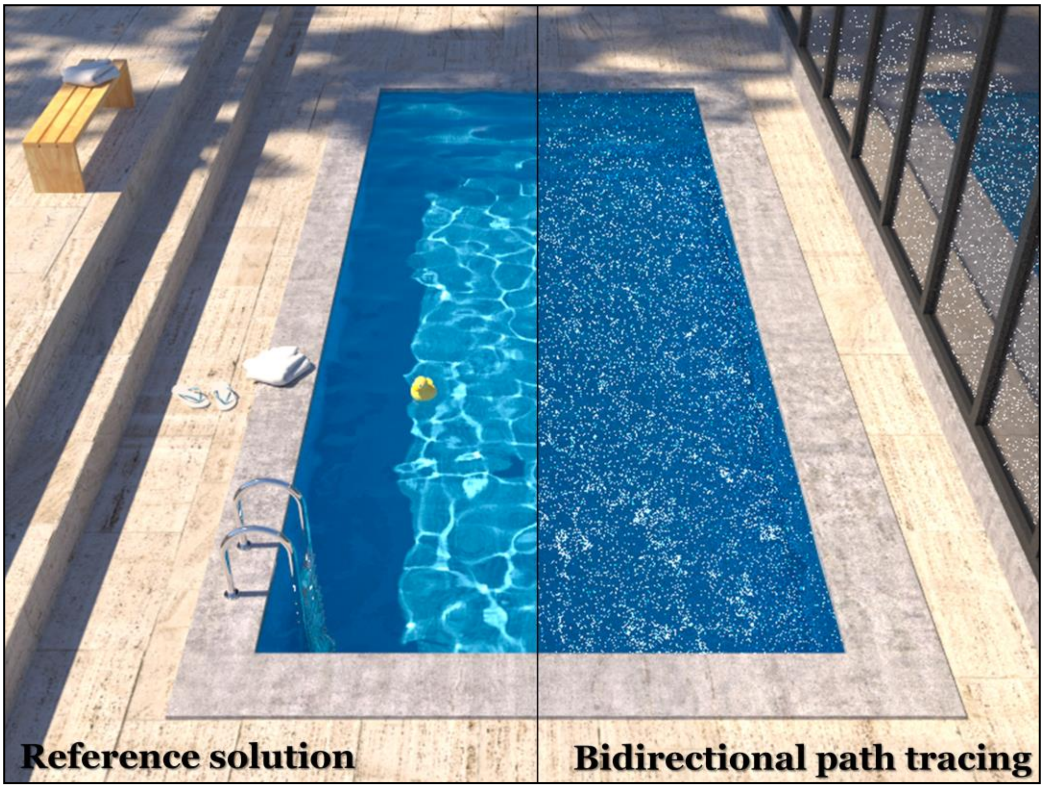
BPT, 25 samples per pixel



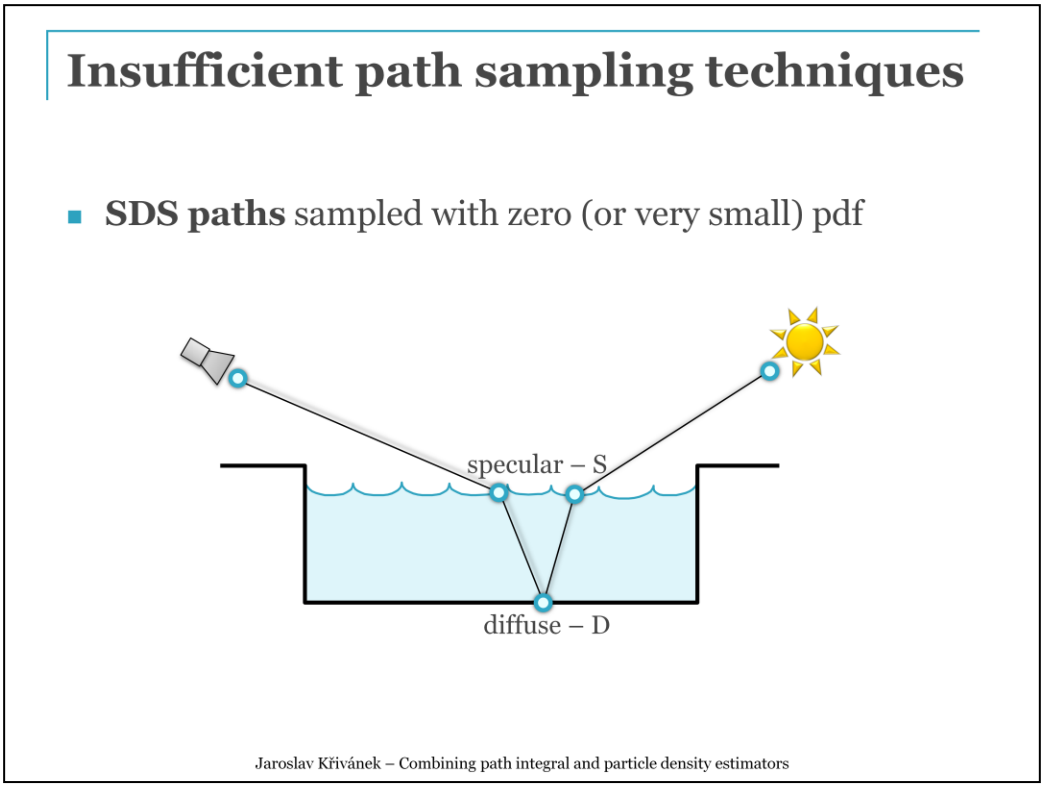
PT, 56 samples per pixel

Images: Eric Veach

LIMITATIONS OF BIDIRECTIONAL PATH TRACING



• In this rendering of the pool, BPT is unable to reproduce the caustics seen at the pool bottom.



• As usual, the culprit is inappropriate path sampling. The problem is that none of the path sampling techniques used in bidirectional path tracing is efficient at sampling the SDS effects, so their combination cannot sample those effects either.

• To see this, consider the example on the slide. We have a pool (diffuse – D) filled with water (specular – S), a pinhole camera, and a small light source.

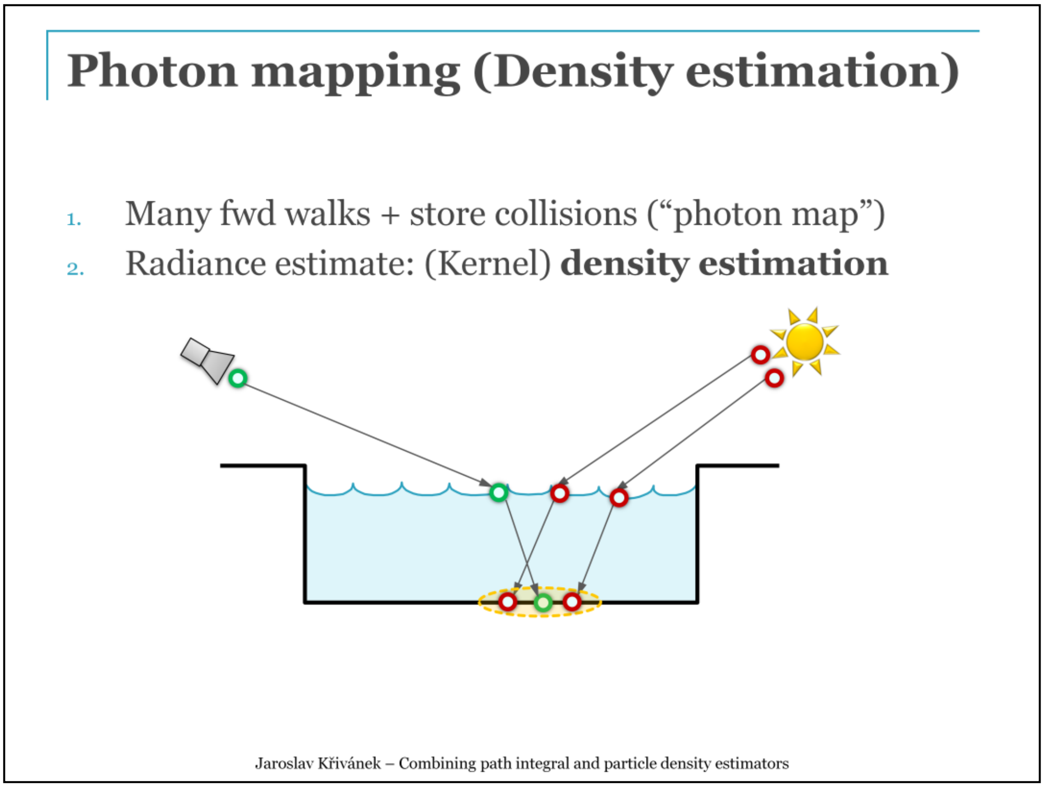
• No path connections are possible because of the two specular vertices. Unidirectional sampling from the light source is not possible either because of the pinhole camera.

• So we are left with one single (unidirectional) path sampling technique that starts from the camera, and hope to randomly strike the light source. The smaller the source the lower the probability of hitting it and the higher the estimator variance.

• In the limit, for point sources and pinhole camera, the SDS effects cannot be sampled by local path sampling at all.

PHOTON MAPPING

(DENSITY ESTIMATION)



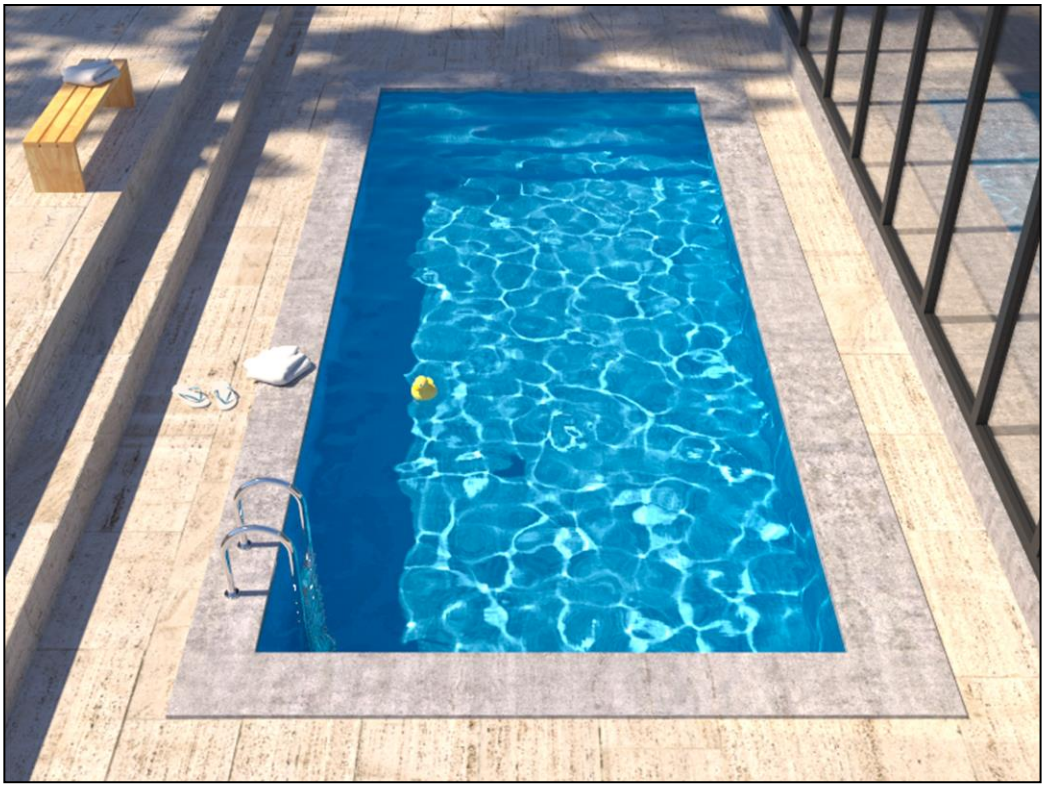
• Photon mapping – a computer graphics name for a light transport technique based on non-parametric density estimation – is well known for its efficient handling of caustics and reflected caustics.

• The algorithm proceeds in two stages.

• In the first stage, a large number of particle histories (light paths) are traced starting from the light source. Every time the particle scatters, we store it in a data structure called the "photon map".

• In the second stage, we generate the image by tracing sub-paths starting from the camera and whenever we hit a non-specular object, we estimate the radiance by a "photon map lookup" and terminate the walk.

• The photon map lookup estimates the radiance using a non-parametric density estimation technique; k-NN density estimation is most common. This involves finding the k nearest neighboring particles in the photon map and estimating the radiance as a weighted density of these particles.

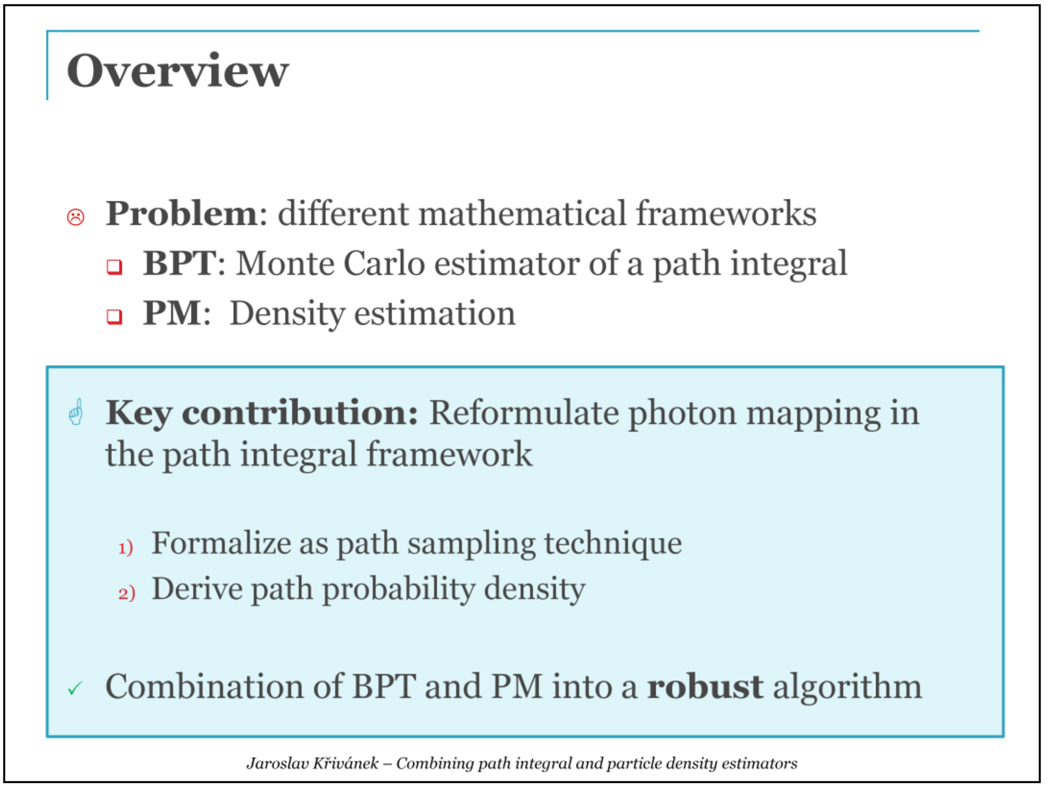


• This approach has absolutely no problem handling SDS paths because – unlike bidirectional path tracing – it does not rely on connecting sub-path with an edge.

• Instead, the connection is made using the density estimate, which can be readily performed at the bottom of the pool.

COMBINING

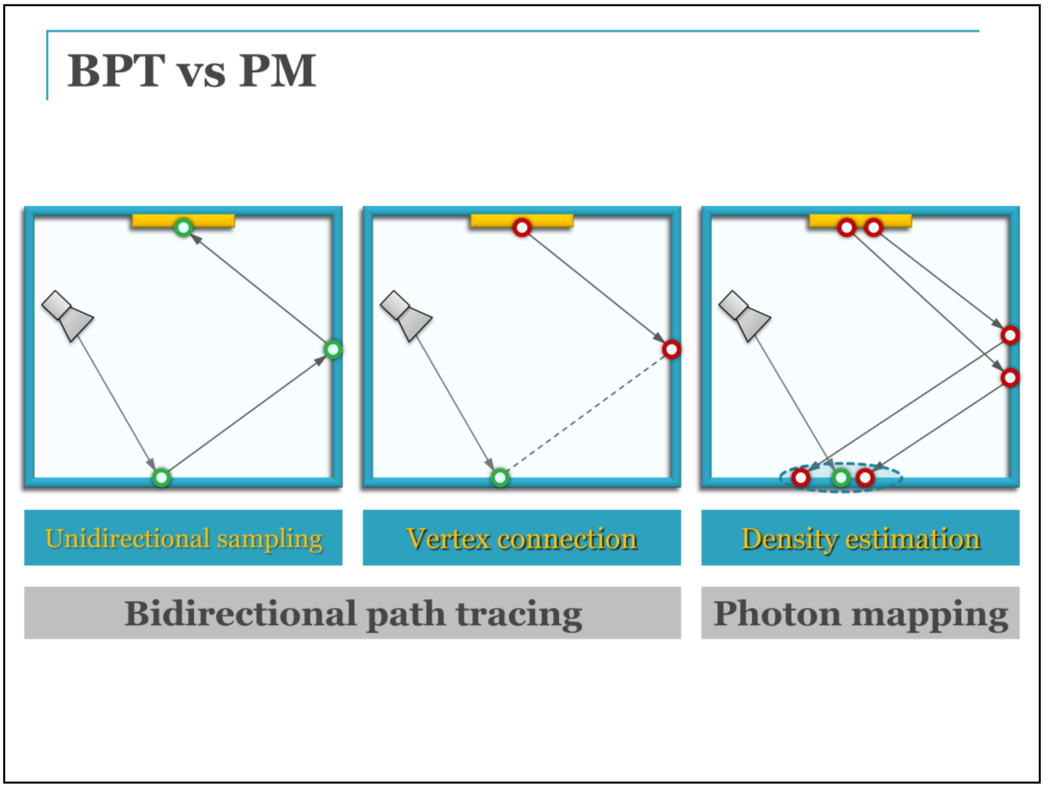
BIDIRECTIONAL PATH TRACING
&
PHOTON MAPPING



However, even though both methods have been published more than 15 years ago, neither a rigorous analysis of their relative performance nor an efficient combination had been shown until very recently. The reason for this is that BPT and PM have originally been defined in different theoretical frameworks – BPT as a standard Monte Carlo estimator to the path integral, and PM as an outgoing radiance estimator based on photon density estimation.

The first step toward combining these two methods is to put them in the same mathematical framework. We choose Veach's path integral formulation of light transport, as it has a number of good properties and also because BPT is already naturally defined in this framework.

We need two key ingredients: (1) express PM as a sampling technique that constructs light transport paths that connect the light sources to the camera, and (2) derive the probability densities for paths sampled with this technique. This will give us a basis for reasoning about the relative efficiency of BPT and PM. And more importantly, it will lay the ground for combining their corresponding estimators via *multiple importance sampling*.

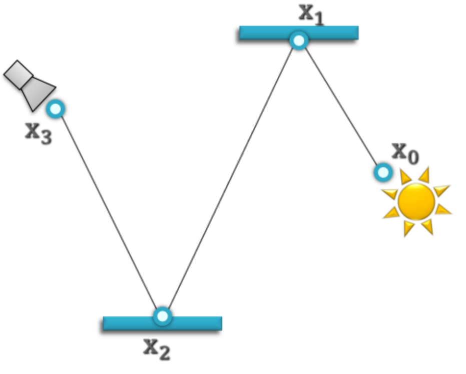


Let us start by reviewing how bidirectional path tracing (BPT) and photon mapping (PM) sample light transport paths that connect the light sources to the camera:

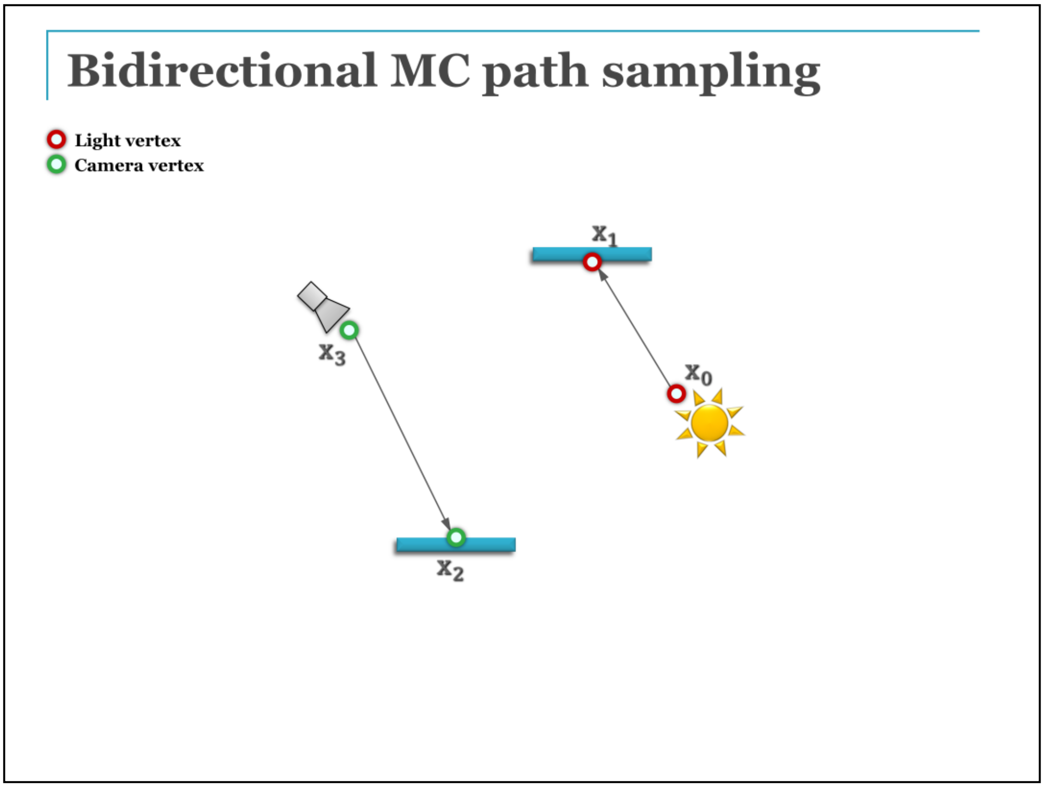
[CLICK] The techniques BPT employs can be roughly categorized to *unidirectional sampling* (US) and *vertex connection* (VC). [CLICK] US constructs a path by starting either from a light source or the camera and tracing a random walk in the scene until termination. [CLICK] On the other hand, VC traces one subpath from a light source and another one from the camera, and then completes a full path by connects their endpoints.

[CLICK] In contrast, PM first traces a number of light subpaths and stores their vertices, a.k.a. photons. It then traces subpaths from the camera and computes the outgoing radiance at the hit points using density estimation by looking up nearby photons.

Bidirectional MC path sampling

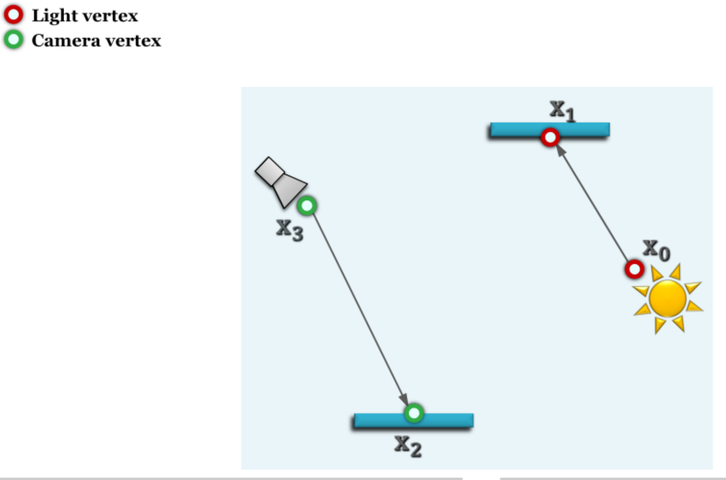


Let us start by taking a simple length-3 path and see how it can be constructed bidirectionally.



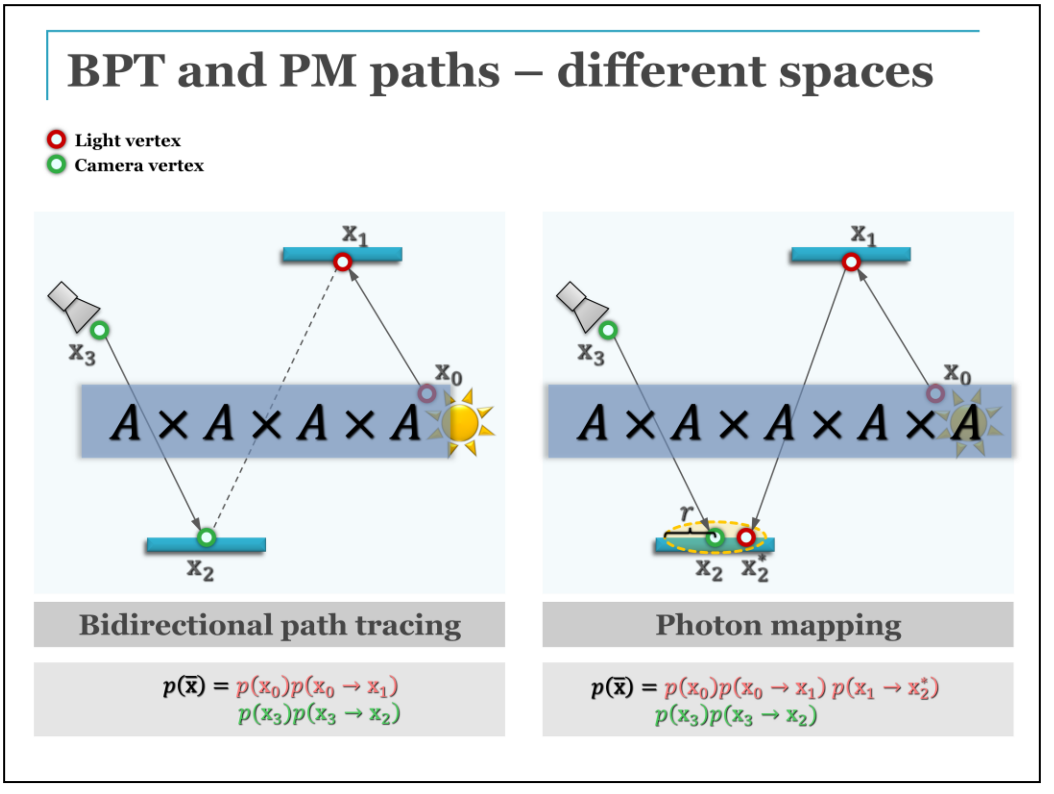
We first trace one subpath from the camera and another one from a light source.

Bidirectional MC path sampling



Bidirectional path tracing | Photon mapping

Now let's see how we complete a full path in BPT and PM.



Bidirectional path tracing connects the subpath endpoints deterministically.

[CLICK] We call this technique *vertex connection*. The PDF of the resulting full path is well known, and is simply the product of the PDFs of two subpaths, which have been sampled independently.

[CLICK] Photon mapping, on the other hand, will extend the light subpath by sampling one more vertex from $\mathbf{x}_1$, and will concatenate the two subpaths only if the "photon" hit-point $\mathbf{x}_2^*$ lies within a distance r from $\mathbf{x}_2$.

[CLICK] We label this technique *vertex merging*, as it can be intuitively thought to weld the endpoints of the two subpaths if they lie close to each other.

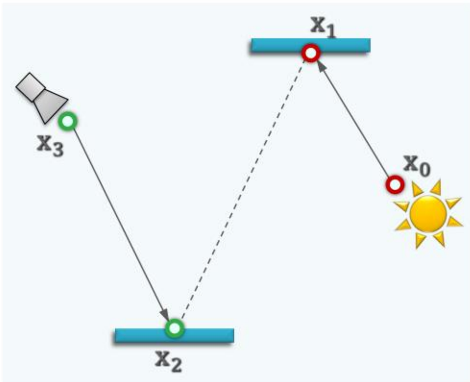
[CLICK] What remains is to derive the PDF of the resulting full path. To do this, we can interpret the last step as establishing a regular vertex connection between $\mathbf{x}_1$ and $\mathbf{x}_2$, but conditioning its acceptance on the random event that a vertex $\mathbf{x}_2^*$ sampled from $\mathbf{x}_1$ lands within a distance r to $\mathbf{x}_2$. This probabilistic acceptance is nothing more than a Russian roulette decision. The full path PDF is then again the product of the subpath PDFs, but in addition multiplied by the probability of sampling the point $\mathbf{x}_2^*$ within a distance r of $\mathbf{x}_2$. This acceptance probability is equal to the integral of the PDF of $\mathbf{x}_2^*$ over the r-neighborhood of $\mathbf{x}_1$.

[CLICK] Under the reasonable assumptions that the surface around $\mathbf{x}_1$ is locally flat, i.e. that this neighborhood is a disk, and that the density of $\mathbf{x}_2^*$ is constant inside this disc, the integral can be well approximated by the PDF of the actual point $\mathbf{x}_2^*$ we have sampled, multiplied by the disc area πr^2.

Vertex Connection and Merging

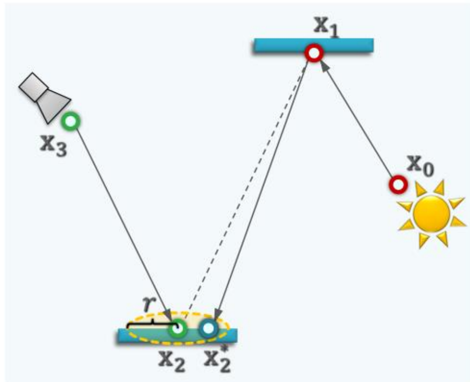
- Light vertex
- Camera vertex

$$A \times A \times A \times A$$



Bidirectional path tracing / **Vertex connection**

$$p_{VC}(\bar{\mathrm{x}}) = p(\mathrm{x}_0)p(\mathrm{x}_0 \to \mathrm{x}_1)\; p(\mathrm{x}_3)p(\mathrm{x}_3 \to \mathrm{x}_2)$$



Photon mapping / **Vertex merging**

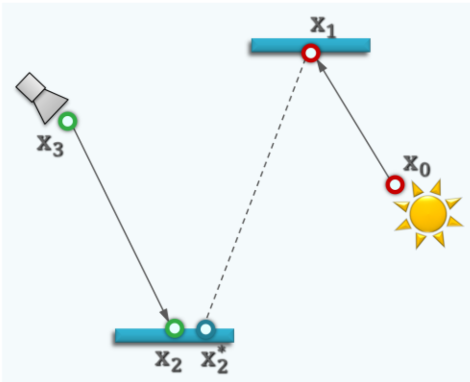
$$p_{VM}(\bar{\mathrm{x}}) \approx p(\mathrm{x}_0)p(\mathrm{x}_0 \to \mathrm{x}_1)\; P(\|\mathrm{x}_2 - \mathrm{x}_2^*\| < r)\; p(\mathrm{x}_3)p(\mathrm{x}_3 \to \mathrm{x}_2)$$

$$p(\mathrm{x}_1 \to \mathrm{x}_2^*)\pi r^2$$

Extended path space [Hachisuka et al.]

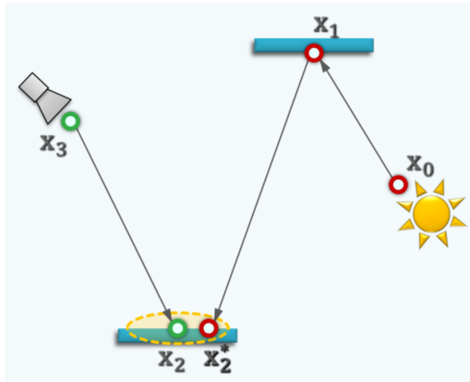
Light vertex
Camera vertex

$$A \times A \times A \times A \times A$$



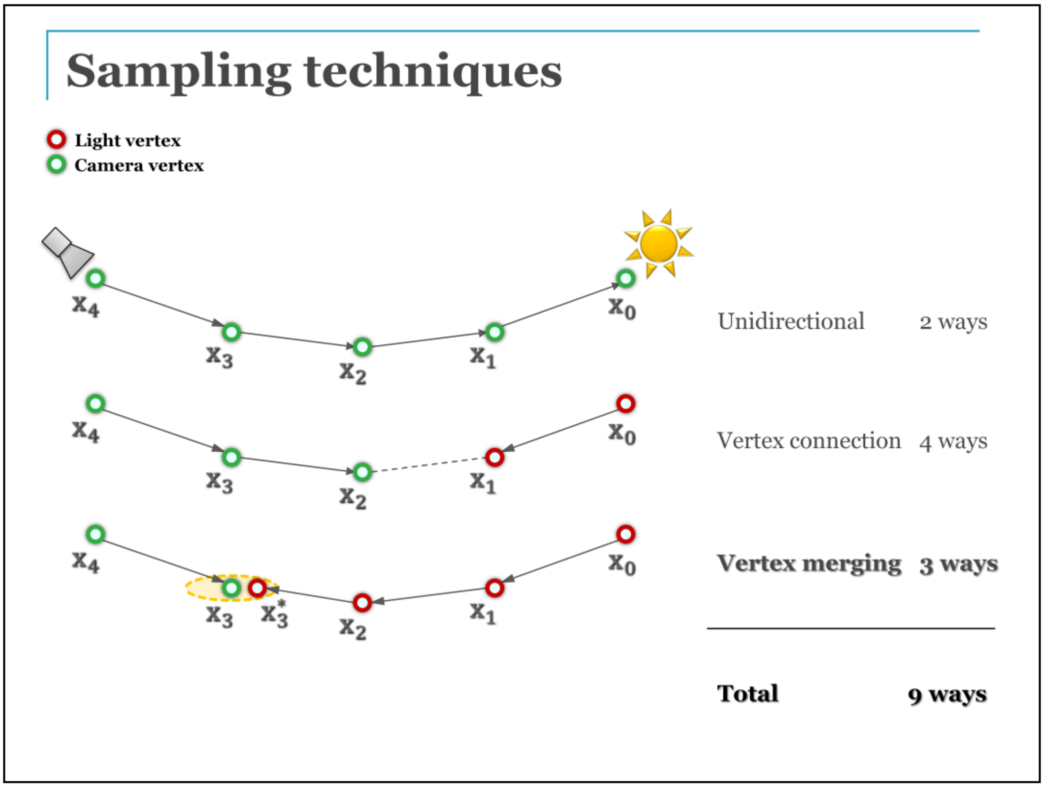
Bidirectional path tracing

$$p(\bar{x}) = p(x_0)p(x_0 \to x_1)\, p(x_3)p(x_3 \to x_2)/(\pi r^2)$$



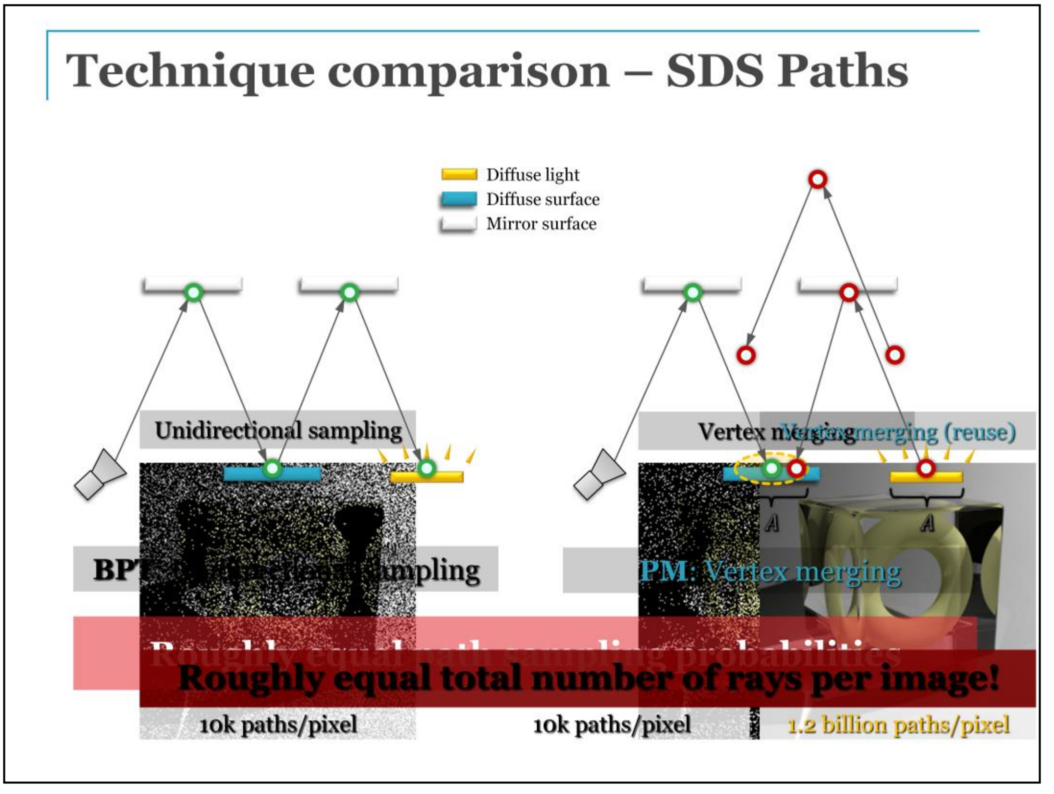
Photon mapping

$$p(\bar{x}) = p(x_0)p(x_0 \to x_1)\ p(x_1 \to x_2^*)\ p(x_3)p(x_3 \to x_2)$$



Now that we have formulated the vertex merging path sampling technique, we can put it side by side with the already available techniques in BPT. There are two ways to sample a length-4 path unidirectionally, and four ways to sample it via vertex connection. Vertex merging adds five new ways to sample the path, corresponding to merging at the five individual path vertices. In practice, we can avoid merging at the light source and the camera, as directly evaluating emission and sensitivity is usually cheap.

But with so many ways to sample the same light transport path, a question naturally arises in the mind of the curious: which technique is the most efficient for what types of paths?



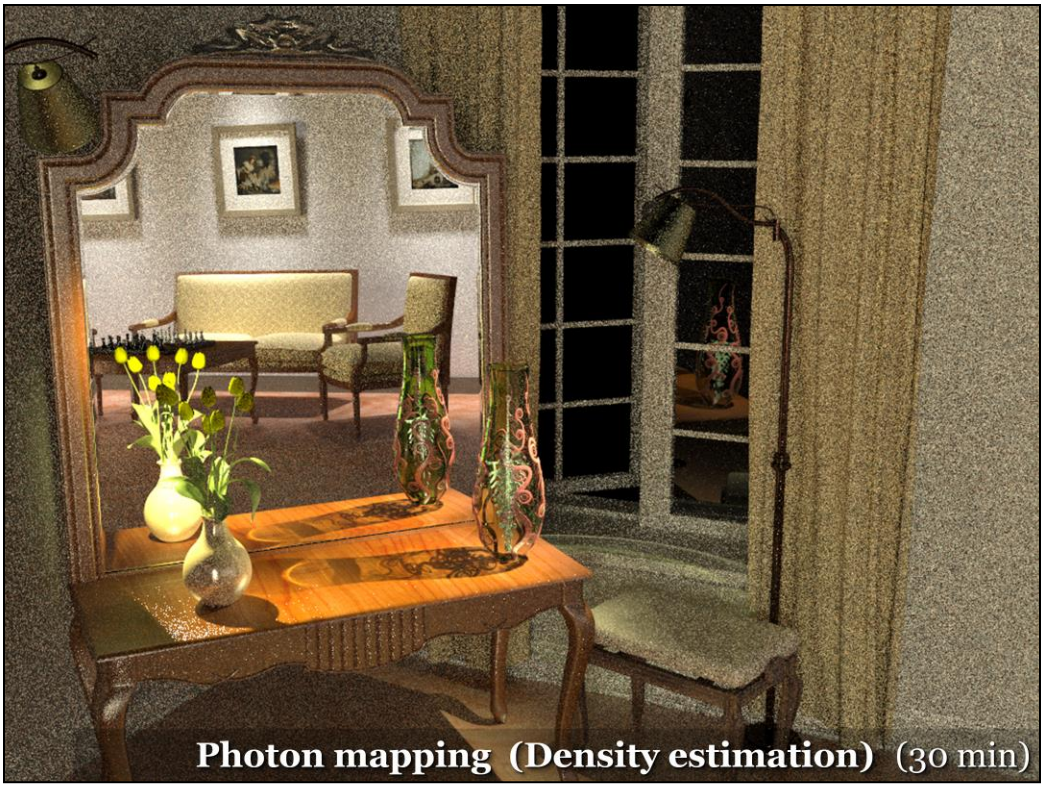
To answer this question, let us first take a look at specular-diffuse-specular (SDS) paths. Here, bidirectional path tracing can only rely on unidirectional sampling: it traces a path from the camera hoping to randomly hit the light source. With vertex merging, we can trace one light and one camera subpath, and merge their endpoints on the diffuse surface.

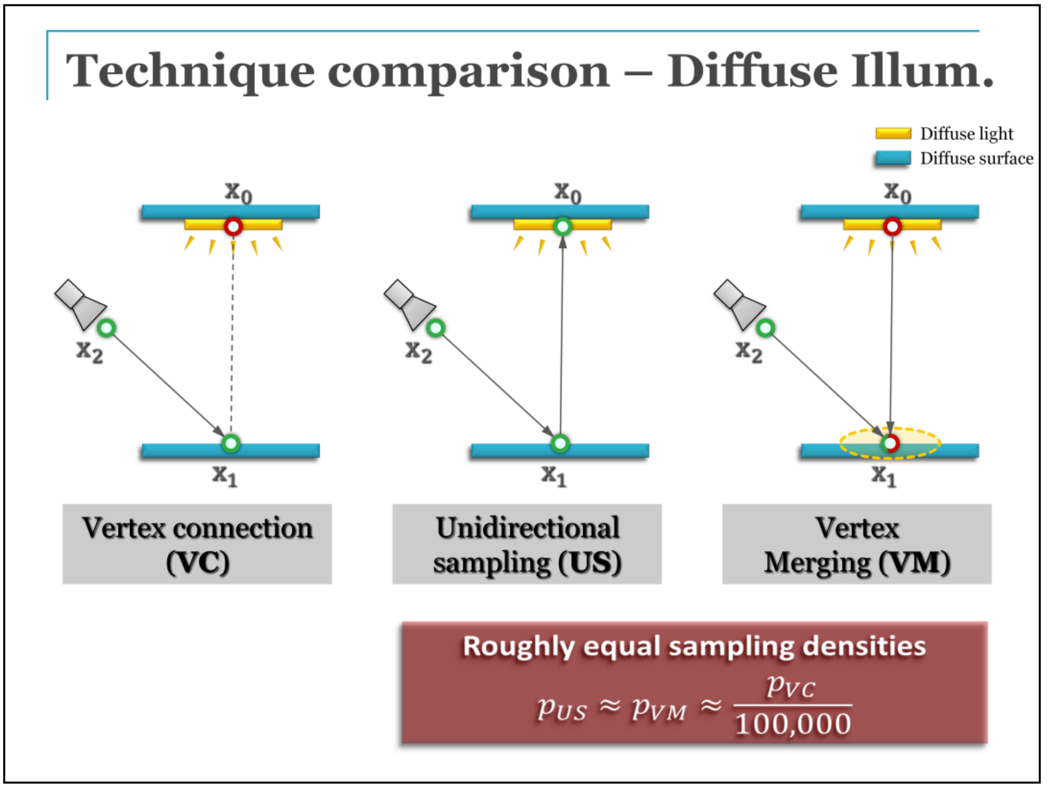
[CLICK] It can be shown that if the light source and the merging disk have the same area A, then unidirectional sampling and vertex merging sample paths with roughly the same probability density. This means that we should expect the two techniques to perform similarly in terms of rendering quality.

[CLICK] We render these two images progressively, sampling one full path per pixel per iteration. For the left image we trace paths from the camera until they hit the light. For image on the right, we trace subpaths from both ends, and merge their endpoints if they lie within a distance $r = \sqrt{A/\pi}$ from each other. Both images look equally noisy, even after sampling 10,000 paths per pixel. This confirms that vertex merging, and thus photon mapping, is *not* an intrinsically more robust sampling technique for SDS paths than unidirectional sampling.

[CLICK] However, the strength of vertex merging is computational efficiency – we can very efficiently reuse the light subpaths traced for *all* pixels at the cost of a single range search query. This allows us to quickly construct orders of magnitude more light transport estimators from the same sampling data, with a minimal computational overhead, resulting in a substantial quality improvement.

[CLICK] For all these three images we have traced roughly the same number of rays, and the only difference between the one in the center and the one on the right is that the for right image we have enabled path reuse, by storing, and looking up, the light subpath vertices in a photon map at every rendering iteration.


Photon mapping (Density estimation) (30 min)



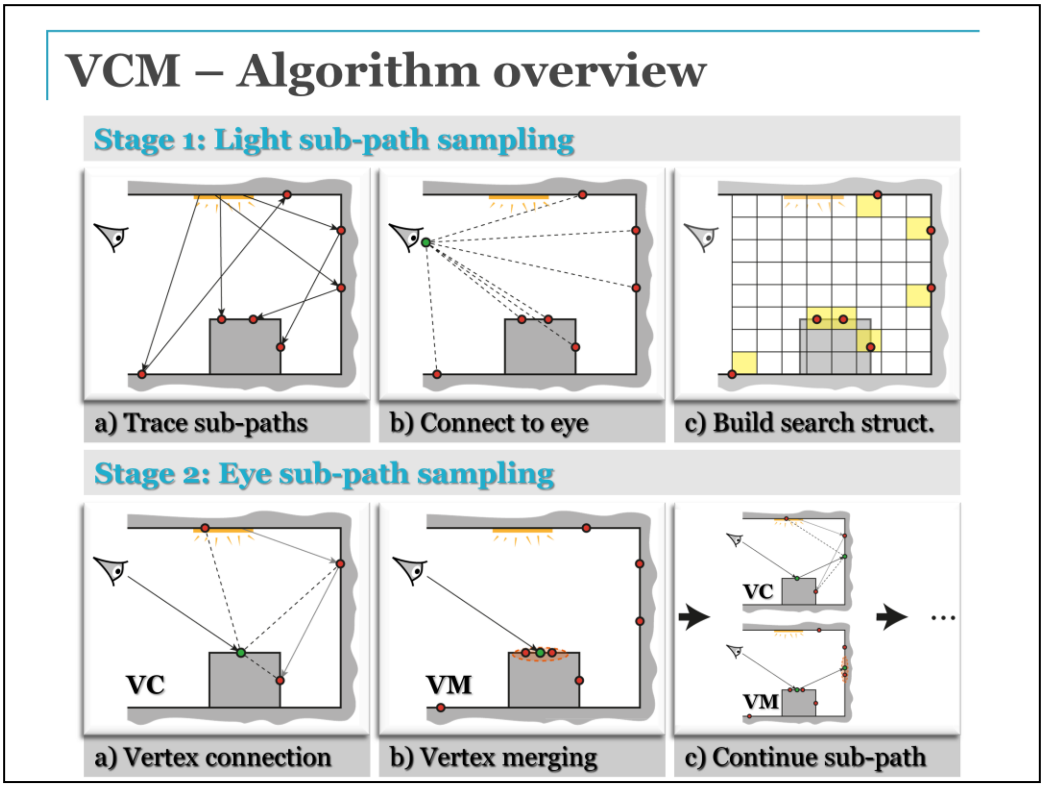
Now let's look at another extreme example – diffuse illumination. Note that vertex connection (VC) constructs the edge between $\mathbf{x}_1$ and $\mathbf{x}_2$ deterministically, while unidirectional sampling (US) and vertex merging (VM) both rely on random sampling.

Once again, it can be shown that if the light source and the merging disk have the same area, then US and VM sample this path with roughly the same probability density.

[CLICK] For the specific case shown on this slide, this density is about 100,000 lower than that of VC. This demonstrates that VM is not an intrinsically more robust sampling technique than VC either. This is not surprising – if we recall the expression for the VM path PDF, we see that it can only be lower than that of the corresponding VC technique, as their only difference is the probability factor in the VM PDF, which is necessarily in the range $[0; 1]$. Still, by reusing paths across pixels, vertex merging, and thus photon mapping, gains a lot of efficiency over unidirectional sampling.

All these useful insights emerge from the reformulation of photon mapping as a path sampling technique.


Bidirectional path tracing (30 min)



Even more usefully, we now have the necessary ingredients for combining photon mapping and bidirectional path tracing into one unified algorithm. The vertex merging path PDFs tell us how to weight all sampling techniques in multiple importance sampling, and the insights from the previous two slides command to strive for path reuse.

The combined algorithm, which we call *vertex connection and merging* (VCM), operates in two stages.

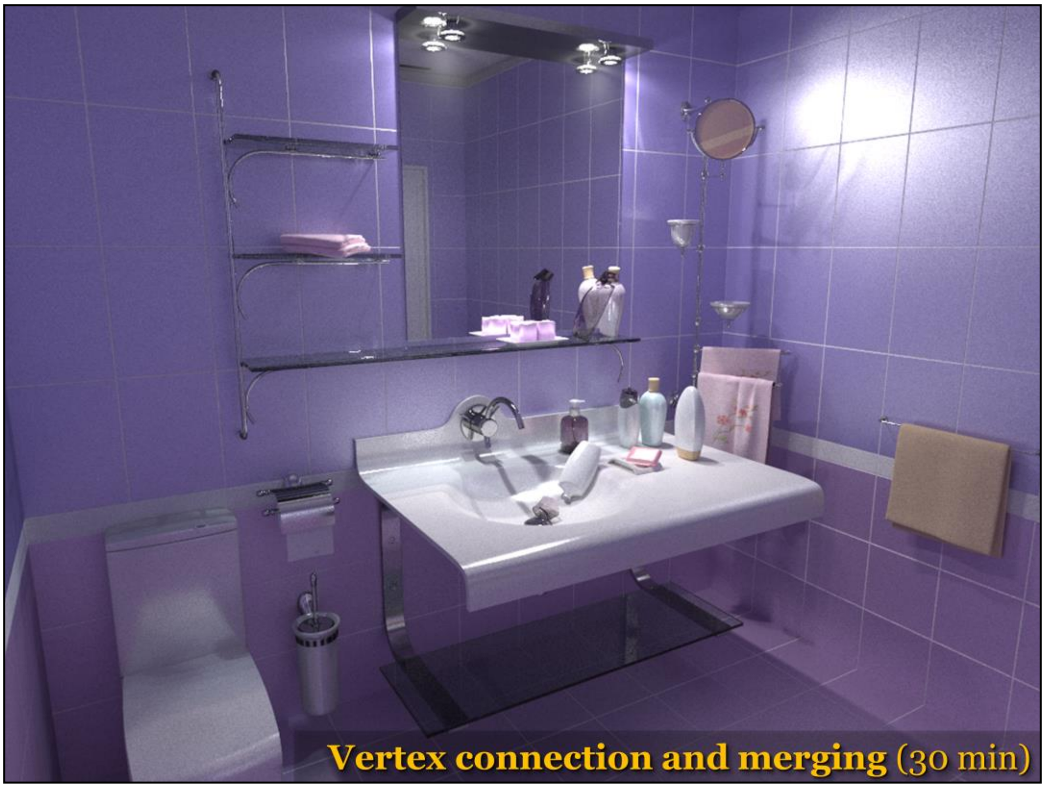
1. In the first stage, we
   a) [CLICK] trace the light subpaths for all pixels,
   b) [CLICK] connect them to the camera, and
   c) [CLICK] store them in a range search acceleration data structure (e.g. a kd-tree or a hashed grid).

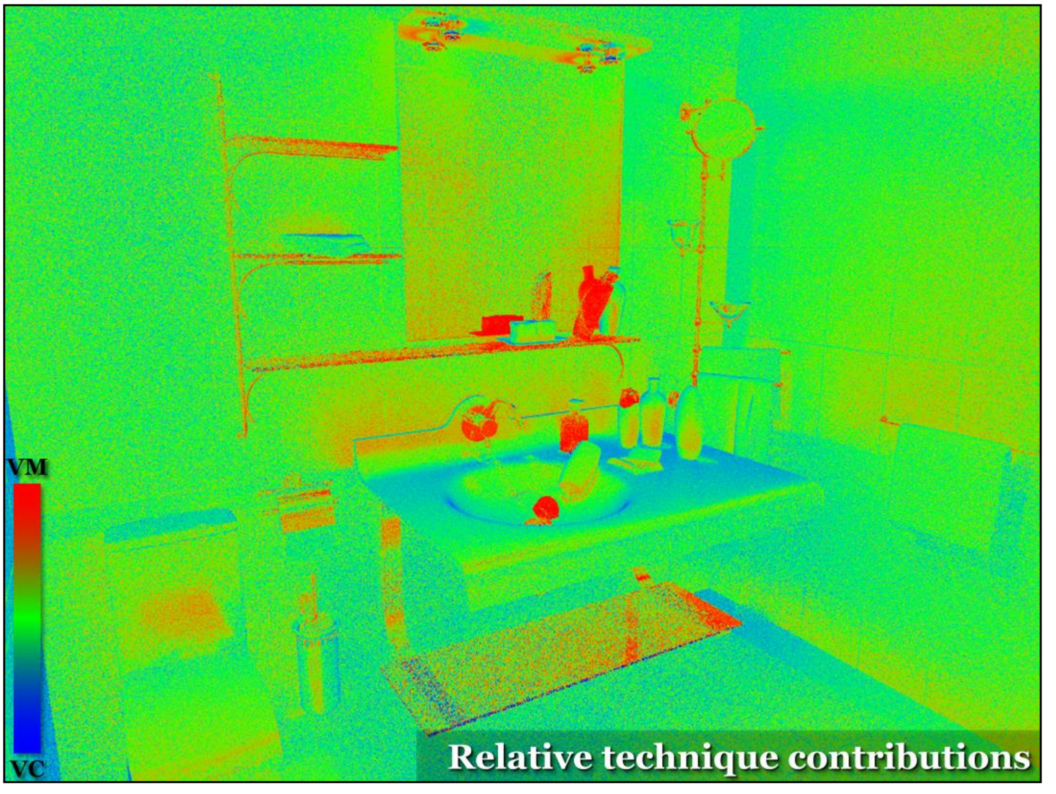
2. [CLICK] In the second stage, we trace a camera subpath for every pixel.
   a) [CLICK] Each sampled vertex on this path is connected to a light source (a.k.a. next event estimation), connected to the vertices of the light subpath corresponding to that pixel, and
   b) [CLICK] merged with the vertices of *all* light subpaths.
   c) [CLICK] We then sample the next vertex and do the same.

In a progressive rendering setup, we perform these steps at each rendering iteration, progressively reducing the vertex merging radius . For details on this, please refer to the cited papers below for details.

Let us now see how this combined algorithm stacks up against bidirectional path tracing and stochastic progressive photon mapping on a number of scenes with complex illumination.


Stochastic progressive photon mapping (30 min)


Vertex connection and merging (30 min)

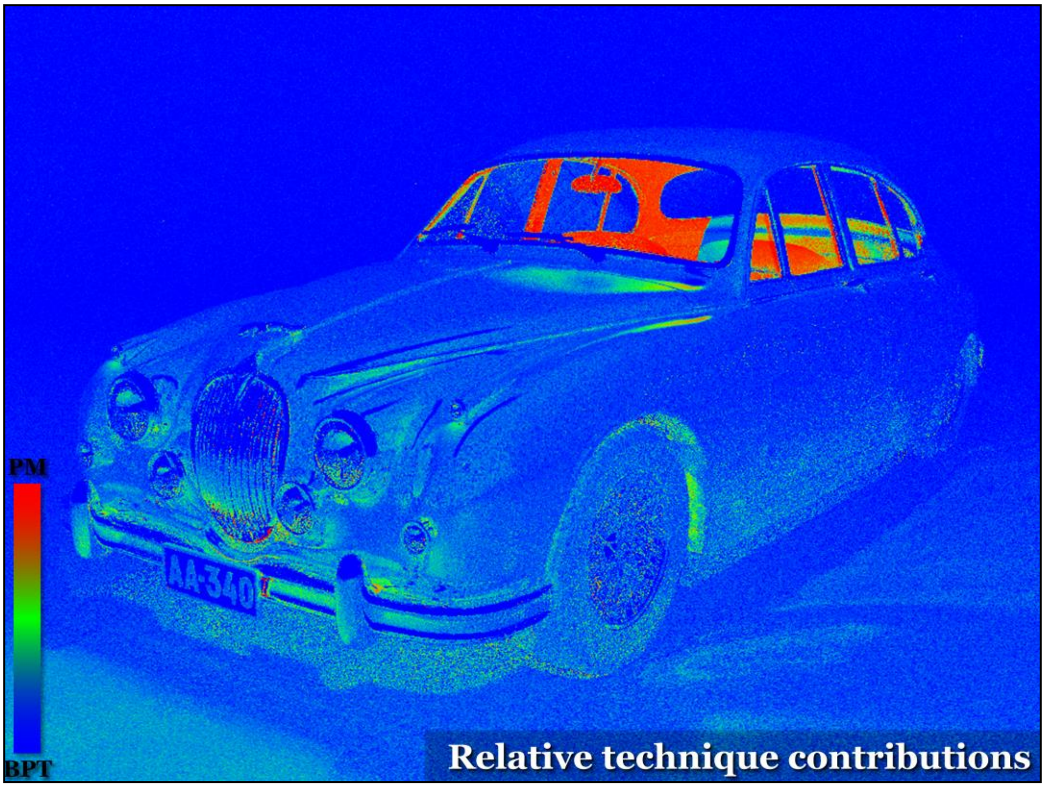


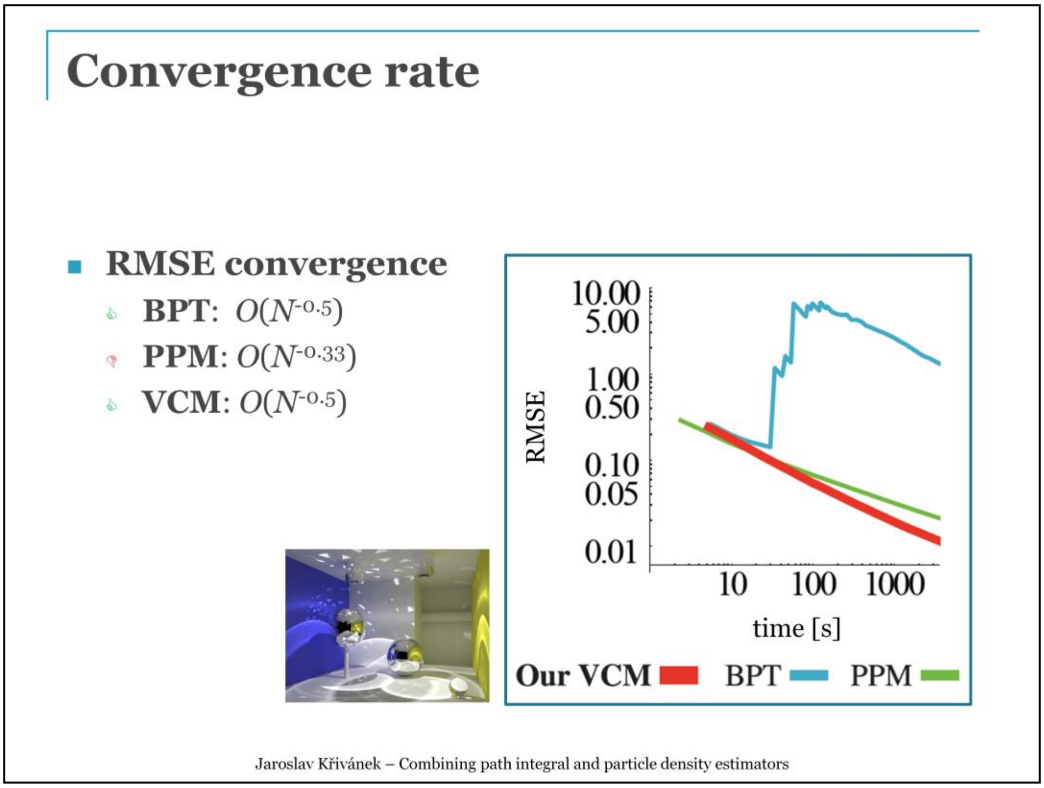
Here, we visualize the relative contributions of VM and VC techniques to the VCM image from the previous slide. This directly corresponds to the weights that VCM assigned to these techniques.


AA-340
Bidirectional path tracing (30 min)


AA-340
Stochastic progressive photon mapping (30 min)


AA-340
Vertex connection and merging (30 min)

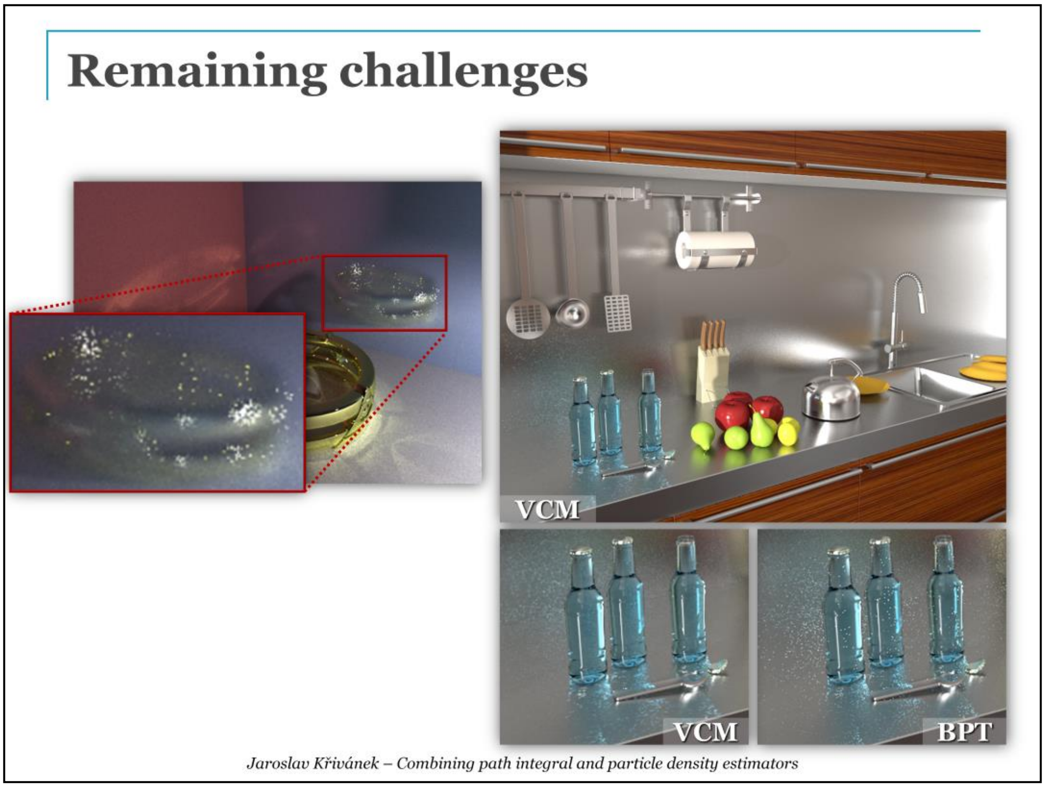

PM
BPT
AA-340
Relative technique contributions



An important property of the algorithm is that it retains the higher order of convergence of BPT, meaning that it approaches the correct solution faster than PPM as we spend more computational effort (i.e. sample more paths). The asymptotic analysis can be found in the VCM paper.

Summary

- Image synthesis requires **robust** estimators
- **Vertex Connection and Merging**
  - Bidirectional path tracing + Photon mapping (density est.)
  - Photon mapping (density est.) as a path sampling technique
- **Invaluable tools**
  - Multiple importance sampling
  - Path integral view of light transport



Even though VCM is a step forward in Monte Carlo rendering and has proven very useful in practice, it doesn't come without limitations. Specifically, it cannot handle more efficiently those types of light transport paths that are difficult for both BPT and PM to sample.

[CLICK] A prominent example are caustics falling on a glossy surface.

[CLICK] On this kitchen scene, even though VCM brings practical improvements over BPT, there is still a lot to be desired from the caustics on the glossy surface.

Resources

- **Implementation technical report**
  Image comparisons
  [iliyan.com]

- **SmallVCM** – *open-source VCM implementation*
  [SmallVCM.com]

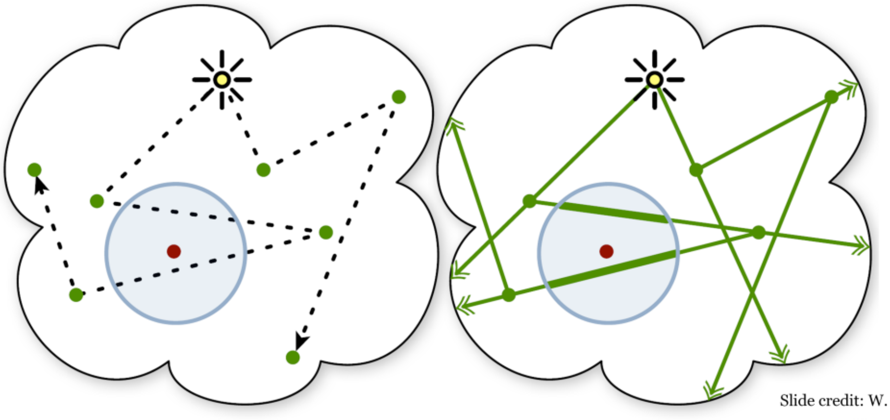
A number of companies have announced VCM integration in the upcoming releases of their commercial renderers.

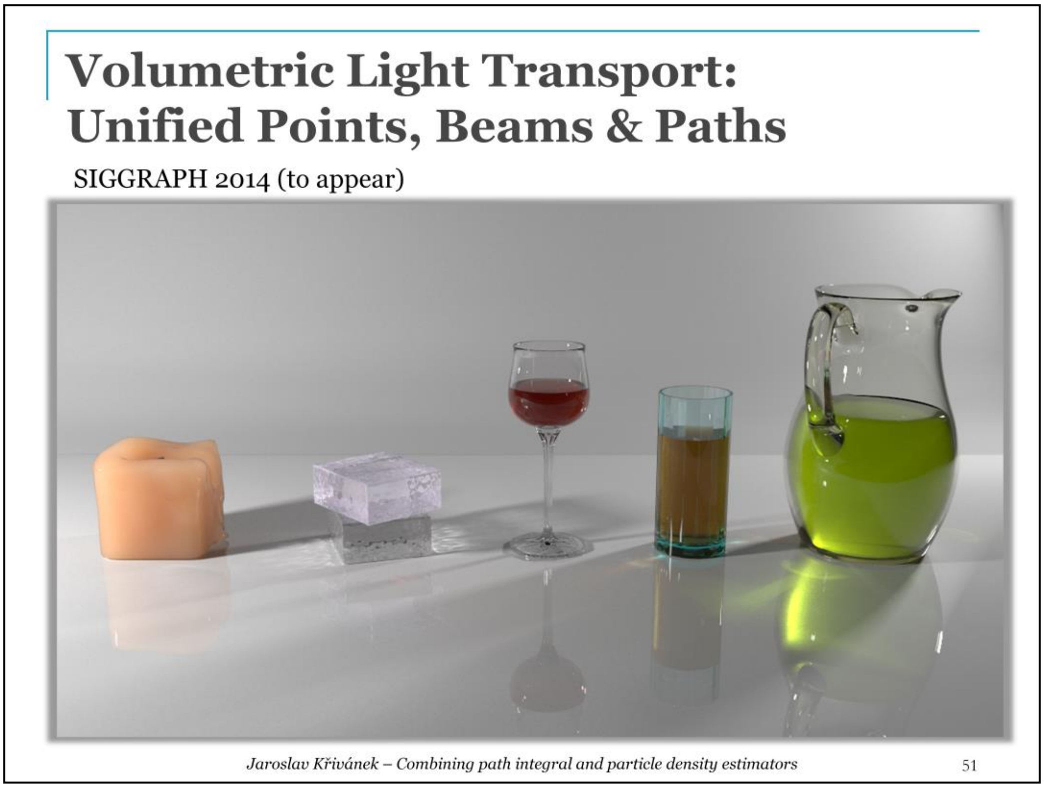
For example, VCM is coming in Pixar's Photorealistic RenderMan v19, …

EXTENSION TO VOLUMETRIC SCATTERING

Volumetric Light Transport

- "Photon beams" [Jarosz et al. 2011]





• We have recently extended VCM to participating media and this work just got accepted to SIGGRAPH.

• And again, understanding all the existing photon point and photon beams estimators in the path integral framework proved to be essential.

References

- *Georgiev et al.,* "Light Transport Simulation with Vertex Connection and Merging", SIGGRAPH Asia 2012
- *Hachisuka et al.* "A Path Space Extension for Robust Light Transport Simulation" , SIGGRAPH Asia 2012
- *Křivánek et al.* "Unifying Points, Beams, and Paths in Volumetric Light Transport Simulation" , SIGGRAPH 2014 (to appear)

Acknowledgements

- **Ondra Karlík**
- **Czech Science Foundation**
  - grant no. P202-13-26189S

THANK YOU!

Questions?

Jaroslav Křivánek

Combining path integral and particle density estimators in light transport simulation

Computer Graphics Charles University

[cgg.mff.cuni.cz/~jaroslav]