CHARLES UNIVERSITY PRAGUE

faculty of mathematics and physics

On-line Learning of Parametric Mixture Models for Light Transport Simulation

Jiří Vorba[1] Ondřej Karlík[1] Martin Šik[1]
Tobias Ritschel[2] Jaroslav Křivánek[1]

[1]Charles University in Prague
[2]MPI Informatik Saarbrücken

Good morning. Thank you for the introduction.

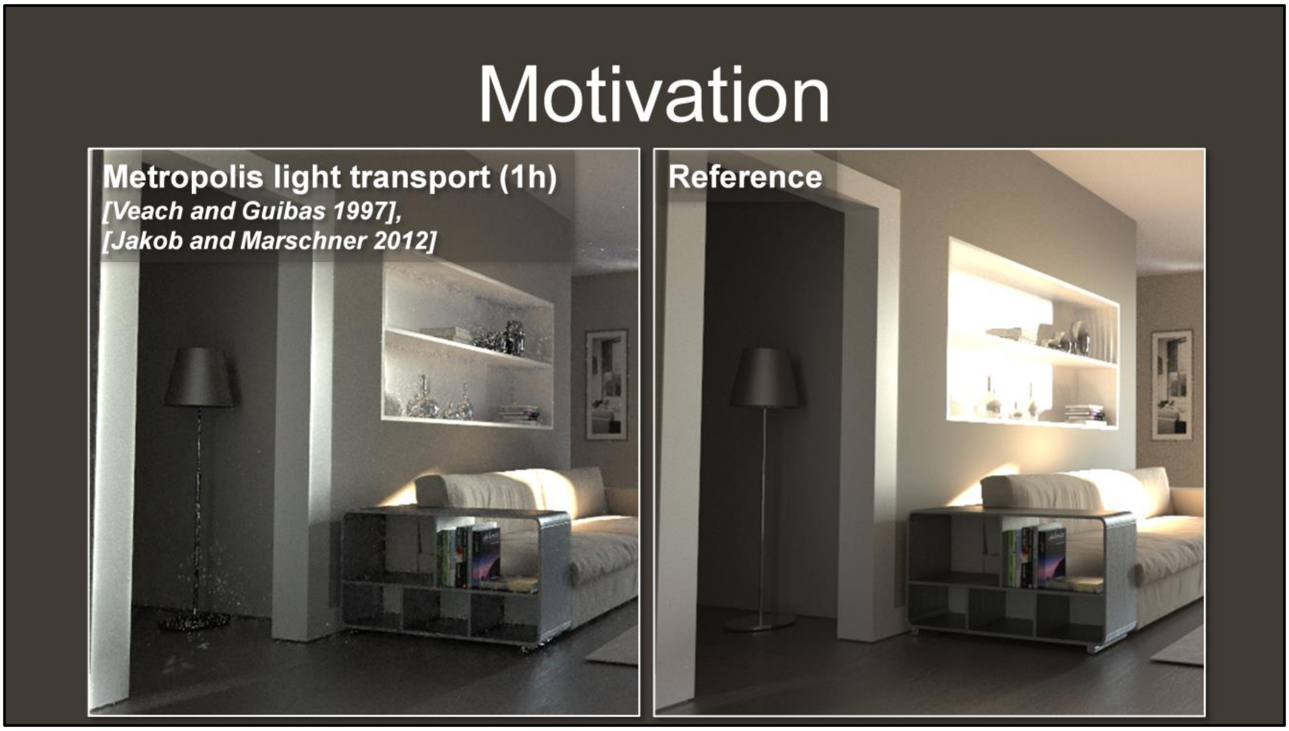
I am going to present a new method …

… that aims at rendering scenes with difficult visibility.

Such scenes comprise complex geometry and light reaches the camera only after several interactions with surfaces.

In this example, the sunlight reaches the interior through a window and the small gap between the curtains and undergoes many interactions before it enters the dim room on the left.

Now I will demonstrate, that rendering of such scenes remains an open problem.



It is generally considered that Metropolis light transport algorithm is a remedy for this problem.

However, as you can see in the image, the result after 1h is very far from the reference.

Recently, Vertex Connection and Merging algorithm was introduced.

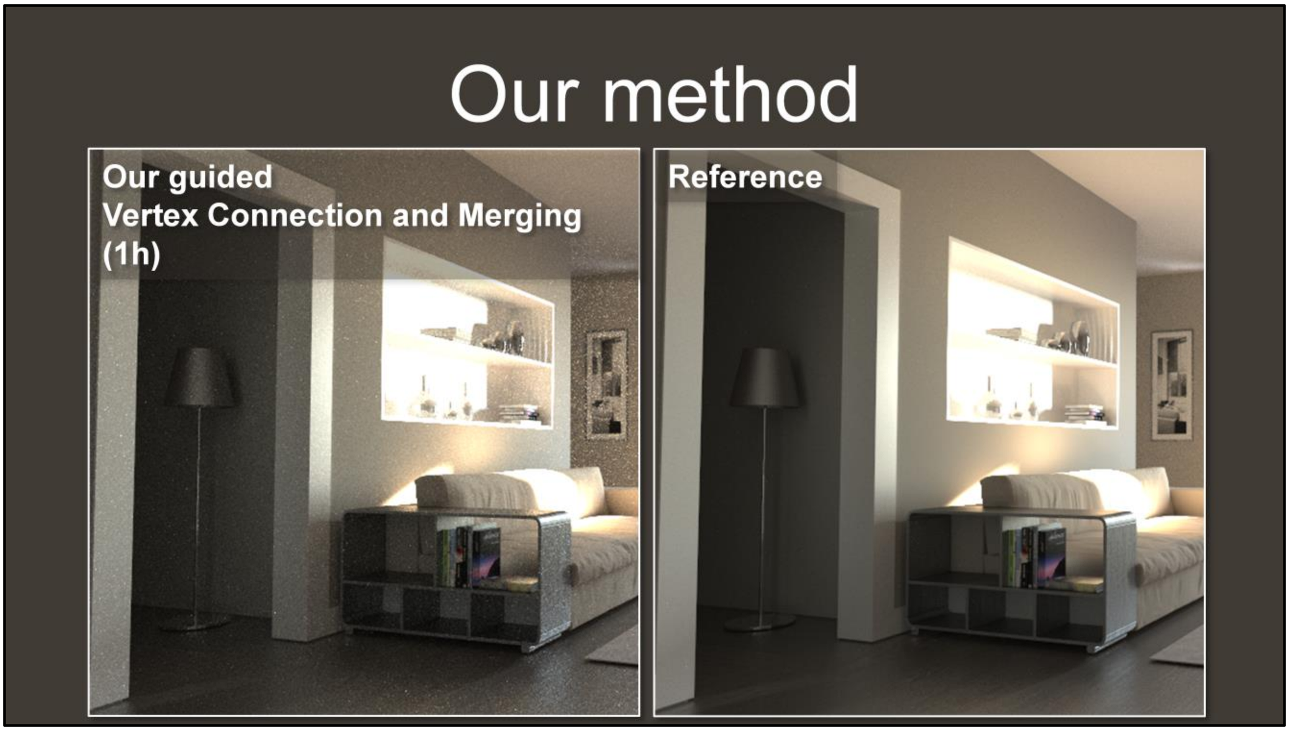
This algorithm can efficiently handle high variety of lighting conditions thanks to connecting and merging of many light paths.

However, as you can see, the difficult visibility remains a problem.

There are simply too few light paths that could be connected or merged. The result is a noisy image.

We aim to solve this problem by guiding of the path-sampling process.

Now, I will show you the result of our guiding applied to VCM.



As you can see, guiding of path-sampling process substantially reduced the noise.

Poor importance sampling

$$L_{out} = \int_{\Omega} L_{in}(\omega) \cdot f_r(\omega) \cos\theta \mathrm{d}\omega$$

6

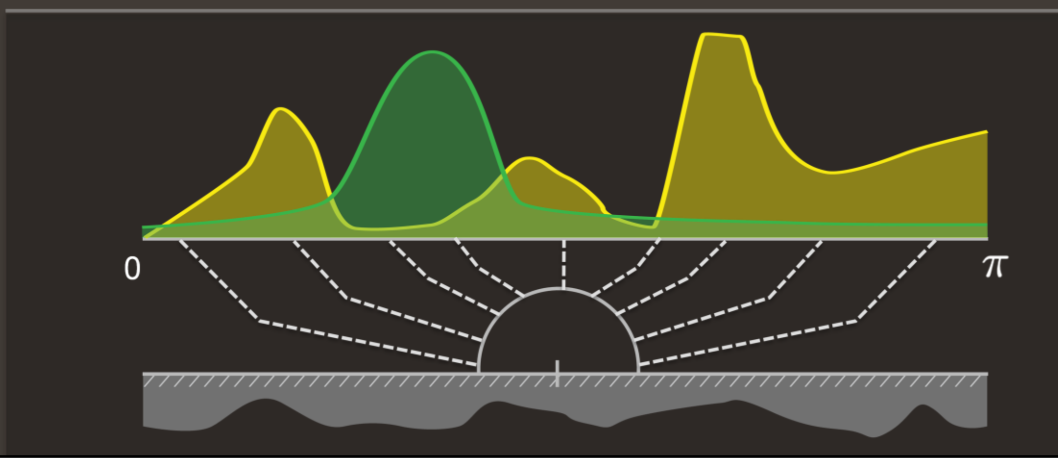
To understand why such scenes are so difficult for rendering, we need to look at the rendering equation.

The product of incoming radiance L_in and BRDF (which is highlighted in green) is integrated over the hemisphere of directions.

State-of-the-art MC-based algorithms strive for importance sampling of the BRDF only and they often ignore the incoming radiance term.

Poor importance sampling

$$L_{out} = \int_{\Omega} L_{in}(\omega) \cdot f_r(\omega) \cos\theta \mathrm{d}\omega$$



7

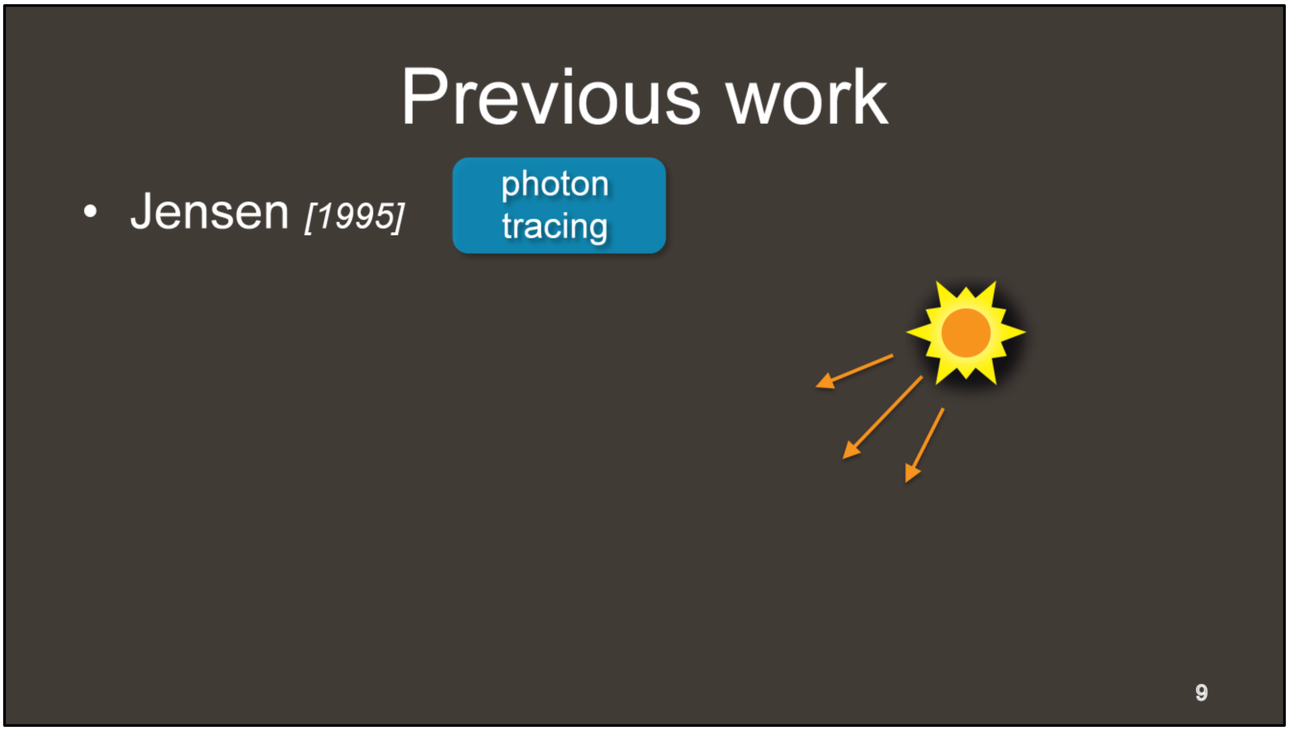
But this term can be the source of high variance due to possible caustic lighting and highlights in the scene.

And it also includes the visibility. So we could say that ignoring the radiance term makes the sampling blind to the rendered scene and results in waste of samples.

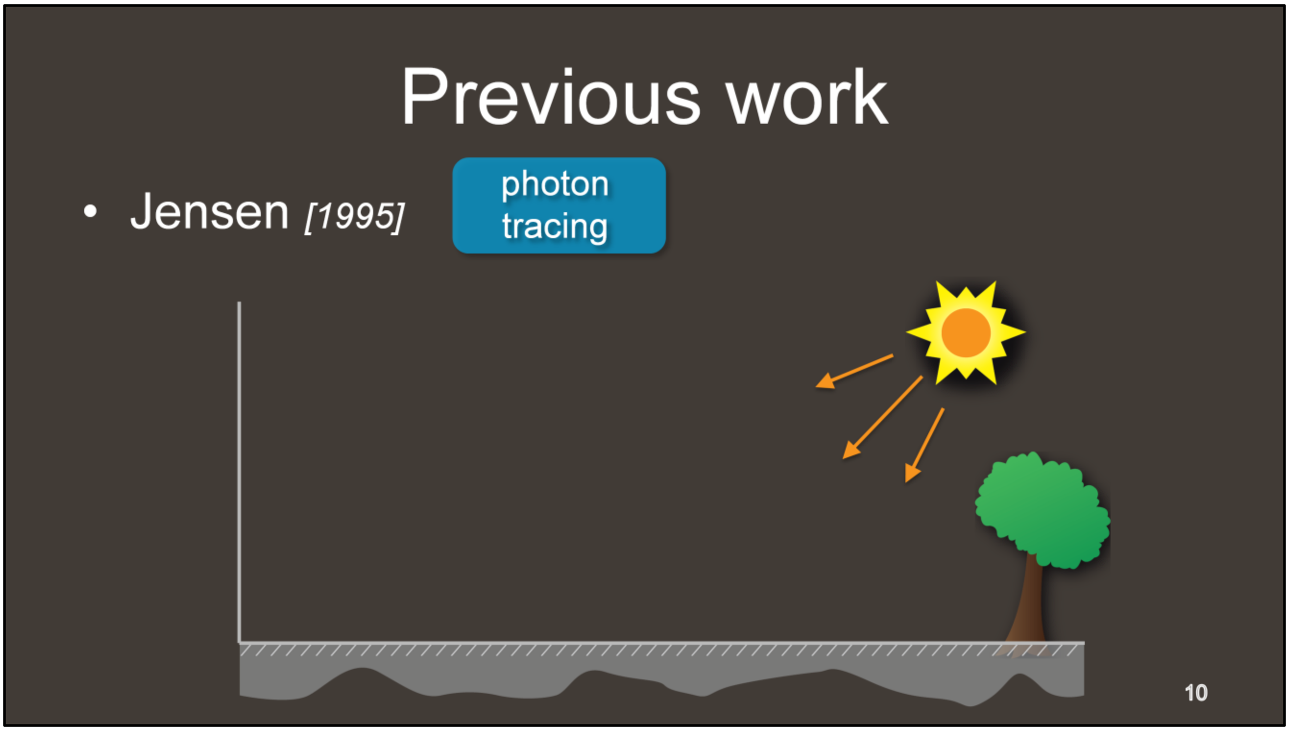
Previous work

8

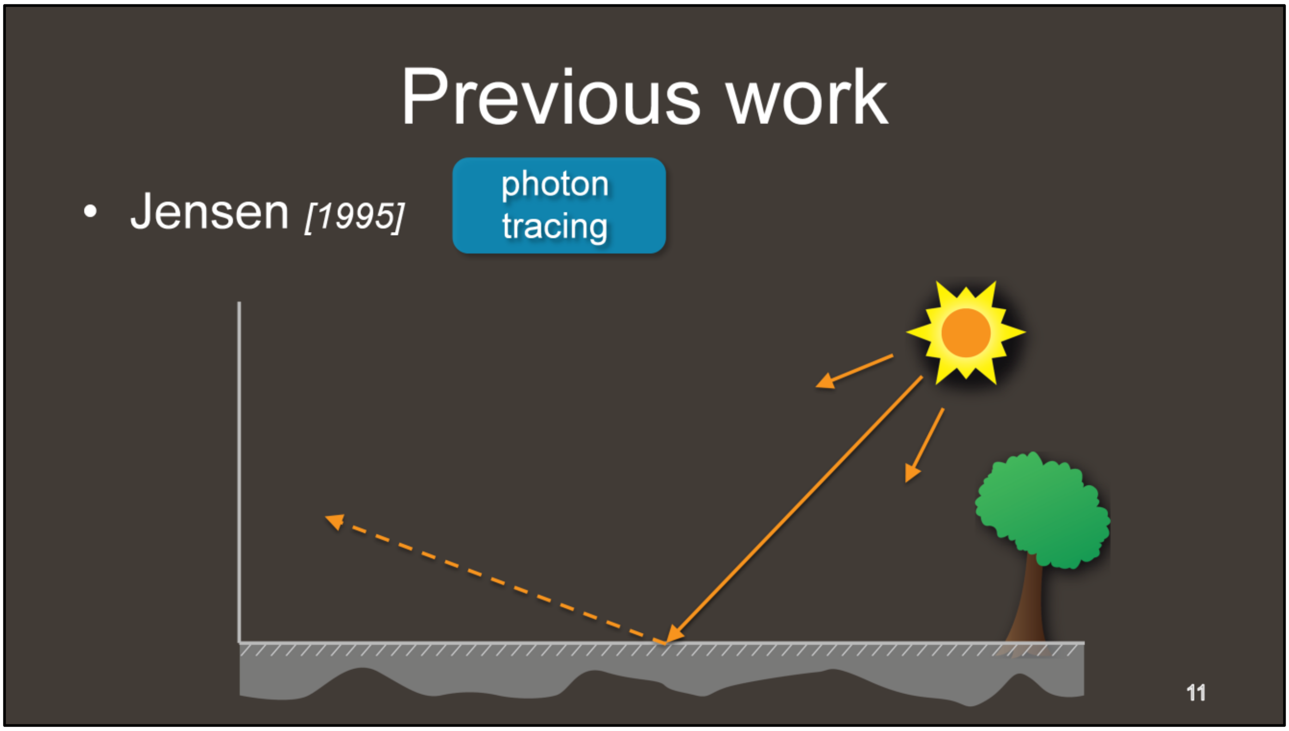
People tried to address this inefficient sampling in the past.



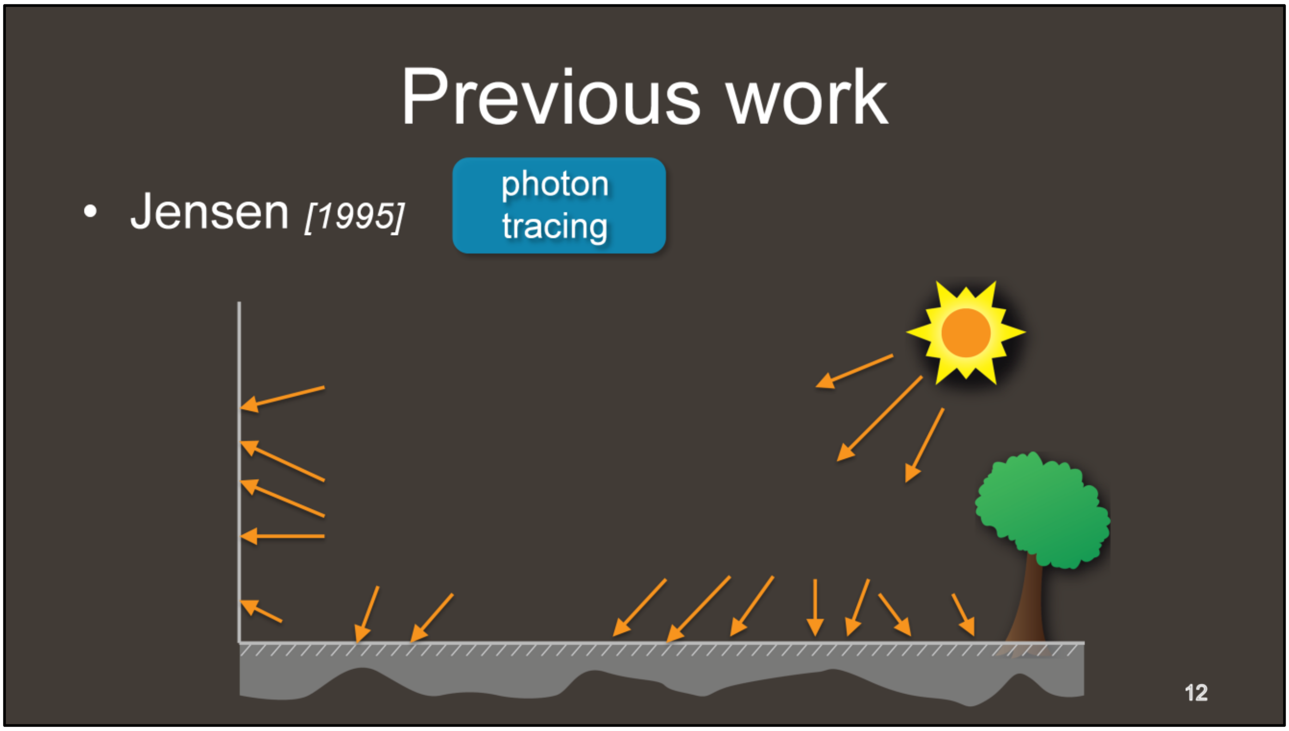
Jensen [1995], in his previous work, uses light particles – that is photons – to reconstruct the distribution of incoming radiance (Li).



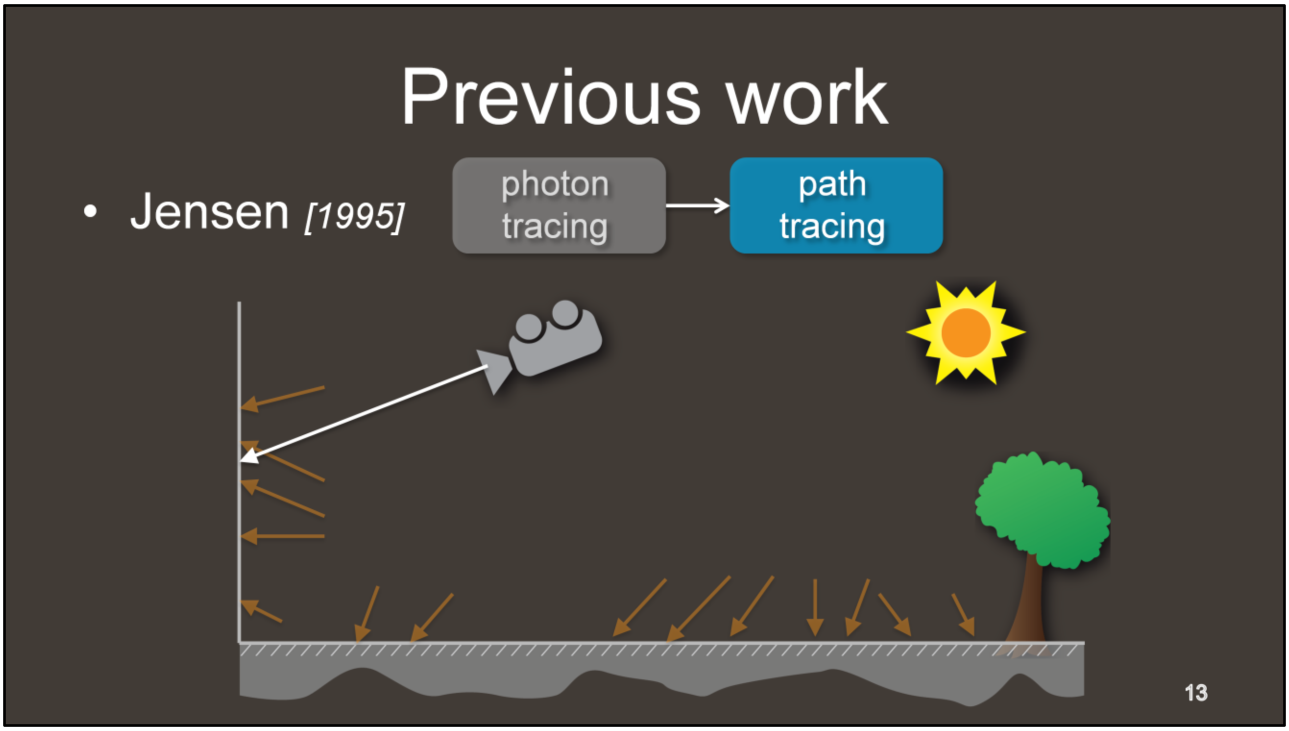
They trace the photons in a preprocessing phase…



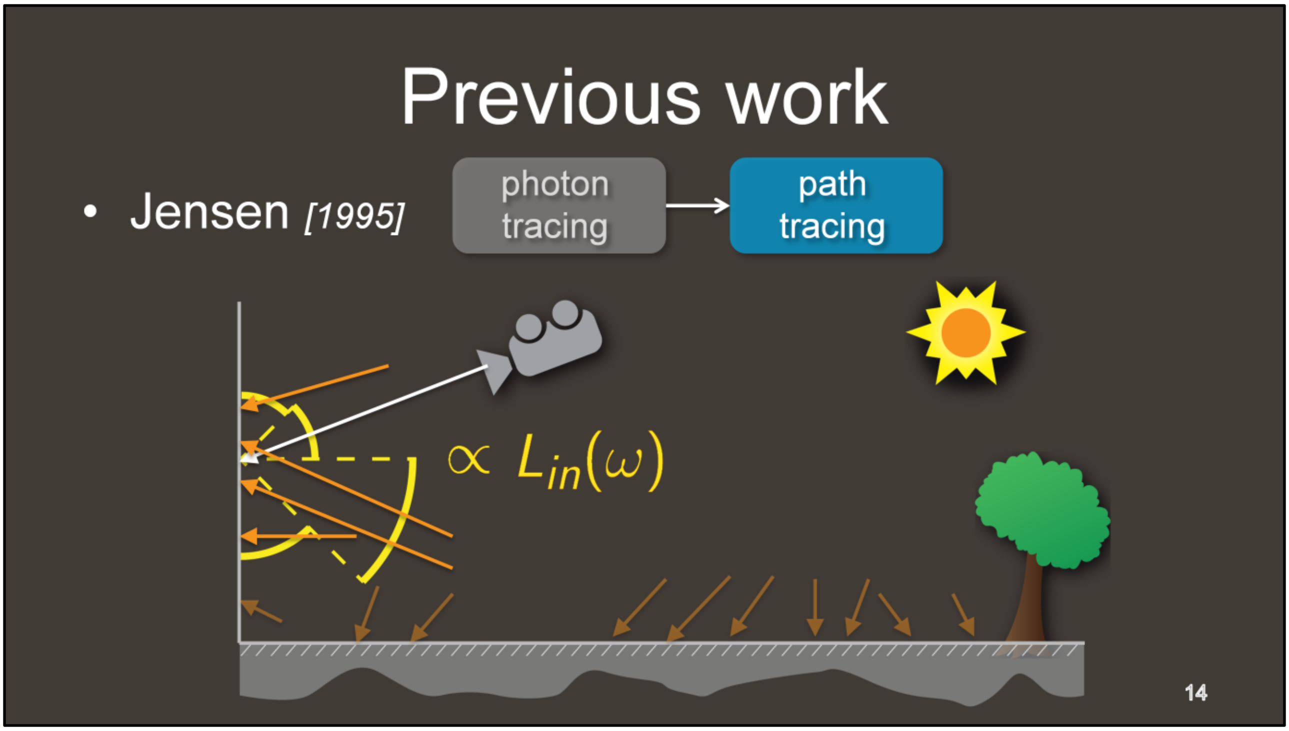
They trace the photons in a preprocessing phase…



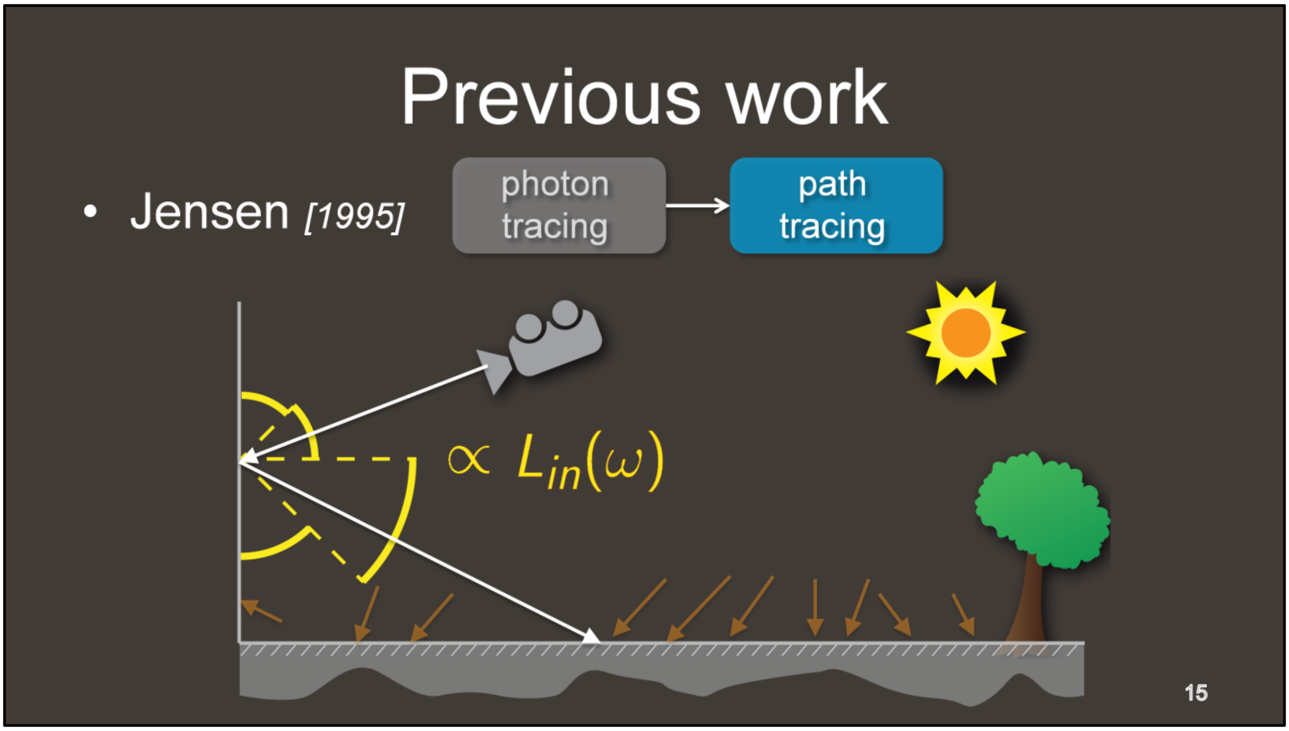
and store them on surfaces for later use in path-tracing.



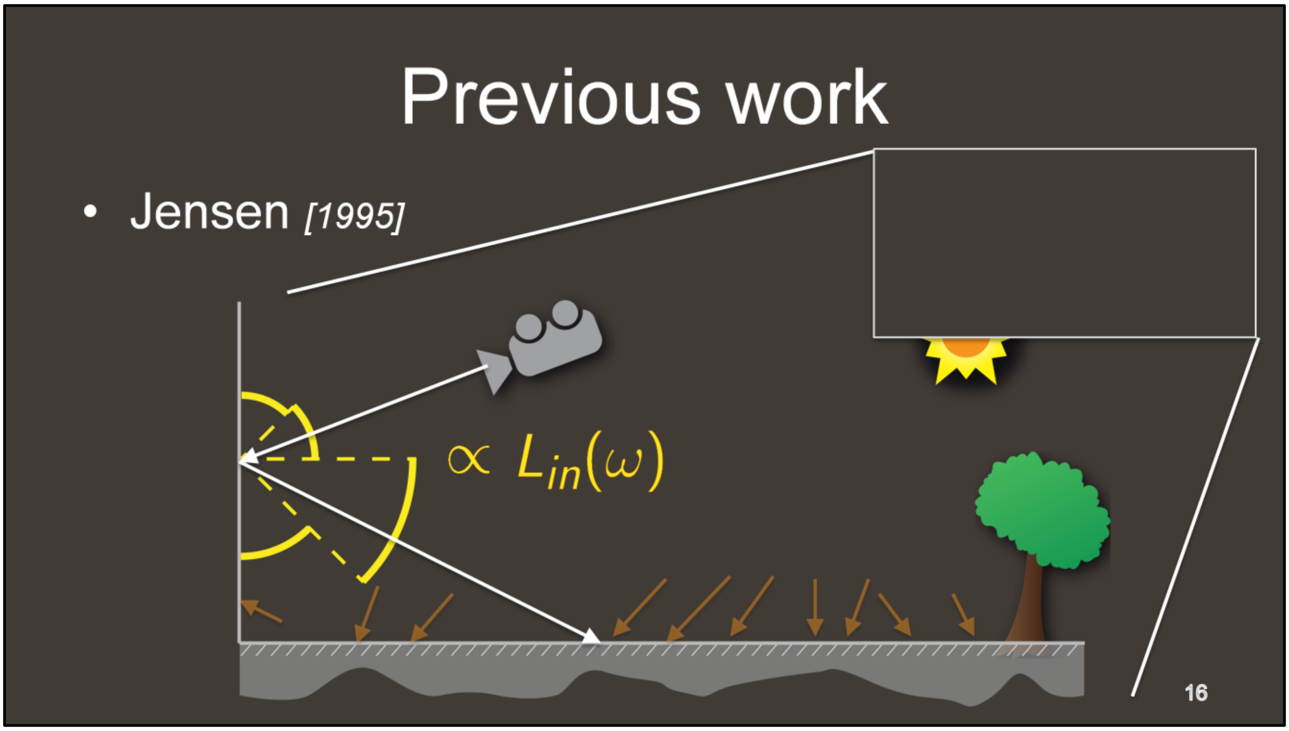
During the path-tracing, …



…, they reconstruct the directional distributions of radiance from nearest photons…



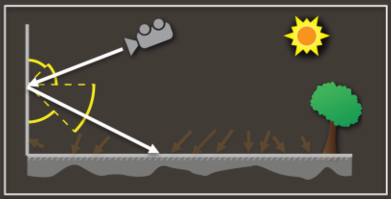
and use the distributions for sampling of reflected directions. Note that contrary to photon mapping, the method produces an unbiased image.



and use the distributions for sampling of reflected directions. Note that contrary to photon mapping, the method produces an unbiased image.

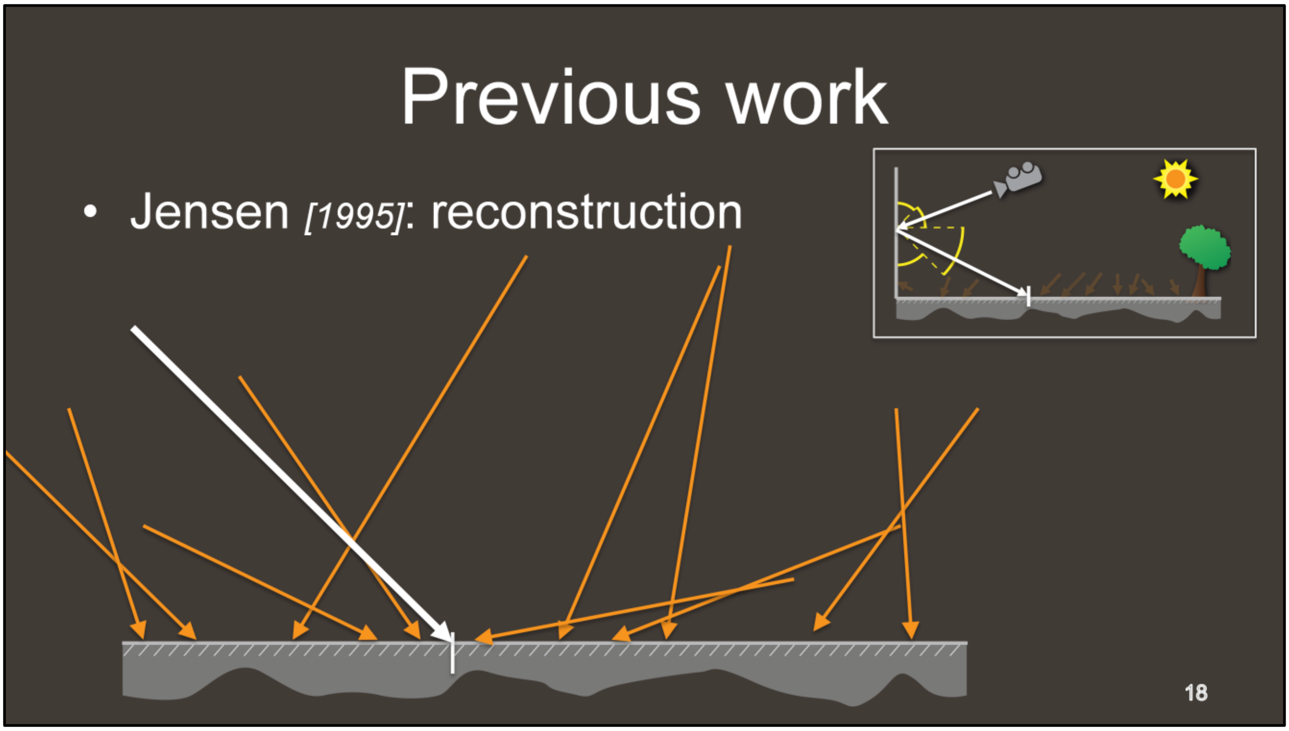
Previous work

- Jensen *[1995]*

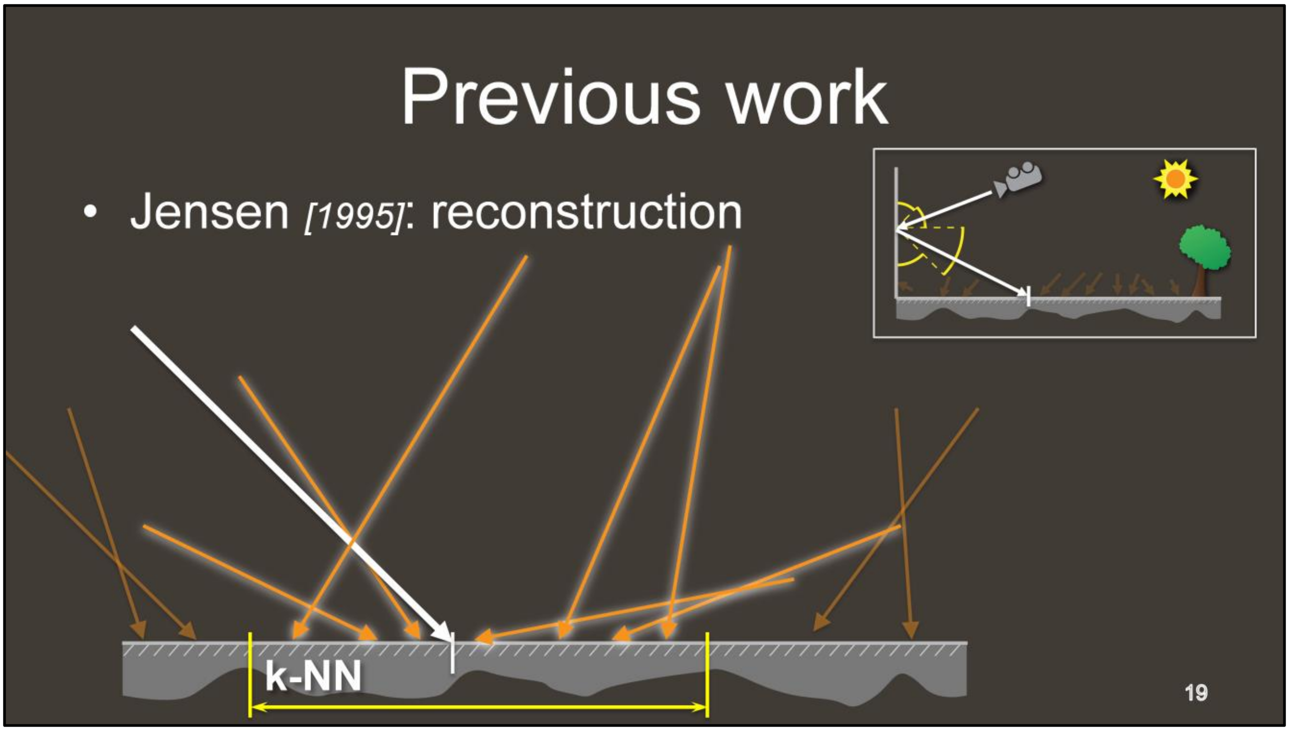


17

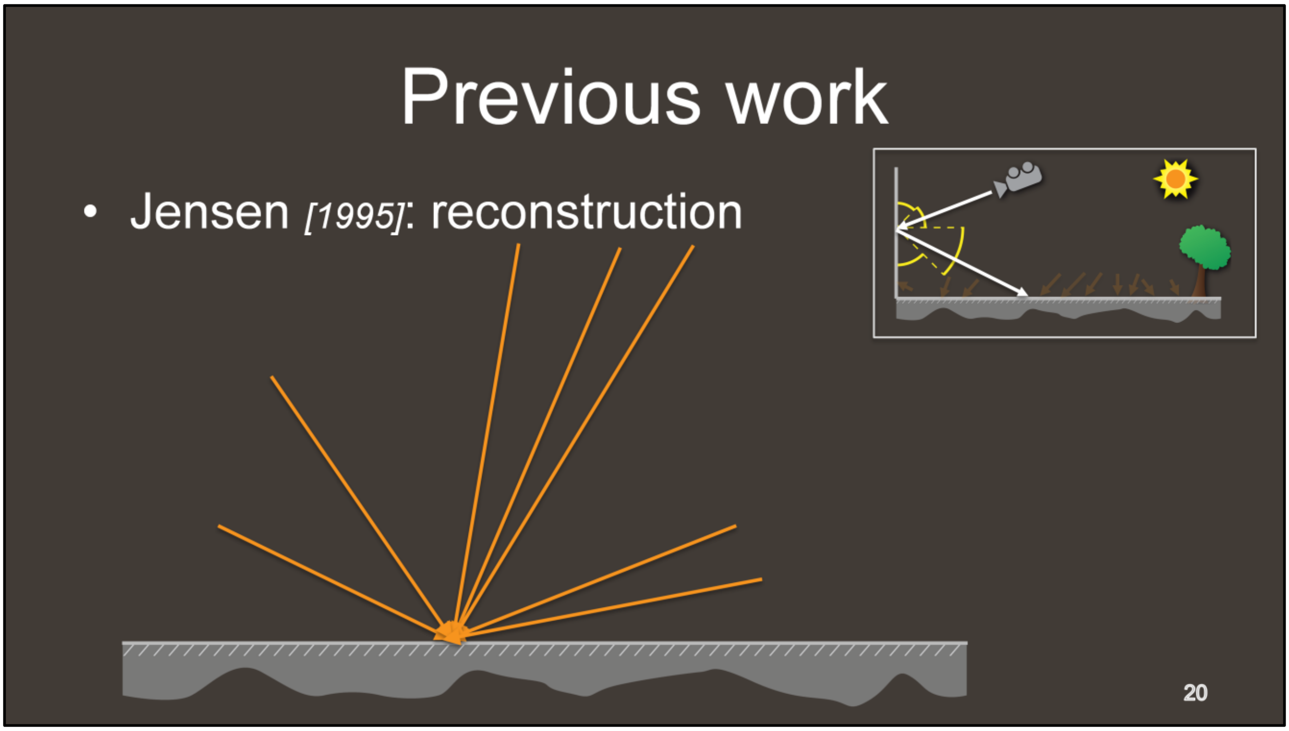
and use the distributions for sampling of reflected directions. Note that contrary to photon mapping, the method produces an unbiased image.



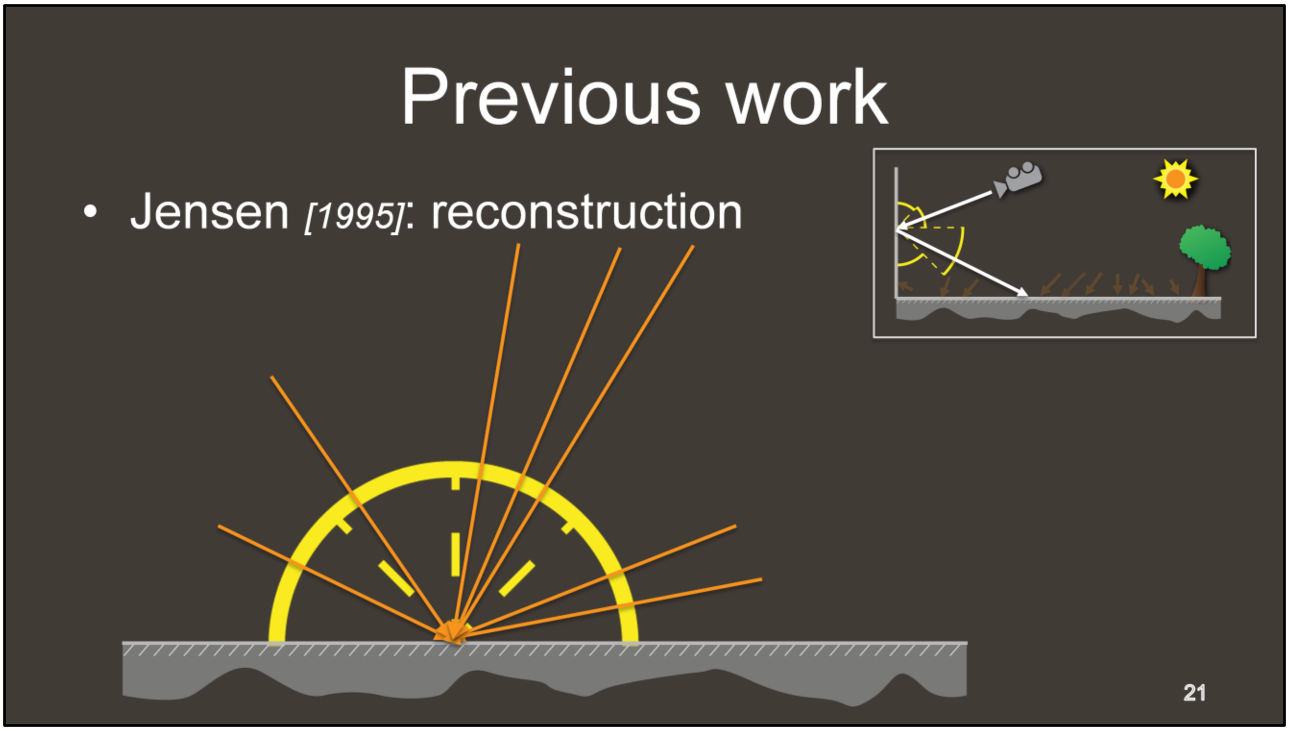
The reconstruction of a distribution starts with a search for nearest photons from the surface intersection.



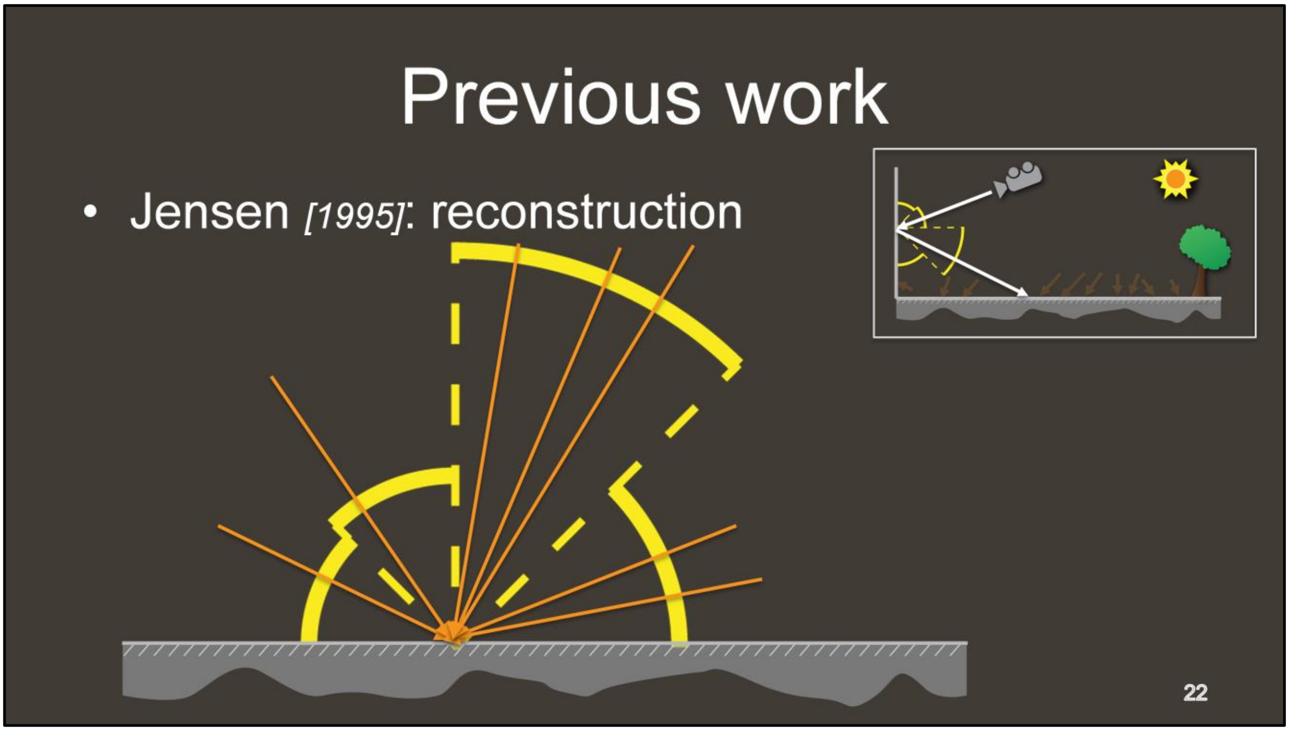
They are interested in a directional distribution so they make the assumption that all particles hit the surface at one point.



To reconstruct the directional probability distribution …



they discretize the hemisphere into equal-sized bins and count the number of particle directions falling into each bin.



The result is a histogram over hemisphere.

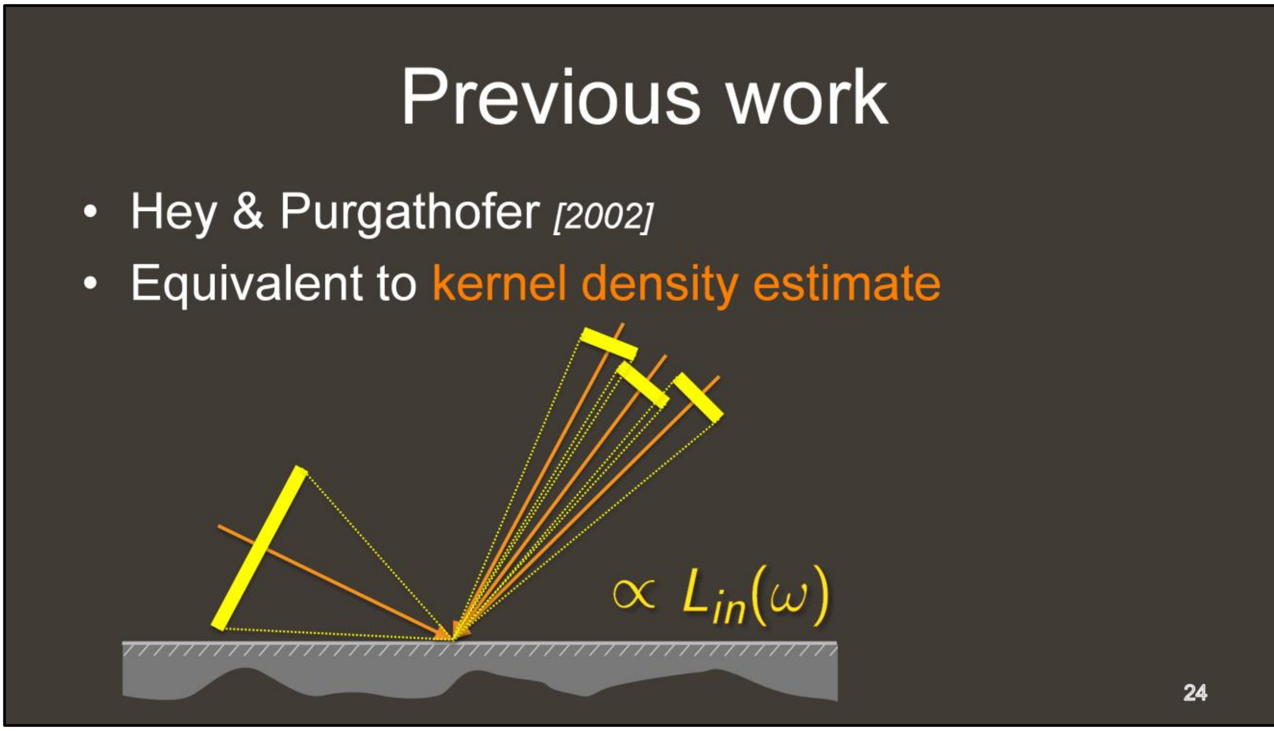
However, histogram density estimate is known to be a poor estimation method prone to over and under estimation.

Previous work

- Hey & Purgathofer *[2002]*

23

Hey & Purgathofer [2002] alleviate the problem by using a different model to reconstruct and represent the directional distributions.



They represent the directional pdf by overlapping cones with adaptive width.

The cones are centered around incoming photon directions
and the size of the cones is inversely proportional to directional density of photons.
[CLICK]
This corresponds to a kernel density estimate with adaptive kernel size so the method allows to have more refined pdf than Jensen's histograms.

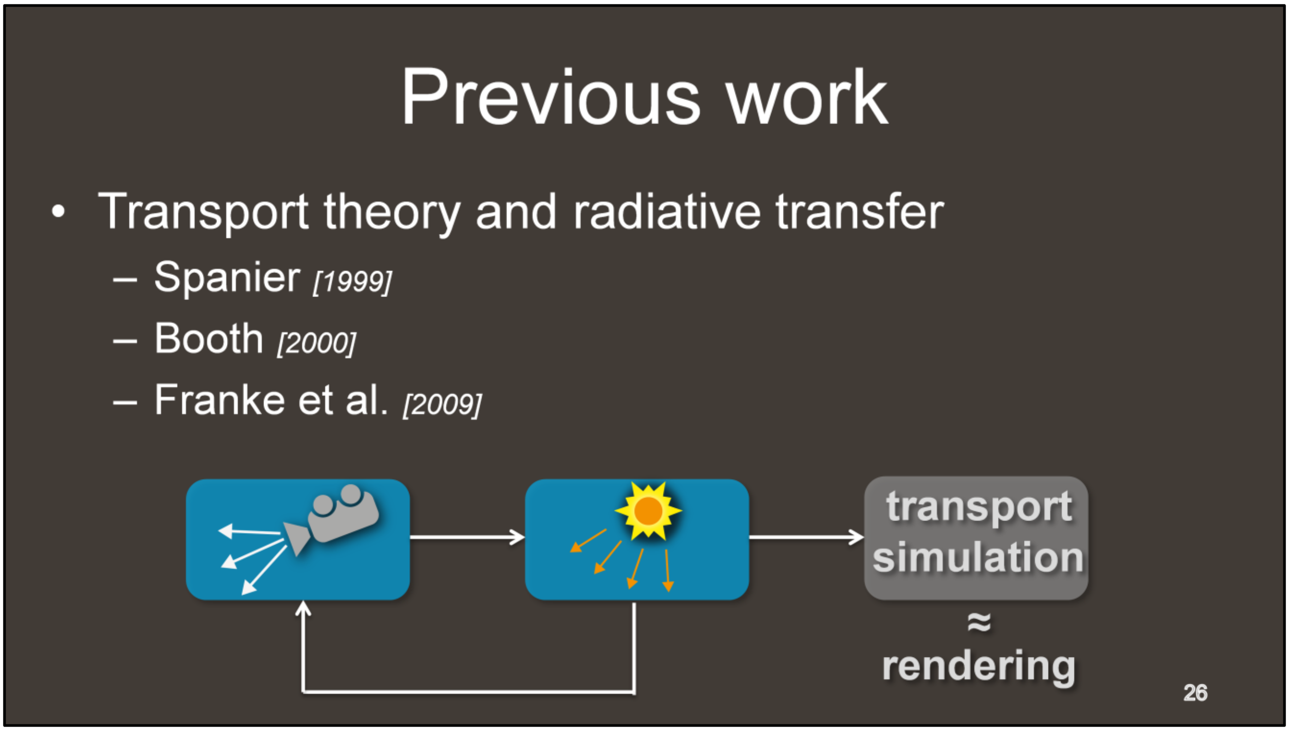
Previous work

- Transport theory and radiative transfer
  - Spanier *[1999]*
  - Booth *[2000]*
  - Franke et al. *[2009]*

25

The light transport can be considered a subfield to radiative transfer and transport theory of neutral particles.

I will point out works of Spanier [1999], Booth [2000] and Franke et al. [2009].



Similarly to their work, we also interleave photon and particle tracing steps to refine our distributions.

Limitations of previous work

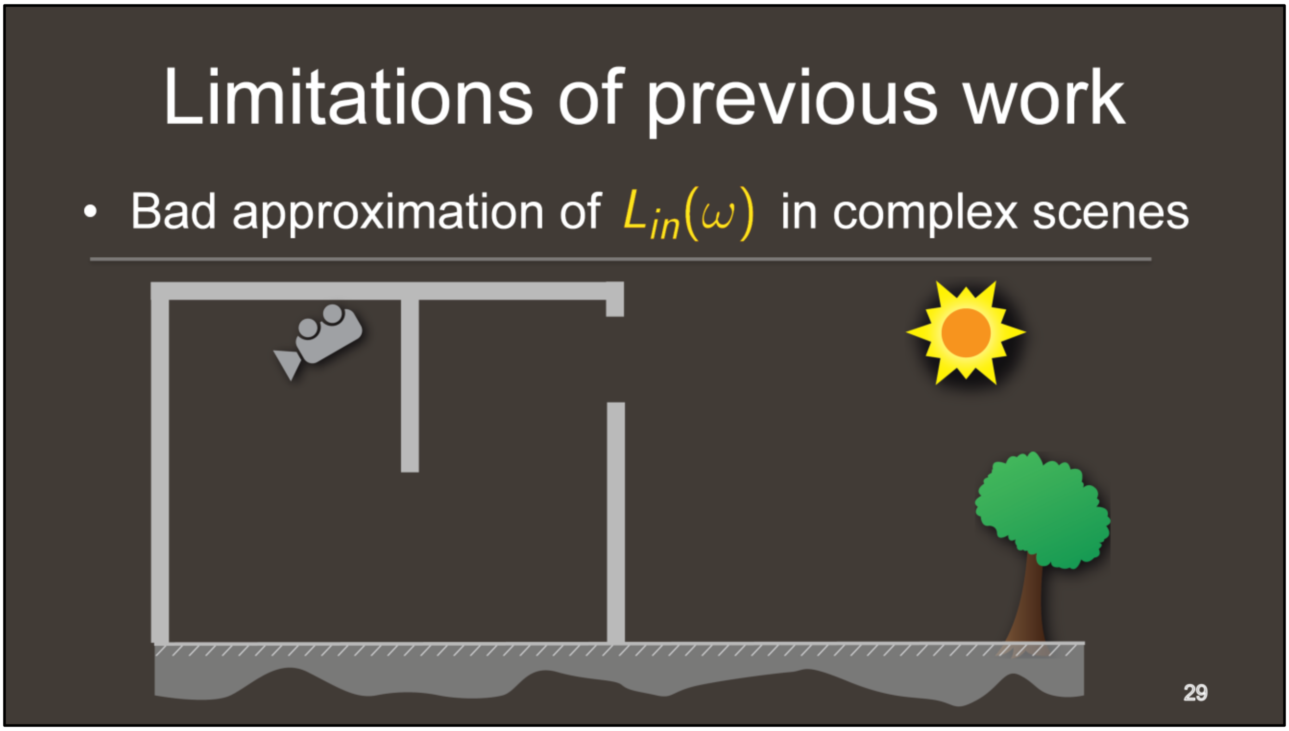
27

Now I will explain the main issue connected with previous methods.

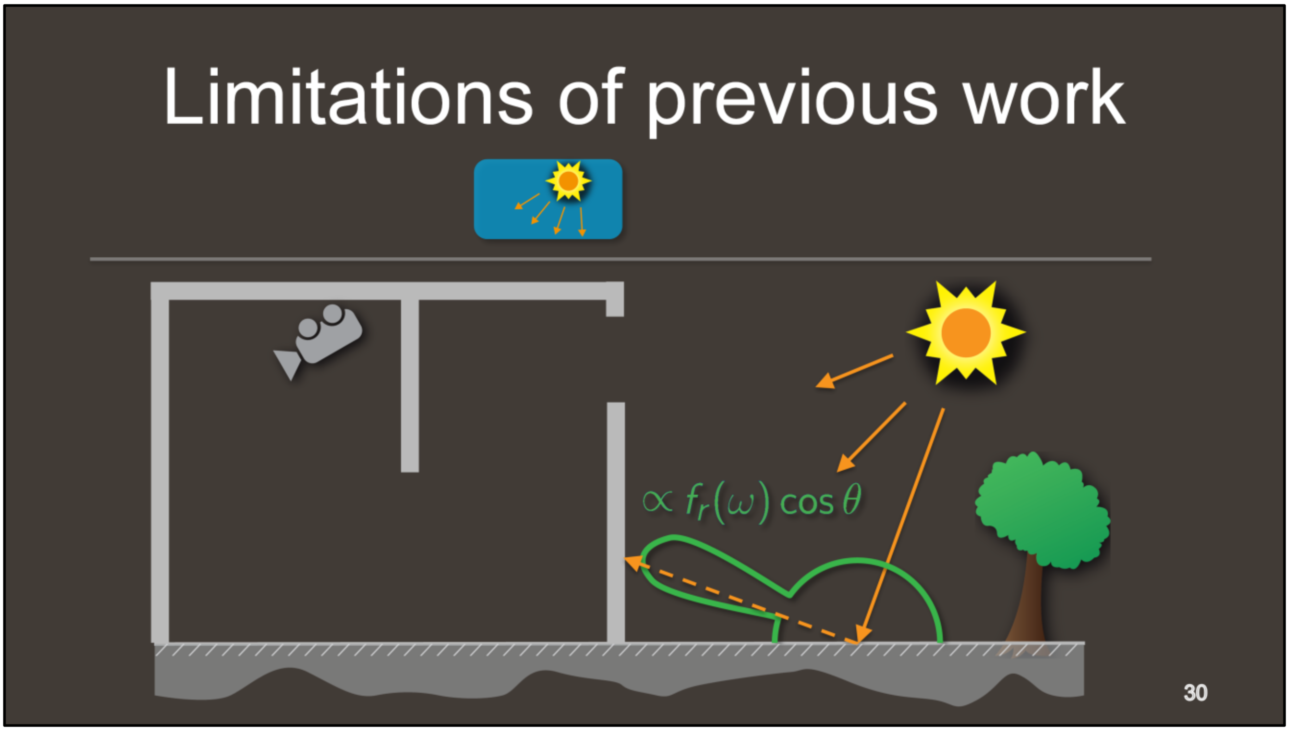
Limitations of previous work

- Bad approximation of $L_{in}(\omega)$ in complex scenes

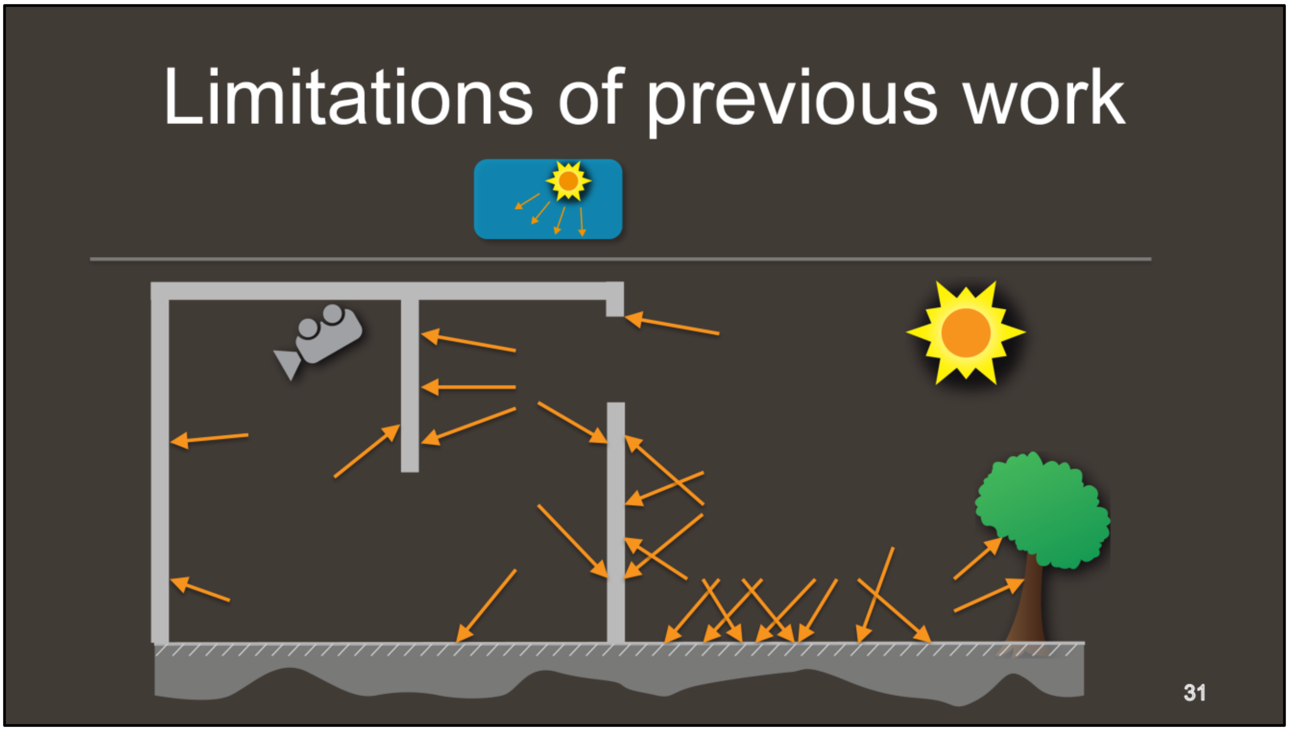
Although they approximate the radiance term for importance sampling, they struggle in a wide range of scenes.



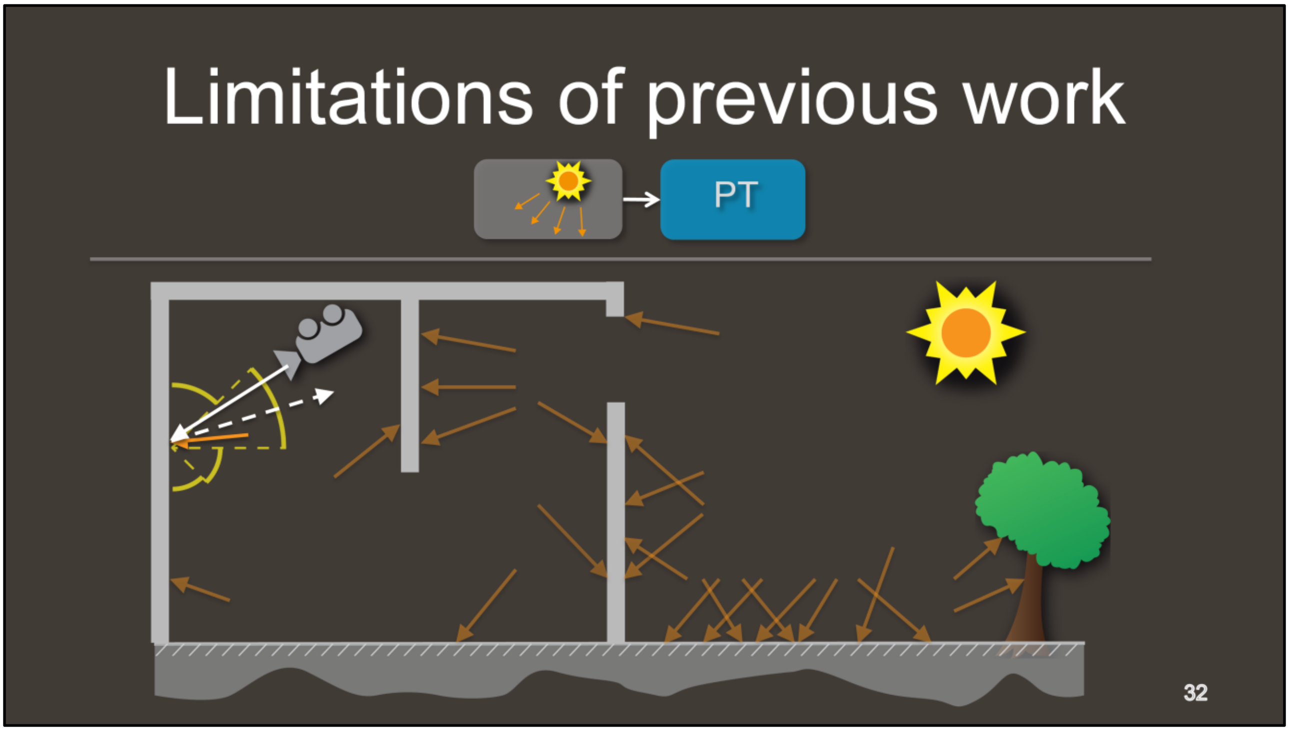
Let’s consider an interior scene where the camera is in a dim room and light is entering from the outside.



As the particles are scattered through BRDF at the beginning, it is not possible to obtain a sufficient number of particles everywhere for good reconstruction.

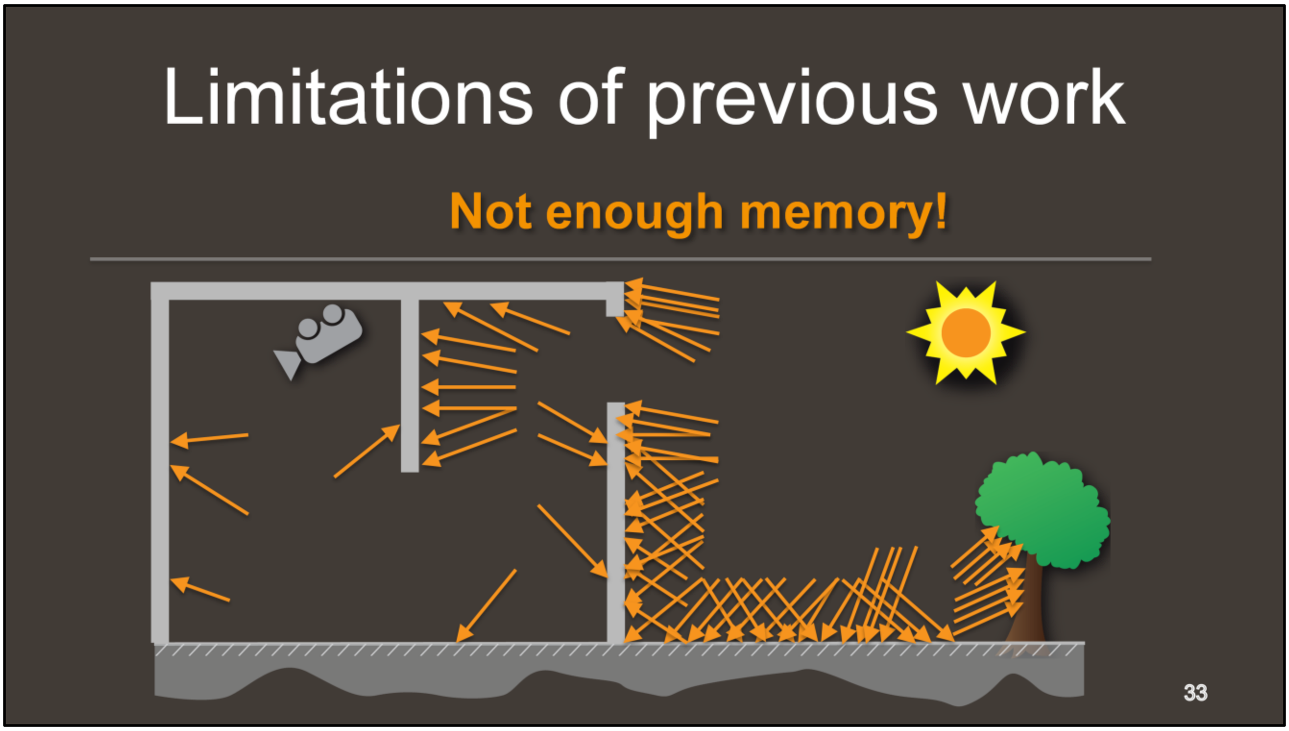


Note that, in the illustration, we have only one photon in front of the camera.



Using such a poor reconstruction in path-tracing would even increase the noise level in the rendered image.

What we need is to get many more particles into the dim room, …



... but the problem is that we are limited by memory where all the particles need to be stored (including the particles outside of the room, where we do not need them at all).

Our solution

34

Now I will present our solution to this problem that forms the cornerstone of our method.

Our solution

GMM

- The Gaussian mixture model (GMM)

35

We follow the same line of work. But instead of histogram or cones, we use a Gaussian mixture model to represent directional distributions.

This model can be progressively trained from a potentially infinite stream of particles. The important thing is that the particles do not need to be stored in memory.

Our solution

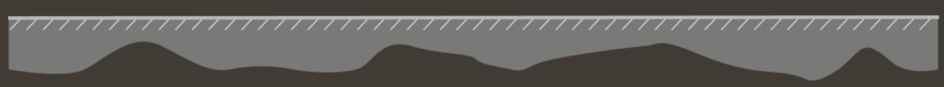
GMM ⇒ **on-line learning** ⇒ **constant memory**

- The Gaussian mixture model (GMM)

37

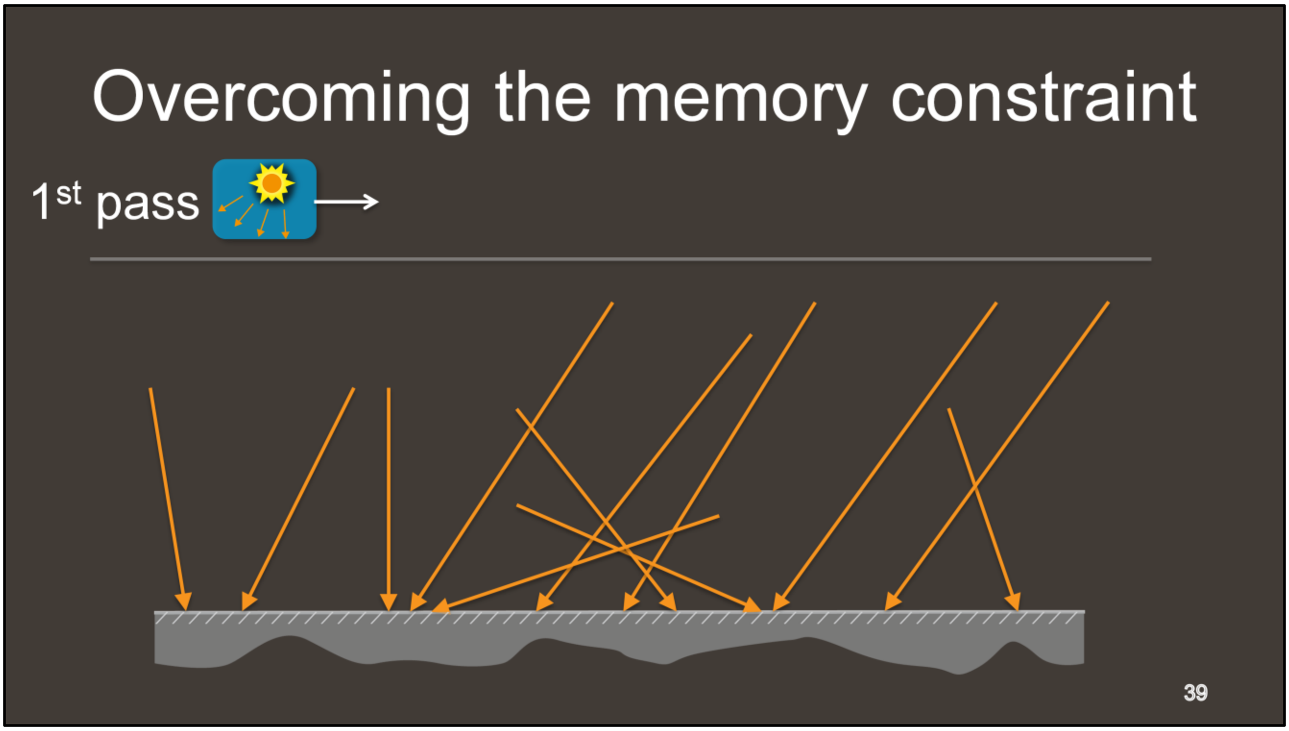
So in this way our on-line learning allows to overcome the memory constraint.

Overcoming the memory constraint

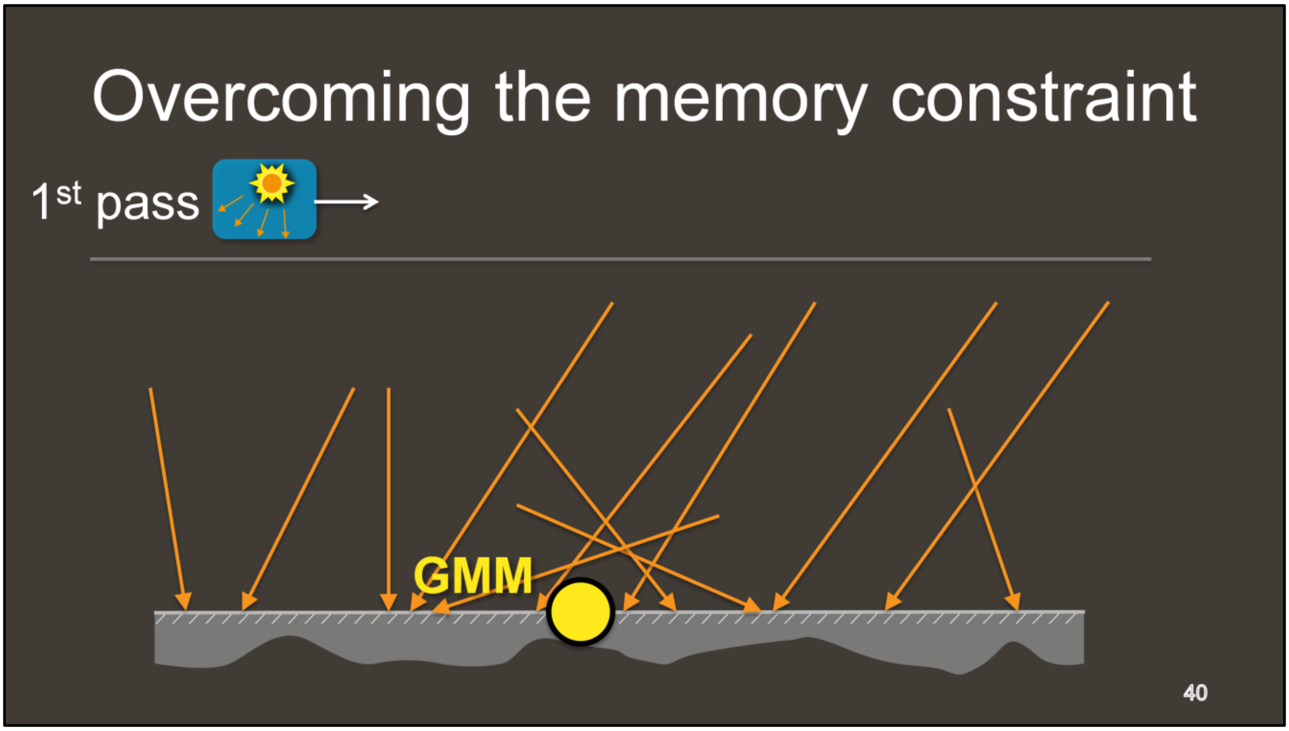


38

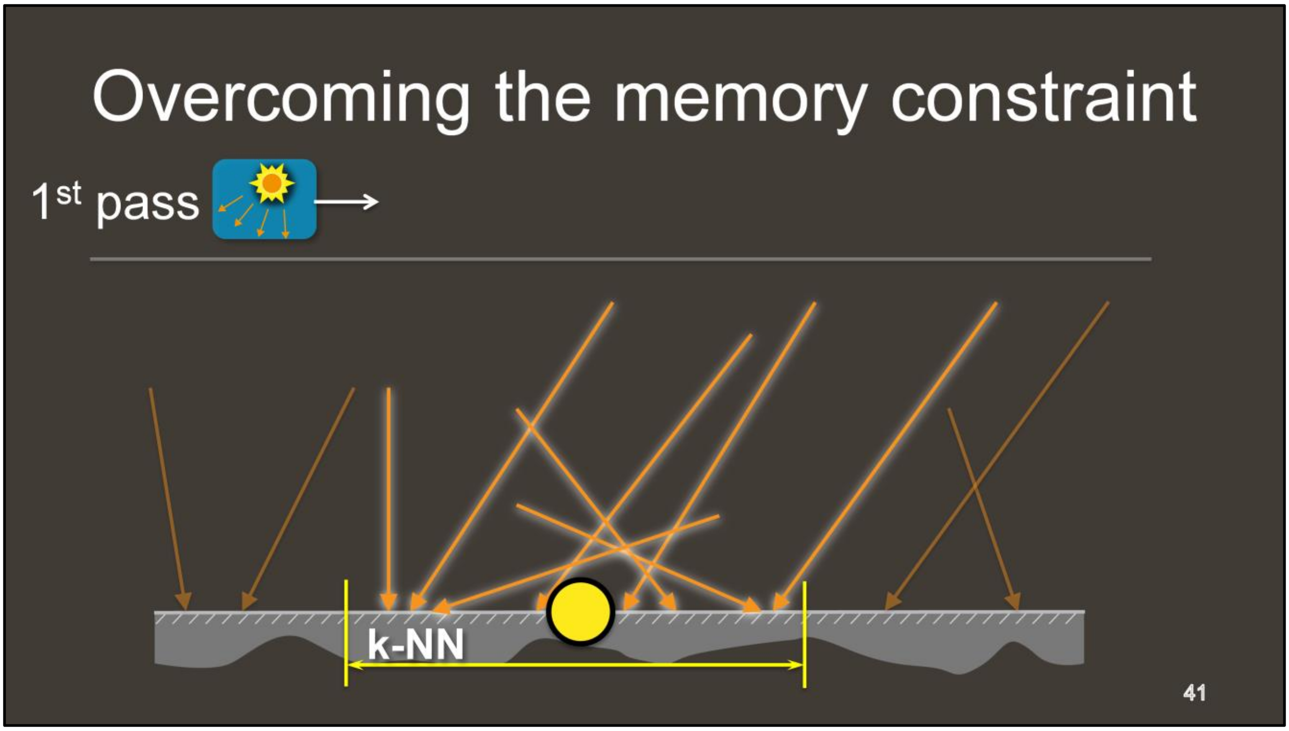
This is how it works.



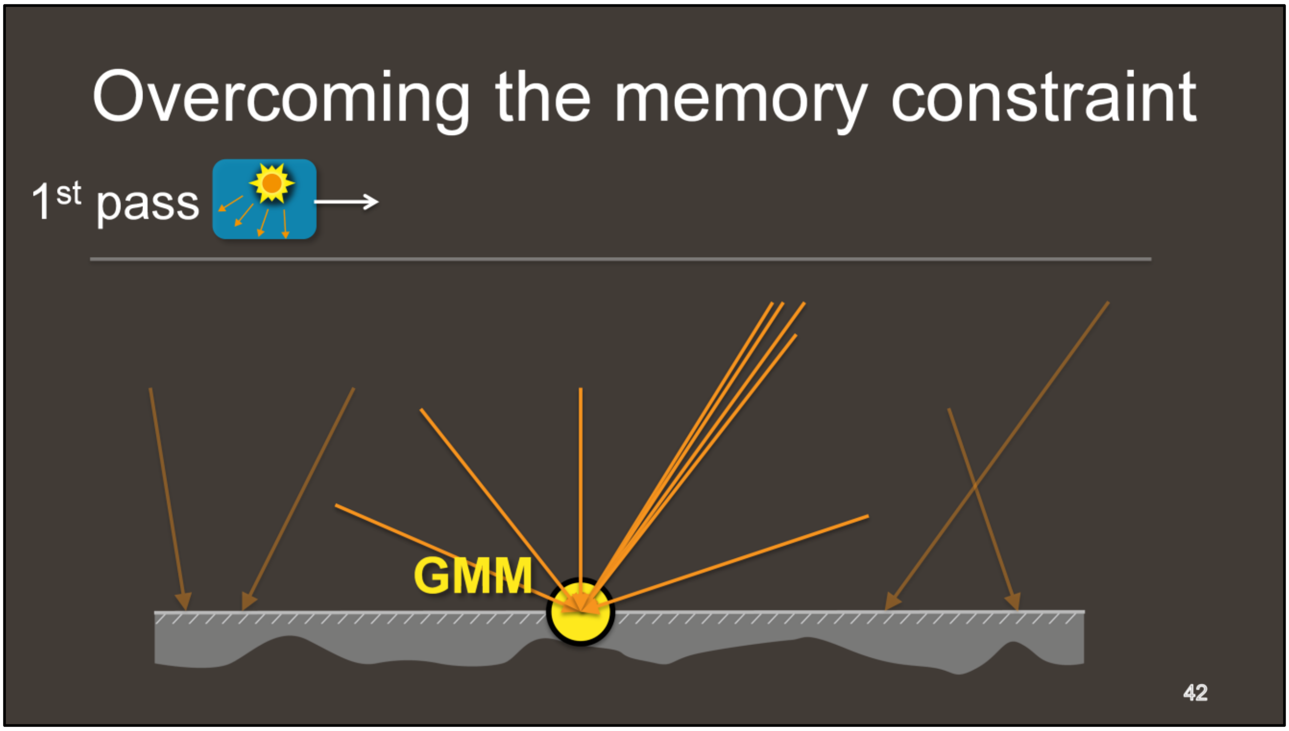
We trace particles in batches that can fit into memory. So we trace a first batch of photons, …

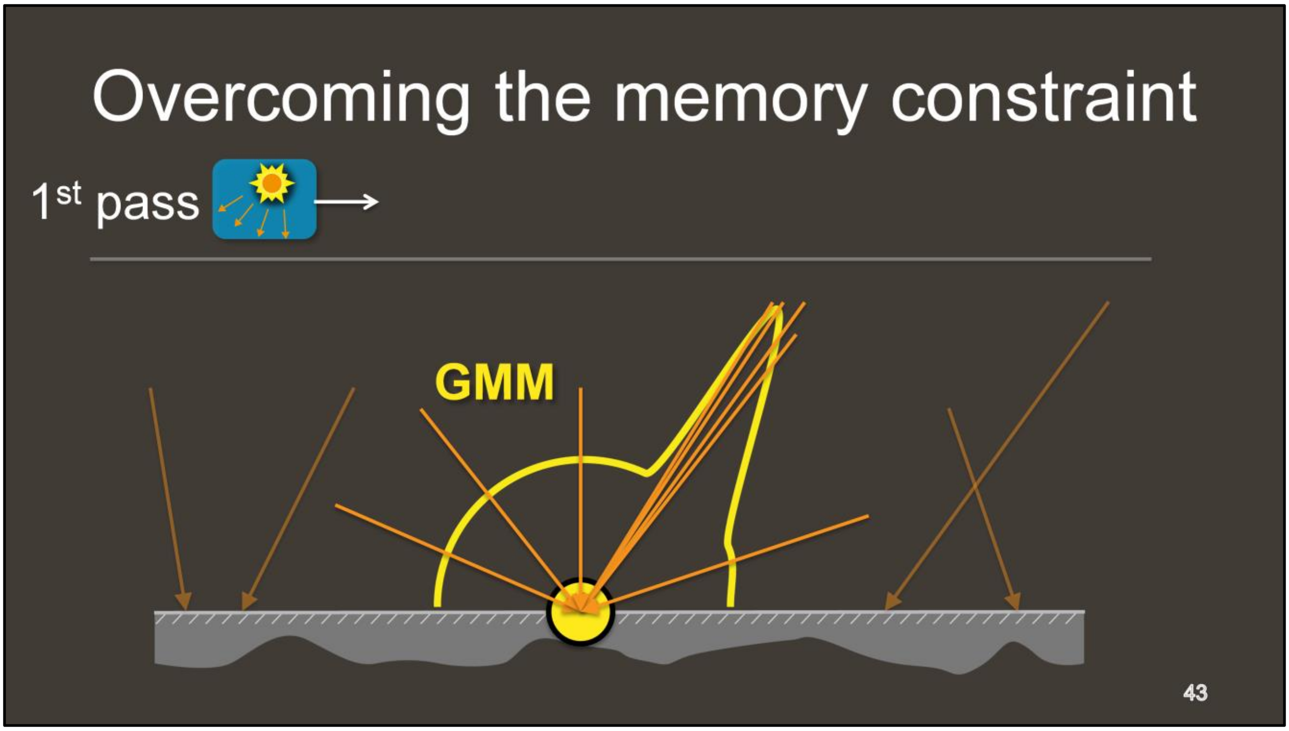


… we create a distribution on a scene surface…

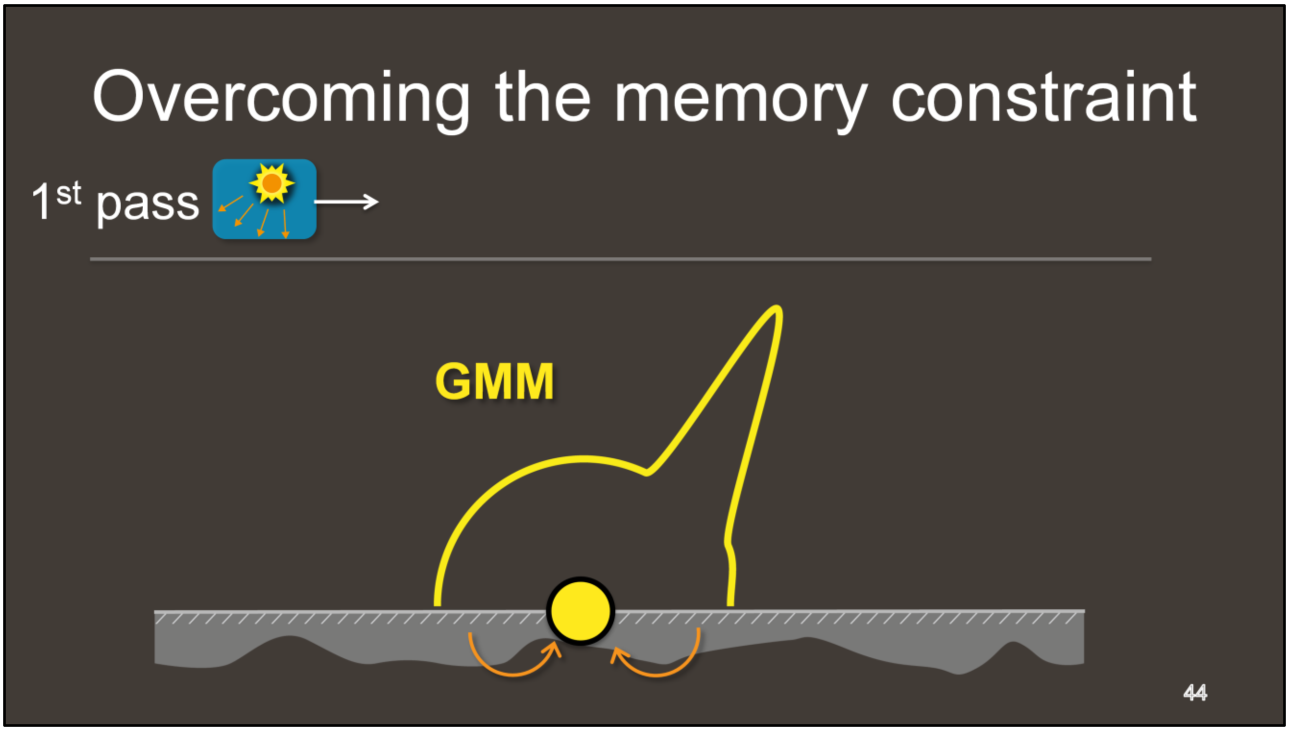


and we use the nearest photons for its initial training.

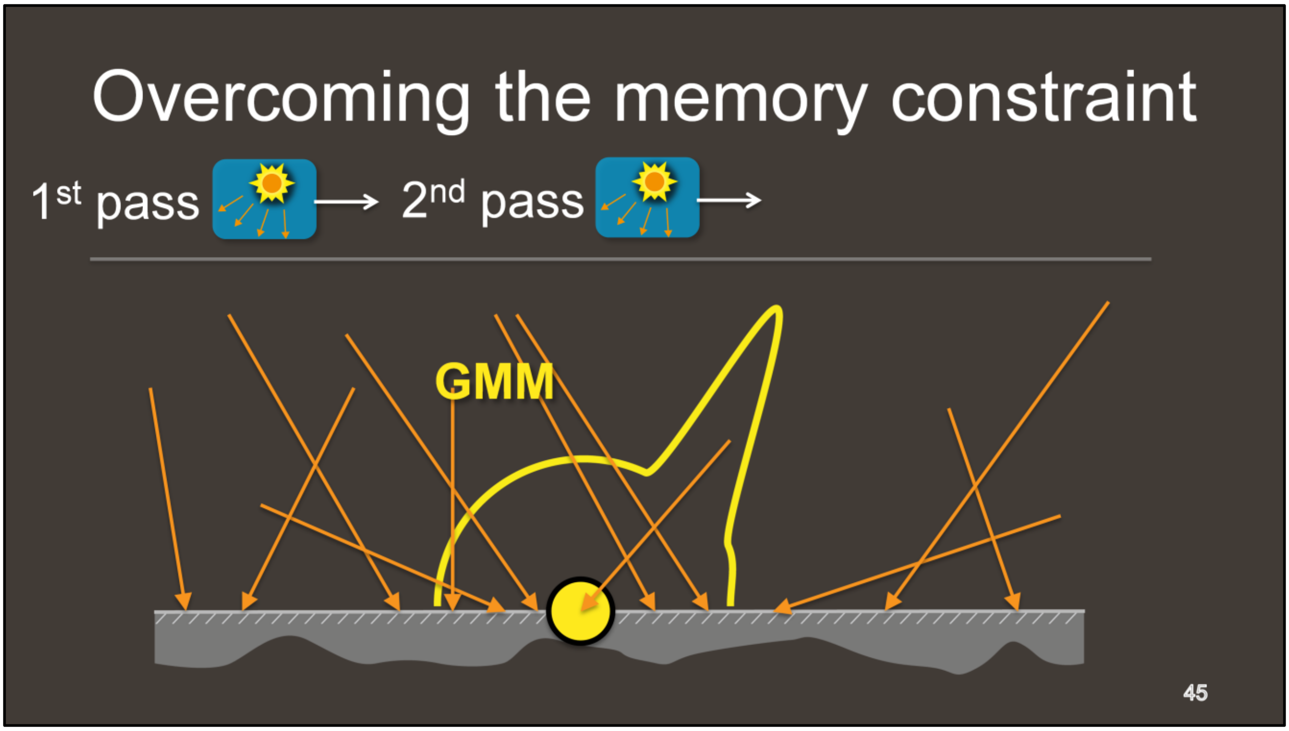

Overcoming the memory constraint
1st pass
GMM
42



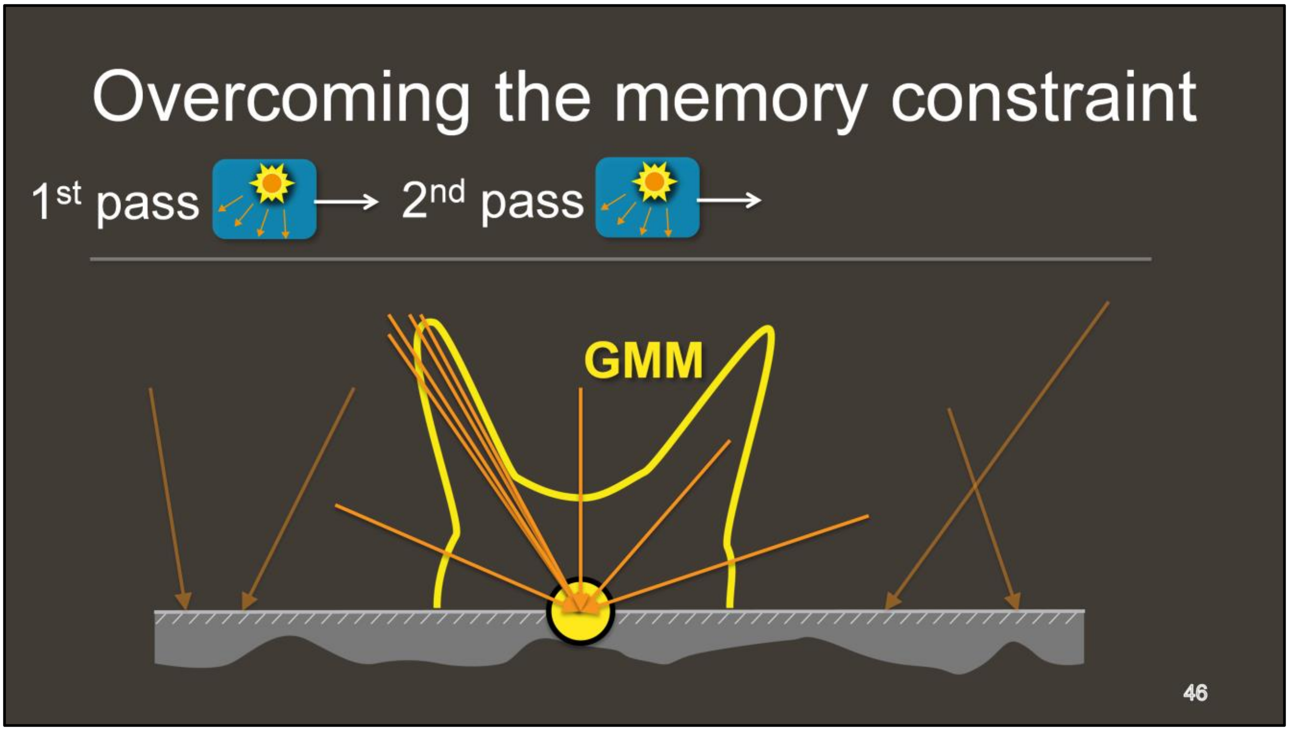
Now the particles can be removed …



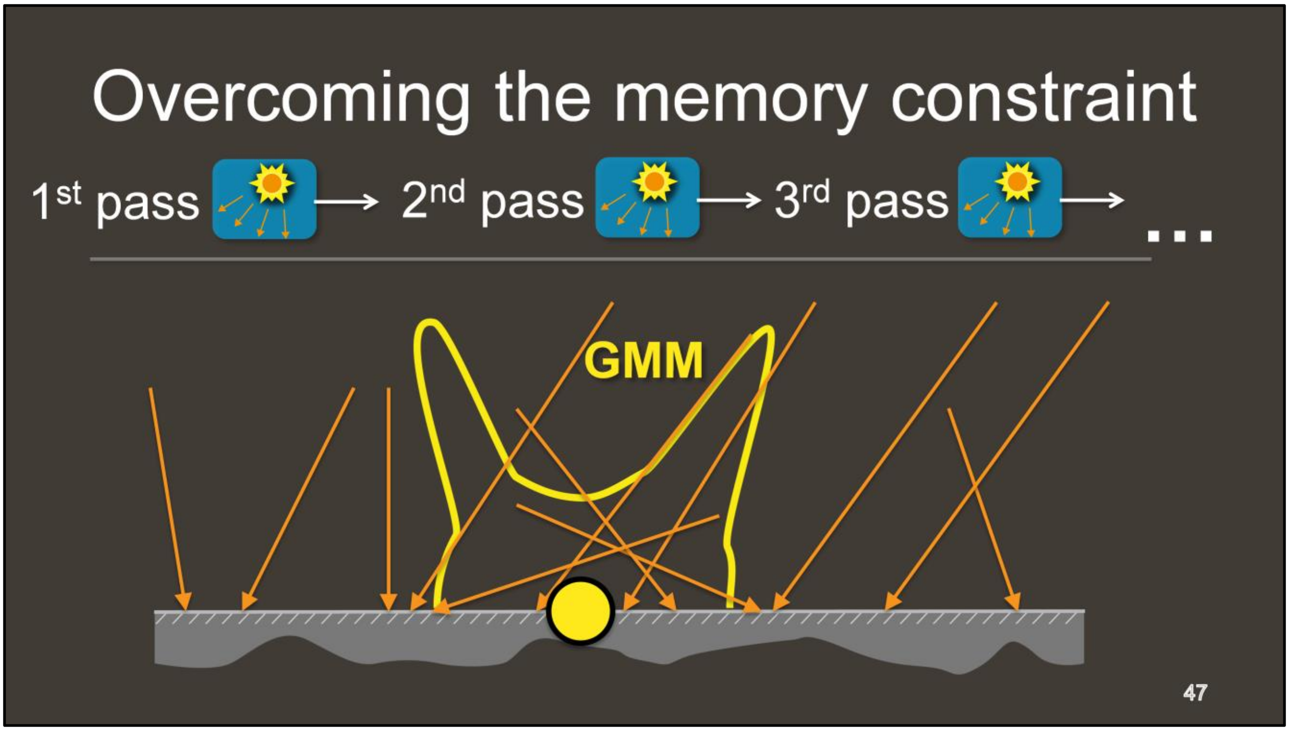
… because the information is absorbed in the distribution.



Then we trace another batch of particles,

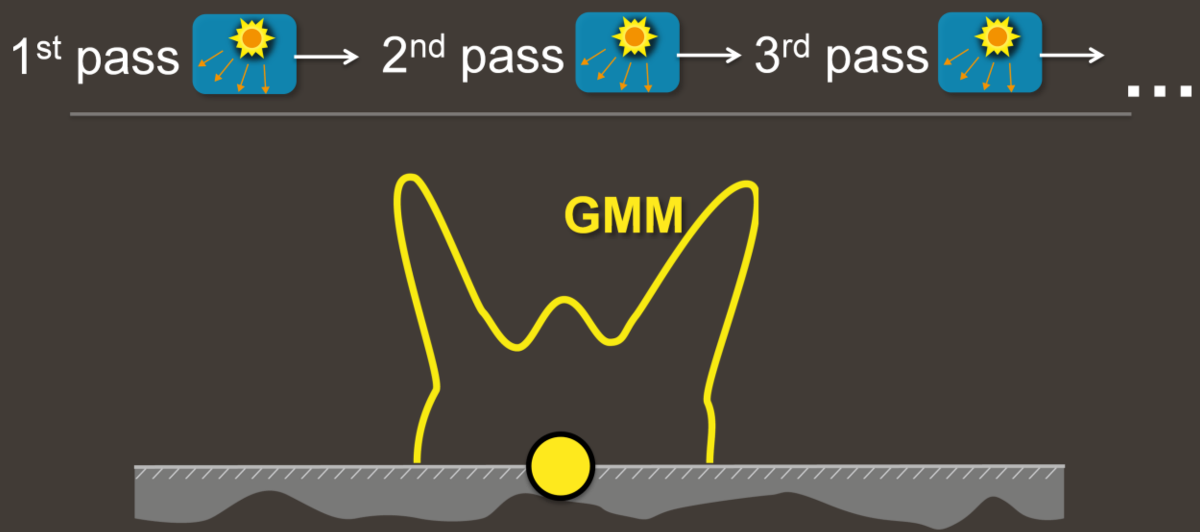


… and we use the nearest ones to progressively update the distribution.

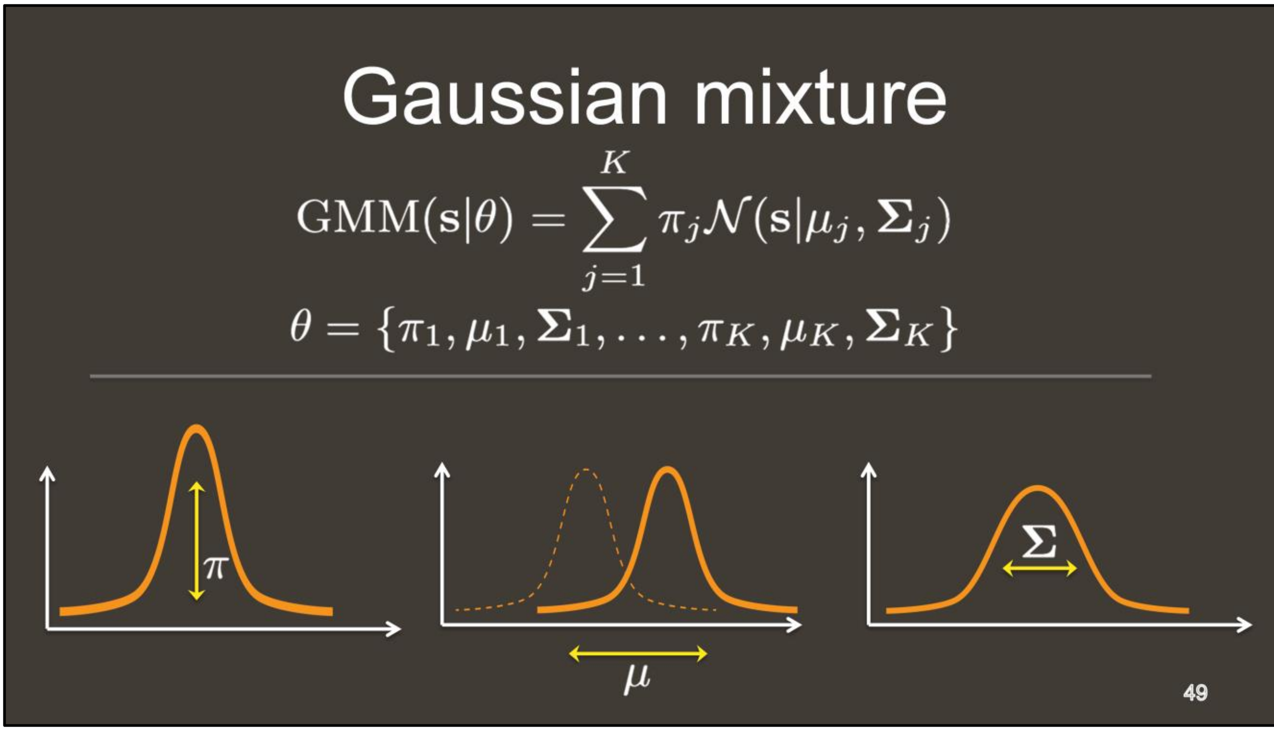


We can repeat this process until the distribution is fully trained.

Overcoming the memory constraint



48



Lets see what the Gaussian mixture actually is.

It is a convex combination of Gaussian distributions that is fully described by [CLICK] a set of parameters theta.

Each Gaussian is defined by its [CLICK] mixture weight, [CLICK] its mean and [CLICK] its covariance matrix.

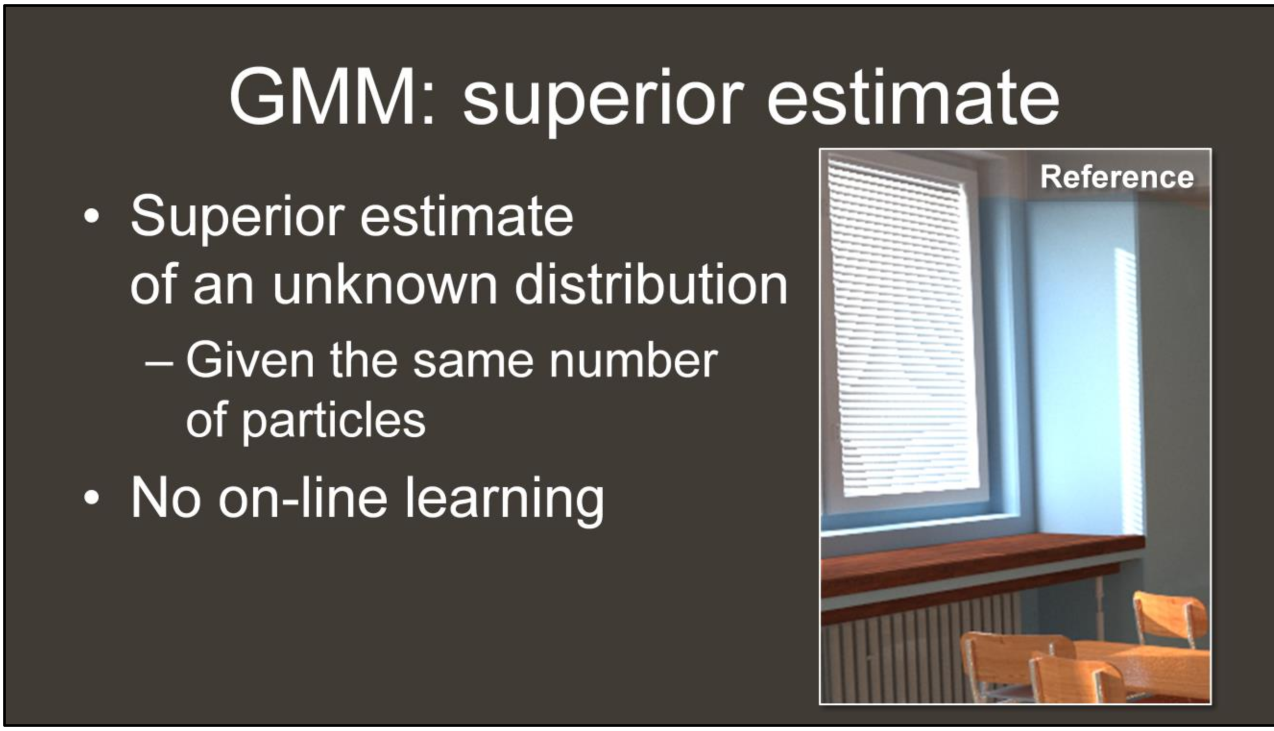
Gaussian mixture

$$\mathrm{GMM}(\mathbf{s}|\theta) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{s}|\mu_j, \mathbf{\Sigma}_j)$$

$$\theta = \{\pi_1, \mu_1, \mathbf{\Sigma}_1, \ldots, \pi_K, \mu_K, \mathbf{\Sigma}_K\}$$

50

The Gaussians are defined in plane, [CLICK] however, to obtain a distribution over the hemisphere, [CLICK] we simply use [CLICK] the mapping of Shirley and Chiu [1997].



I would like to stress that on-line learning is not the only advantage of Gaussian mixture models over the previous directional distribution representations.
[CLICK]
It is also better at estimating of unknown distributions.

Let me demonstrate on an example.
[CLICK]
We rendered a complex scene where we have traced as many photons in one single batch as we could.
[CLICK]
There is no on-line learning involved – that means that all models were given the same number of particles.

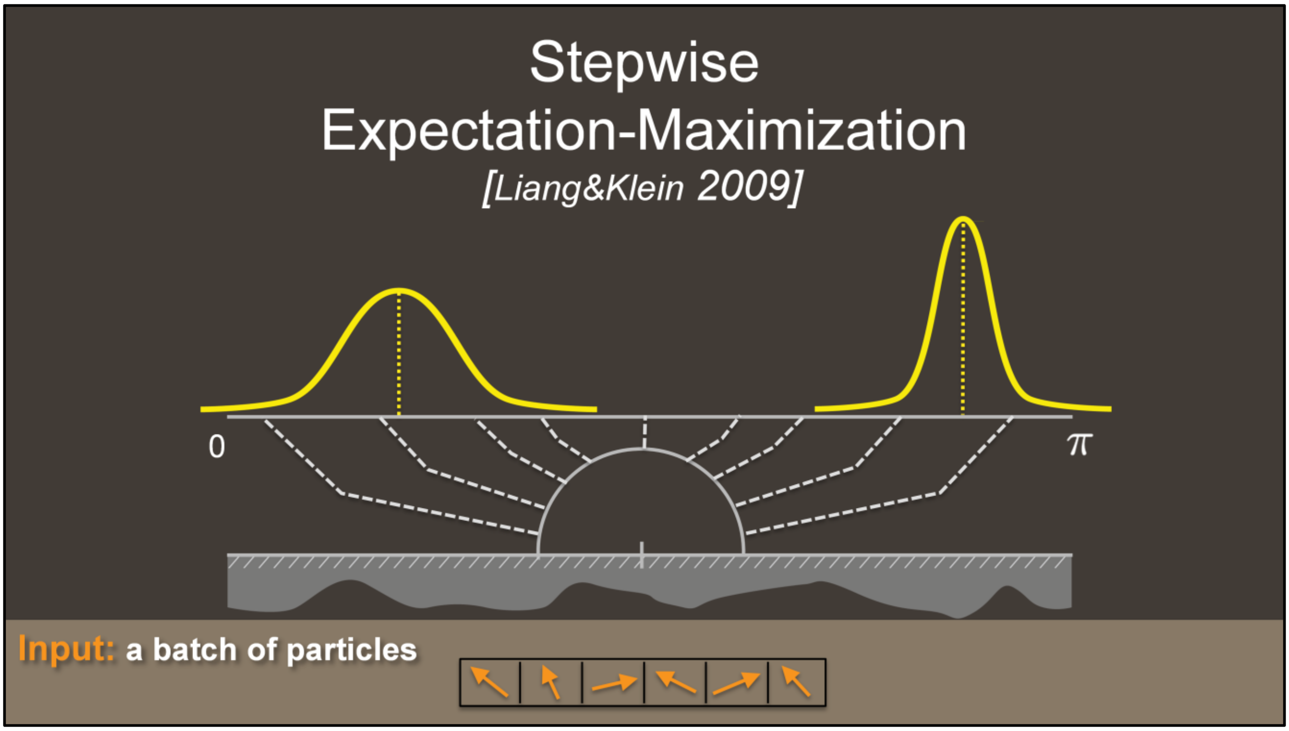
Then we used path-tracing guided by Jensen's histograms, …
[CLICK]
… Hey and Purgathofer's cones …
[CLICK]
… and our Gaussian mixtures.

The amount of noise after 1h of rendering suggests that the Gaussian mixture model is clearly superior to the previous models.

GMM: learning

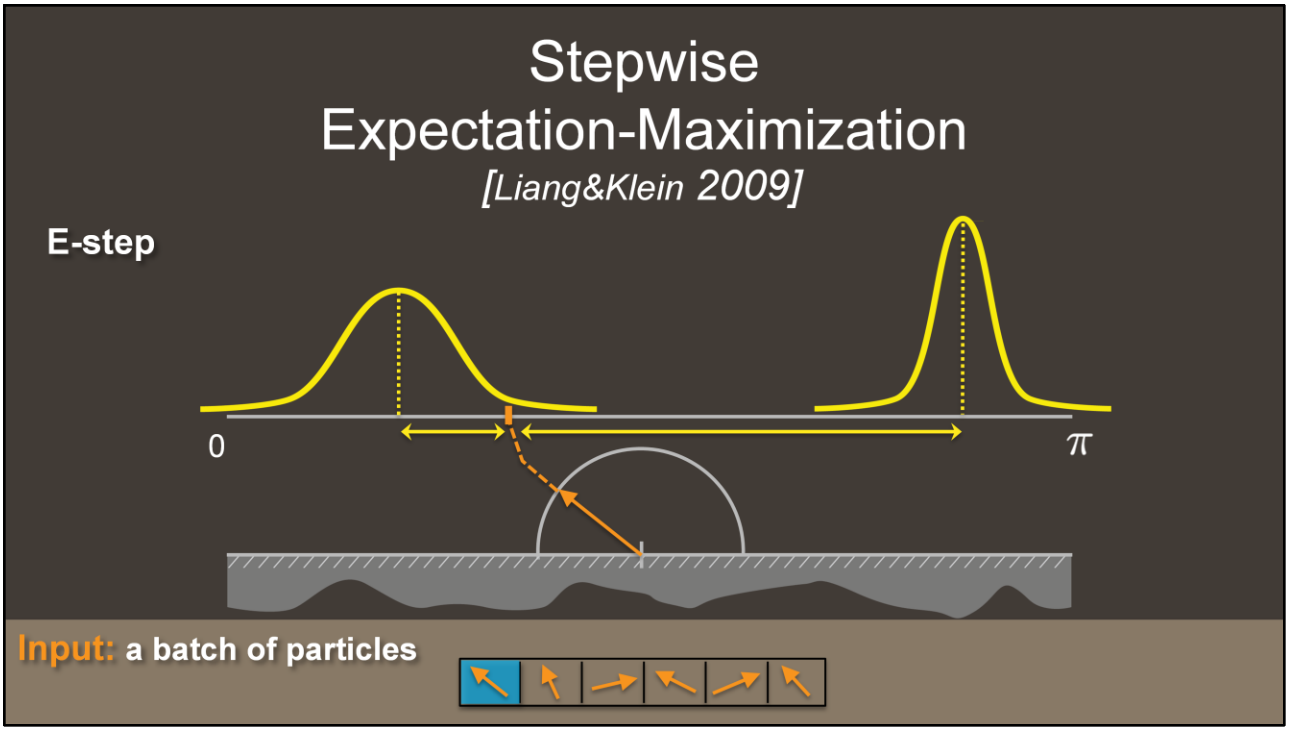
53

For learning of Gaussian mixtures …



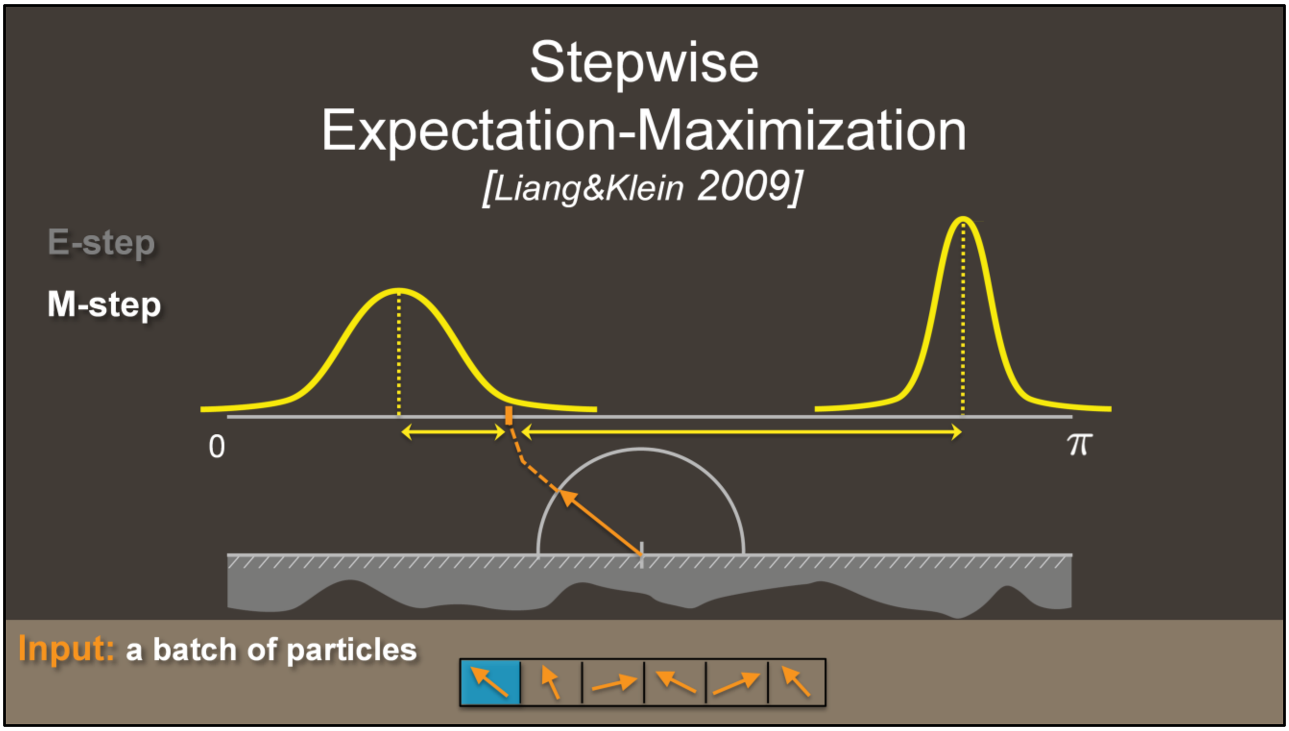
… we use the stepwise Expectation-Maximization.
This is different from the usual EM algorithm previously used in graphics in that the E and M steps are performed immediately after observing each particle.

[CLICK] In this 1D scheme we have a hemisphere
[CLICK] that is mapped to a plane.
The example mixture has only two Gaussians [CLICK]
and the input of the algorithm is [CLICK] a batch of particles.

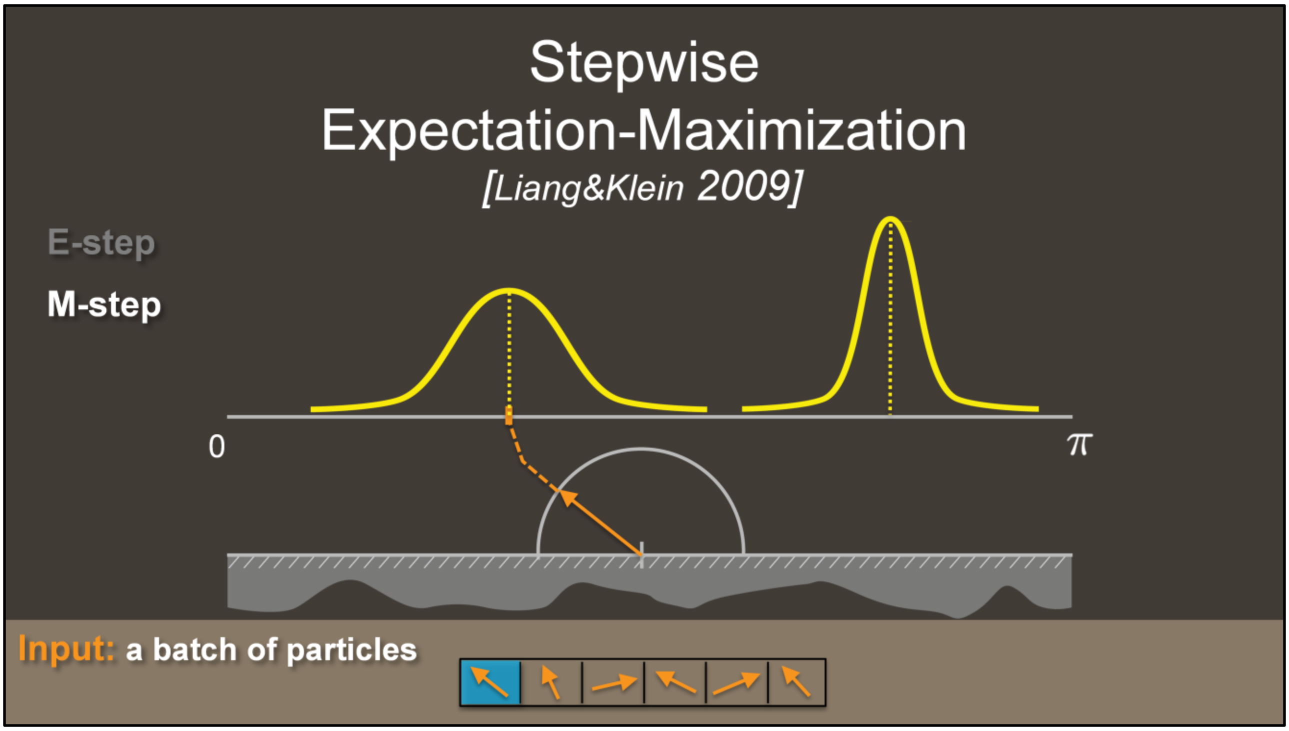


The algorithm takes the particles one-by-one [CLICK] and every particle is processed in two steps. First, in the E-step, [CLICK] it determines its “soft” assignment to all Gaussians in the mixture [CLICK].

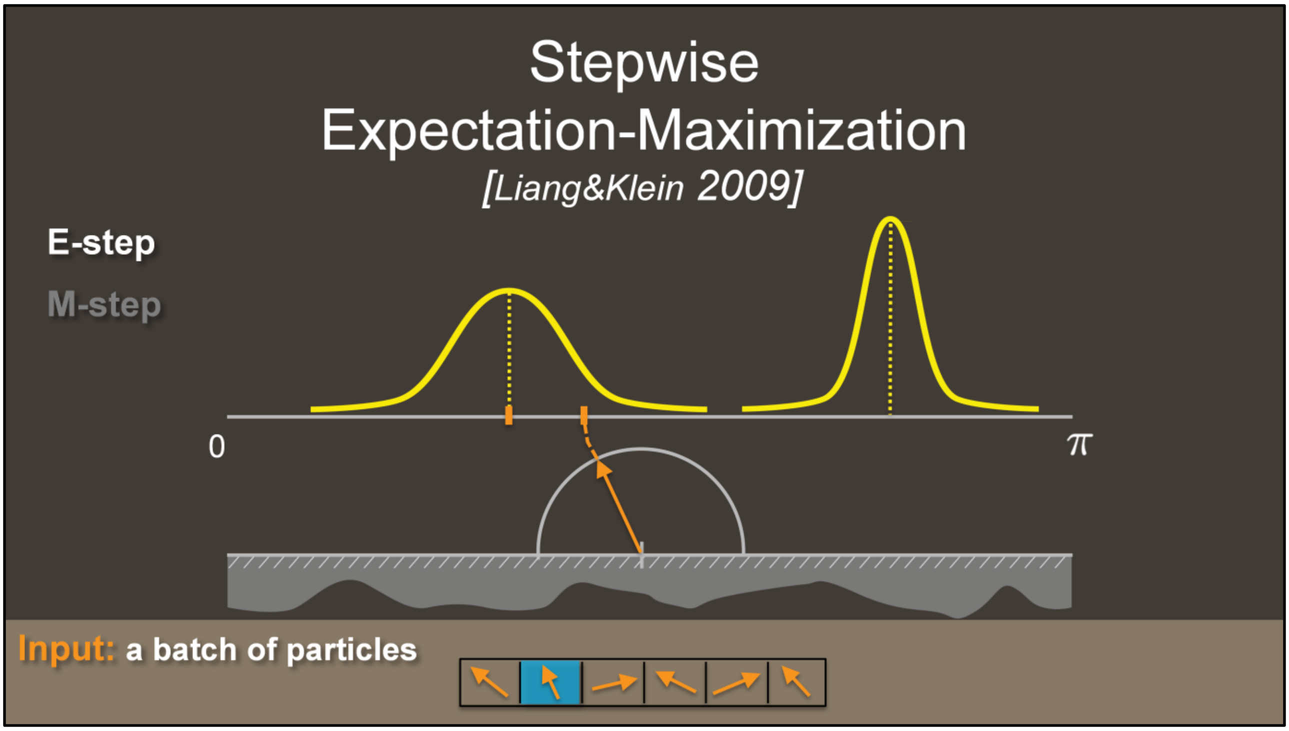
Then follows the M-step,



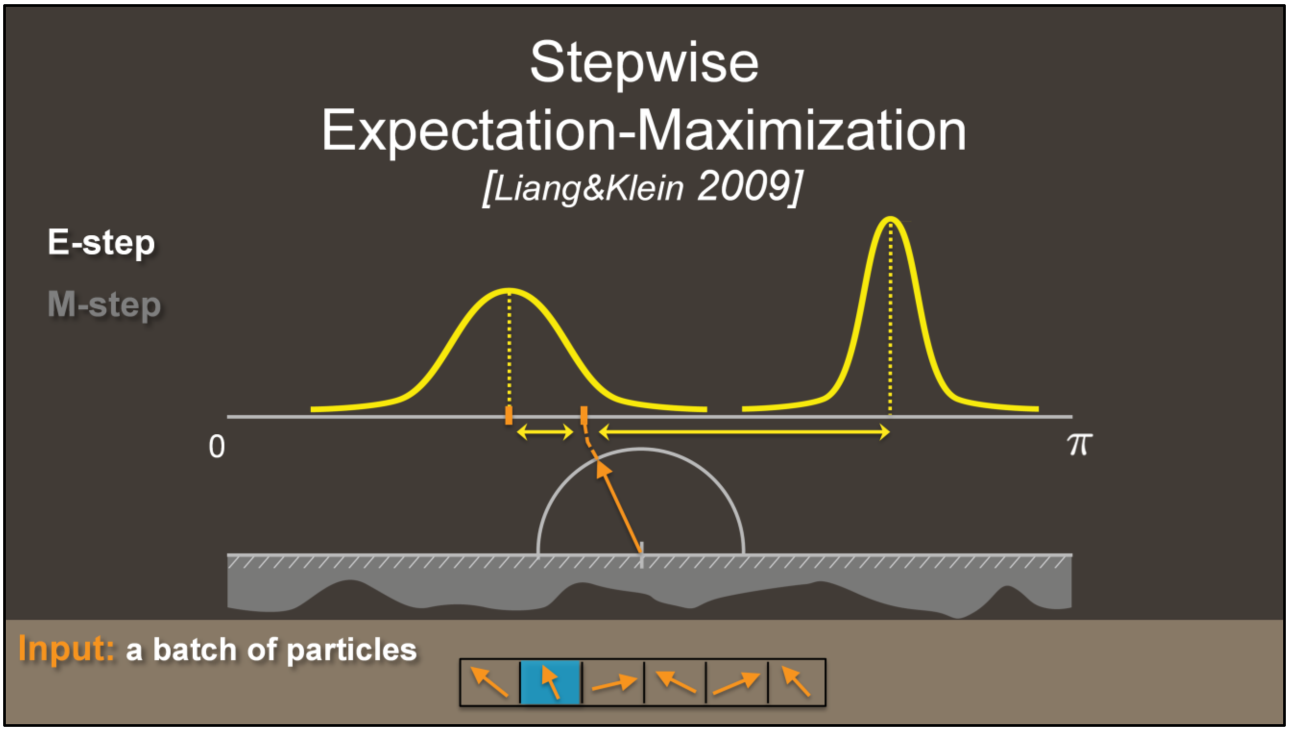
in which the parameters of all Gaussians are updated.



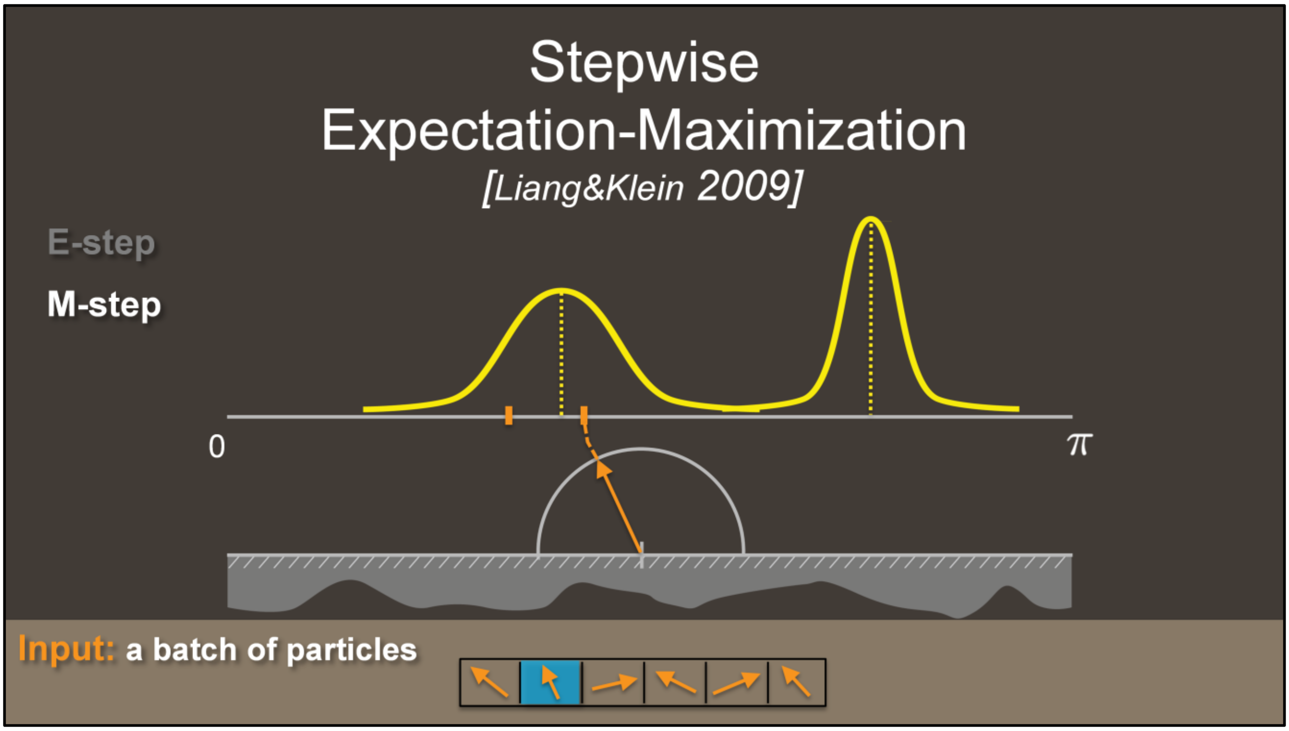
The update is based on the assignment from the E-step.



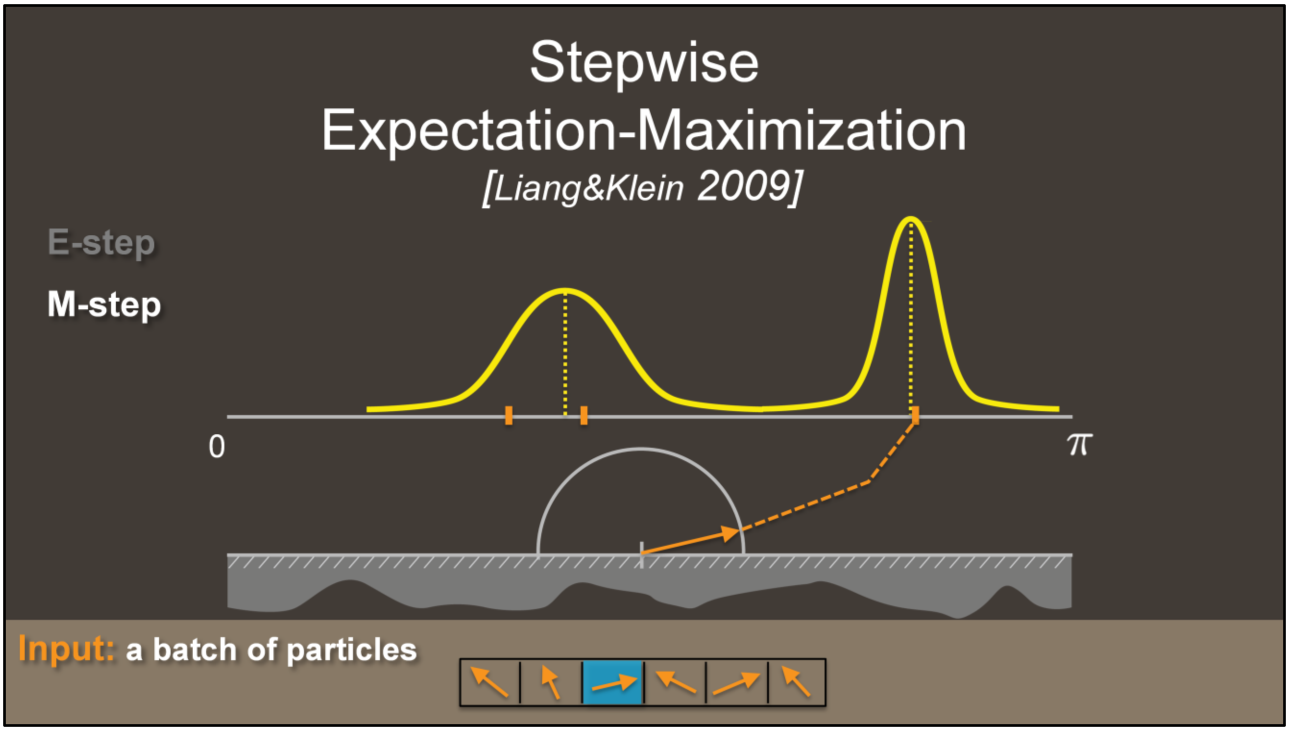
The algorithm proceeds to another particle, it goes through the E-step [CLICK] and updates parameters in the M-step. [CLICK]



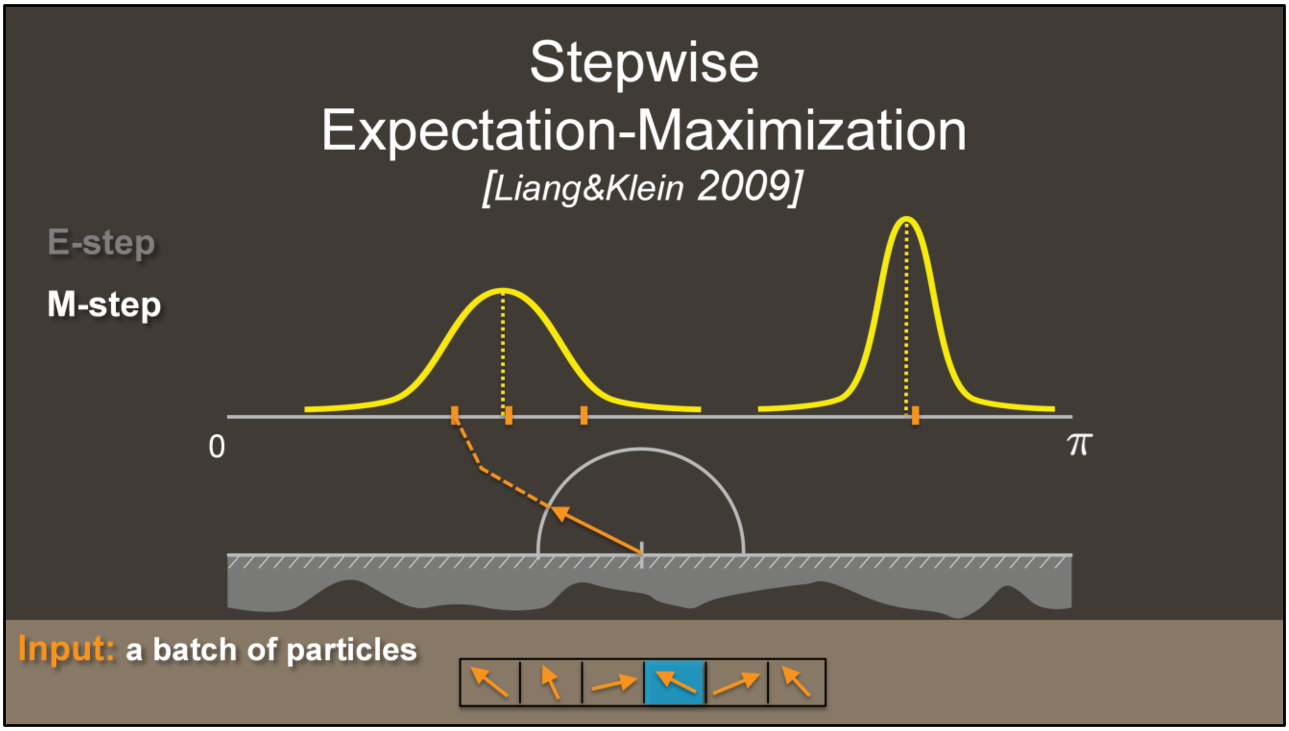
The algorithm proceeds to another particle, it goes through the E-step [CLICK] and updates parameters in the M-step. [CLICK]



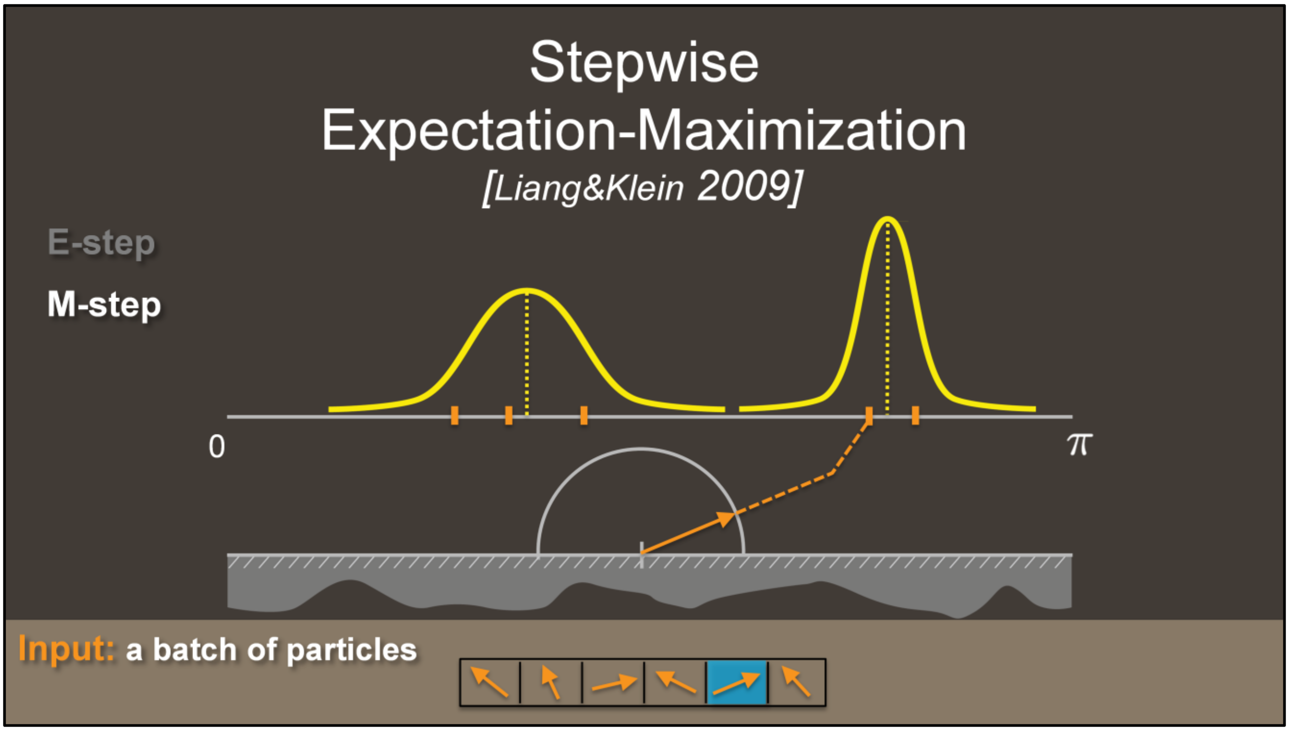
The algorithm proceeds to another particle, it goes through the E-step [CLICK] and updates parameters in the M-step. [CLICK]



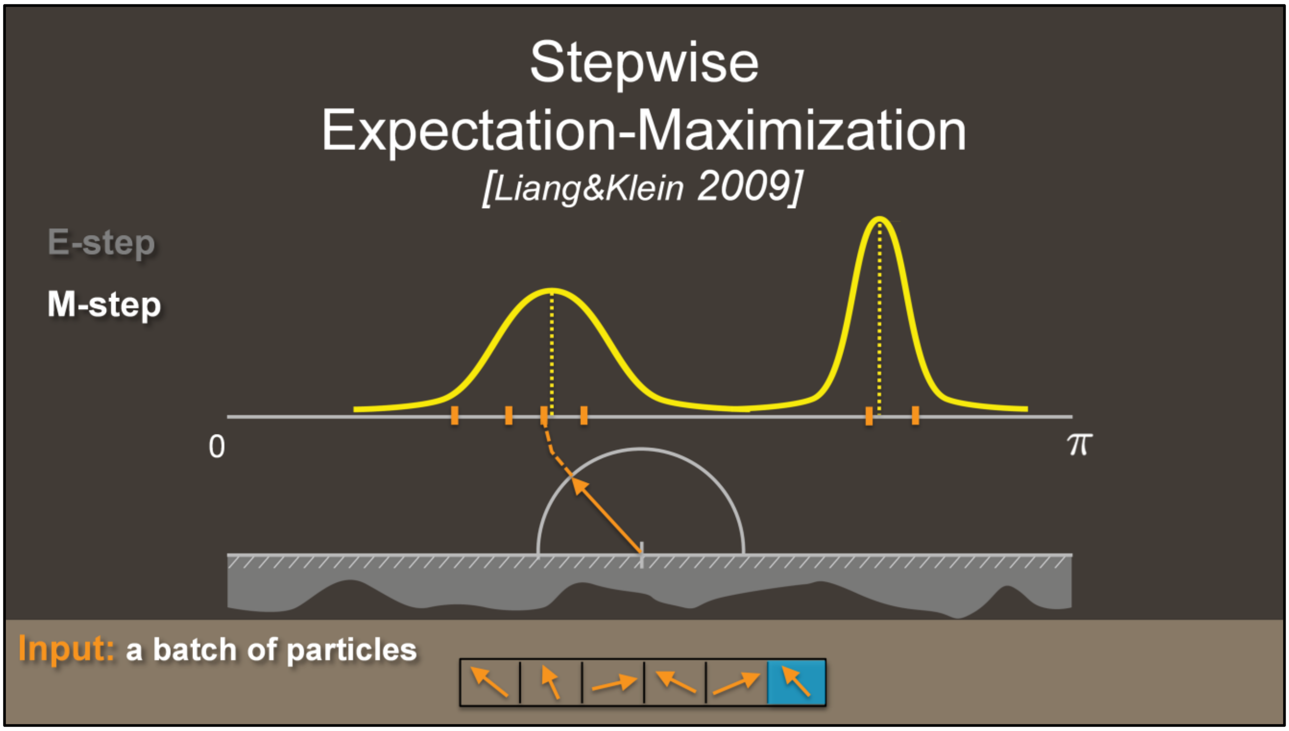
These steps are repeated for all particles in the batch…



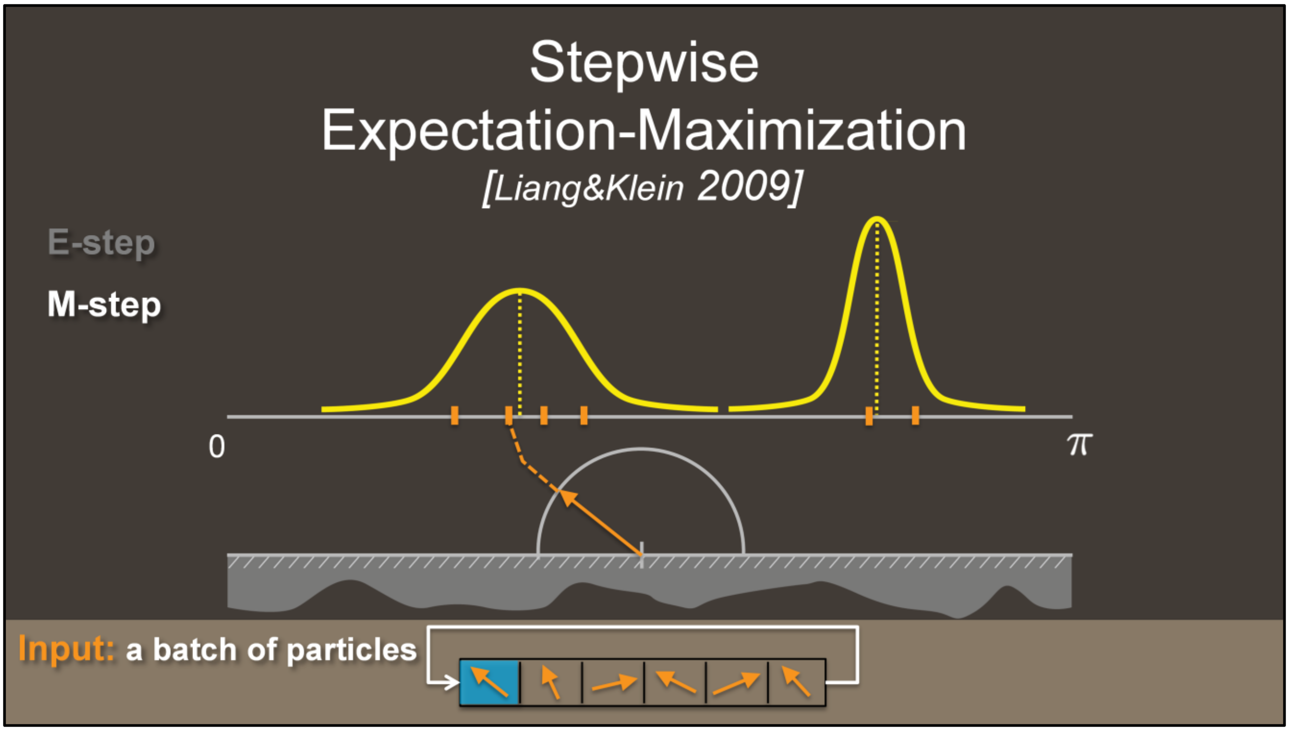
These steps are repeated for all particles in the batch…



These steps are repeated for all particles in the batch…



… and when the last particle from the input batch is finally processed …



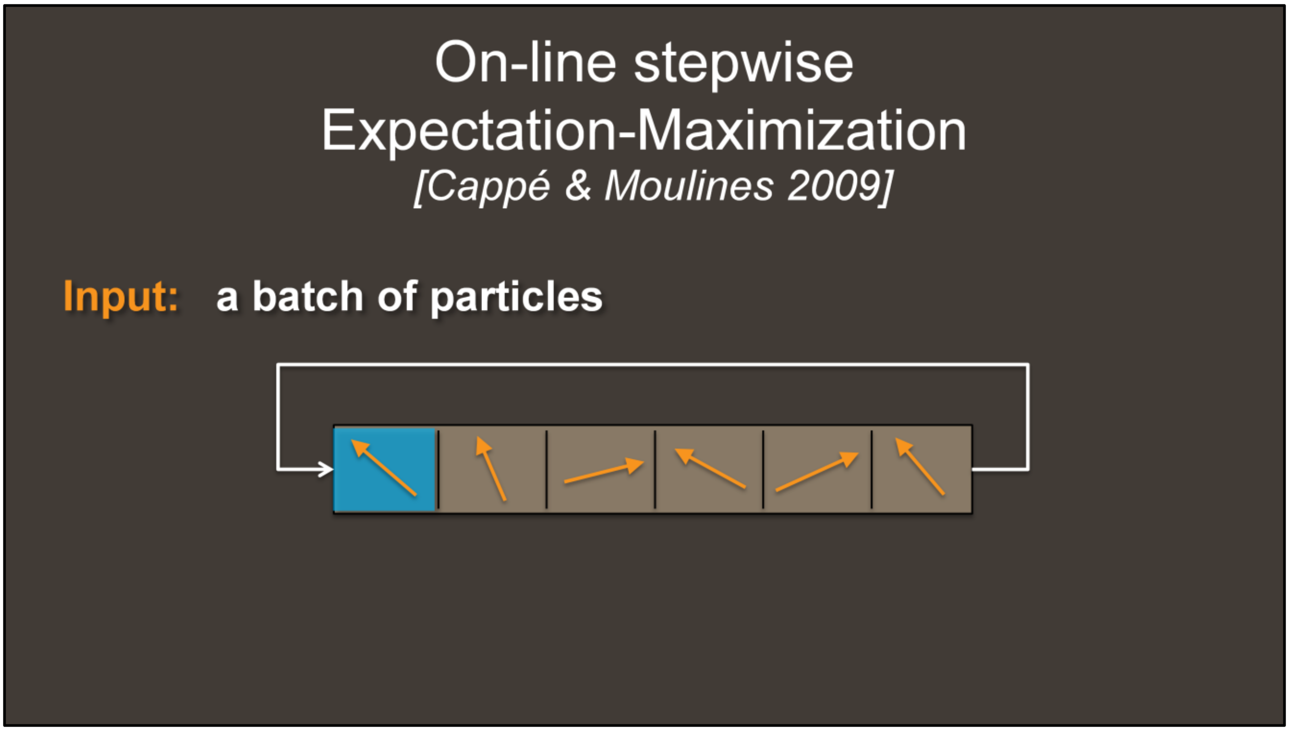
the algorithm returns to the first particle again.

This process continues until convergence.

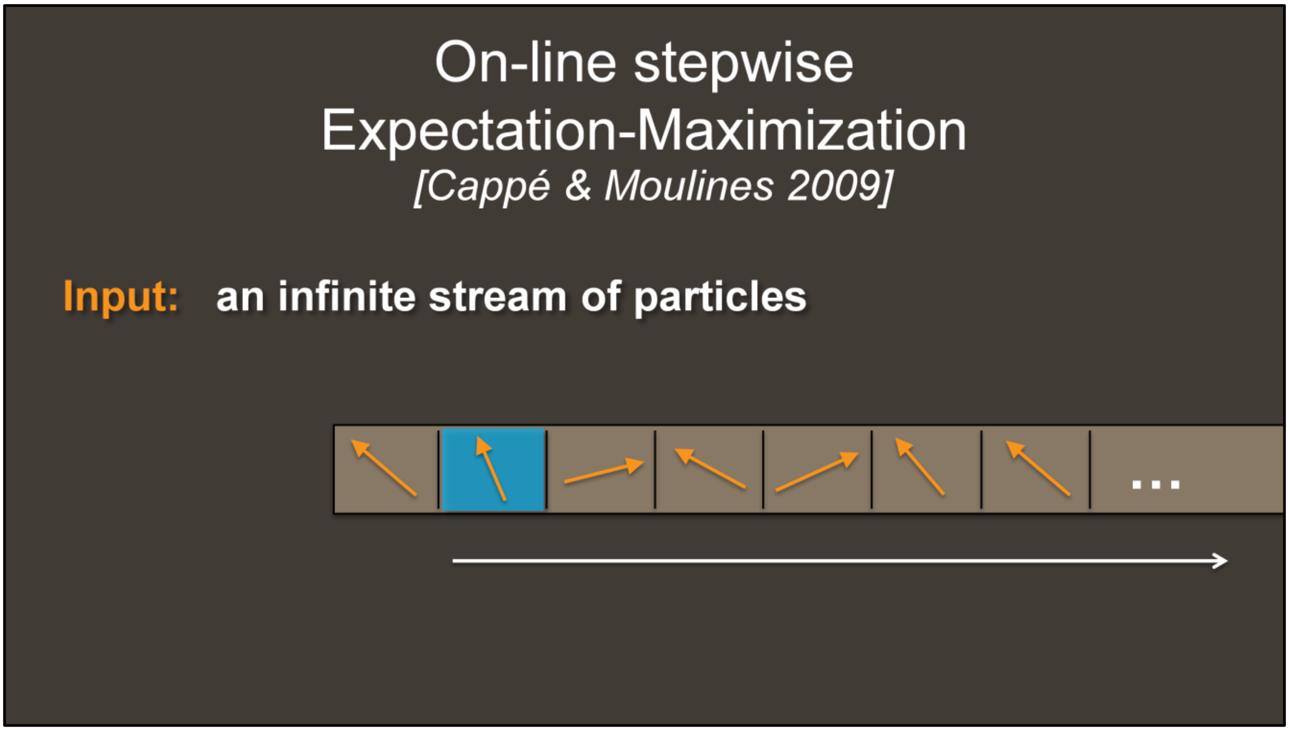
Stepwise Expectation-Maximization

[Liang&Klein 2009]

Input: **a batch of particles**



Thanks to the stepwise formulation, making the algorithm on-line is very simple.

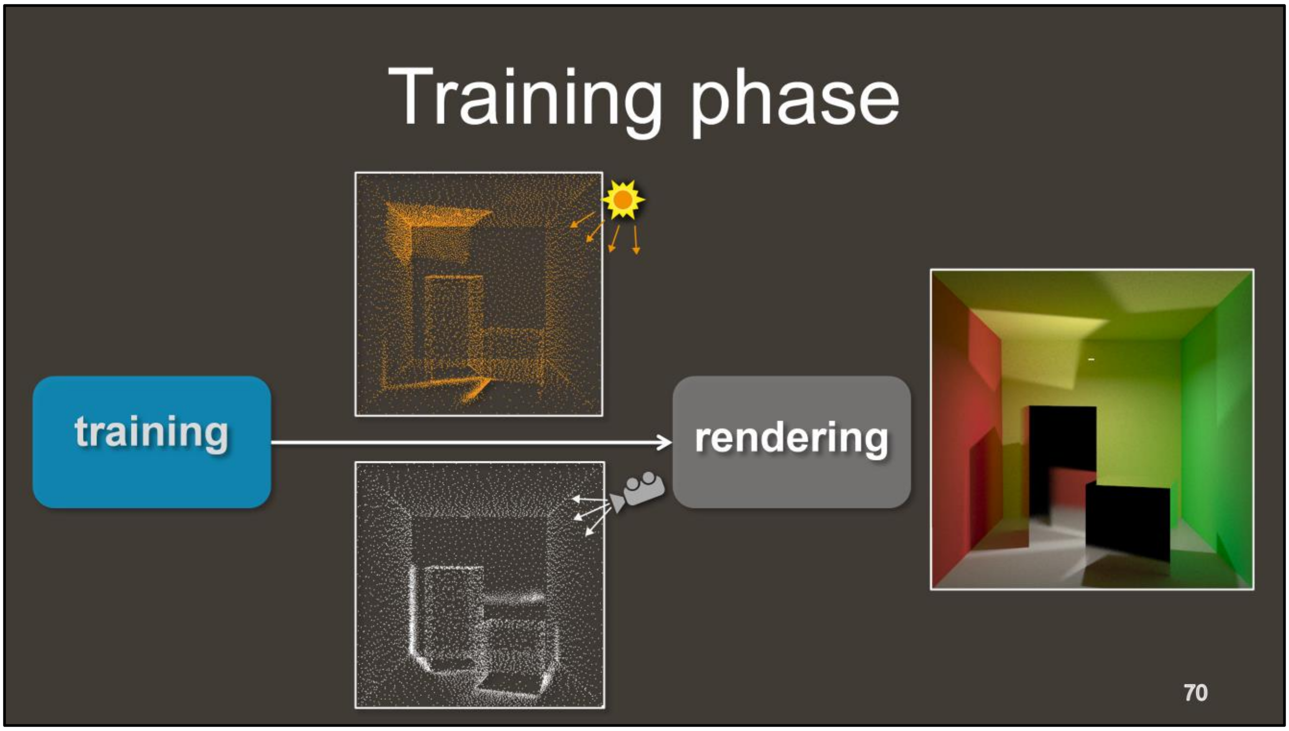


All we need to do is to read particles from potentially infinite stream and never return to previously used particles.

Note that this would not be possible with the usual E-M algorithm previously used in graphics.

Method's outline

Now I will finally outline the complete picture of our method.



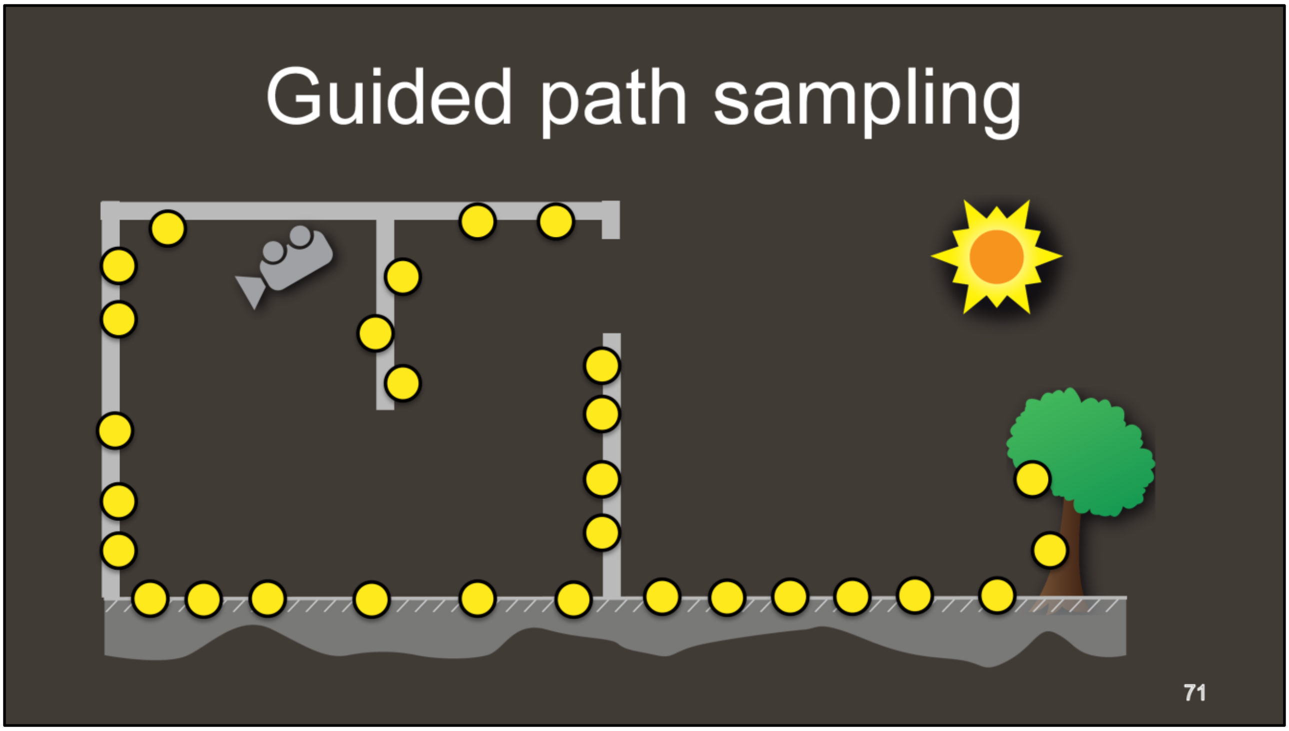
Before rendering, we train our Gaussian mixtures in a training phase.

The result of this phase are two spatial caches of distributions.

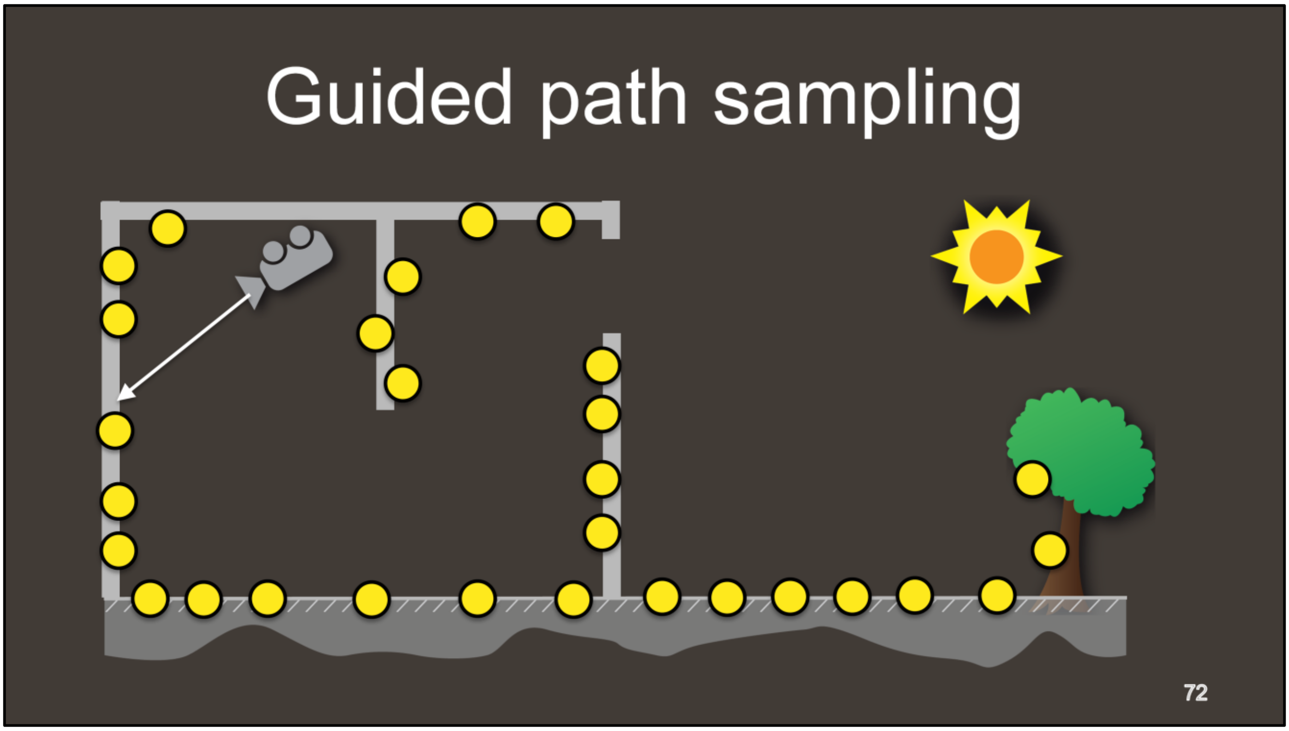
[CLICK]

One of them is trained from photons and thus contains directional distributions of radiance
[CLICK]
while the other is trained from camera particles and contains distributions of visual importance.

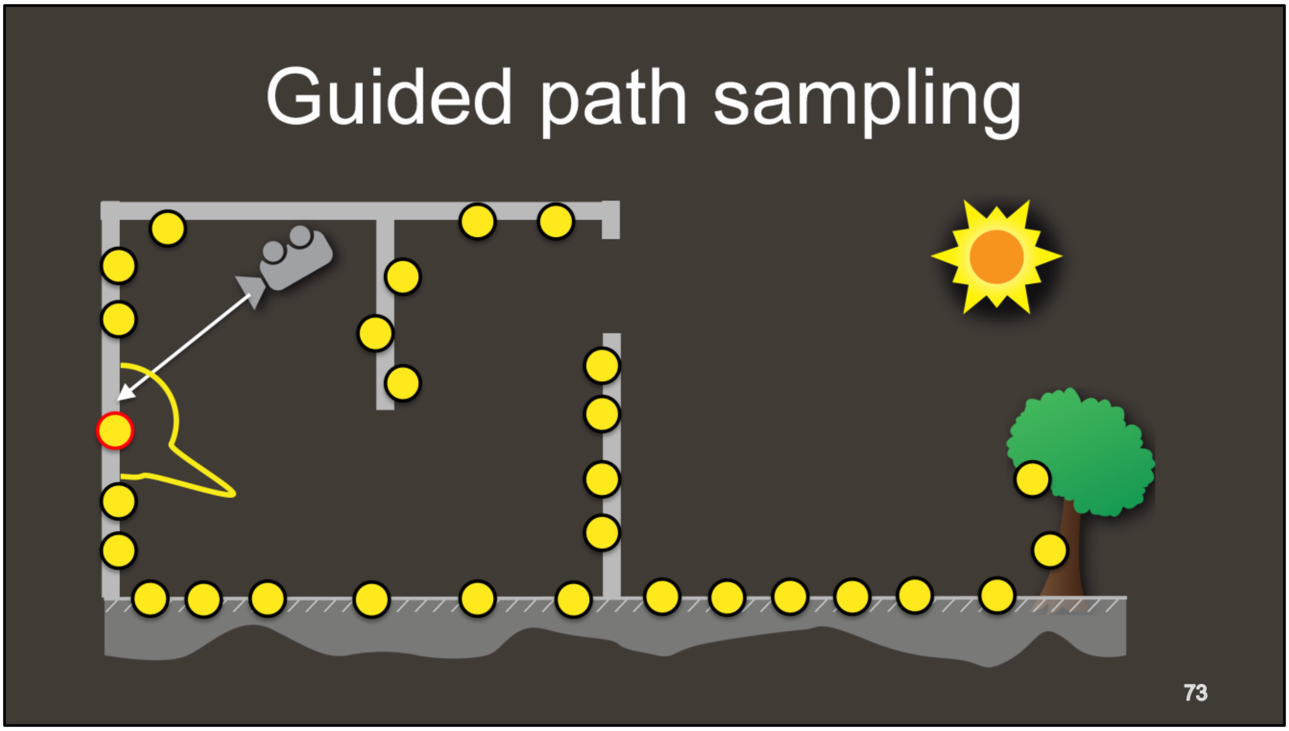
These distributions are used to guide path-sampling during the rendering phase.



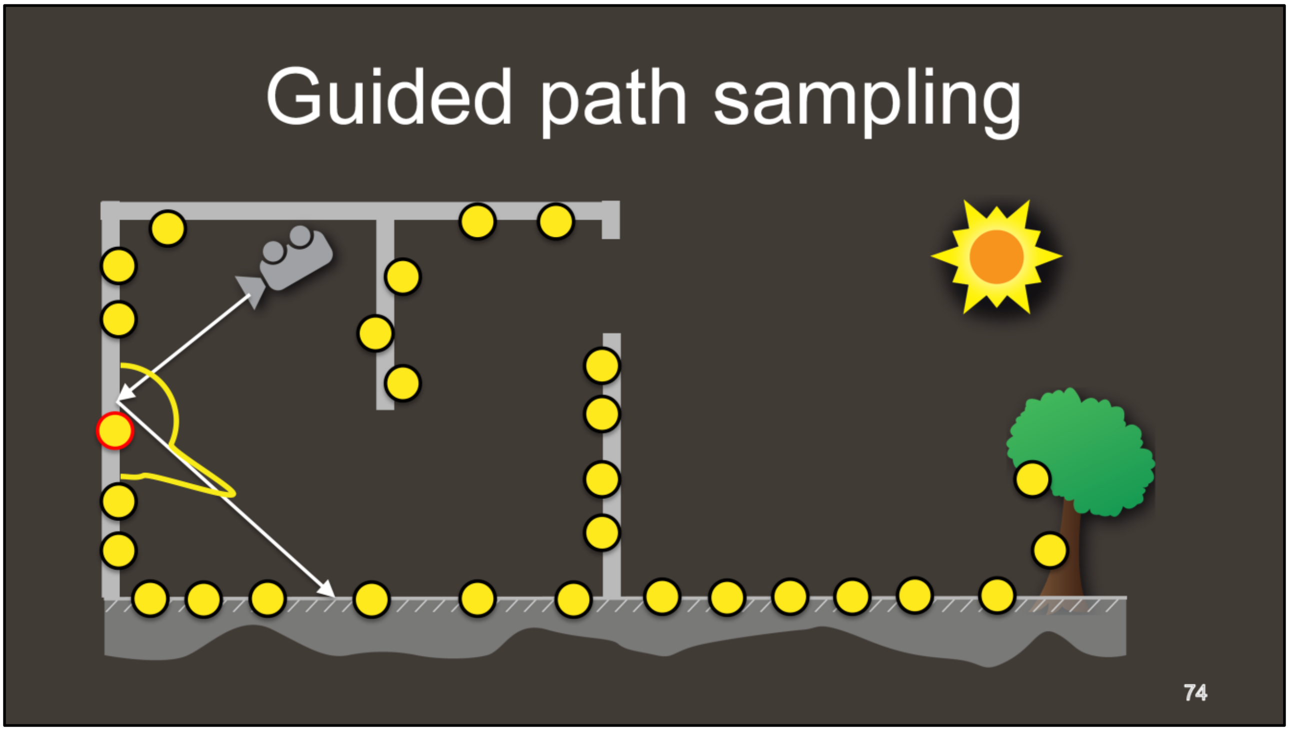
I will explain the guided path-sampling on the illustration.
[CLICK]
Here, we trace a path from the camera and we use the nearest radiance distributions for its guiding towards the light sources.



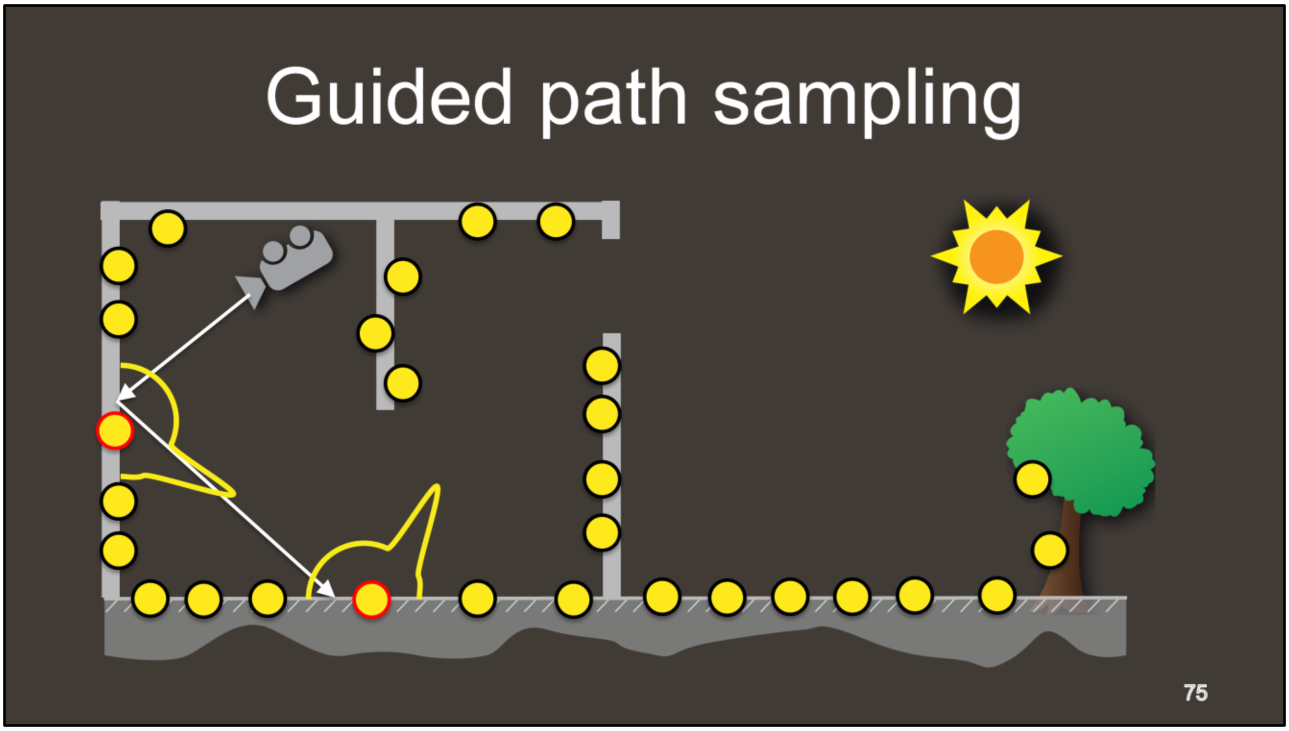
I will explain the guided path-sampling on the illustration.
[CLICK]
Here, we trace a path from the camera and we use the nearest radiance distributions for its guiding towards the light sources.



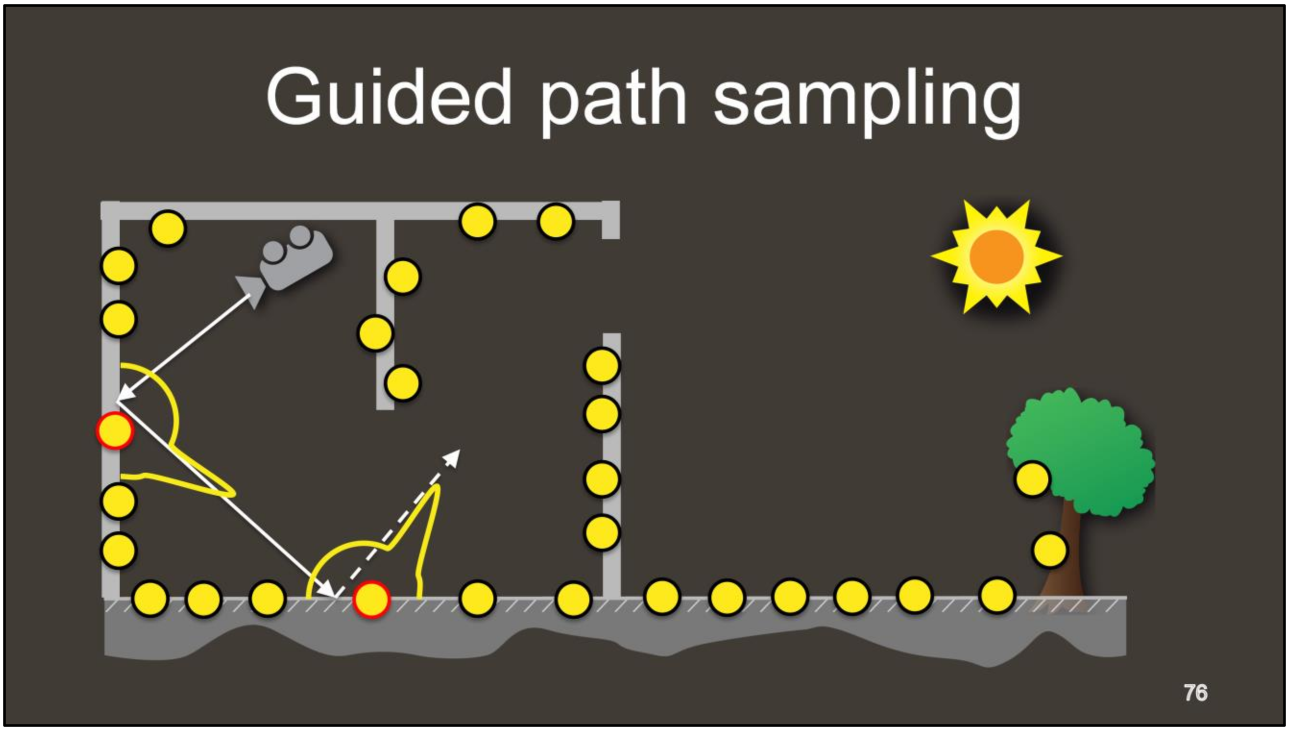
I will explain the guided path-sampling on the illustration.
[CLICK]
Here, we trace a path from the camera and we use the nearest radiance distributions for its guiding towards the light sources.



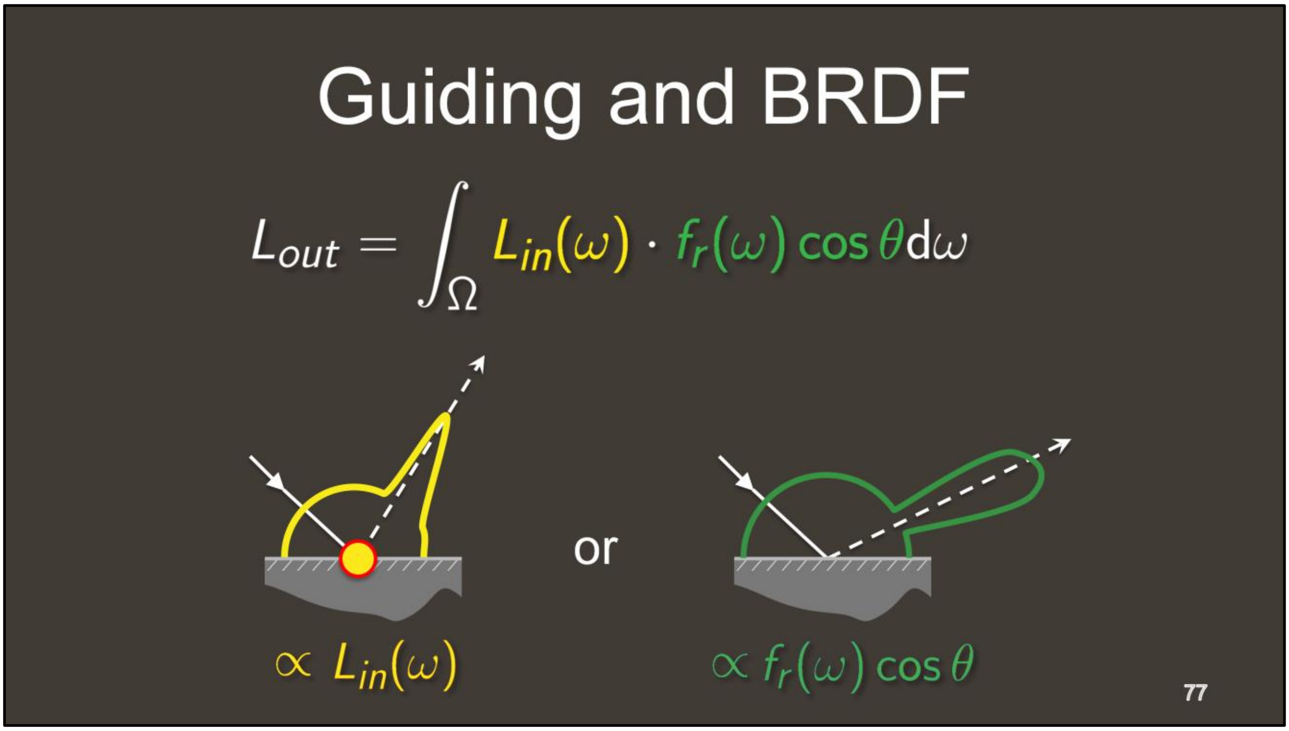
I will explain the guided path-sampling on the illustration.
[CLICK]
Here, we trace a path from the camera and we use the nearest radiance distributions for its guiding towards the light sources.



I will explain the guided path-sampling on the illustration.
[CLICK]
Here, we trace a path from the camera and we use the nearest radiance distributions for its guiding towards the light sources.



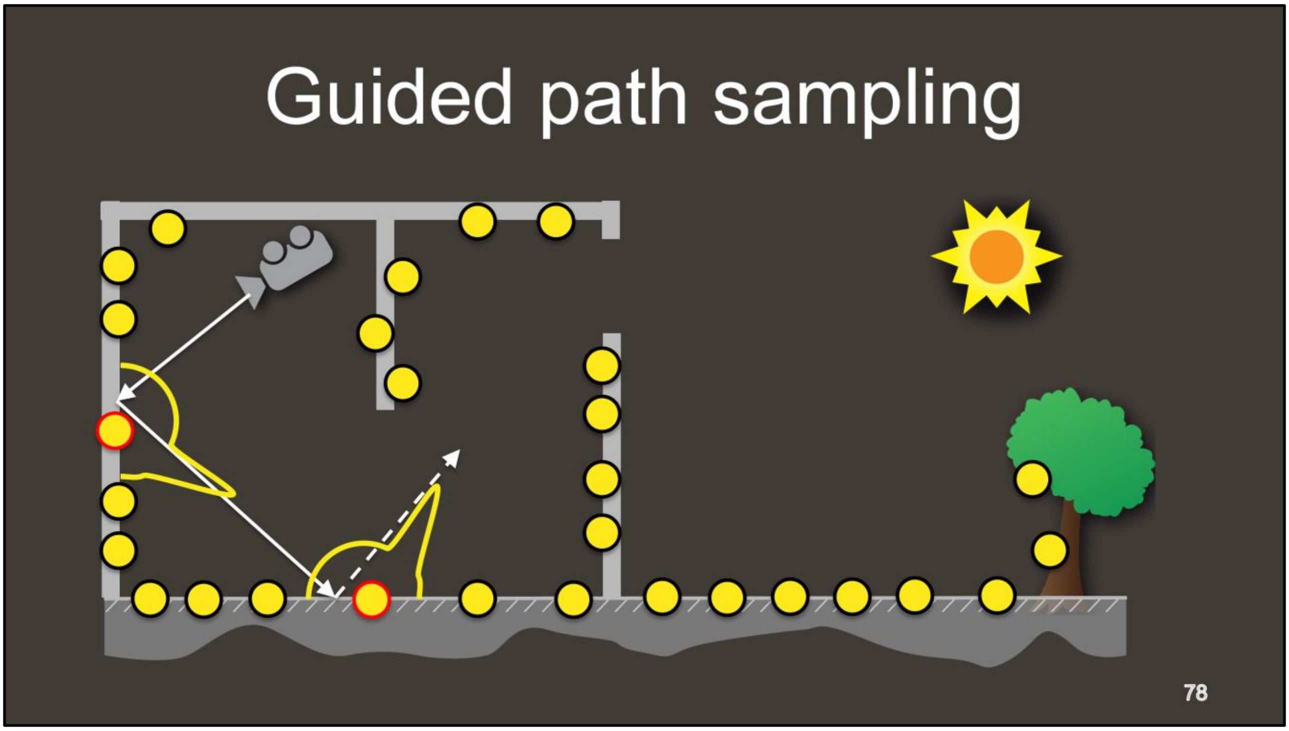
I will explain the guided path-sampling on the illustration.
[CLICK]
Here, we trace a path from the camera and we use the nearest radiance distributions for its guiding towards the light sources.



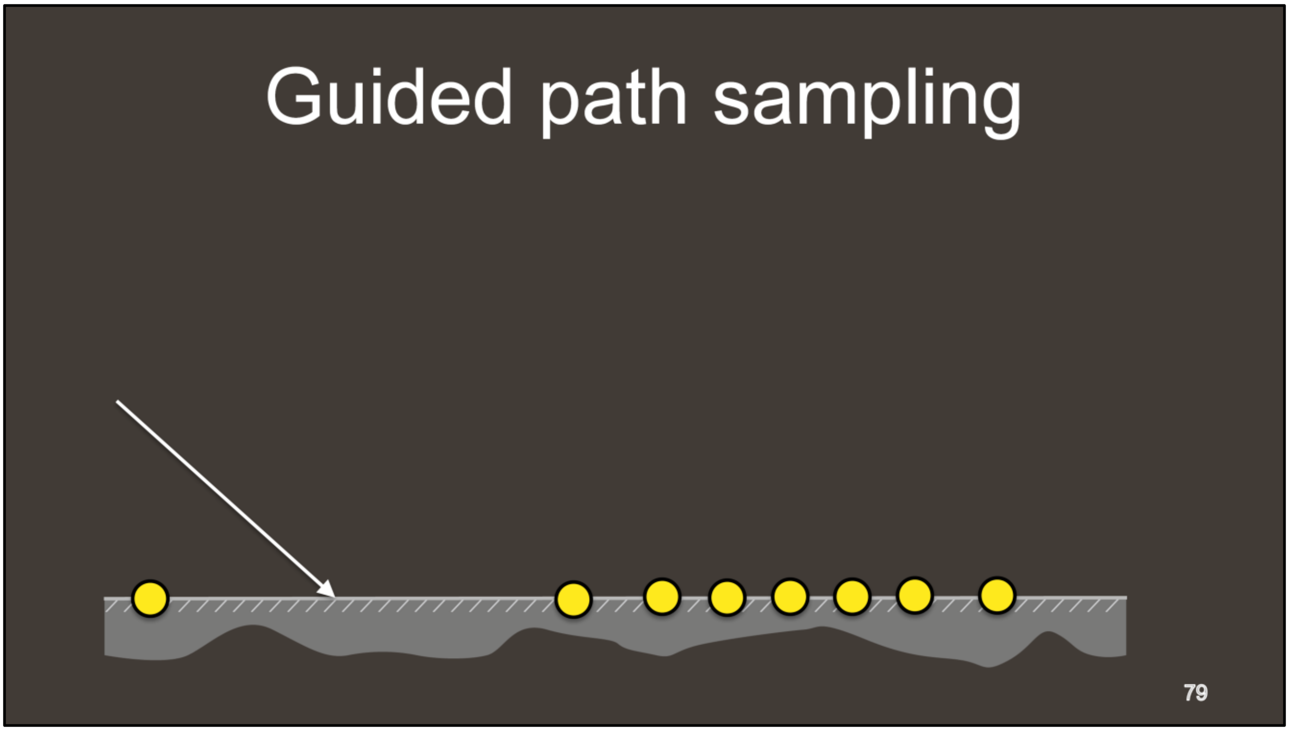
In our method we also sample the BRDF.

To sample a new direction we randomly use [CLICK] either our guiding distribution [CLICK] or the BRDF.

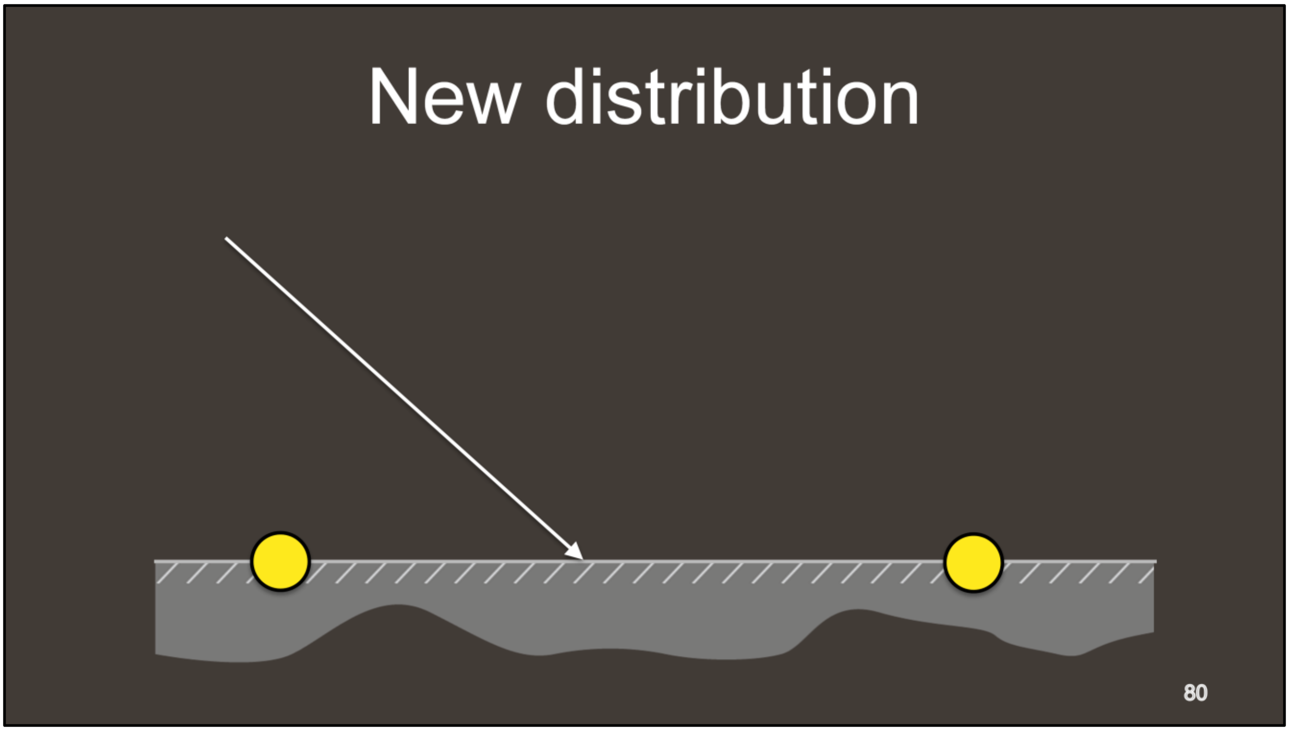
Both sampling strategies, are combined via multiple importance sampling.



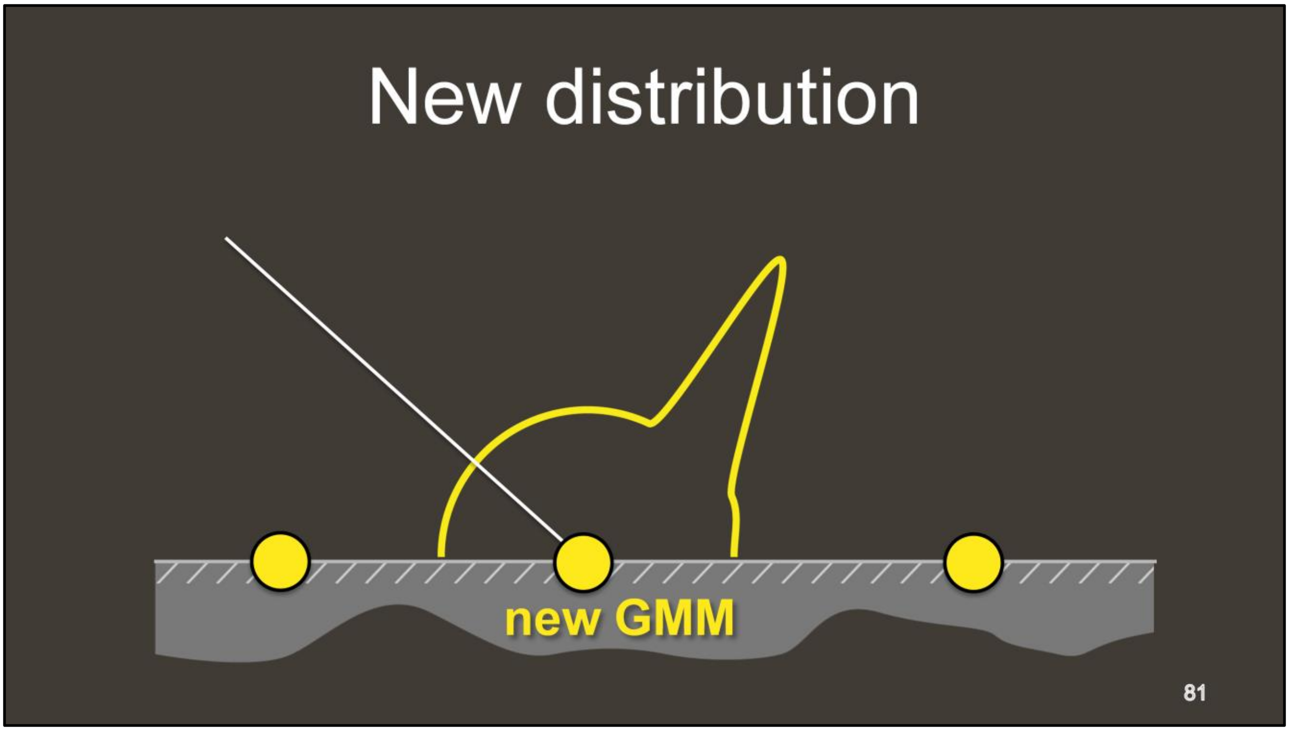
During the guided path-sampling, we also create new distributions on-the-fly as needed.



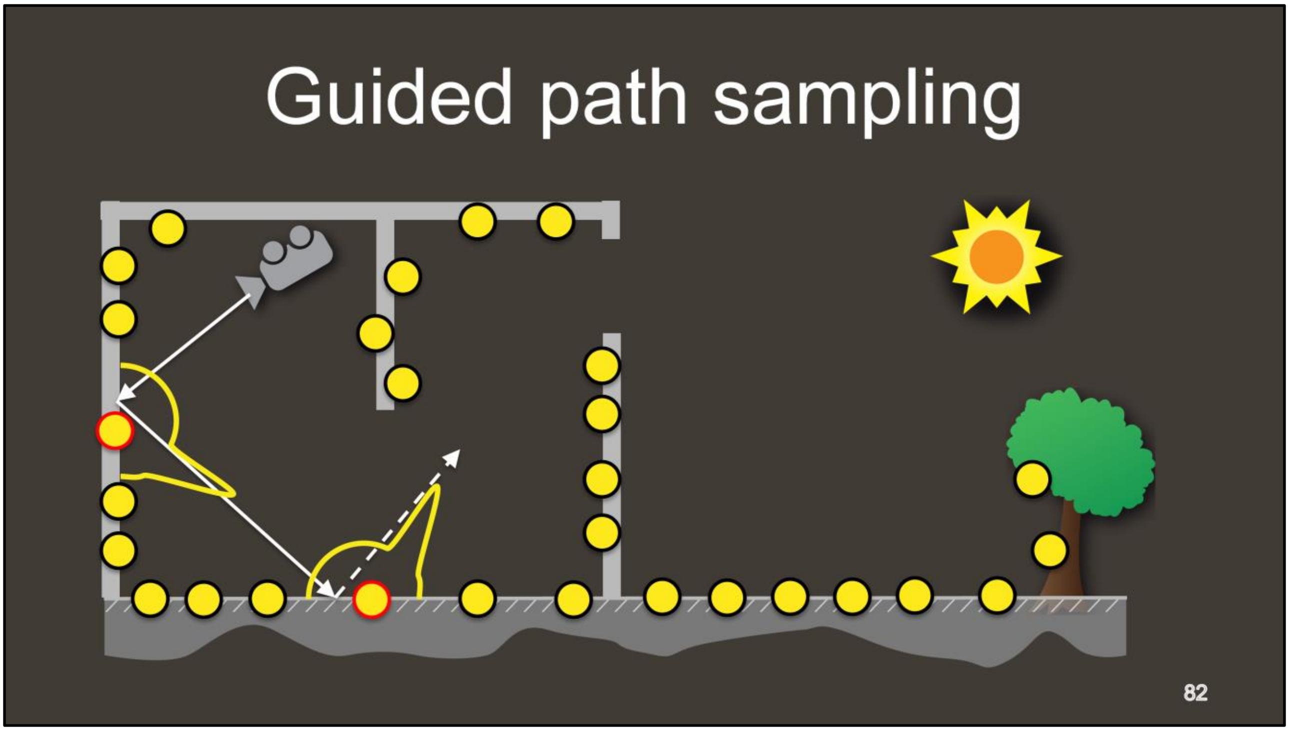
During the guided path-sampling, we also create new distributions on-the-fly as needed.



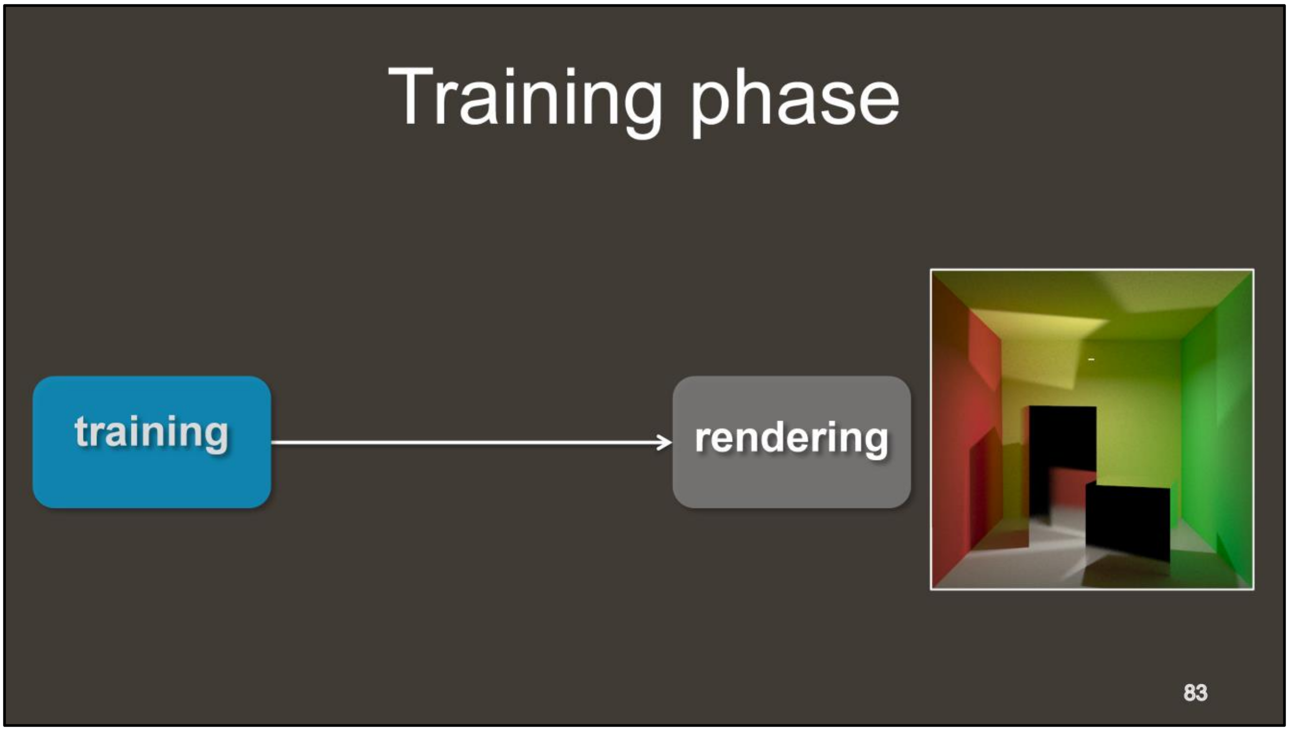
Similarly to irradiance caching, if there is no distribution that could be reused at the intersection point …



… a new distribution is trained and placed into the spatial cache.

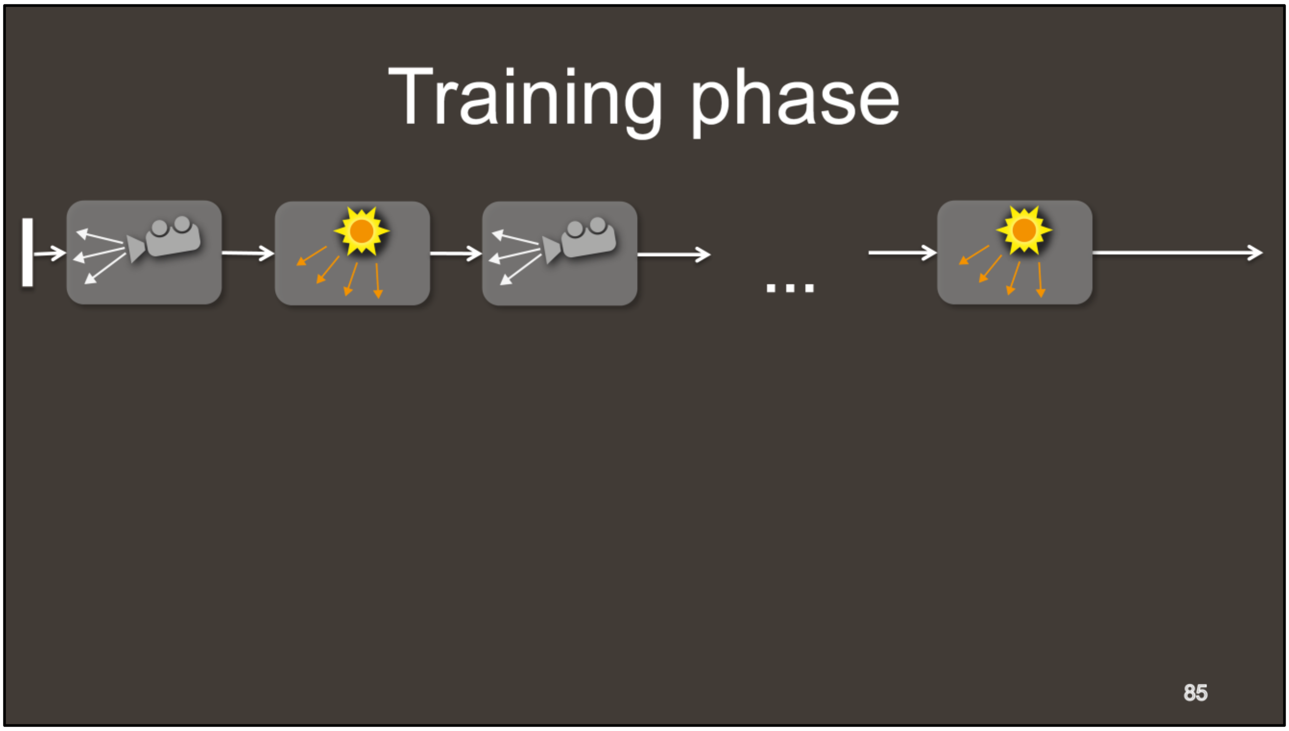


Now that I've finished the description of the guided path-sampling, …

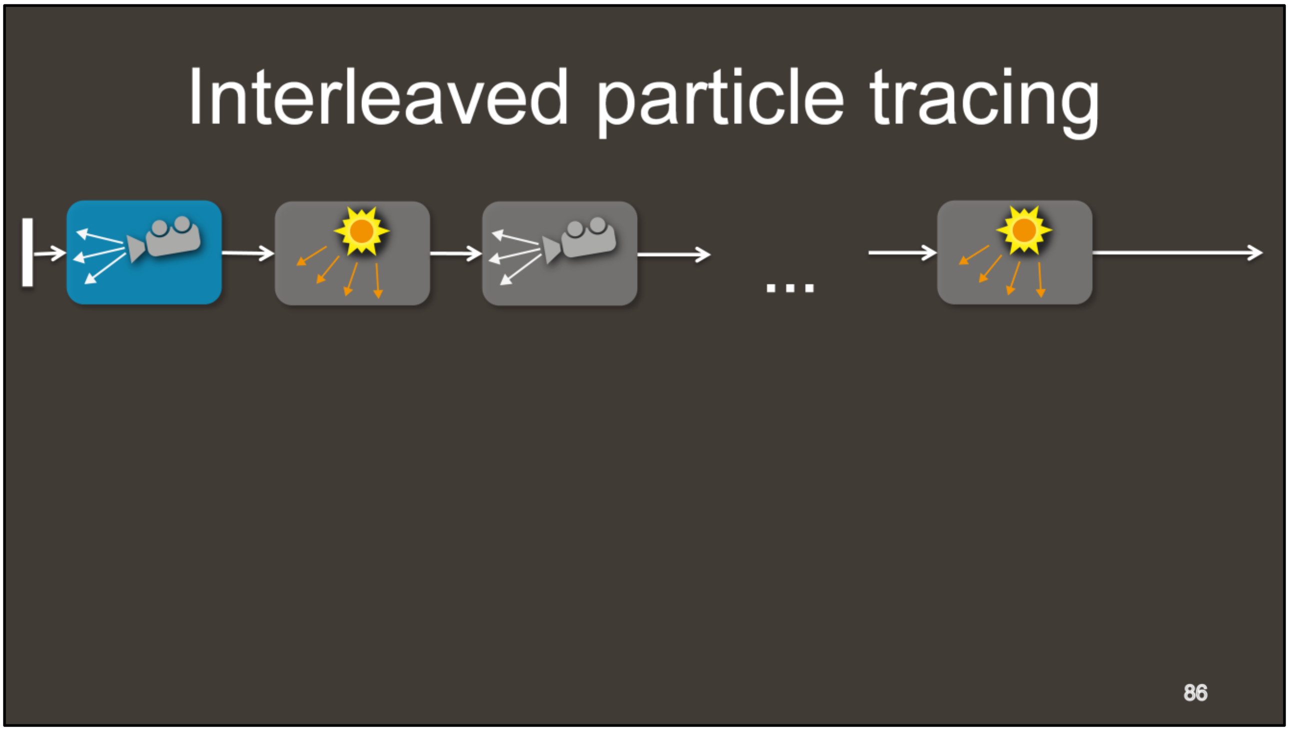


we can look in detail at the training phase.

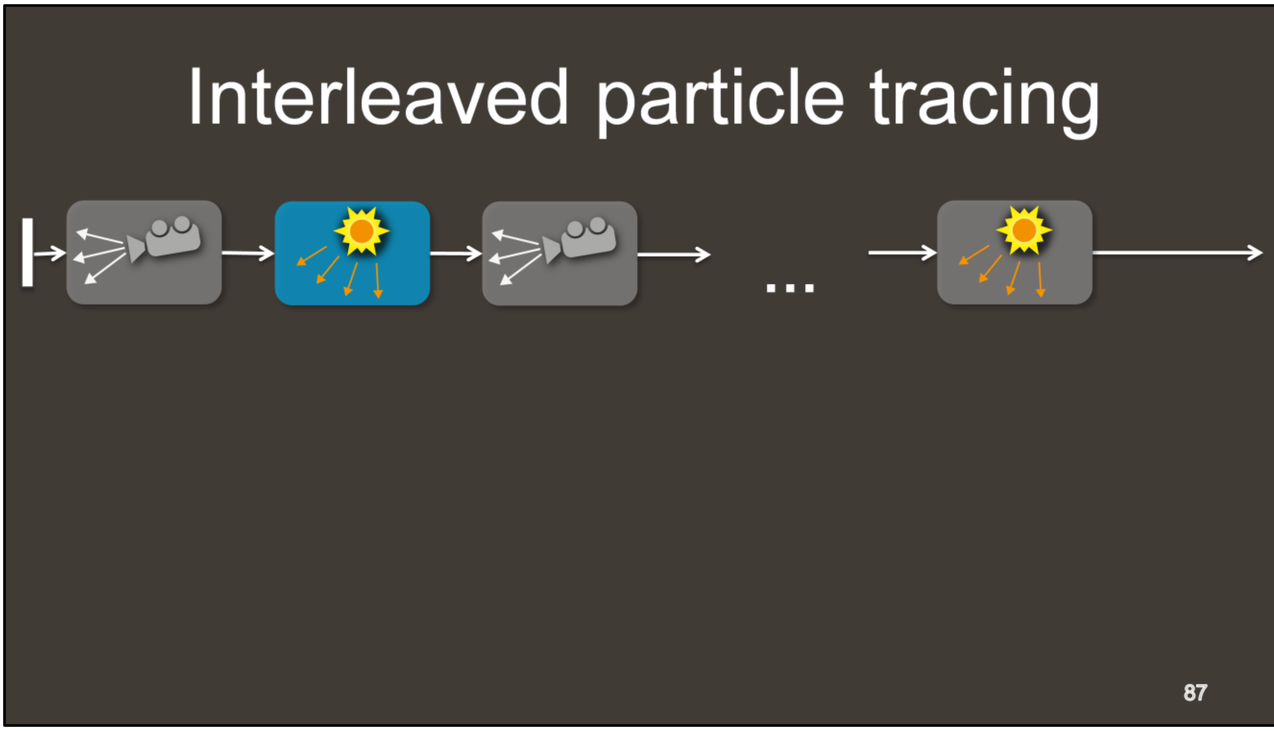
we can look in detail at the training phase.



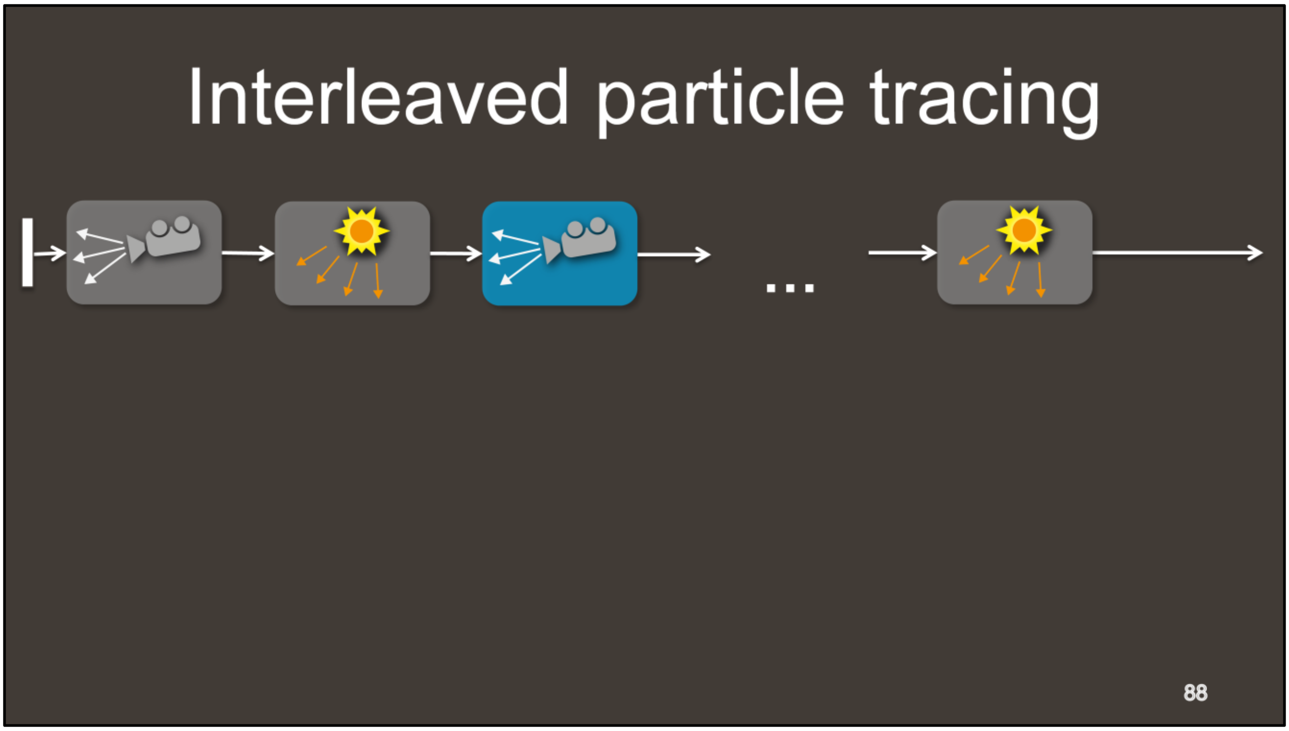
We interleave the particle tracing steps which is possible thanks to on-line learning.



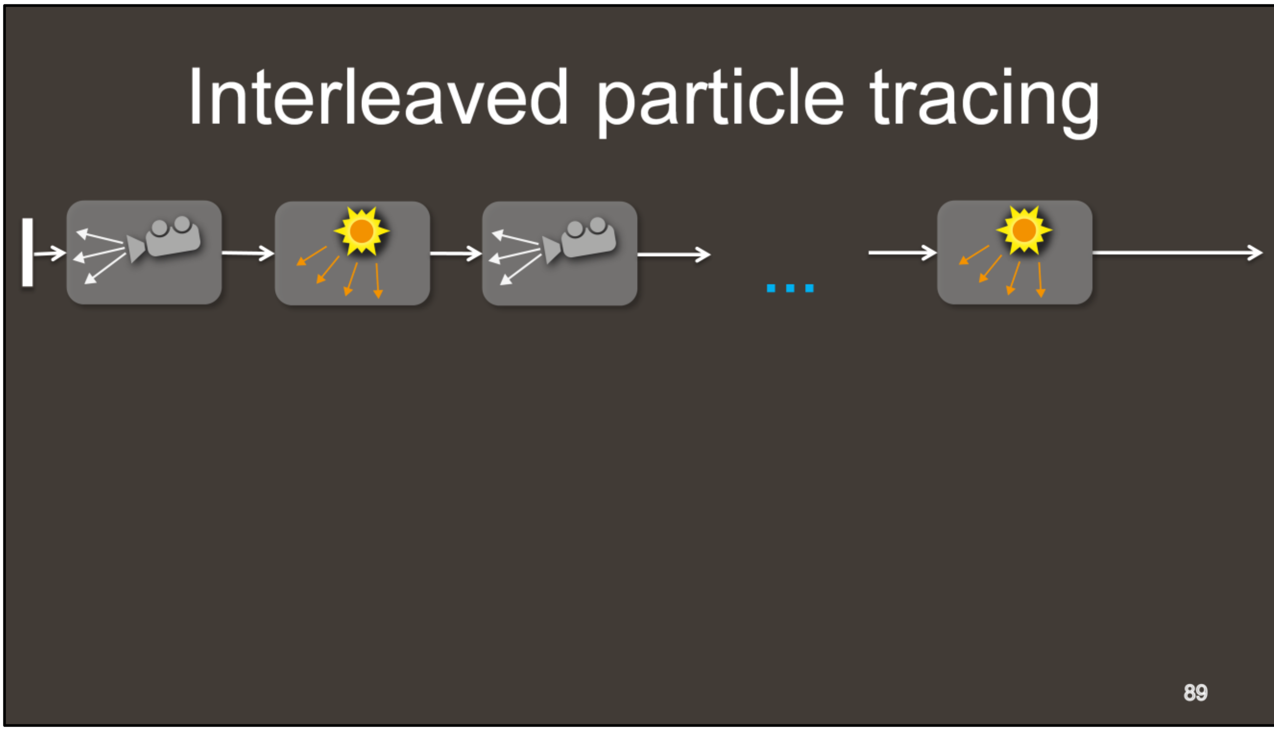
We start the training phase by blindly tracing importons using only importance sampling according to the BRDF.



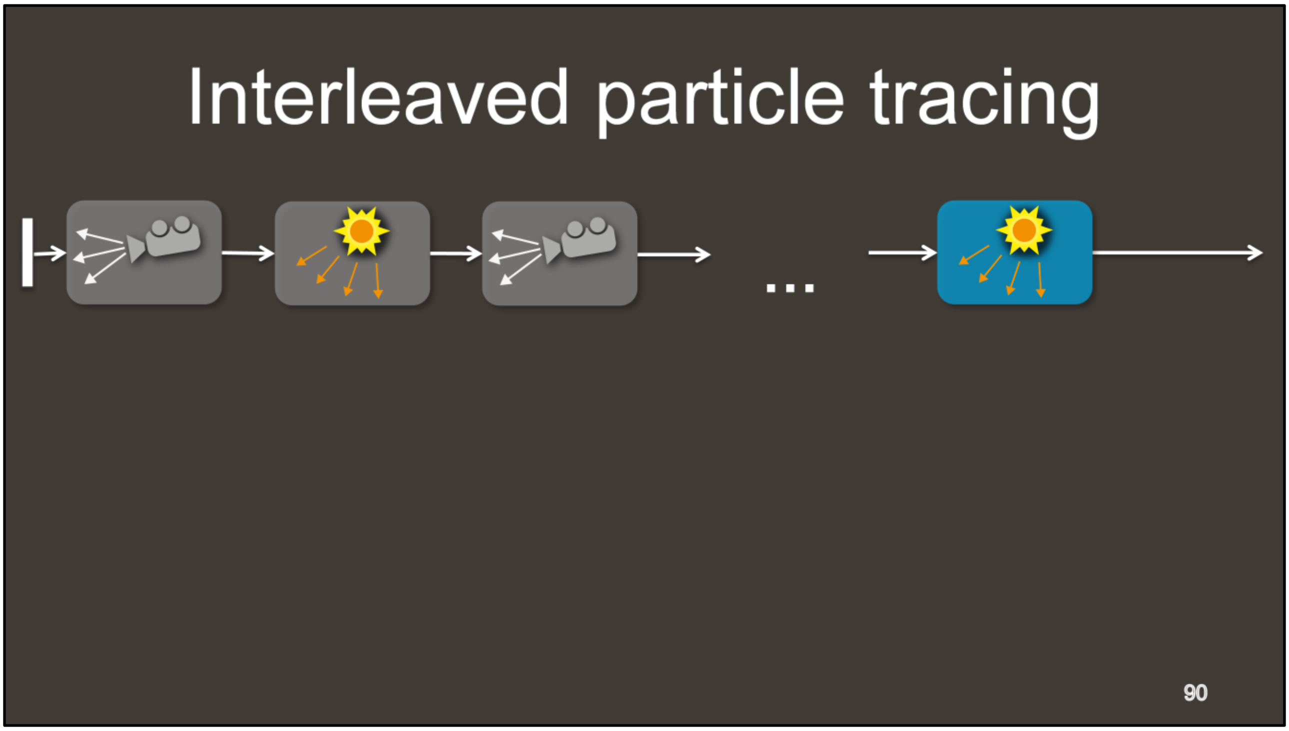
Then, we trace photons, guiding them towards visually important places.



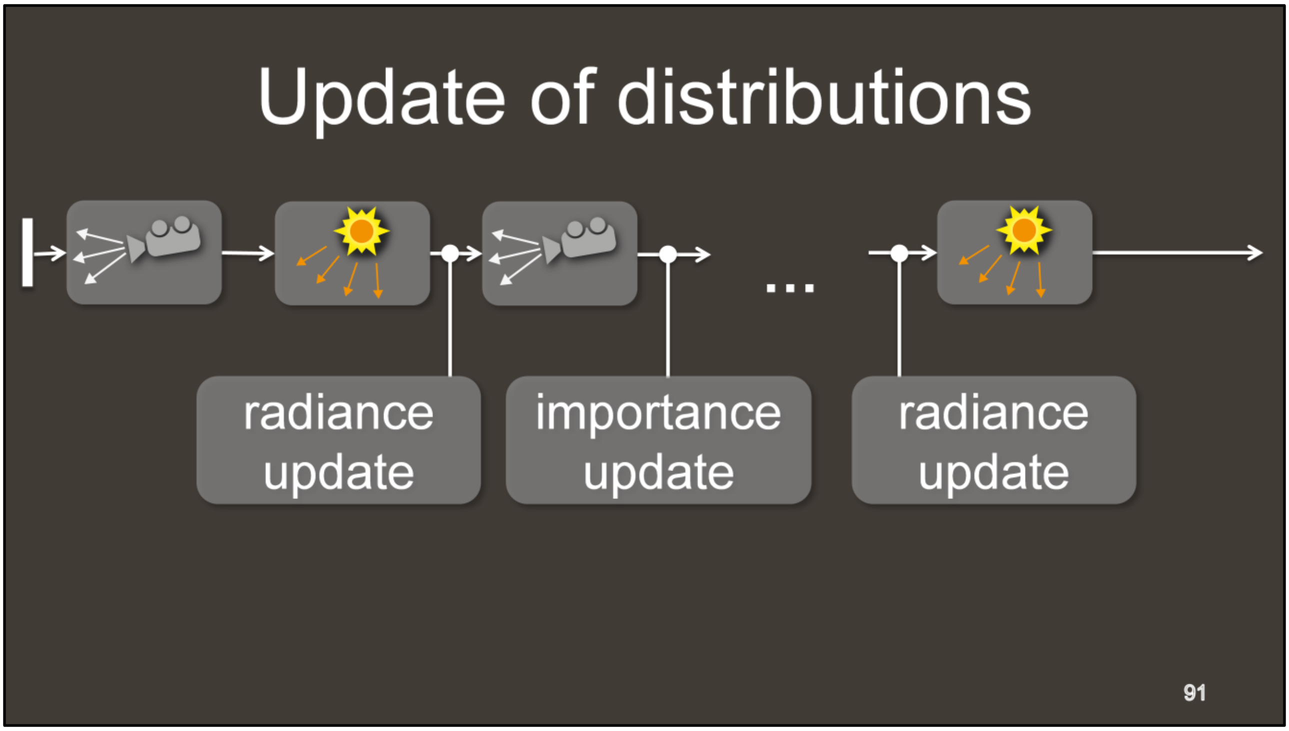
Then, we trace importons again and we guide them towards light sources.



This is repeated several times …



… until the training phase is concluded by photon tracing.



Between the particle tracing steps we use gathered particles to update the cached distributions.

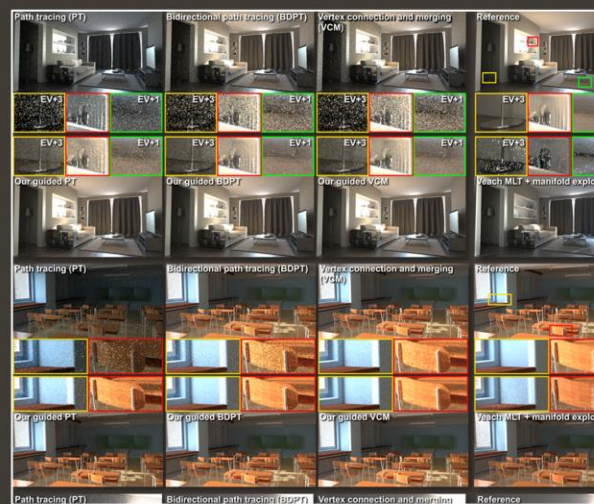
So the distributions are getting more refined and in turn provide better importance sampling for the subsequent tracing.

For details about the presented algorithms, please, refer to the paper.

Now I will present our results.

Results

- **Application to**
  - PT
  - BDPT
  - PPM *[Hachisuka 2008]*
  - VCM *[Georgiev et al. 2012]*
- Three scenes
- **Comparison to**
  - MLT + Manifold Exploration *[Veach and Guibas 1997], [Jakob and Marschner 2012]*



93

We have applied our method to [CLICK] PT, BDPT, PPM and VCM and we provide extensive comparison [CLICK] conducted on three different scenes.

[CLICK]
We compare the results also to Veach-style MLT with Manifold exploration.

Results

- Same-time (1h) comparisons
- Preprocessing time included
  - Took 4 to 10 minutes

94

However, I will show you only fraction of the results.

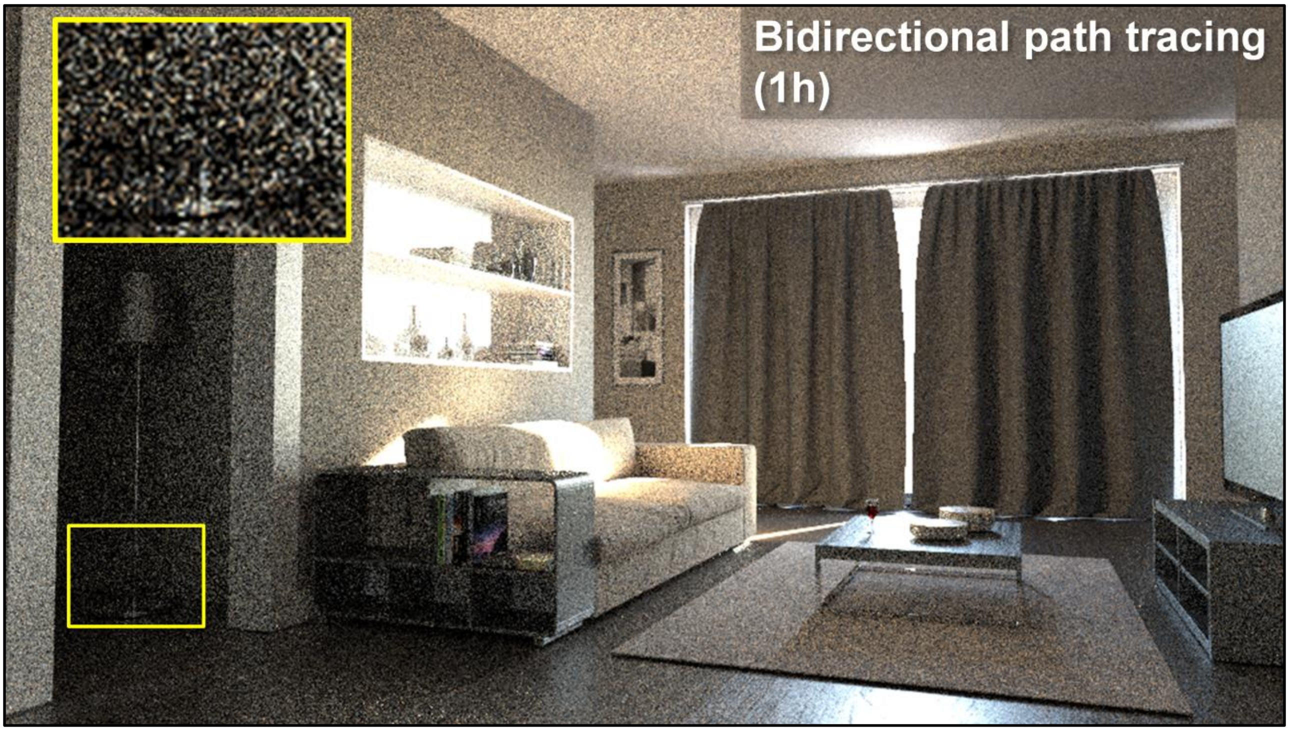
All the images were rendered for 1h including the preprocessing time.

This slide shows the Living room scene that I have presented at the very beginning.

[CLICK]

I will draw your attention to a small region that is especially difficult to render.

Note that this reference image was rendered by plain BDPT for 60 days. And you might still notice some noise.



This very noisy result was achieved with bidirectional path tracing after one hour.

Now I will show you the result after application of our guiding method.

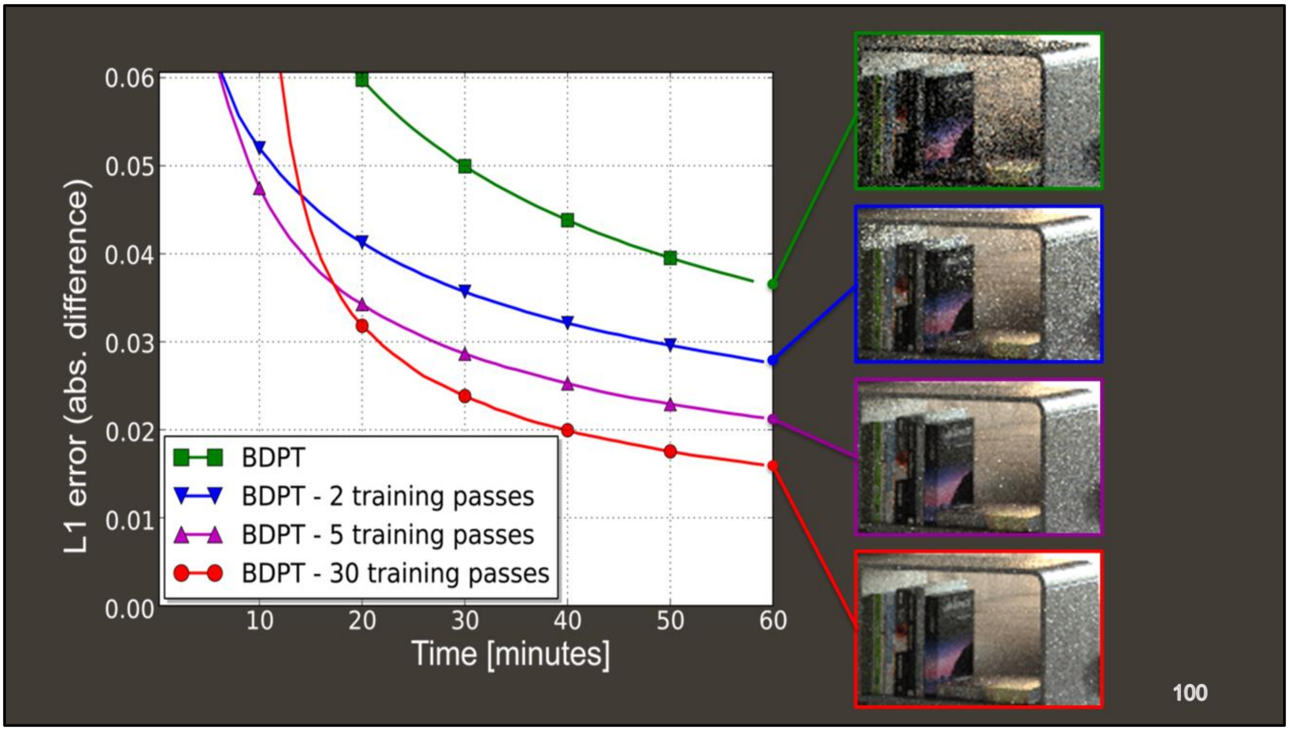
As you can see, the noise was substantially reduced.

However, you may also notice some residual high-frequency noise visible especially around the picture on the wall. We would like to address this in our future work by learning of distributions also during the rendering phase.

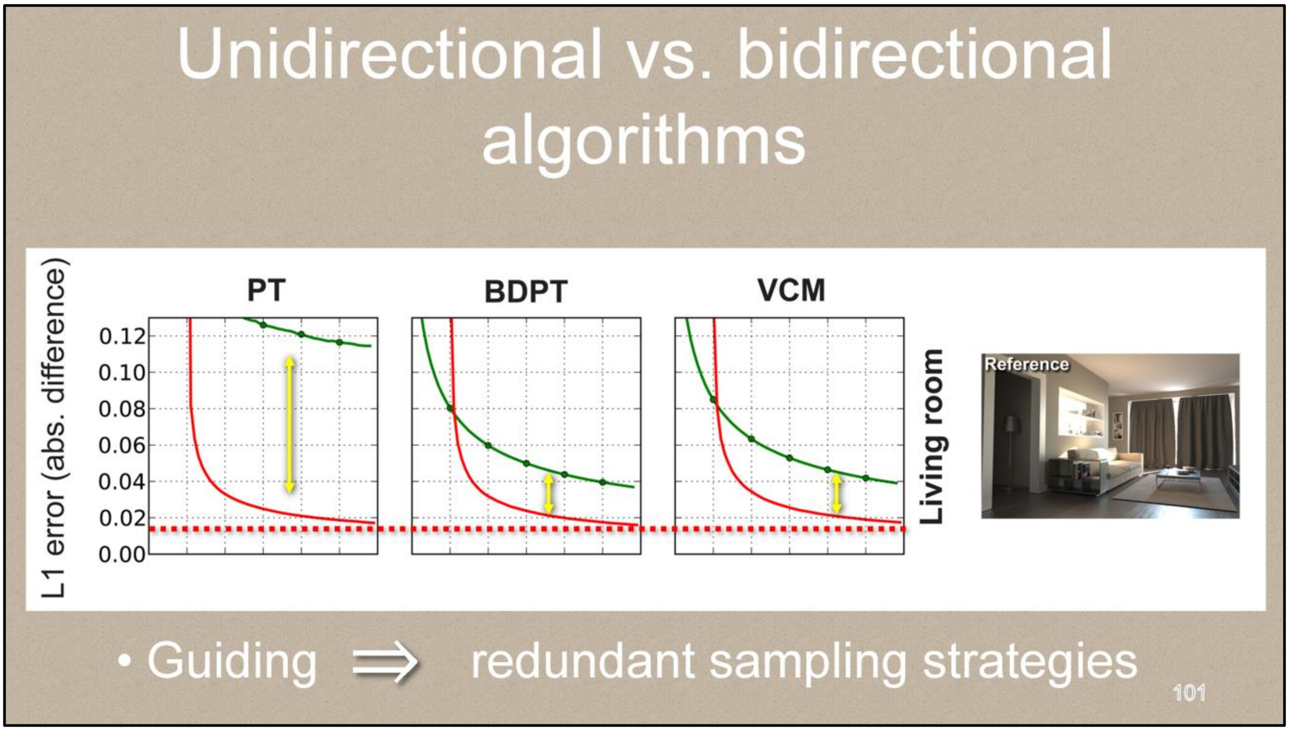
On another detail of the same scene I will demonstrate that the time spent on training is amortized by faster convergence of rendering.
[CLICK]
We use different number of training passes in the preprocessing.
[CLICK]
1 training pass corresponds to tracing importons followed by tracing photons.

You can see that more passes produced better result even when we account for the preprocessing time.

The improvement is visible in the insets …



… and is also underlined by L1 error plots.



Based on our results, we suspect that guiding might render some path sampling techniques in bidirectional algorithms less important.

[CLICK]

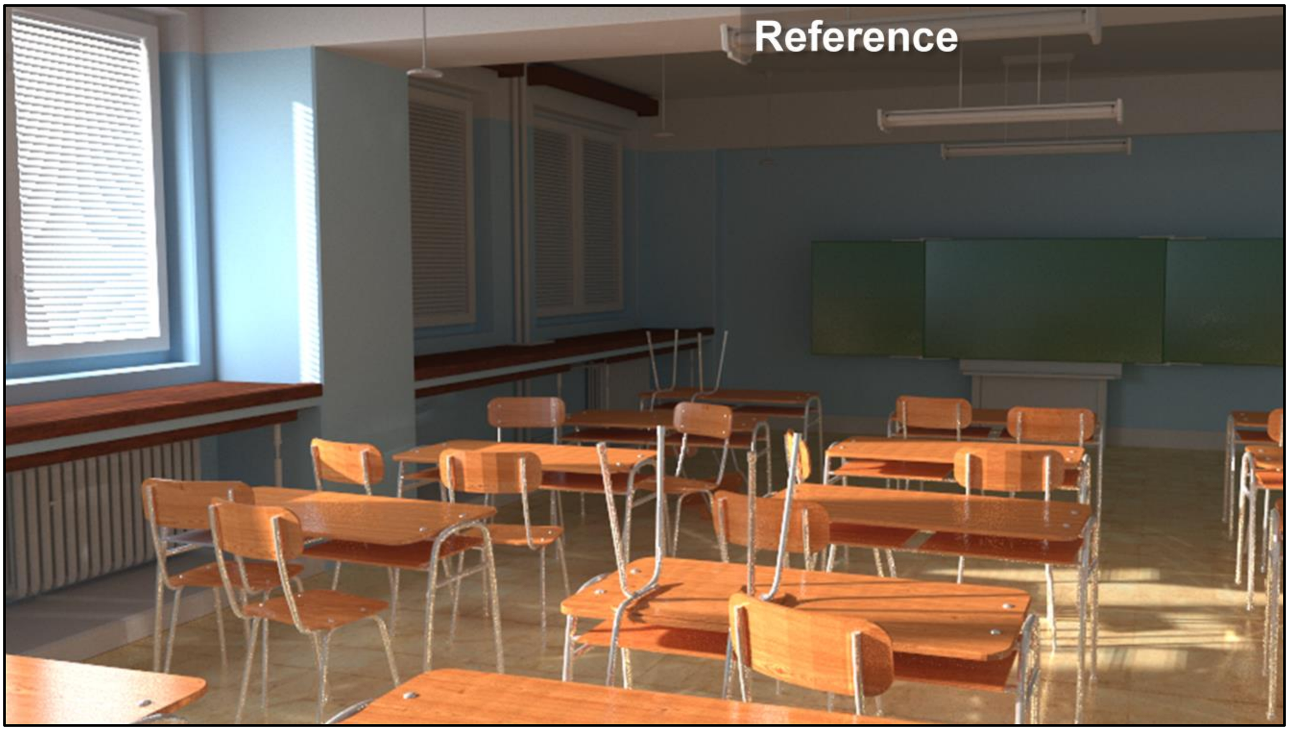
The L1 error plots correspond to the living room scenes rendered with three different algorithms. (Path tracing, Bidirectional path tracing and VCM). Green color is used for unguided methods while red depicts the guided algorithm.

[CLICK]

Clearly the highest relative improvement is always achieved for path tracing and …

[CLICK]

… guided bidirectional algorithms are only slightly superior to guided path tracing.

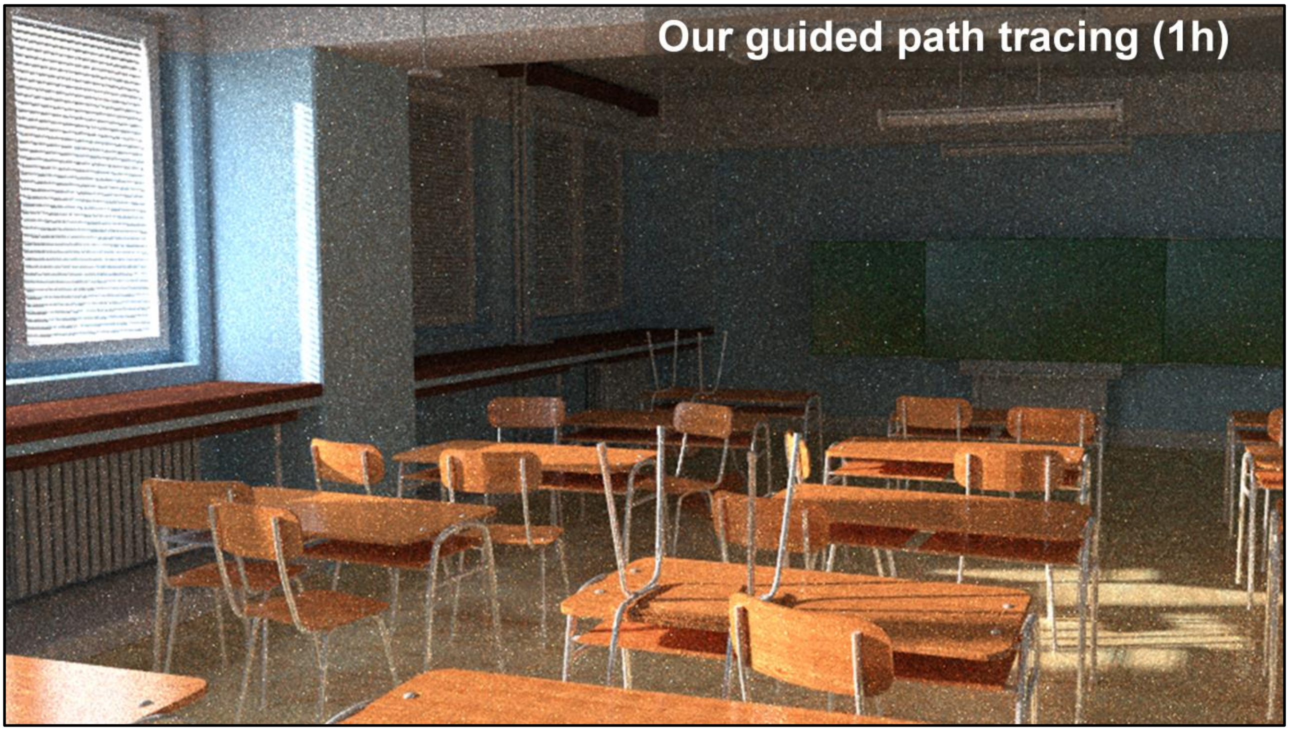


On another scene I will show the improvement achieved for simple path-tracing.

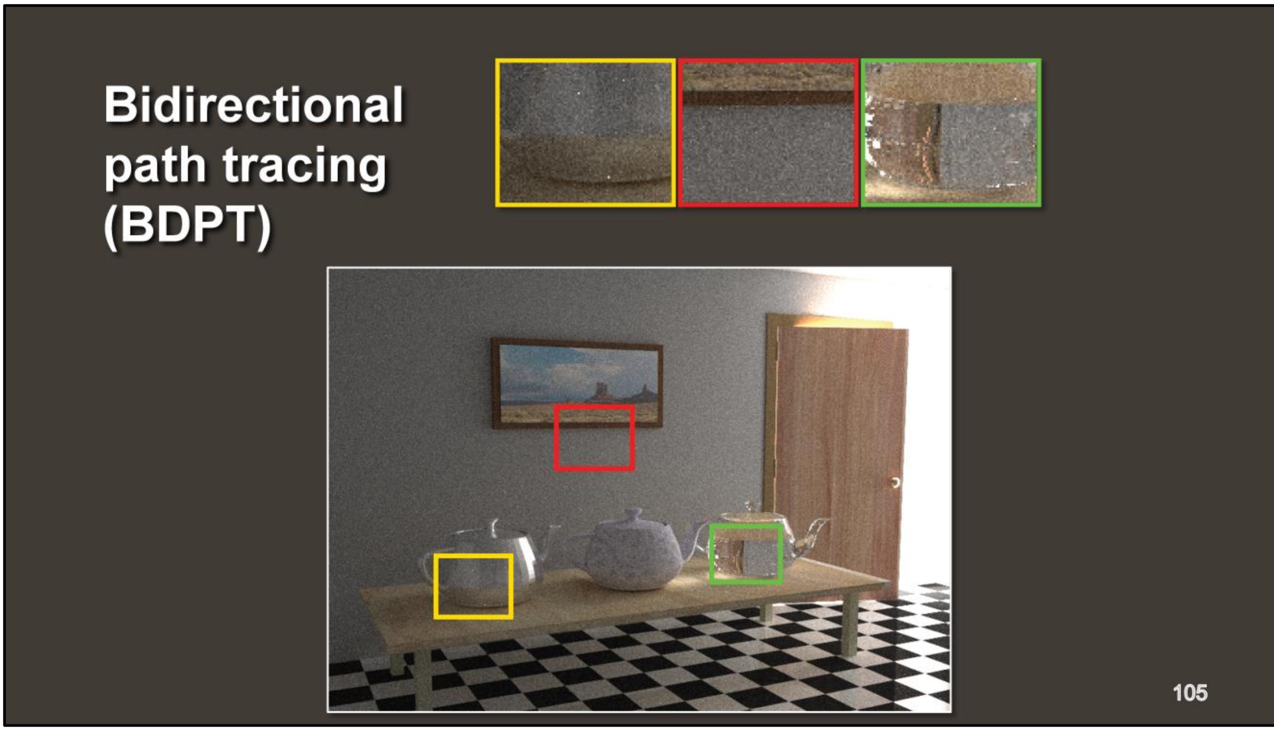
The sunlight reaches the scene after passing through windows with jalousies and undergoes complex interactions in the classroom.

The simple path tracing fails to render the scene. In fact, all the sunlight is almost completely missing.

Now I am going to show you the result after the application of our guiding method.



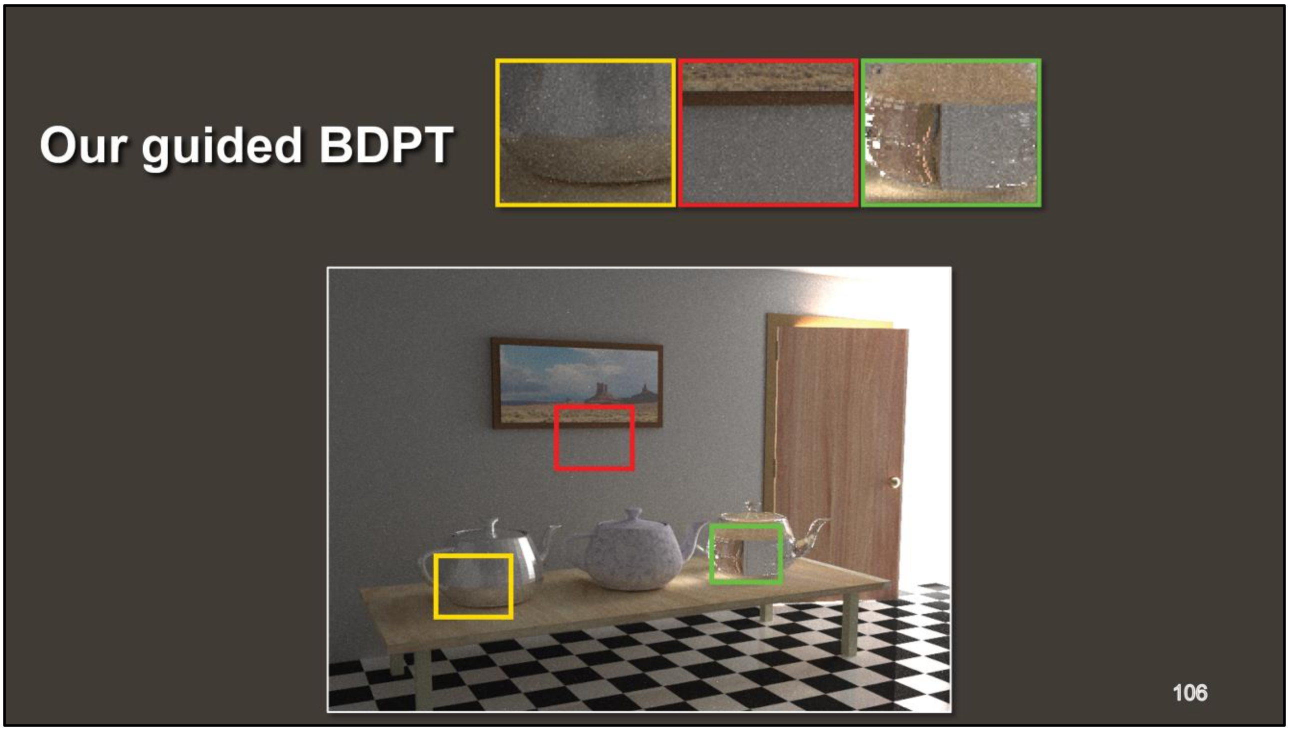
Again, the noise reduction is substantial and even higher then in the case of bidirectional path tracing.



On this famous scene I show the result using bidirectional path tracing.

The light is coming from the other room through a small door slit.

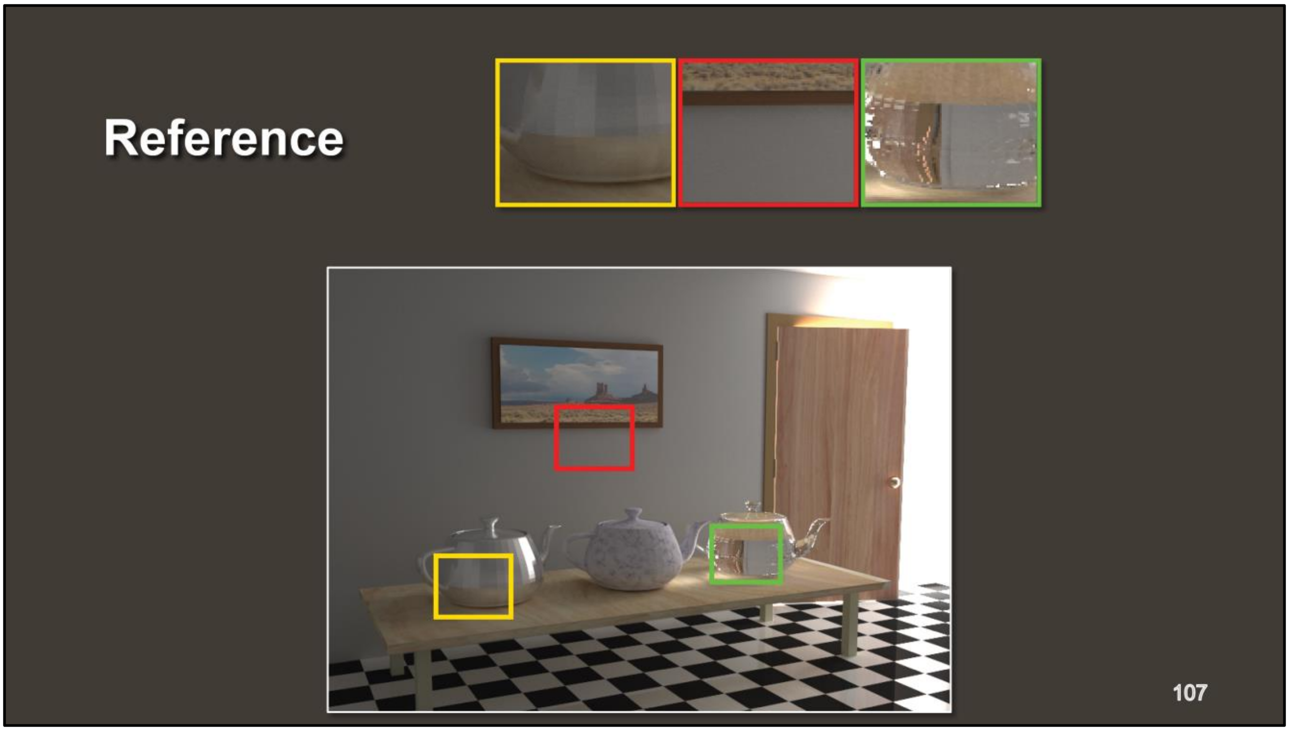
This is our guided version and finally the reference.



On this famous scene I show the result using bidirectional path tracing.

The light is coming from the other room through a small door slit.

This is our guided version and finally the reference.



On this famous scene I show the result using bidirectional path tracing.

The light is coming from the other room through a small door slit.

This is our guided version and finally the reference.

Conclusion

- Parametric mixtures
- Importance sampling
- On-line training
- Scenes with difficult visibility

108

To sum up, we propose the use of a parametric mixture model to represent directional distributions.

These distributions are trained in the preprocessing phase and used for importance sampling in the rendering.

The proposed model allows on-line training from an infinite stream of particles which in turn enables rendering scenes with difficult visibility.

Limitations and future work

Limitations

- Overhead (from caching)

Future Work

- Adaptive refinement in rendering
- Combination with Metropolis Ligth Transport

109

However, the overhead of our method, that comes from querying the distribution cache, may offset its advantages in simple scenes.

[CLICK]

In the future, we would like to refine the distributions adaptively also during the rendering phase.

We also plan to combine our method with MLT.

We think that these two approaches could complement each other to achieve further benefits.

The source code of our method is publicly available at the project home page.

[CLICK]

The method is implemented as a library and its sample usage for path tracing is available in the Mitsuba renderer.

We would like to thank the following people and institutions for their help and funding

(We also thank to anonymous reviewers for their thoughtful comments.)

CHARLES UNIVERSITY PRAGUE

faculty of mathematics and physics

Thank you!

http://cgg.mff.cuni.cz/~jirka/papers/2014/olpm/index.htm

Thank you for your attention.

Additional Slides

113

Guiding emission

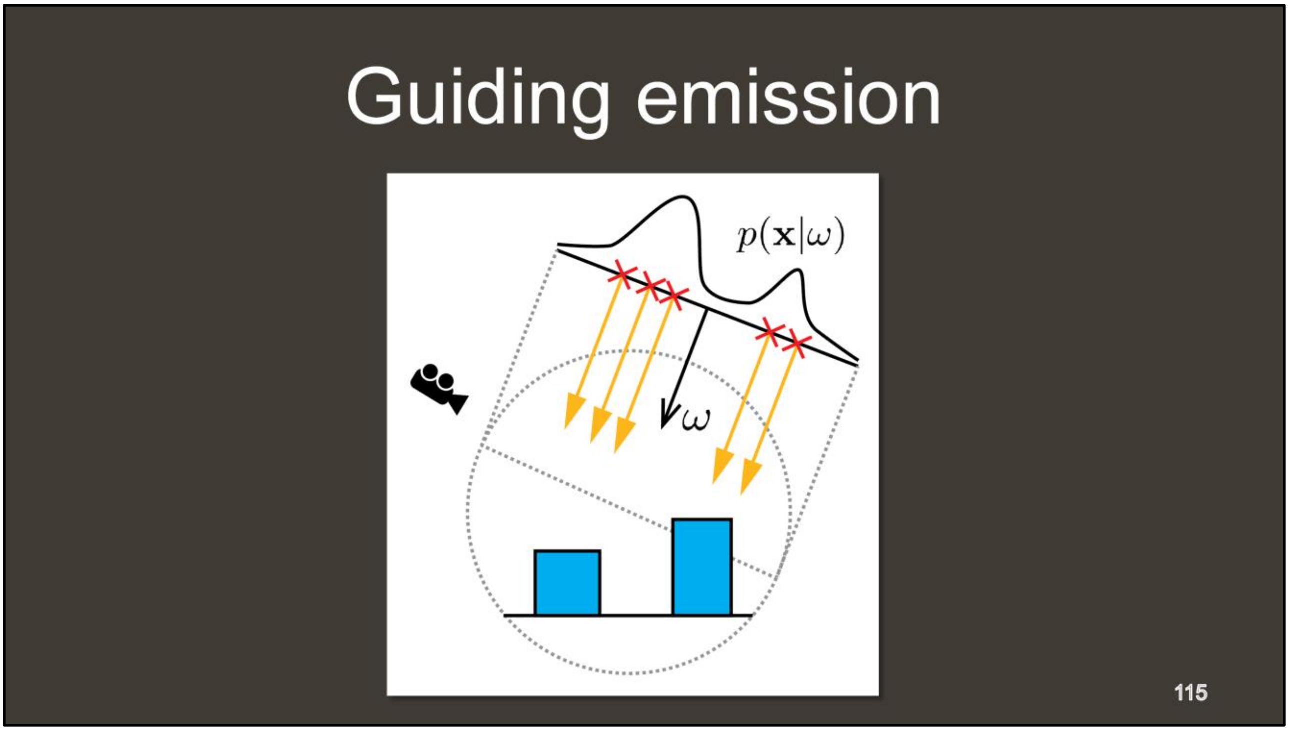
- Environment light source
- One progressive framework:
  - Scattering and emission

114

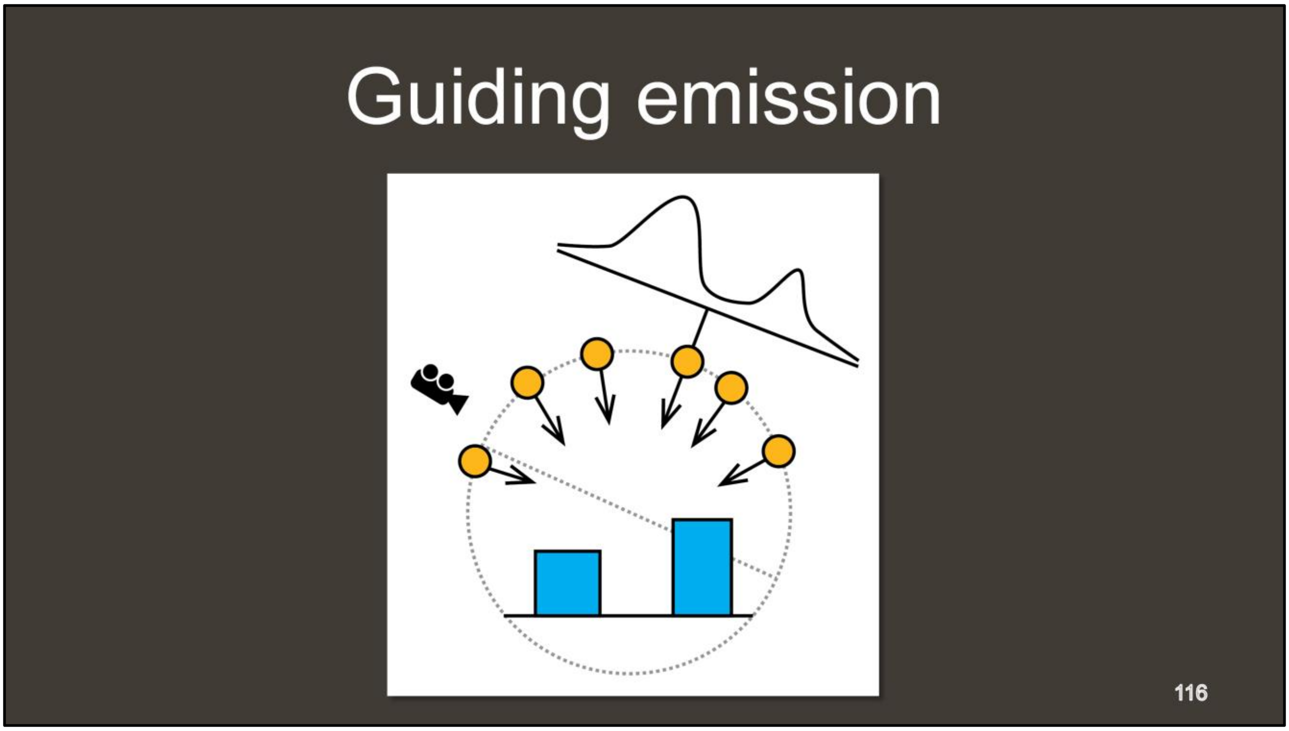
We use our method also for guiding of emission sampling from environment light source.

In other words, we drive the emission by visual importance.

Both guiding of scattering and the emission fit nicely into the same framework.

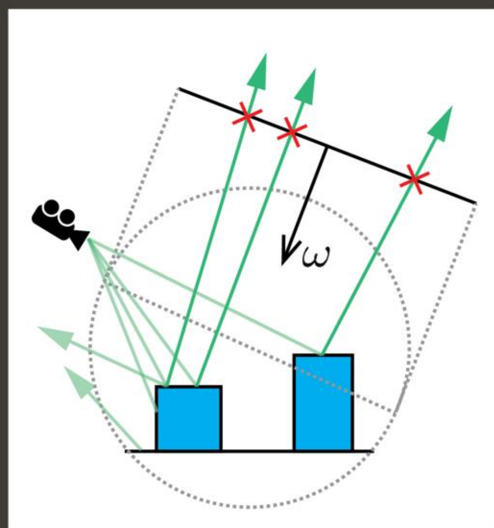


Our Gaussian mixtures for sampling starting position are conditioned by a direction omega and are defined on a disk perpendicular to omega.



To allow progressive training, we store the distributions in a directional cache.

Guiding emission

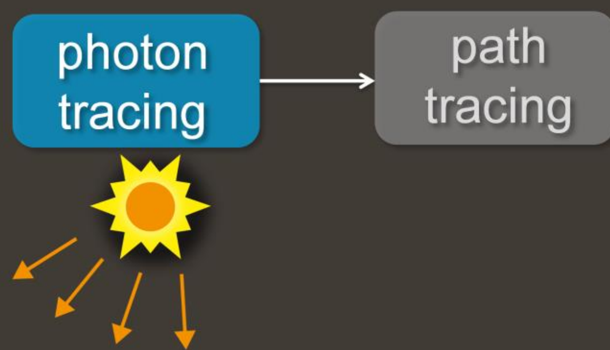


117

After every particle tracing step, distributions are trained from importons that left the scene.

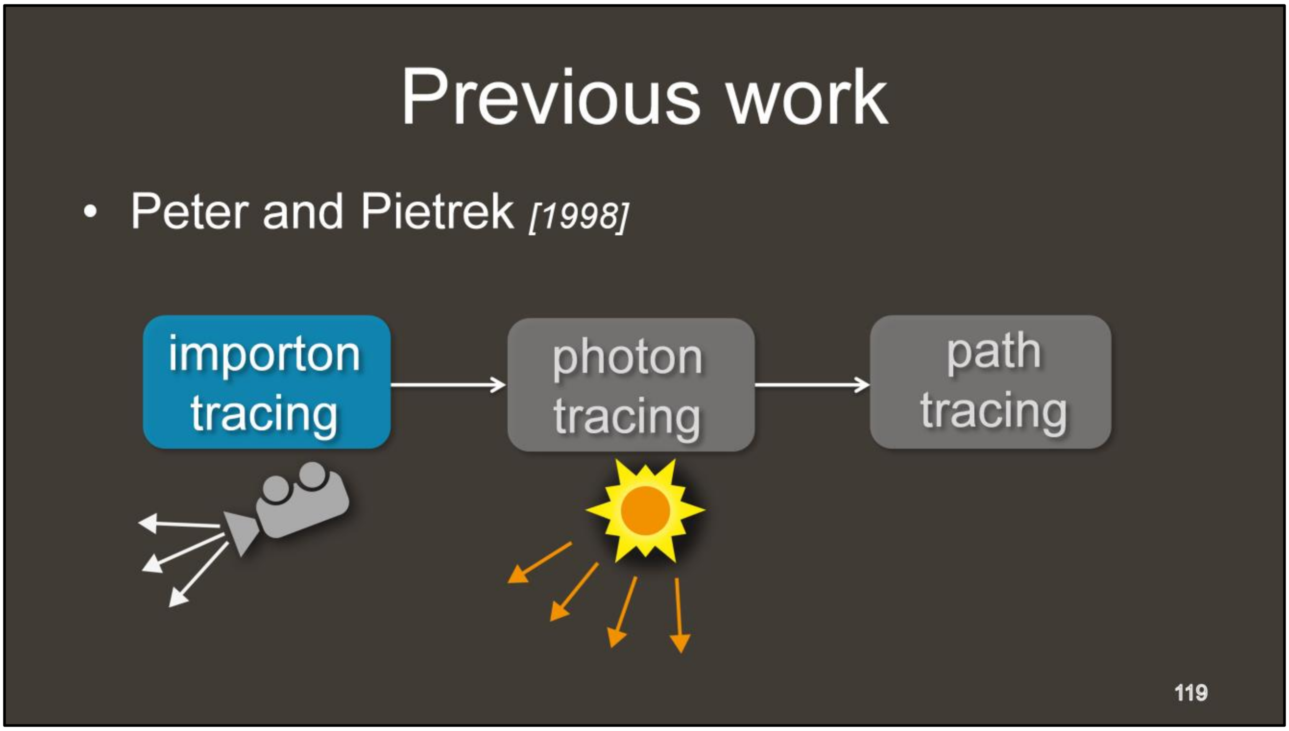
Previous work

- Peter and Pietrek *[1998]*

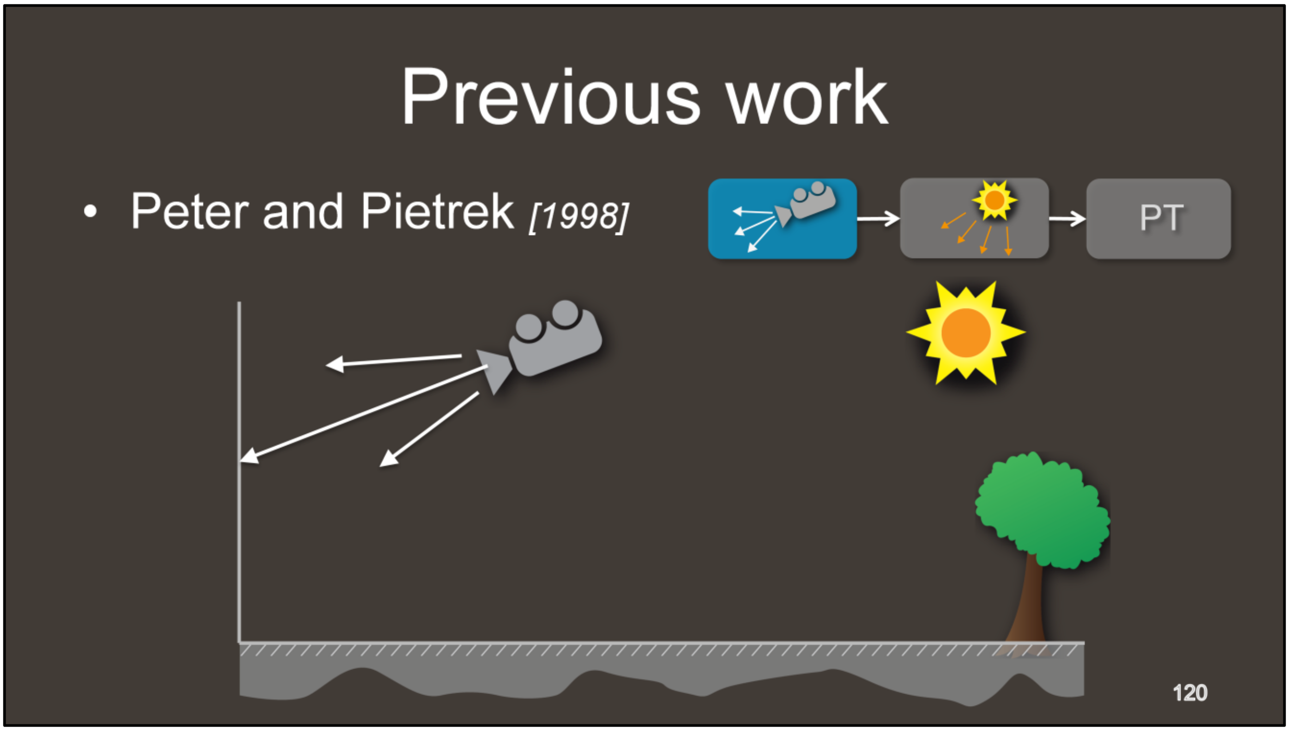


118

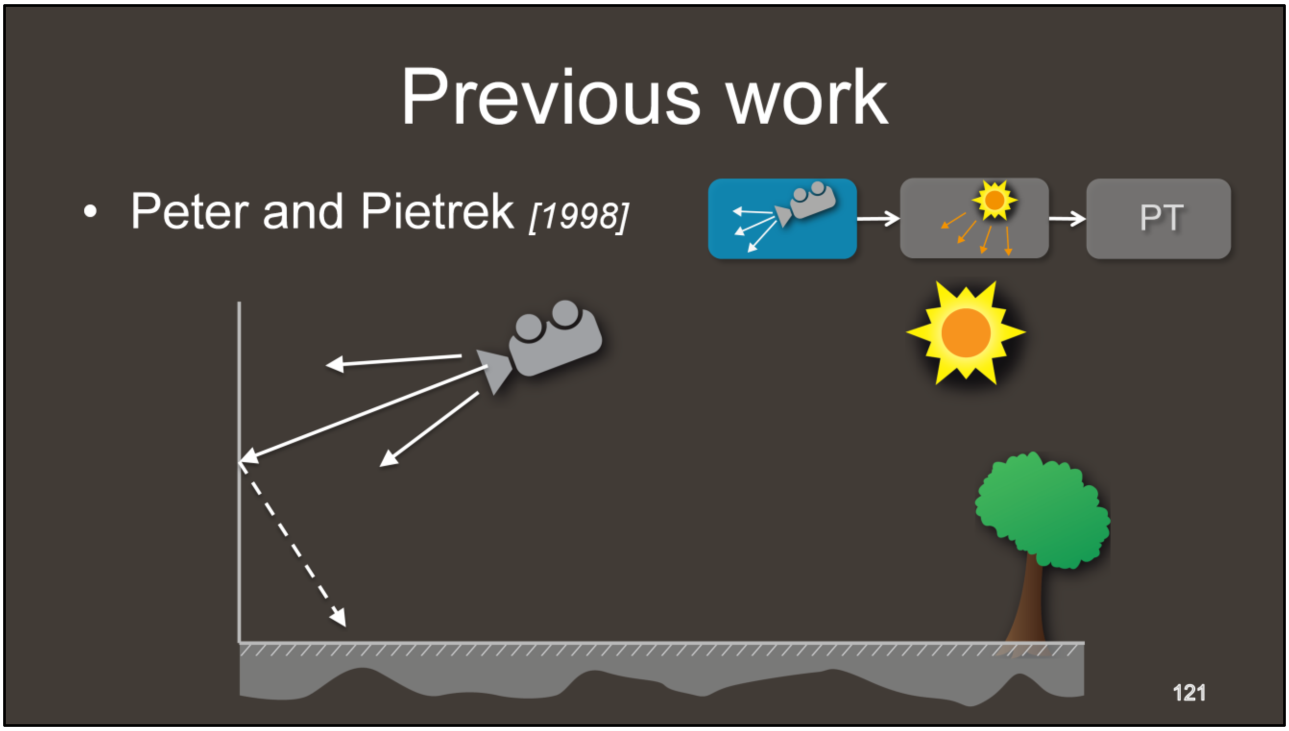
The work of Peter and Pietrek [1998] extended Jensen’s idea …

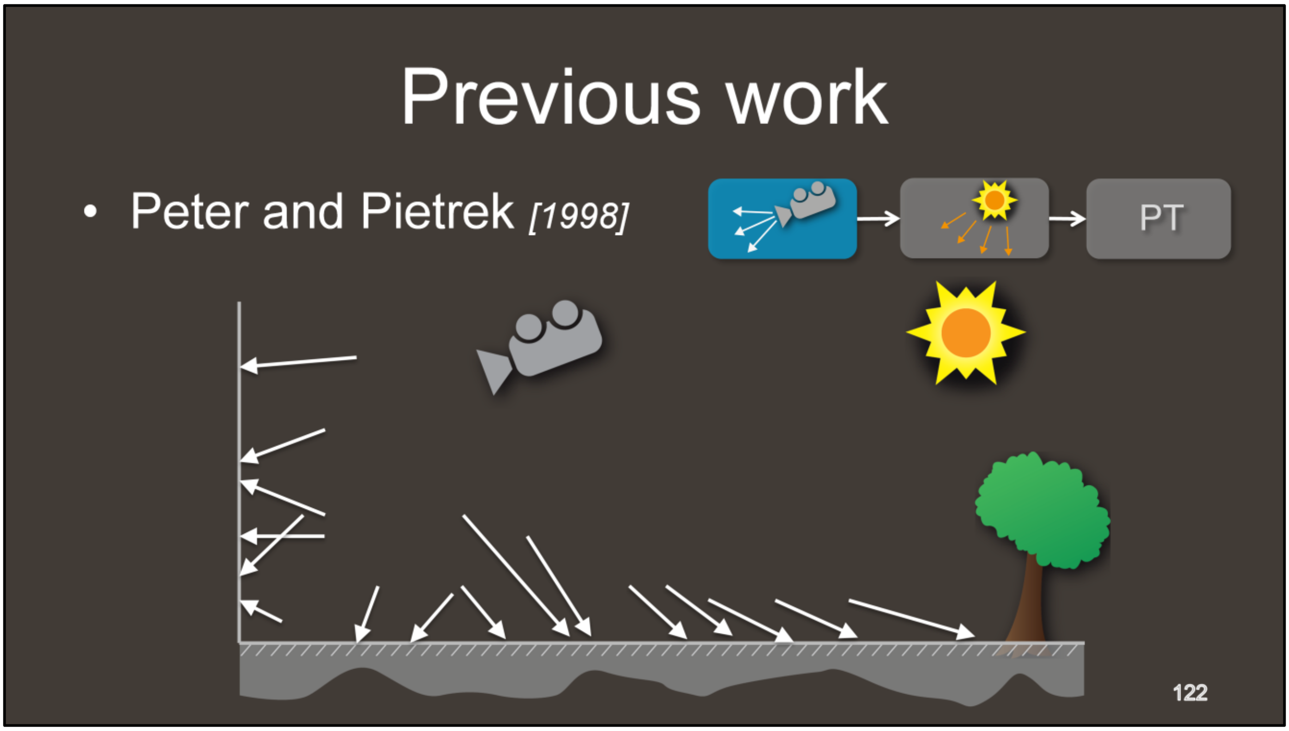


… by using camera particles – importons – to guide photons to visually important places.

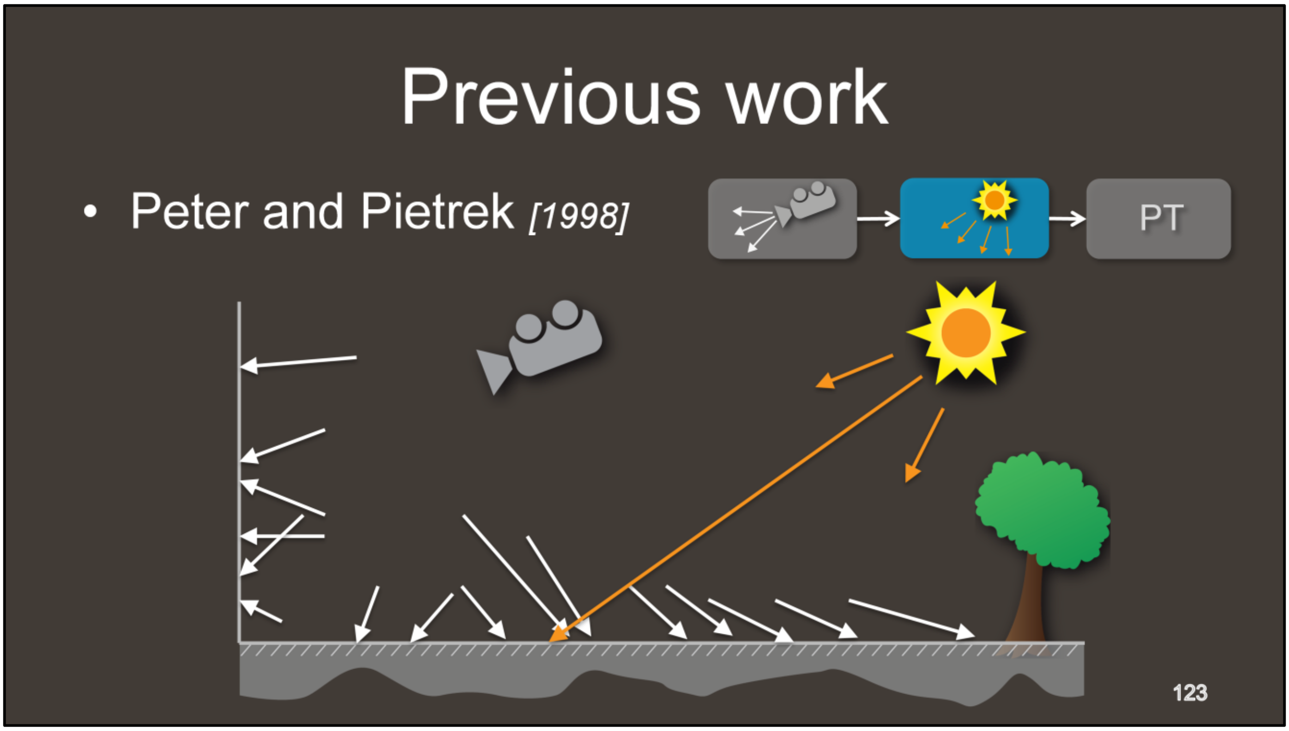


The algorithm starts by tracing particles from the camera – called importons.

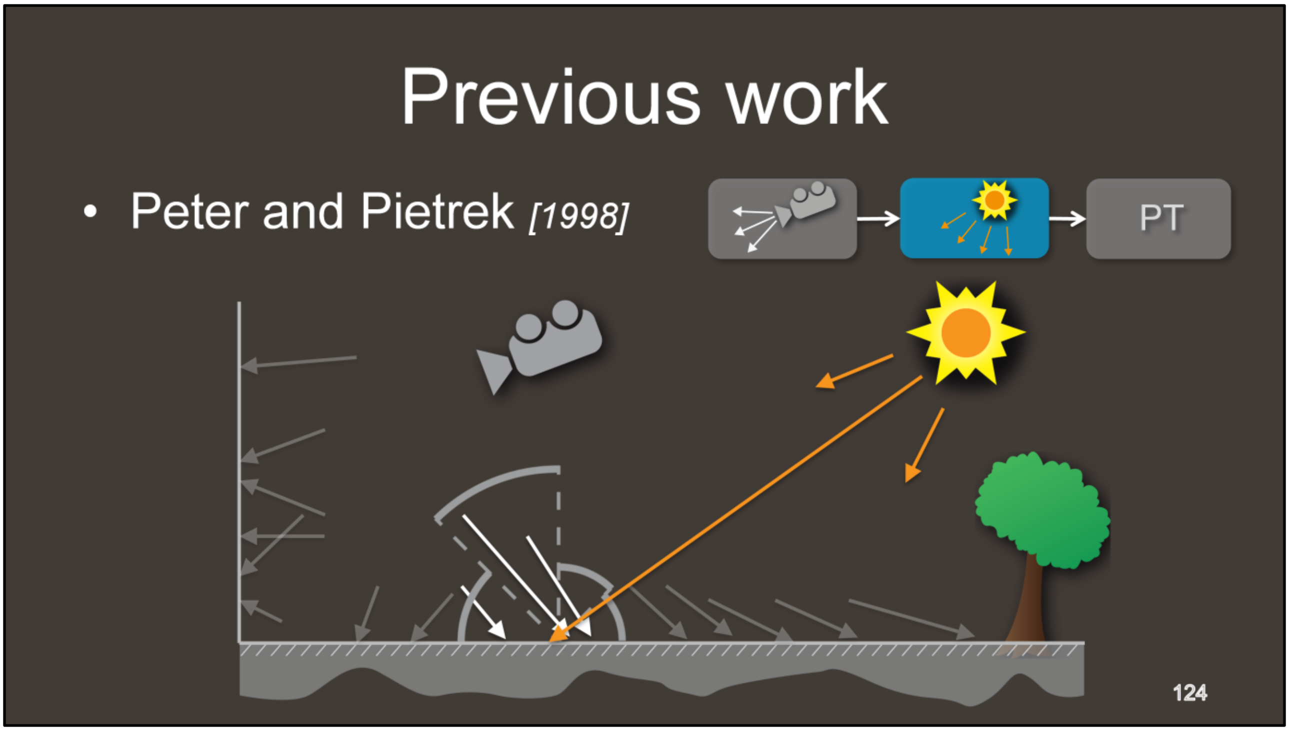

Previous work
• Peter and Pietrek [1998]
PT
121



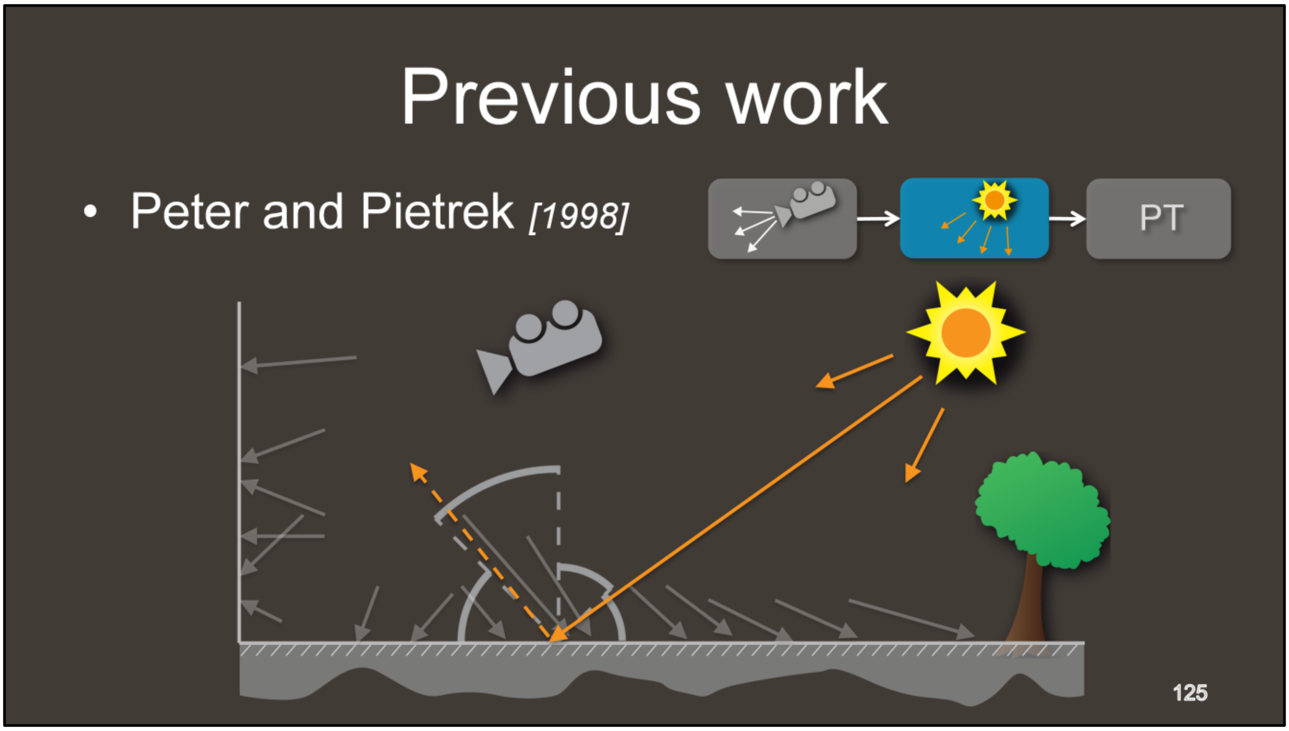
Then the photon tracing pass uses the importons …



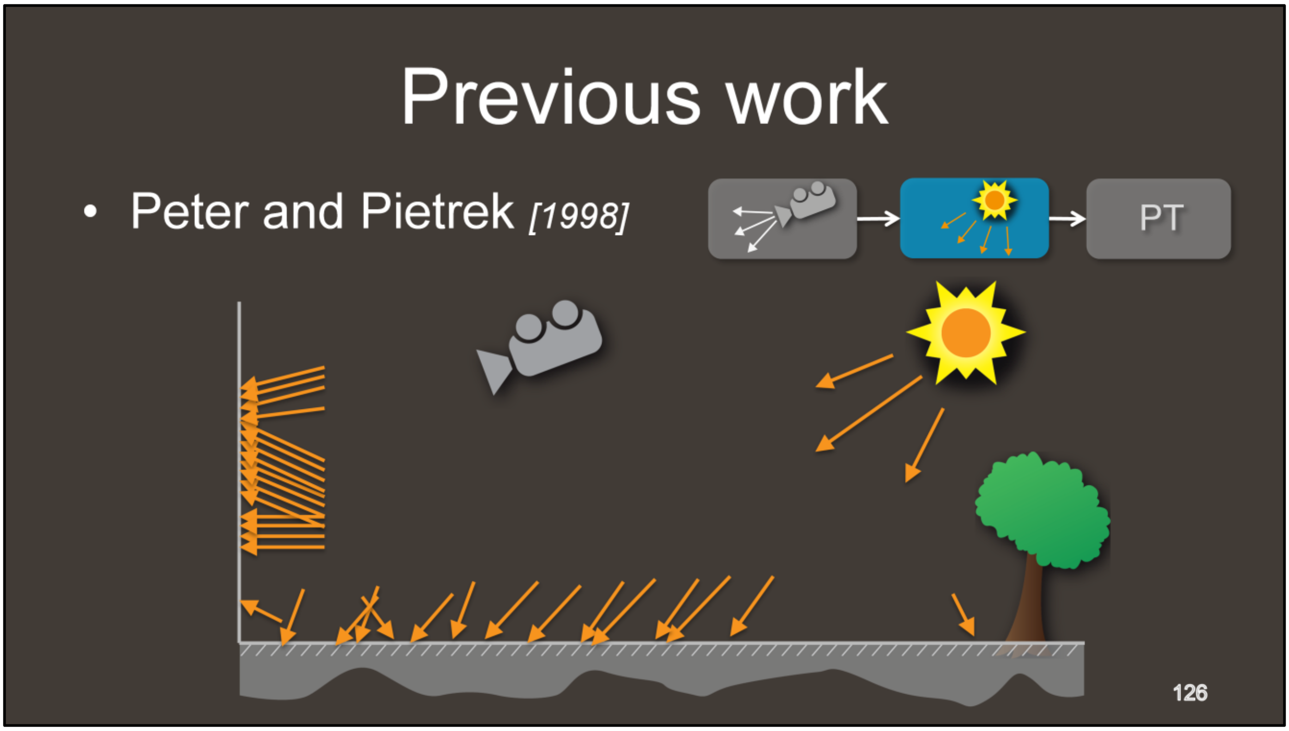
Then the photon tracing pass uses the importons …



… to reconstruct the visual importance distributions for scattering.



Thus the photons are guided towards the camera and during the path-tracing phase allow …



for better reconstruction of radiance in visually important places.