UNIFYING POINTS, BEAMS, AND PATHS IN VOLUMETRIC LIGHT TRANSPORT SIMULATION

Jaroslav Křivánek
Charles University in Prague

Iliyan Georgiev
Light Transportation Ltd.

Toshiya Hachisuka
Aarhus University

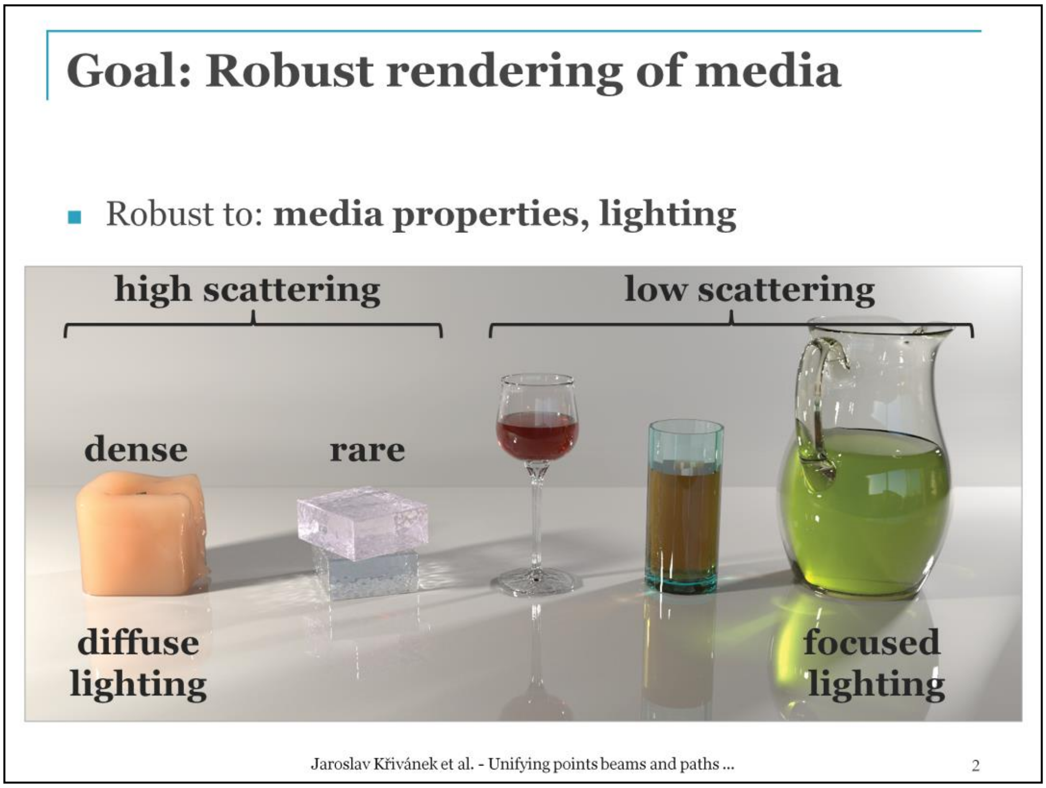
Petr Vévoda
Charles University in Prague

Martin Šik
Charles University in Prague

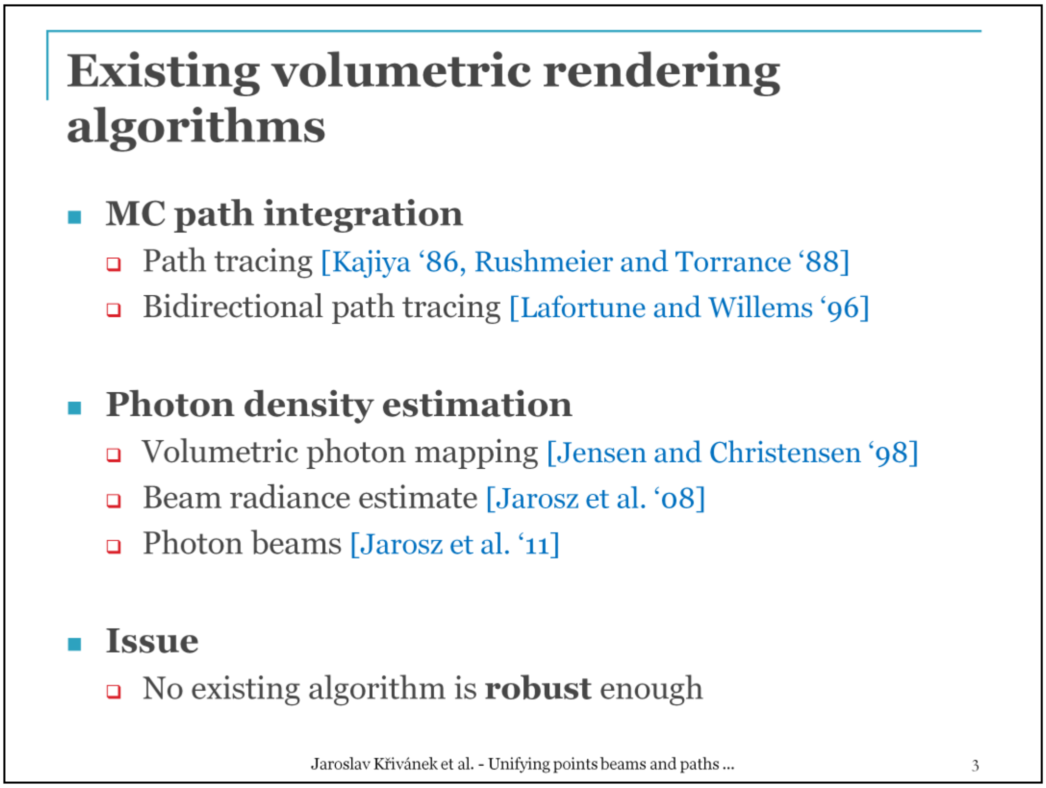
Derek Nowrouzezahrai
Univesity of Montreal

Wojciech Jarosz
Disney Research Zurich

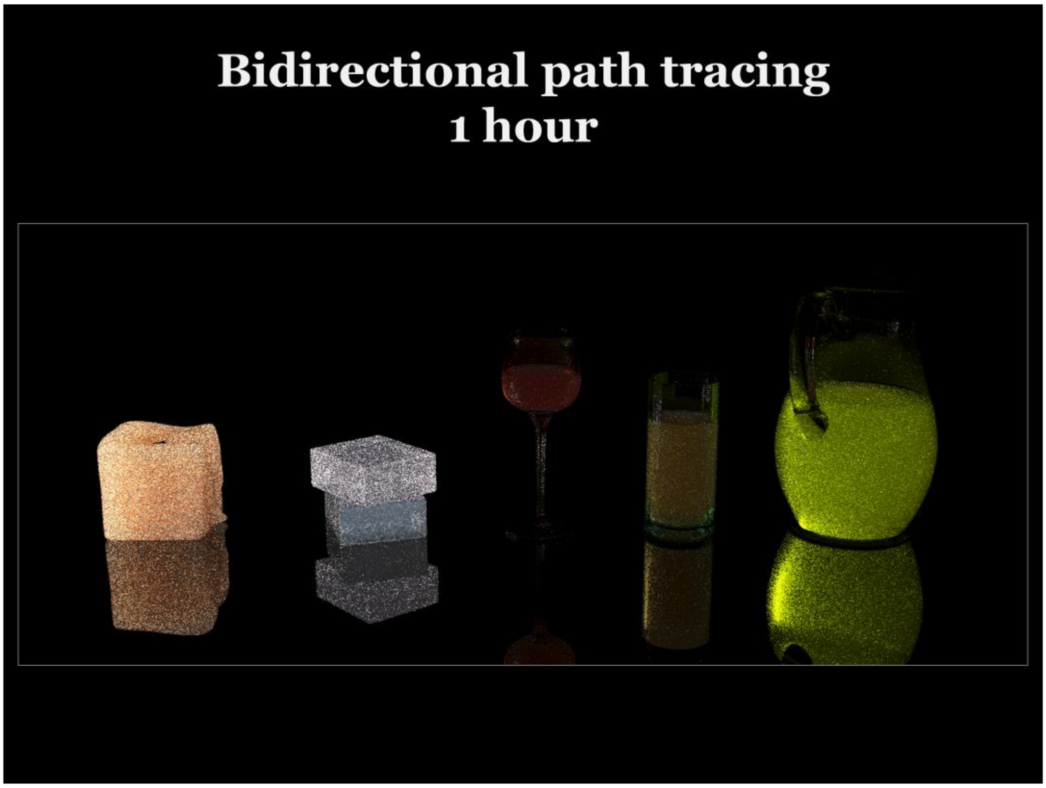
CHARLES UNIVERSITY
AARHUS UNIVERSITY
Université de Montréal
Disney Research, Zurich



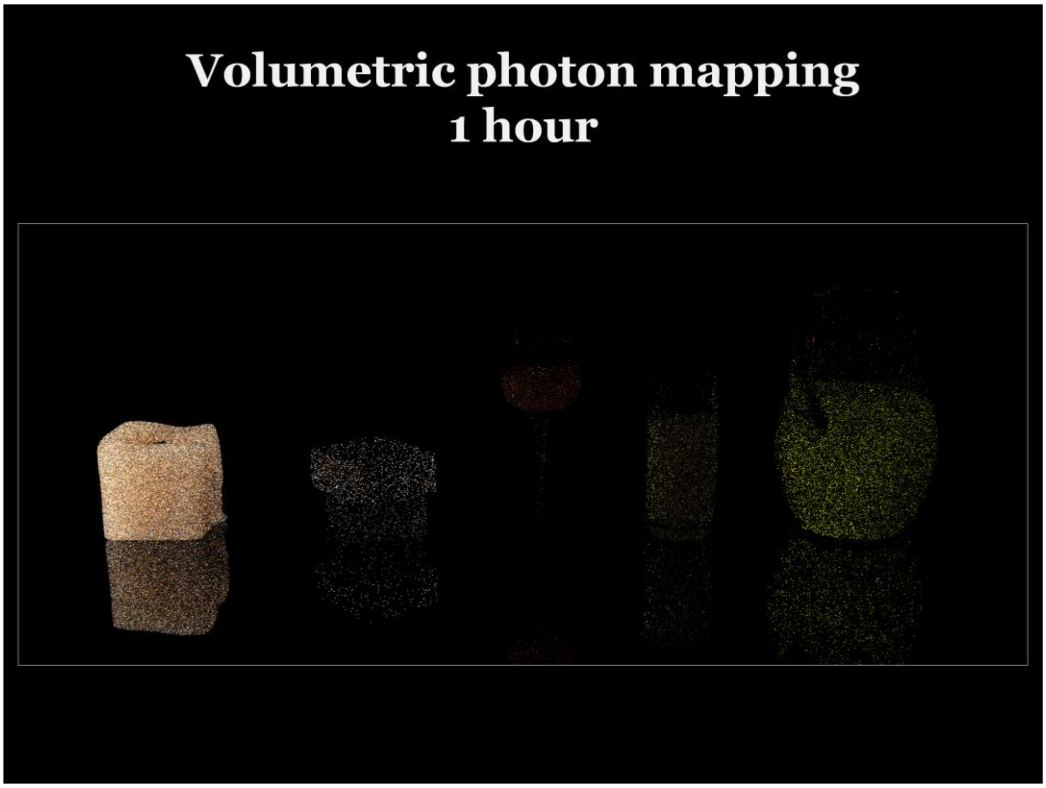
- In this work, we aim to render participating media in a manner that is robust to media properties and to lighting.
- We want to handle optically dense or rare media with high or low scattering albedo.
- We want to handle diffusive multiple scattering (as in subsurface scattering) or highly focused lighting (as in volumetric caustics).
- The algorithm we've developed has all these features and it was actually used to render the image shown here.



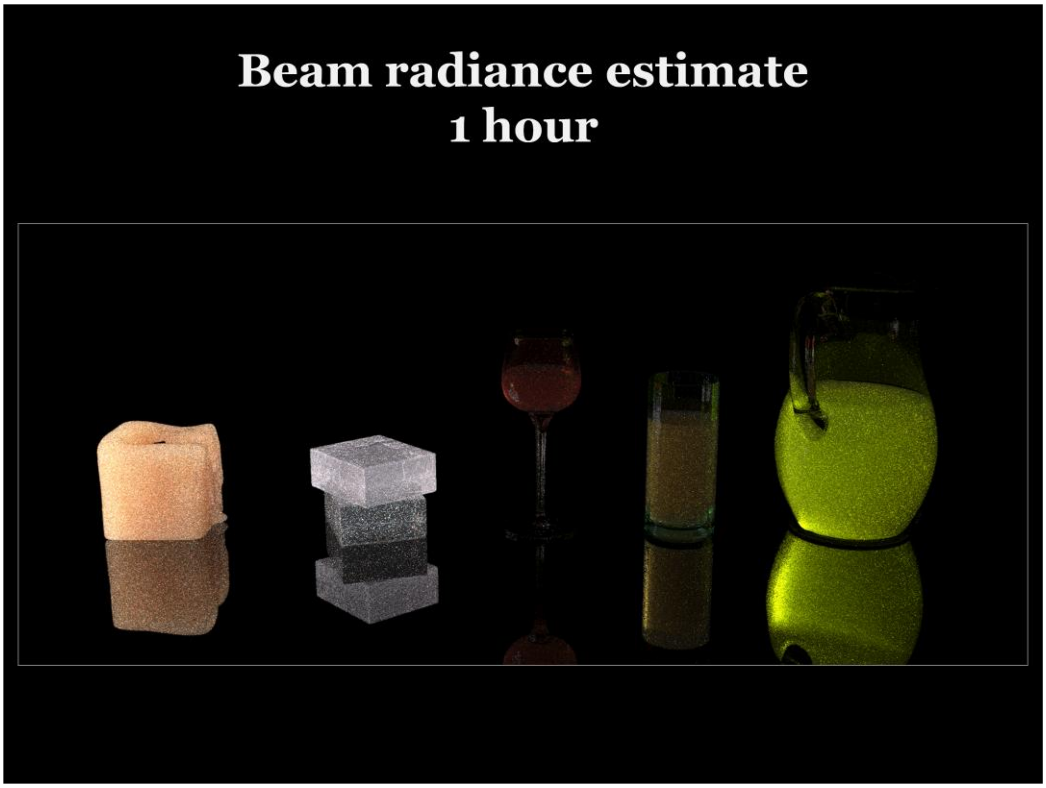
- The most robust existing approaches for volumetric light transport can be divided into two categories.
- First, we have Monte Carlo path integration, such as bidirectional path tracing.
- And second, we have techniques derived from photon density estimation, such as volumetric photon mapping, the beam radiance estimate or photon beams.
- While each of these techniques is great in certain types of media, it may fail for other types.
- We address this problem in our work.



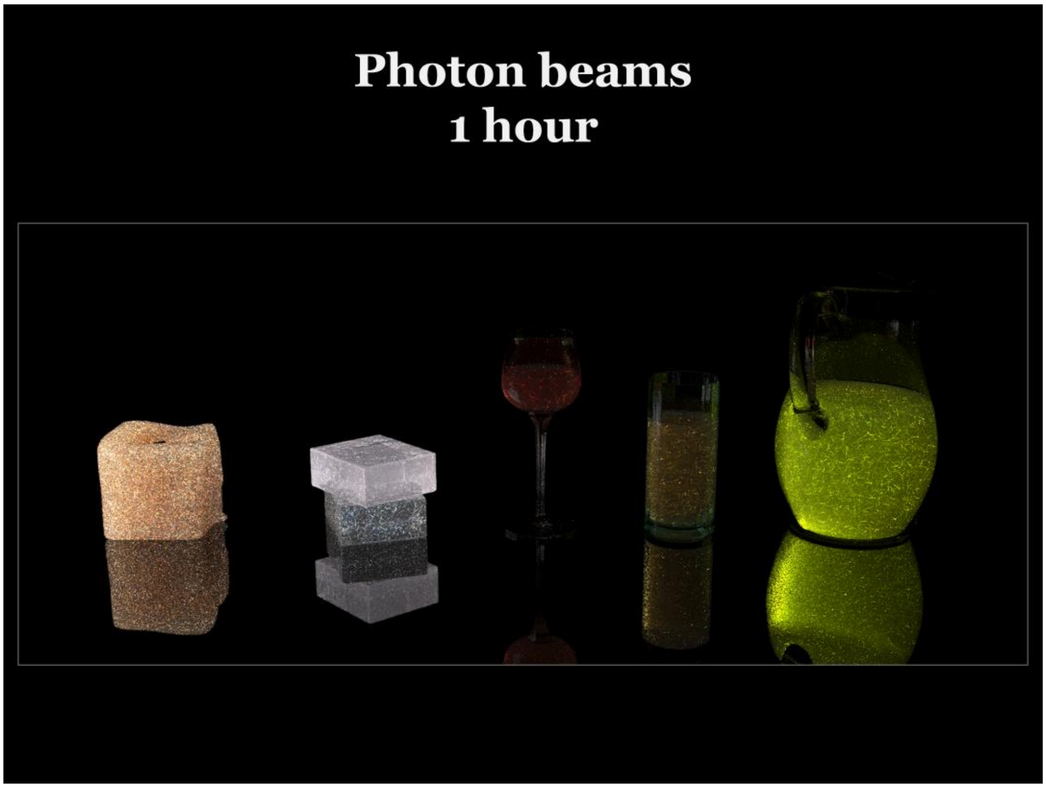
- To further motivate our work, let's look at the volumetric light transport in the previously shown scene as rendered by some of the existing algorithms.
- This is bidirectional path tracing, and we can see that the image is pretty noisy even after an hour.



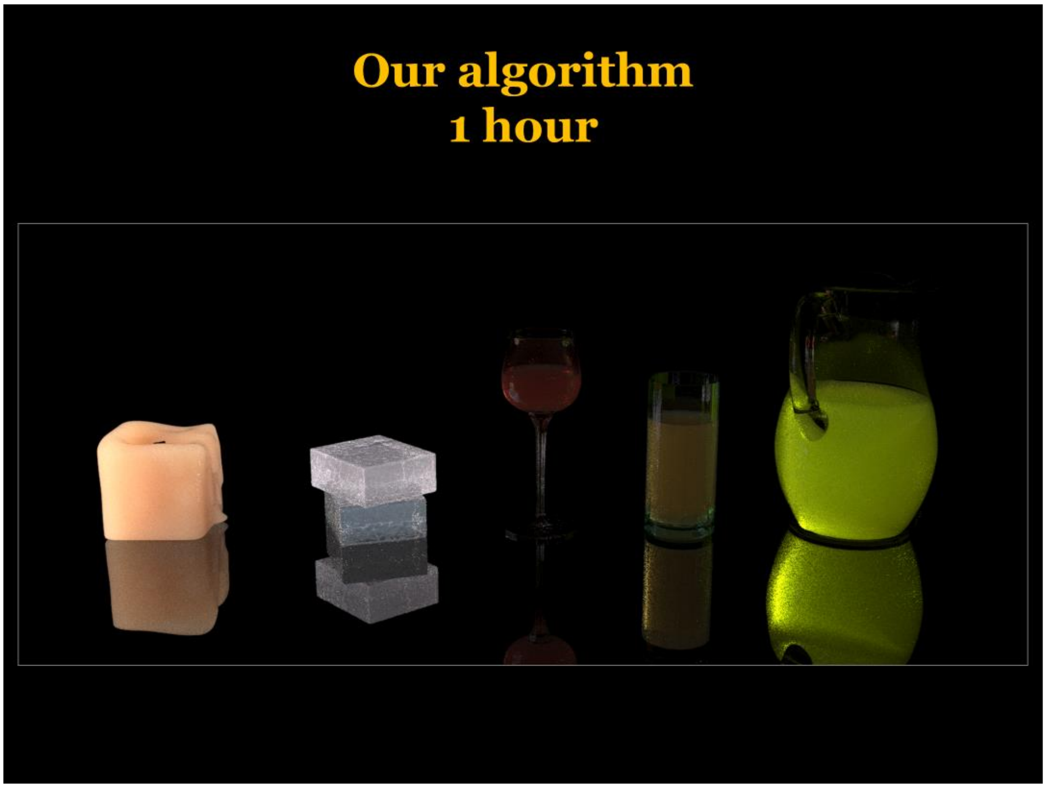
- Volumetric photon mapping



- Beam radiance estimate, much better but still not great.



- Photon beams.



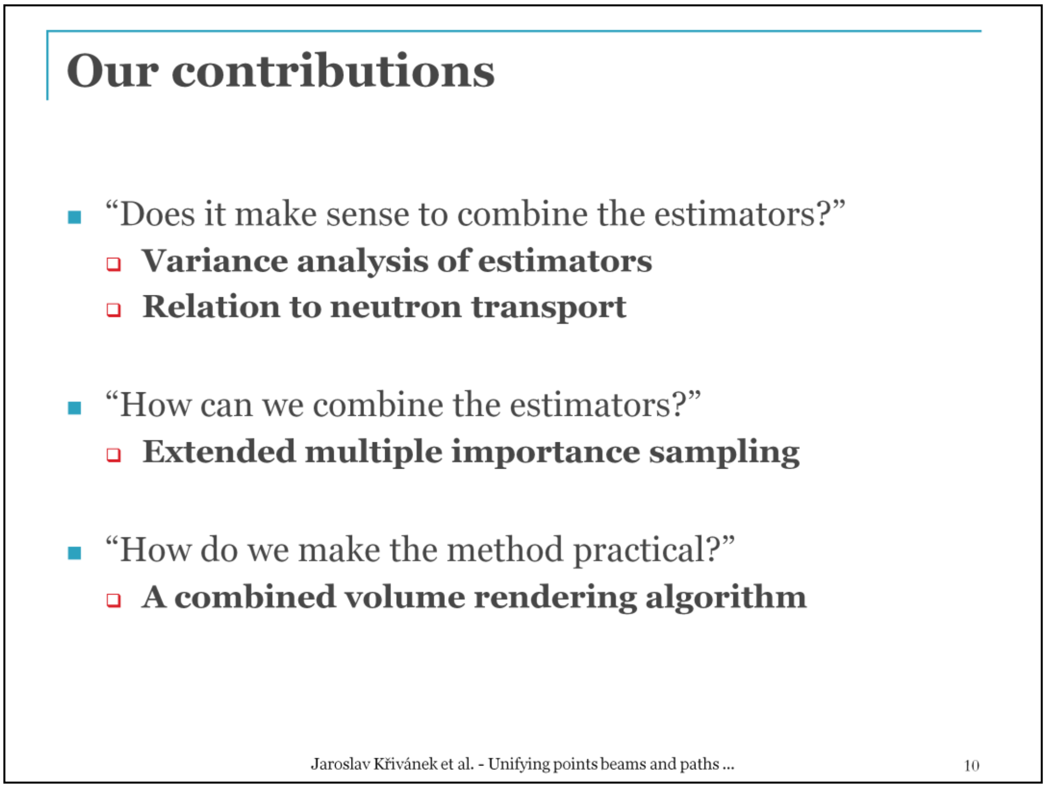
- And finally, our algorithm is able to produce a much cleaner image in the same amount of time.

Our approach: Combine estimators

- **Multiple Importance Sampling** [Veach and Guibas '95]
- **Previous work**
  - Bidirectional path tracing (**BPT**) [Veach and Guibas '95]
  - Vertex connection and merging (**VCM**) [Georgiev et al. '12]
  - Unified path sampling (**UPS**) [Hachisuka et al. '12]
- **Our algorithm**
  - **"Unified points beams and paths" (UPBP)**

Jaroslav Křivánek et al. - Unifying points beams and paths ... 9

- To achieve these results, we follow previous work that has shown that combining different estimators using Multiple Importance Sampling is an excellent way to achieve robustness.
- Notably, the Vertex Connection and Merging and Unified Path Sampling frameworks have recently combined MC path integration with photon density estimation.
- We apply the idea of combining estimators to volumetric light transport.
- We call the resulting algorithm "**unified points, beams and paths**" to reflect the multitude of different estimators in the mixture.



- We have addressed some interesting open questions related to combining estimators in volumetric light transport.

- First, there are more estimators in volumes than on surfaces.
- Do they have some complementary advantages to justify their combination?
- To answer this question, we derived their variance and found out that the variance behavior is indeed complementary, so the combination makes sense.
- And as a bonus, we've shown that there is a very tight connection between what we in graphics call the photon points and beams and the so-called collision and track-length estimators used in neutron transport.

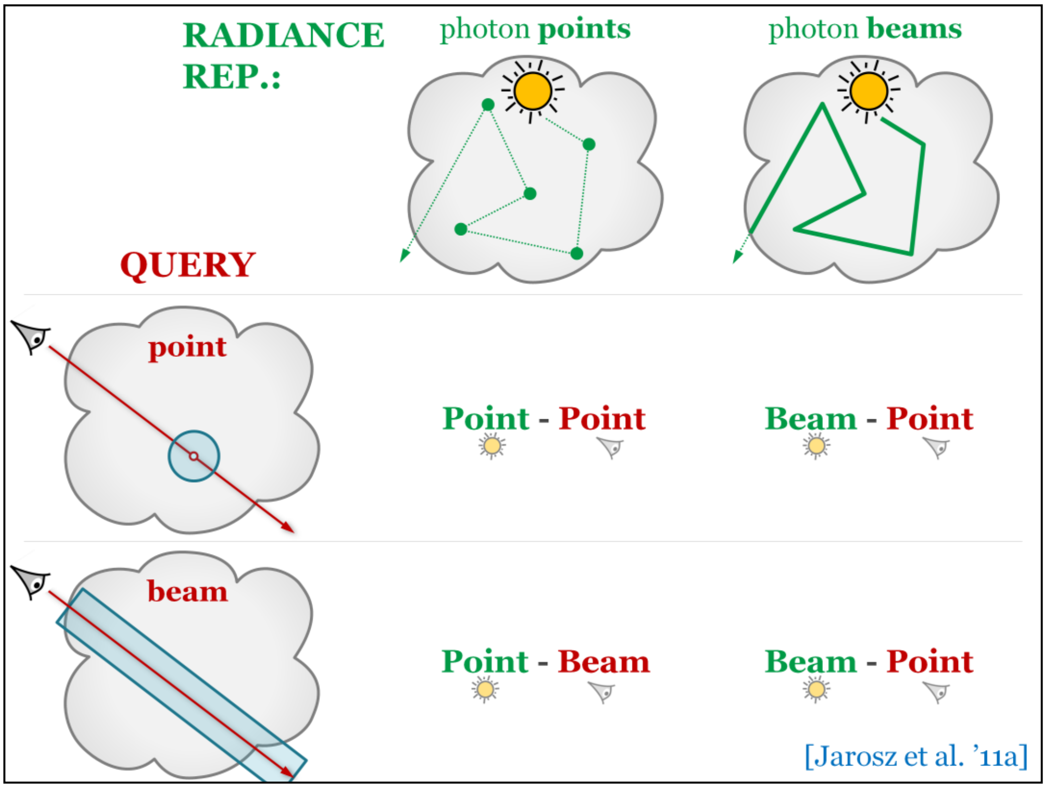
- The second question is how exactly to combine the estimators.
- To do this, we've developed a new generalization of Multiple Importance Sampling.

- Third, we developed a practical combined rendering algorithm robust to different media properties.

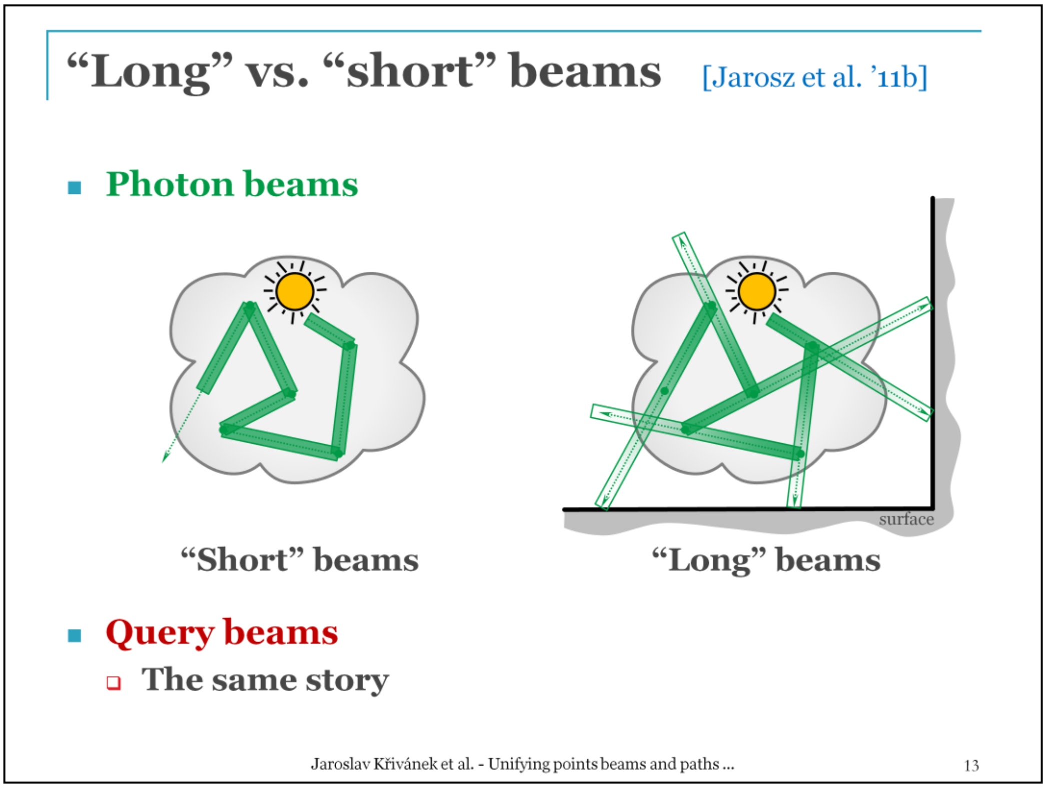
- Before giving details on these points....

VOLUMETRIC PHOTON DENSITY ESTIMATORS

… let me briefly review the different volumetric photon density estimators.

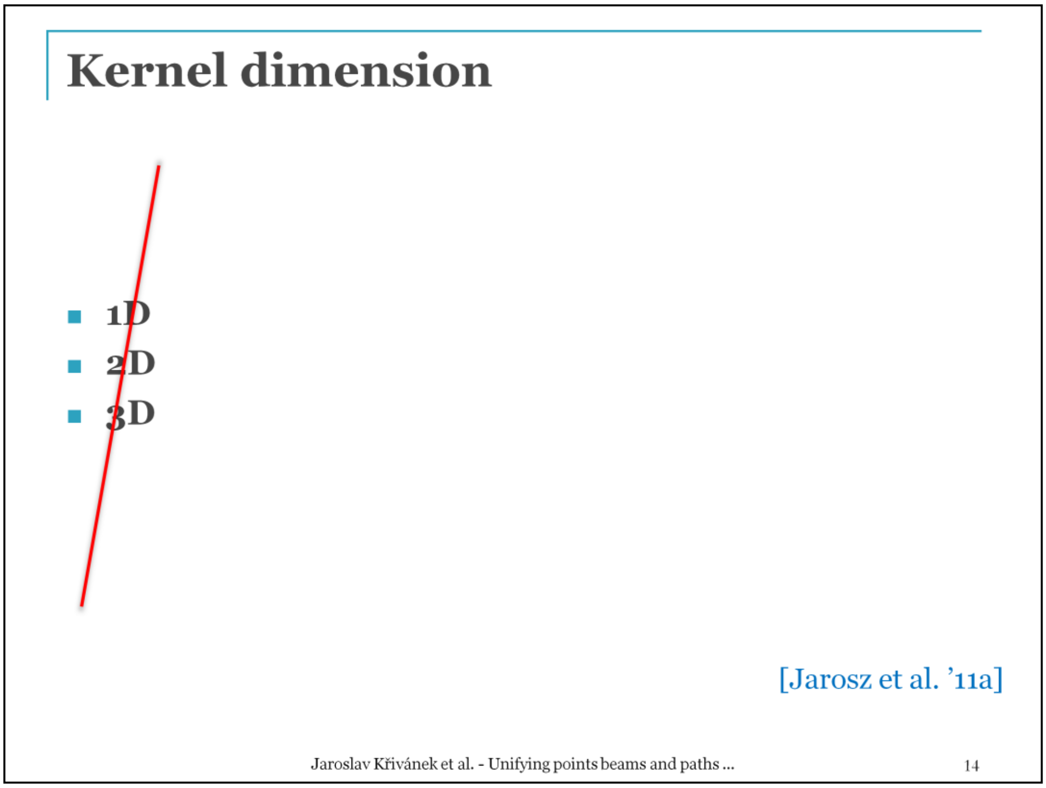


- Photon density estimation works in two passes.
- In the first pass, we trace paths from light sources and store a representation of equilibrium radiance.
- In media, we can represent the radiance either by particles or **photon points**, or by particle tracks, or **photon beams**.
- In the second pass, we query this representation to render an image.
- Here, we can use a radiance estimate at a certain **query point**, or along an entire ray, or **query beam**.
- This gives us four basic types of estimators: Point-point, Beam-point, Point-Beam, Beam-Beam.

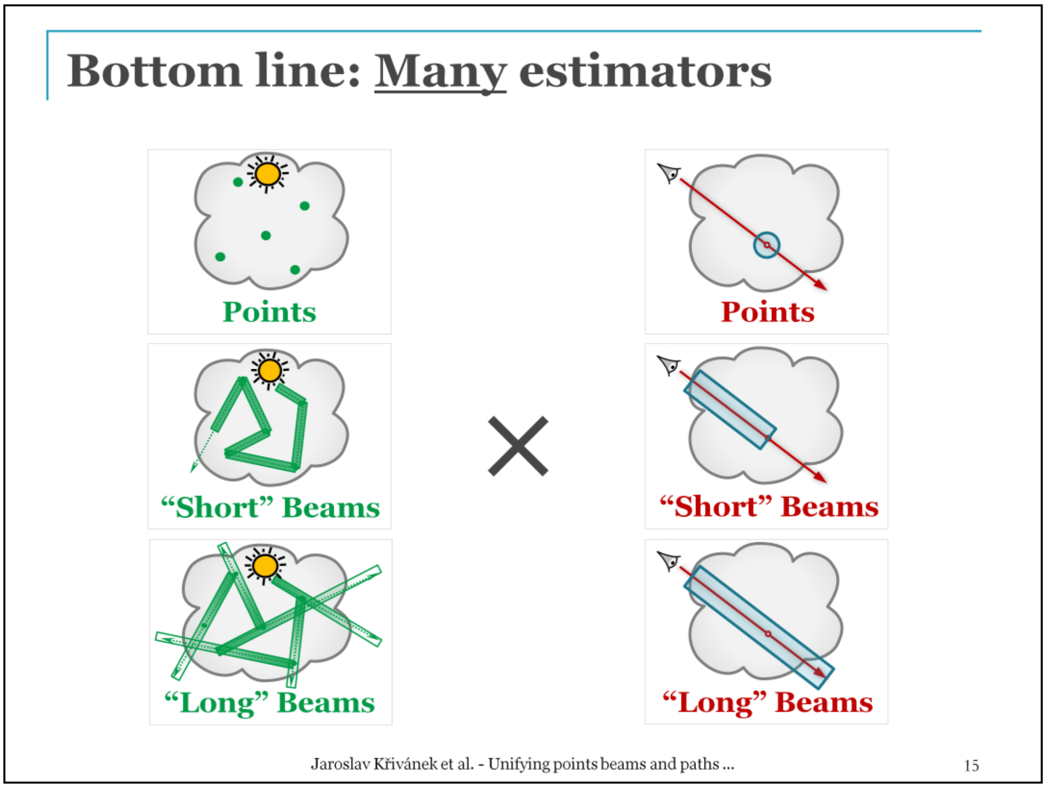


- In addition, the photon beams may either be limited to the actual trajectory of the path that generated them, which we call **“short” beams**.
- Or they may extend all the way to the next surface, which we call **“long” beams**.
- This difference has a significant impact on the estimator variance.

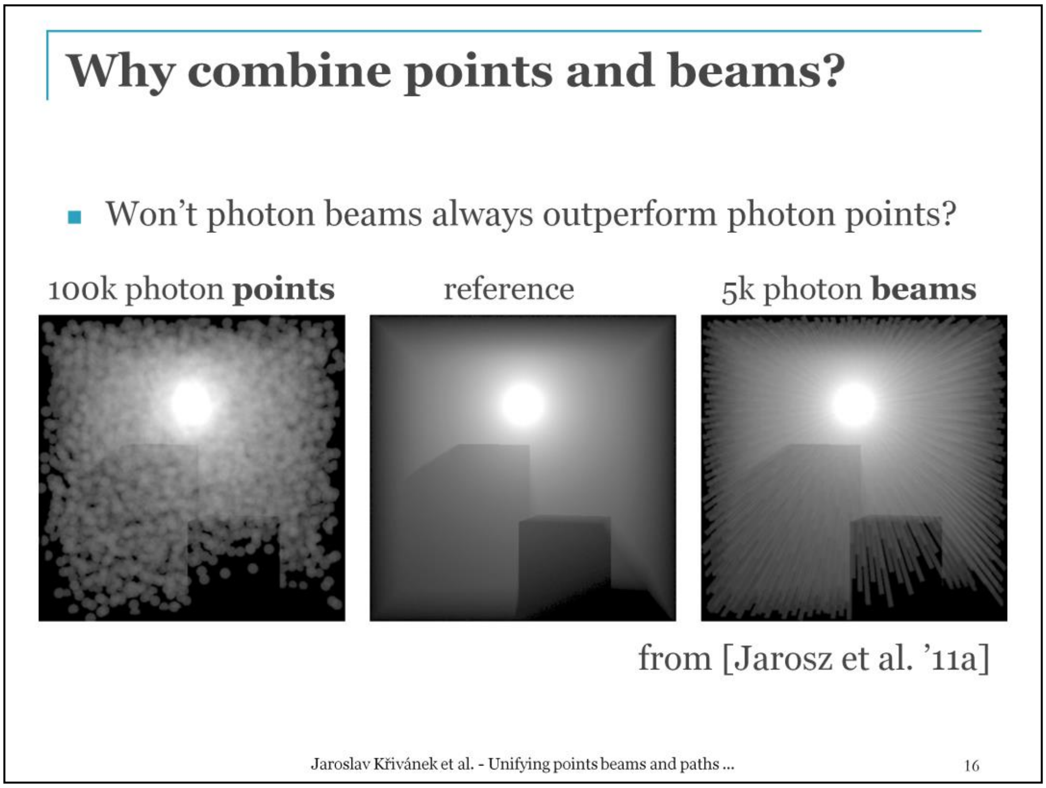
- We can apply the exact same thing also to the query beams, so we can have short and long query beams.



- The density estimation kernels may have various dimension, which increases the number of estimator even more.
- In practice, we follow the previous work and choose the estimators with the lowest kernel dimension.
- But note that all the theory derived in the paper applies to all the estimators.



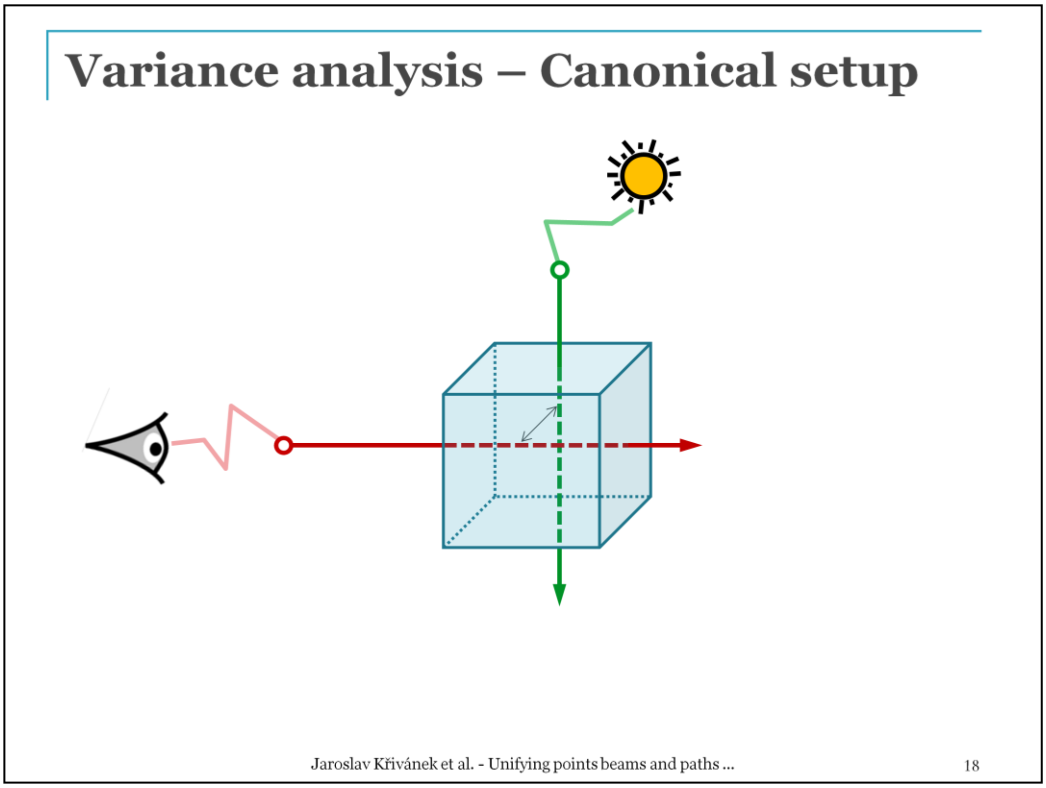
- The bottom line is, there are many volumetric estimators.
- Does it make sense to combine them?



- For example, intuitively, one could expect that because the beams fill up the space so much faster, they might be always better than points.
- But we will see that while photon beams are great in some media, they may be outperformed by points in other media.

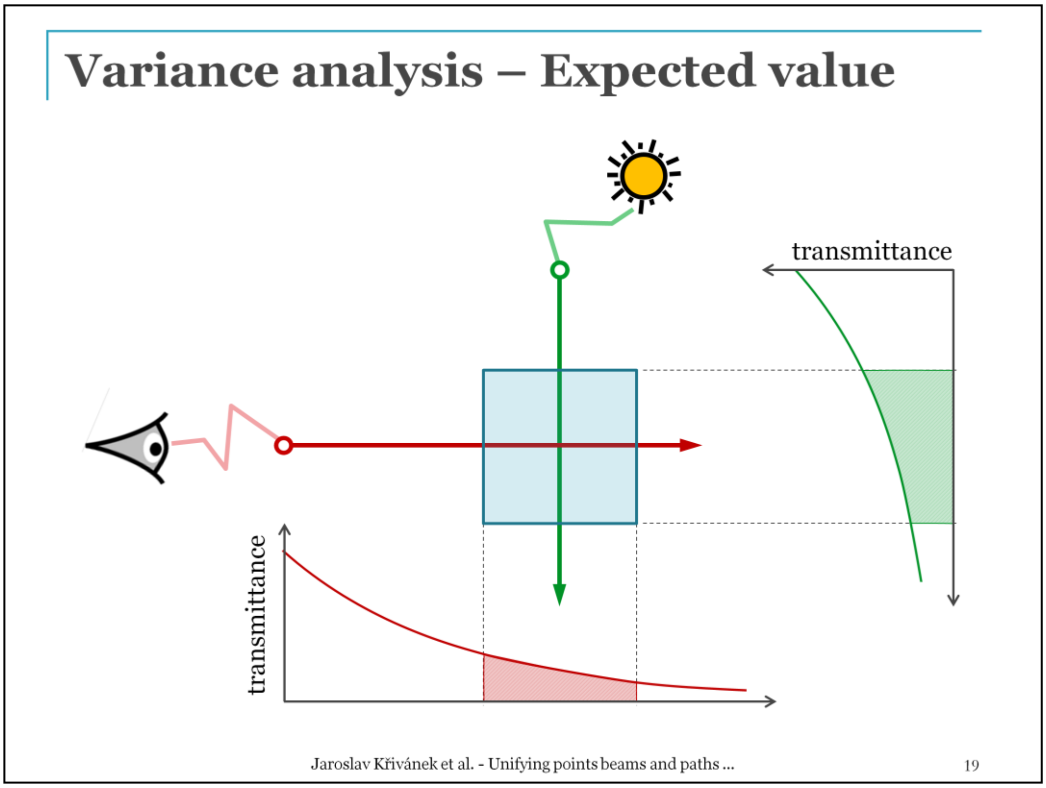
VARIANCE ANALYSIS

- To formally asses the performance of the different estimators, we derived their variance in a canonical configuration.



- The configuration that we consider consists of two fixed perpendicular rays in a homogeneous medium.
- The green one is at the end of a light sub-path and the red one at the end of an eye sub-path.
- We choose a constant cube kernel and assume that both rays pass through the kernel.

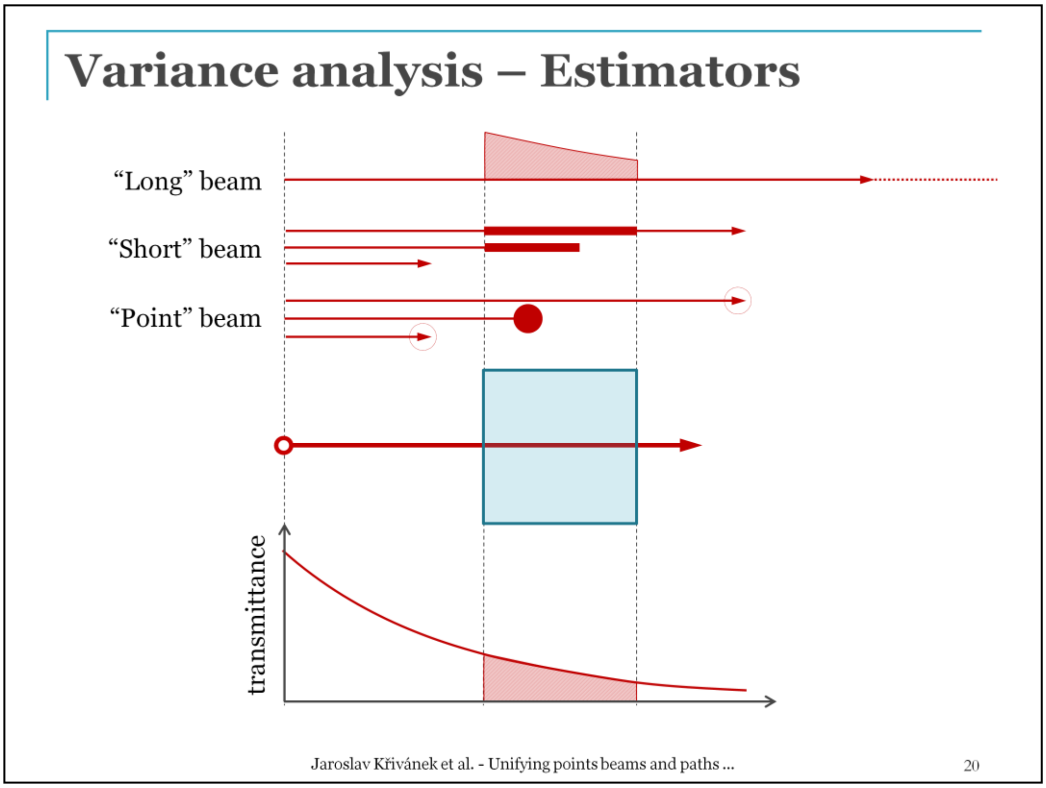
- In rendering, this configuration is sampled randomly which incurs some extra variance.
- But this variance is the same for all the estimators so we won’t need to worry about it here because our goal is to compare the variance of the different estimators.



- To simplify the diagram, I'll draw it flat.

- In this setup, the expected value of the estimators is the integral of transmittance over the kernel along the light ray …
- … times the same thing for the eye ray.

- And each of the estimators estimates this value in a different way with different variance for which we have derived analytical expressions.
- And this variance depends in an interesting way on the size of the kernel compared to the mean free path of the medium.



- Let’s have a closer look at how the integral is estimated by the three estimator types along one of the two rays.

- The **long beam** estimator shoots an infinite ray and simply always return the right answer calculated analytically, which is a zero-variance estimators of the integral.

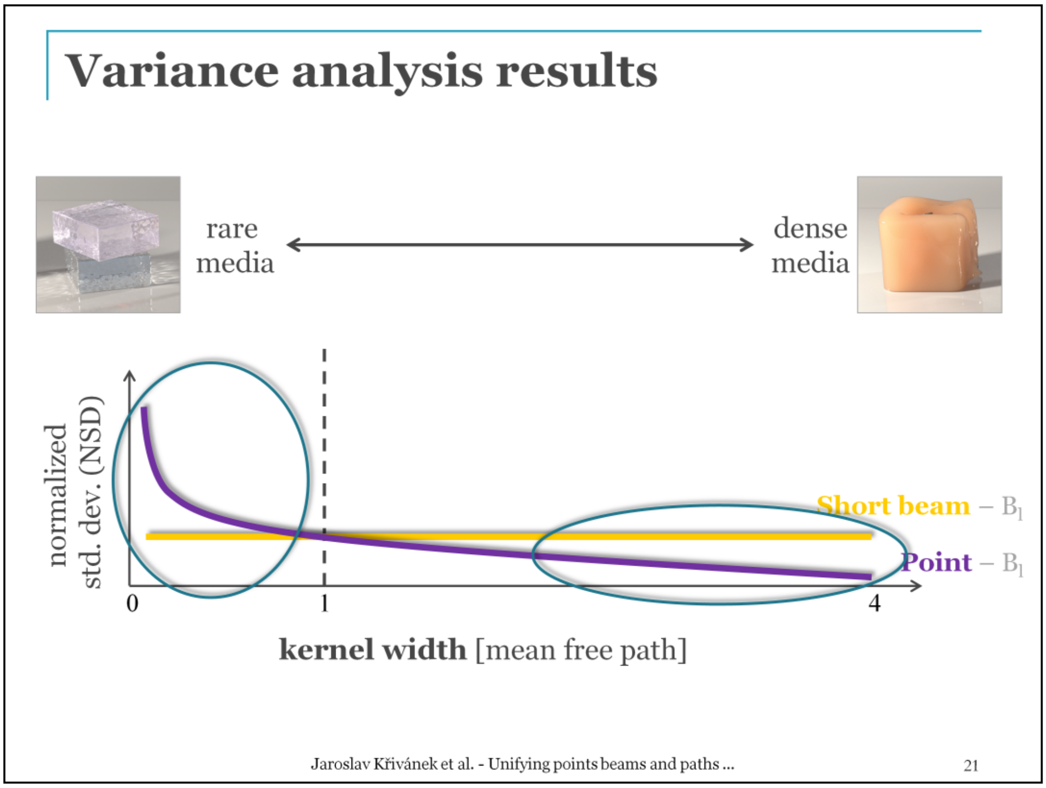
- The **short beam** estimator samples a finite ray with length proportional to transmittance and returns the length of the portion of the ray that lies inside the kernel. This could be zero if the ray does not reach the kernel. The variance in this case is non-zero and stems from two factors: whether or not the kernel is reached at all, and if it is, what portion of the ray actually lies inside the kernel.

- The **point** estimator samples a finite ray as before, but it returns a constant number if the end point fall within the kernel and zero otherwise. So the variance is only due to the chance of ‘hitting’ the kernel.

- Let’s now explain the variance behavior of the short beam and point estimators on an intuitive level:
- If the kernel is really large, the point estimator will have low variance

because it often hits the kernel and there is not other source of variance. The short beam, on the other hand, will show high variance because of the varying length of the ray segment that overlaps the kernel.

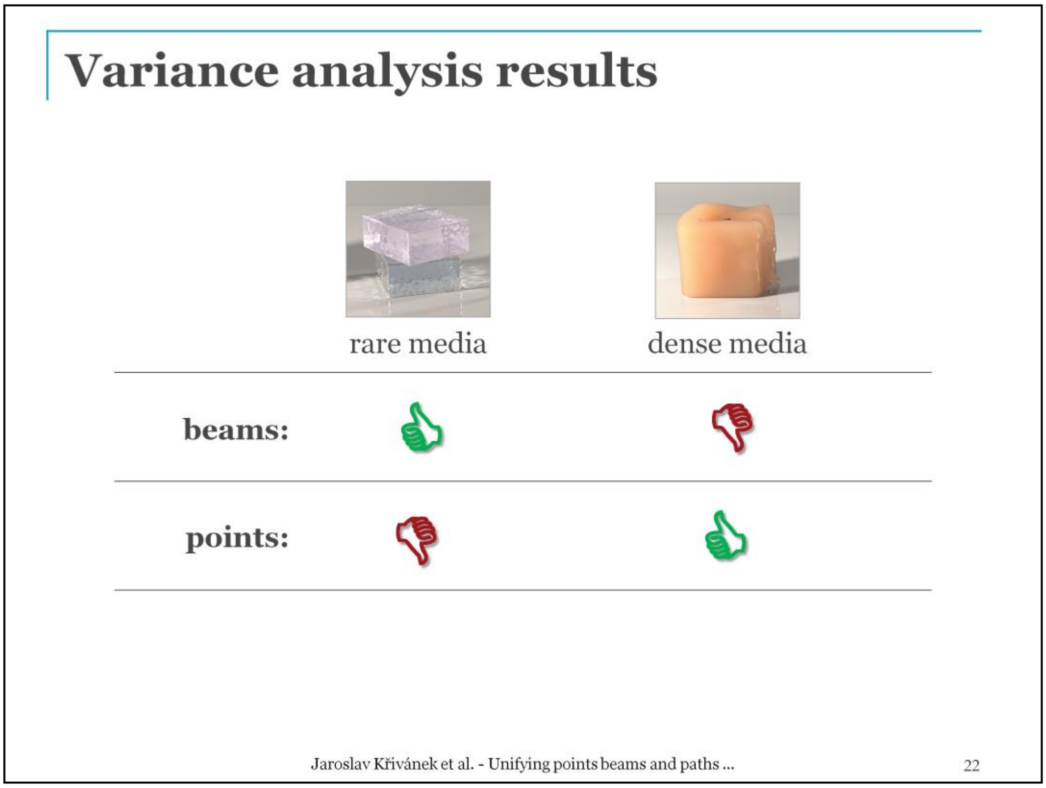
- If, on the other hand, the kernel is small, the point estimator will have a high variance because it will have hard times sampling a position in the kernel. The short beam variance will be low because the variability of the ray segment over the kernel will be small (simply because the kernel itself is small).



- Let's now plot the normalized standard deviation (NSD), which is a measure of relative variance, against the kernel width (expressed in the units of the mean free path length of the medium).
- Equivalently, if we fix the kernel size, the horizontal axis tells us how dense or rare the medium is.
- On the left there are rare media or small kernels, on the right, there are dense media or large kernels.

- We plot the NSD for two selected estimators, short-beam – long-beam and point – long-beam.
- The long beam contributes zero variance, so we're really comparing short beams to points.
- And we see that while the NSD happens to be constant for the short beams, it has an interesting behavior for the points.
- As the kernel gets smaller, or equivalently, the medium gets thinner, the NSD of the point estimator diverges.
- On the other hand, for large kernels or dense media, the NSD of points approaches zero.
- There's a cross point between the short beams and points at the kernel width of 1 mean free path.

- This behavior exactly corresponds to the intuition given on the previous slide.

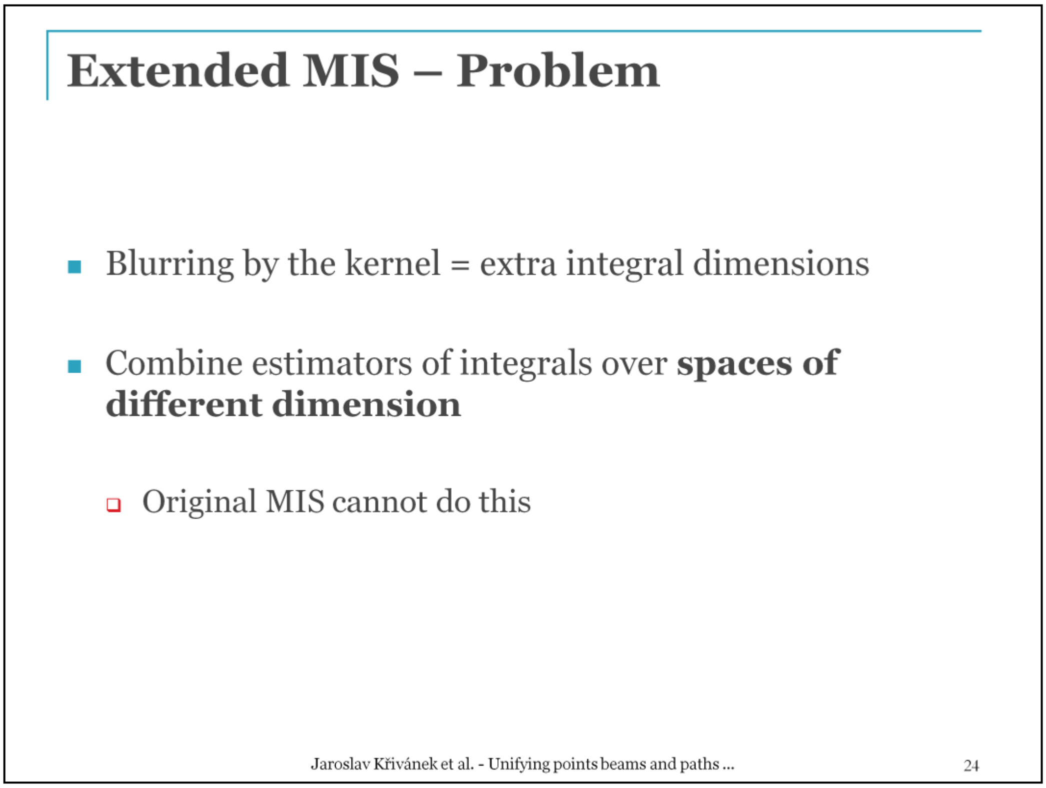


- The take-home message from this analysis is: beams are better in rare media, where the mean free path is much longer than the kernel size.
- On the other hand, in dense media, when the mfp is shorter than the kernel size, points perform better.
- We believe this is a really interesting results, and we consider the variance analysis one of the major contributions of the paper, because so far, the relative performance of point- and beam-based estimators has been unknown.

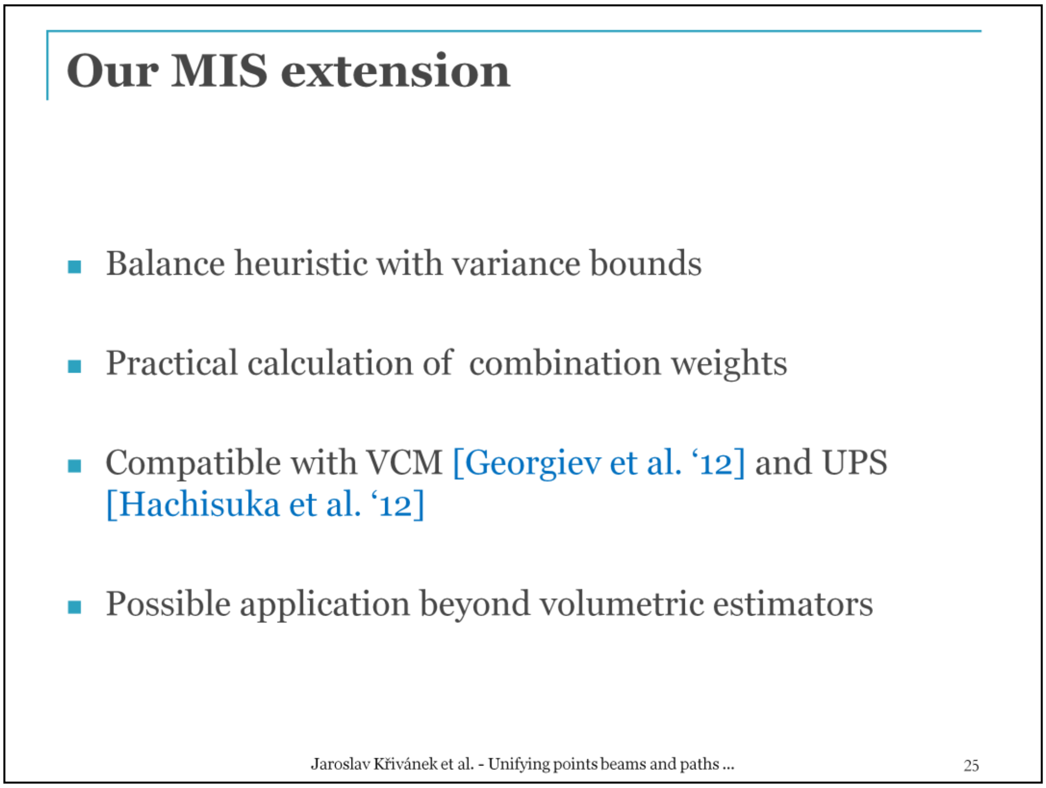
“HOW TO COMBINE?”

EXTENDED MIS

- The next question is, how to combine the estimators.
- To do this, we’ve developed a new generalization, or extension of Multiple Importance Sampling.



- Why did we need it, why not use the MIS as is?
- The problem is that blurring by the density estimation kernel corresponds to extra integral dimensions over the usual path integral.
- So we are actually combining estimators of integrals over spaces of different dimension and the original MIS just isn’t designed to do this.

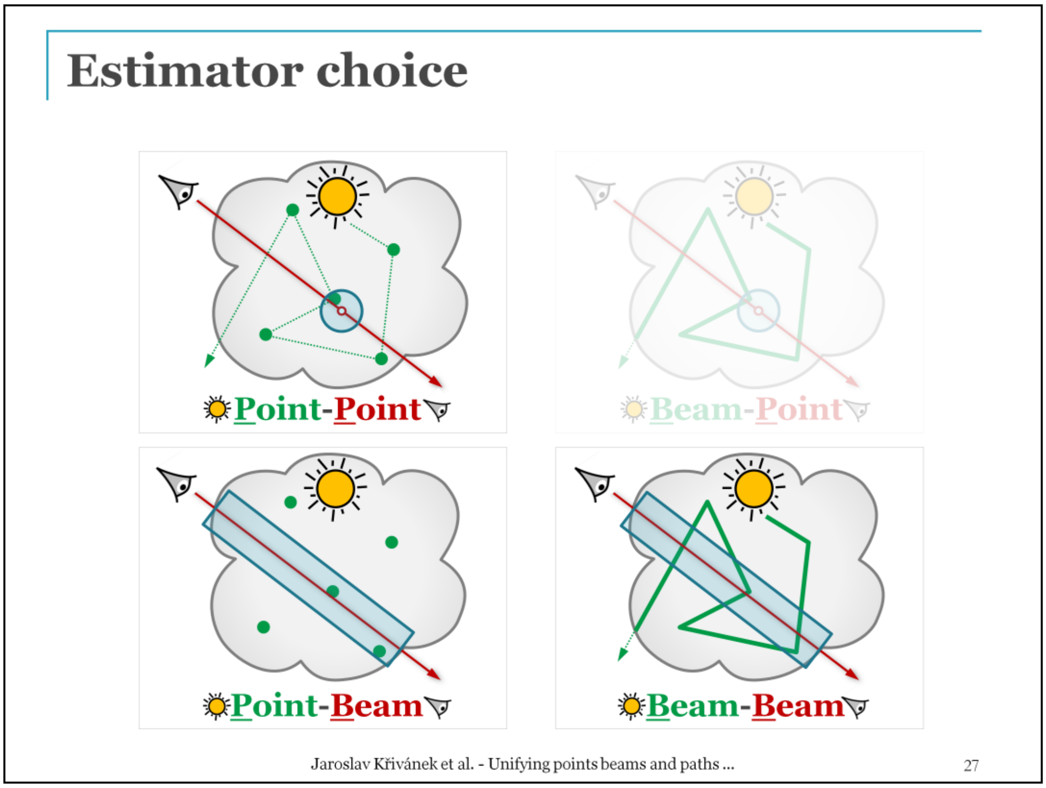


- We extend the MIS to directly recognize that some of the estimators in the mixture may have some extra dimensions.
- We've also developed the corresponding balance heuristic that allows us to calculate the combination weights.
- The result is compatible with the VCM and UPS frameworks but our formulation is more general and possibly applicable beyond the problem of combining volumetric estimators.
- And I'll have to refer to the paper for more details.

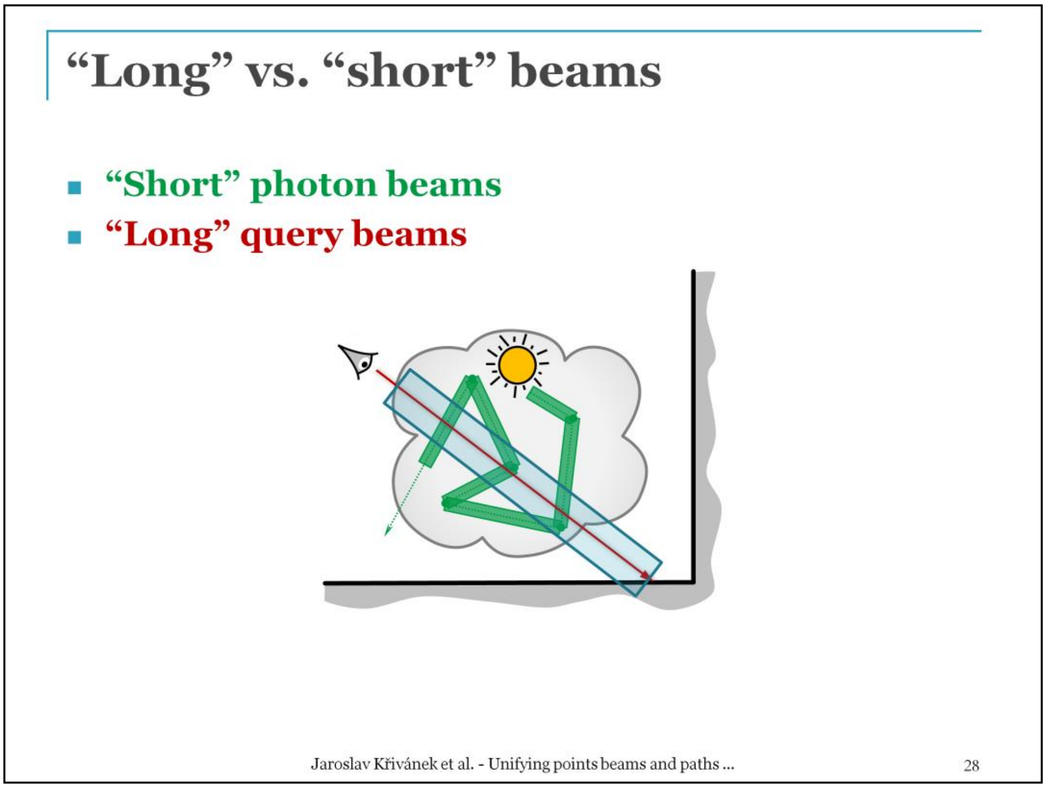
“HOW TO IMPLEMENT IT?”

THE COMBINED ALGORITHM

- Now we have a lot of theory and we want to use it to implement a practical combined algorithm. Here’s what we did.



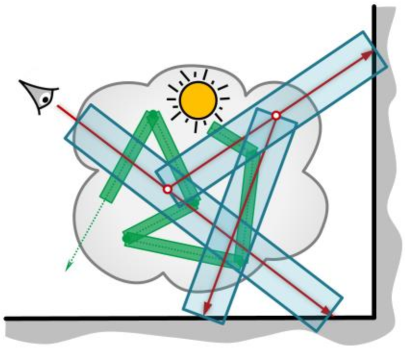
- First, we need to choose the estimators to combine.
- We use the point-point, point-beam, beam-beam estimators.
- We do not use Beam-Point estimator because it has similar properties to the Point-Beam but its implementation is much less efficient.



- Second, it makes little sense to combine long and short beam version of the same estimator so we need to choose one.
- In our test, the best performance was obtained with short photon beams but we use long query beams.

Family of estimators

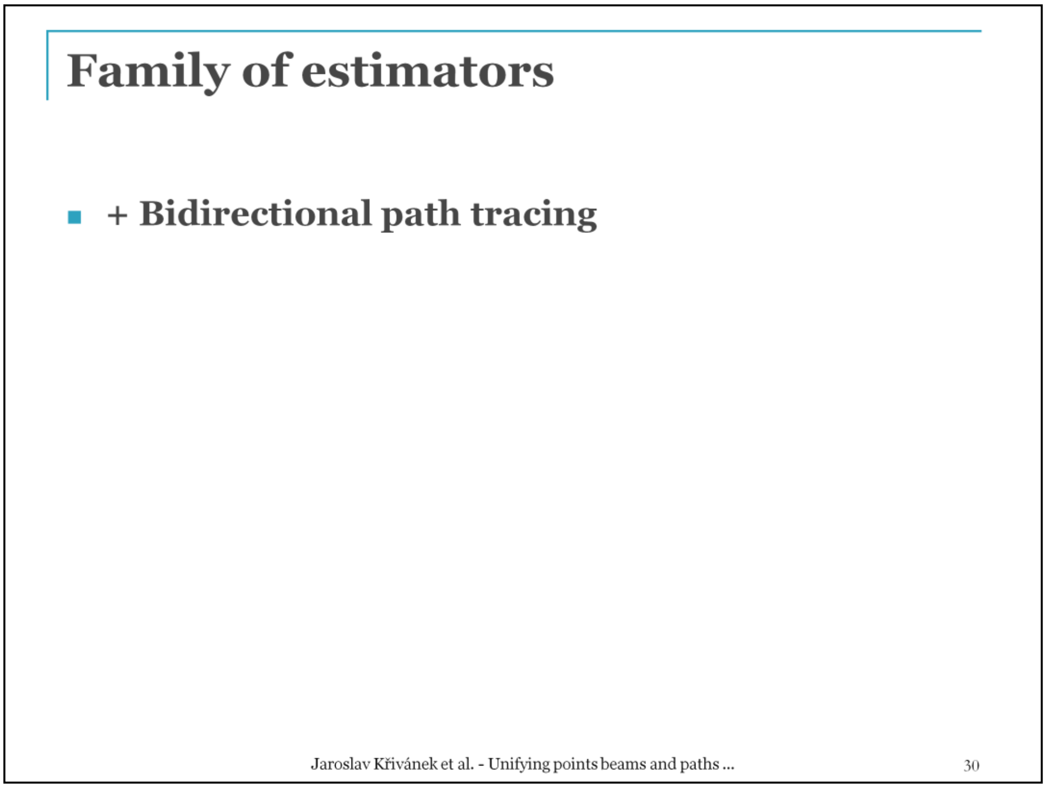
- Queries along all vertices / segments of the eye path



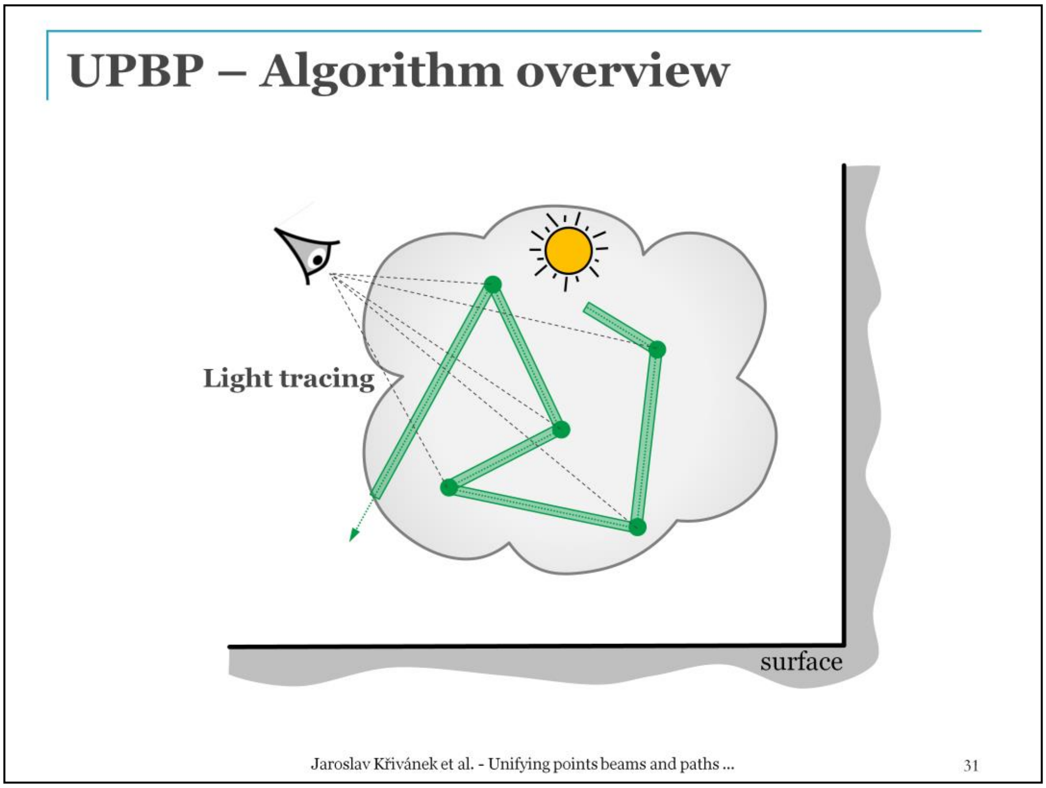
- Combined using our extended MIS

Jaroslav Křivánek et al. - Unifying points beams and paths ... 29

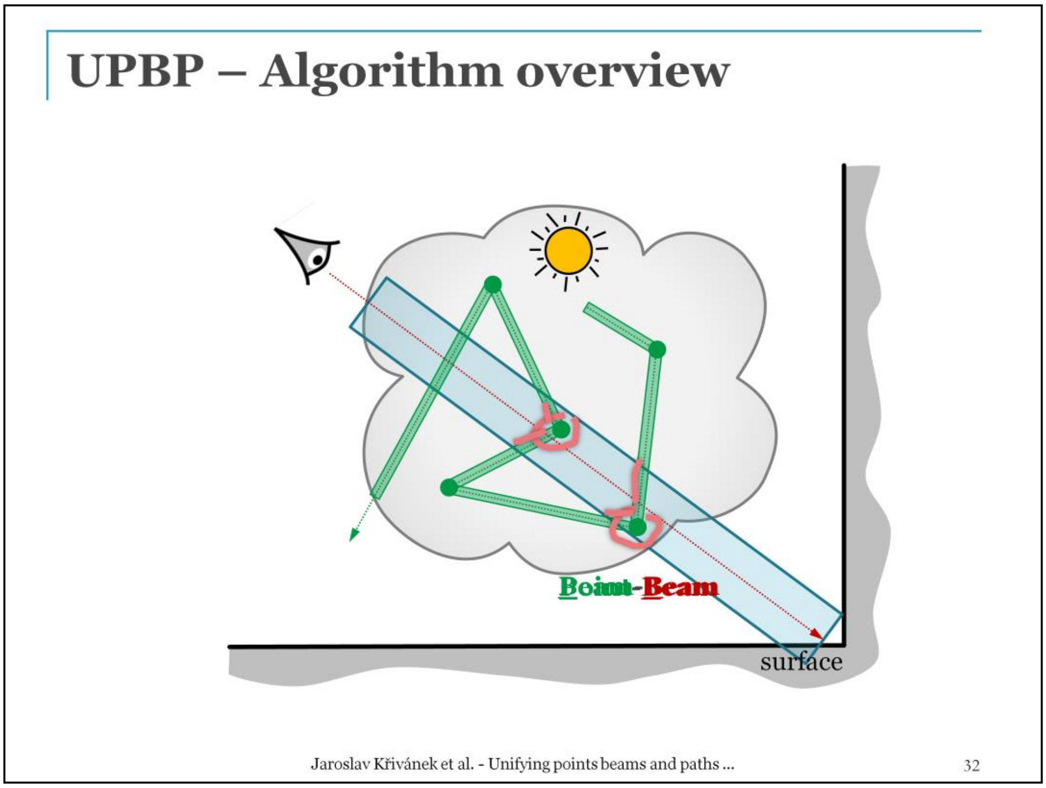
- It is important to stress that while the original estimators apply the point/beam queries just along the primary ray from the eye, we use the queries for all the segments along the eye paths, and we combine the results using our extended MIS.
- So each of the estimators results in a full family of estimators.



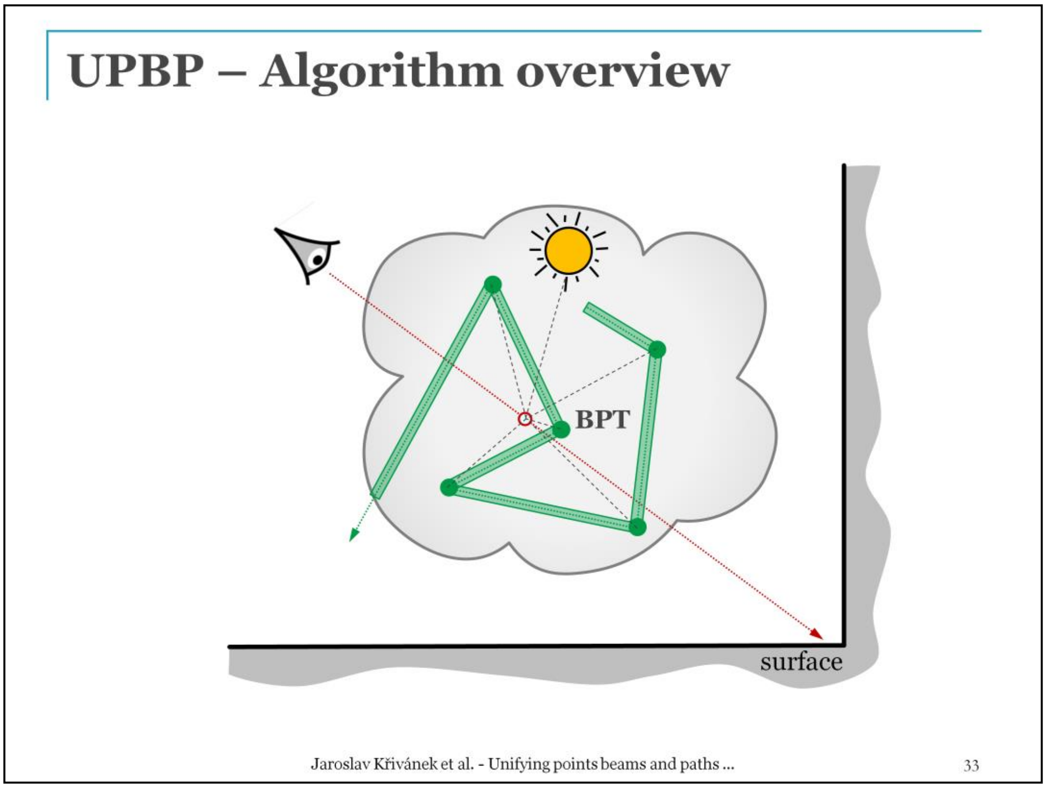
- And we add the sampling techniques from bidirectional path tracing.



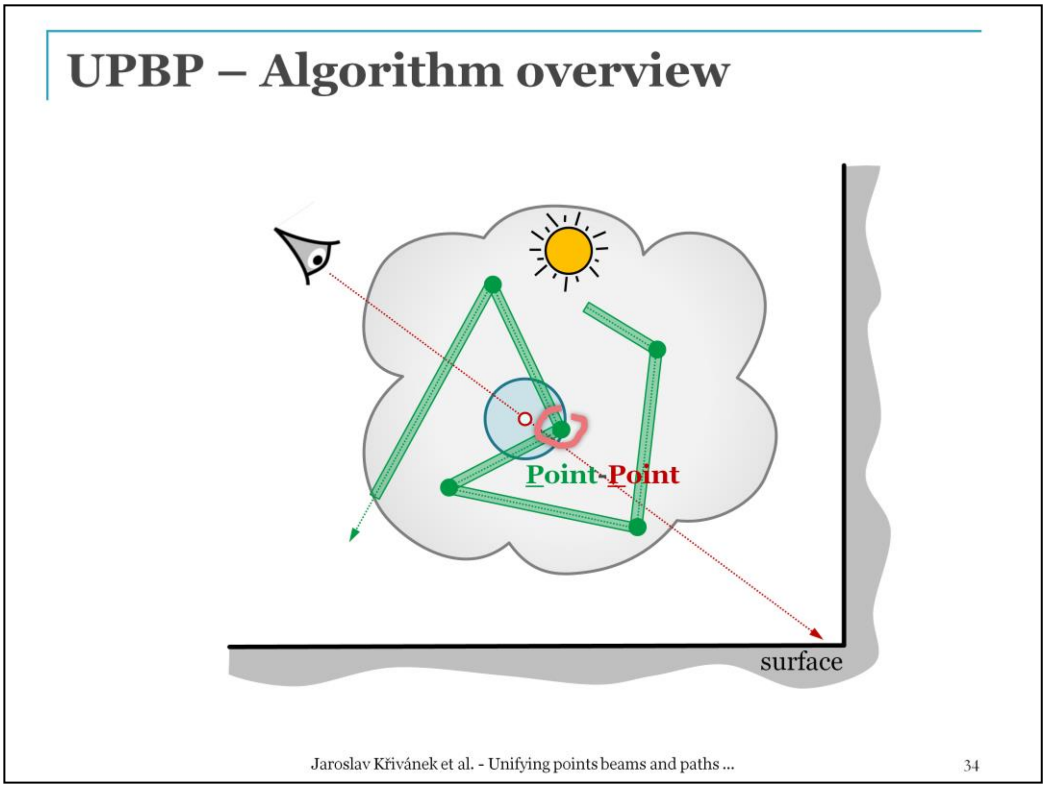
- Here’s how the algorithm works.
- In each iteration we start by tracing a number of paths from the light sources.
- We connect their vertices to the eye, which corresponds to light tracing.
- We store the vertices as photon points, and the path segments as photon beams.
- And we do this for multiple paths from the lights.



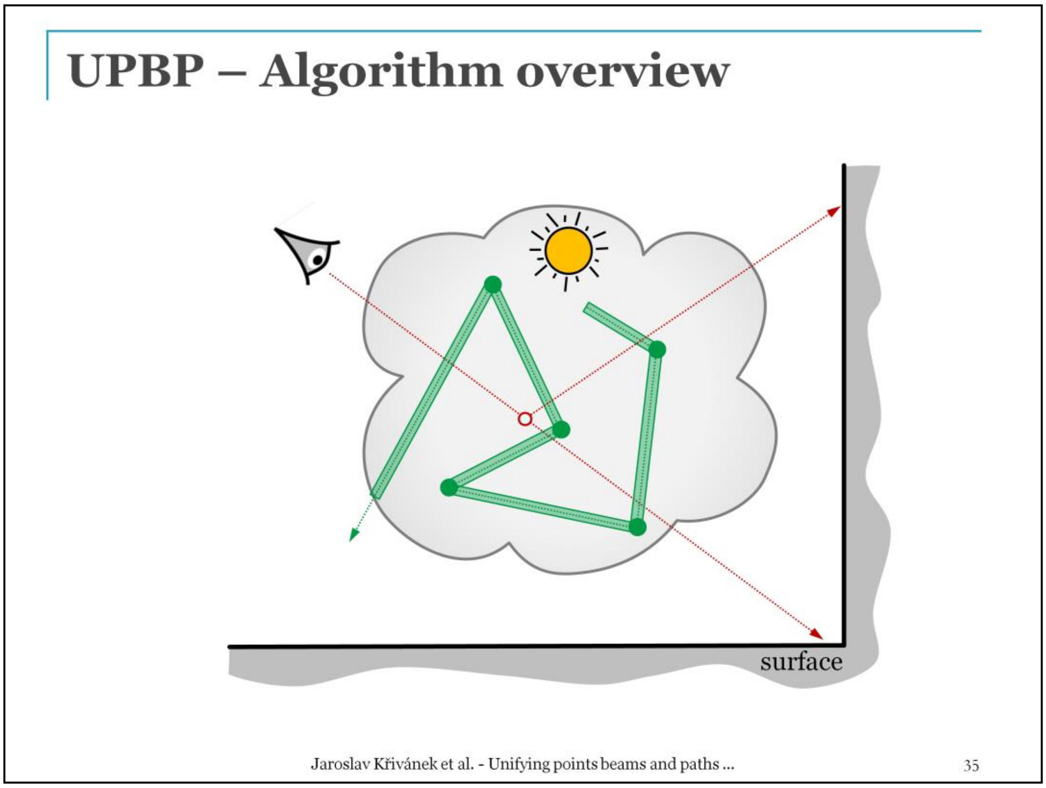
- In the second phase of the iteration, we trace eye paths through each pixel.
- For each segment of the path, we look up the photons and evaluate to the point-beam estimator.
- We then look up the beams, which is the beam-beam estimator.



- After that, we sample the scattering distance along the eye ray and connect to the light path vertices. This corresponds to bidirectional path tracing.



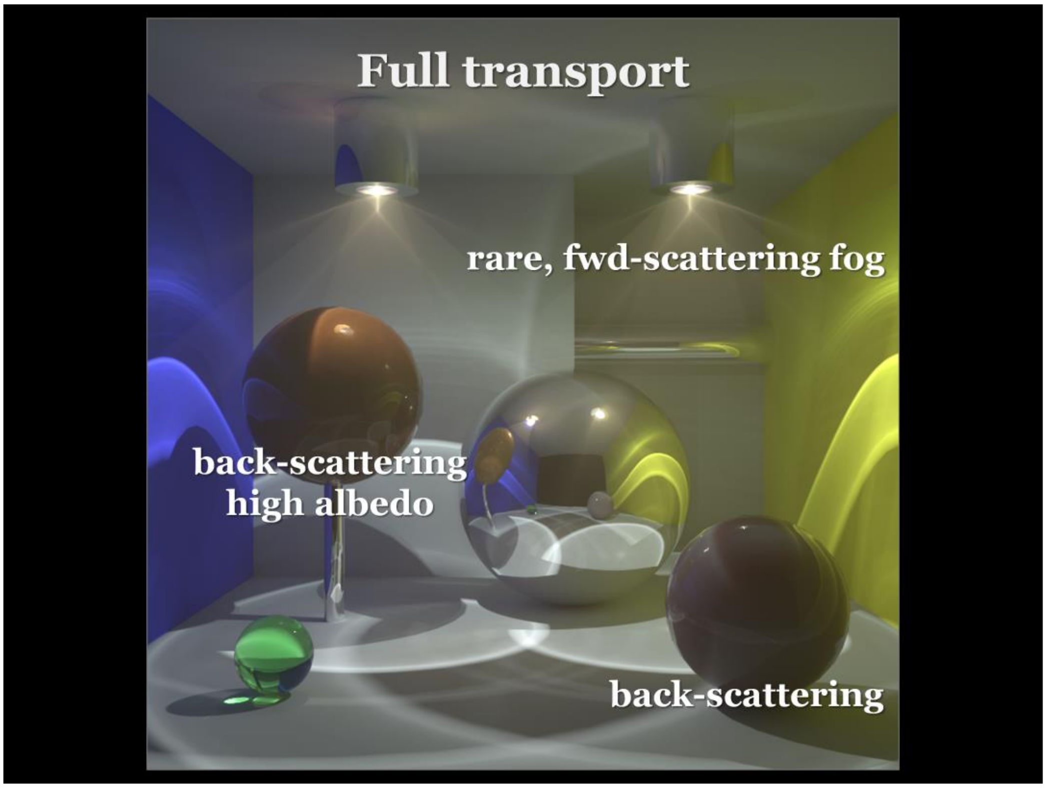
- Then we query the photons around the scattering location and evaluate the point-point estimator.



- And then we extend the eye path and repeat.

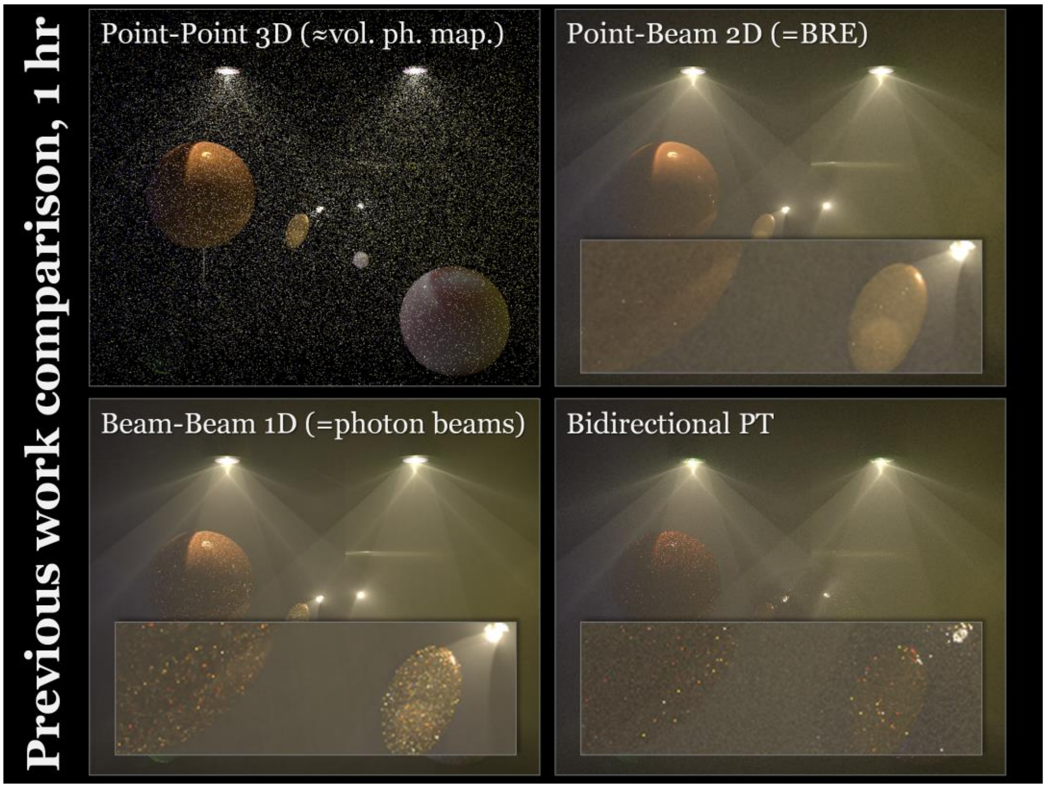
RESULTS

- Let’s see what the algorithm does.

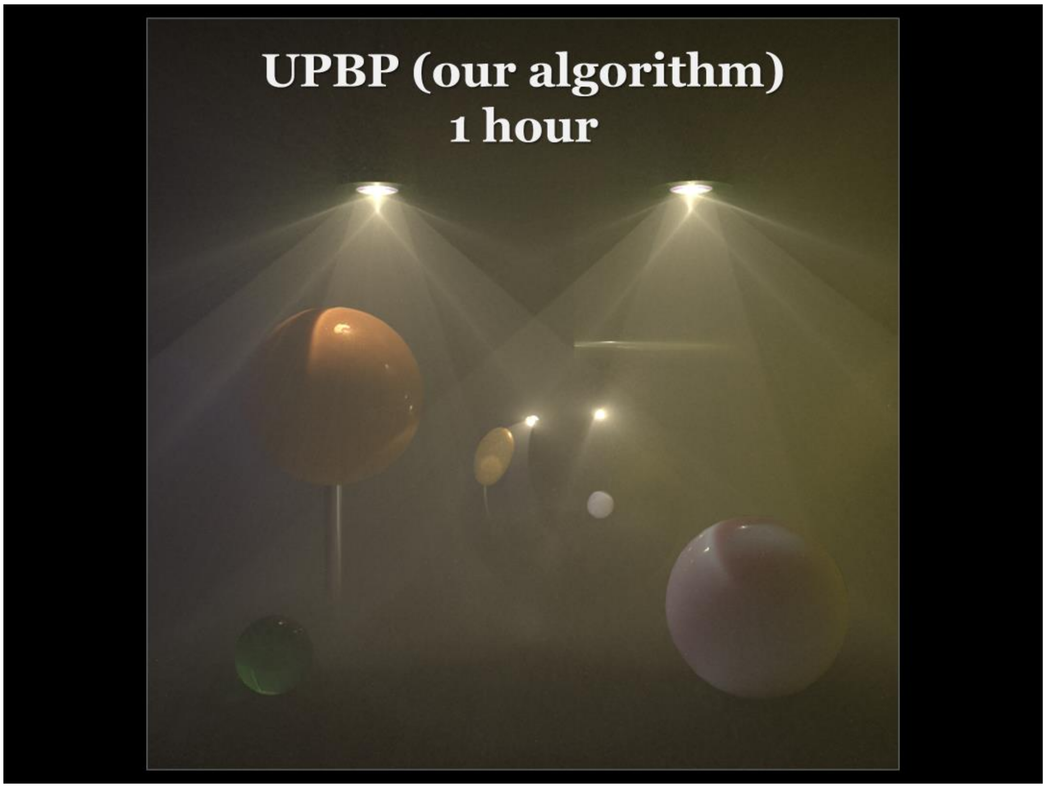


- We have set up this test scene which is filled with rare, forward scattering fog.
- There are two spheres filled with a dense back-scattering medium.
- And the scene is illuminated almost entirely by caustic lighting.

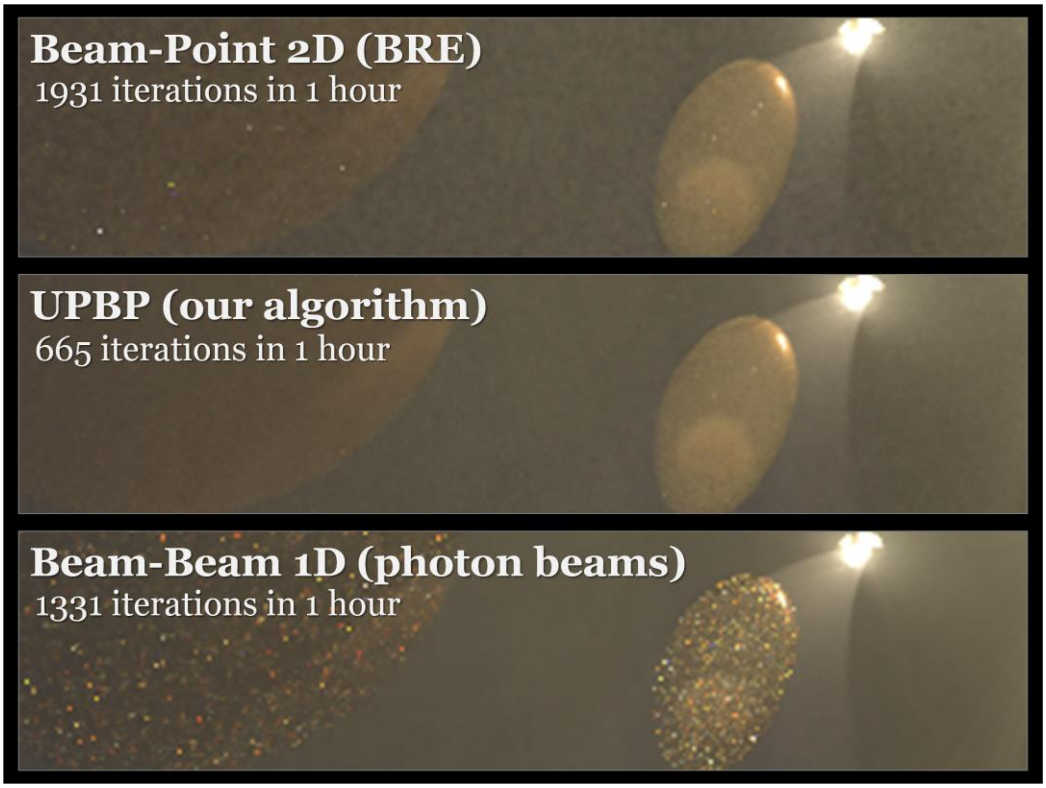
- For the algorithm comparison, we again consider only transport in media.



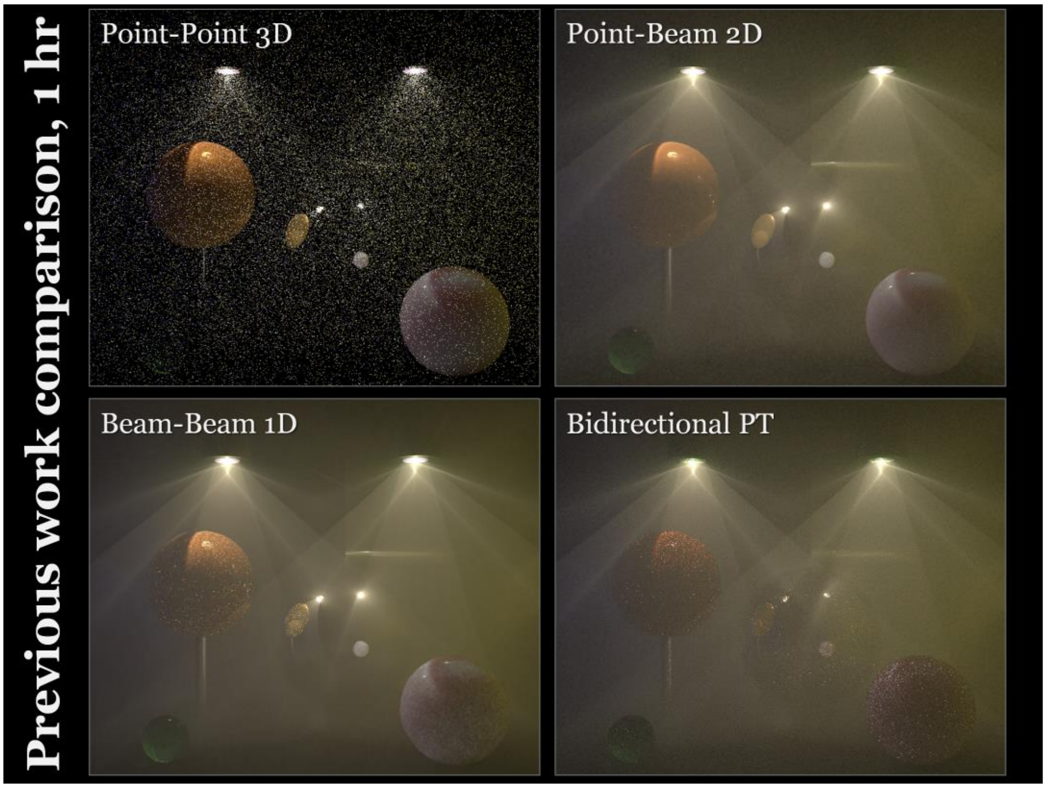
- And here's how the previous work does.
- The point-point estimator, which is equivalent to volumetric photon mapping without ray marching, does a pretty good job at rendering the dense spheres but cannot handle the sparser fog.
- The point-beam estimator, or the beam radiance estimate, does a much better job at rendering the fog, though it still fairly noisy as you can see in the inset.
- The beam-beam estimator, or photon beams, provides excellent results for the fog, bug the spheres suffer from some nasty noisy artifacts.
- Bidir handles the fog quite well, though not as well as the photon beams. It produces bad artifacts in the dense spheres because much of the illumination there is essentially a reflected caustic.



- Our algorithm is able to take the best from the individual techniques to produce a much cleaner image.



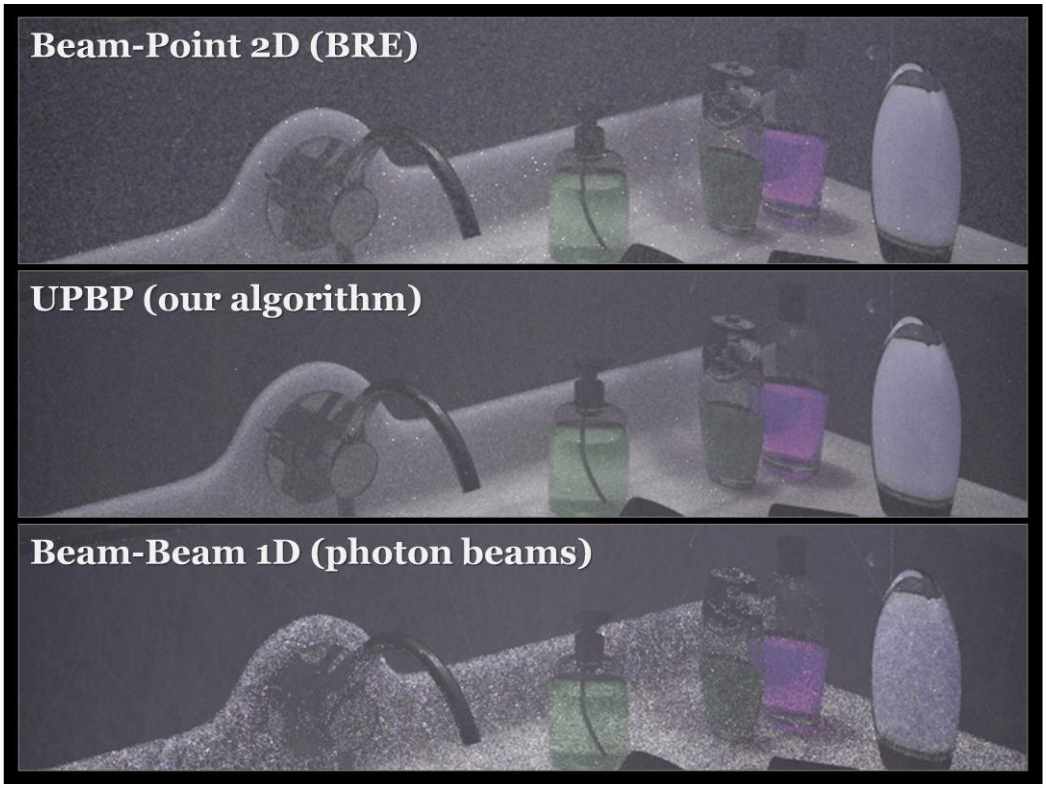
- If we compare our result to the BRE (at the top) and photon beams (at the bottom), we see that the fog is rendered much better than with the BRE but not as well as with the photon beams.
- This is the price we have to pay for combining all the estimators: if one medium is best handled by just one estimators, running the other ones only incurs overhead.



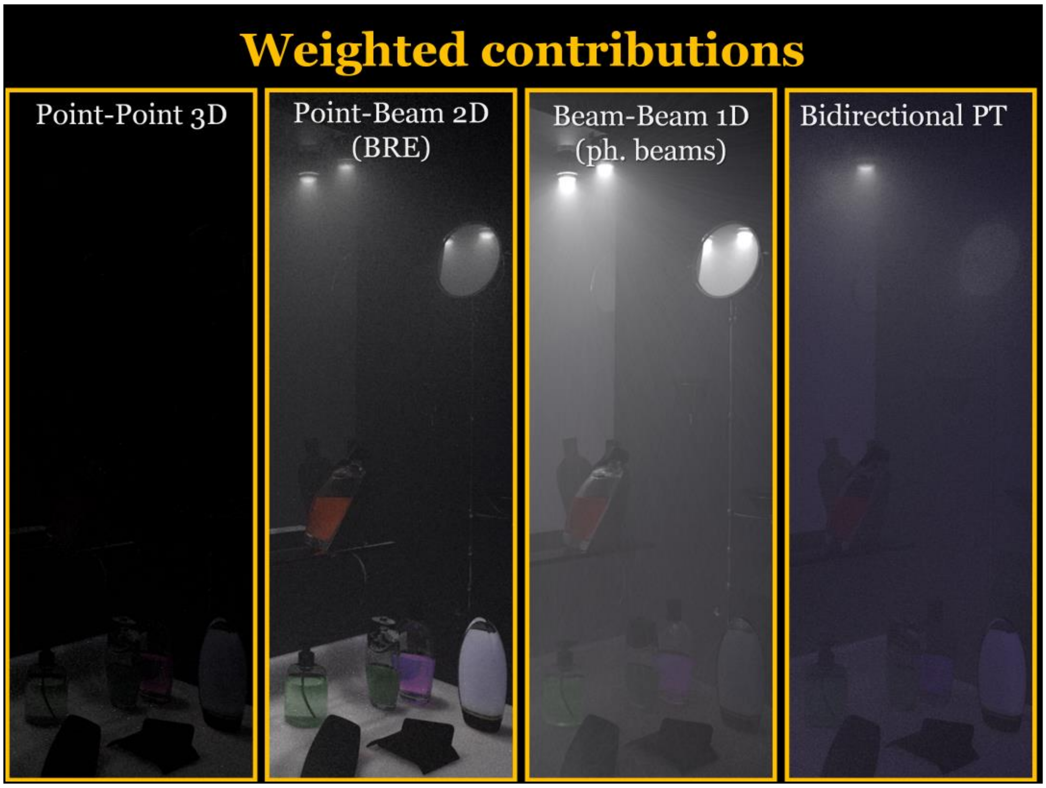
- Let’s go back to the results of the previous work and ….

- … and let's start replacing it by the weighted contribution of the individual estimators to our combined result.
- PP, PB, BB, BPT.
- We see that most of the image is made up from the contributions from the point-beam and beam-beam estimators, where the point-beam takes care of rendering the dense, back-scattering spheres, and the beam-beam estimator renders the rarer fog.
- Bidirectional path tracing contributes mostly the surface-to-medium transport, which is visible as the blue tint of the right sphere.

- The setup of this more realistic scene is fairly similar to the previous one and we have been able to confirm the observations even here.

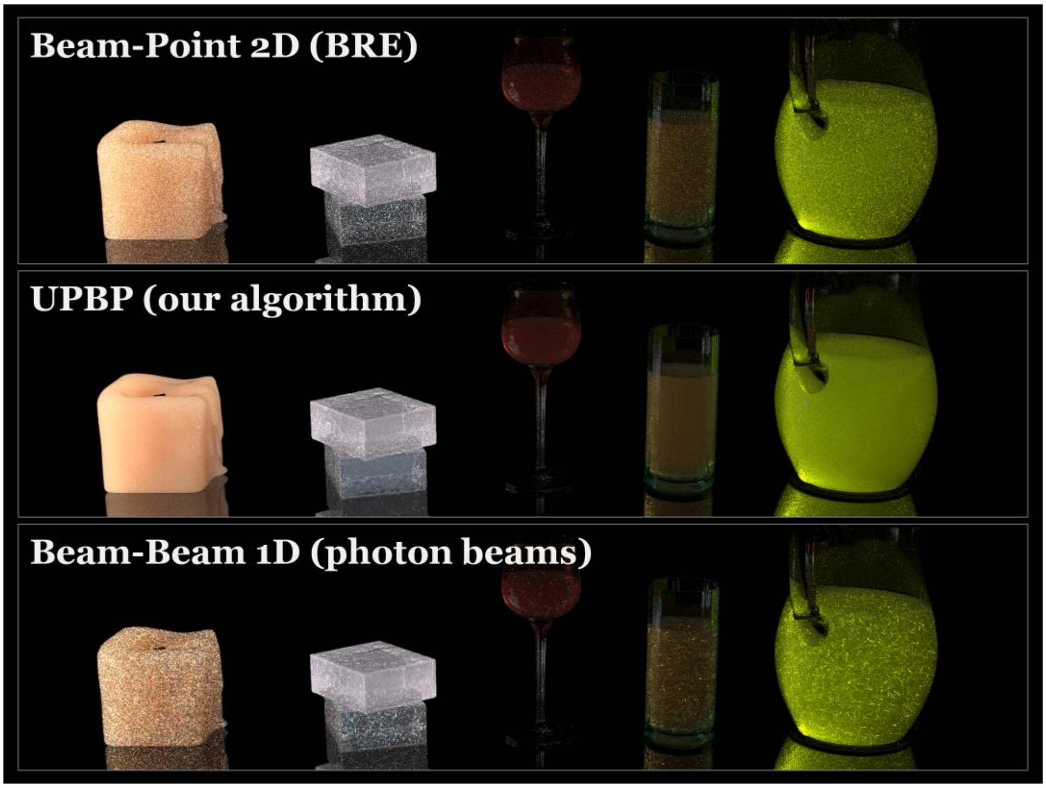


- Indeed, we see the same behavior as before: the BRE (top) handles the dense media much better than photon beams (bottom) and vice-versa for the sparser fog.
- Our algorithm takes the best of both.



- Similarly, when we look at the weighted contributions, dense media like the wash-basing are mostly covered by the BRE, thinner media like the fog by the photon beams.
- The fact that BPT is in charge of the surface-to-media transport is quite apparent here: it provides the blue tint to the media due to reflections from the blue tiles on the walls.

- I’ve already shown the results for this scene…



- … but I want to use it to point out that in this case, even though none of the previous algorithm handles this scene well, the combination is almost clear.
- This provides some evidence that our MIS-based combination is more robust than a heuristic combination that would be based on selecting a particular estimator for each medium.

Limitations & future work

- **Asymptotic variance and bias analysis**
  - Radius reduction / consistent results
- Take **bias** and **efficiency** into account
- **Number of samples** from techniques
- Implementation improvements
  (**beams data structure**)

- Of course our work does not come without its limitations.
- First, in all our results, we use a fixed kernel radius. To enable radius reduction and therefore consistent rendering, we would need an asymptotic variance and bias analysis which we haven’t done so far.
- Furthermore, the estimator combination relies *only* on the variance considerations, but taking bias and efficiency into account could significantly improve the results.
- Having a solid theory that would tell us how many samples to take from each estimator would be extremely useful, especially in the cases where some estimators could be completely disabled.
- And last but not least, we need a better data structure for looking up the photon beams.

Conclusions

- **Photon beams are not always better than photon points!**
  - Rare media: beams
  - Dense media: points
- **Extended MIS**
  - Applications beyond combining volumetric estimators
- **Practical combined algorithm**
  - Wide range of media properties

- The take-home message from this talk could be: Beams are not always better than points.
- Beams are great for rare media.
- But dense media are better handled by points.
- We have provided evidence for this through our theoretical variance analysis and the rendered images.
- Furthermore, our extension of Multiple Importance Sampling has potential application beyond just the combination of volumetric estimators.
- Finally, we have shown a practical volumetric light transport simulation algorithm robust to a wide range of media properties.

- The source code that was used to generate all our results is available, including scripts to reproduce all the results.
- It’s called SmallUPBP for historical reasons – it’s built on SmallVCM, but it’s not really that small anymore.

- This work has also received an extensive coverage on fxguide, so please make sure check it out!

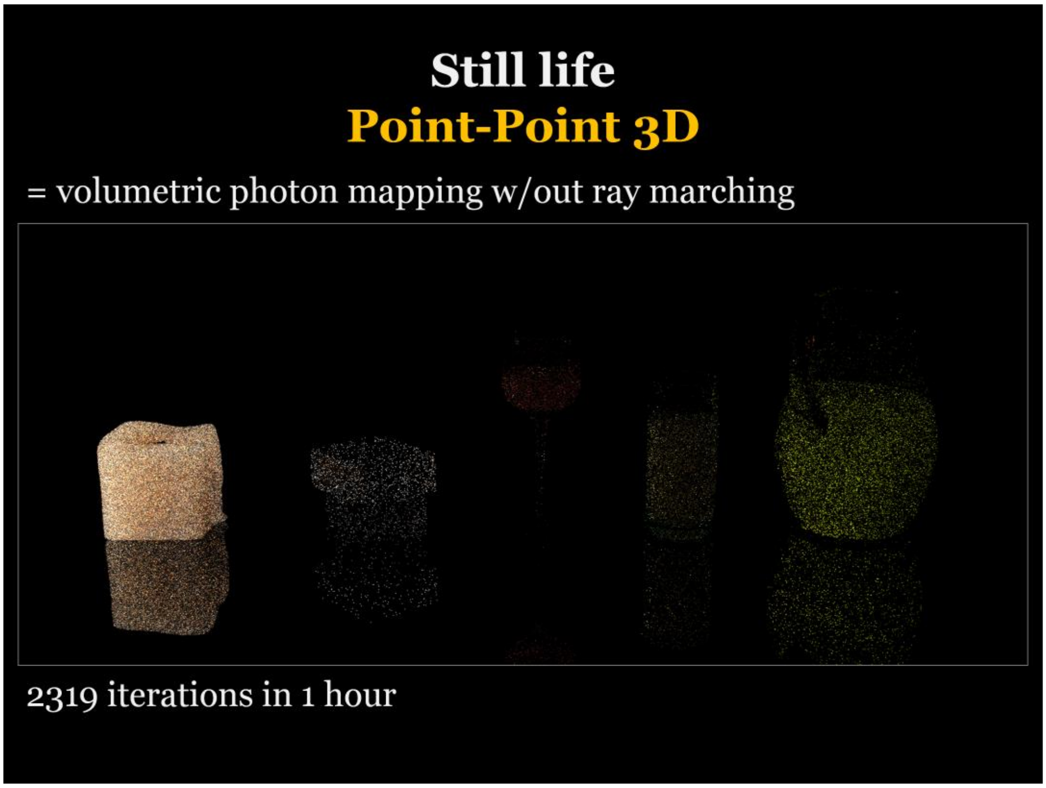
Acknowledgements

- **Funding**
  - Czech Science Foundation
  - Charles University in Prague
  - Natural Sciences and Engineering Research Council of Canada
- Chaos Group
- Ondra Karlík
- Anonymous reviewers

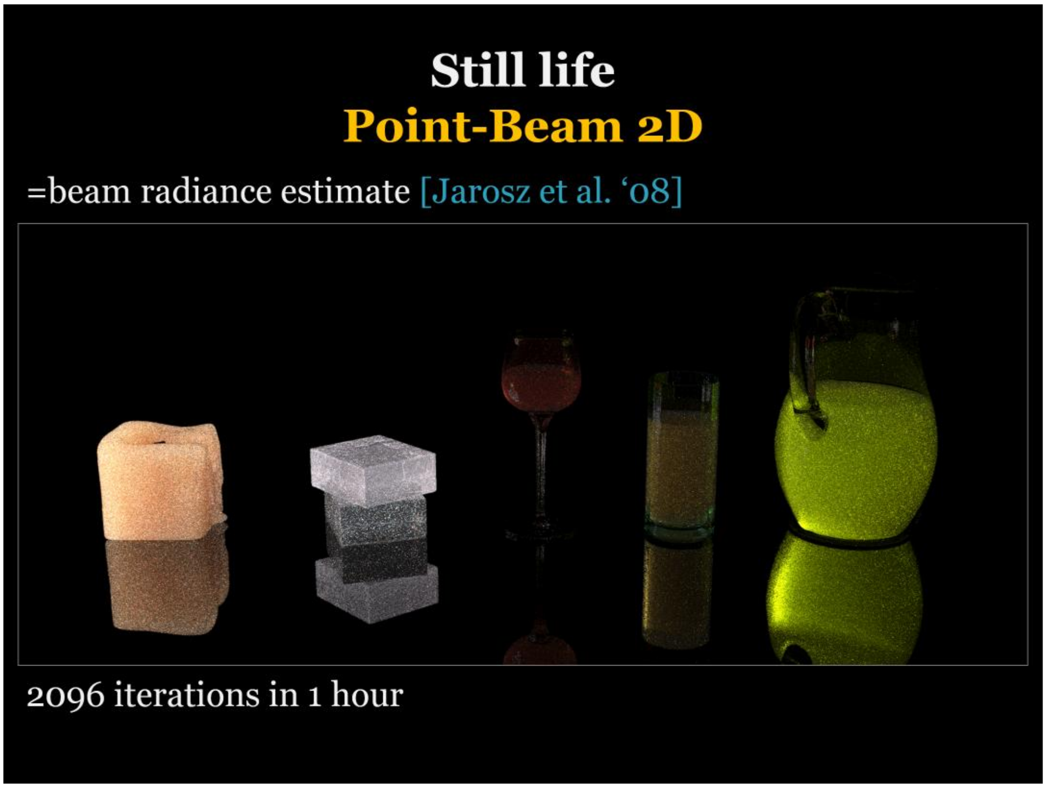
- Let me thanks the funding agencies, Chaos Group and Ondra Karlík for the scenes, and the SIGGRAPH reviewers for their helpful comments.

- I want to thank you for your attention.

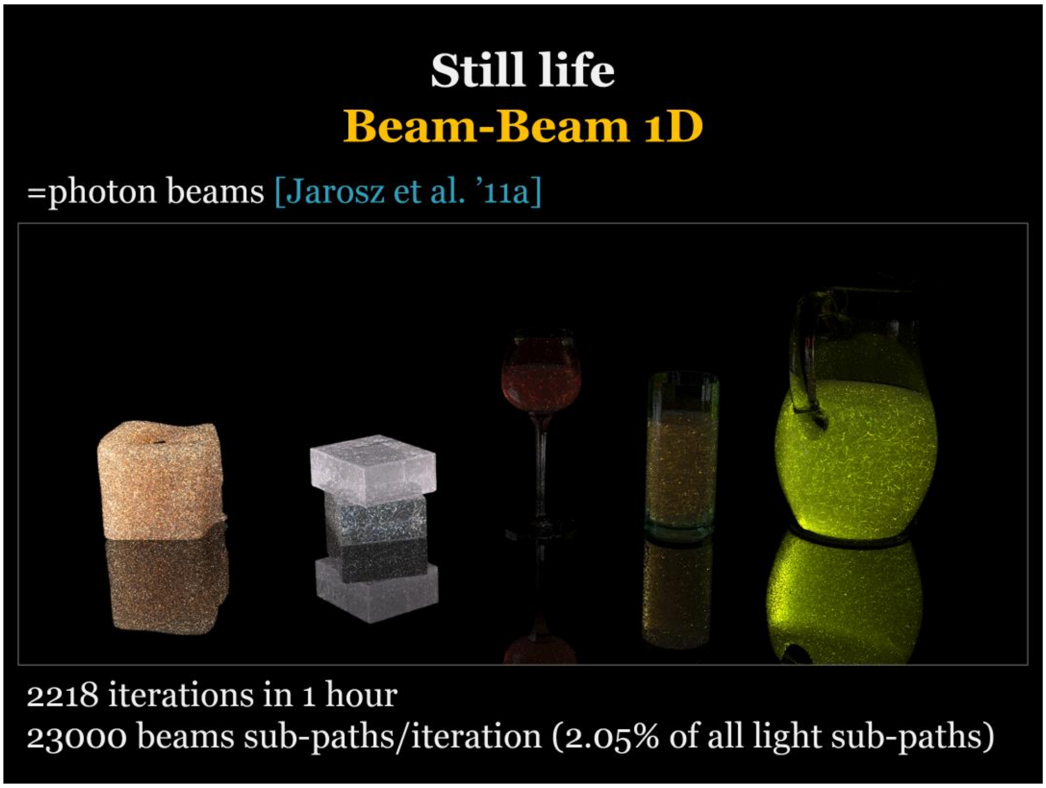
ADDITIONAL SLIDES



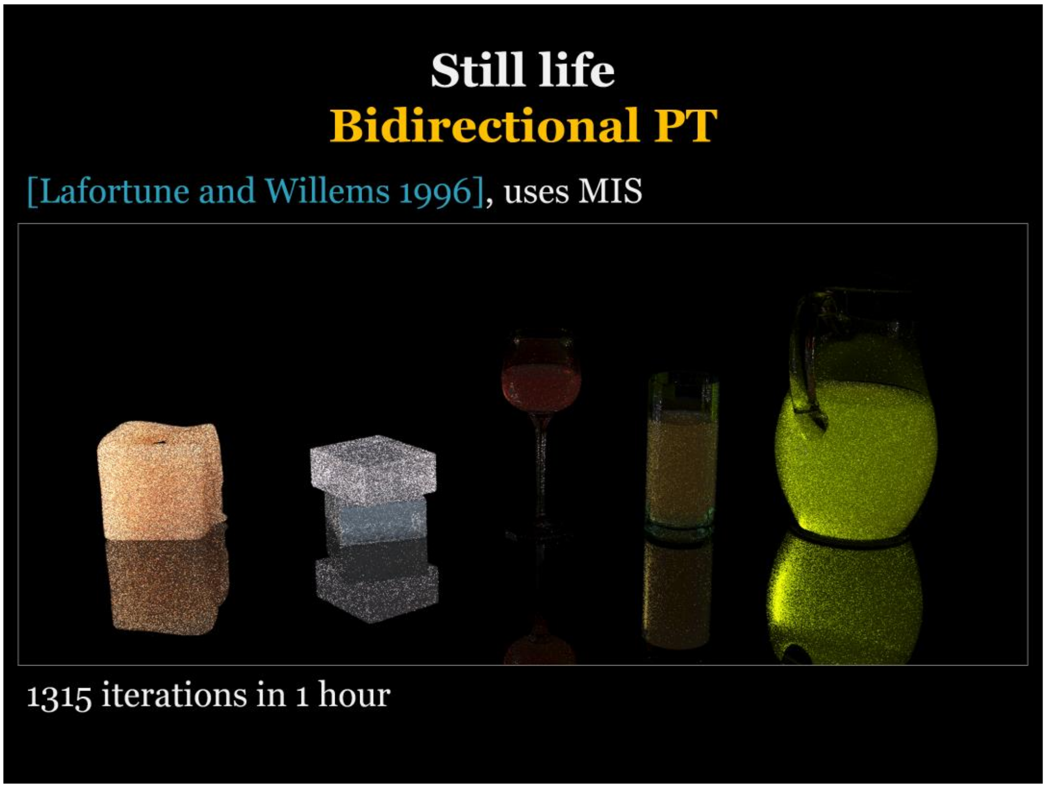
- The Point-Point 3D estimator is equivalent to volumetric photon mapping but without ray-marching.
- So there’s a single point along the camera ray where we look-up the photons.
- You can see that there’s quite some noise even after one hour of rendering.



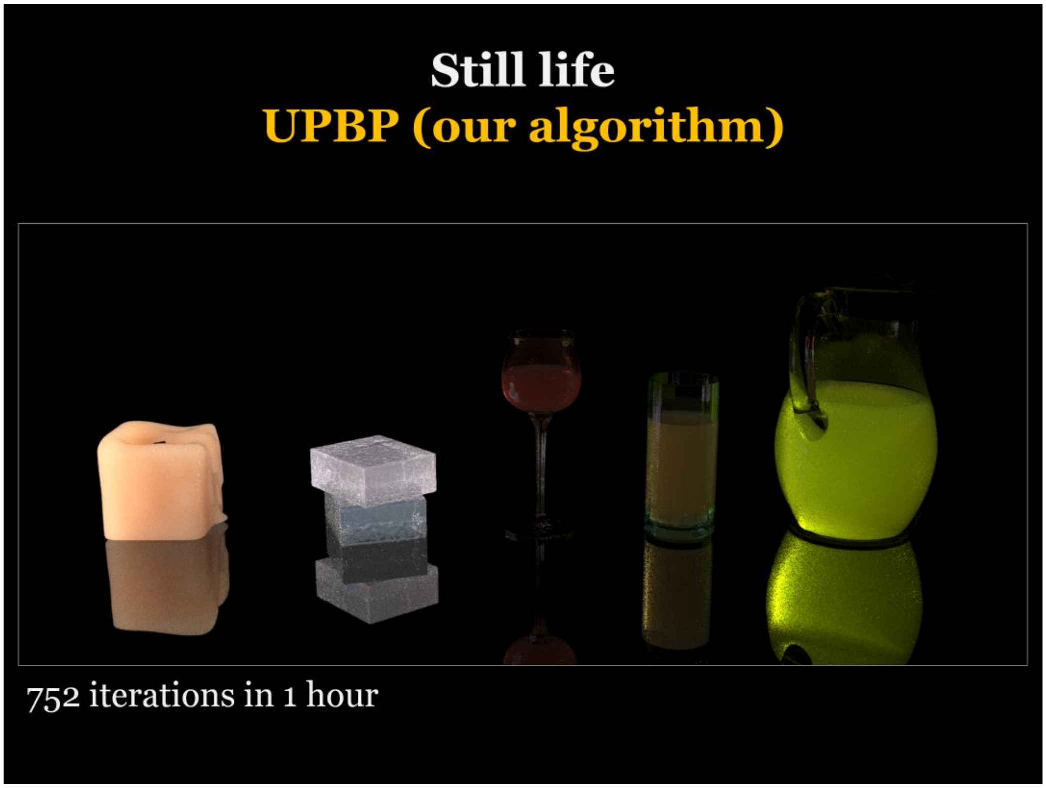
- Now we move to the Point-Beam 2D estimator, that is the beam radiance estimate.
- The performance in terms of the number of iterations is lower but you can see that the image quality has much improved.



- Now let us move to the beam-beam 1D estimator, or photon beams.
- I have to point out that the number of stored beams is only 2% of the number of photon points.
- This is because building the data structure and looking-up beams is much slower than for points.
- So we've selected the number of beams that leads to roughly the same iteration time.
- And we see that while the results in the soap block have improved, in other, dense media, the results are not great.



- Next, bidirectional path tracing does not work very well here because much of the illumination is essentially due to reflected caustics.
- But the bidir path tracer is really important for rendering surfaces and the surface-to-medium transport.



- And finally our combined algorithm achieves the best result in spite of the fact that it only manages to run about 750 iteration in one hour.