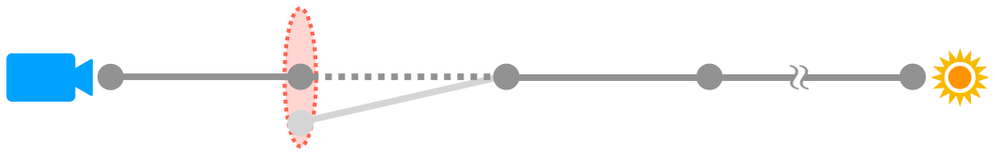

SIGGRAPH
2025
The Premier Conference & Exhibition on
Computer Graphics & Interactive Techniques
SEGMENT-BASED LIGHT TRANSPORT SIMULATION
WENYOU WANG, REX WEST, TOSHIYA HACHISUKA

Background

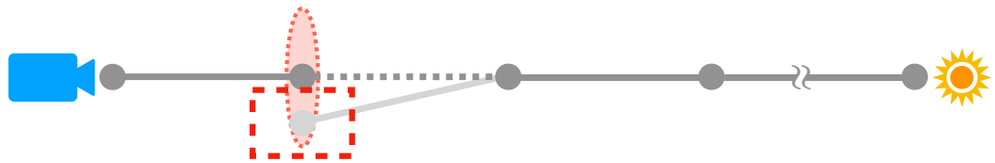
Classic Monte Carlo Rendering methods simulate light transport by sampling light paths with connected vertices.

Background



Alternatively, we could also consider sampling disconnected sub-paths and then construct complete paths. Typically, these methods can enable efficient path reuse based on spatial proximity.

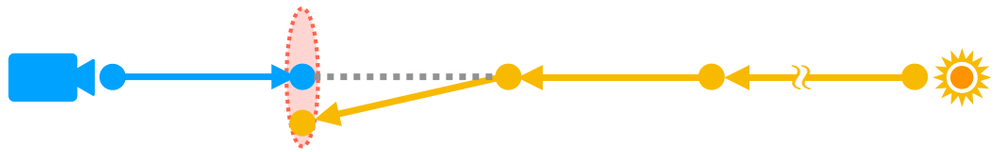
Background



But also keep in mind that the disconnection is not free because it will introduce extra vertices. A proper formulation that supports these extra vertices are necessary.

Context

- Photon density estimation [Jensen 1996]

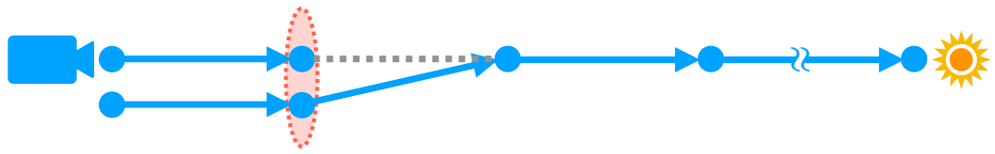


Consider photon density estimation as an example. The endpoint of a light sub-path is disconnected from the endpoint of a camera sub-path. The extra vertices are considered as random variables for density estimation.

Context

SIGGRAPH 2025
Vancouver+ 10-14 August

- Path-space filtering [Keller et al. 2014]



And we can also explore constructing full paths with disconnected camera sub-paths using path space filtering that efficiently reuses sub-paths by filtering over the kernel.

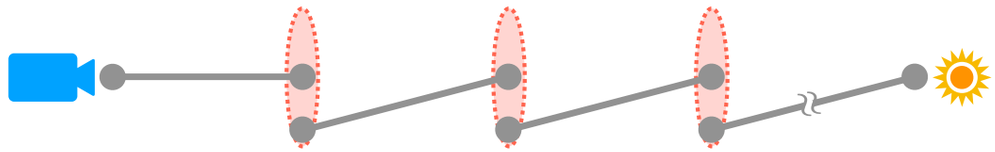
Context

- The extended path integral formulation[Hachisuka et al. 2012]

Different from these two methods that try to support extra vertices under the original path formulation, we could also address the extra vertices at the level of formulation. This is the key idea of the extended path integral.

Background

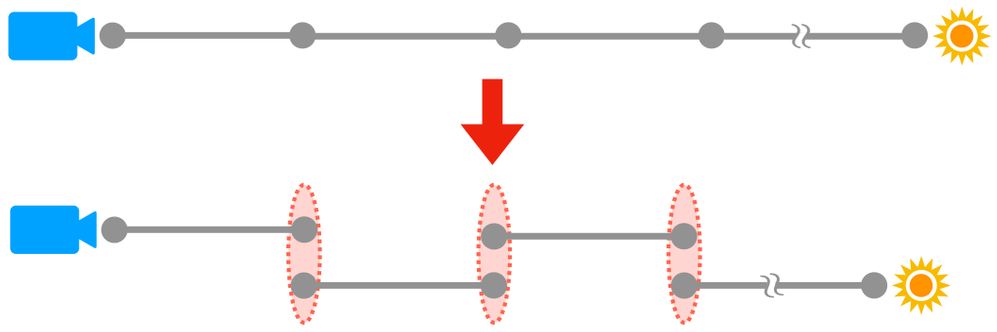
- The extended path integral formulation[Hachisuka et al. 2012]



Under the extended path integral, we directly extend the space of the original path integral by inserting spatial kernel functions. So the path samples with disconnections are just usual monte carlo samples in the extended path space. In its extreme form, we could consider disconnection at every path vertex.

Background

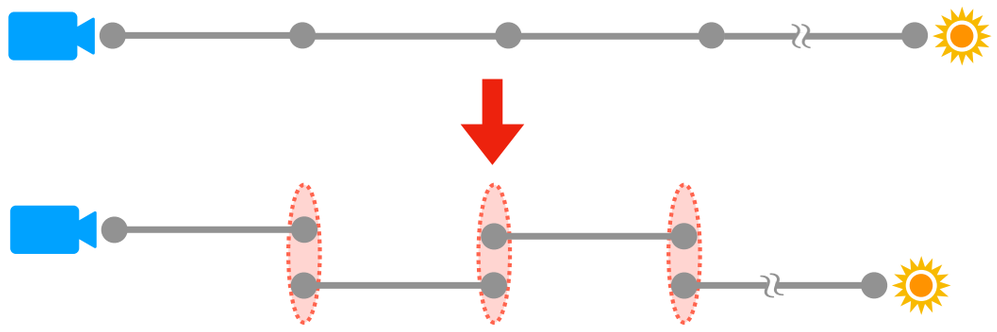
- The extended path integral formulation[Hachisuka et al. 2012]



A very interesting way to look at this form is it actually turns path sampling from sampling a sequence of vertices into sampling a sequence of disconnected segments.

Background

- The extended path integral formulation[Hachisuka et al. 2012]



- The concept of segment sampling has been left unexplored

Despite this segment-based view, the existing rendering algorithms did not really explore this concept of segment sampling and consider sampling segments as the basic path-construction primitives.

Background

- We revisit this formulation and introduce a segment-based light transport framework.

We revisit this formulation and introduce a segment-based light transport framework.

Background

- We revisit this formulation and introduce a segment-based light transport framework.
- Our frameworks includes:
    - A segment-based path integral formulation
    - Segment sampling techniques
    - Practical estimation strategies

And our frameworks includes a segment-based formulation, sampling techniques for segments and estimation strategies for segment paths.

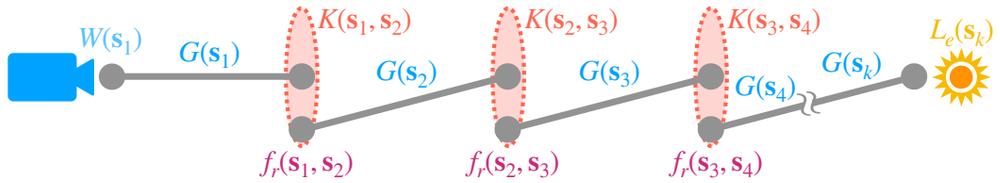
Background

- We revisit this formulation and introduce a segment-based light transport framework.
- Our frameworks includes:
  - A segment-based path integral formulation
  - Segment sampling techniques
  - Practical estimation strategies
- We show how our framework enables a performant bidirectional filtering algorithm

As one example, we show how our framework enables a performant bidirectional filtering algorithm

Our method - formulations

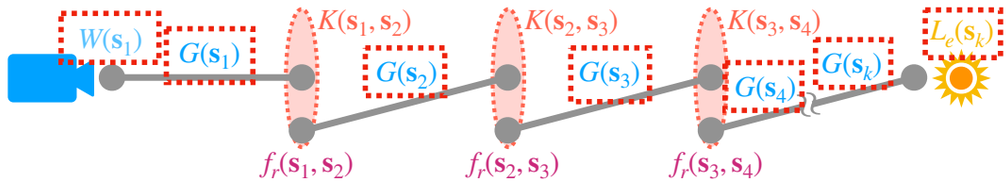
- Segment path contribution function



We first rewrite the terms in the original extended path integral formulation using segments.

Our method - formulations

- Segment path contribution function

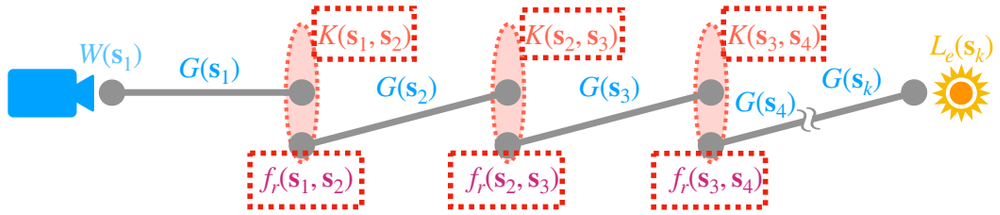


- $W(s_1)$, $G(s_i)$, $L_e(s_k)$ are effectively unchanged from the extend path integral formulation

The responsivity term W, geometry term G, and emission term Le are effectively unchanged because their original definitions are defined with a segment or a pair of vertices.

Our method - formulations

- Segment path contribution function

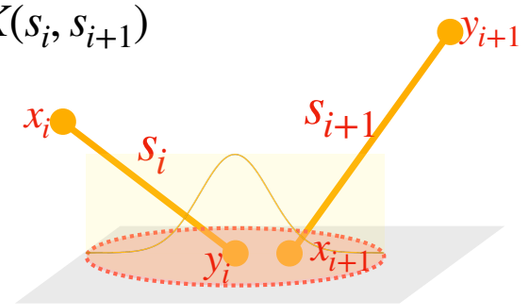


- $W(s_1)$, $G(s_i)$, $L_e(s_k)$ are effectively unchanged from the extend path integral formulation
- $K(s_i, s_{i+1}), f_r(s_i, s_{i+1})$ are operating over pairs of segments

However, the kernel function and the segment-based BSDF term are now operating on a pair of segments, so we need to redefine them.

Our method - formulations

- Kernel function $K(s_i, s_{i+1})$



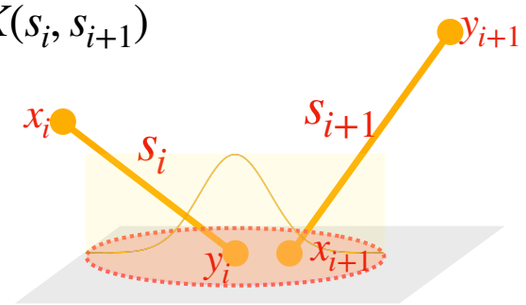
- A practical uniform kernel function

$$K(s_i, s_{i+1}) = \frac{1}{(\int_{\mathcal{K}(y_i)} dx_{i+1})}$$

A simple choice of kernel function is the uniform kernel function and as we demonstrated in the paper’s results, it is actually good and effective.

Our method - formulations

- Kernel function $K(s_i, s_{i+1})$



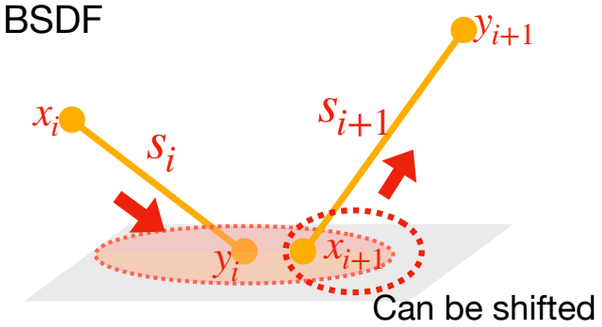
- A practical uniform kernel function
- Directional kernel functions are accessible via our formulation

But you could also consider advanced variants like directional kernel function since our formulation allows the kernel function to be function of segment directions.

Our method - formulations

SIGGRAPH 2025
Vancouver+ 10-14 August

- Segment-based BSDF



- The shift invariant form

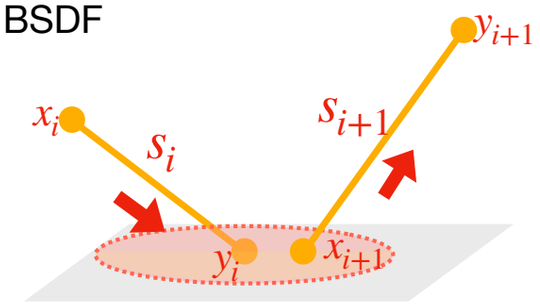
$$f_r(s_i, s_{i+1}) = f_r(x_i, y_i, y_{i+1} - (x_{i+1} - y_i))$$

As for the segment-based BSDF term, we define this shift invariant form, which is consistently equal to the standard triplet BSDF no matter how we shift the vertex x{i+1}.

Our method - formulations

SIGGRAPH 2025
Vancouver+ 10-14 August

- Segment-based BSDF

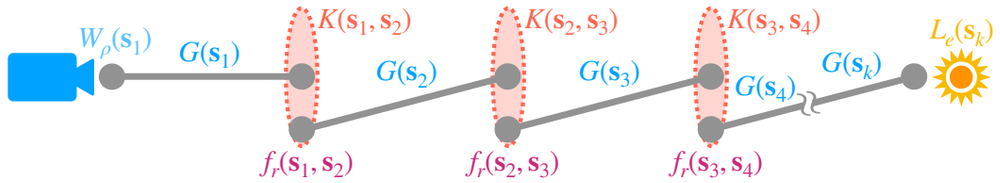


- The shift invariant form
- Aligns with the actual implementation of filtering methods[Deng et al. 2021;West et al. 2022]

And this form aligns closely with the actual implementation of previous filtering methods where incoming radiance is reused without any correction.

Our method - formulations

- Segment path contribution function

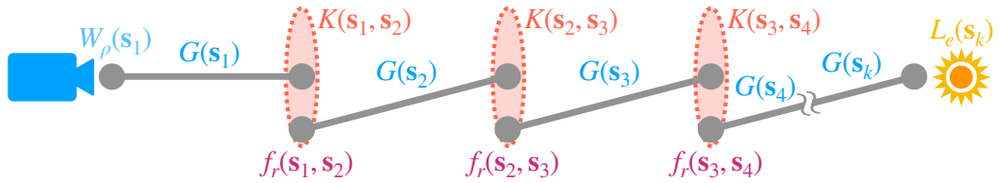


$$g(\bar{s}) = W(s_1)\prod_{i=1}^{k} G(s_i)\prod_{i=1}^{k-1} K(s_i, s_{i+1}) f_r(s_i, s_{i+1}) L_e(s_k)$$

With these definitions, we could express the segment path contribution function g as a product of the terms.

Our method - formulations

- Segment path contribution function

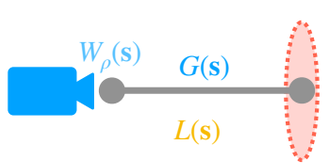


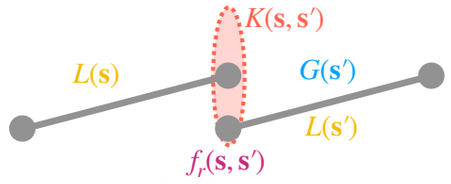
$$J = \int_{\mathcal{S}} g(\bar{s}) d\bar{s}$$

Then the segment path integral is naturally defined as an integral of the contribution function over the segment path space.

Our method - formulations

- Segment path contribution function





$$J = \int_{\mathcal{S}_1} W(s)G(s)L(s)d(s)$$

$$L(s) = L_e(s) + \int_{\mathcal{S}_1} K(s,s')f_r(s,s')G(s')L(s')ds'$$

Pixel integral

Recursive segment integral

And we could also rewrite the segment path integral into a recursive form so we get a segment-based redefinition of the rendering equation.

Our method

- Similar to vertex-base counterparts, we could characterize our methods into a two-step process:

① sampling a set of segments.

② constructing estimators based on the sampled segments.

With the formulations, now we could start to consider sampling and estimation. Following a two-step process as in vertex-based methods we first samples a set of segments and then we construct the estimators with the segment samples.

Our method - segment samplers

- Contribution function considers sequence of segments not independent segments

Let’s start from sampling segments first. The path contribution function is defined on a sequence of segments. For importance sampling, It is natural to consider sampling a sequence of segment rather than independent segments.

Our method - segment samplers

- Contribution function considers sequence of segments not independent segments
- Efficient segment sampler - sequential segment sampler

Based on this, we introduce the sequential segment samplers, which is very similar to the vertex-based path sampler.

Our method - segment samplers

SIGGRAPH 2025
Vancouver+ 10-14 August

- Contribution function considers sequence of segments not independent segments
- Efficient segment sampler - sequential segment sampler
- Start sampling the sequence from the camera

$$p_W(s_1) \sim W(s_1)G(s_1)$$

We could start sampling the first segment either from the camera by importance sampling the responsivity function,

Our method - segment samplers

- Contribution function considers sequence of segments not independent segments
- Efficient segment sampler - sequential segment sampler
- Start sampling the sequence from the camera

$$p_W(s_1) \sim W(s_1)G(s_1)$$

- Or start sampling the sequence from light sources

$$p_L(s_1) \sim L_e(s_1)G(s_1)$$

Or start from the light sources by importance sampling the emission term.

Our method - segment samplers

- Contribution function considers sequence of segments not independent segments
- Efficient segment sampler - sequential segment sampler
- Start sampling the sequence from the camera

$$p_W(s_1) \sim W(s_1)G(s_1)$$

- Or start sampling the sequence from light sources

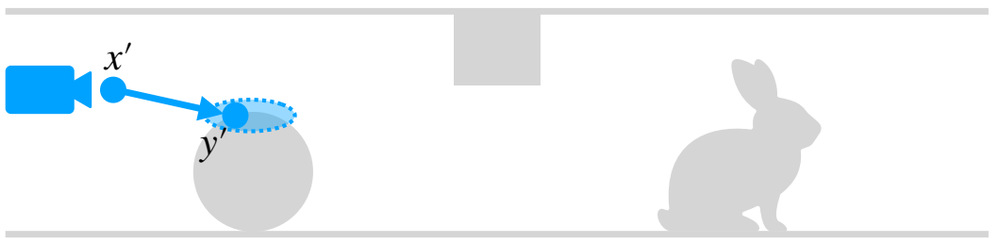
$$p_L(s_1) \sim L_e(s_1)G(s_1)$$

- Repeat conditional sampling after the first segment generated

Then after we have the first segment, we could repeat doing conditional sampling to form a segment sequence.

Our method - segment samplers

- Efficient segment sampler - sequential segment sampler

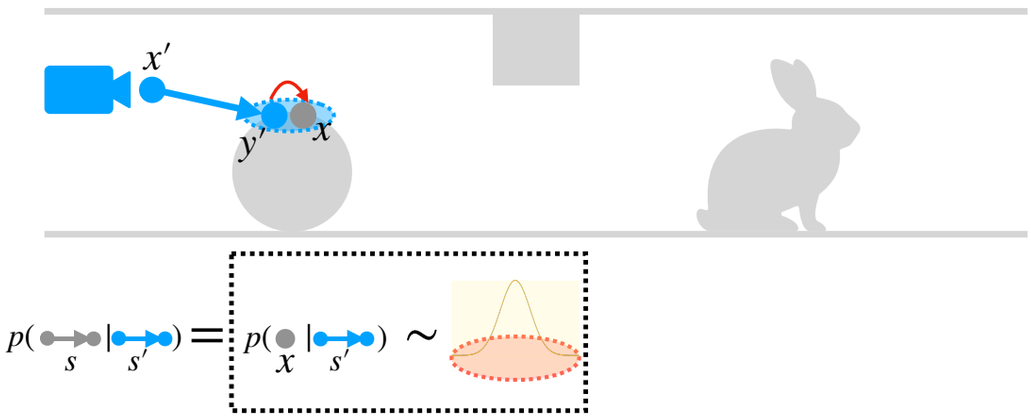


$p($  $) =$

Assume we start from the camera and have the first segment generated,

Our method - segment samplers

- Efficient segment sampler - sequential segment sampler

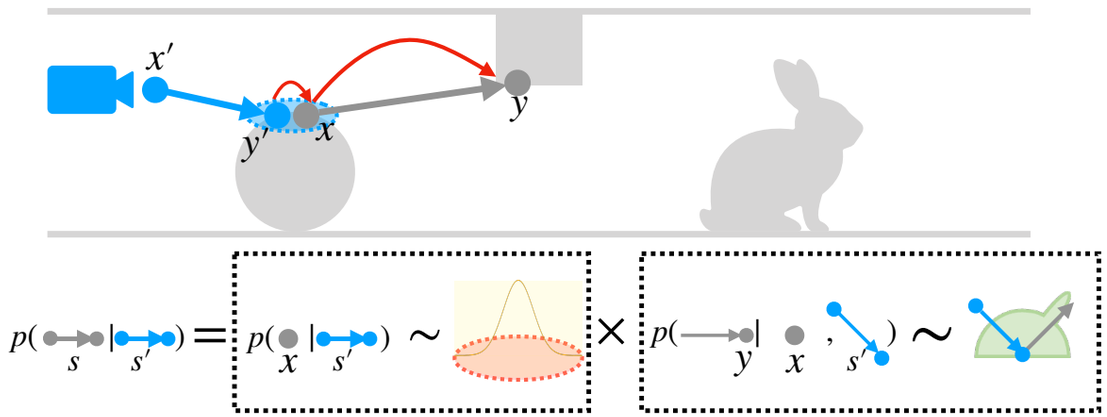


We first importance sampling the kernel function to get the first vertex of the segment.

Our method - segment samplers

SIGGRAPH 2025
Vancouver+ 10-14 August

- Efficient segment sampler - sequential segment sampler

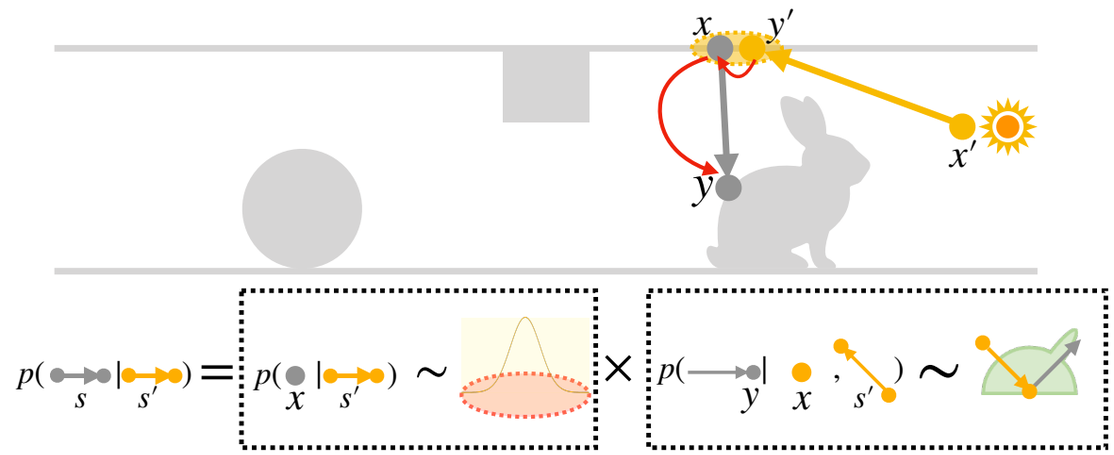


Then we consider the previous segment s’ as an auxiliary direction that conditions the sampling of next segment direction. So by importance sampling the BSDF term given the direction of s’, we get the new direction. Then we trace a ray from the first vertex x to get the second vertex y.

Our method - segment samplers

SIGGRAPH 2025
Vancouver+ 10-14 August

- Efficient segment sampler - sequential segment sampler

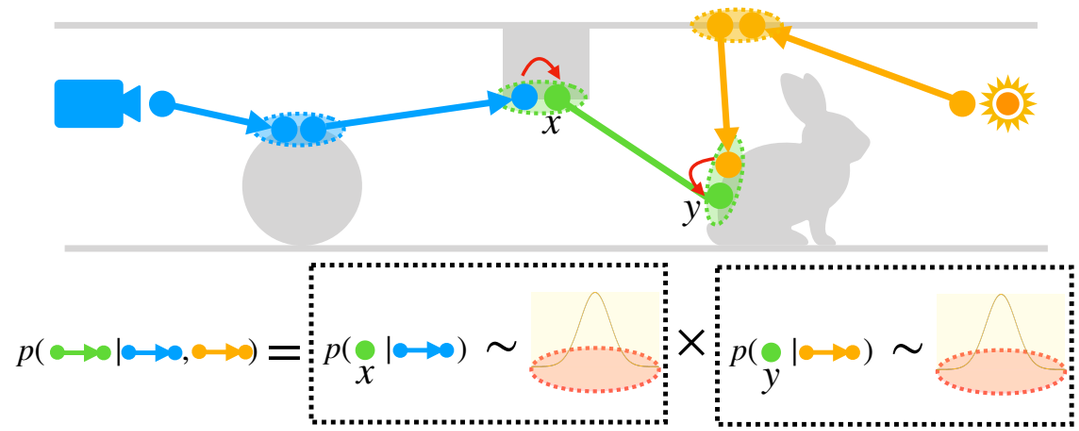


Similarly, we could also sample segments starting from light sources to get a sequence of segments.

Our method - segment samplers

SIGGRAPH 2025
Vancouver+ 10-14 August

- Efficient segment sampler - bridge segment sampler



Another interesting variant is what we call the bridge segment sampler. Basically, we importance-sample the two kernel functions defined by a pair of segments—one from the camera side and one from the light side, to form a segment, which is very much like the connections in classic path sampling.

Our method - estimation strategies

- Sequential estimation

$$\langle J_k \rangle = \frac{g(\overline{s})}{p(\overline{s})} \ , \ p(\overline{s}) = p(s_1, s_2, \ldots, s_k)$$

Given these segment sampling techniques, now we could look at how we
use them to construct estimators. The first and the most straightforward one is the sequential estimation strategy.

Our method - estimation strategies

- Sequential estimation

$$\langle J_k \rangle = \frac{g(\overline{s})}{p(\overline{s})}\ ,\ p(\overline{s}) = p(s_1, s_2, \ldots, s_k)$$

- Under-utilize the potential of segment path sampling

But this strategy under-utilizes the potential of segments. Making a single segment path from the sampled segments is quite wasteful. Ideally, we want to reuse segments as much as possible.

Our method - estimation strategies

- Sequential estimation

$$\langle J_k \rangle = \frac{g(\overline{s})}{p(\overline{s})} \ , \ p(\overline{s}) = p(s_1, s_2, \ldots, s_k)$$

- Under-utilize the potential of segment path sampling
- Multiple sampling techniques

$$\langle J_k \rangle_{MIS} = \sum_{i=1}^{T} \sum_{j=1}^{n_i} \frac{w_i(\overline{s}_{i,j}) g(\overline{s}_{i,j})}{n_i p(\overline{s}_{i,j})} \ , \ \sum_{i=1}^{T} w_i(\overline{s}) = 1$$

In the spirit of BDPT, we can use MIS to combine different segment path sampling techniques.

Our method - estimation strategies

- Sequential estimation

$$\langle J_k \rangle = \frac{g(\overline{s})}{p(\overline{s})} \ , \ p(\overline{s}) = p(s_1, s_2, \ldots, s_k)$$

- Under-utilize the potential of segment path sampling
- Multiple sampling techniques

$$\langle J_k \rangle_{MIS} = \sum_{i=1}^{T} \sum_{j=1}^{n_i} \frac{w_i(\overline{s}_{i,j}) g(\overline{s}_{i,j})}{n_i p(\overline{s}_{i,j})} \ , \ \sum_{i=1}^{T} w_i(\overline{s}) = 1$$

- Sampling cost is amortized compared to sequential estimation

By reusing segment sub-paths, sampling cost is amortized, so it is much better than the sequential one.

Our method - estimation strategies

- Recursive estimation

$$L(s) = L_e(s) + \int_{\mathcal{S}_1} K(s, s') f_r(s, s') G(s') L(s') d\, s'$$

But that’s not the end of the story. We could Push reusing further with the recursive form.

Our method - estimation strategies

- Recursive estimation

$$L(s) = L_e(s) + \int_{\mathcal{S}_1} K(s,s')f_r(s,s')G(s')L(s')d\,s'$$

$$\langle L(s)\rangle \approx L_e(s) + \frac{1}{N}\sum_{s'}^{N} \frac{K(s,s')f_r(s,s')G(s')\langle L(s')\rangle}{p(s')}$$

A simple application is just to directly build a basic recursive monte carlo estimator for the integral so we consider segments with non-zero kernel function as continuation segments for each recursive expansion.

Our method - estimation strategies

- Recursive estimation

$$L(s) = L_e(s) + \int_{\mathcal{S}_1} K(s,s')f_r(s,s')G(s')L(s')d\,s'$$

$$\langle L(s)\rangle \approx L_e(s) + \frac{1}{N}\sum_{s'}^{N} \frac{K(s,s')f_r(s,s')G(s')\langle L(s')\rangle}{p(s')}$$

- continuation segment samples may have been sampled by different segment sampling techniques

However, the set of continuation segment samples during the recursive process may have been sampled by different sampling techniques, so basic monte carlo estimator is insufficient.

Our method - estimation strategies

- Marginal multiple importance sampling recursive estimator

$$\langle L(s)\rangle \approx L_e(s) + \frac{1}{N}\sum_{s'}^{N} \frac{K(s,s')f_r(s,s')G(s')\langle L(s')\rangle}{p(s')}$$

MMIS

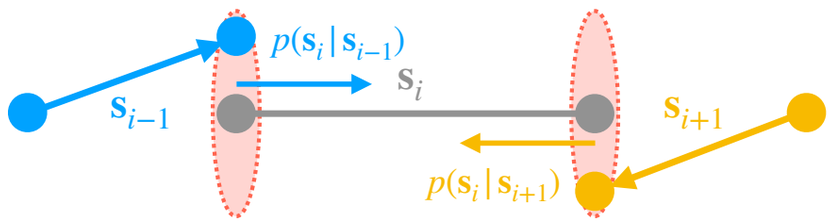
$$\langle L(s)\rangle \approx L_e(s) + \sum_{i=1}^{T}\sum_{j=1}^{N_i} \frac{K(s,s_{i,j})f_r(s,s_{i,j})G(s_{i,j})\langle L(s_{i,j})\rangle}{\sum_{i'=1}^{T}\sum_{j'=1}^{N_i} p(s_{i,j}\,|\,t_{i',j'})}$$

To handle the situations we could directly apply the theory of marginal multiple importance sampling that supports combining different conditional sampling techniques based on auxiliary variable t. The MMIS recursive estimator allows us to combine segments generated with different techniques.

Our method - estimation strategies

SIGGRAPH 2025
Vancouver+ 10-14 August

- Marginal multiple importance sampling recursive estimator



For example, if we consider to use both camera and light sequential sampler, an arbitrary segment s_i in the middle can be generated conditionally on segments that comes from either direction.

Our method - Practical algorithms

- Segment-based classic light transport algorithms

Path tracing = Camera Segment Sampler ⊕ Sequential estimation

Light tracing = Light Segment Sampler ⊕ Sequential estimation

Now with the samplers and estimation strategies, we could start to design some algorithms. For example, in the segment world, path tracing or light tracing is just a combination of sequential estimation with the corresponding segment sampler.

Our method - Practical algorithms

SIGGRAPH 2025
Vancouver+ 10-14 August

- Segment-based classic light transport algorithms

Path tracing = Camera Segment Sampler ⊕ Sequential estimation

Light tracing = Light Segment Sampler ⊕ Sequential estimation

BDPT = Camera/Light Segment Sampler ⊕ Bridge Segment Sampler ⊕ Multiple sampling techniques

And we could also combine them with the bridge segment sampler using a MIS estimator, which leads to the segment-based BDPT.

Our method - Practical algorithms

SIGGRAPH 2025
Vancouver+ 10-14 August

- Segment-based classic light transport algorithms

Path tracing = Camera Segment Sampler ⊕ Sequential estimation

Light tracing = Light Segment Sampler ⊕ Sequential estimation

BDPT = Camera/Light Segment Sampler ⊕ Bridge Segment Sampler ⊕ Multiple sampling techniques

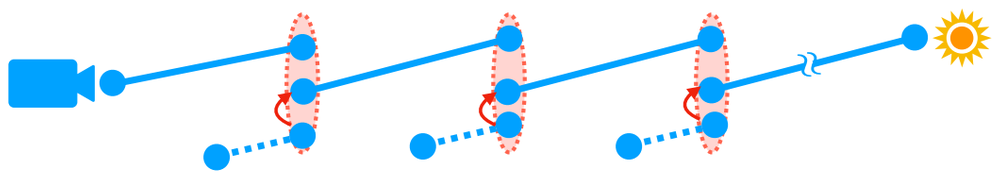
- Theoretically intriguing, but do not offer tangible benefit over their vertex-based counterparts

While these variants are theoretically intriguing, they do not offer tangible benefits compared to vertex-based counterparts.

Our method

SIGGRAPH 2025
Vancouver+ 10-14 August

- Generalization of previous methods



Multi-vertex filtering = Camera Segment Sampler ⊕ Recursive Estimation

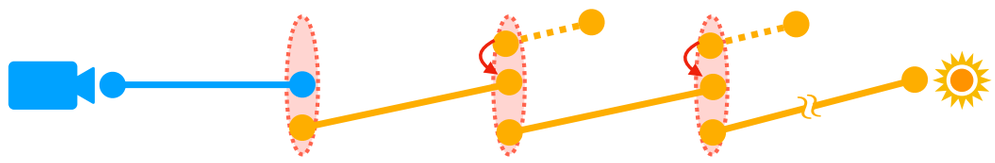
[Deng et al. 2021]

But If we start to think of using the recursive strategy, we will have some interesting variants. For example, multi-vertex filtering—which performs filtering multiple times along a path—can be generalized simply by plugging our camera segment sampler into the recursive MMIS estimator so we get a segment-based multi-vertex filtering.

Our method

SIGGRAPH 2025
Vancouver+ 10-14 August

- Generalization of previous methods



Photon filtering = Light Segment Sampler ⊕ Recursive Estimation

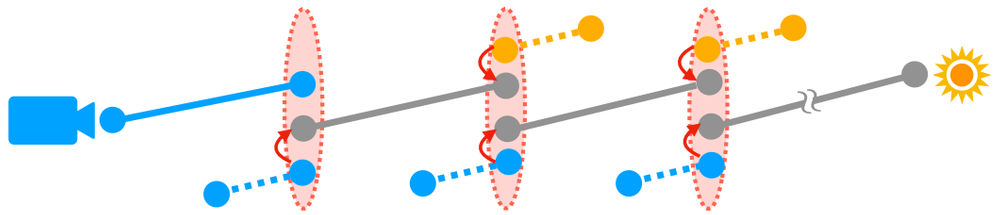
[West et al. 2022]

Likewise, photon filtering—can be seen as using our light segment sampler together with the recursive estimator.

Our method

SIGGRAPH 2025
Vancouver+ 10-14 August

- Combining samplers enables bidirectional filtering



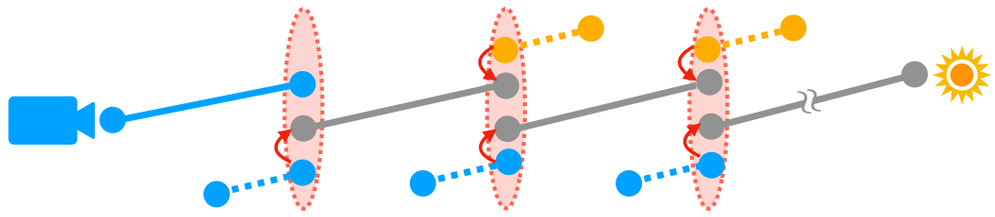
Bidirectional filtering = Camera Segment Sampler ⊕ Light Segment Sampler ⊕ Recursive Estimation

Moreover, to unify previous filtering approaches, it is just as simple as combine the camera and light segment sampler with recursive estimation. This bring us a bidirectional filtering algorithm.

Our method

SIGGRAPH 2025
Vancouver+ 10-14 August

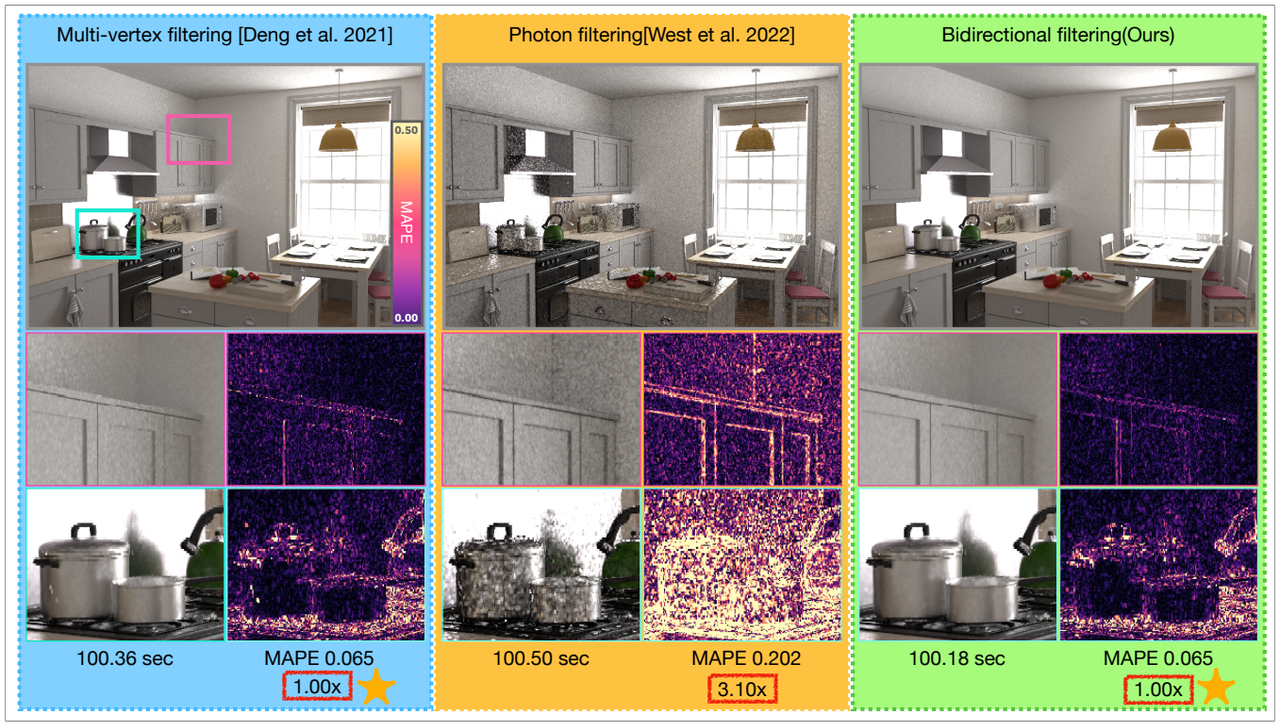
- Combining samplers enables bidirectional filtering



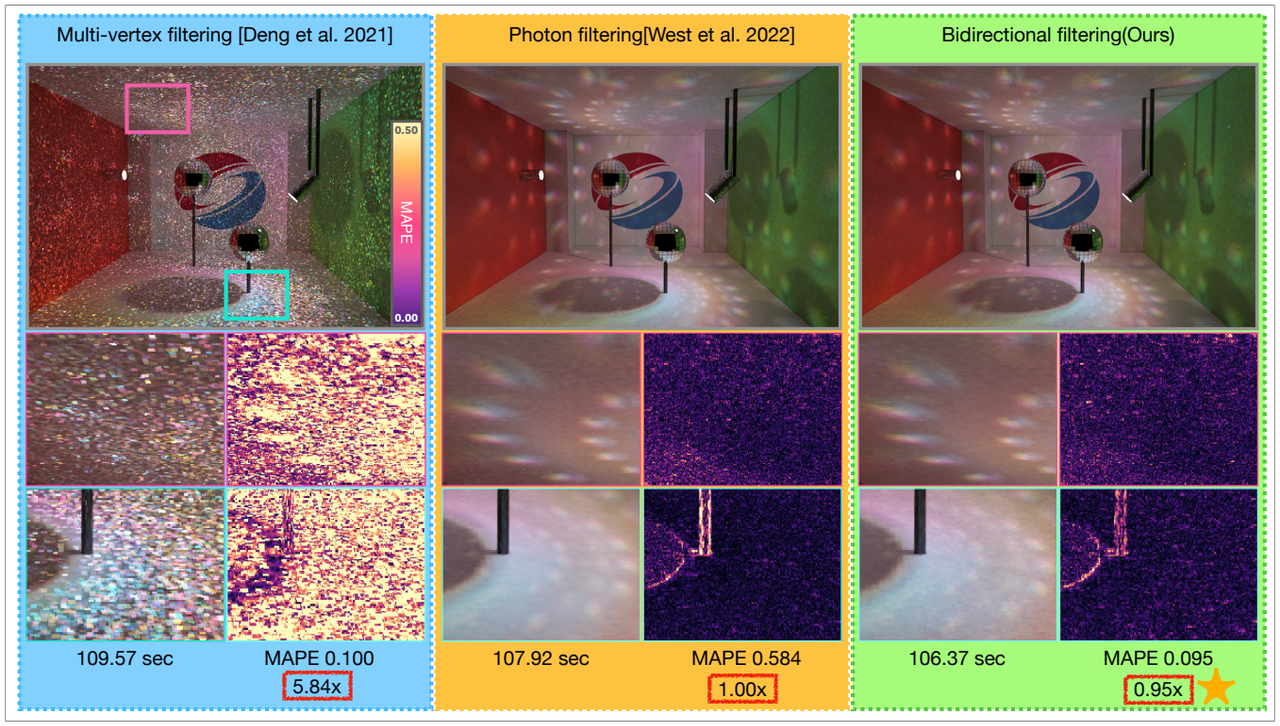
Bidirectional filtering = Camera Segment Sampler ⊕ Light Segment Sampler ⊕ Recursive Estimation

- Leverage the strength of filtering methods from both ends, avoid failure cases

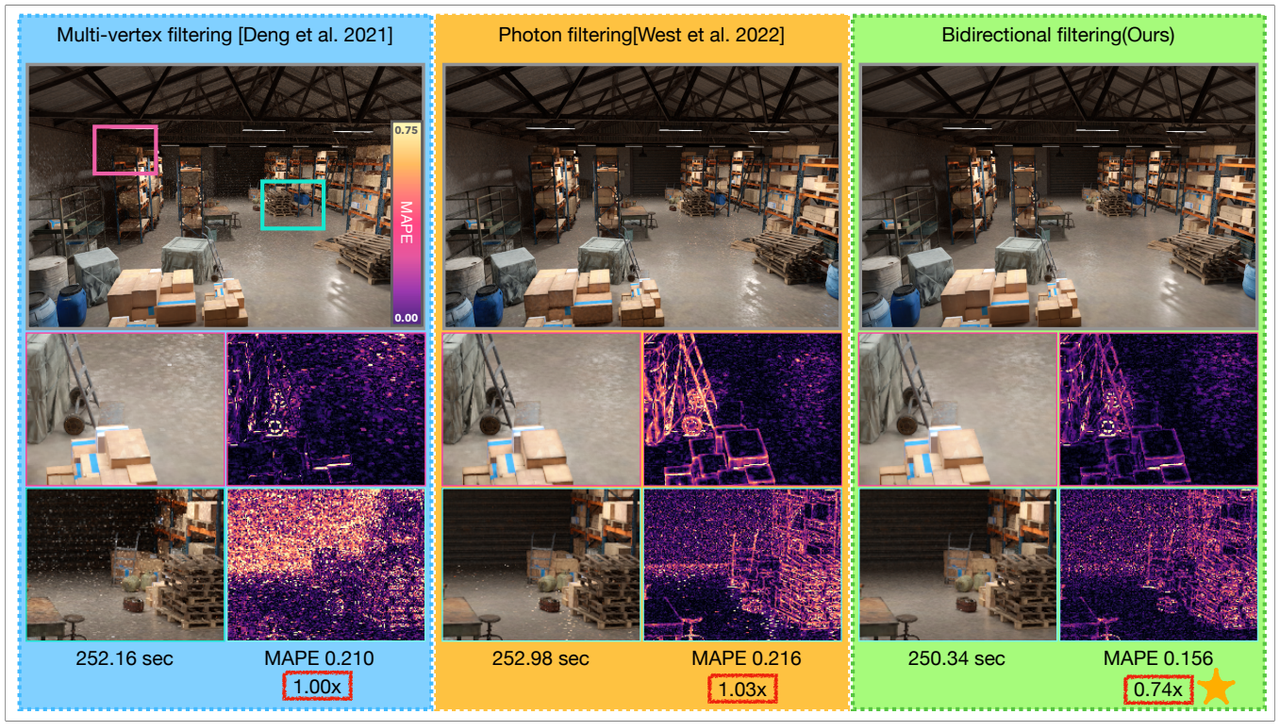
As we will demonstrate in the results, bidirectional filtering is robust, it inherits the strength of both prior unidirectional filtering, but with none of their failure cases



To see how our method performs, let’s see some equal-time comparison results. In this kitchen scene, multi-vertex filtering performs pretty well, while photon filtering struggles. Even our method combines both techniques, it still matches the performance of the better one, the multi-vertex filtering.



In this disco-ball scene with challenging caustics, the multi-vertex filtering struggles to capture the caustics, while our method matches—and even slightly surpasses—the performance of photon filtering.



For the warehouse scene—it provides an ideal case for our bidirectional filtering. This scene has indirect area lights, glossy materials, and very complex occlusion geometry. Thus our approach can outperform both unidirectional filtering variants and bring impressive improvements.

Future work

- Ideal distribution of sampling segments is unknown

  Segment guiding or introducing MCMC segment samplers

To wrap up, let’s look at future work directions.
Currently we just combine segments sampled from different distributions. One interesting question to ask what would the single ideal distribution for sampling segments?
If we can characterize such a distribution, we can develop a segment-guiding sampler—or even a MCMC-based sampler.

Future work

- Ideal distribution of sampling segments is unknown

  Segment guiding or introducing MCMC segment samplers

- Filtering not only spatially but also temporally

- Local filtering is still as slow as O(n^2)

About the filtering part, we’re still limited to spatial kernels—so exploring temporal filtering could also be promising.
At the same time, local filtering is still costly, so reducing its complexity can also be valuable.

Future work

- Ideal distribution of sampling segments is unknown

  Segment guiding or introducing MCMC segment samplers

- Filtering not only spatially but also temporally

- Local filtering is still as slow as O(n^2)

- Volumetric segment-based formulations

Finally, I think it would be interesting to extend our segment-based framework to volumetric rendering—building on prior work in volumetric photon-density estimation.

Okay, That brings me to the end of my talk. Thank you all for your attention!