Light Transport Simulation in the Gradient Domain

BINH-SON HUA, The University of Tokyo
ADRIEN GRUSON, The University of Tokyo and JFLI CNRS UMI 3527
MATTHIAS ZWICKER, University of Maryland
TOSHIYA HACHISUKA, The University of Tokyo

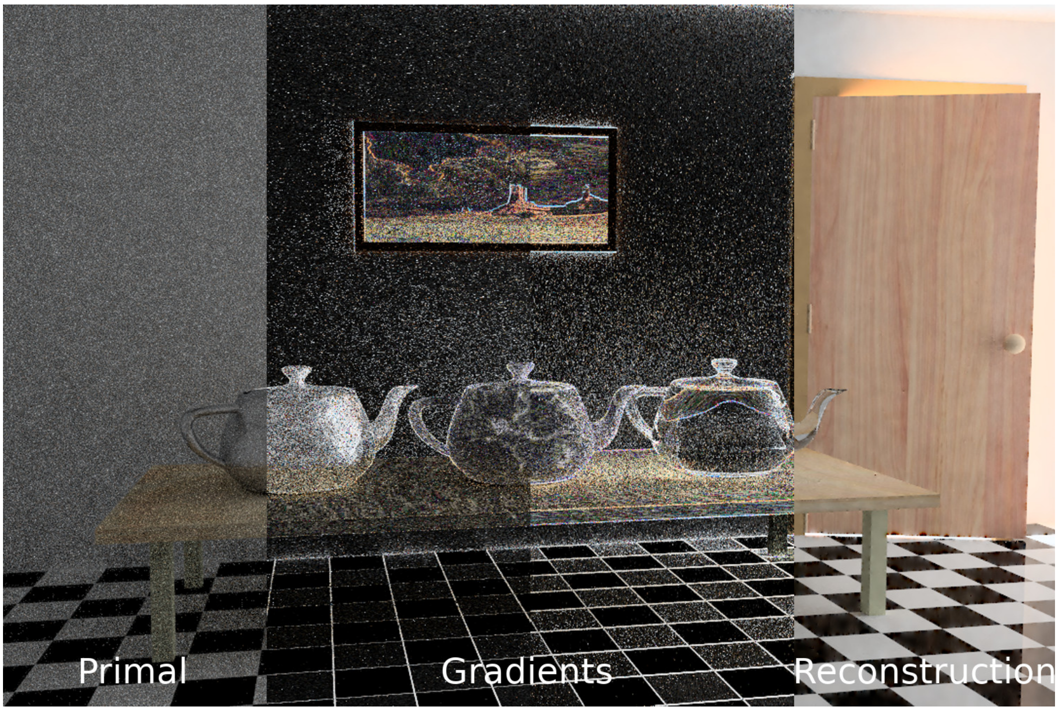


Fig. 1. Gradient-domain light transport.

Despite the wide adoption in film production and animation industry nowadays, Monte Carlo light transport simulation is still prone to producing noisy images within short rendering time. Accelerating the convergence of Monte Carlo rendering without sacrificing its accuracy is by far a challenging task. In this course, we will learn about gradient-domain light transport simulation, a recent family of techniques in physically based rendering introduced in the past five years that can accelerate traditional Monte Carlo rendering up to approximately an order of magnitude based on gradient estimation and image reconstruction. Particularly, we will introduce the fundamentals of gradient-domain rendering with gradient-domain path tracing, and then extend the discussion to gradient-domain bidirectional path tracing and photon density estimation. We also discuss volume rendering in the gradient domain before diving into advanced topics in recent state-of-the-art papers in this direction. We further discuss tips and tricks in open-source implementations of such algorithms, and provide ideas for future research directions.

ACM ISBN 978-1-4503-6026-5/18/12. https://doi.org/10.1145/3277644.3277761

Contents

1 INTRODUCTION

Accurately simulating light transport remains a long standing challenge in computer graphics. While the high-dimensional rendering integral can be well approximated by Monte Carlo estimation, existing techniques still take from minutes to hours to output images that are clean to human perception. Making fast rendering techniques even faster is therefore still an active research topic.

Several recent extensions of conventional rendering methods to the gradient domain have greatly improved visual convergence [Kettunen et al. 2015; Lehtinen et al. 2013; Manzi et al. 2015] by exploiting image-space smoothness and path coherency. The seminal work on gradient-domain Metropolis light transport [Lehtinen et al. 2013] led to gradient-domain variants of uni- and bi-directional path tracing [Kettunen et al. 2015; Manzi et al. 2015], as well as gradient-domain density estimation on surfaces [Hua et al. 2017; Sun et al. 2017]. These methods directly estimate image *gradients*, instead of pixel intensities, and apply Poisson reconstruction to generate the final image. Gradient-domain approaches tend to be more efficient than their conventional counterparts since images tend to be piecewise smooth in many scenes.

This course is an introductory resource about theoretical understanding of gradient-domain rendering and practical issues and solutions of how to bring traditional rendering algorithms to the gradient domain. It also serves as a comprehensive technical report about the state-of-the-art gradient-domain rendering techniques. The course is designed for researchers, developers, students, and enthusiasts in computer graphics who are intrigued about how to make existing physically based light transport simulation more efficient.

Audience with basic background about Monte Carlo rendering techniques such as path tracing or photon mapping is required. Upon the completion of this course, the audience can apply the knowledge to implement typical gradient-domain light transport algorithms such as gradient-domain path tracing or gradient-domain photon density estimation in their own renderers.

We will start with a brief revision to basic theories of light transport and traditional rendering techniques such as path tracing [Kajiya 1986] and photon mapping [Hachisuka and Jensen 2009; Jensen 2001] at the beginning of the course before diving into basic concepts of gradient-domain rendering. We will particularly explore gradient-domain path tracing [Kettunen et al. 2015], and extend the discussion towards bidirectional techniques that includes gradient-domain bidirectional path tracing [Manzi et al. 2015] and gradient-domain photon density estimation [Hua et al. 2017] and their combinations [Sun et al. 2017] that uses more sophisticated path exploration techniques [Jakob and Marschner 2012]. Continuing the success of gradient-domain rendering for surfaces, we then explore rendering scenes with participating media in the gradient domain [Gruson et al. 2018], which also often exhibit image-space smoothness despite of an additional dimension in the rendering integral. Advanced but orthogonal aspects of gradient-domain rendering follow, including more robust image reconstruction [Manzi et al. 2016b; Rousselle et al. 2016], reusing paths [Bauszat et al. 2017], temporal rendering [Manzi et al. 2016a], adaptive sampling [Back et al. 2018; Lehtinen et al. 2013; Manzi et al. 2014] and spectral rendering [Petitjean et al. 2018]. Finally, we will discuss practical implementation details and potential ideas for future work.

2 SYLLABUS

1. Opening

(5 minutes) Welcome speech and a brief introduction about the course, its objectives and schedule.

2. Basic theories in light transport

(15 minutes) We give a brief overview about the state-of-the-art Monte Carlo light transport simulation techniques and highlighted path tracing and photon mapping. This is a warming up section for those who are not very familiar with rendering before we dive into the technical details of gradient-domain rendering.

3. Fundamentals of gradient-domain light transport

(30 minutes) The key ideas of gradient-domain rendering is introduced. Particularly, the audience will learn the concepts of the modified rendering equation for gradient-domain rendering, the shift mapping function and its Jacobian. Gradient-domain path tracing will be introduced along with the simple half-vector shift mapping. The classical reconstruction technique using screened Poisson reconstruction will also be introduced.

Break (10 minutes)

4. Gradient-domain bidirectional light transport

(30 minutes) We continue with the discussion of bidirectional path tracing, progressive photon mapping, and their combinations in the gradient domain. In this part, more complex shift mapping techniques will be introduced such as manifold exploration.

5. Gradient-domain volumetric rendering

(30 minutes) Beyond surfaces, we discuss volumetric rendering with the gradient-domain volumetric photon density estimation techniques. More complex concepts in volumetric rendering such as photon points, beams, and planes are introduced in this section.

Break (10 minutes)

6. Advanced topics in gradient-domain rendering

(30 minutes) We discuss a collection of many advanced techniques in gradient-domain rendering such as temporal rendering, adaptive rendering, Metropolis light transport, path reuse, spectral rendering, and improved reconstruction.

7. Practical tips and tricks of implementing gradient-domain rendering

(15 minutes) We discuss practical issues and solutions for implementing gradient-domain rendering in Mitsuba, an open-source physically based renderer. We also share our experience in implementing a hobby-time gradient-domain renderer from scratch in Rust, a system programming language designed to prevent segmentation faults.

8. Conclusion and Q&A

(15 minutes) We will summarize the key concepts we learn so far, and brainstorm ideas for future work.

Estimated total time: 2 hours 50 minutes of presentations and two 10-minute breaks.

3 PRESENTERS

3.1 Binh-Son Hua, The University of Tokyo

Binh-Son Hua is currently a post-doctoral researcher at the University of Tokyo. Before that, he was a postdoctoral researcher in Singapore University of Technology and Design. He received his PhD degree in Computer Science from National University of Singapore in 2015. His research interests are physically based rendering and 3D scene understanding. His recent works are published in both computer graphics and computer vision venues, including Eurographics, TVCG, 3DV, CVPR, and SIGGRAPH.

3.2 Adrien Gruson, The University of Tokyo and JFLI CNRS UMI 3527

Adrien Gruson is a post-doctoral researcher in Computer Graphics Group at the University of Tokyo (Prof. Toshiya Hachisuka's lab). Before that, he received his Ph.D. in Computer Science from the University of Rennes 1. His research interests include physically-based rendering based on MCMC and gradient-domain approaches. He is also a member of Japanese French Laboratory of Informatics (JFLI).

3.3 Matthias Zwicker, University of Maryland

Matthias Zwicker has joined the University of Maryland in March 2017 as the Reginald Allan Hahne Endowed E-Nnovate Professor in Computer Science. He obtained his PhD from ETH in Zurich, Switzerland, in 2003. From 2003 to 2006 he was a post-doctoral associate with the computer graphics group at the Massachusetts Institute of Technology, and then held a position as an Assistant Professor at the University of California in San Diego from 2006 to 2008. From 2008-2017, he was a professor in Computer Science at the University of Bern, Switzerland, and the head of the Computer Graphics Group. His research focus is on efficient high-quality rendering, signal processing techniques for computer graphics, data-driven modeling and animation, and point-based methods.

3.4 Toshiya Hachisuka, The University of Tokyo

Toshiya Hachisuka is an Associate Professor in the Department of Creative Informatics at the University of Tokyo. He is interested in the intersection of computational statistics, numerical computation, and physics based computer graphics. He has published multiple work on those topics including some recent work on gradient-domain rendering via photon density estimation. He received his Ph.D. in Computer Science from University of California at San Diego in 2011 and B.Eng. from the University of Tokyo in 2006.

ACKNOWLEDGMENTS

The Door scene in the teaser is modelled after the famous one found in Eric Veach and Leo Guibas' original Metropolis Light Transport paper by Miika Aittala, Samuli Laine, and Jaakko Lehtinen.

This project was partly funded by JSPS KAKENHI grant numbers 15H05308 and 17K19958, and Swiss National Science Foundation grant number 163045.

REFERENCES

Jonghee Back, Sung-Eui Yoon, and Bochang Moon. 2018. Feature Generation for Adaptive Gradient-Domain Path Tracing. In *Computer Graphics Forum (Proceedings of Pacific Graphics).*

Pablo Bauszat, Victor Petitjean, and Elmar Eisemann. 2017. Gradient-Domain Path Reusing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* (2017).

Adrien Gruson, Binh-Son Hua, Nicolas Vibert, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2018. Gradient-domain Volumetric Photon Density Estimation. *ACM Transactions on Graphics* (2018).

Toshiya Hachisuka and Henrik Wann Jensen. 2009. Stochastic progressive photon mapping. *ACM Transactions on Graphics (TOG)* 28, 5 (2009).

Binh-Son Hua, Adrien Gruson, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2017. Gradient-Domain Photon Density Estimation. *Computer Graphics Forum* 36, 2 (2017).

Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (TOG)* 31, 4 (2012).

Henrik Wann Jensen. 2001. *Realistic image synthesis using photon mapping.*

James T. Kajiya. 1986. The Rendering Equation. *ACM Transactions on Graphics (TOG)* 20, 4 (1986).

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain path tracing. *ACM Transactions on Graphics (TOG)* 34, 4 (2015).

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain metropolis light transport. *ACM Transactions on Graphics (TOG)* 32, 4 (2013).

Marco Manzi, Markus Kettunen, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain bidirectional path tracing. *Eurographics Symposium on Rendering* (2015).

Marco Manzi, Markus Kettunen, Frédo Durand, Matthias Zwicker, and Jaakko Lehtinen. 2016a. Temporal gradient-domain path tracing. *ACM Transactions on Graphics (TOG)* (2016).

Marco Manzi, Fabrice Rousselle, Markus Kettunen, Jaakko Lehtinen, and Matthias Zwicker. 2014. Improved Sampling for Gradient-domain Metropolis Light Transport. *ACM Trans. Graph.* (2014).

Marco Manzi, Delio Vicini, and Matthias Zwicker. 2016b. Regularizing Image Reconstruction for Gradient-Domain Rendering with Feature Patches. *Computer Graphics Forum* 35, 2 (2016).

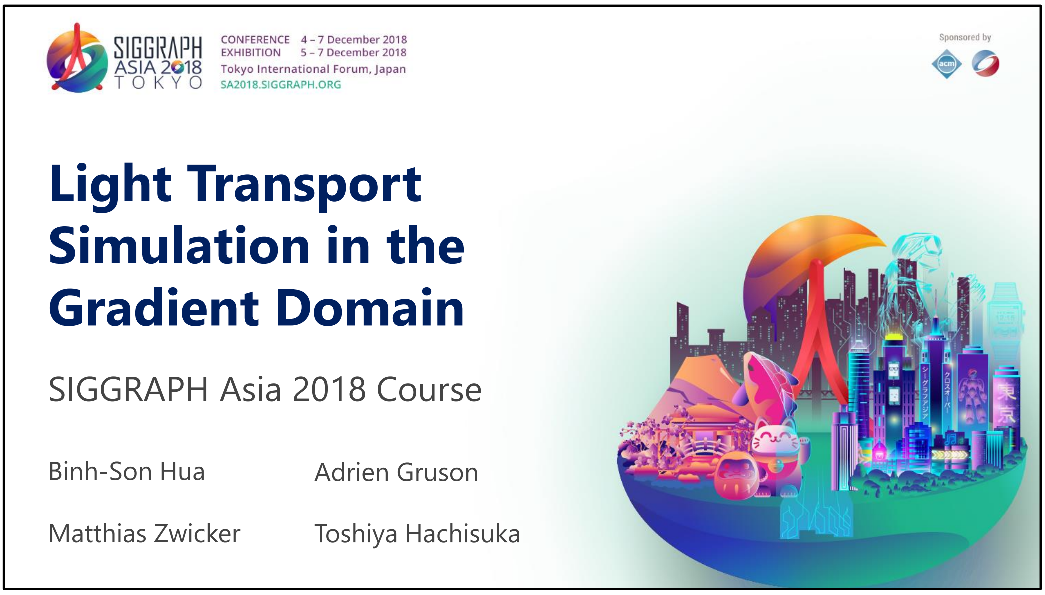
Victor Petitjean, Pablo Bauszat, and Elmar Eisemann. 2018. Spectral Gradient Sampling for Path Tracing. In *Computer Graphics Forum (Proceedings of EGSR).*

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-space Control Variates for Rendering. *ACM Transactions on Graphics (TOG)* 35, 6 (2016).

Weilun Sun, Xin Sun, Nathan A Carr, Derek Nowrouzezahrai, and Ravi Ramamoorthi. 2017. Gradient-Domain Vertex Connection and Merging. (2017).

4 COURSE NOTES

4.1 Opening



SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

Sponsored by

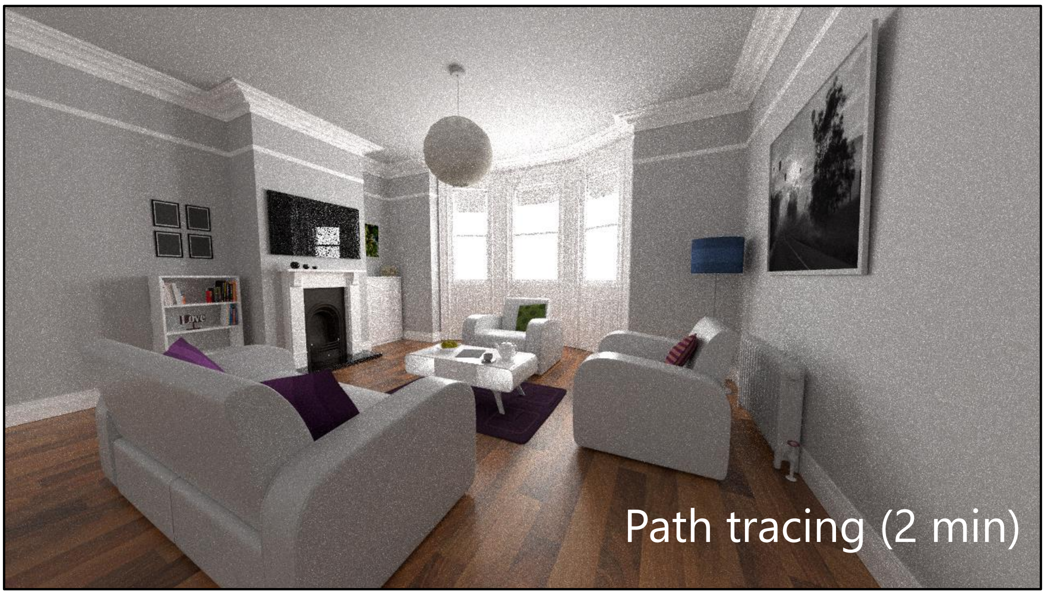
Light Transport Simulation in the Gradient Domain

SIGGRAPH Asia 2018 Course

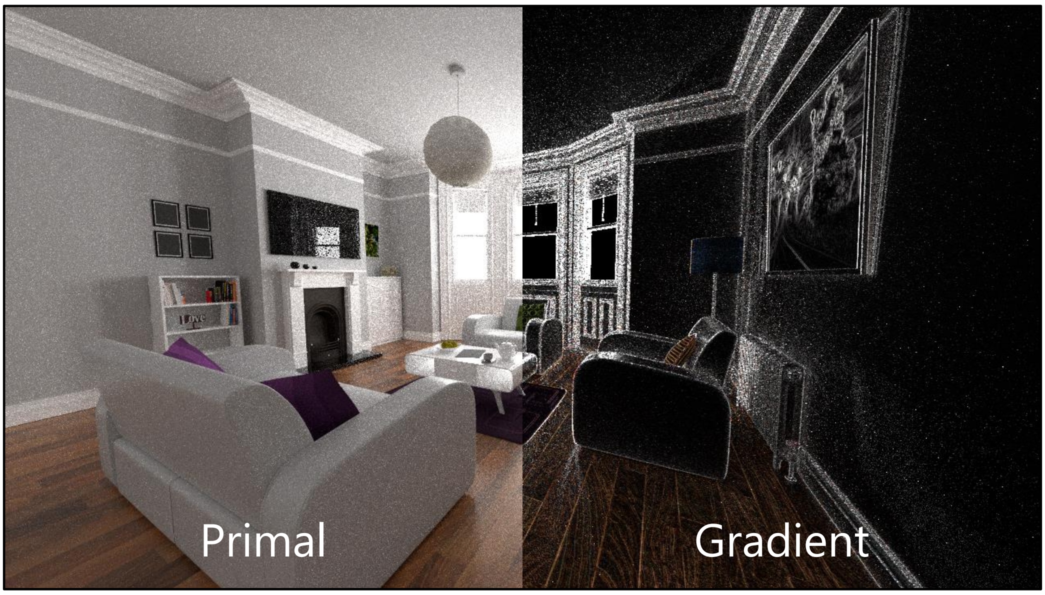
Binh-Son Hua
Adrien Gruson
Matthias Zwicker
Toshiya Hachisuka

Reference (1 hour)

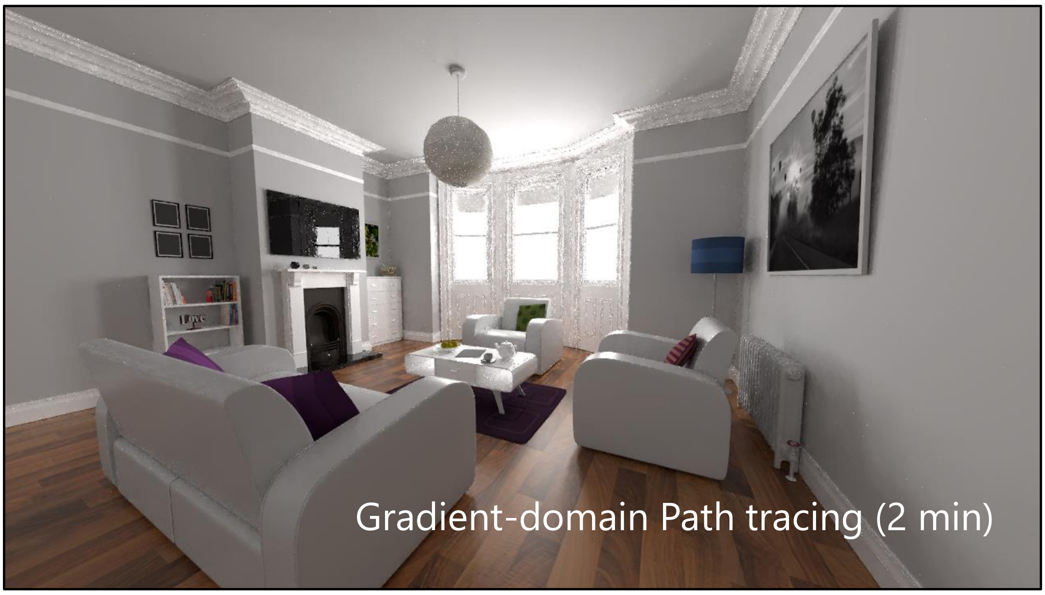
Light transport simulation in the modern days boils down to solving a rendering integral using Monte Carlo estimation. Many different algorithms to estimate image pixel values can be used, from path tracing, bidirectional path tracing, to photon mapping, virtual point lights, and many others. The common point of such algorithms (when they are consistent) is that they will converge to the same image after spending long enough rendering time. For example, we render this image of a living room in 1 hour.



1 hour is rather long to produce a clean image. If we only spend two minutes, we usually get images with visible noise given such a complex scene. In production of feature films, architectural visualization, or animation, making current fast rendering faster has been the goal of the computer graphics community in the past decades.



In this course, we will learn about a family of rendering techniques that can further accelerate existing rendering algorithms. We build upon the concept of estimating image-space gradients alongside with rendering the original (primal) image. We will see that estimating gradients turns out to be a nice and consistent way to make noisy images less noisy for many common cases with the same computation effort.



Here is an example. With the same two minutes, gradient-domain rendering can deliver cleaner results compared to existing path tracing.

GRADIENT-DOMAIN RENDERING

Gradient-Domain Photon Density Estimation
Gradient-Domain Path Reusing
Gradient-domain Volumetric Photon Density Estimation
Gradient-Domain Metropolis Light Transport
Gradient-Domain Path Tracing

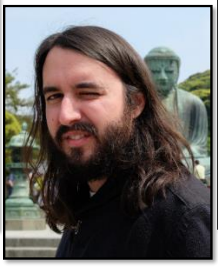
>13 papers on this subject

In recent years, there have been several top-tier papers published to push the direction of gradient-domain rendering forward. This course builds on the materials of such papers, from introductory to advanced topics in gradient-domain rendering.

PRESENTERS

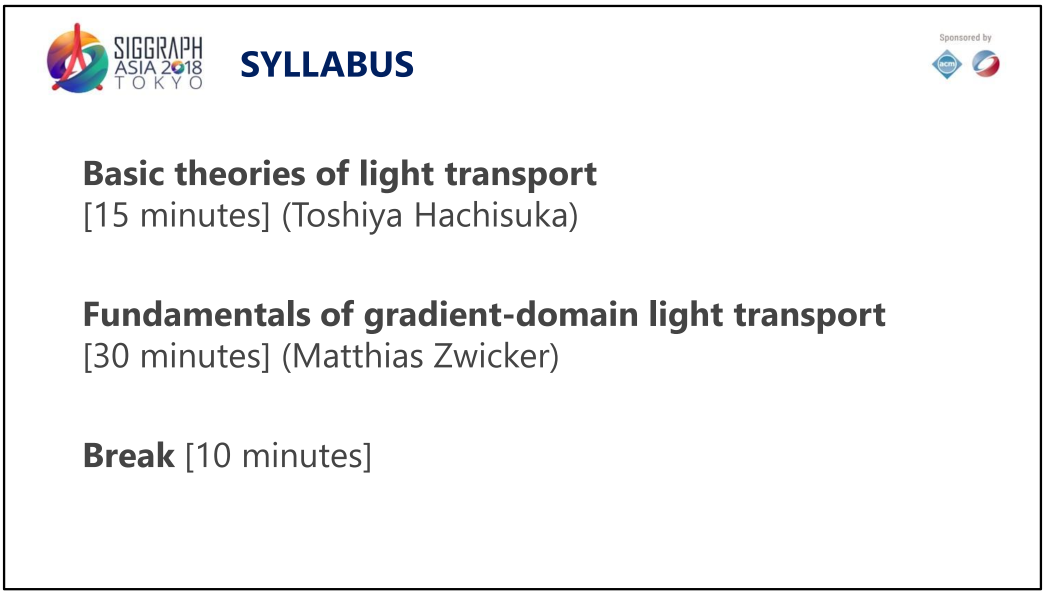
Binh-Son Hua
(The University of Tokyo)

Matthias Zwicker
(University of Maryland)



Adrien Gruson
(The University of Tokyo, JFLI)

Toshiya Hachisuka
(The University of Tokyo)



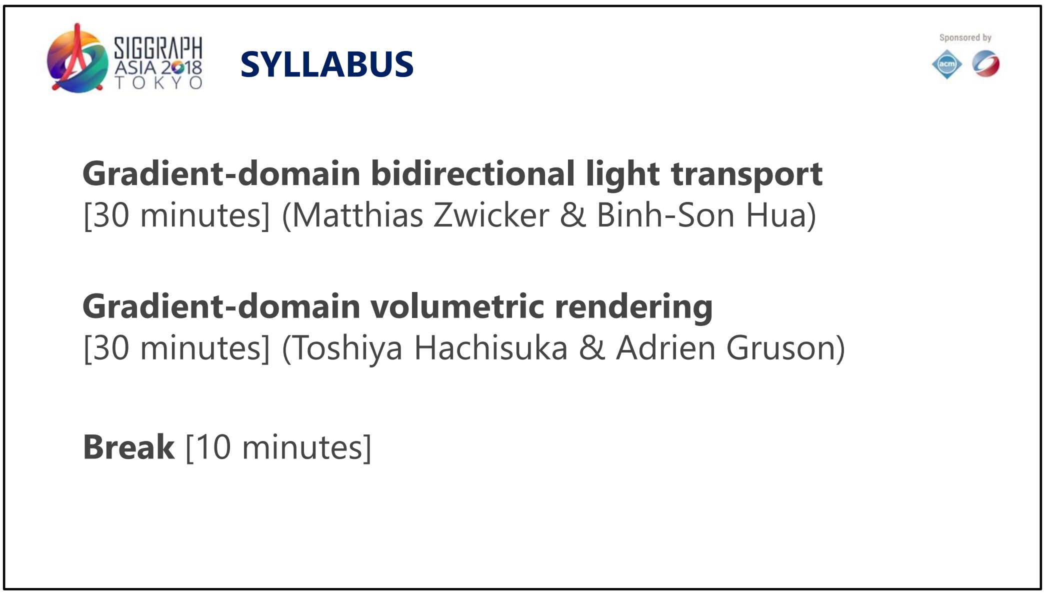
SYLLABUS

Basic theories of light transport
[15 minutes] (Toshiya Hachisuka)

Fundamentals of gradient-domain light transport
[30 minutes] (Matthias Zwicker)

Break [10 minutes]

Here is the syllabus.

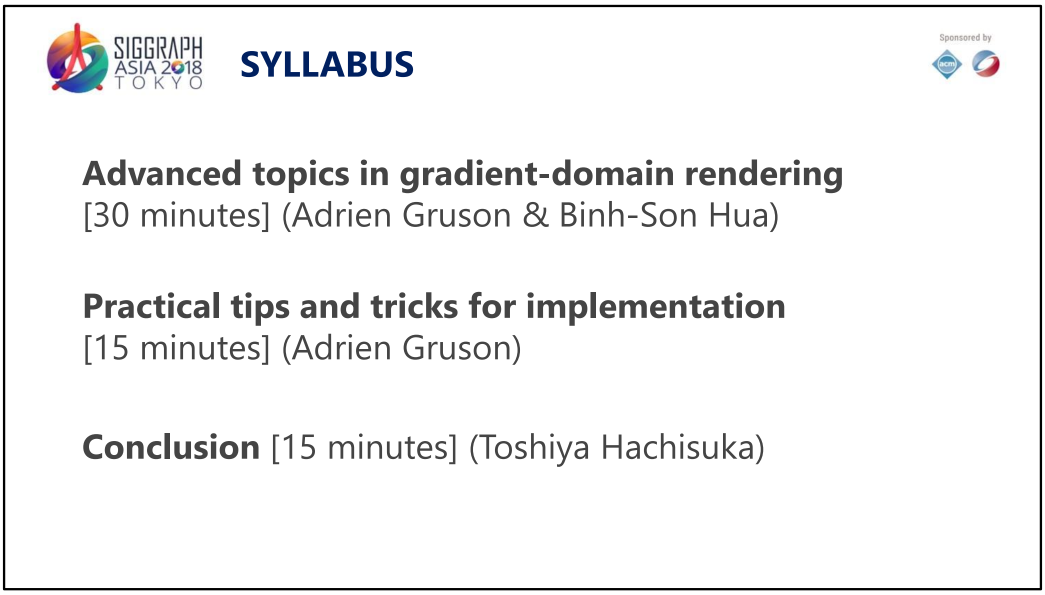


SYLLABUS

Gradient-domain bidirectional light transport
[30 minutes] (Matthias Zwicker & Binh-Son Hua)

Gradient-domain volumetric rendering
[30 minutes] (Toshiya Hachisuka & Adrien Gruson)

Break [10 minutes]

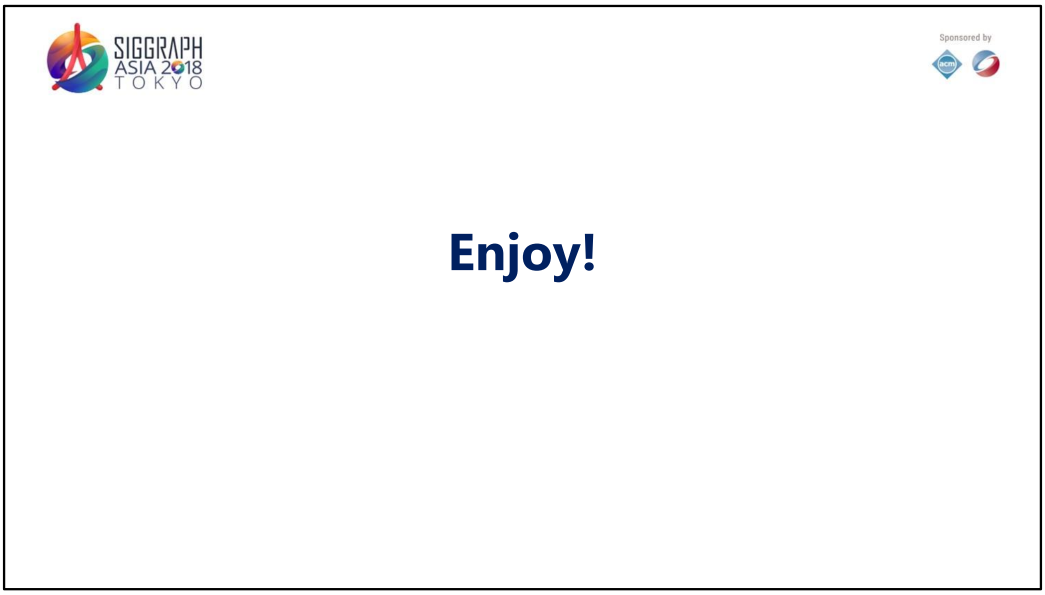


SYLLABUS

Advanced topics in gradient-domain rendering
[30 minutes] (Adrien Gruson & Binh-Son Hua)

Practical tips and tricks for implementation
[15 minutes] (Adrien Gruson)

Conclusion [15 minutes] (Toshiya Hachisuka)



SIGGRAPH ASIA 2018 TOKYO

Sponsored by

Enjoy!

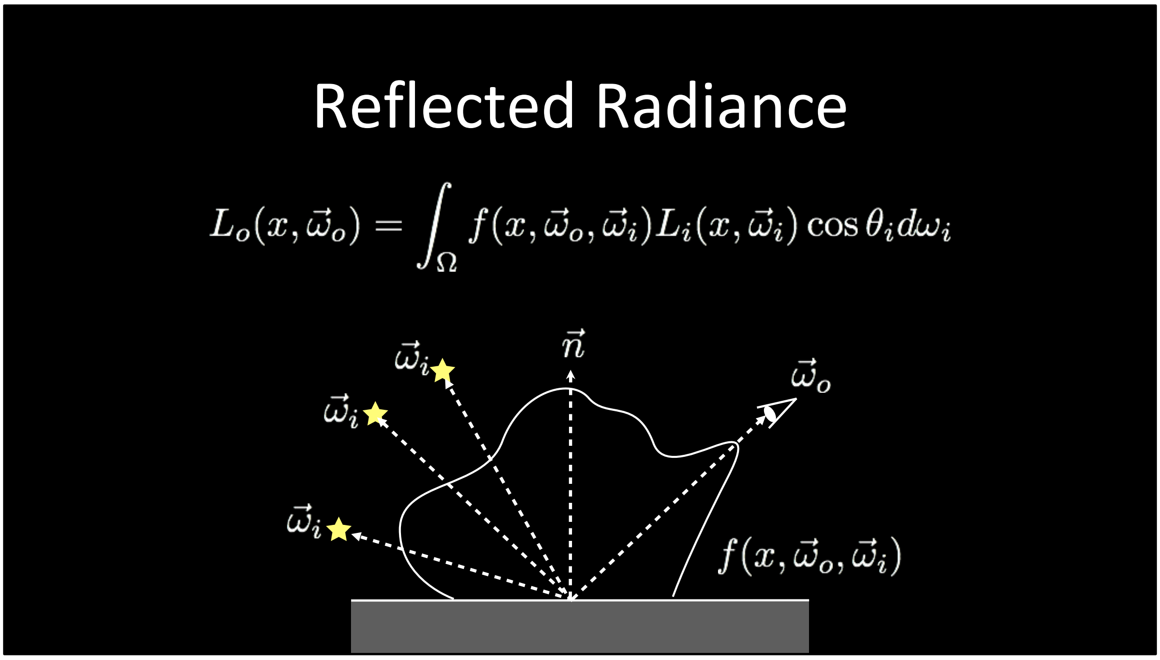
4.2 Basic theories in light transport

SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

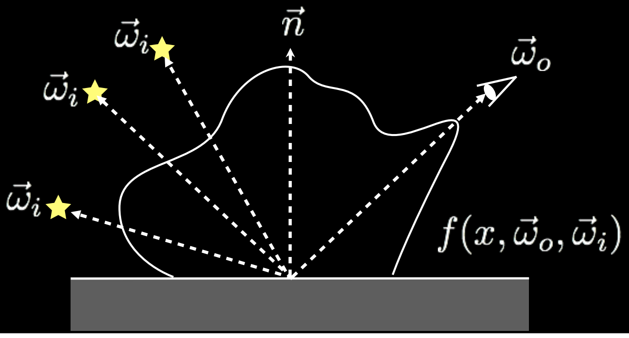
Sponsored by

Basic Theories of Light Transport Simulation

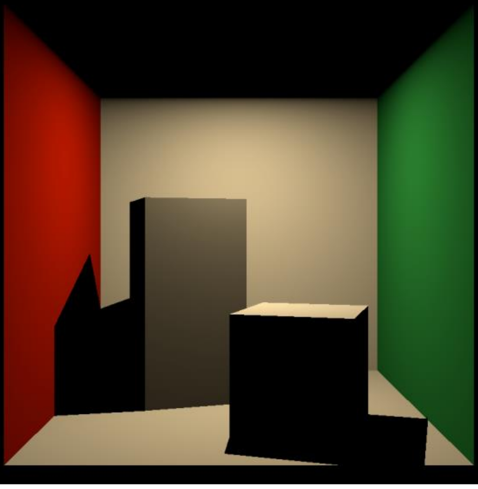


Reflected Radiance

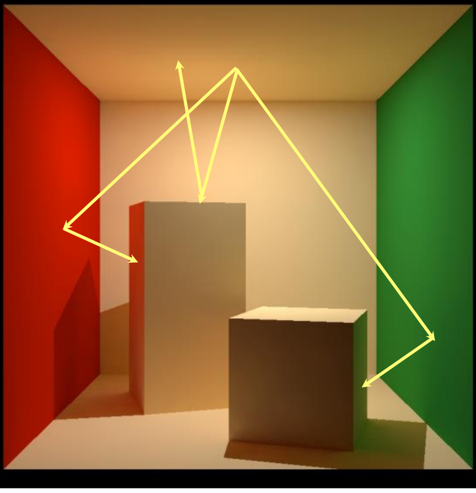
$$L_o(x,\vec{\omega}_o) = \int_\Omega f(x,\vec{\omega}_o,\vec{\omega}_i) L_i(x,\vec{\omega}_i)\cos\theta_i d\omega_i$$



Direct Illumination



Direct + Indirect Illumination

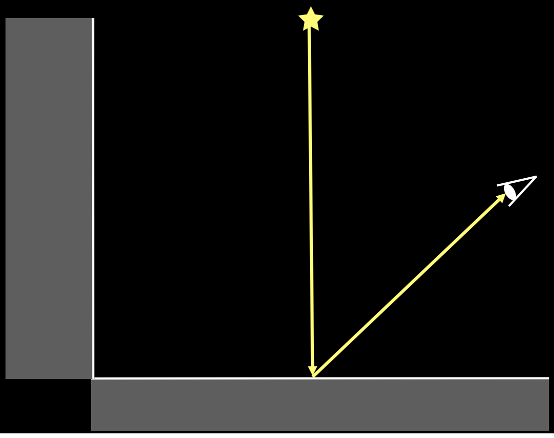


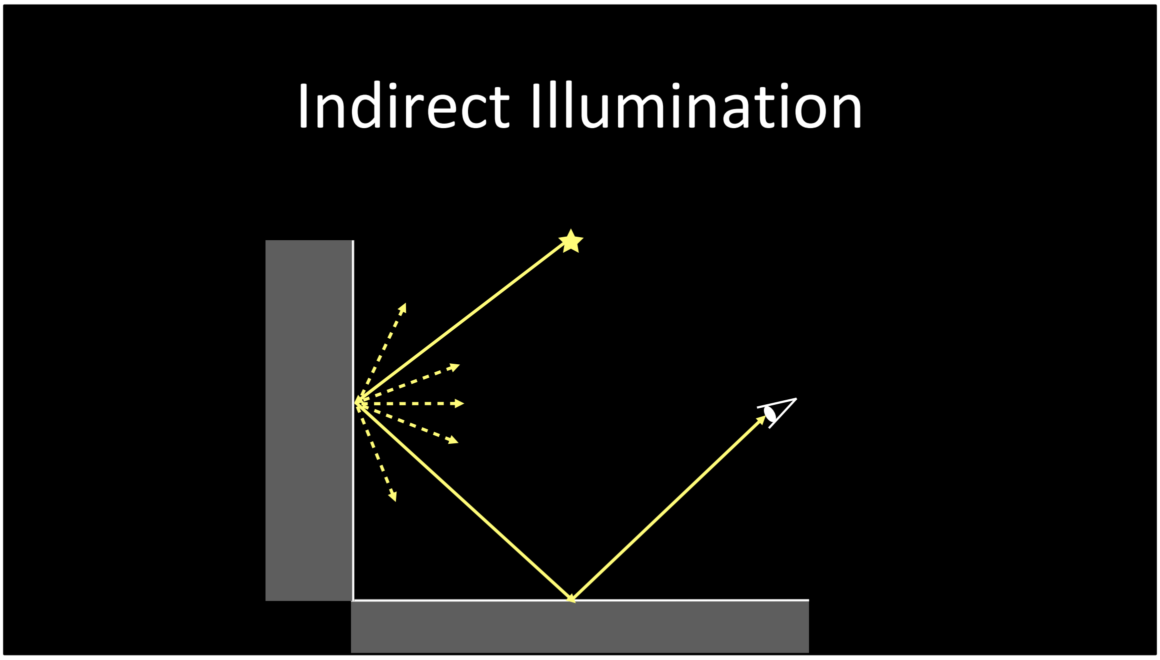
Indirect lighting

- Light can bounce off from other surfaces
- Multiple bounces

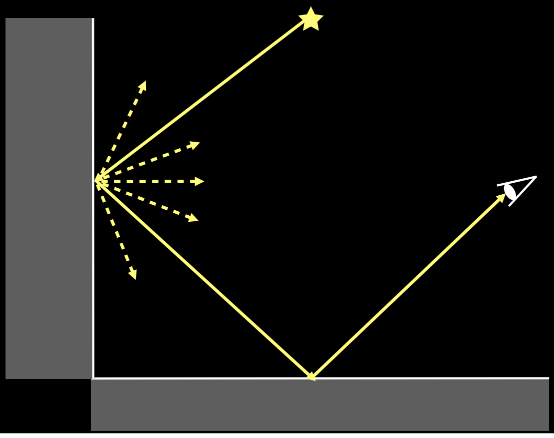
- Global illumination
  - Other objects affect illumination
  - Shadowing is one example
  - In contrast to local illumination

Indirect Illumination

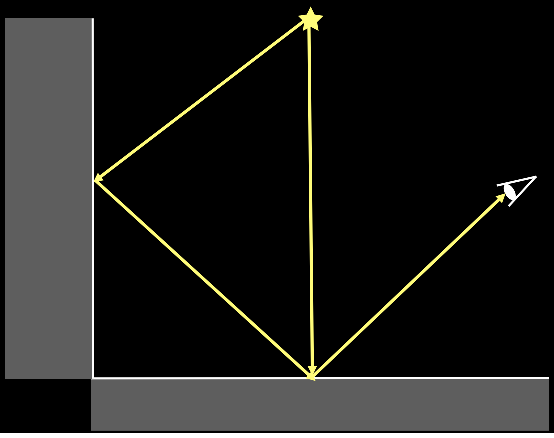




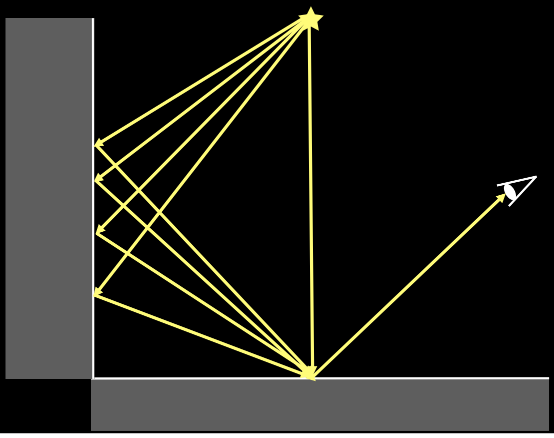
Indirect Illumination



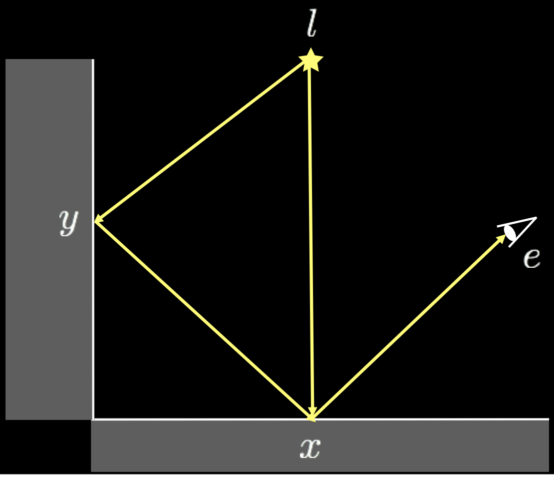
Indirect Illumination



Indirect Illumination



Indirect Illumination



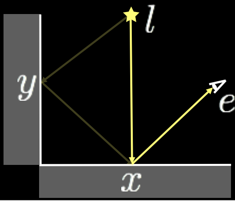
Including Indirect Bounce

$$I^0 L_o(x \to e) = \int_\Omega f(l \to x \to e) L_i(l \to x) \cos\theta d\omega$$

$$L(x \to e)$$

$$f(l \to x \to e)$$

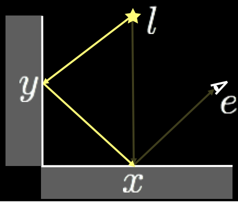
$$I^0$$

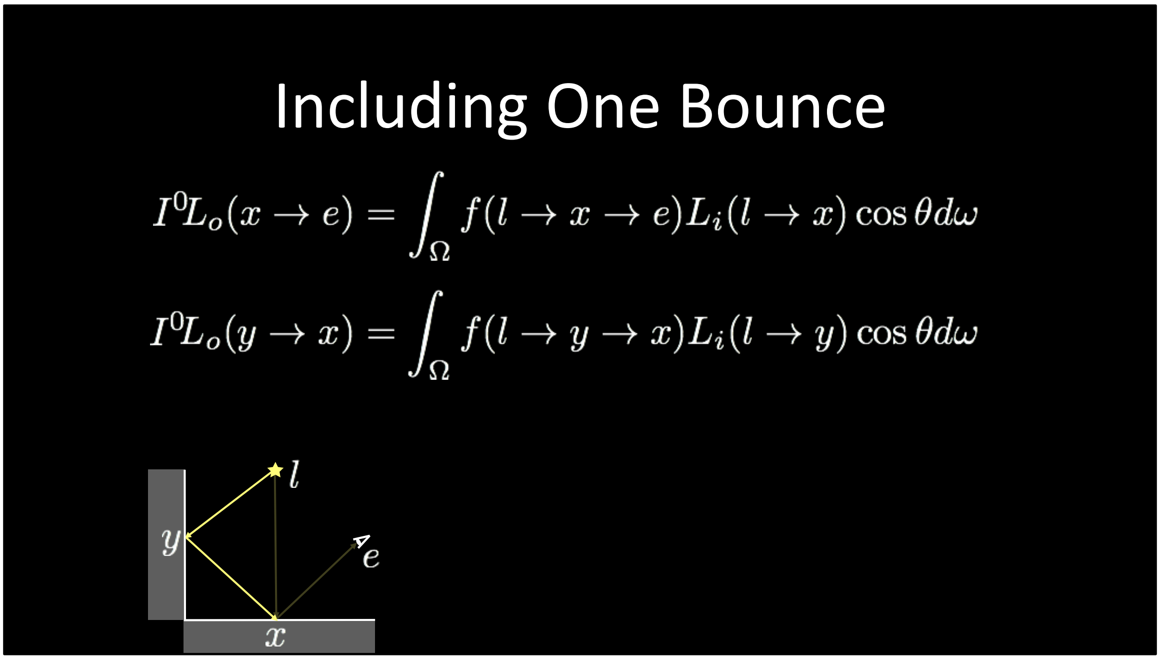


Including Indirect Bounce

$$I^0 L_o(x \to e) = \int_\Omega f(l \to x \to e) L_i(l \to x) \cos\theta d\omega$$

$$I^0 L_o(y \to x) =$$

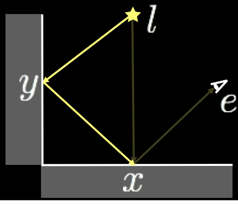




Including One Bounce

$$I^0 L_o(x \to e) = \int_\Omega f(l \to x \to e) L_i(l \to x) \cos\theta d\omega$$

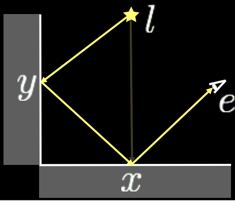
$$I^0 L_o(y \to x) = \int_\Omega f(l \to y \to x) L_i(l \to y) \cos\theta d\omega$$

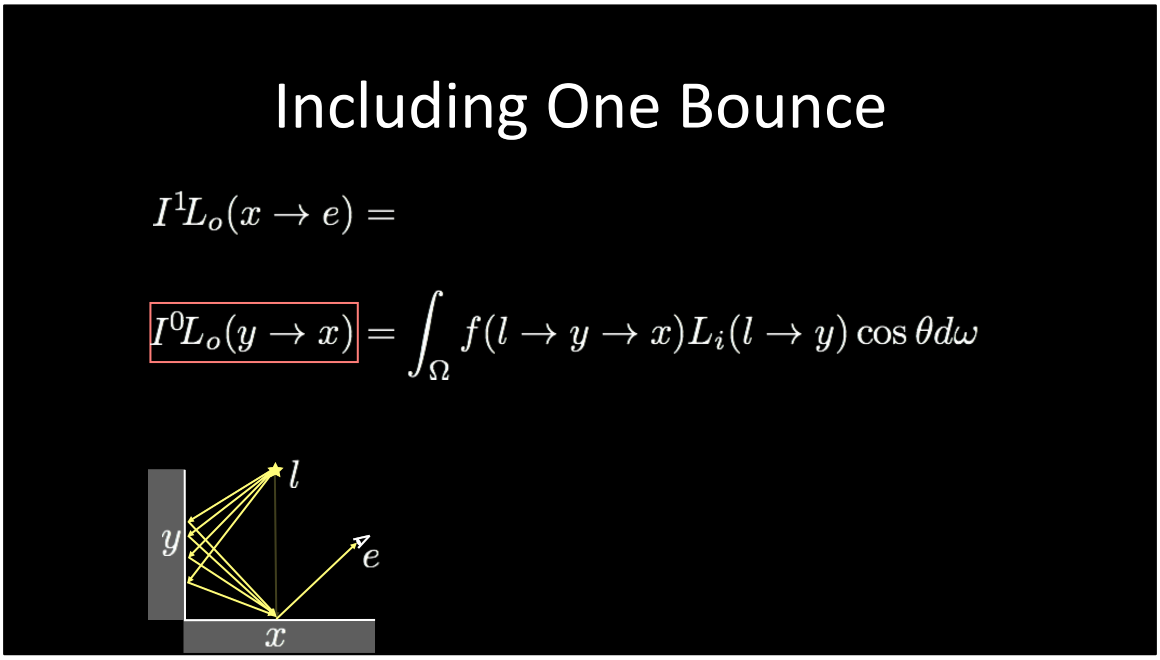


Including One Bounce

$$I^1 L_o(x \to e) =$$

$$I^0 L_o(y \to x) = \int_\Omega f(l \to y \to x) L_i(l \to y) \cos\theta d\omega$$

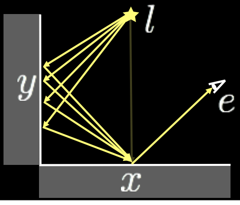




Including One Bounce

$$I^1 L_o(x \to e) =$$

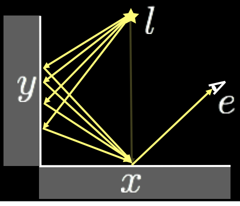
$$\boxed{I^0 L_o(y \to x)} = \int_\Omega f(l \to y \to x) L_i(l \to y) \cos\theta d\omega$$



Including One Bounce

$$I^1 L_o(x \to e) = \int_\Omega f(\boxed{y \to x \to e})\boxed{I^0 L_o(y \to x)} \cos\theta d\omega$$

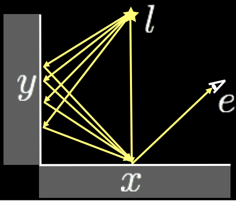
$$\boxed{I^0 L_o(y \to x)} = \int_\Omega f(l \to y \to x) L_i(l \to y) \cos\theta d\omega$$



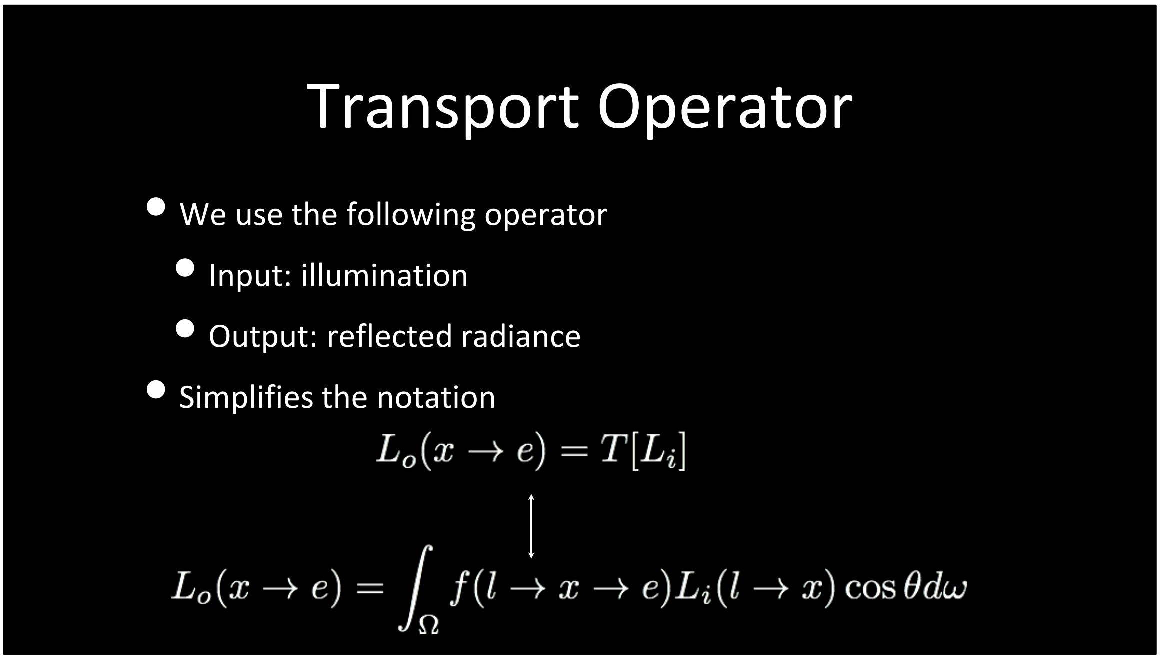
Including One Bounce

$$I^1 L_o(x \to e) = \int_\Omega f(y \to x \to e)\, I^0 L_o(y \to x) \cos\theta d\omega$$

$$I^0 L_o(y \to x) = \int_\Omega f(l \to y \to x) L_i(l \to y) \cos\theta d\omega$$



$$\boxed{\begin{aligned} L_o(x \to e) = I^0 L_o(x \to e) \\ + I^1 L_o(x \to e) \end{aligned}}$$



Transport Operator

- We use the following operator
  - Input: illumination
  - Output: reflected radiance
- Simplifies the notation

$$L_o(x \to e) = T[L_i]$$

$$\Big\updownarrow$$

$$L_o(x \to e) = \int_\Omega f(l \to x \to e) L_i(l \to x) \cos\theta d\omega$$

Using Transport Operator

- One bounce (i.e., direct)

$$I^0 L_o(x \to e) = T[L_i]$$

- Two bounces

$$I^1 L_o(x \to e) = T[I^0 L_o]$$

Using Transport Operator

- One bounce (i.e., direct)

$$\boxed{I^0 L_o(x \to e)} = T[L_i]$$

- Two bounces

$$I^1 L_o(x \to e) = T[\boxed{I^0 L_o}]$$

Using Transport Operator

- One bounce (i.e., direct)

$$I^0 L_o(x \to e) = T[L_i]$$

- Two bounces

$$\begin{aligned} I^1 L_o(x \to e) &= T[I^0 L_o] \\ &= T[T[L_i]] = T^2[L_i] \end{aligned}$$

Using Transport Operator

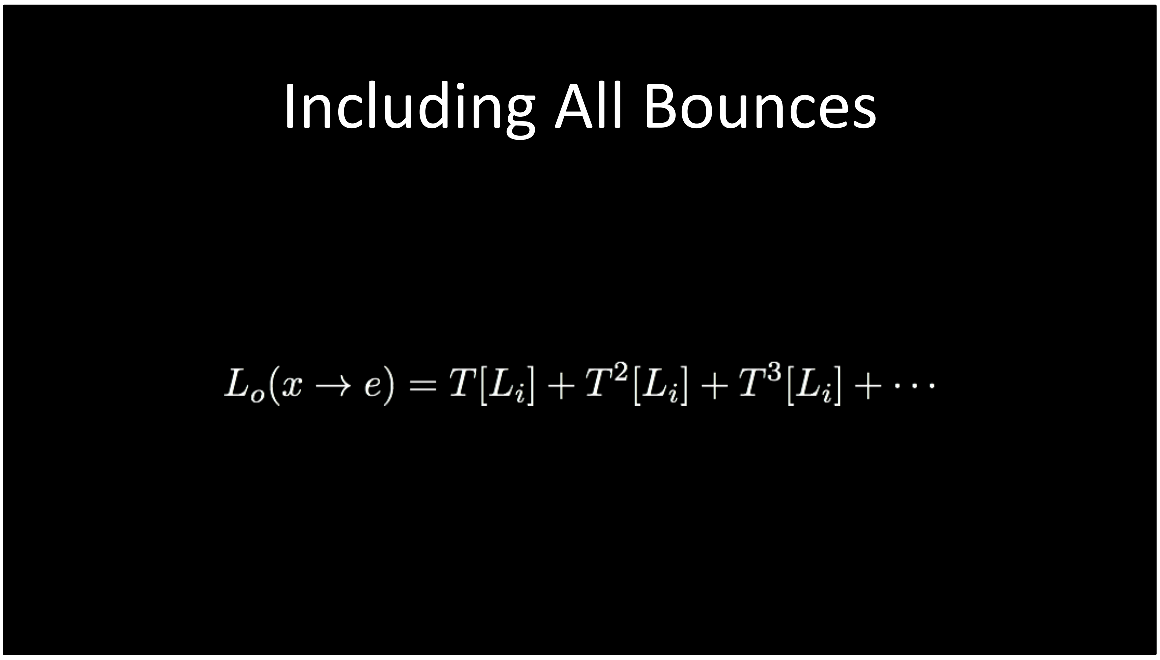
- One bounce (i.e., direct)

$$I^0 L_o(x \to e) = T[L_i]$$

- Two bounces

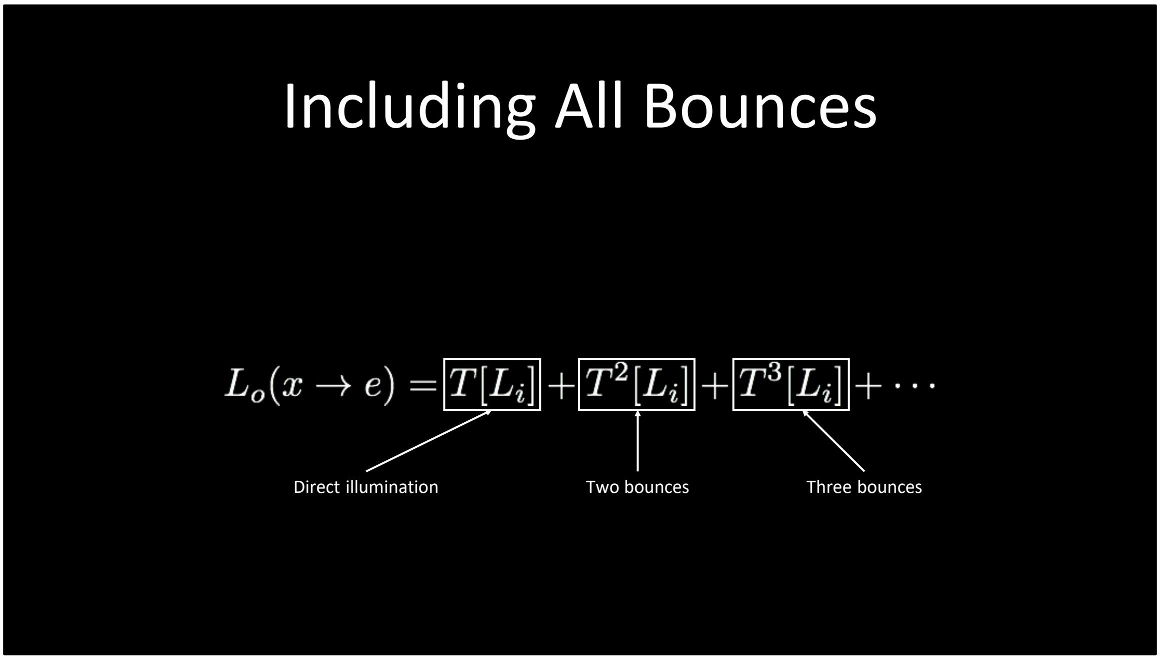
$$\begin{aligned} I^1 L_o(x \to e) &= T[I^0 L_o] \\ &= T[T[L_i]] = T^2[L_i] \end{aligned}$$

$$\boxed{L_o(x \to e) = T[L_i] + T^2[L_i]}$$



Including All Bounces

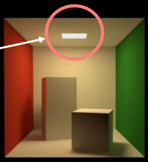
$$L_o(x \to e) = T[L_i] + T^2[L_i] + T^3[L_i] + \cdots$$



Including All Bounces

$$L_o(x \to e) = \underbrace{\boxed{T[L_i]}}_{\text{Direct illumination}} + \underbrace{\boxed{T^2[L_i]}}_{\text{Two bounces}} + \underbrace{\boxed{T^3[L_i]}}_{\text{Three bounces}} + \cdots$$

Including All Bounces



Directly visible light sources

$$L_o(x \to e) = \boxed{L_i} + \boxed{T[L_i]} + \boxed{T^2[L_i]} + \boxed{T^3[L_i]} + \cdots$$

Direct illumination

Two bounces

Three bounces

To the Rendering Equation

- Remember the Neumann series

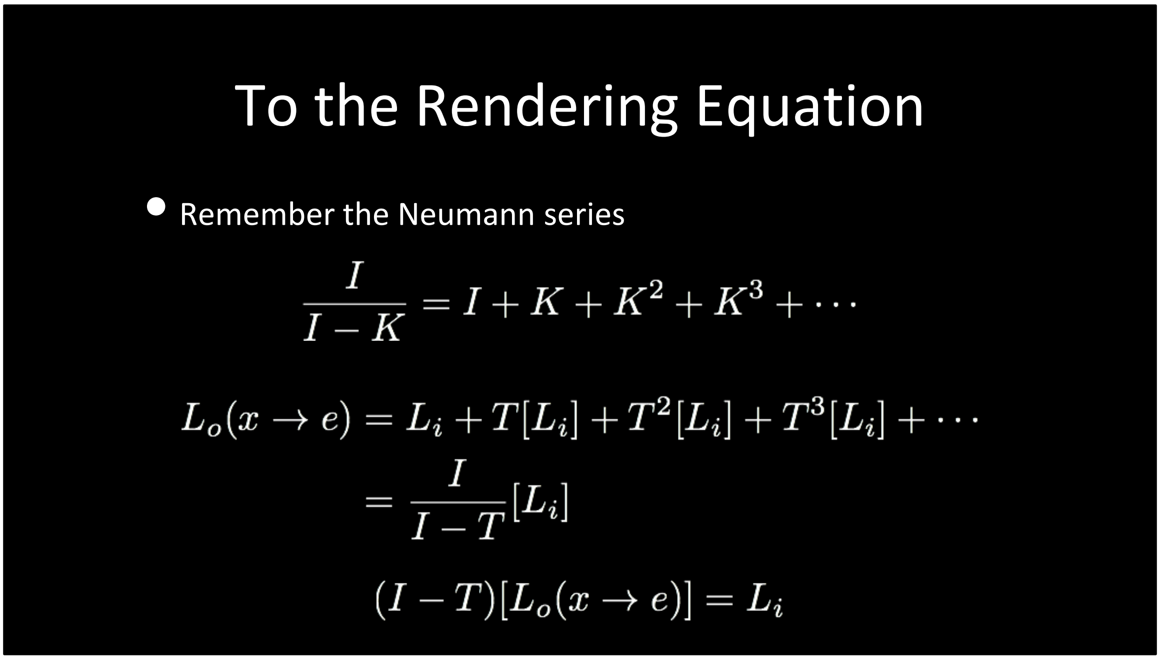
$$\frac{I}{I-K} = I + K + K^2 + K^3 + \cdots$$

To the Rendering Equation

- Remember the Neumann series

$$\frac{I}{I-K} = I + K + K^2 + K^3 + \cdots$$

$$\begin{aligned} L_o(x \to e) &= L_i + T[L_i] + T^2[L_i] + T^3[L_i] + \cdots \\ &= \frac{I}{I-T}[L_i] \end{aligned}$$



To the Rendering Equation

- Remember the Neumann series

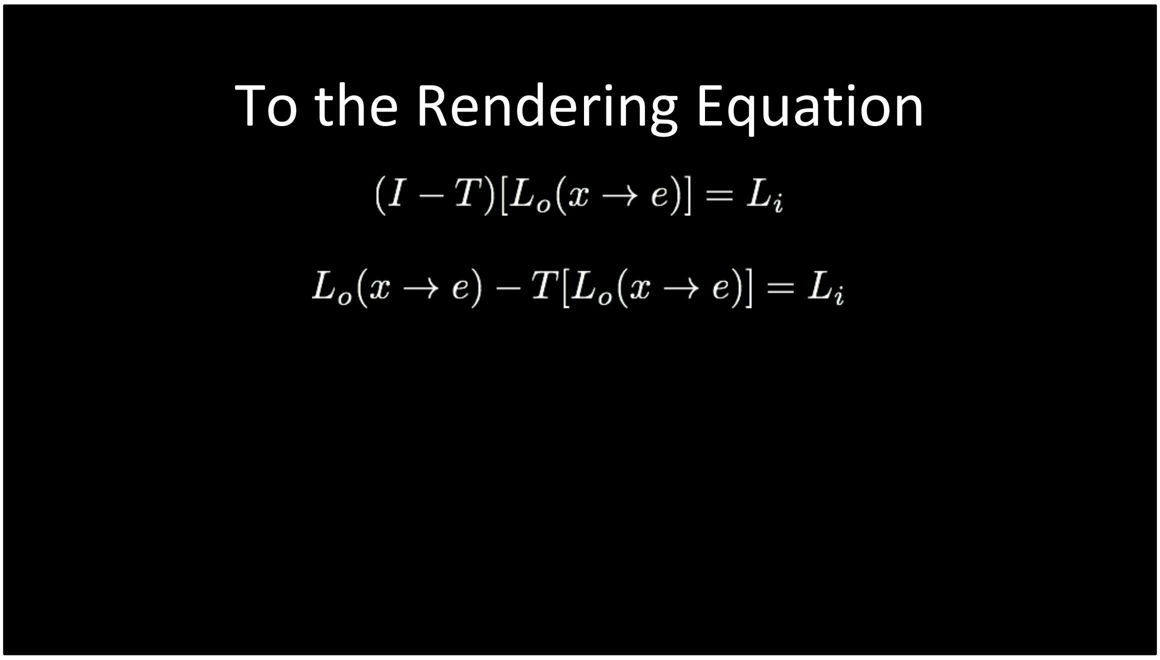
$$\frac{I}{I-K} = I + K + K^2 + K^3 + \cdots$$

$$\begin{aligned} L_o(x \to e) &= L_i + T[L_i] + T^2[L_i] + T^3[L_i] + \cdots \\ &= \frac{I}{I-T}[L_i] \end{aligned}$$

$$(I-T)[L_o(x \to e)] = L_i$$

To the Rendering Equation

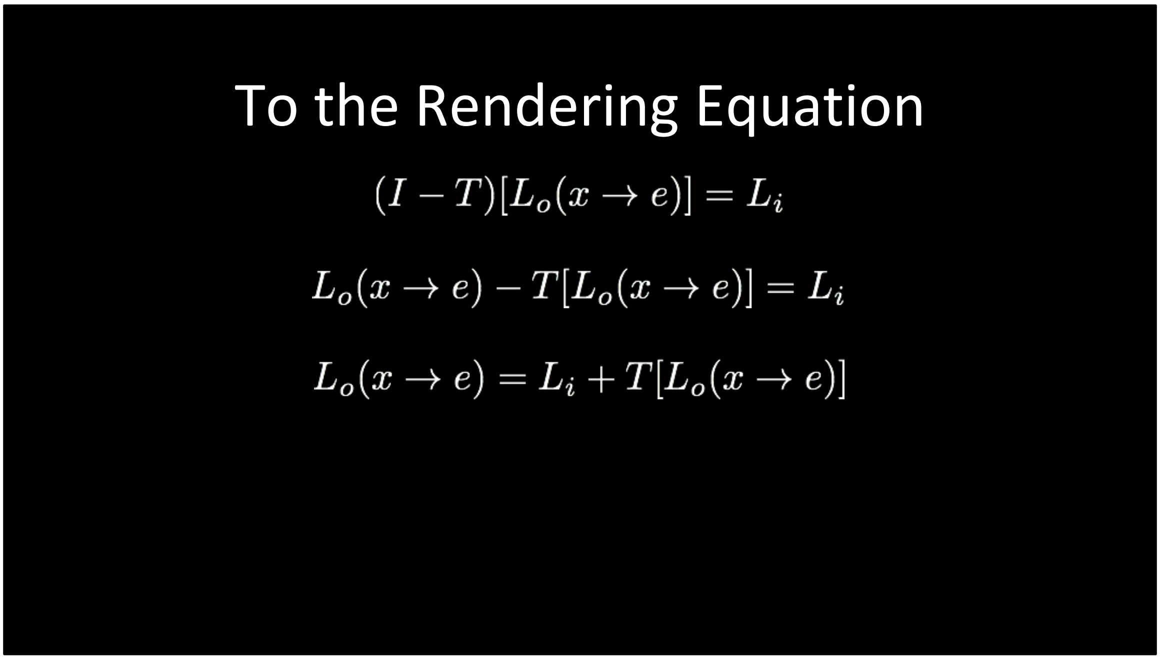
$$(I - T)[L_o(x \to e)] = L_i$$



To the Rendering Equation

$$(I - T)[L_o(x \to e)] = L_i$$

$$L_o(x \to e) - T[L_o(x \to e)] = L_i$$

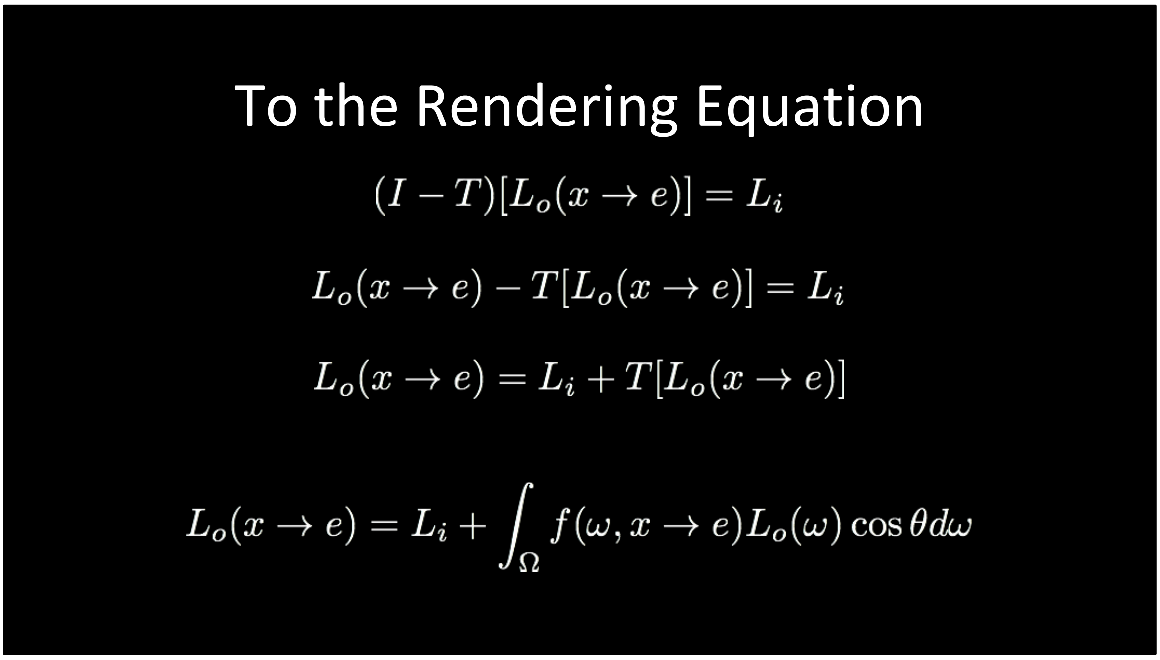


To the Rendering Equation

$$(I - T)[L_o(x \rightarrow e)] = L_i$$

$$L_o(x \rightarrow e) - T[L_o(x \rightarrow e)] = L_i$$

$$L_o(x \rightarrow e) = L_i + T[L_o(x \rightarrow e)]$$



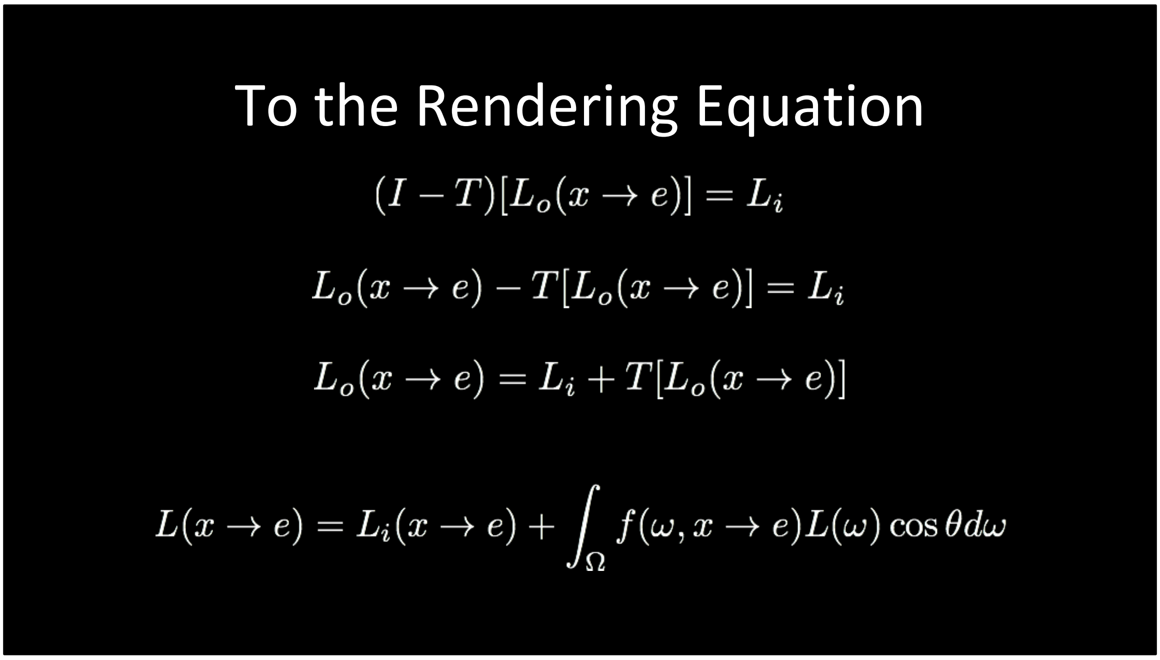
To the Rendering Equation

$$(I - T)[L_o(x \to e)] = L_i$$

$$L_o(x \to e) - T[L_o(x \to e)] = L_i$$

$$L_o(x \to e) = L_i + T[L_o(x \to e)]$$

$$L_o(x \to e) = L_i + \int_\Omega f(\omega, x \to e) L_o(\omega) \cos\theta d\omega$$



To the Rendering Equation

$$(I - T)[L_o(x \to e)] = L_i$$

$$L_o(x \to e) - T[L_o(x \to e)] = L_i$$

$$L_o(x \to e) = L_i + T[L_o(x \to e)]$$

$$L(x \to e) = L_i(x \to e) + \int_\Omega f(\omega, x \to e) L(\omega) \cos\theta d\omega$$

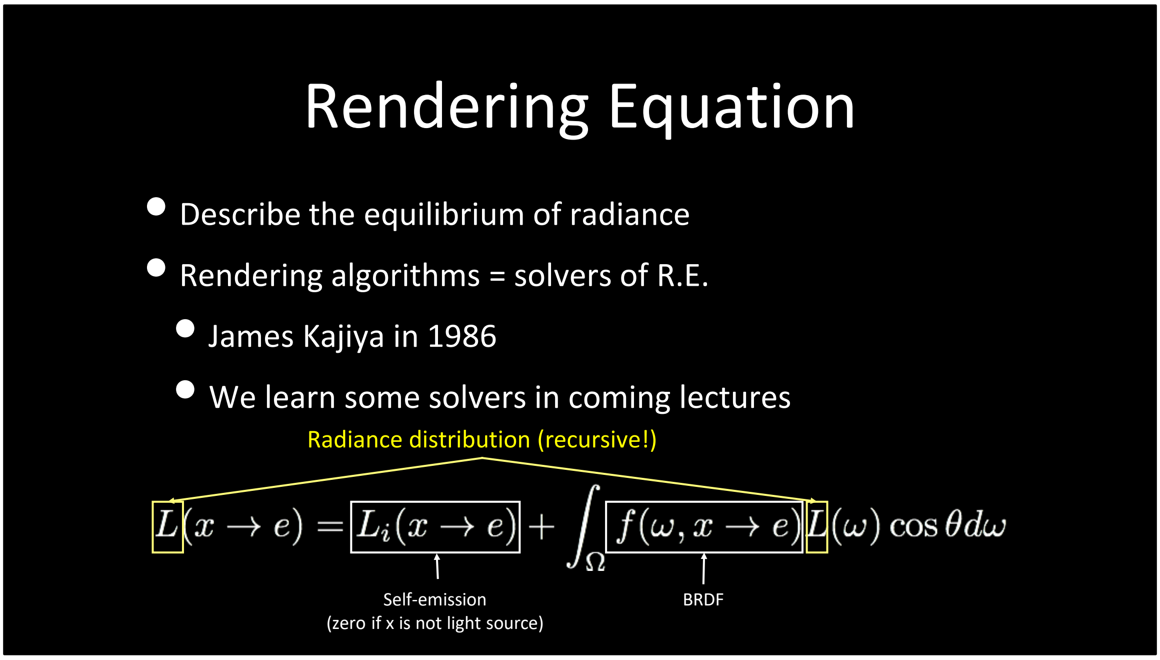
To the Rendering Equation

$$(I - T)[L_o(x \to e)] = L_i$$

$$L_o(x \to e) - T[L_o(x \to e)] = L_i$$

$$L_o(x \to e) = L_i + T[L_o(x \to e)]$$

$$\boxed{L(x \to e) = L_i(x \to e) + \int_\Omega f(\omega, x \to e) L(\omega) \cos\theta d\omega}$$

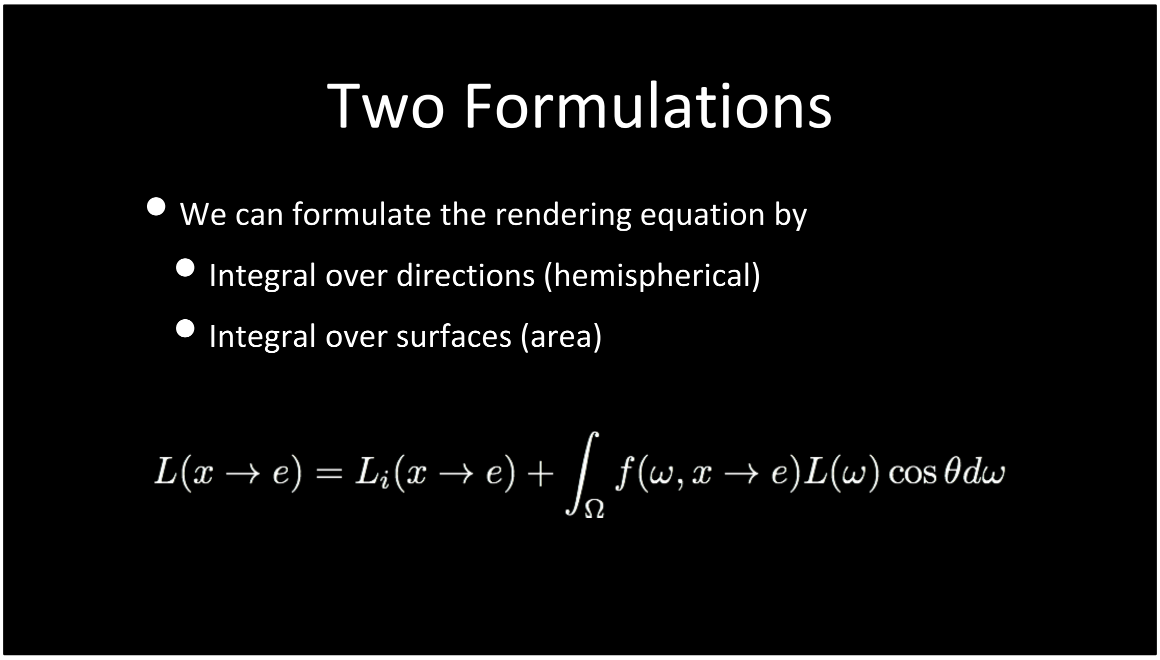


Rendering Equation

- Describe the equilibrium of radiance
- Rendering algorithms = solvers of R.E.
  - James Kajiya in 1986
  - We learn some solvers in coming lectures

Radiance distribution (recursive!)

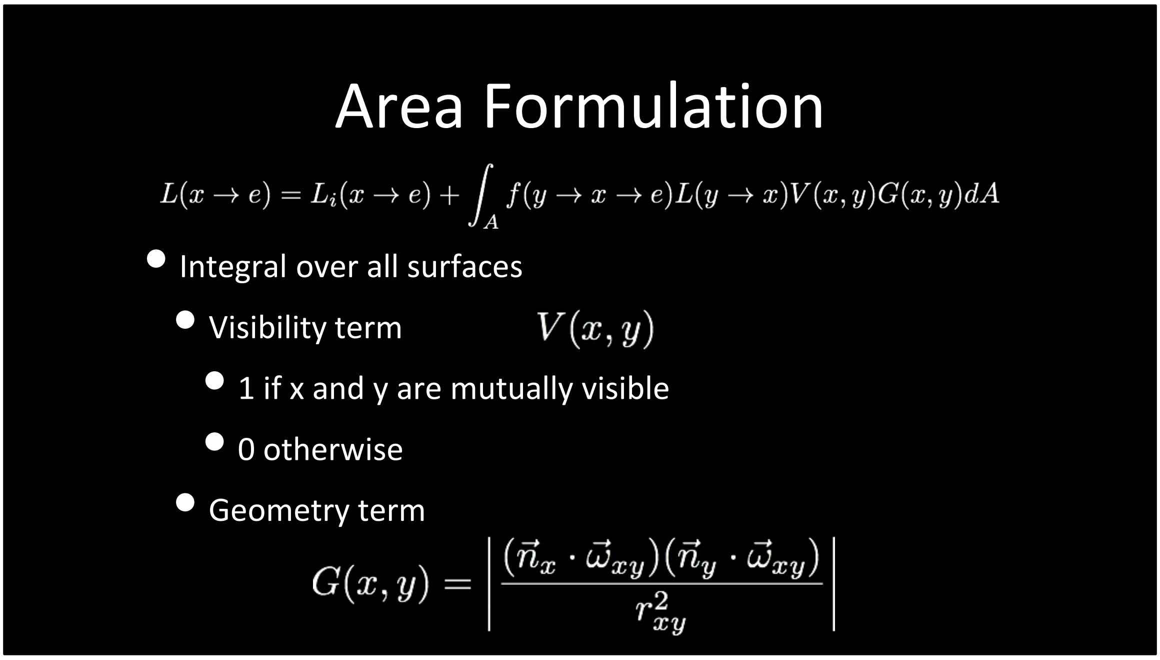
$$\boxed{L}(x \rightarrow e) = \underbrace{\boxed{L_i(x \rightarrow e)}}_{\substack{\text{Self-emission} \\ \text{(zero if x is not light source)}}} + \int_{\Omega} \underbrace{\boxed{f(\omega, x \rightarrow e)}}_{\text{BRDF}} \boxed{L}(\omega) \cos\theta d\omega$$



Two Formulations

- We can formulate the rendering equation by
  - Integral over directions (hemispherical)
  - Integral over surfaces (area)

$$L(x \to e) = L_i(x \to e) + \int_\Omega f(\omega, x \to e) L(\omega) \cos\theta d\omega$$



Area Formulation

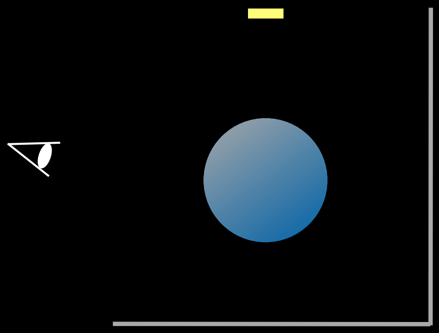
$$L(x \to e) = L_i(x \to e) + \int_A f(y \to x \to e) L(y \to x) V(x,y) G(x,y) dA$$

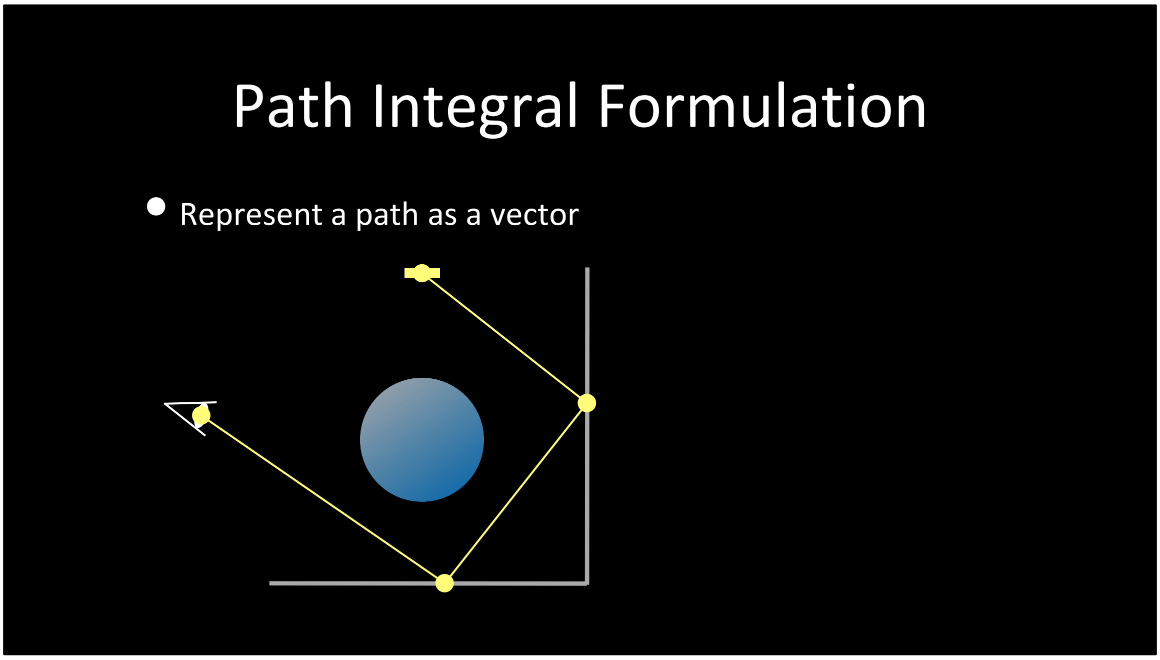
- Integral over all surfaces
  - Visibility term $V(x,y)$
    - 1 if x and y are mutually visible
    - 0 otherwise
  - Geometry term

$$G(x,y) = \left| \frac{(\vec{n}_x \cdot \vec{\omega}_{xy})(\vec{n}_y \cdot \vec{\omega}_{xy})}{r_{xy}^2} \right|$$

Path Integral Formulation

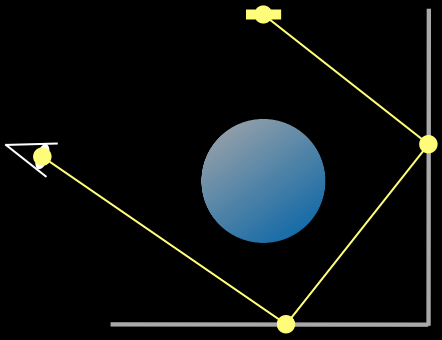
- Represent a path as a vector





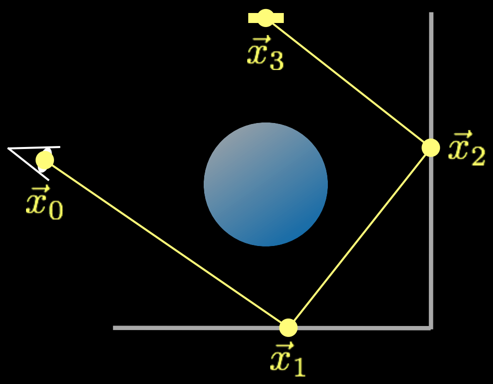
Path Integral Formulation

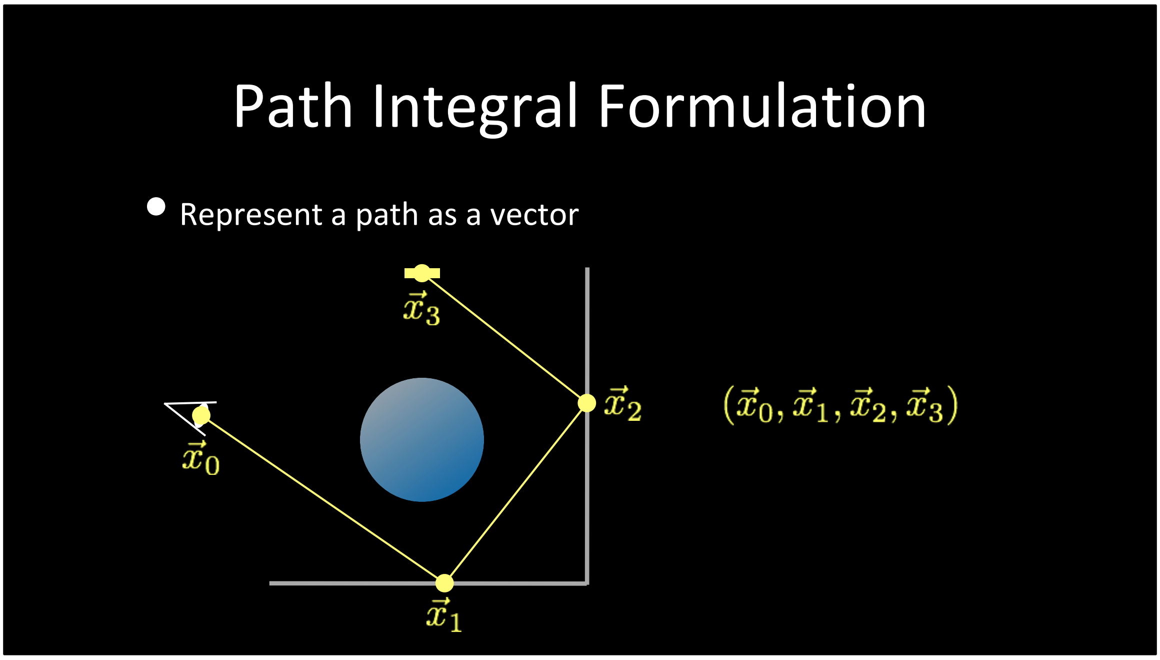
- Represent a path as a vector



Path Integral Formulation

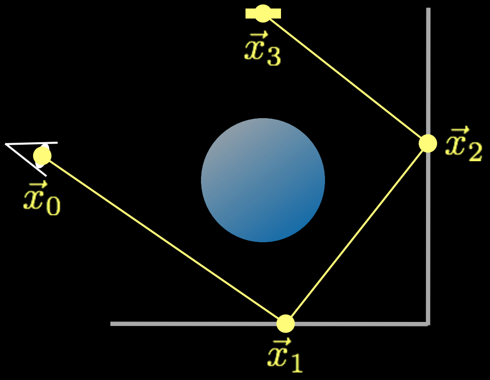
- Represent a path as a vector





Path Integral Formulation

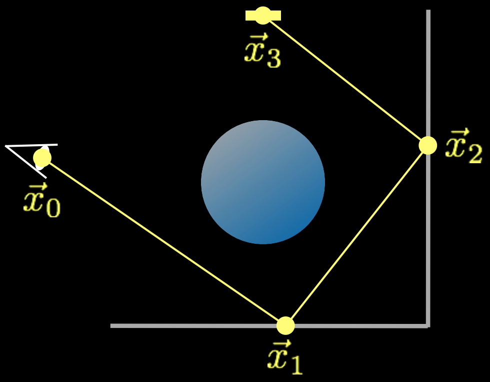
- Represent a path as a vector



$(\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3)$

Path Integral Formulation

- Represent a path as a vector

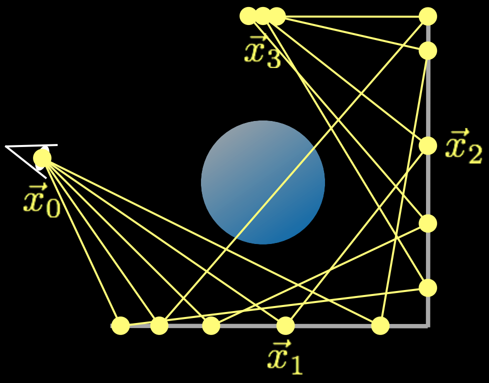


$$T(\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3)$$

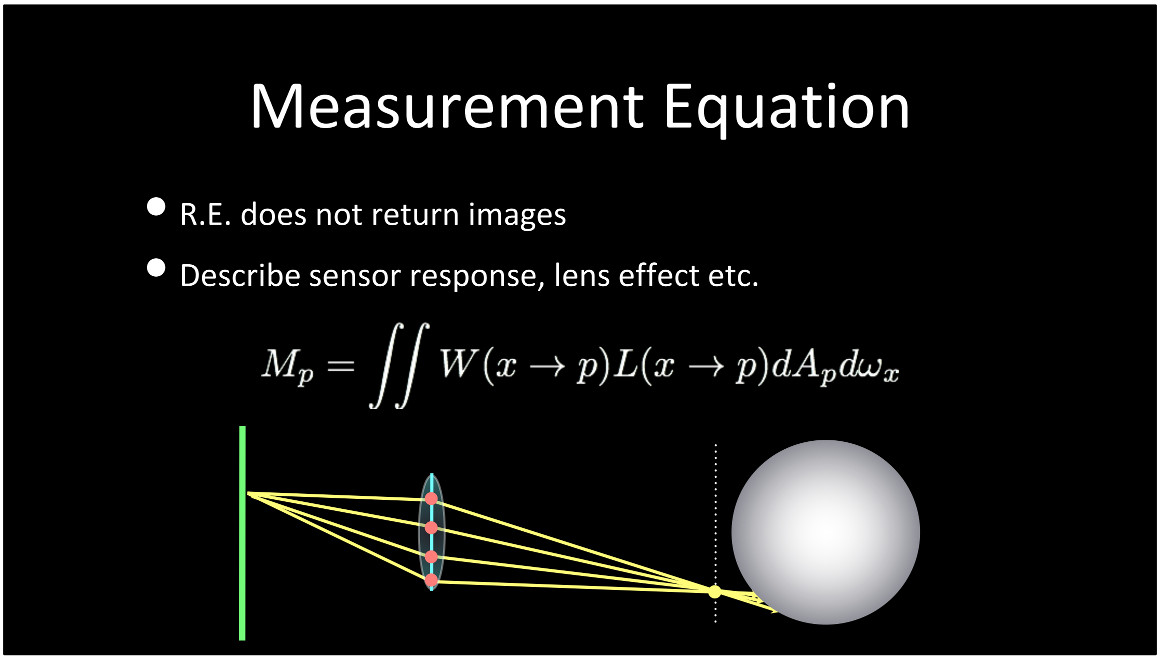
Path throughput
(via the rendering eqn.)

Path Integral Formulation

- Represent a path as a vector



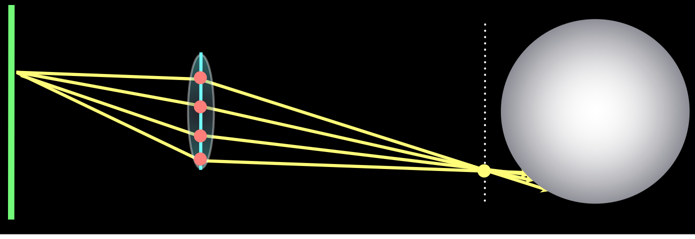
$$\int T(\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3)\, dA^4$$

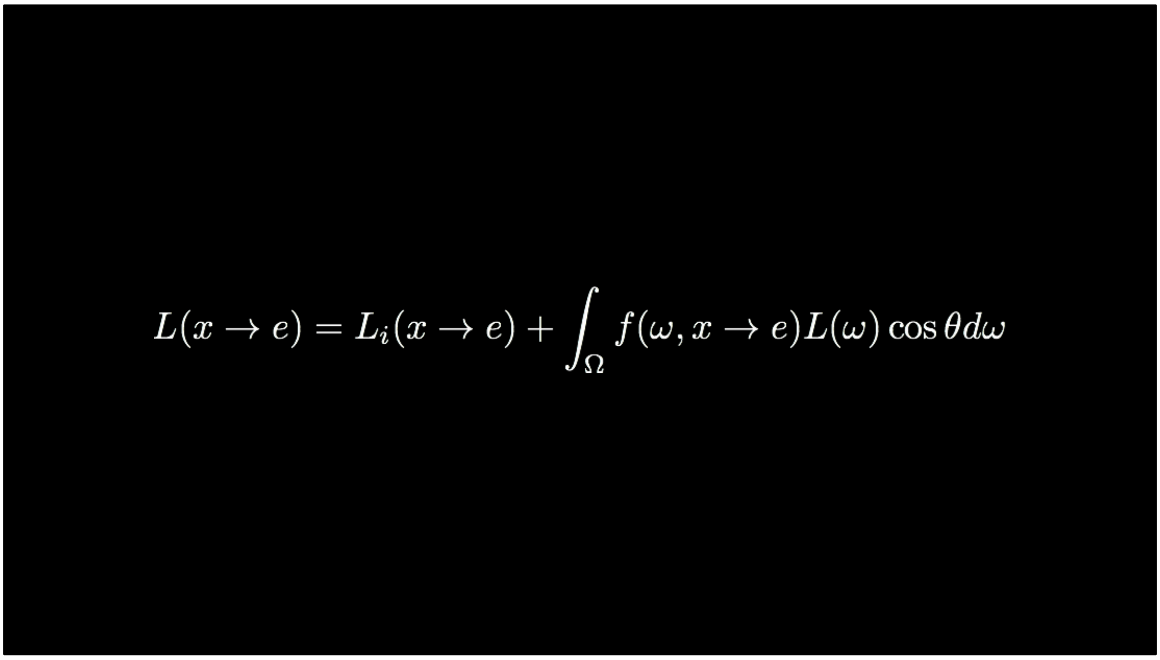


Measurement Equation

- R.E. does not return images
- Describe sensor response, lens effect etc.

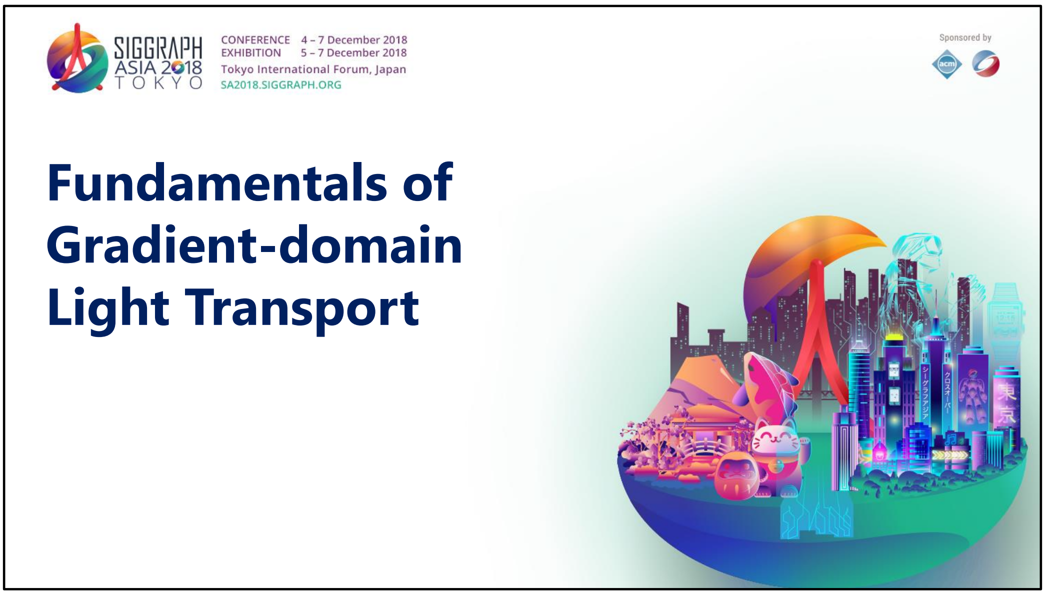
$$M_p = \iint W(x \to p) L(x \to p) dA_p d\omega_x$$





$$L(x \to e) = L_i(x \to e) + \int_{\Omega} f(\omega, x \to e) L(\omega) \cos\theta d\omega$$

4.3 Fundamentals of gradient-domain light transport

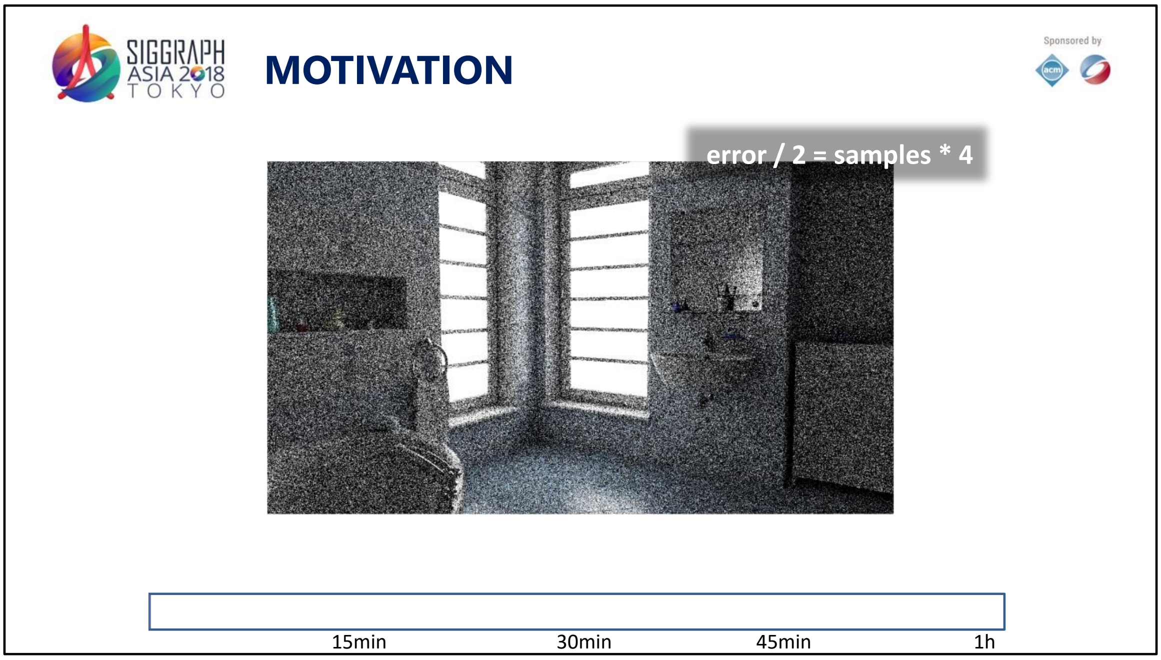


SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

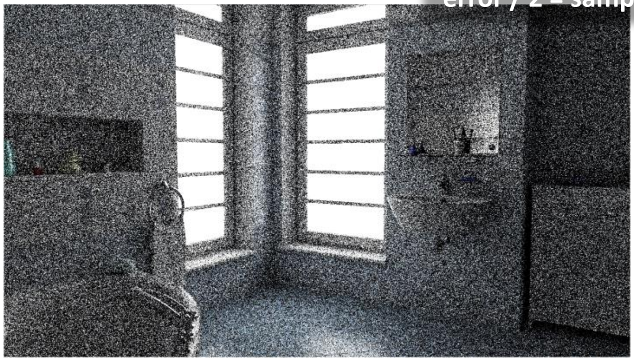
Sponsored by

Fundamentals of Gradient-domain Light Transport



MOTIVATION

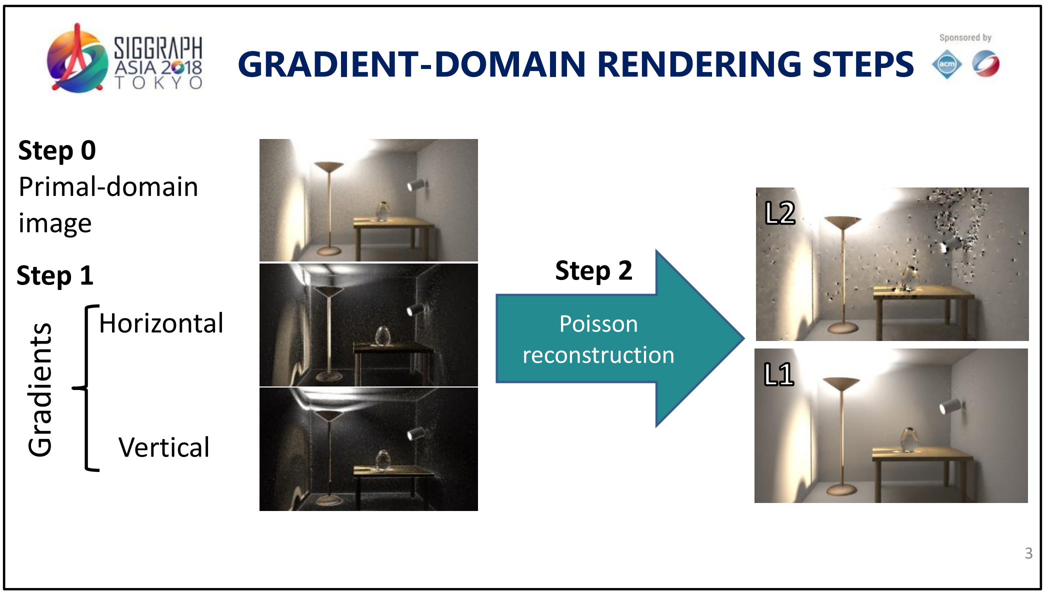
error / 2 = samples * 4



15min 30min 45min 1h

Gradient-domain rendering is motivated by the slow convergence of Monte Carlo light transport algorithms. As can be seen in the rendering of this bathroom scene, unconverged images exhibit high frequency noise that can only be reduced by increasing rendering time. A well known theory is that to halve the error however we have to increase rendering time by 4, which makes this process painfully slow.

Gradient-domain rendering aims at developing an algorithm that **speeds up the rendering process** and hence reduces the amount of time required to get an image of a certain quality.



GRADIENT-DOMAIN RENDERING STEPS

Step 0
Primal-domain image

Step 1
Gradients
Horizontal
Vertical

Step 2
Poisson reconstruction

L2
L1

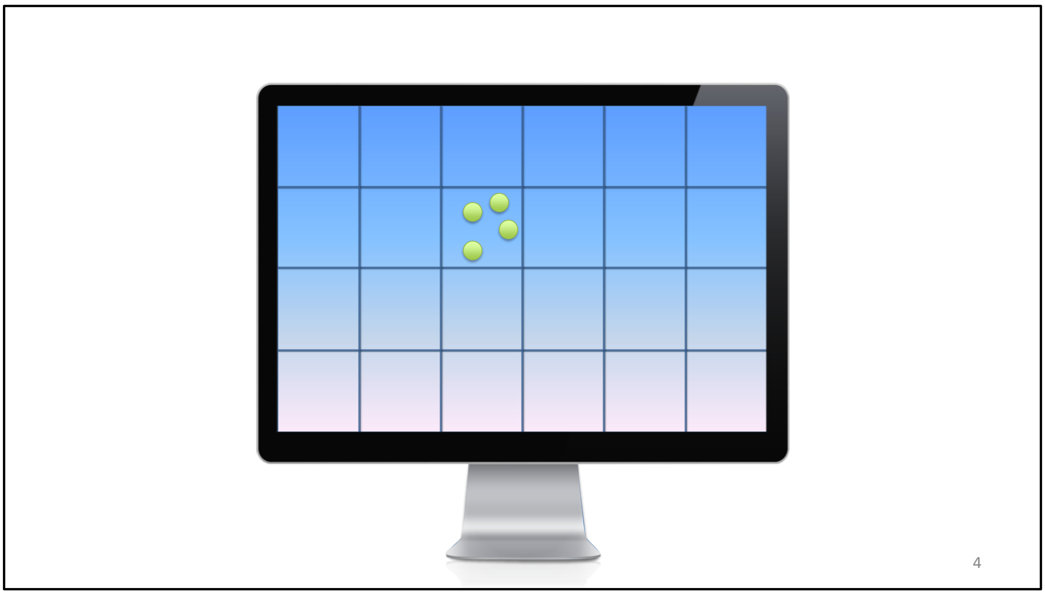
3

We ask the question whether doing computation in some other domains, e.g., gradients, is more efficient than estimating image pixels directly. If we only estimate the gradients by taking finite difference of pixel values, this will not yield any better efficiency. And today we will see that by carefully designing the gradient estimation, we can do better when rendering in the gradient domain.

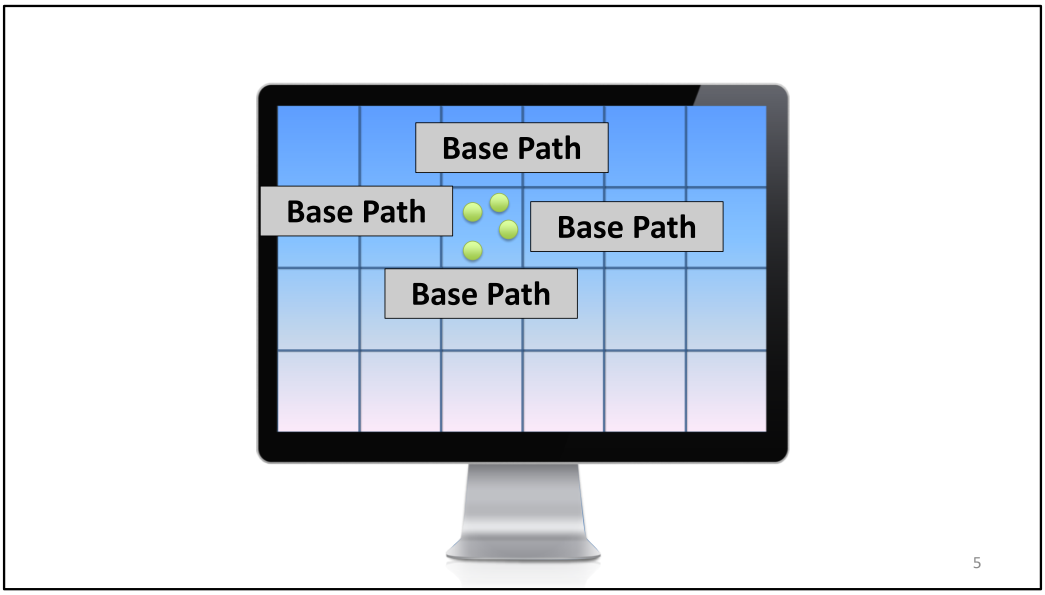
An overview of gradient-domain rendering is shown in this diagram. The main steps include (1) estimating the image-space gradients, and (2) perform a Poisson reconstruction by taking as input the original image and its estimated gradients.

In step 1, gradient estimation is usually performed by exploring path coherency. This results in more accurate gradients than simply taking finite difference directly on the original image. In step 2, depending on the norm used in the Poisson reconstruction we can have: unbiased results with L2 norm that is subject to visual artifacts, or more robust but biased results with L1 norm.

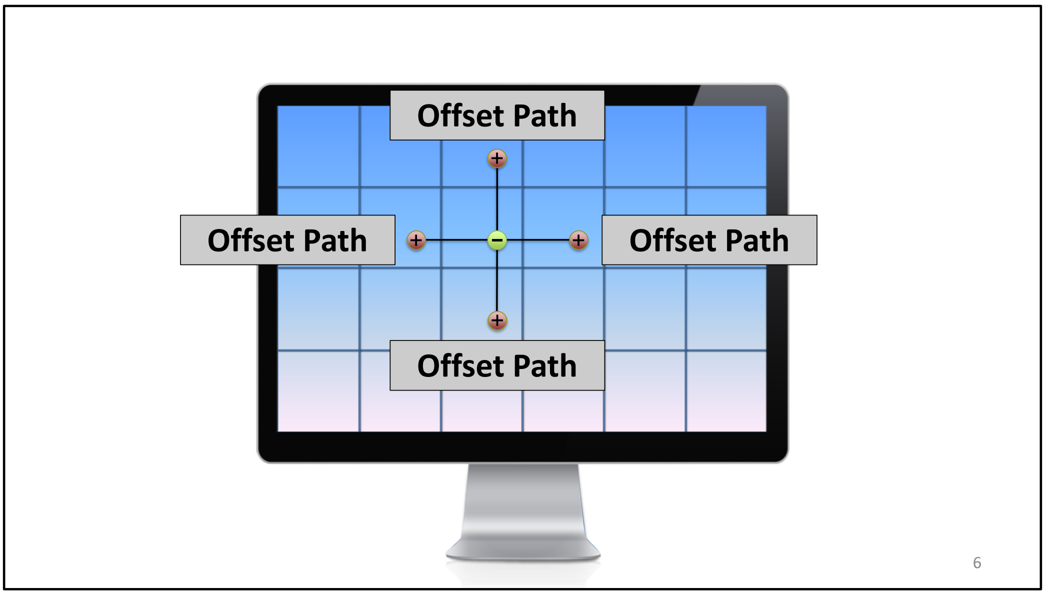
In this section, let’s explore gradient-domain rendering with a simple algorithm: path tracing.



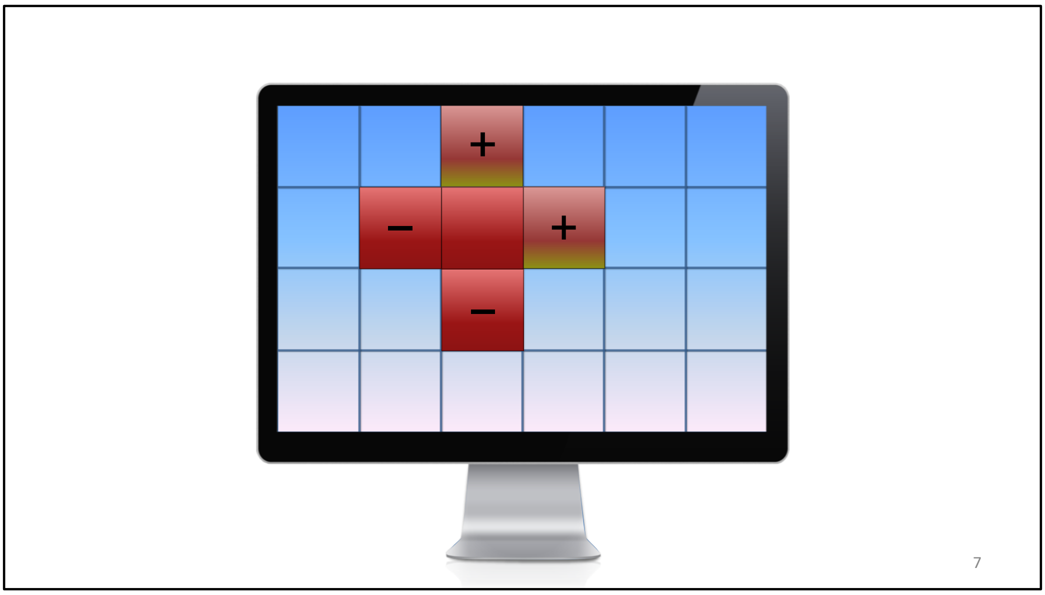
We drive gradient domain path tracing by a standard path tracer that shoots a number of paths through each pixel.



We call these paths **base paths**.

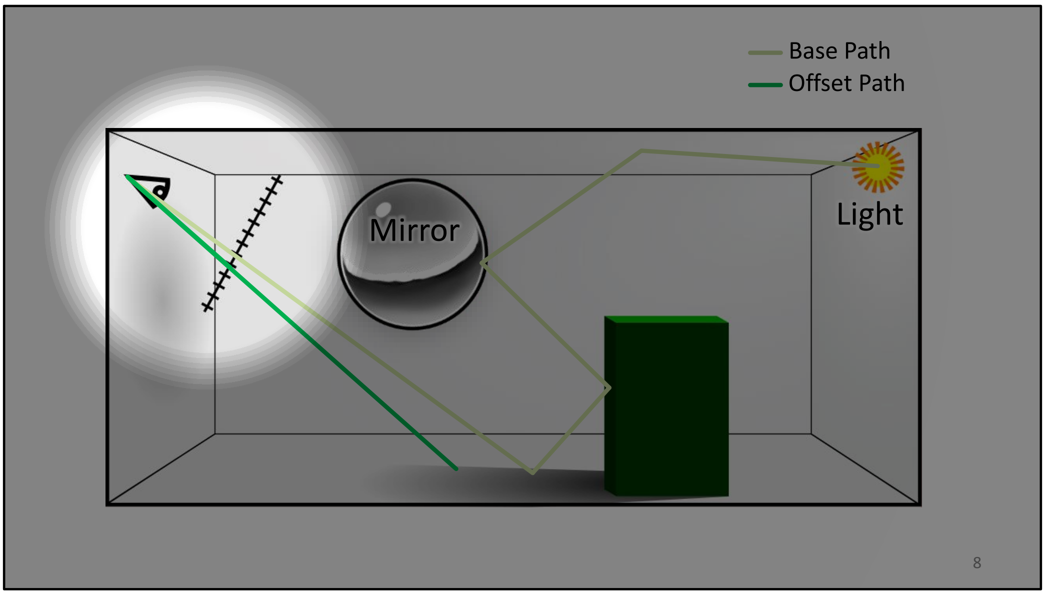


To obtain the finite difference gradients, we **shift each** base path to all four neighboring pixels. We call these paths **offset paths**.



Sampling a base path and shifting it to the four neighbors gives us a throughput estimate and four gradient estimates: left, top, right and bottom. These are accumulated into their respective buffers.

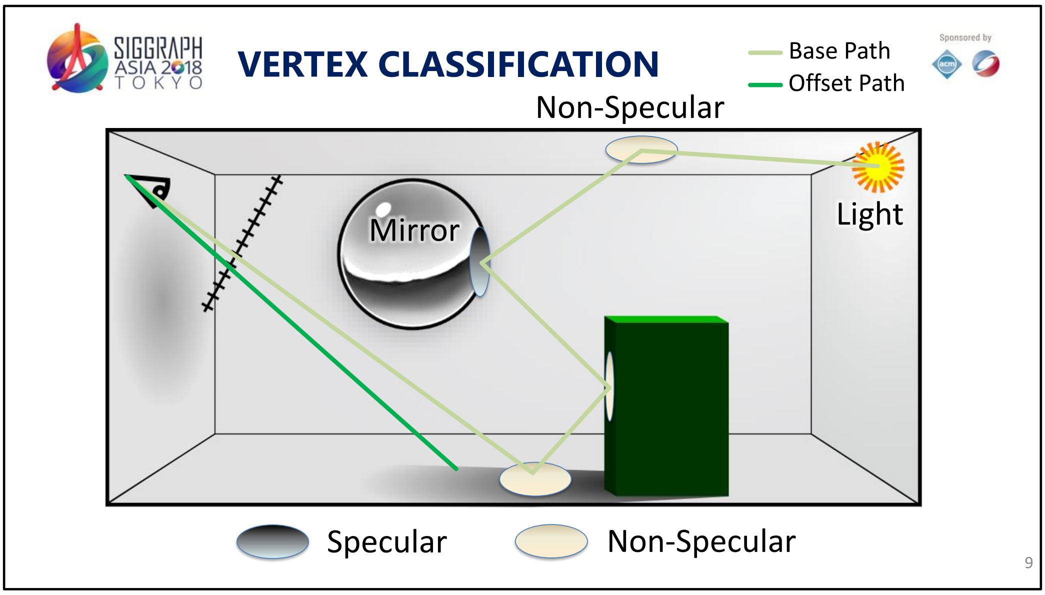
Let’s now have a look at how to actually construct the offset paths.



Let’s look at a simple scene where we have shot this base path using a standard path tracer.

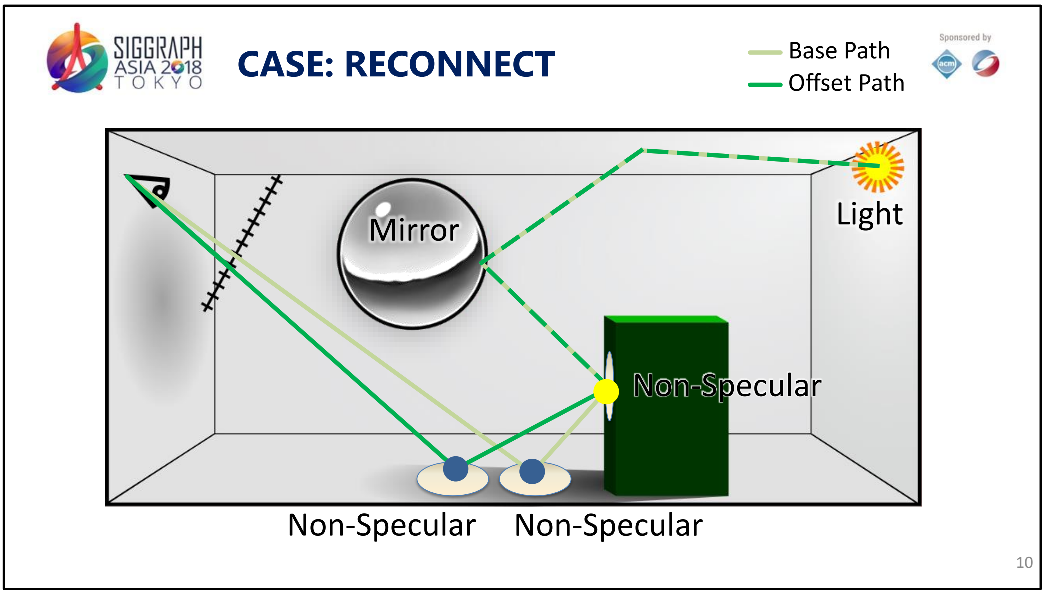
We then generate an offset path, shown in green. It needs to go exactly one pixel apart, and this determines the first leg of the offset path.

As we discuss in the paper, the shift is mathematically a change of integration variables, so we have a lot of possibilities on how to continue. Intuitively, we want to construct offset paths that are as similar as possible to their base paths. This leads to small throughput differences which reduces noise significantly.

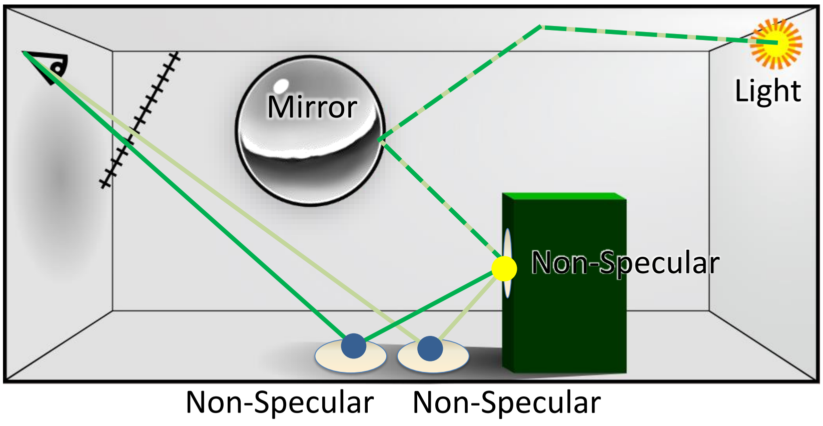


VERTEX CLASSIFICATION

Before we go on, we classify all path vertices as either **specular**, or **non-specular**, based on a threshold on the roughness of the BRDF.



CASE: RECONNECT



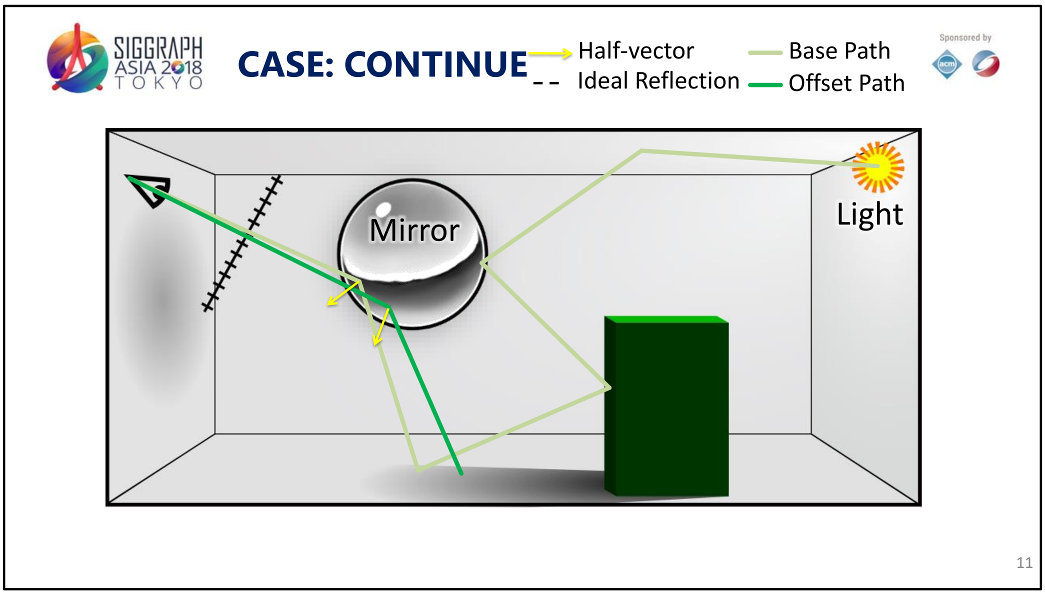
10

We distinguish between two cases:

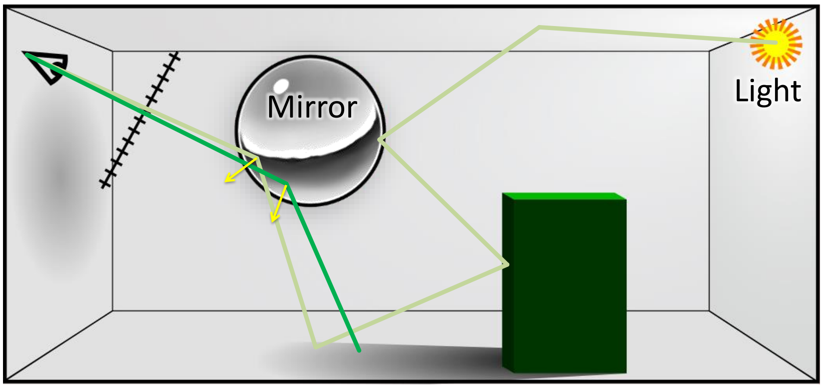
(1) If both the current base and offset vertices, shown in blue here, are non-specular, and the next vertex of the base path, shown in yellow, is also non-specular, then we reconnect the offset path to the base path. After this, we let the offset path follow the base path until we reach the light.

This leads to paths that are as similar as possible. As a result, the throughput difference will be small, because the vertices where the paths differ are non-specular.

(2) However, for highly glossy materials, which we classify as specular, this direct reconnection strategy may not work well because the incoming and outgoing directions change when we reconnect, which might cause the BRDFs at the reconnection segment evaluated to zero.



CASE: CONTINUE

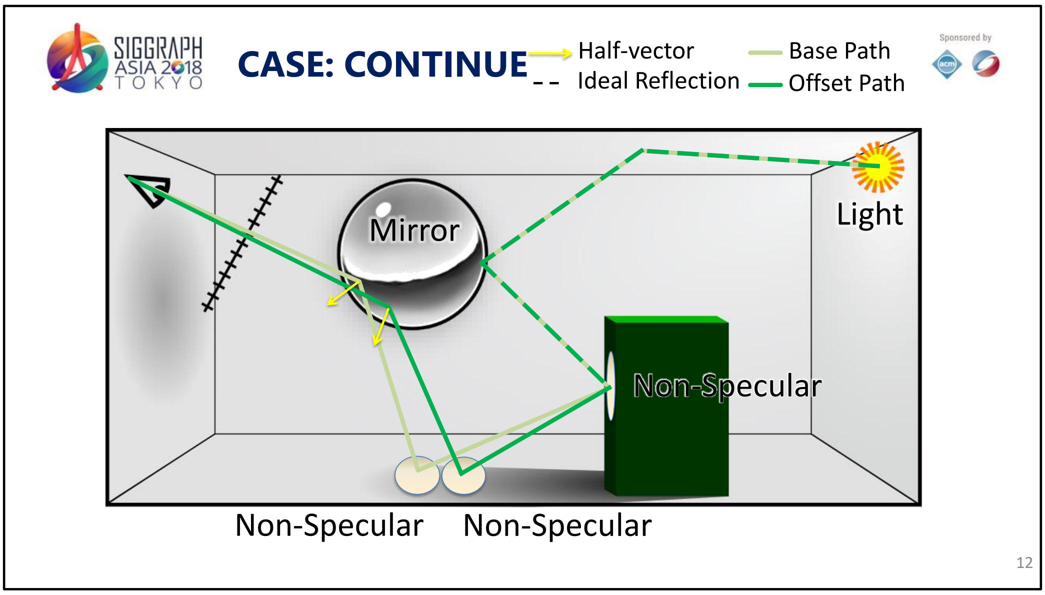


11

We address this issue by continuing to trace the offset path if we hit a vertex classified as specular.

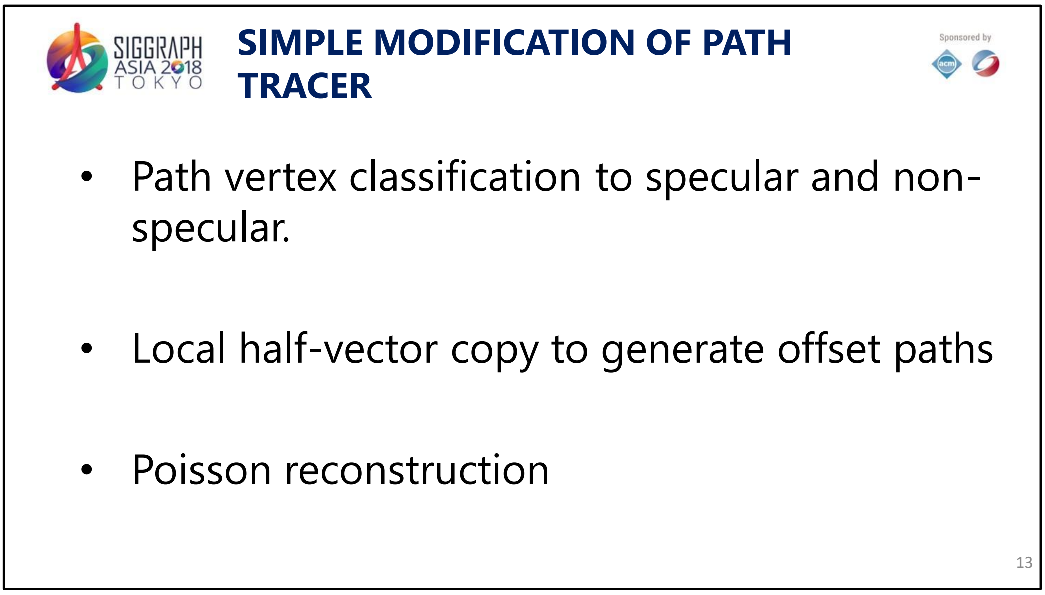
Assume that we have shot this base path whose first vertex is at a specular surface. In this case, the offset path should have a similar BRDF value at the corresponding vertex.

We achieve this by copying the half-vector in local tangent coordinates, and this determines the direction of the next path segment.



CASE: CONTINUE

We continue by duplicating local half-vectors until we encounter two consecutive non-specular vertices in the base path, and the offset path is also sitting at a non-specular vertex. In this configuration we can safely reconnect to the base path.



SIMPLE MODIFICATION OF PATH TRACER

- Path vertex classification to specular and non-specular.
- Local half-vector copy to generate offset paths
- Poisson reconstruction

13

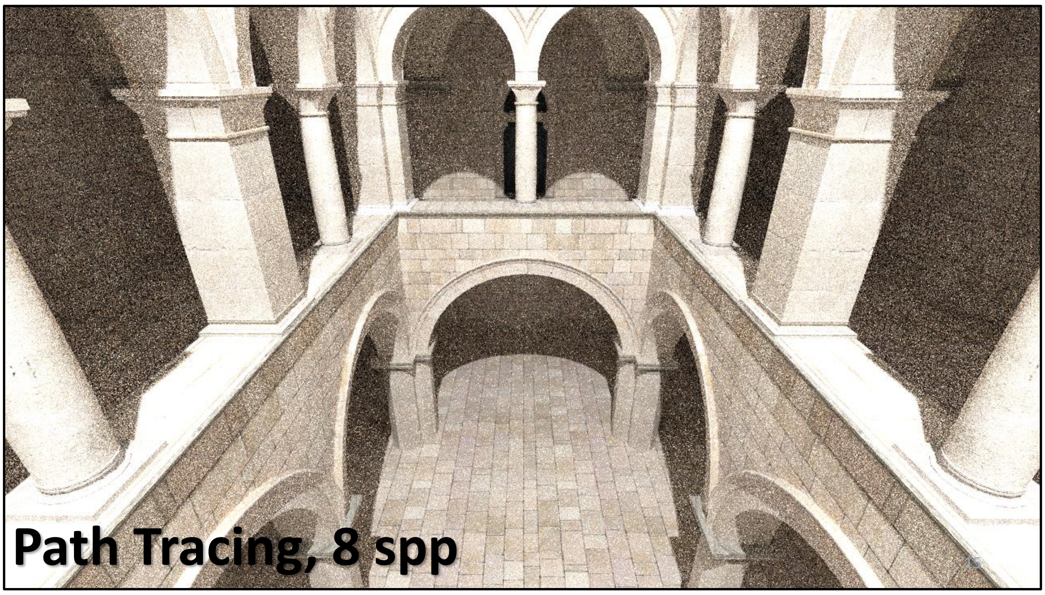
Now, this is almost all you need to know to implement your own gradient-domain path tracer.

Let’s see the some results.

Results

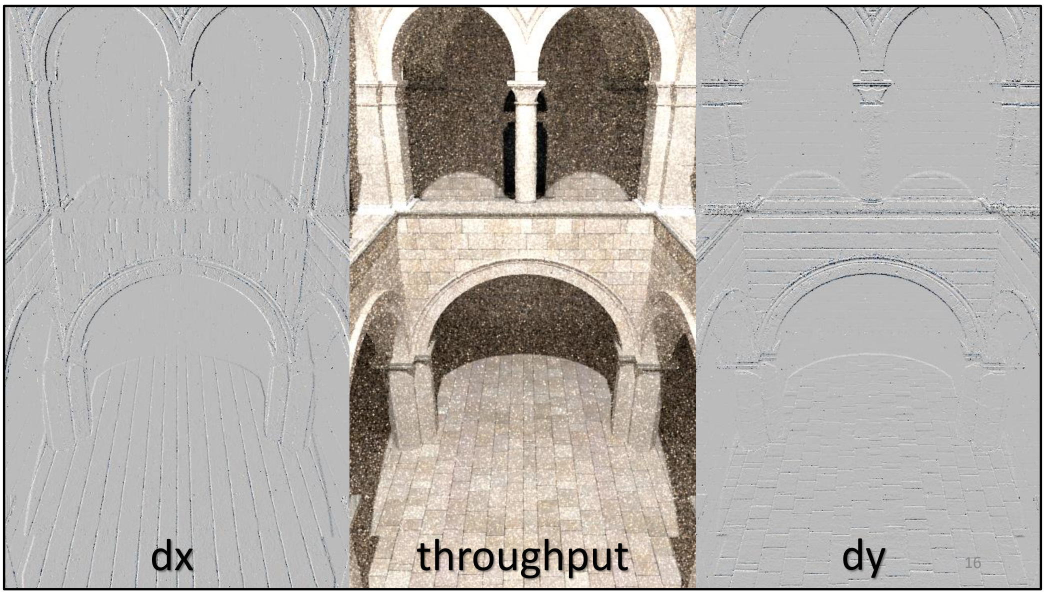
14

So, let’s look at results.



Sponza is a simple diffuse scene, lit from above by a large area light.

Rendering with the path tracer in Mitsuba, with 8 samples per pixel, we get heavy noise.



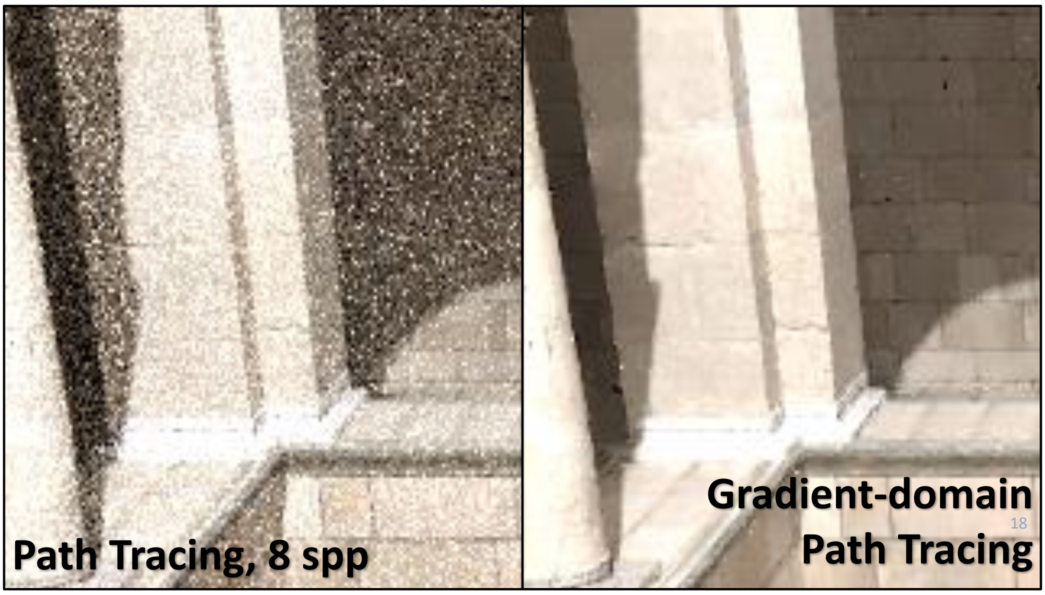
In equal-time, the gradient-domain sampler outputs the horizontal gradients, the vertical gradients, and the throughput image. The throughput image is basically similar to an ordinary noisy path traced image.

We then perform a reconstruction to obtain the final result.

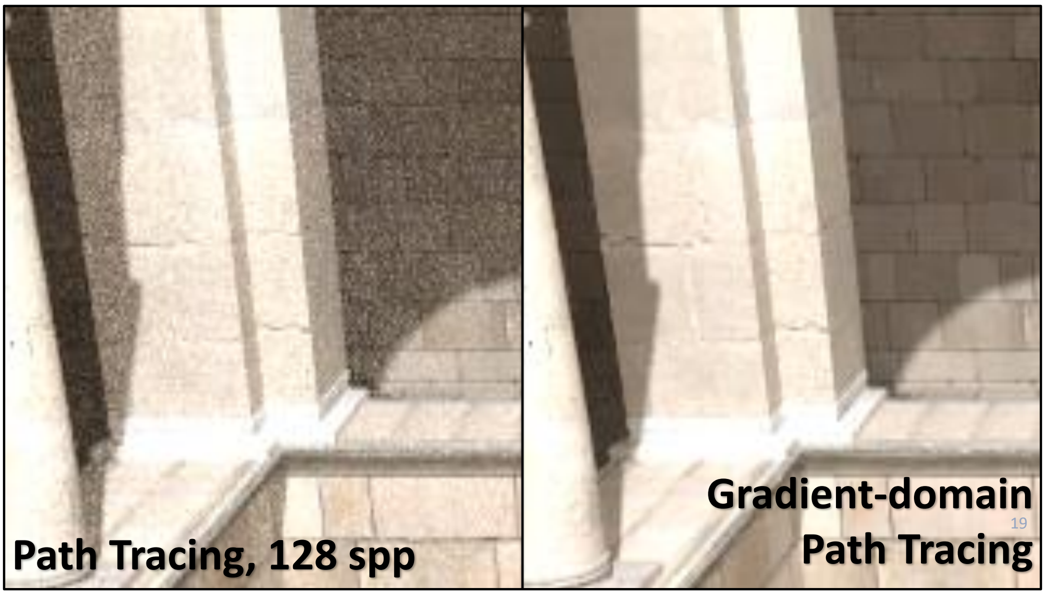
Gradient-domain Path Tracing

17

Here is the result of the reconstruction.

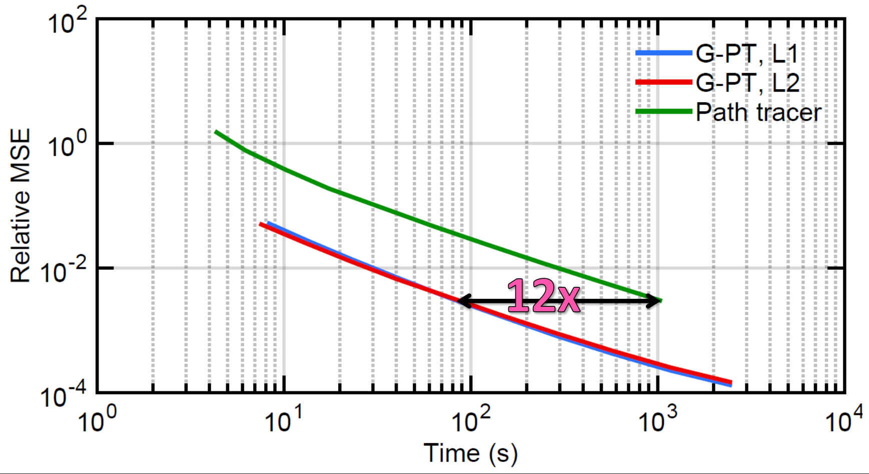


Gradient-domain rendering cleans most of the high frequency noise.



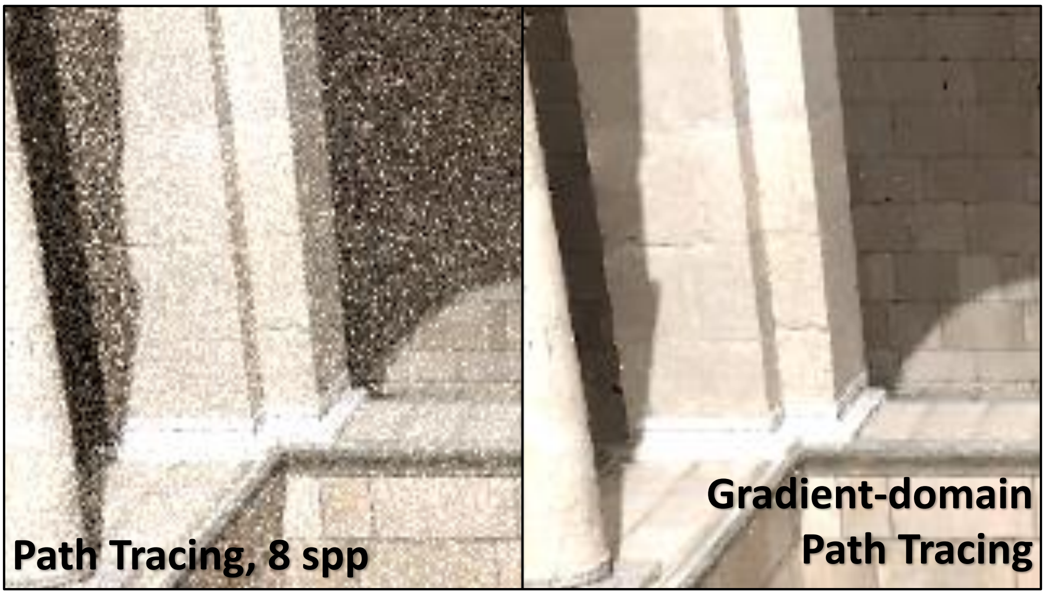
With sixteen times more samples, the path traced image is still very noisy, but our result is essentially ready.

Error vs. Time (Sponza)

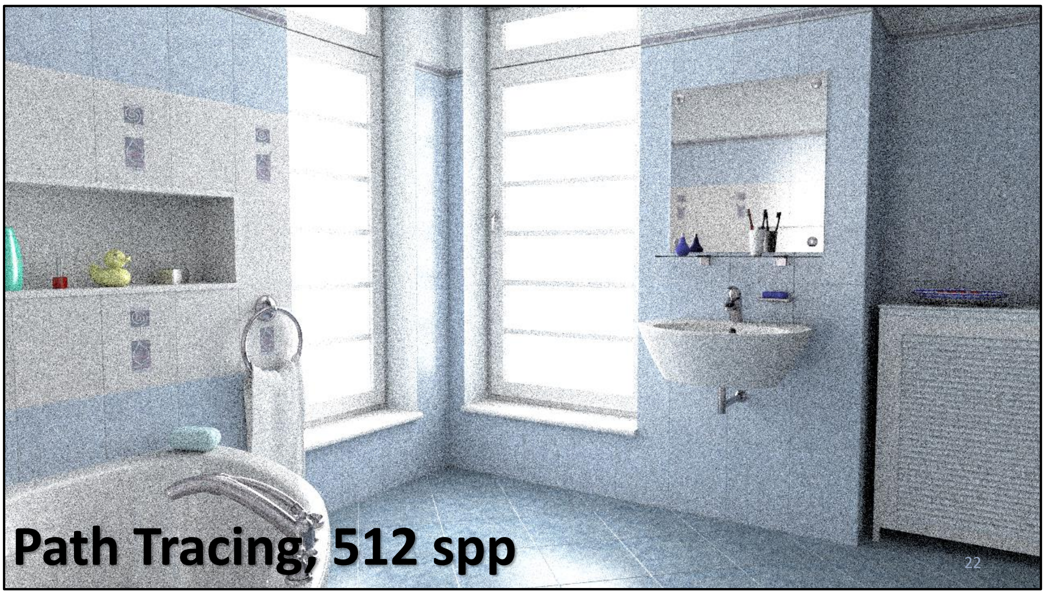


Plotting relative mean square error against time, we're almost twelve times faster.

When we saw how well this works, we were really quite surprised.



Sponza you already saw; in this easy scene the method really shines.



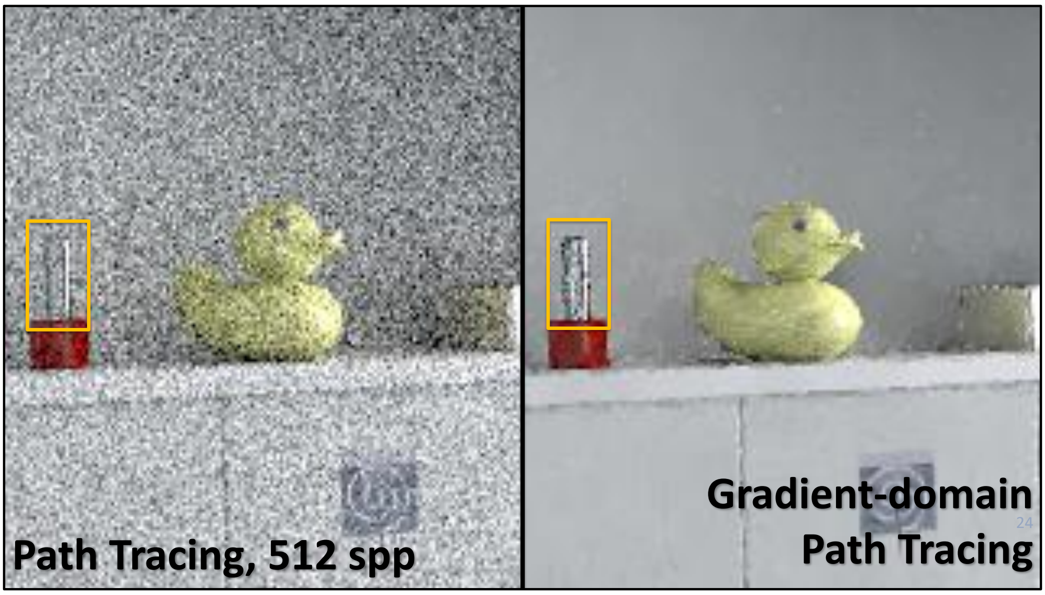
The bathroom scene is hard for path tracing, as it’s illuminated from behind a window, and this makes shadow rays useless. It also has lots of glossy materials and a mirror.

Five hundred samples per pixel are not enough to get a converged image with standard path tracing.

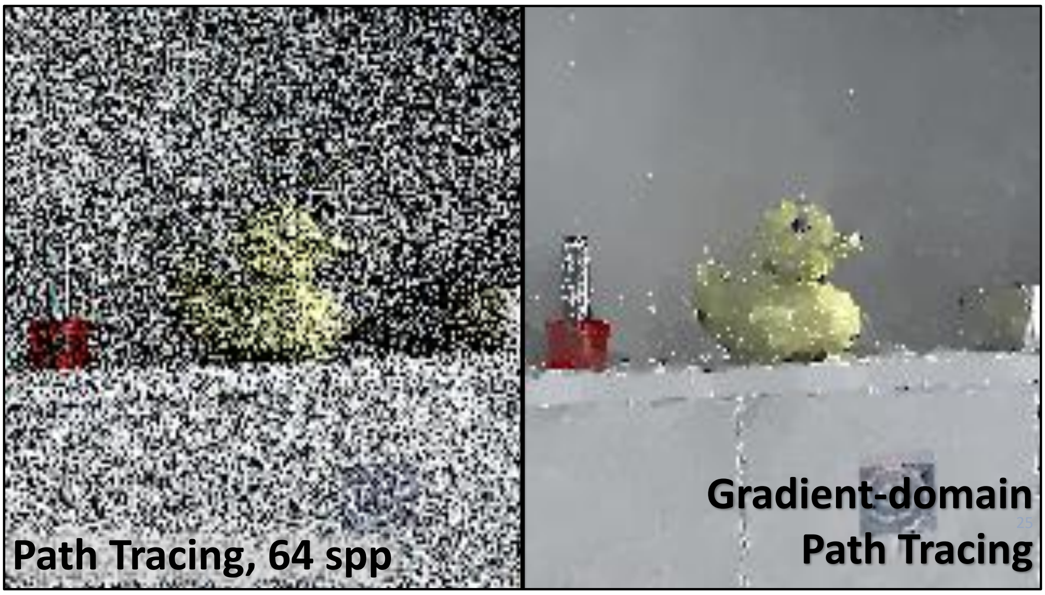
Gradient-domain Path Tracing

23

Despite the somewhat complicated light transport, we manage to kill most of the high frequency noise.

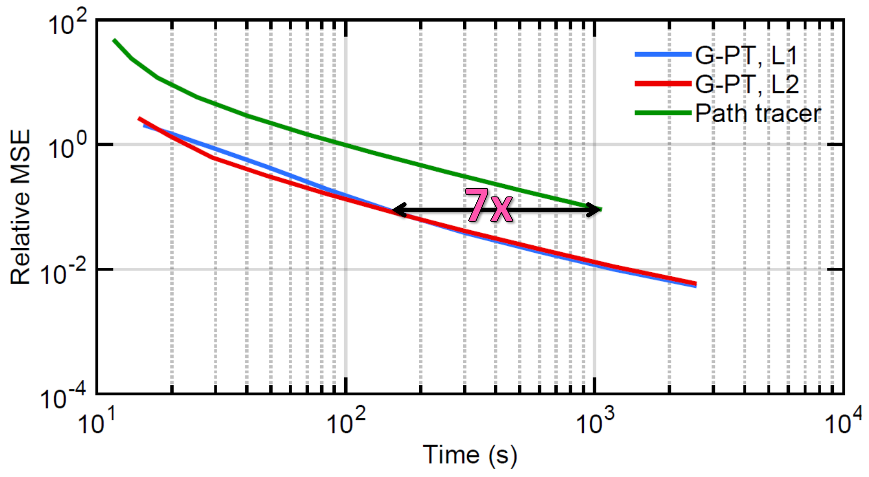


Zooming in, we see that we're doing much better than path tracing, although the level of noise is non-uniform as we can see in this highly curved glossy object. We discuss this more in the paper.



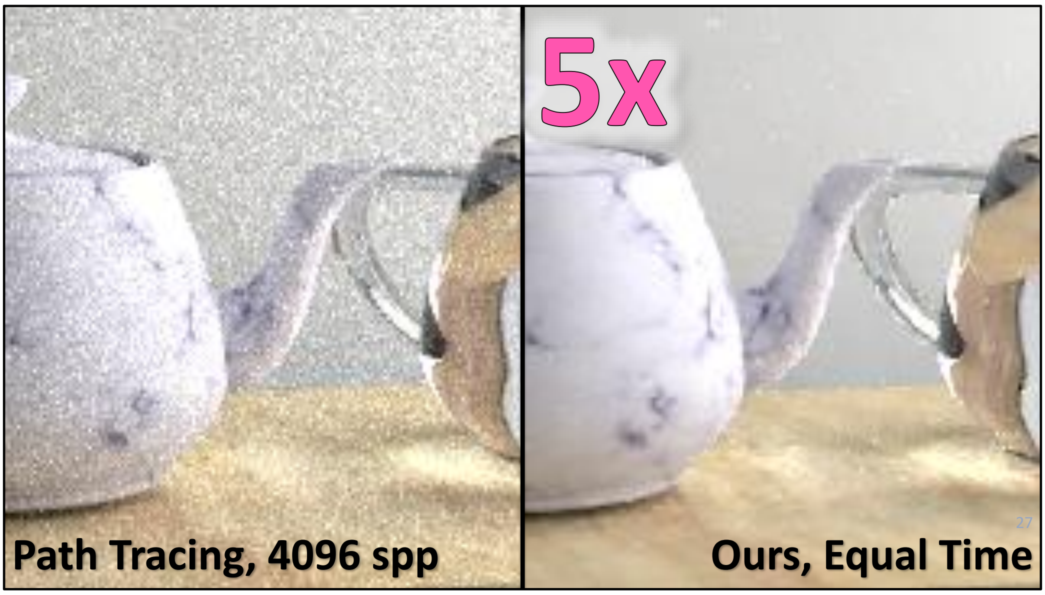
Still, with rendering time that's not enough for standard path tracing to even produce an image, we get something recognizable out, which I think is pretty cool.

Error vs. Time (Bathroom)



26

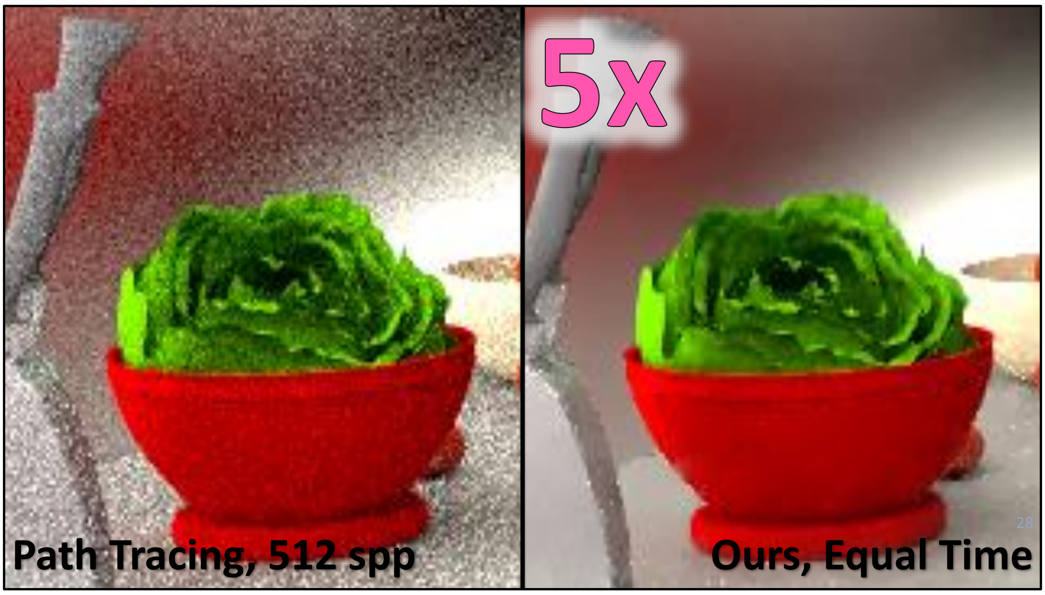
In this scene, we are around seven times faster than standard path tracing.



The Door scene is designed to be hard for path tracing since all light comes from another room through the slightly opened door.

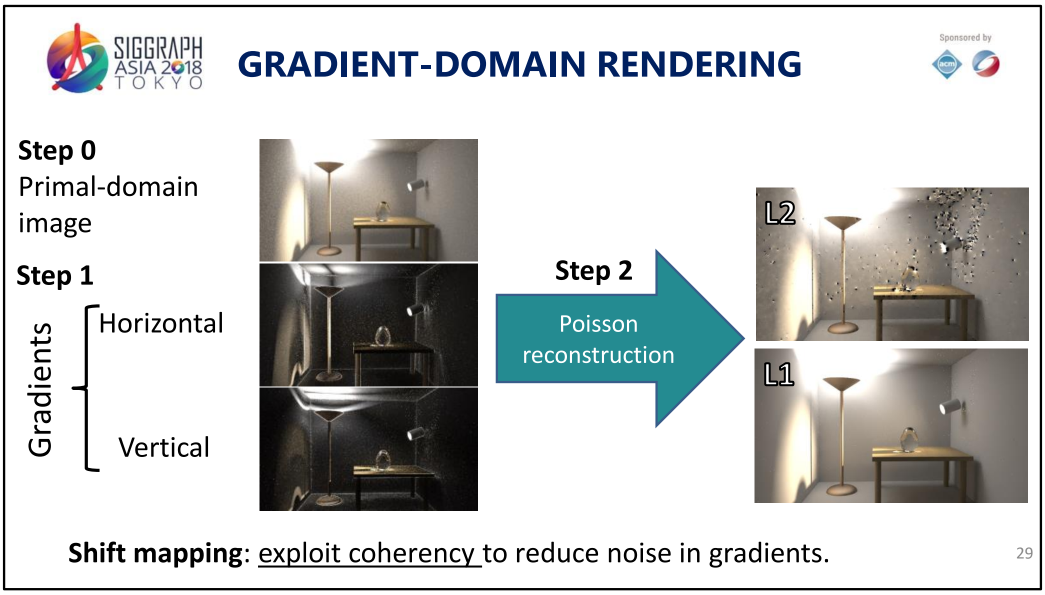
Here too we produce a significantly better image in equal time, despite the complexity of the light transport.

The speedup is roughly a factor of five.



This Kitchen scene is very challenging for the gradient path tracer with its glossy surfaces lit by a small and hard to reach light source.

Still, we again outperform standard path tracing by a factor of five.



GRADIENT-DOMAIN RENDERING

Step 0
Primal-domain image

Step 1

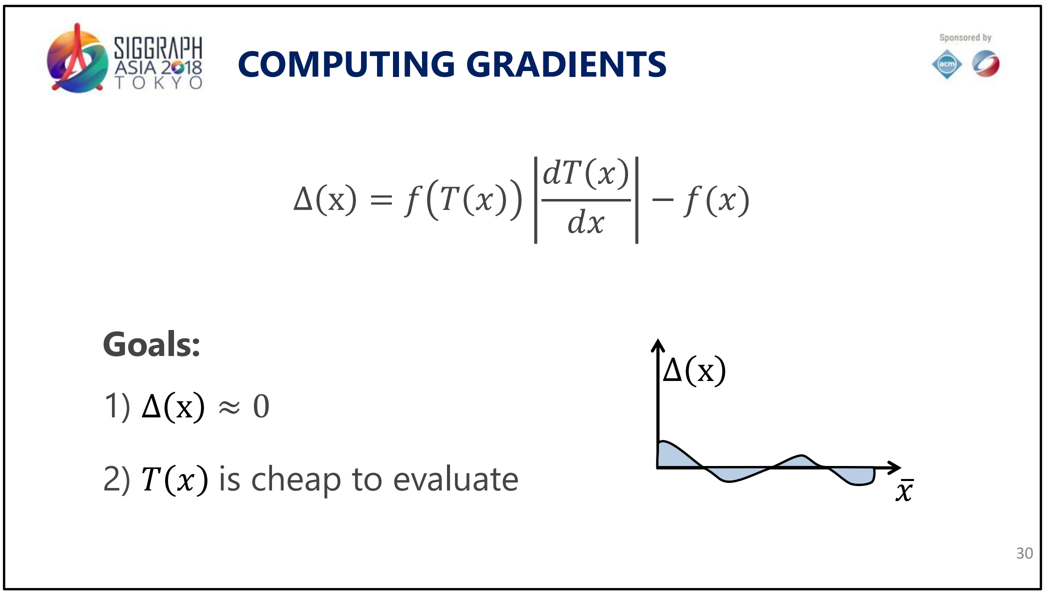
Gradients: Horizontal, Vertical

Step 2

Shift mapping: exploit coherency to reduce noise in gradients.

29

Here is a quick recap. It is important to get familiar with the steps as we will keep discussing about them in the subsequent sections.



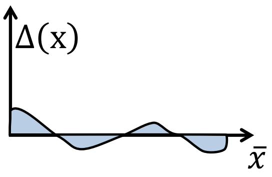
COMPUTING GRADIENTS

$$\Delta(\mathrm{x}) = f(T(x))\left|\frac{dT(x)}{dx}\right| - f(x)$$

Goals:

1) $\Delta(\mathrm{x}) \approx 0$

2) $T(x)$ is cheap to evaluate



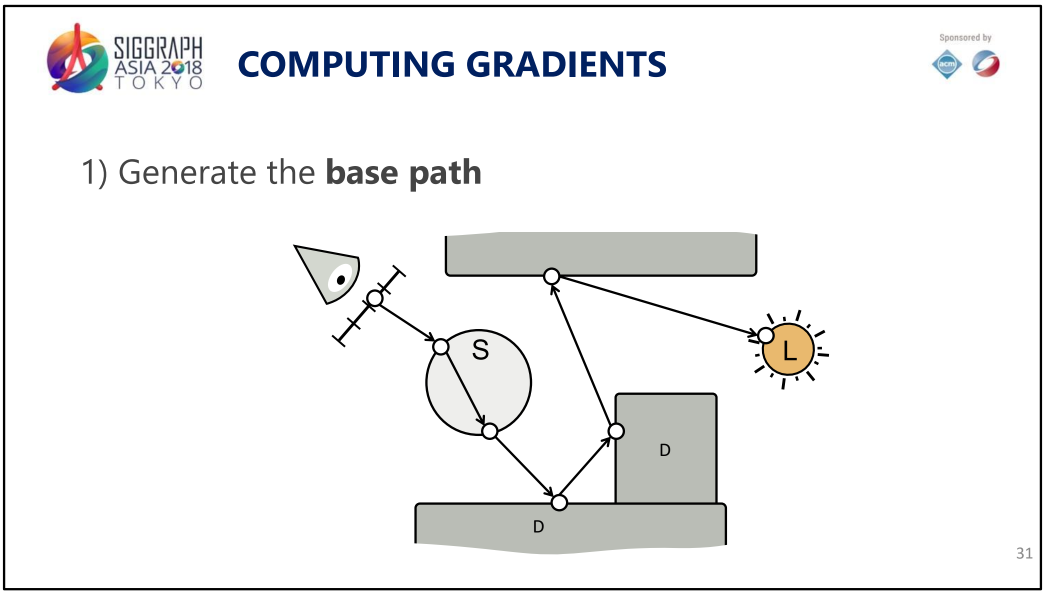
30

Let’s keep the explanation of the formula to a few minutes later, and focus on the goals first.

(1) To compute the image-space gradients, an important observation is that in general, images are smooth and their gradients are very close to zero. Therefore, we want to design an estimator for the gradients such that it outputs values that are very close to zero. To do this, the central idea is to rely on a shift mapping function T(x) that transforms a base light path x to an offset light path that is highly coherent with the base path. This coherency makes the contribution of the base and offset paths to be very similar, or their difference (image-space gradients) very small. This results in low variance estimation for the gradients.

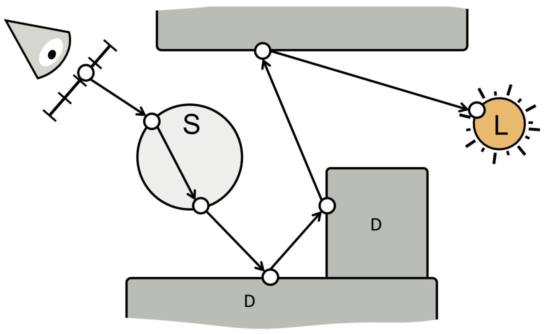
(2) The evaluation of shift mapping function T(x) is fast to avoid causing too much overhead during gradient estimation.

If we are able to achieve these two goals, gradient-domain rendering algorithm is usually very efficient.



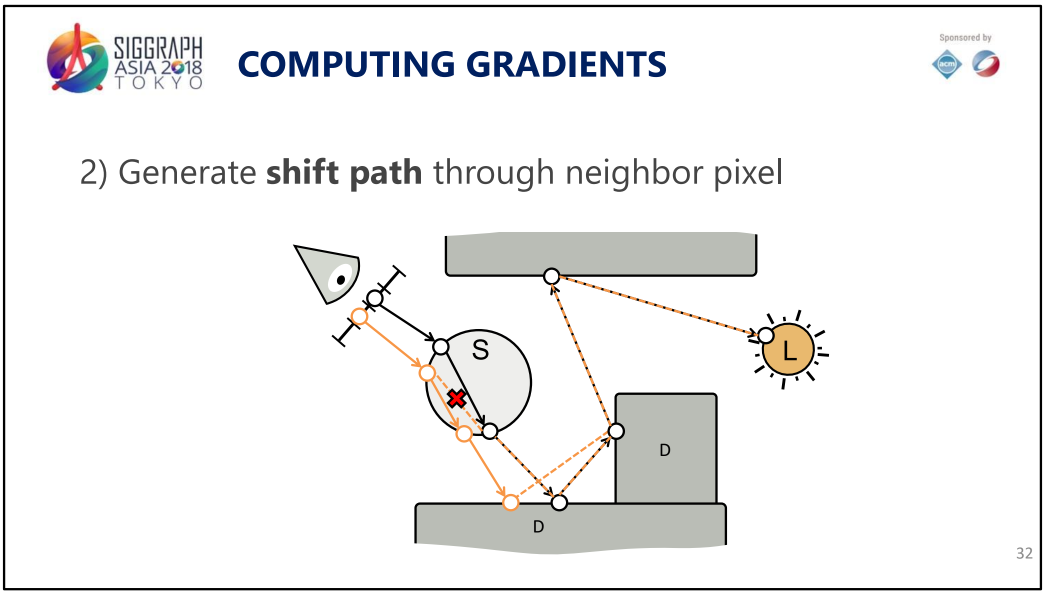
COMPUTING GRADIENTS

1) Generate the **base path**



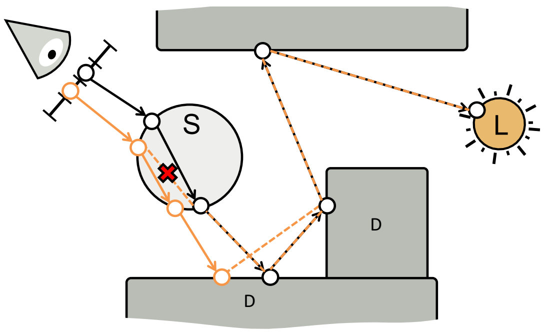
31

Assume a scene as in the diagram. First, we generate a base path in the same way that we do for classical rendering.



COMPUTING GRADIENTS

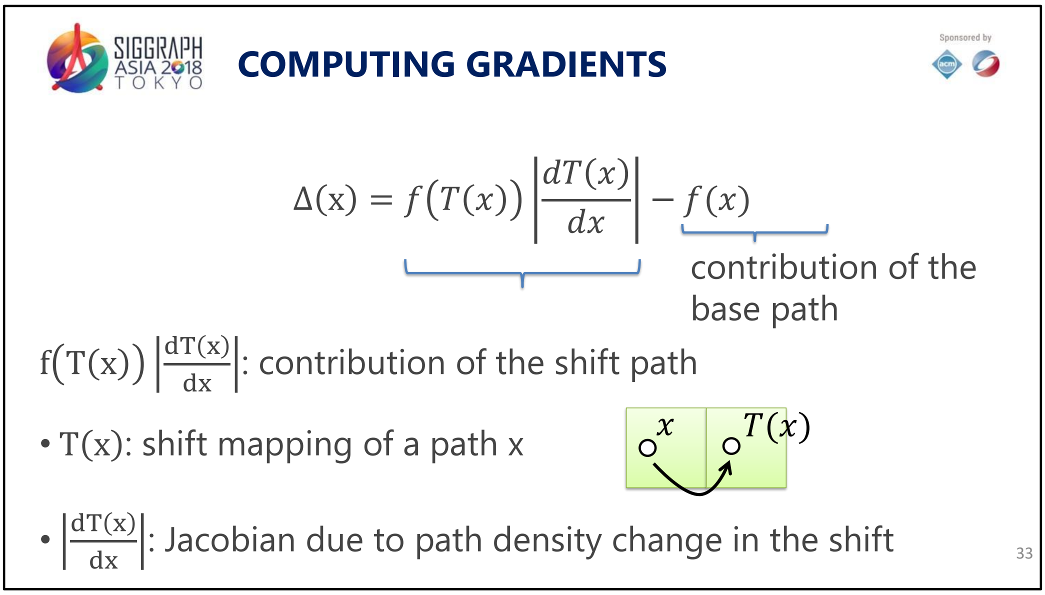
2) Generate **shift path** through neighbor pixel



32

Second, we generate a shift path that passing through neighbor pixel. However, as I said before we want to make the path coherent as much as we can. So we might want to reconnect the shift path to the base path as soon as possible.

Depending on the type of target vertex on the light subpath that we would like to connect to, the reconnection process might vary.
For example, in this diagram, due the specular material on the sphere, we cannot reconnect here. The solution here is to extend the shift path until we reach a diffuse surface to make this reconnection possible.

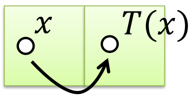


COMPUTING GRADIENTS

$$\Delta(x) = \underbrace{f\left(T(x)\right)\left|\frac{dT(x)}{dx}\right|} - \underbrace{f(x)}_{\text{contribution of the base path}}$$

$f\left(T(x)\right)\left|\frac{dT(x)}{dx}\right|$: contribution of the shift path

- $T(x)$: shift mapping of a path x



- $\left|\frac{dT(x)}{dx}\right|$: Jacobian due to path density change in the shift

33

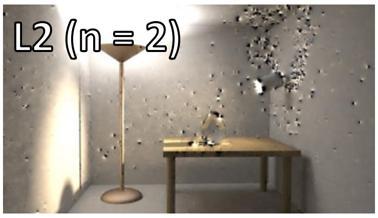
Let’s now explain the terms in details. T(x) takes a path and output a new path that is coherent to the input. The function is deterministic.

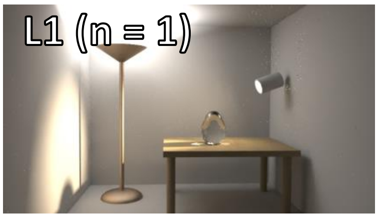
f(x) is the contribution of a path x.

You can see that one additional term here: the Jacobian. This Jacobian accounts for probability change when we shift a base path to create an offset path (the density of the base and offset paths are not the same). This value is important to obtain correct gradient estimation.

POISSON RECONSTRUCTION

$$\min_{L} \underbrace{\left\| \begin{bmatrix} \nabla_{\mathrm{x}} L - G_x \\ \nabla_{\mathrm{y}} L - G_y \end{bmatrix} \right\|_n^n}_{\text{Gradients}} + \underbrace{\lambda \left| L - L_{primal} \right|_n^n}_{\text{Primal (regularization term)}}$$

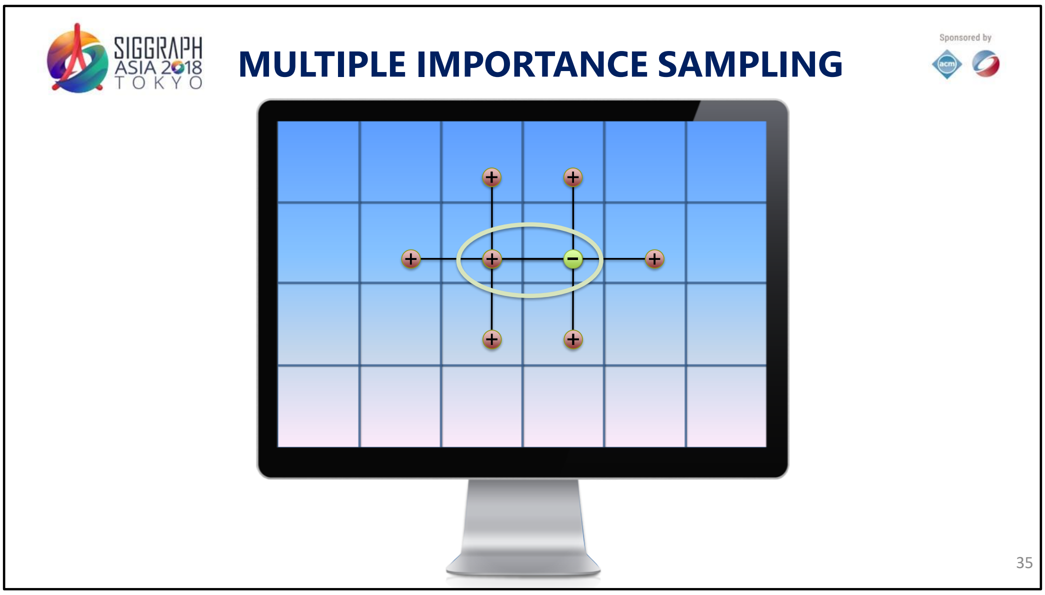




Other reconstruction
[Rousselle et al. 2016]
[Manzi et al. 2016]

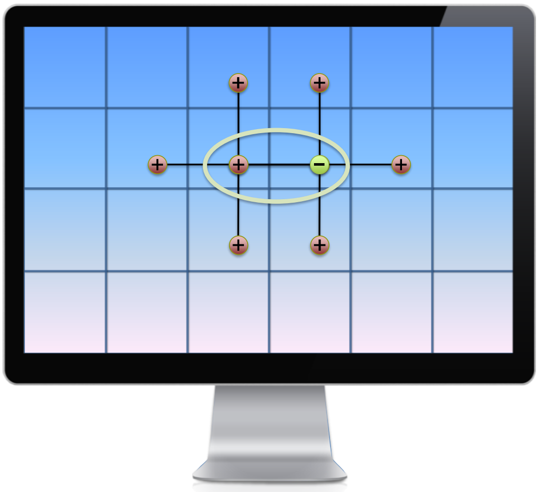
Here is the formula for Poisson reconstruction. The optimization process finds an image that has gradients similar to the input gradients, and at the same time, the image has to be somewhat similar to the throughput (primal) image. The primal image acts as a regularization. This is the default reconstruction used in several papers of gradient-domain rendering so far. Note that we can choose the norm for the cost function. For example, the common situation is to set n = 2 for L2 norm and n = 1 for L1 norm.

For L2 norm, the reconstruction is unbiased but contains some dipole artefacts. For L1 norm, the reconstruction is biased but the artefacts are minimized. If you need more arguments about this, see the seminal paper about gradient-domain rendering [Lehtinen et al., 2013].



MULTIPLE IMPORTANCE SAMPLING

Sponsored by



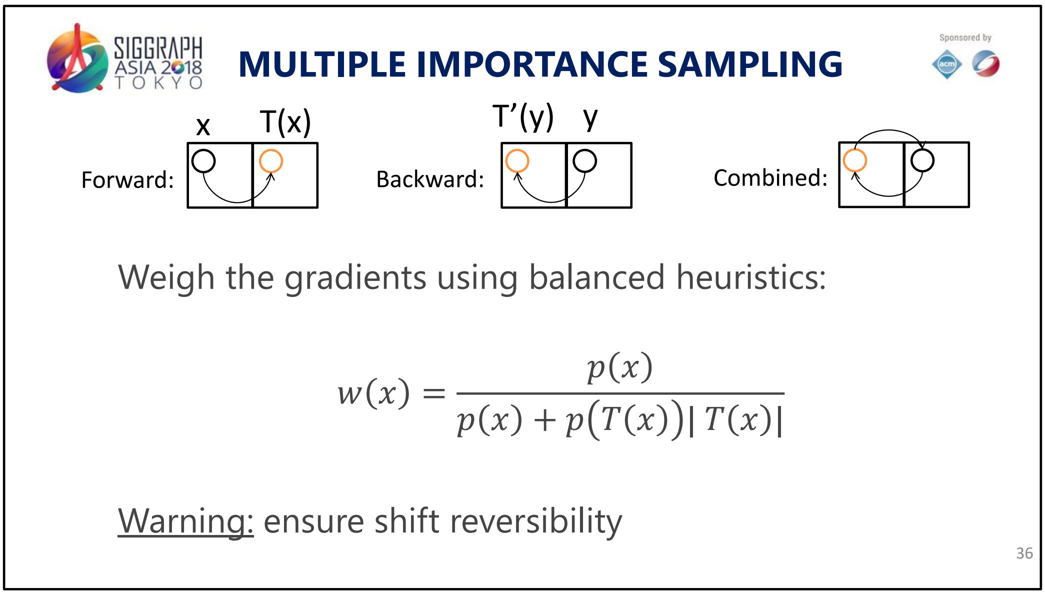
35

Now that we know how to shift paths and do reconstruction, let’s take another look at the previously described sampling process and see if there is any room for improvement.

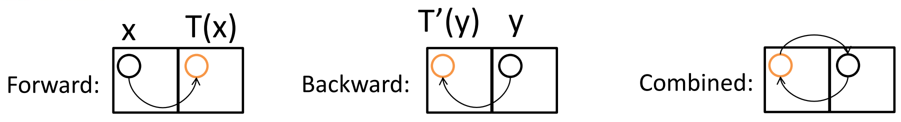
A pair of paths that form a gradient sample can be sampled from two directions:

From the left like we just did, or from the right.

Since the path pair can be sampled from two directions, we can apply multiple importance sampling between them.



MULTIPLE IMPORTANCE SAMPLING



Weigh the gradients using balanced heuristics:

$$w(x) = \frac{p(x)}{p(x) + p\big(T(x)\big)|\,T(x)|}$$

Warning: ensure shift reversibility

36

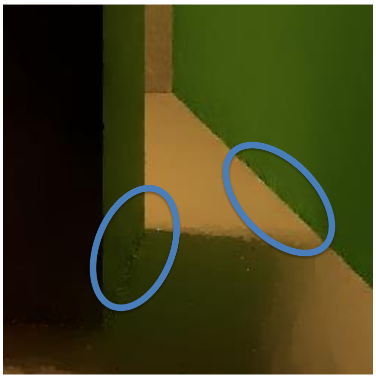
So we can consider two sampling strategies and combine them with multiple importance sampling (MIS).

In general, a path x passing through a pixel could be sampled using the primal-domain path sampling strategy at that pixel, or by first sampling the base path y = T(x) at the pixel's neighbor, and apply inverse shift mapping T' of T to take the base path from the neighbor pixel to the current pixel.

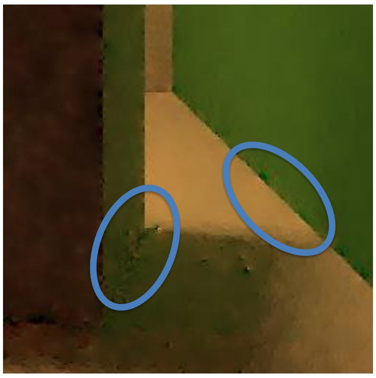
The MIS weight using balanced heuristics can be written as follows. Note that the Jacobian is needed to account for probability change.

MULTIPLE IMPORTANCE SAMPLING

Without MIS
L1 with MIS



With MIS

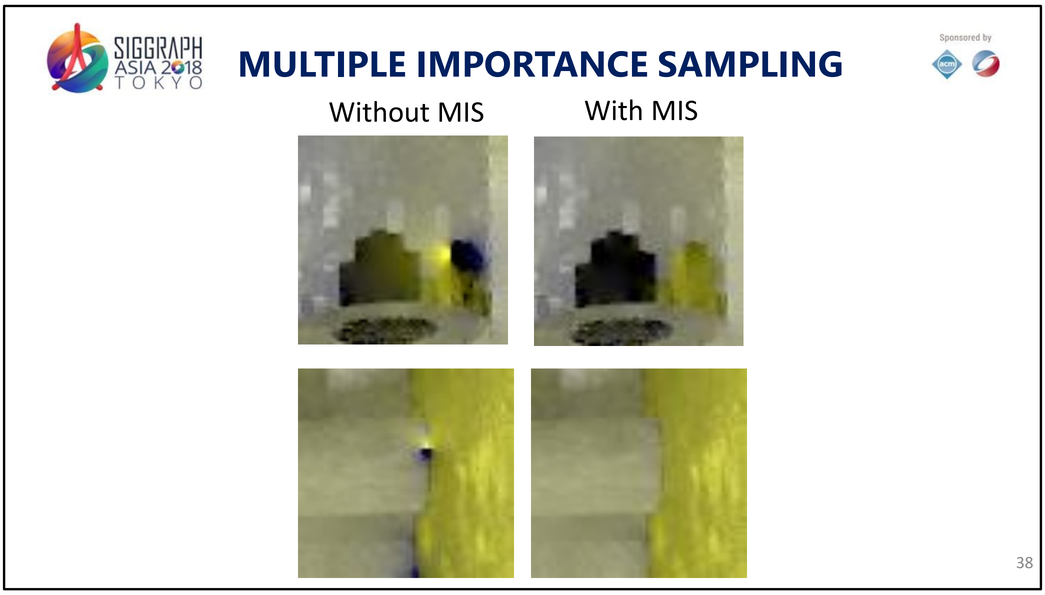


37

Here are some results to demonstrate the effectiveness of multiple importance sampling.

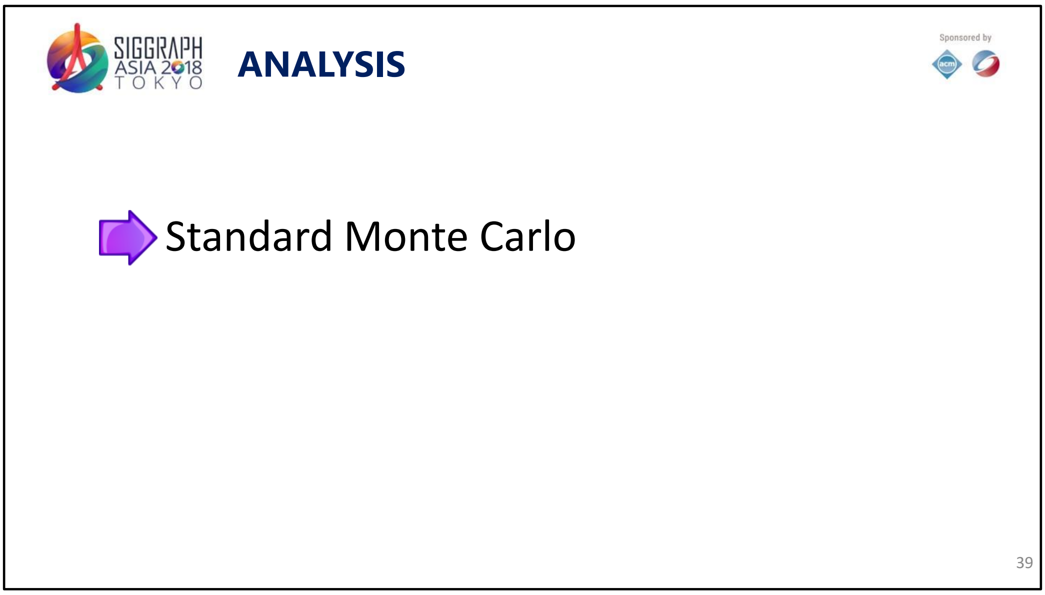
The images are reconstructed with L2 reconstruction. It gets rid of many artifacts appeared in the reconstruction.

To get rid of the rest of the artifacts, we recommend using the slightly biased L1 reconstruction.



MULTIPLE IMPORTANCE SAMPLING

Here is another example.



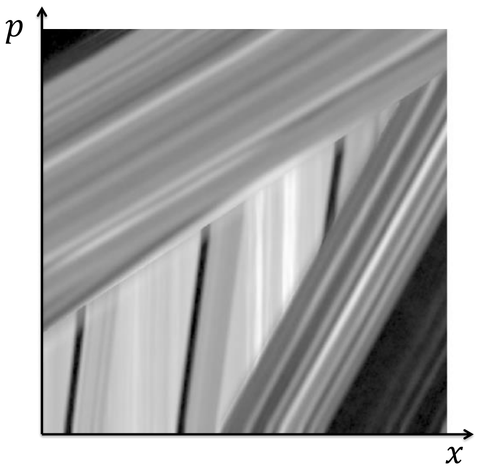
ANALYSIS

Standard Monte Carlo

39

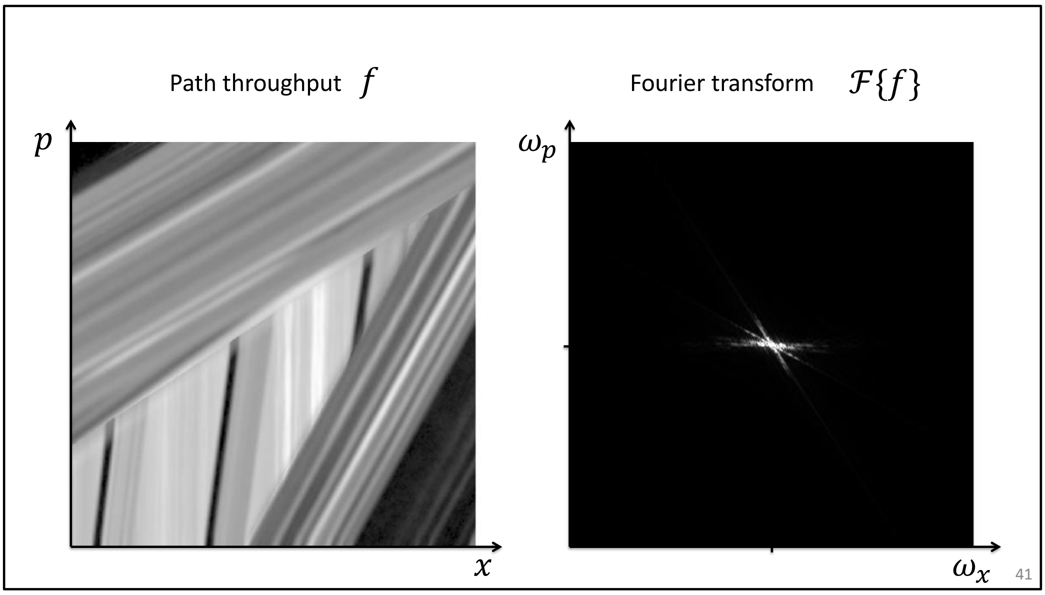
Let’s study some analysis on why gradient-domain rendering actually works by looking at the entire sampling and reconstruction process. The tool we use here is frequency analysis. Let’s begin by reminding ourselves of what happens in standard Monte Carlo random sampling.

Path throughput f



40

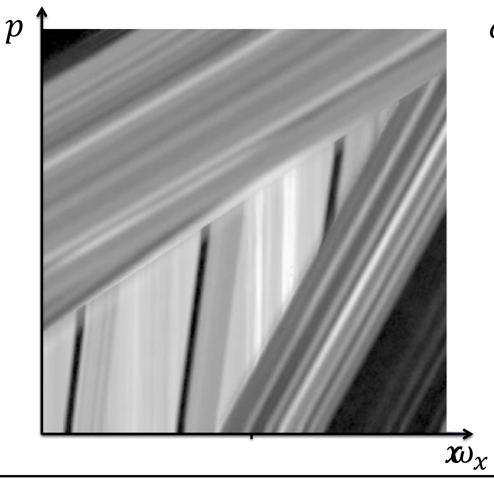
For this, let’s observe a 2D setup, with the image along the x-axis, and a path dimension along the p-axis. f is the path throughput function. We will obtain the final image by integrating over p.



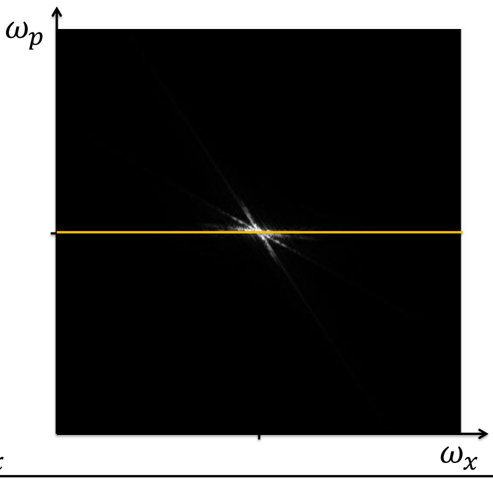
On the right, we see the power spectrum of the throughput function.

Path throughput f

Slice ($\omega_p = 0$)

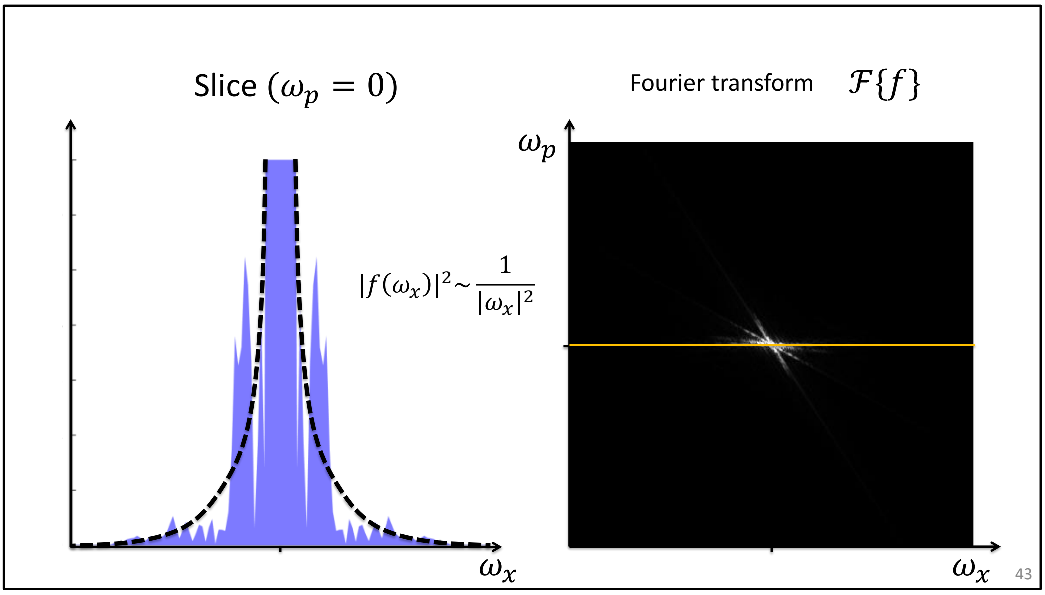


Fourier transform $\mathcal{F}\{f\}$



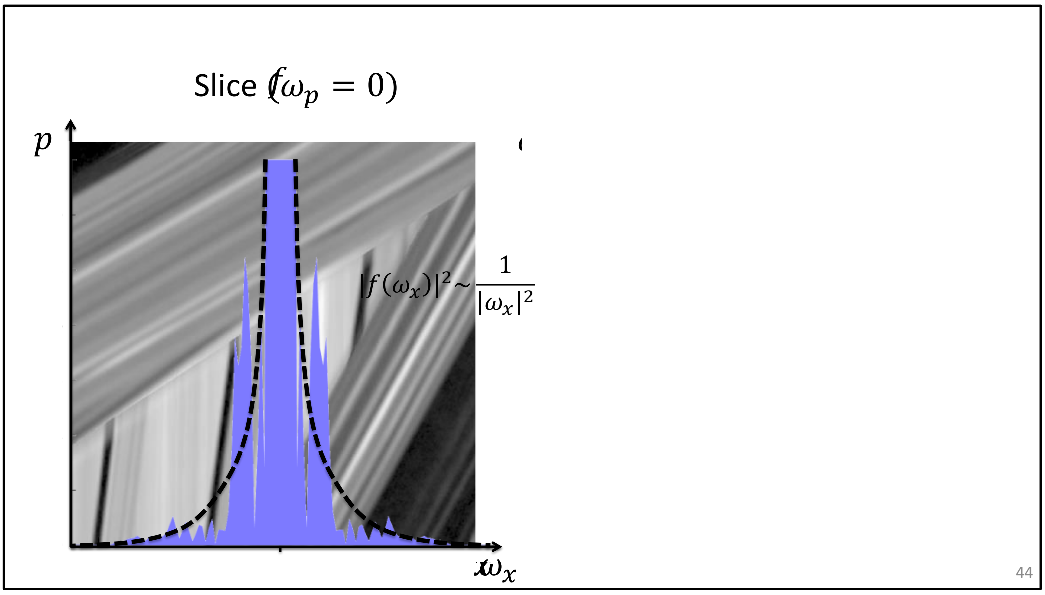
42

To get a better view, let’s concentrate on this horizontal slice.



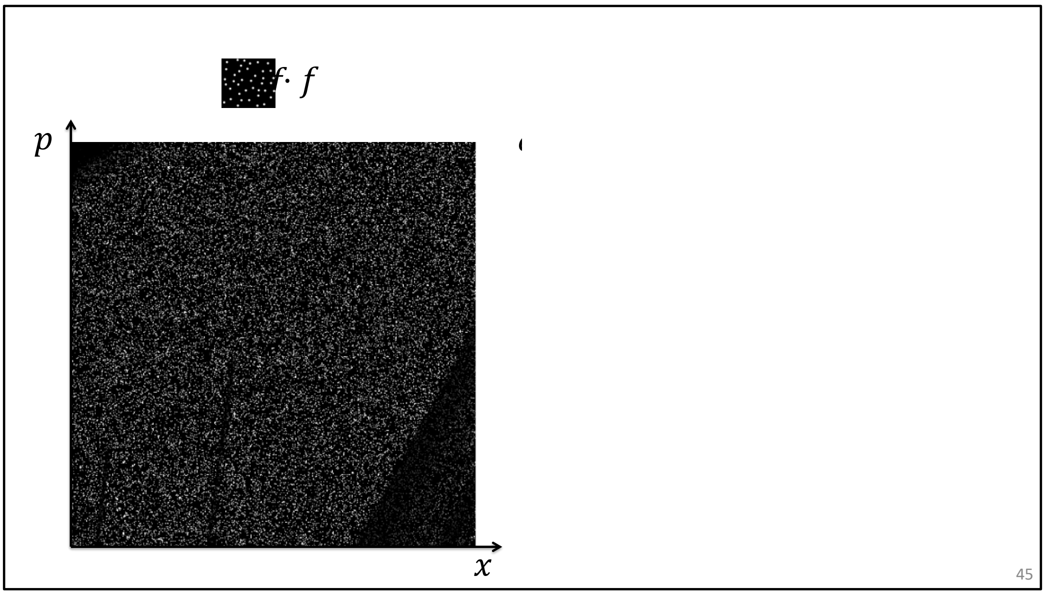
We see that the shape resembles one over frequency squared. This is because the final image, just as the throughput function overall, has discontinuities and resembles a natural image.

The fact that the energy is heavily concentrated near the low frequencies is going to be crucial for us later on.



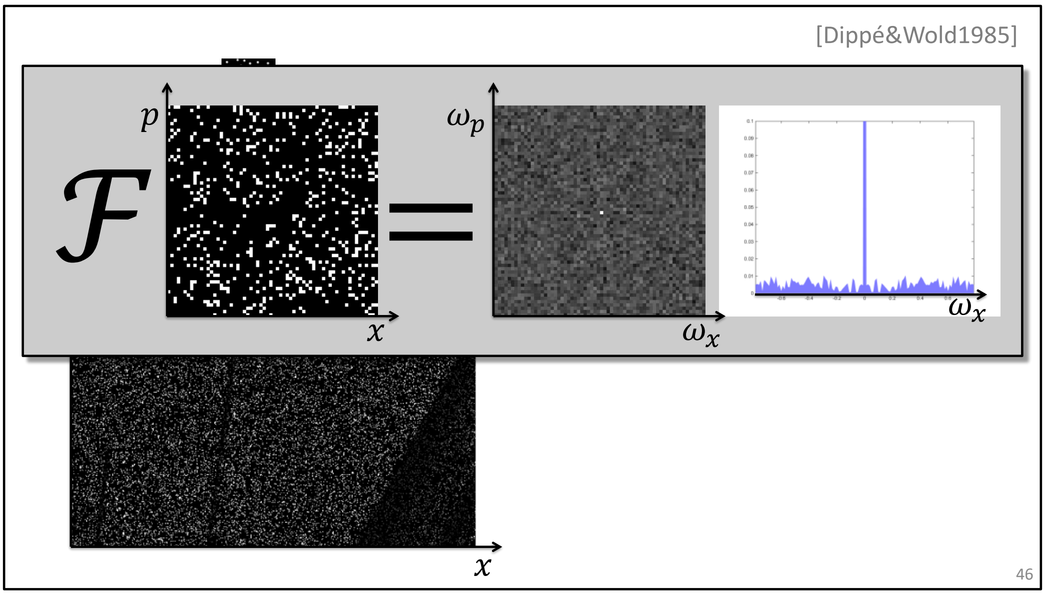
Let us now analyze Monte Carlo sampling.

In the primal domain, random sampling multiplies the signal by a forest of random Dirac impulses.



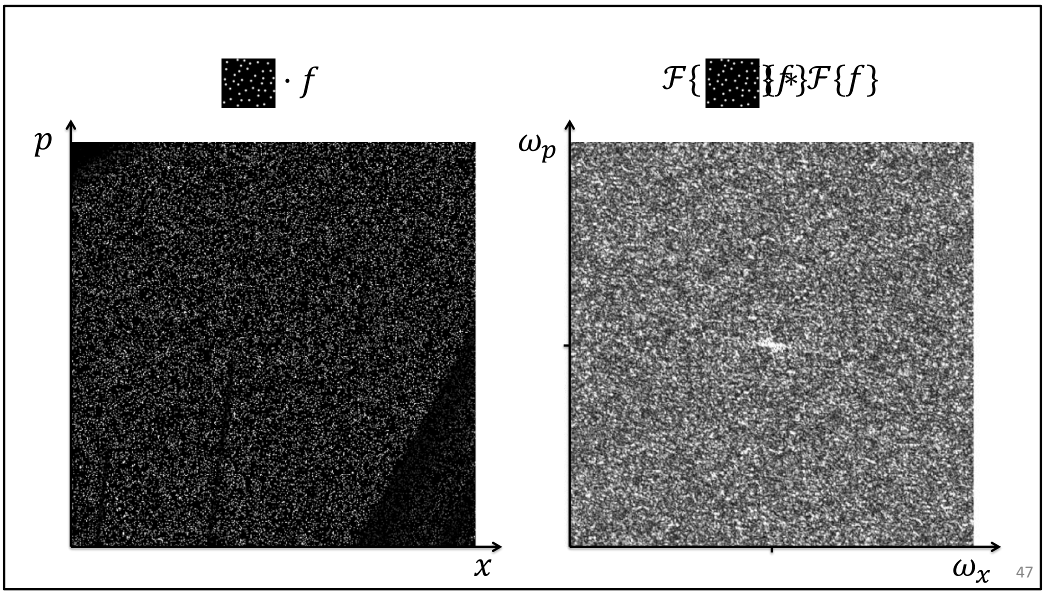
Each impulse corresponds to the location of one sample.

So what does this look like in the frequency domain?



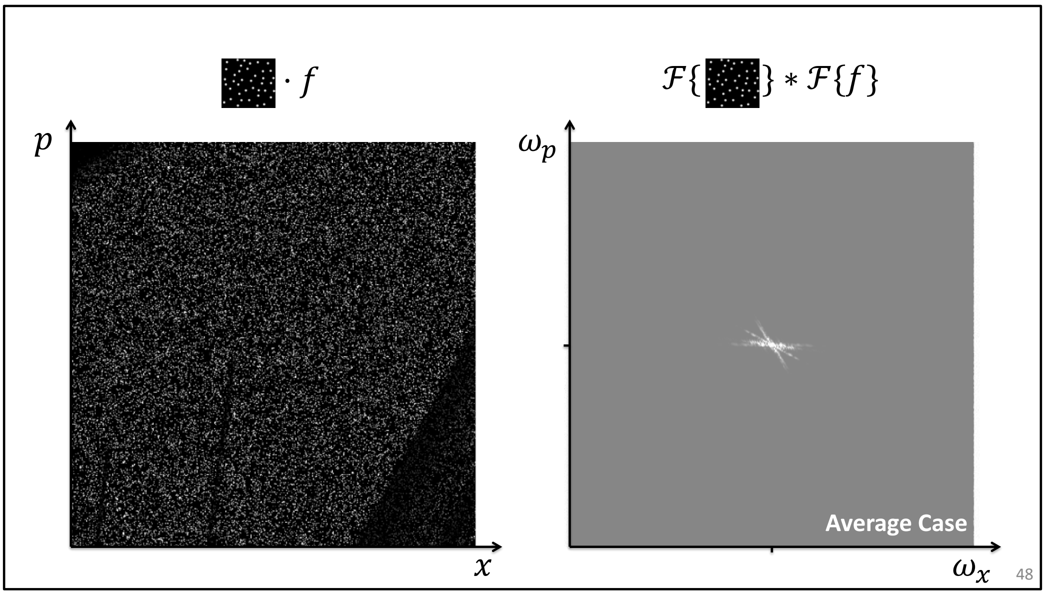
This is the Fourier transform of the random samples. It has a strong peak at the DC, and uniform random noise everywhere else. The slice on the right shows this more clearly.

As we know, multiplication in the primal corresponds to a convolution in the frequency domain.

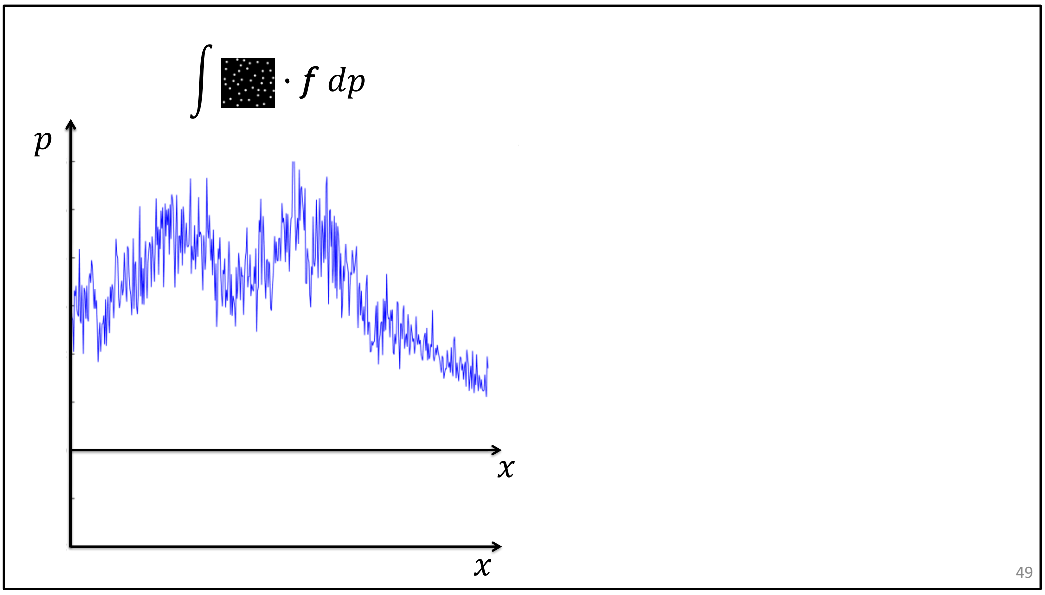


When we do this, we see that this spreads the energy uniformly over the path space as white noise.

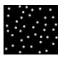
Because we are dealing with random sampling, we are interested in the expected error.

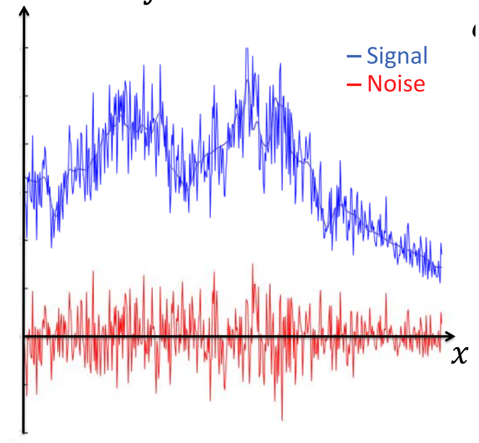


And it is well known that the magnitude of the noise is proportional to the total energy of the signal. This shows up as the gray background in the right.



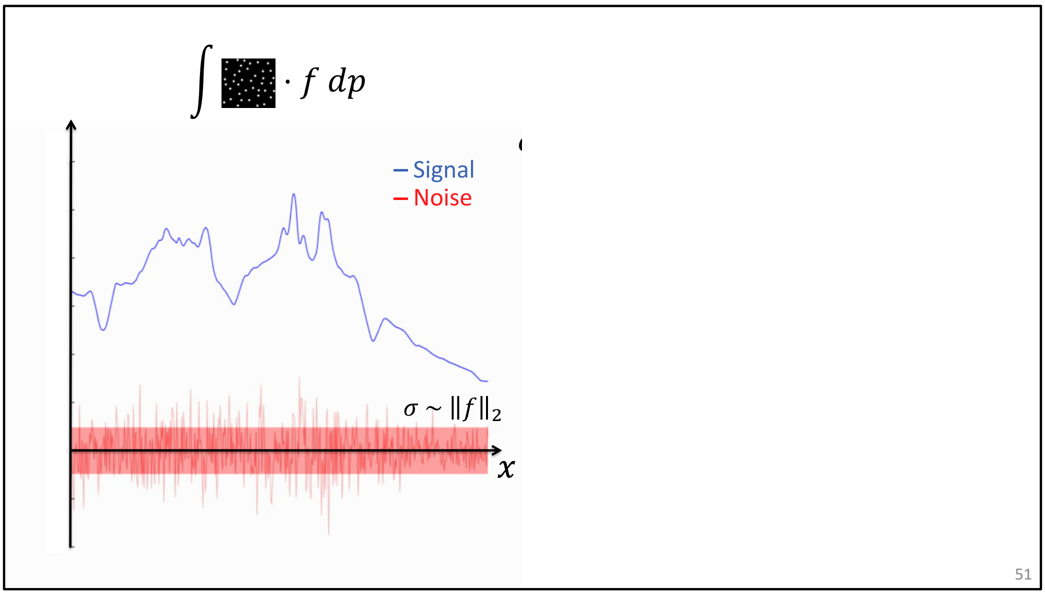
To get the final image, we accumulate the random samples in each pixel. We end up with an estimate of the image with some amount of sampling noise.

$\int$  $\cdot f \, dp$

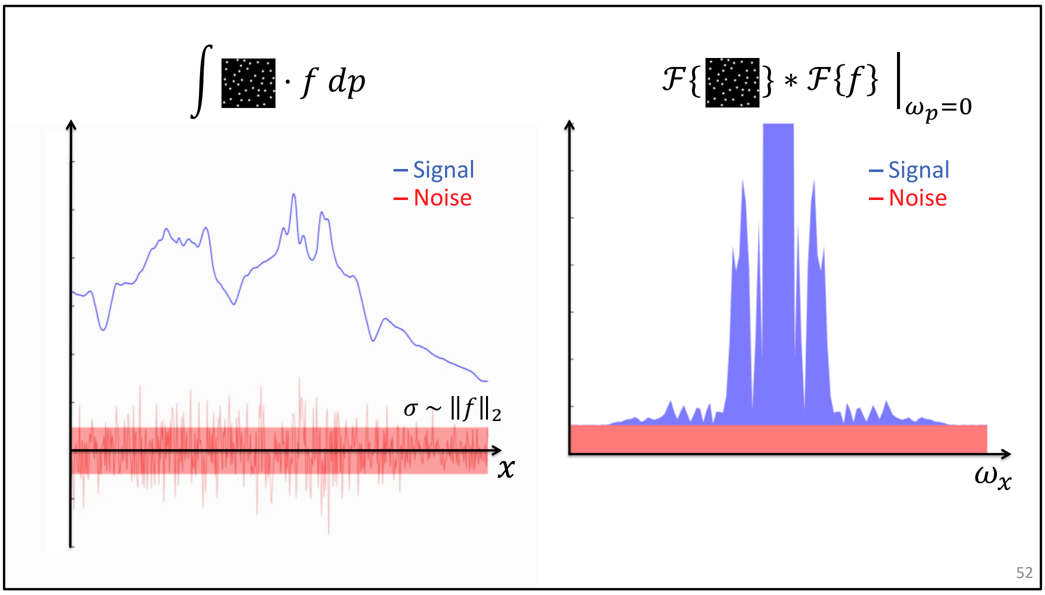


50

This is just white noise with standard deviation proportional to the energy of the signal.

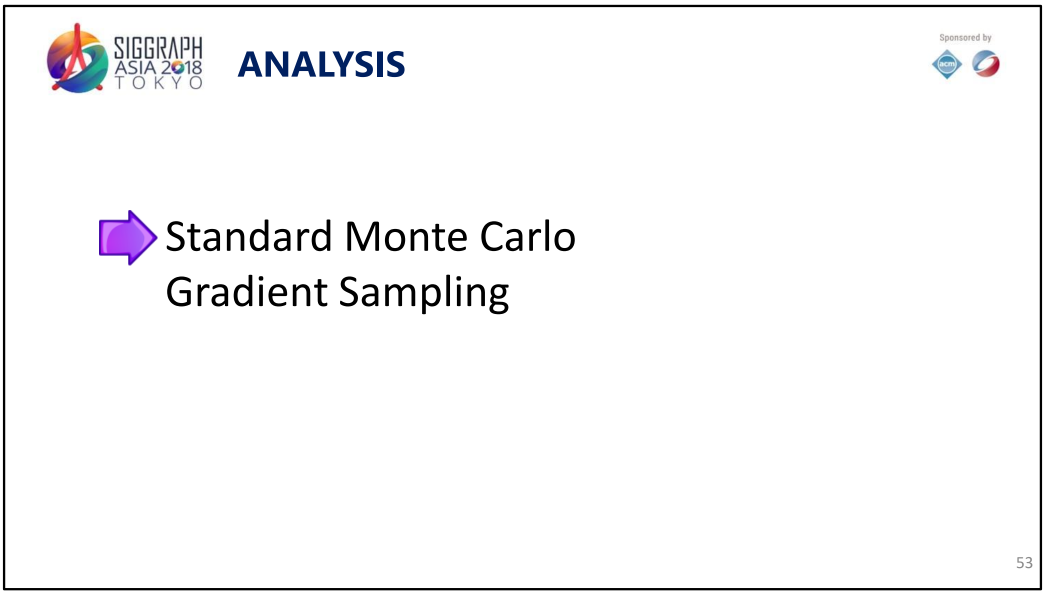


The standard deviation is proportional to the energy of the signal.



The noise shows up in the frequency domain as the red box on the right, which is a slice through the gray background from before.

Here, we have derived the well known result that sampling noise is white and its magnitude is proportional to the signal’s energy.

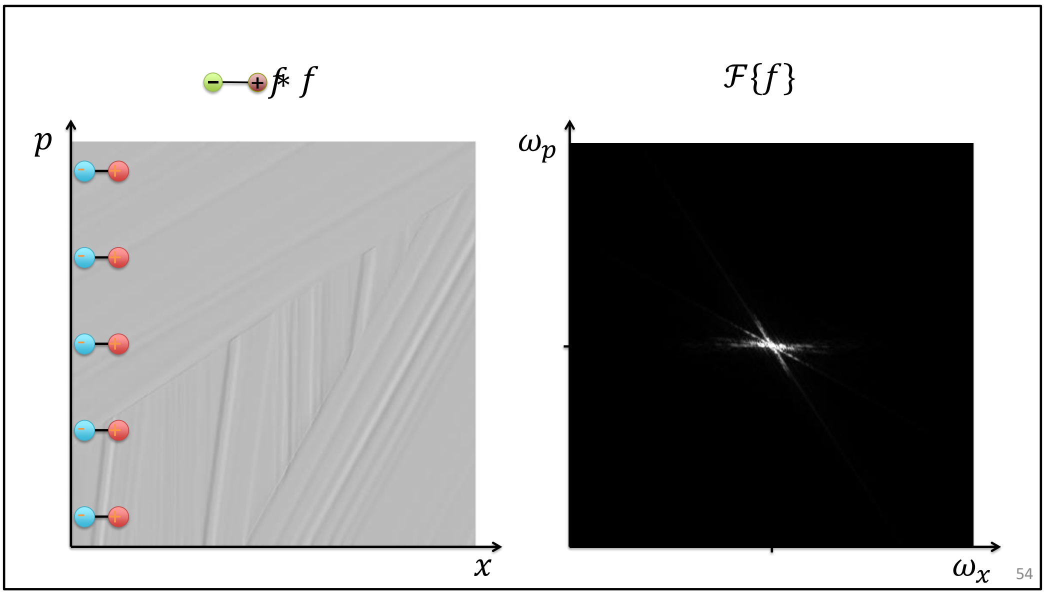


ANALYSIS

Standard Monte Carlo Gradient Sampling

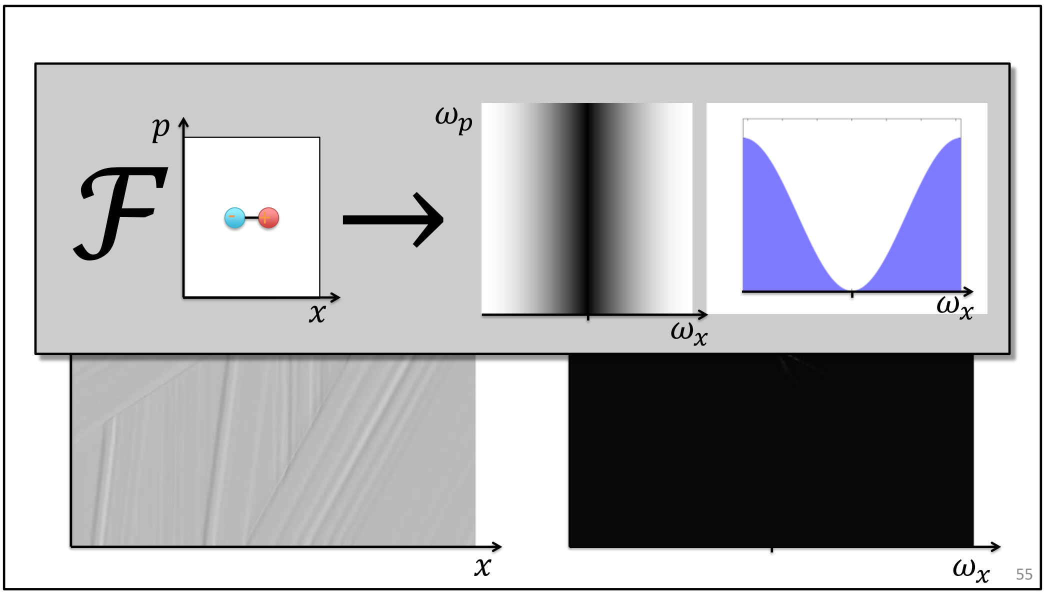
53

With this knowledge equipped, let’s see what happens when we sample **gradients**.



Taking finite differences corresponds to a convolution with a kernel with a positive and a negative Dirac.

As we know, convolution corresponds to multiplying the spectra.

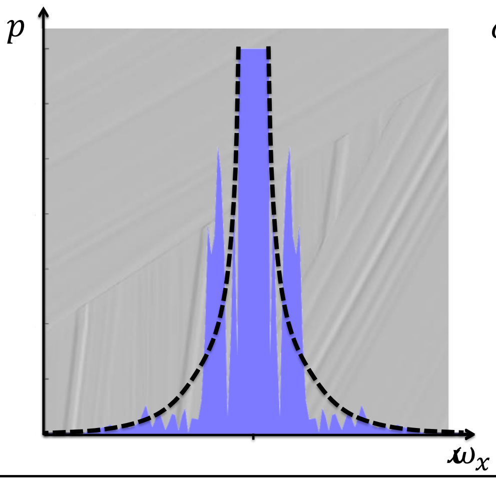


The power spectrum of the finite difference kernel is a squared sine, with the period of the pixel pitch.

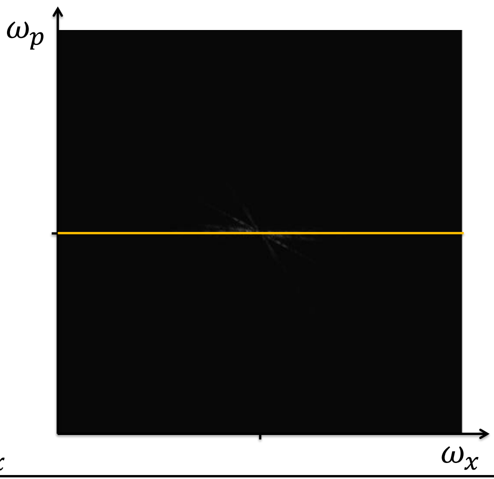
This then multiplies the power spectrum of the signal.

To see clearly what this does, let us again concentrate on this slice.

$\text{Slice}(\omega_p = 0)$ $\oplus\!-\!\oplus * f$



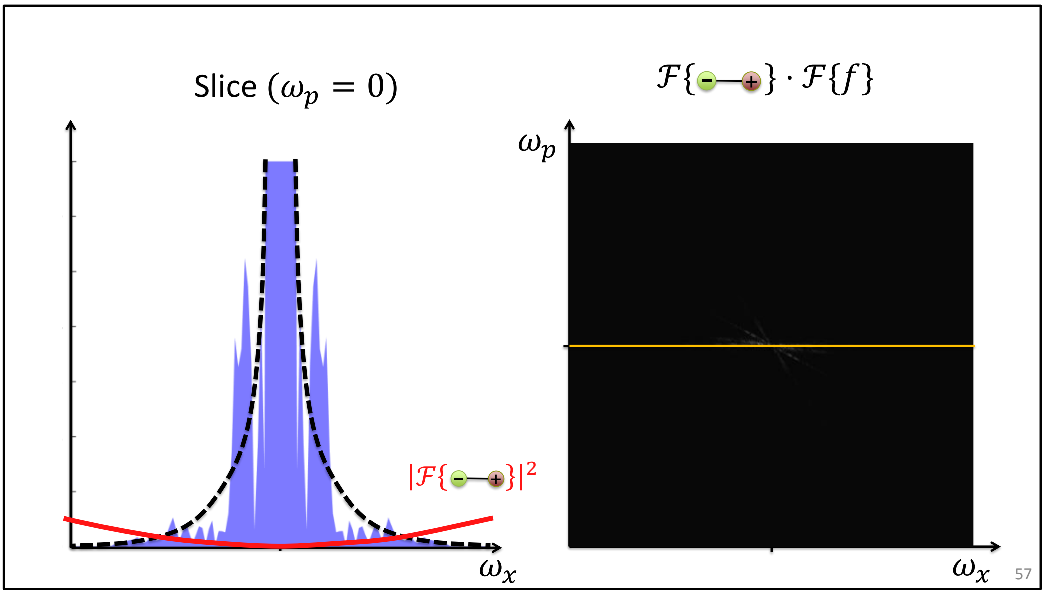
$\mathcal{F}\{\ominus\!-\!\oplus\} \cdot \mathcal{F}\{f\}$



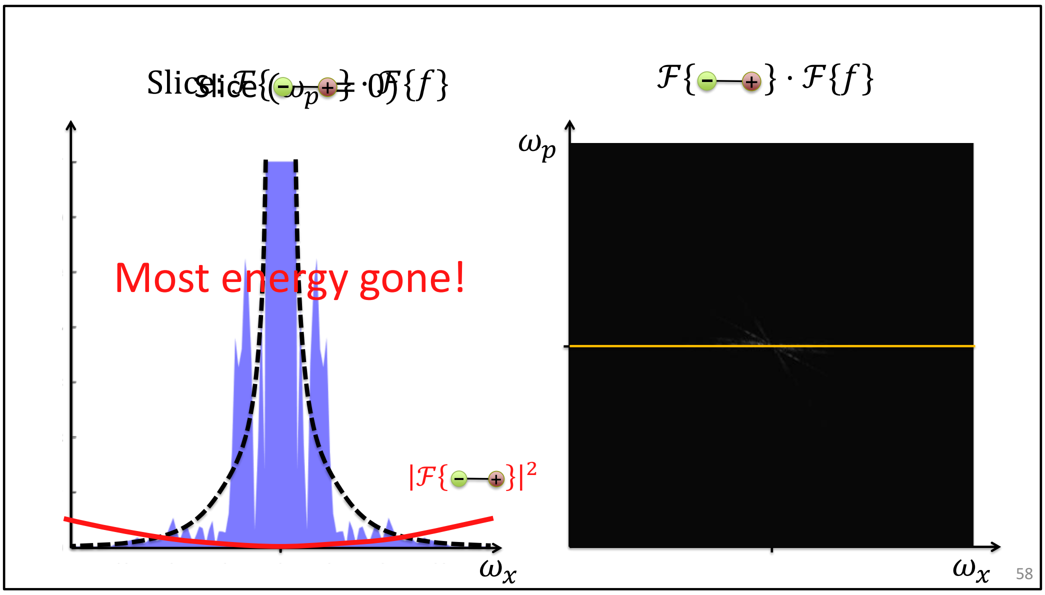
56

Let us again concentrate on this slice.

This is where things get interesting.

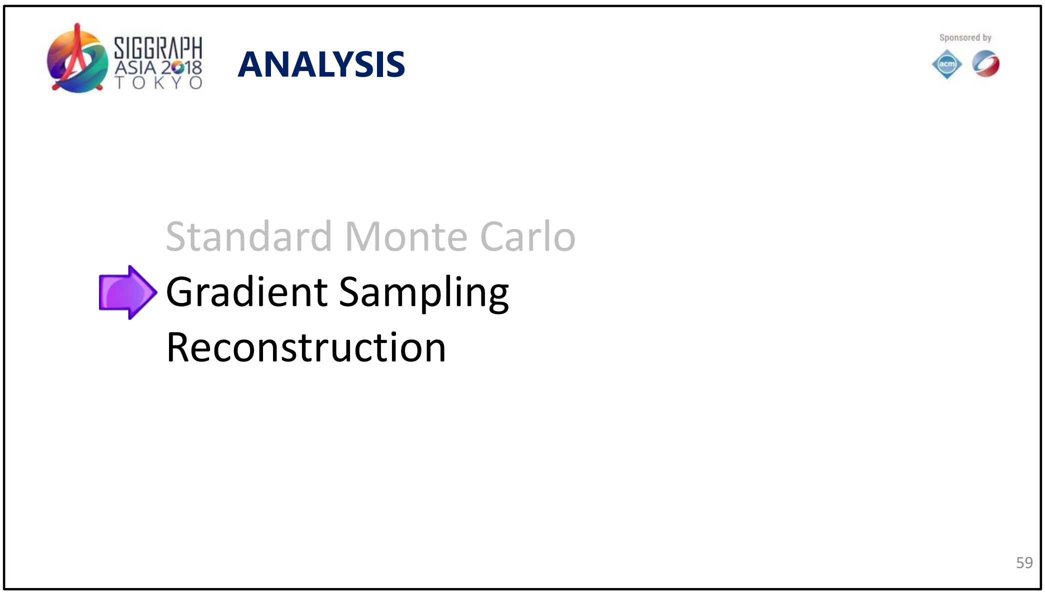


The squared sine, **in red, has low values** near the origin where *most of the energy of the throughput signal is*. Now, clearly, when we multiply these, most of that energy will go away!



We're left with the gradient signal whose magnitude is much lower than the original signal.

Crucially, as the sampling noise is proportional to the energy, this means sampled gradients will have much less noise than the sampled signal itself.



ANALYSIS

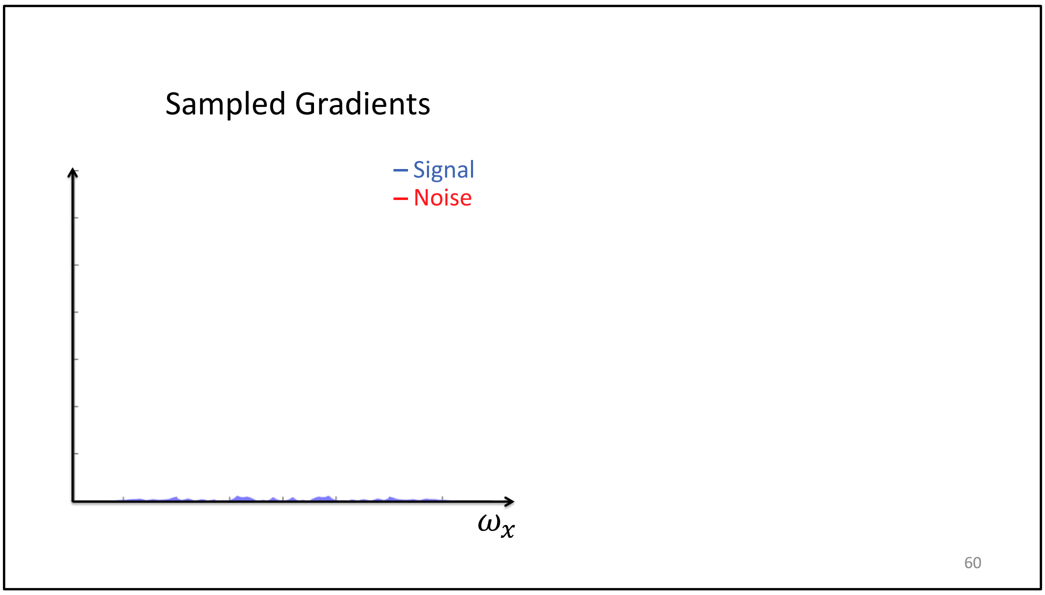
Standard Monte Carlo

➡ Gradient Sampling

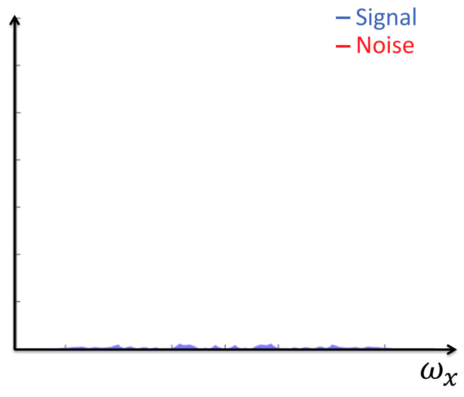
Reconstruction

59

However, we’re not really interested in the gradients, but the image we reconstruct using them. So let’s go on and see what happens when we do Poisson reconstruction.

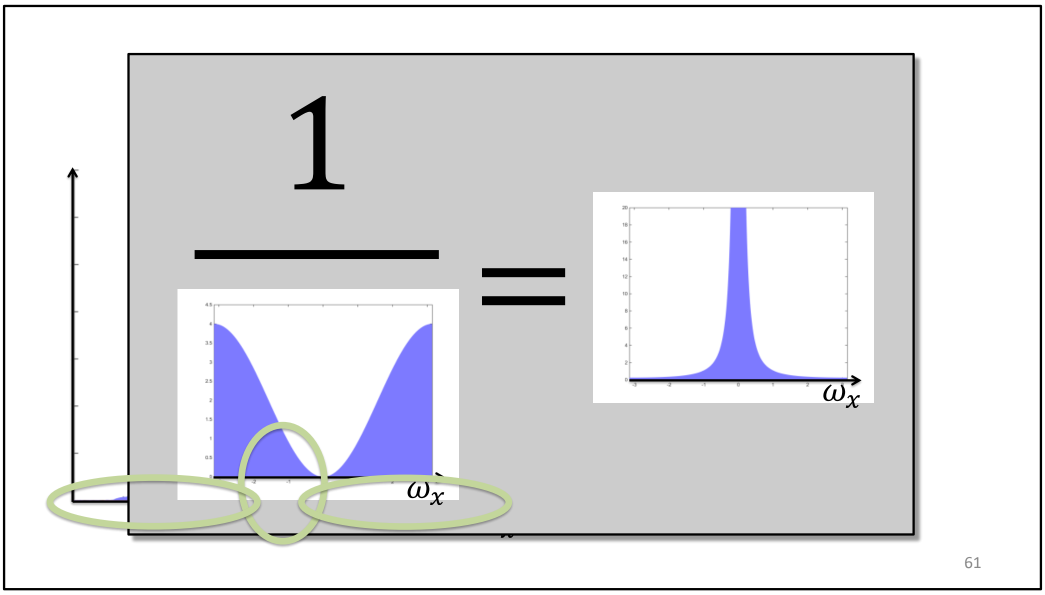


Sampled Gradients



60

Here is the spectrum of the sampled gradients again. Let’s zoom in so that we see something.

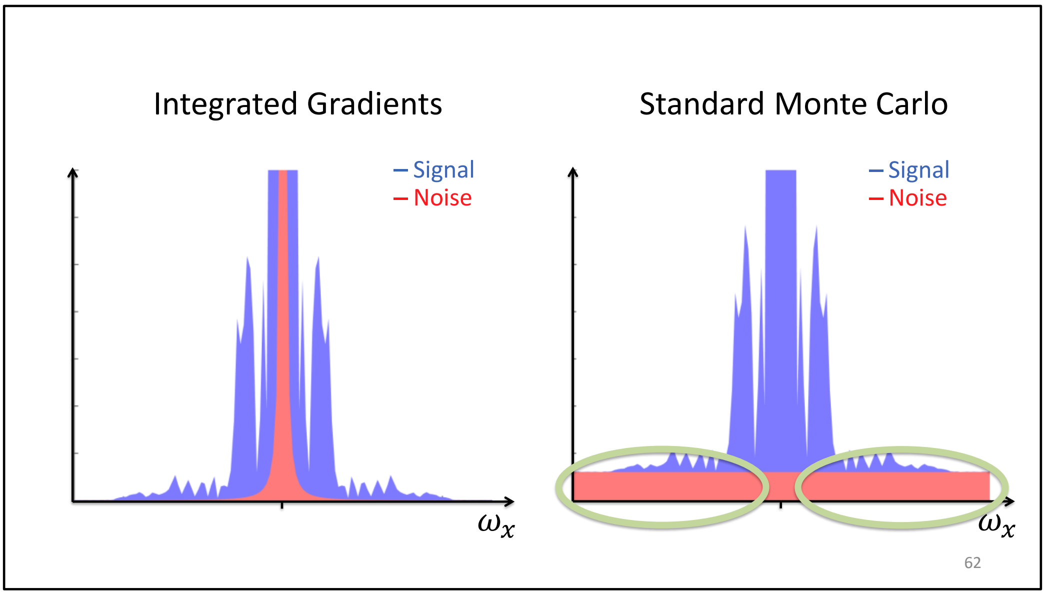


To reconstruct the final image, we invert the gradient operator, by dividing by its power spectrum, the squared sine.

When we do this, the noise in the low frequencies blows up completely, which makes intuitive sense, as the gradients do not contain information about the DC.

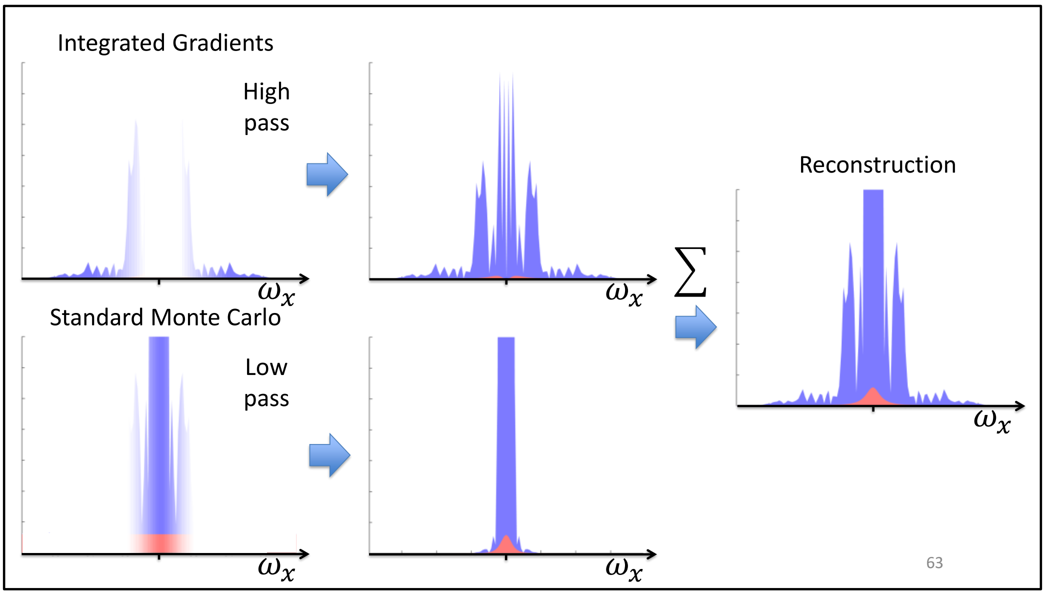
However, the high frequencies are excellent, with much lower error than before.

Compare this to standard sampling on the right.



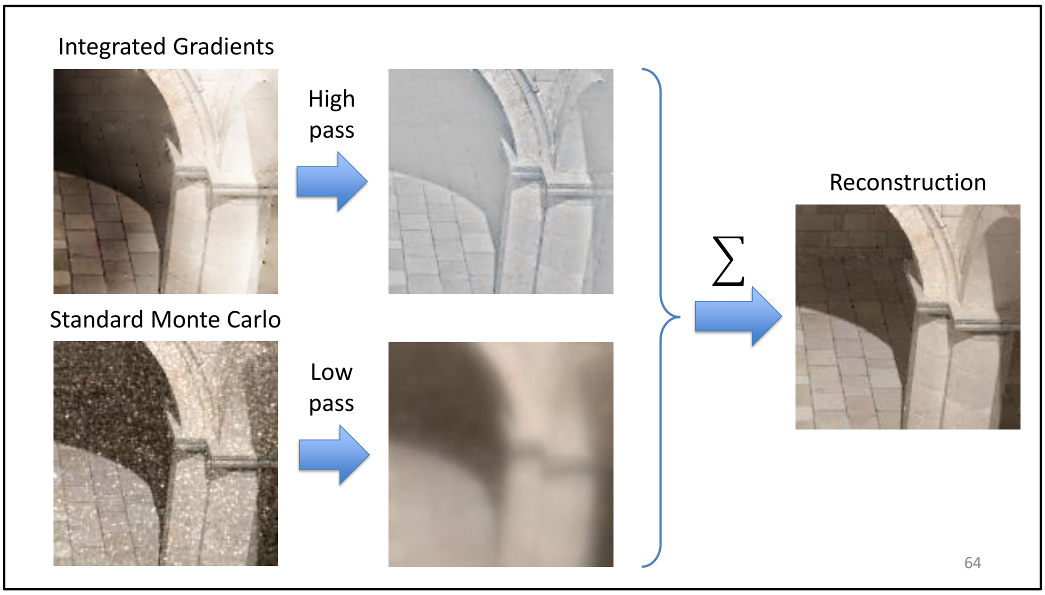
... on the right. Its level of noise is much worse in the high frequencies, whereas the low frequencies are represented better.

Given this, it's not a stretch to figure out that we should take the high frequencies from the integrated gradients, and the low frequencies from standard sampling.



This simply means applying a high pass filter to the gradient reconstruction, and a low pass filter to the regular sampling.

Summing these gives the final reconstruction, which has the best of both worlds, and has a much lower level of overall noise than either of the inputs alone.

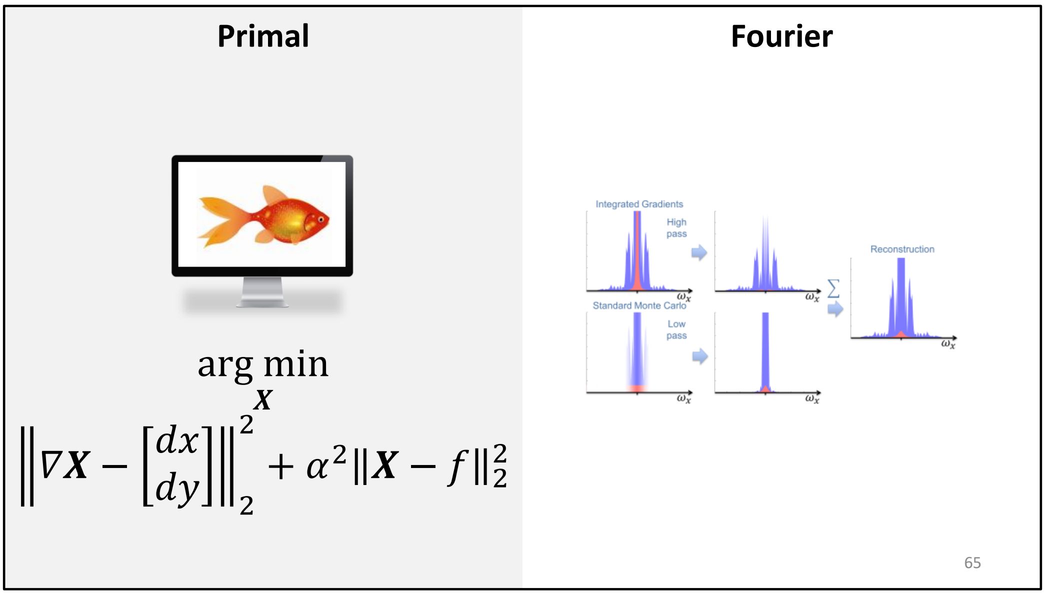


Concretely, it looks like this.

The standard sampling has lots of noise, whereas the gradient reconstruction has good detail but the low frequencies are badly wrong.

High-passing gets rid of the offending low frequencies above, and low-passing gets rid of the noise below.

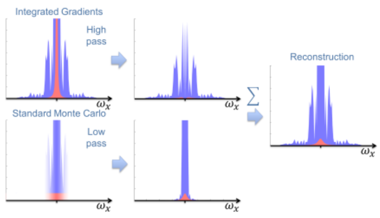
Their sum is much nicer than either alone.



Primal

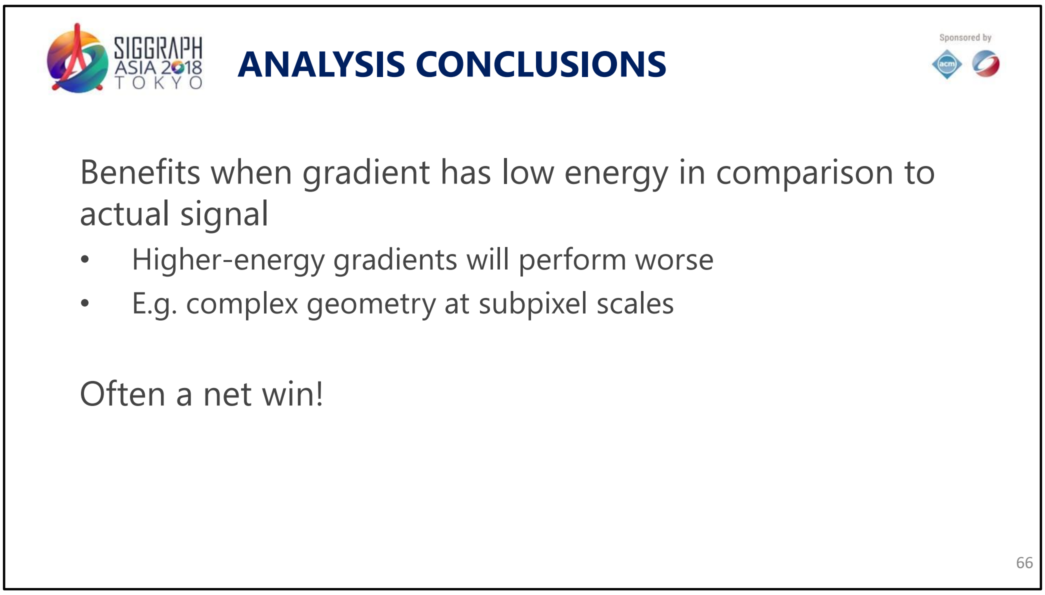
Fourier

$$\underset{\boldsymbol{X}}{\arg\min} \left\| \nabla \boldsymbol{X} - \begin{bmatrix} dx \\ dy \end{bmatrix} \right\|_2^2 + \alpha^2 \| \boldsymbol{X} - f \|_2^2$$



65

As an aside, it turns out solving the Screened Poisson problem in the primal domain does exactly what I just described.



ANALYSIS CONCLUSIONS

Benefits when gradient has low energy in comparison to actual signal

- Higher-energy gradients will perform worse
- E.g. complex geometry at subpixel scales

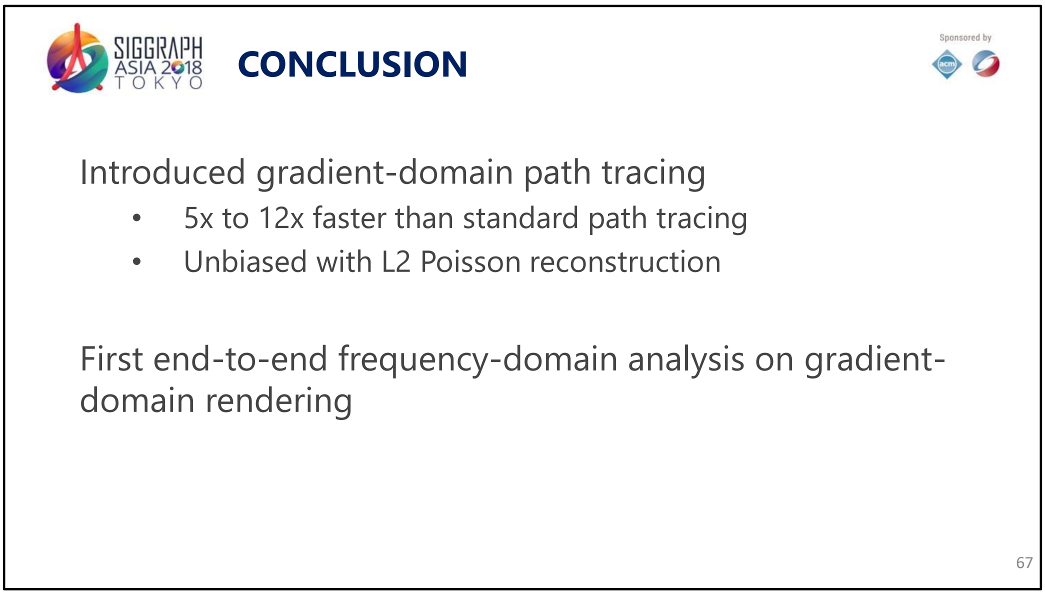
Often a net win!

66

So what does this analysis tell us?

To get benefits out of gradient sampling, the throughput function needs to have most of its energy at the low frequencies. In the paper we show that it’s precisely the ratio of the energies of the signal and its gradient that determines efficiency. The less energy in the gradients, the better.

This tells us that conversely high-energy gradients will perform worse. This happens, for instance, if the image contains really complex geometry like grass or something like that **at the subpixel level**. But a lot of the time, when your scene is modeled in reasonable detail, gradients do provide a benefit.



CONCLUSION

Introduced gradient-domain path tracing

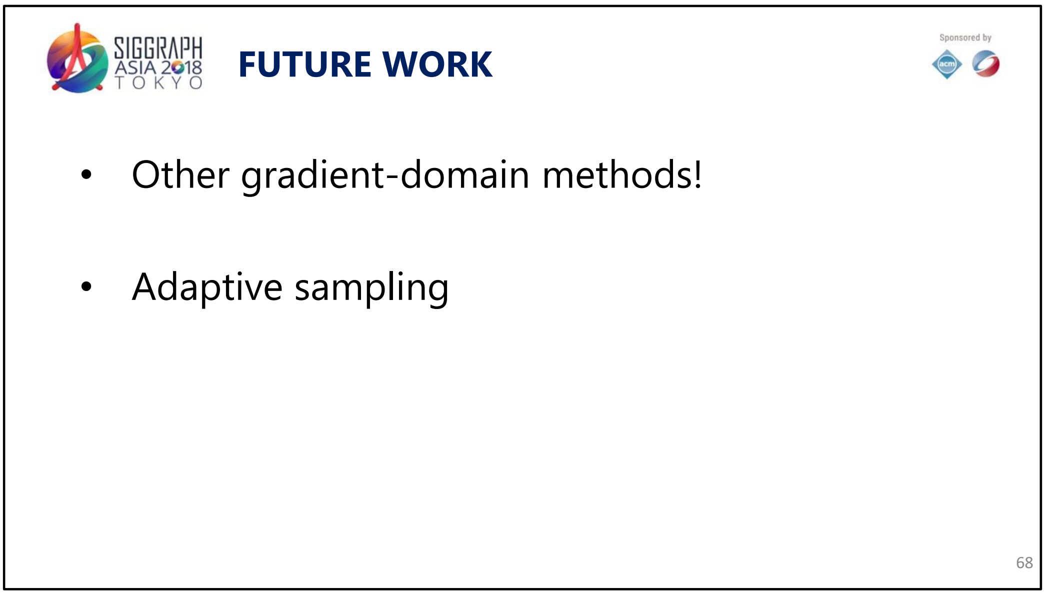
- 5x to 12x faster than standard path tracing
- Unbiased with L2 Poisson reconstruction

First end-to-end frequency-domain analysis on gradient-domain rendering

67

We introduced gradient-domain path tracing, a relatively simple modification of standard path tracing that yields a significant benefit in image quality.

To see the reasons for its initially surprising efficiency, we also presented the first end-to-end frequency-domain analysis about gradient-domain rendering.



FUTURE WORK

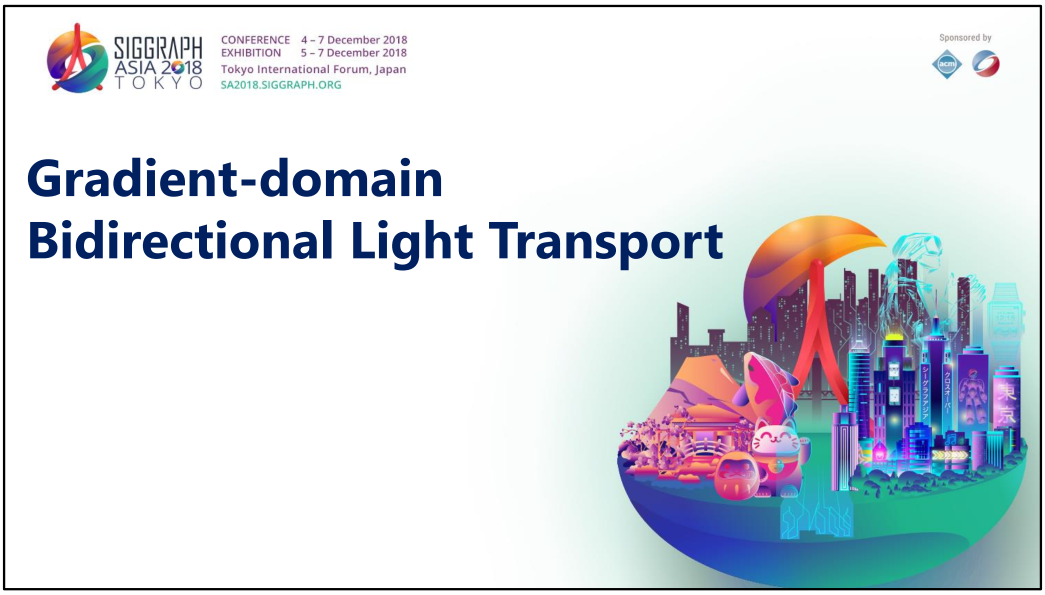
- Other gradient-domain methods!
- Adaptive sampling

68

As for future work, we’d like to dynamically analyze the properties of the sampled signals to adaptively place more samples where they matter the most.

Also, we believe that many common methods could be improved by introducing gradient-domain versions of them. We will explore them in the next section.

4.4 Gradient-domain bidirectional light transport

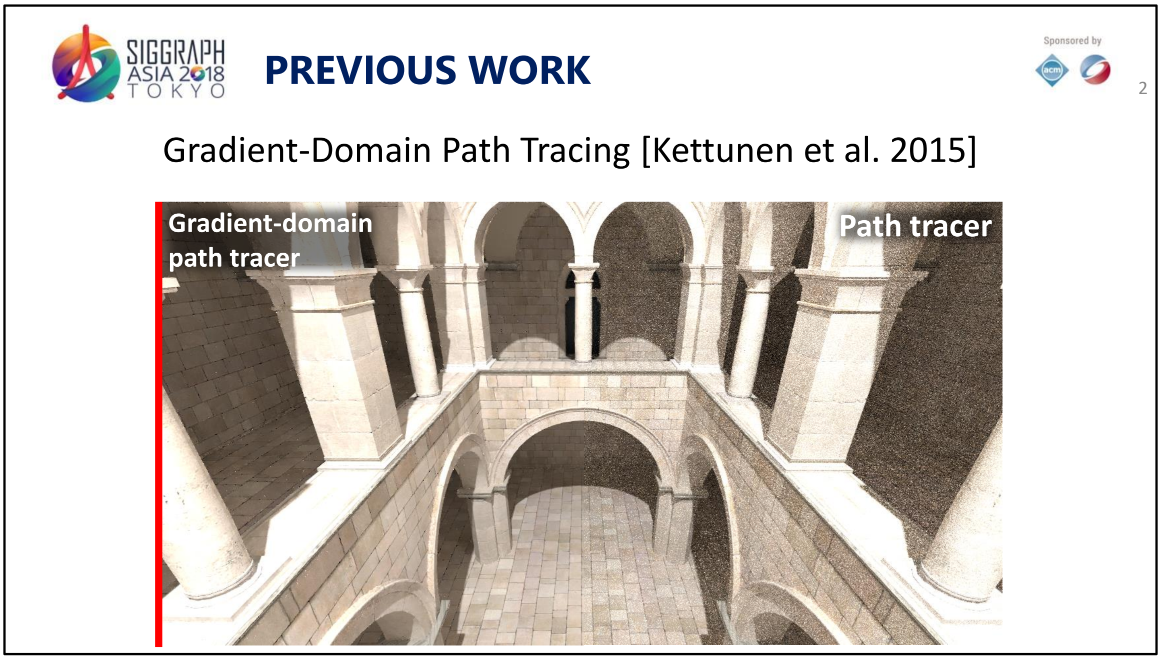


SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

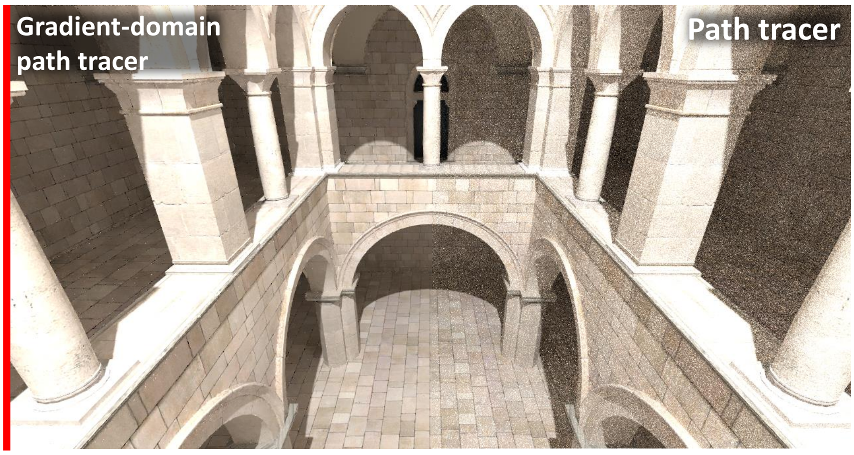
Sponsored by

Gradient-domain Bidirectional Light Transport



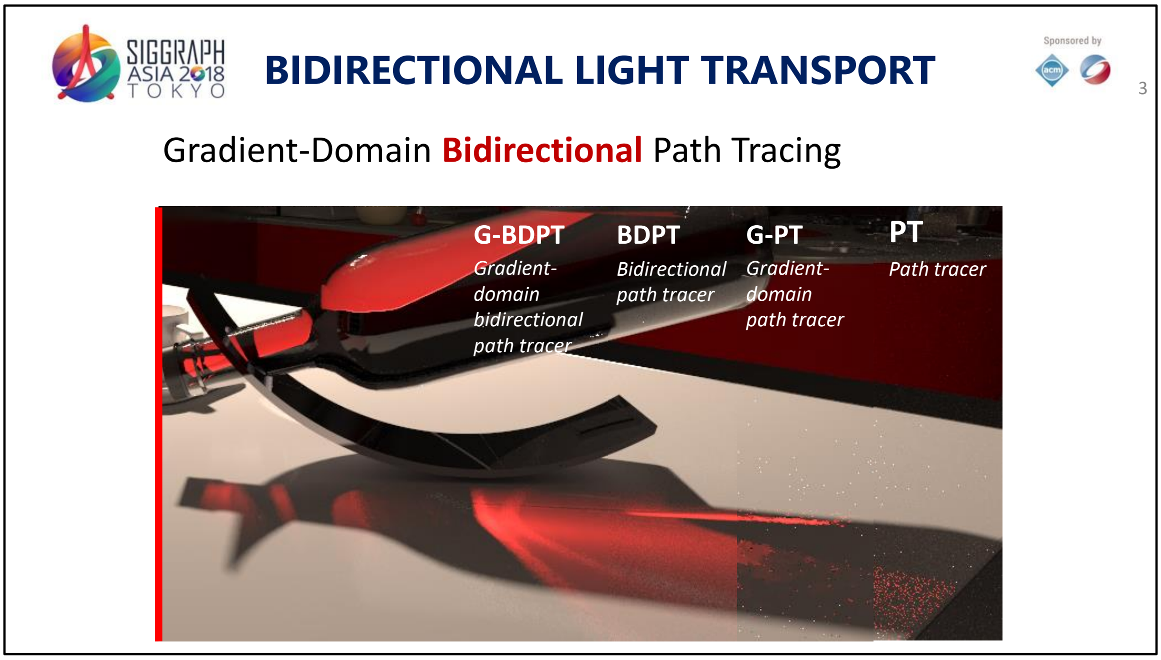
PREVIOUS WORK

Gradient-Domain Path Tracing [Kettunen et al. 2015]



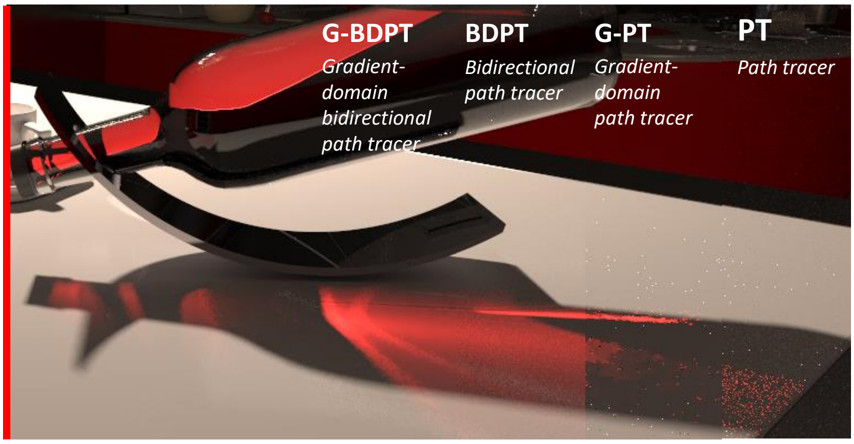
Gradient-domain rendering has been shown to work nicely with path tracing.
To recap, here an example of an image generated with a path tracer compared at equal time with the same scene generated with a gradient-domain path tracer. As you can see the noise reduction is huge.

Our hope is that by adapting the algorithm to a more broadly used light transport algorithms, the usefulness of gradient-domain rendering would increase significantly. In this section, we are going to try this idea with bidirectional path tracing and photon mapping.



BIDIRECTIONAL LIGHT TRANSPORT

Gradient-Domain **Bidirectional** Path Tracing

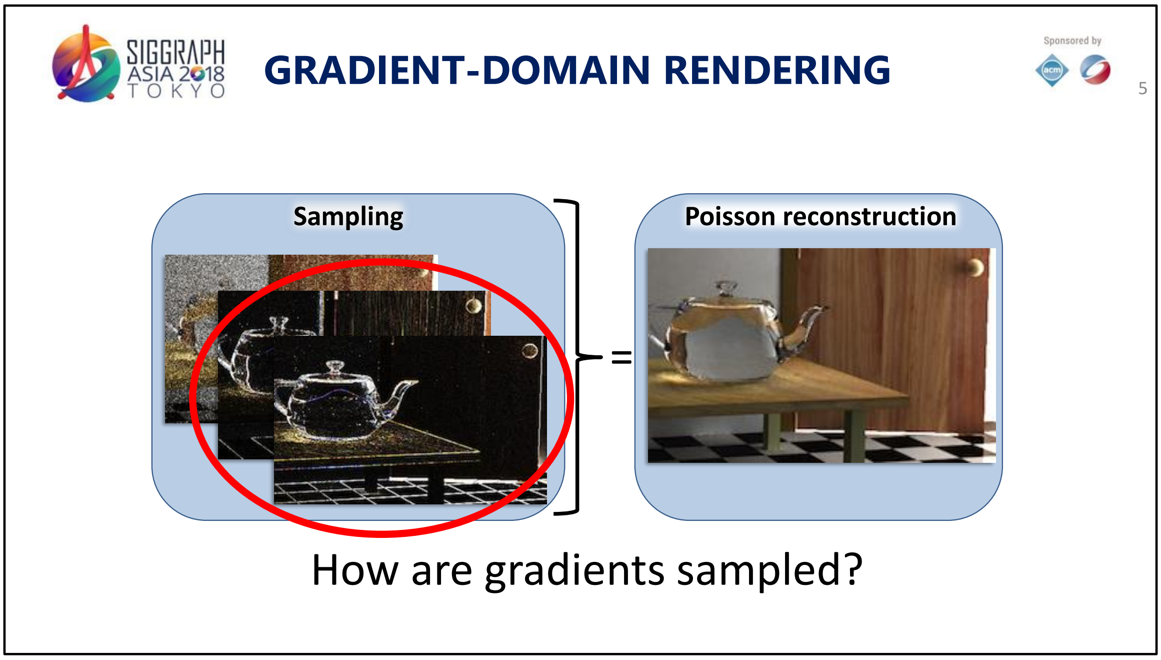

G-BDPT
Gradient-domain bidirectional path tracer
BDPT
Bidirectional path tracer
G-PT
Gradient-domain path tracer
PT
Path tracer

As we learn so far, gradient-domain path tracing (G-PT) is based on path tracing (PT) and PT is notoriously bad at sampling complex specular illumiantion effects like caustics. This behaviour is inherited by G-PT.

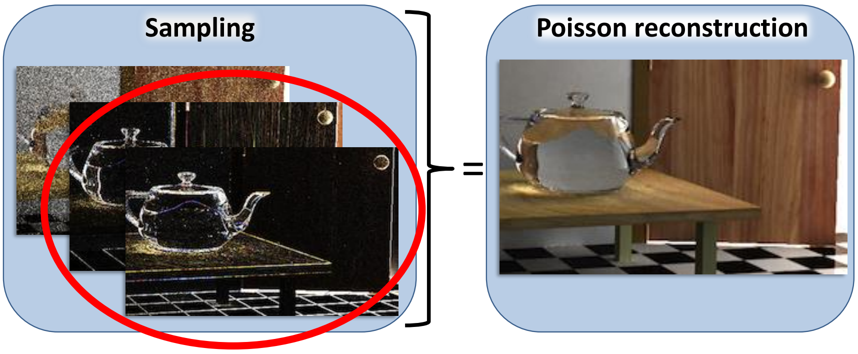
Bidirectional path tracing (BDPT) on the other hand is much more stable at sampling more difficult illumination effects. So our goal is to develop a new algorithm that combines the stability of BDPT and the noise reduction of gradient-domain rendering.

Sponsored by

Gradient-domain Bidirectional Path Tracing

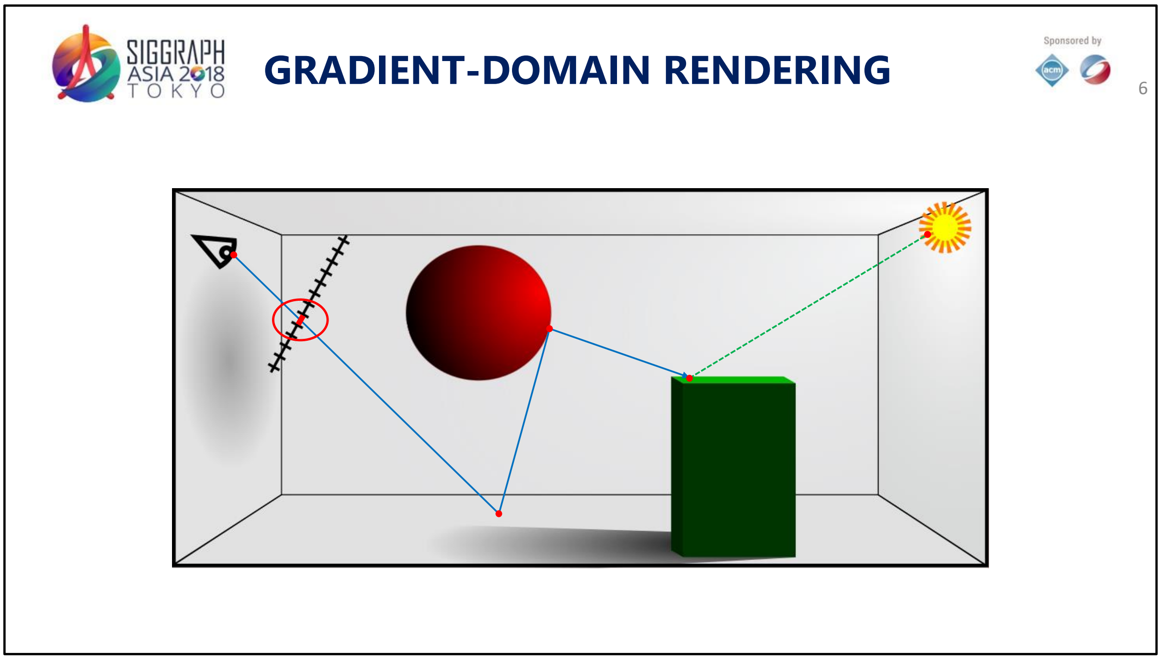


GRADIENT-DOMAIN RENDERING

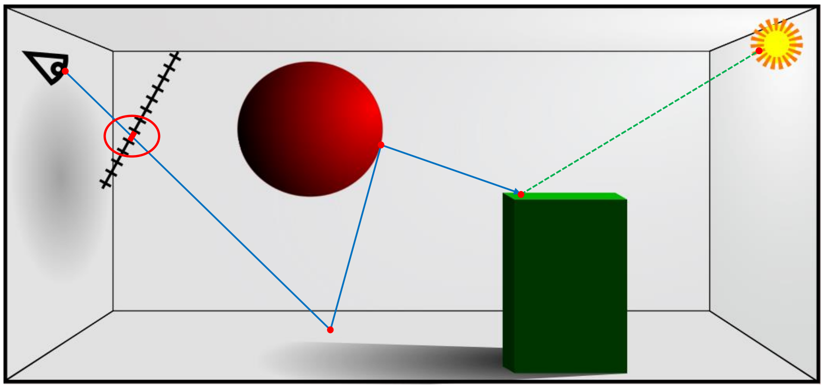


How are gradients sampled?

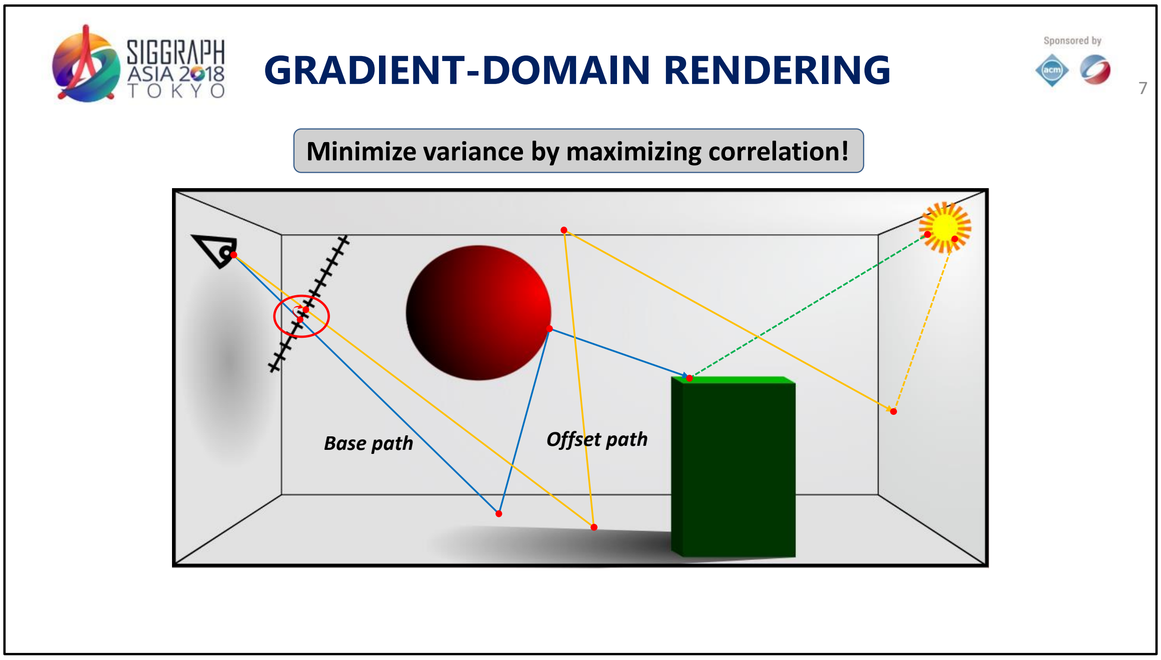
Before we talk about the details of this new algorithm, let us briefly discuss how gradients are sampled in gradient-domain path tracing such that their noise is minimized, since this is crucial for all gradient-domain rendering algorithms.



GRADIENT-DOMAIN RENDERING

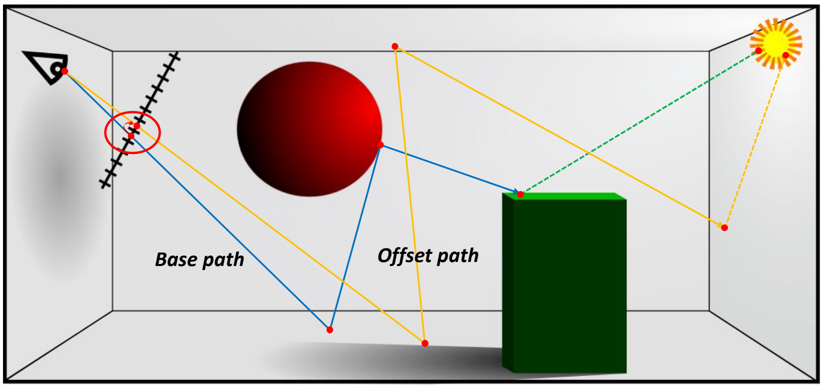


Let us look at a schamtic view of a scene with a camera and image plane, some **diffuse** objects and a lightsource. In an ordinary light transport algorithm we try to estimate pixel values. We do this by generating light ray samples starting at the eye that bounce around in the scene and are eventually connected to a light source. The pixel contribution of many such path samples are then used to estimate the pixel color.

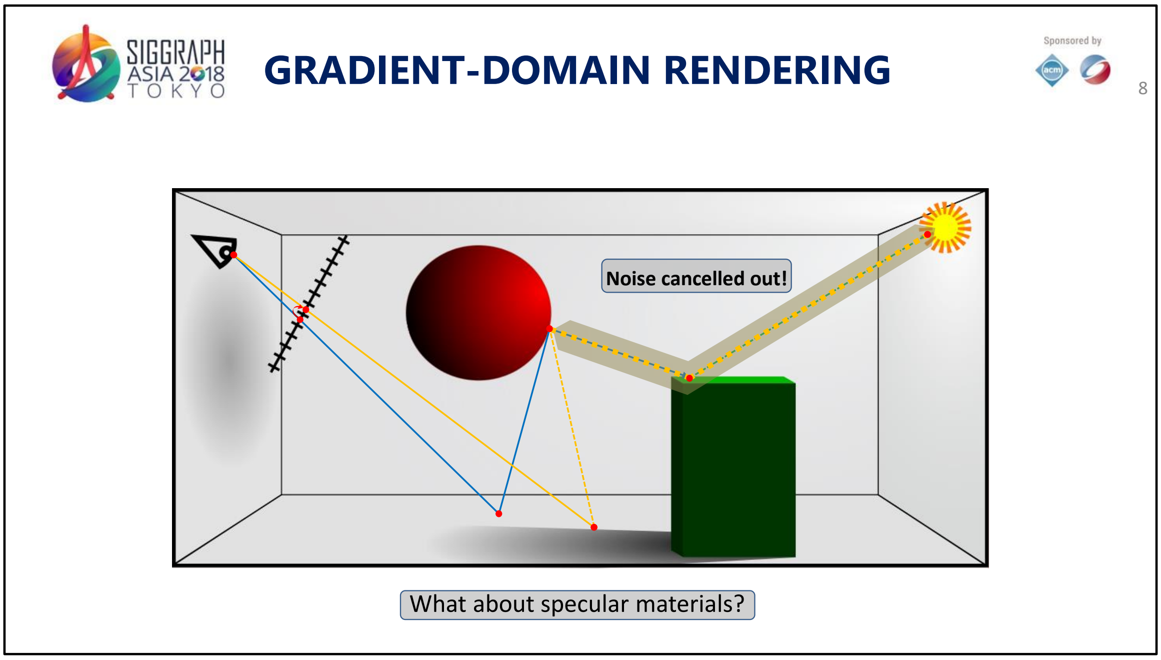


GRADIENT-DOMAIN RENDERING

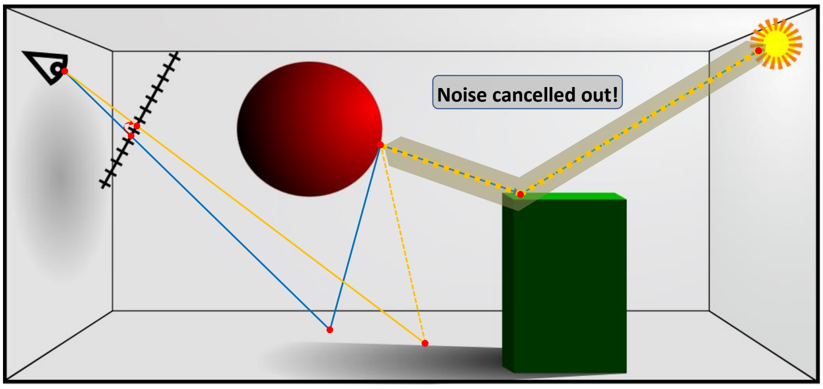
Minimize variance by maximizing correlation!



For sampling gradients we look at the difference between two pixels. We sample this difference by sampling the difference of two light paths going through adjacent pixels, we call these paths base and offset path. Despite both paths being close to each other on the image plane, they might still be very different, since at each additional bounce the path continues in a randomly chosen direction.
To **reduce the variance** of the gradient integral we need to make both paths more similar. This can be achieved by making both paths strongly **correlated**!



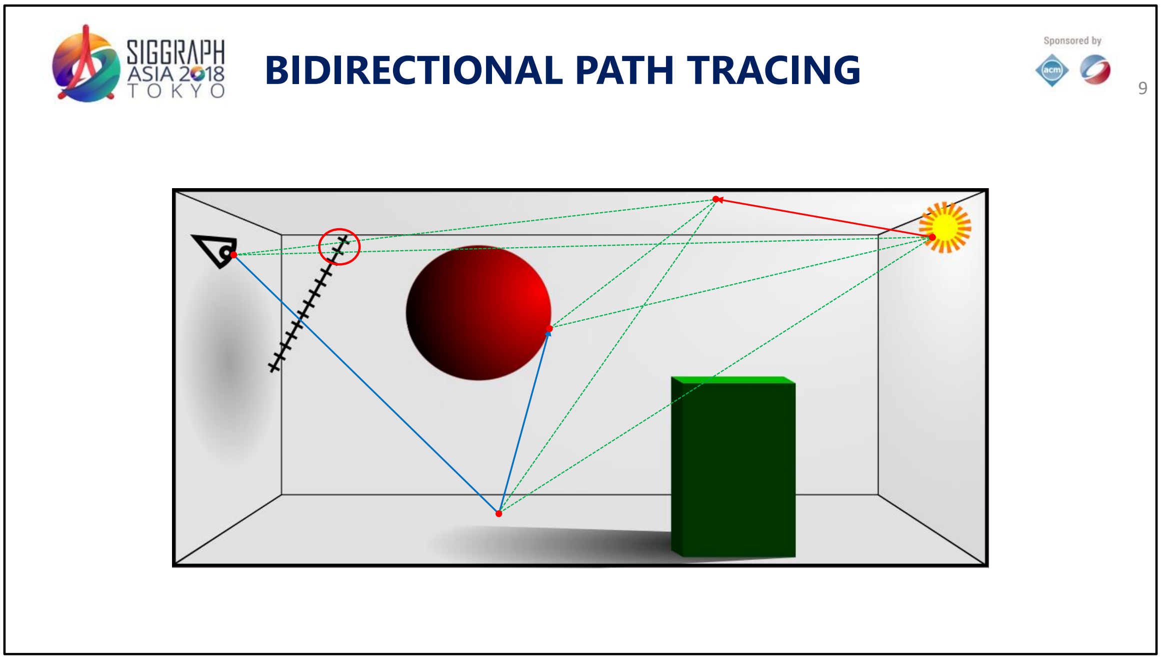
GRADIENT-DOMAIN RENDERING



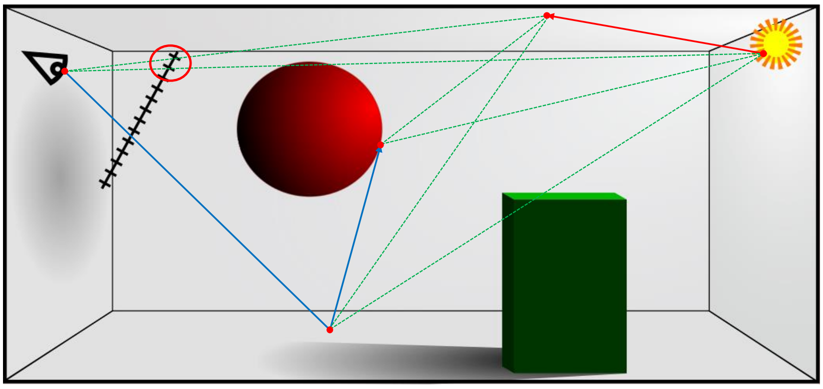
What about specular materials?

We create such correlated pairs of paths by shifting the primary intersection by one pixel and by **reconnection the shifted path as soon as possible back to the base path**. Therefore the remainder of the base and offset path will be exactly the same. As a consequence, noise introduced by the remainder of the path will be **cancelled out**. (To those familiar with Monte Carlo integration this is an applicaiton of variance reduction by common random numbers).

Note that the depicted case only works for diffuse surfaces. To see the generalization to specular materials please consult our paper.



BIDIRECTIONAL PATH TRACING

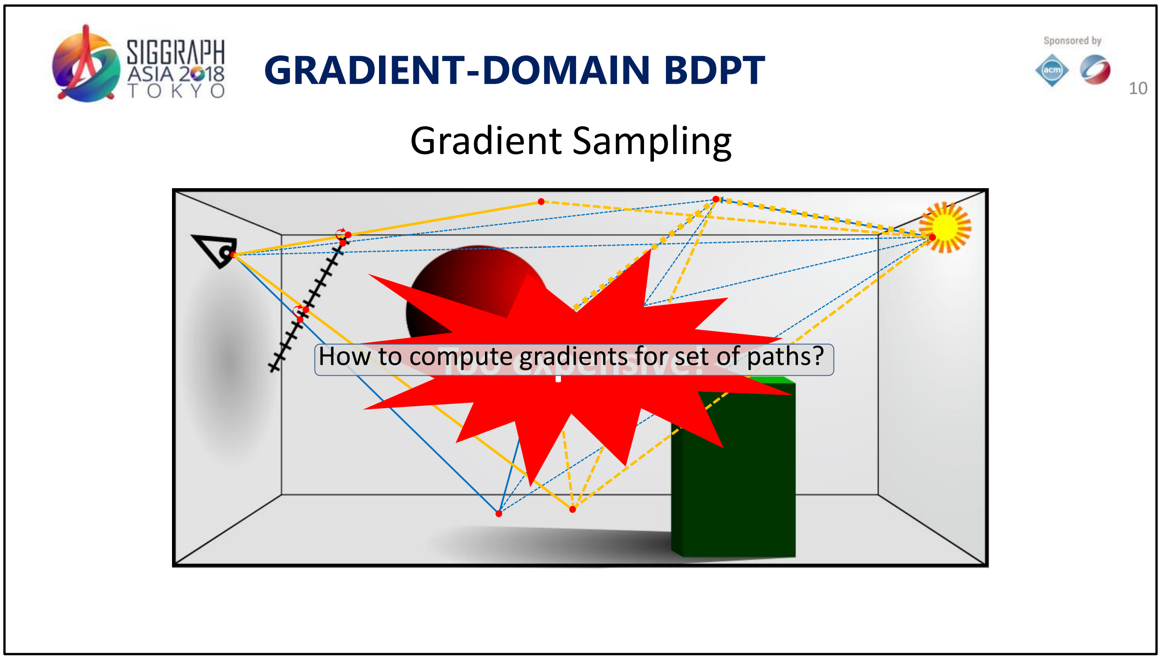


Now that we have the fundamentals of gradient-rendering, let us recap BDPT:

BDPT first traces two subpaths, one from the eye and one from the light, and both may be incomplete. By connecting these subpaths we can generate complete light paths. Connections can be done between arbitrary pairs of vertices and hence a large amount of paths can be generated from two subpaths.

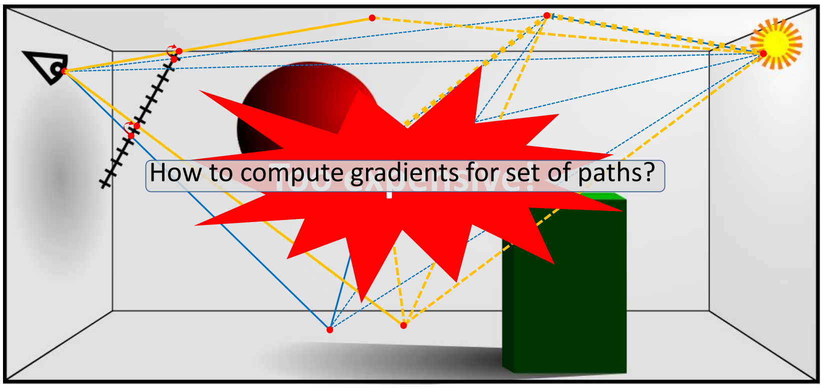
Note that the overhead per connected path is only one additional ray that must be traced which makes this way of sampling very efficient.

Note that some of the connected paths directly connect to the eye and intersect the image plane at a different position. Therefore they may contribute to different pixels in the image.



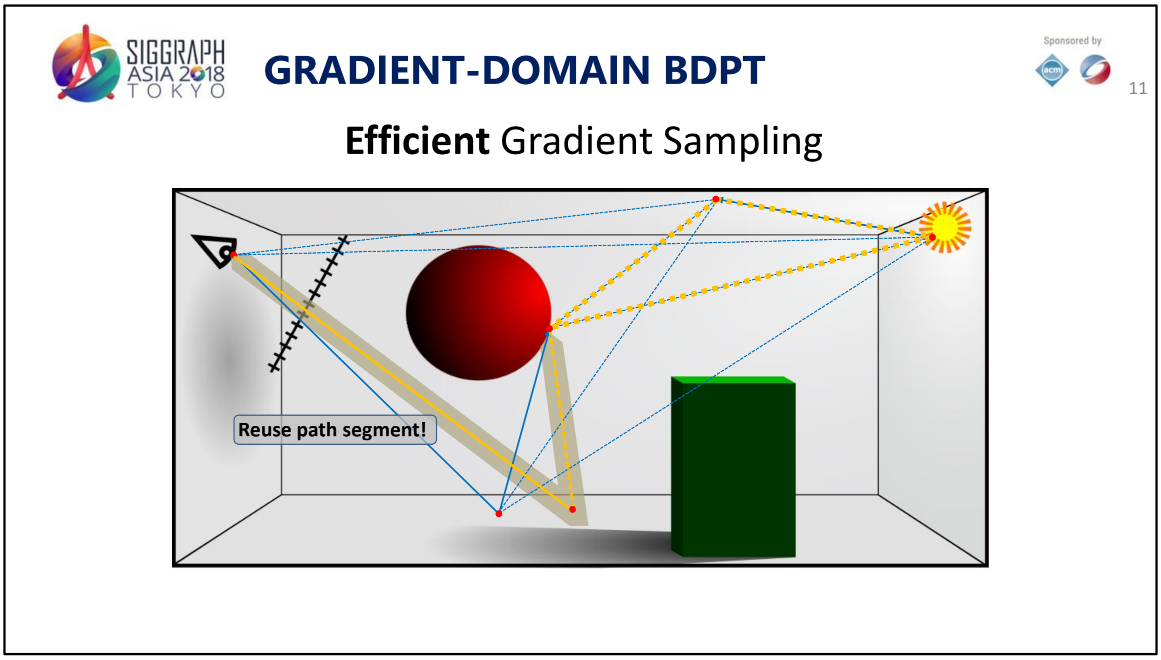
GRADIENT-DOMAIN BDPT

Gradient Sampling



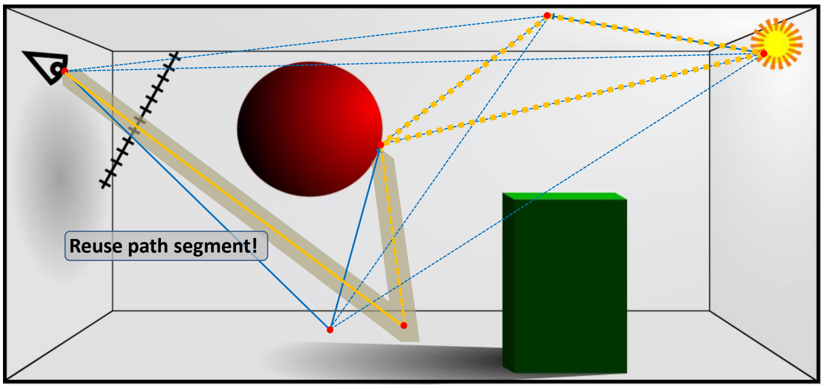
When combining this with gradient-domain rendering, the first question that arises is how to compute gradients of such a set of paths, since path-samples are not generated individually anymore.

The simplest way is to generate a gradient sample for each connected path by creating a offset path individually for each connection strategy. This is however prohibitively expensive since the number of connected paths grows quadratically with the number of vertices in the subpaths.



GRADIENT-DOMAIN BDPT

Efficient Gradient Sampling

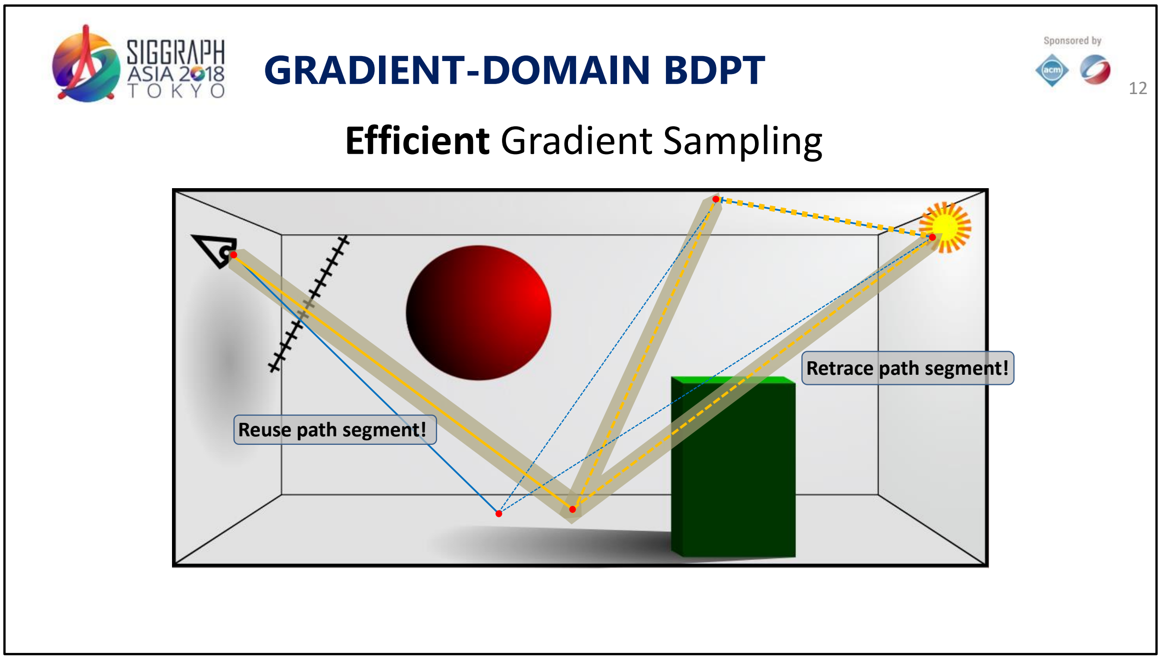


So we need to think about this can be done more efficiently. We need to think about whether there are parts of such shifted paths that can be reused between the different connected paths. In order to simplify the problem, let us look at subsets of the whole set separately.

First, let us look at connection paths that connect to the second or later vertex of the eye path. In this example we have only two such paths, but again note that in more realistic setups we have much more strategies.

Here the upper path has this offset path. And the lower path has a different offset path.

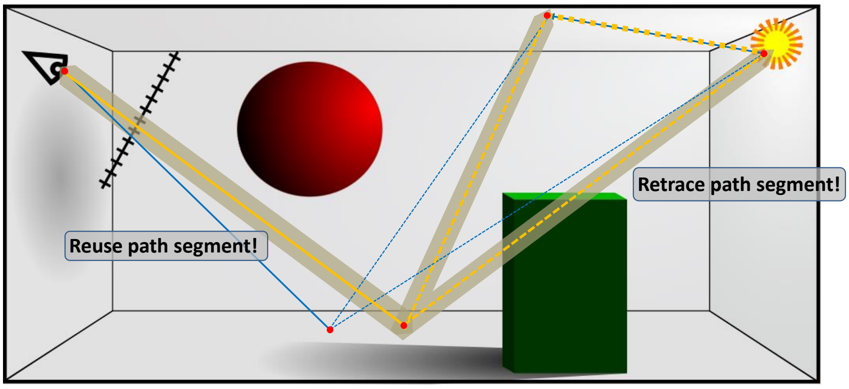
If we compare both offset paths we see that the parts of the shift that aren't equal to the base path, are actually the same for both connection paths. We can therefore compute this shared part of the offset path once and reuse it for all such connection strategies.



GRADIENT-DOMAIN BDPT

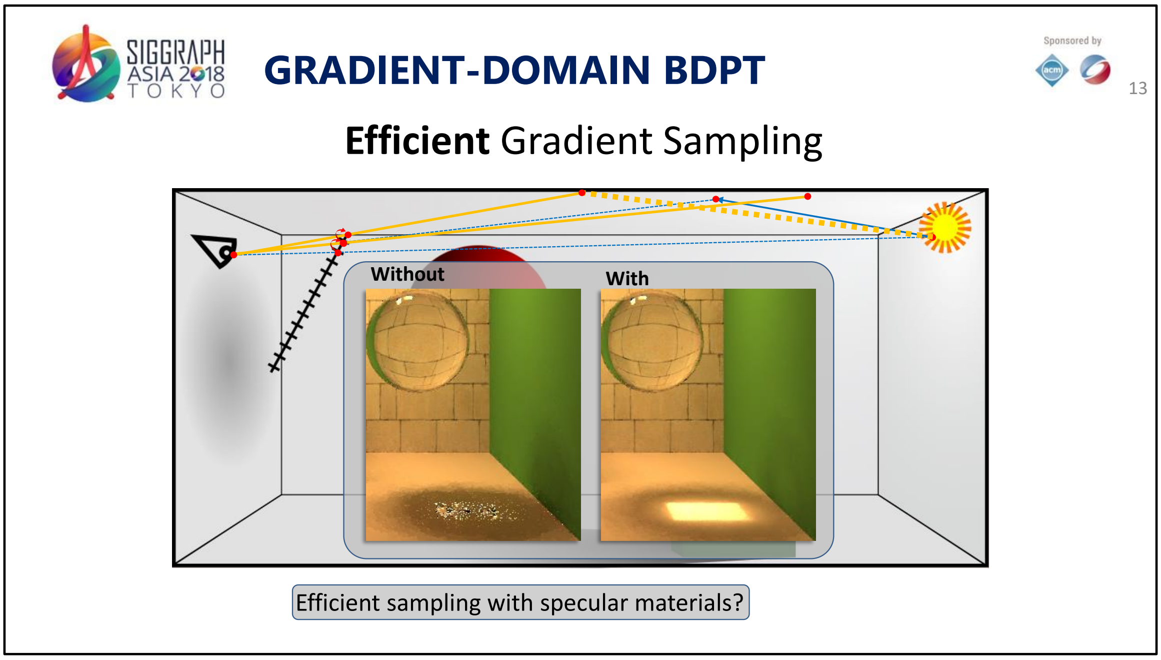
Sponsored by

Efficient Gradient Sampling



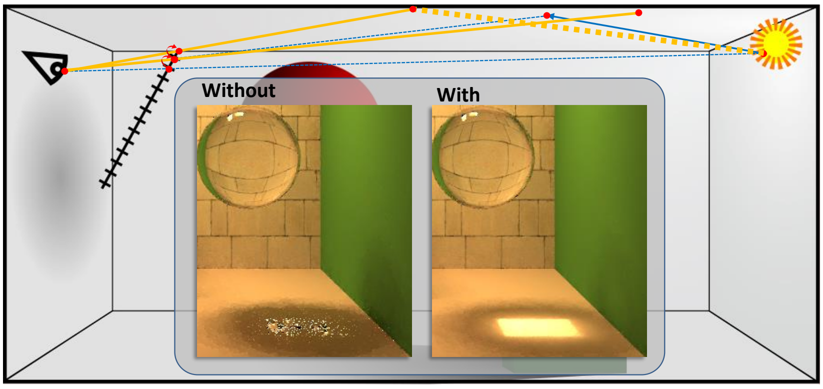
Let us now look at connection paths that connect to the primary vertex of the eye path.

Here again let us first look at the offset path of the base path at the top, and now at the offset path of the base path at the bottom.
There we see that only the primary ray of the offset path is shared for these paths.
The ray that reconnects to the basepath must be computed for each such strategy separately.



GRADIENT-DOMAIN BDPT

Efficient Gradient Sampling

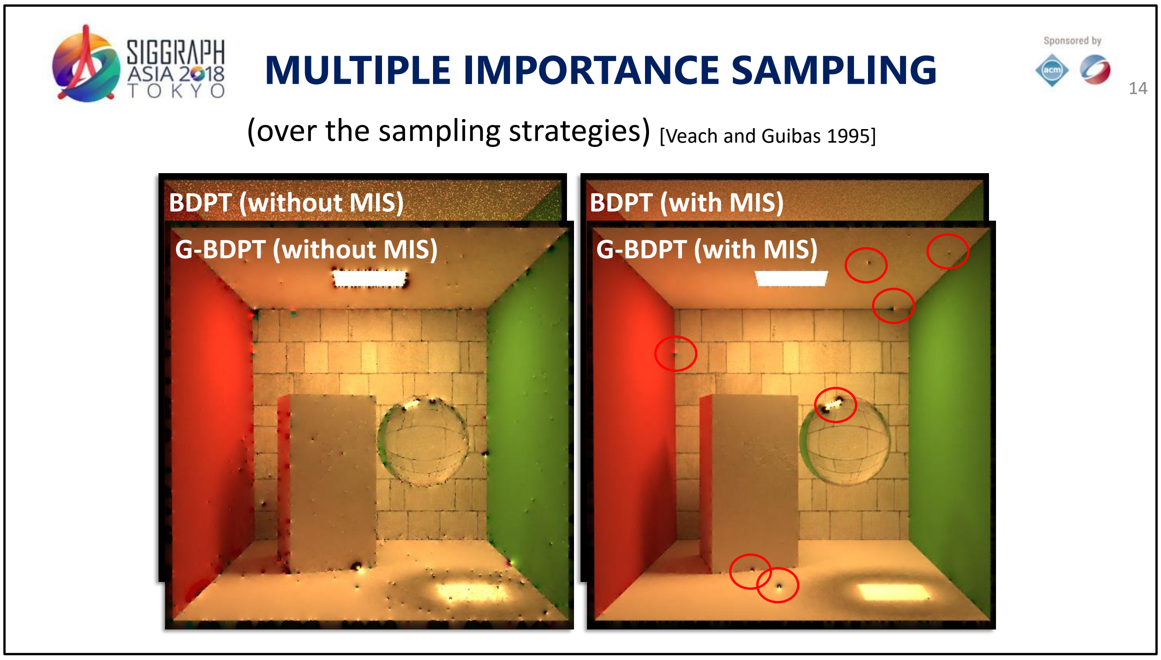


Efficient sampling with specular materials?

The last subset of the paths that we look at are light tracing paths; These are stategies that connect directly to the eye. This type of paths is important to compute caustics and therefore we also want to compute gradient samples for them.

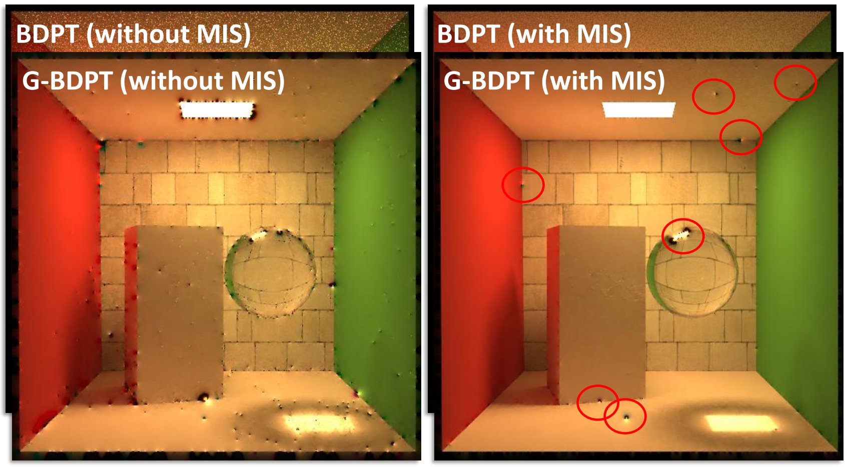
Since the base paths of light tracing paths contribute to different their offset paths might be completely different from each other, therefore we must compute the entire offsetpath for each light tracing path from scratch! This is not very efficient, but since only a small fraction of all connection strategies are light tracing paths this is still affordable.

As a sidenote: Our explanations are only valid for diffuse paths, as soon as specularity comes into play, the efficient graident sampling becomes slightly more complicated. In general, we have to employ manifold exploration [Jakob et al., 2012] for generating the offset path.

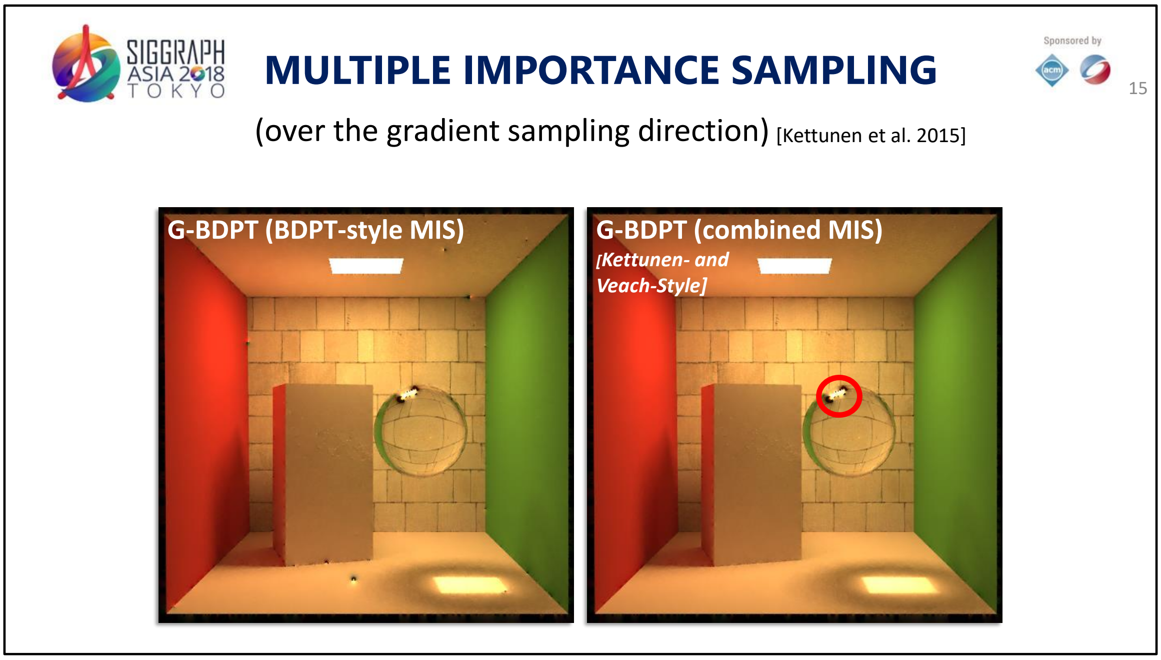


MULTIPLE IMPORTANCE SAMPLING

(over the sampling strategies) [Veach and Guibas 1995]

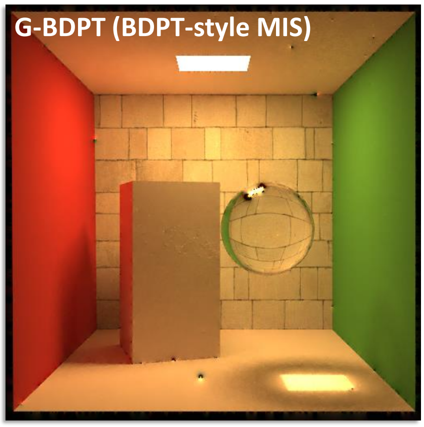


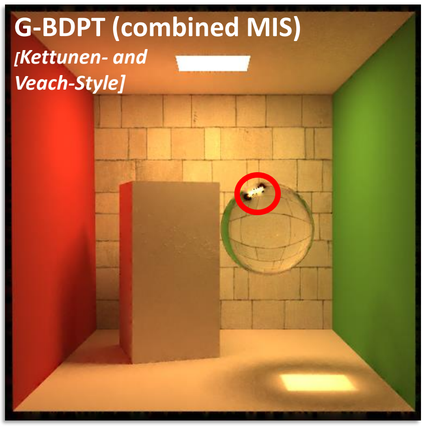
In 1995, Veach and Guibas described that in BDPT each path can be generated in different ways with different sampling strategies. They showed that by weighing these strategies in an optimal way, noise can be reduced significantly. If we apply exactly the same MIS on gradients we can significantly reduce reconstruciton artefacts of G-BDPT. However some problems still remain.



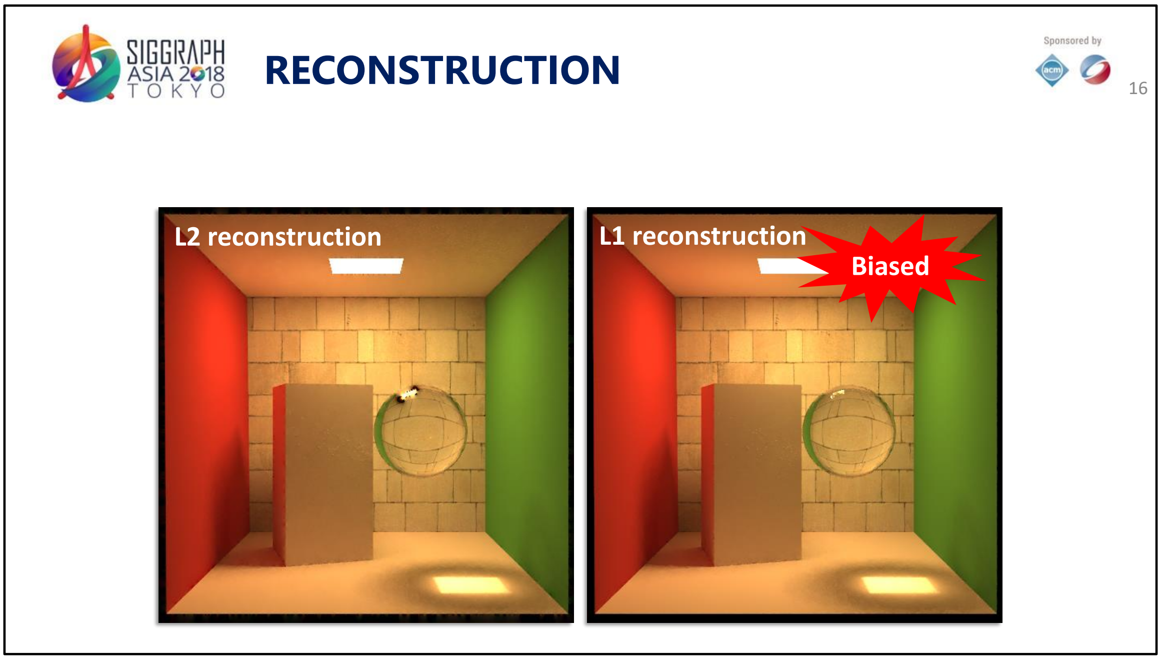
MULTIPLE IMPORTANCE SAMPLING

(over the gradient sampling direction) [Kettunen et al. 2015]

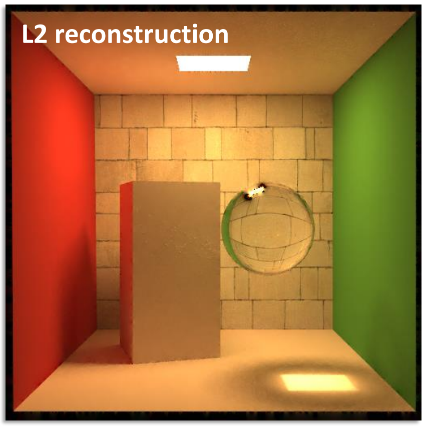


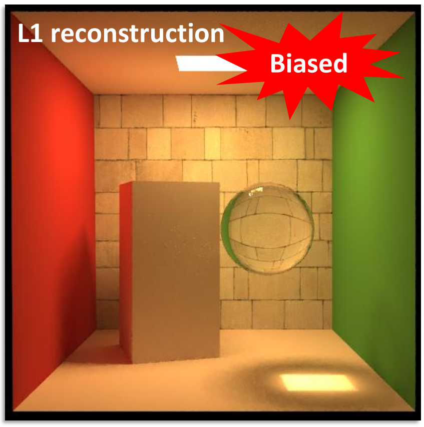


We came up with a way to extend MIS for gradient samples. The main insight here is that a gradient sample can be generated in twice as many ways as a ordinary sample. This comes from the fact that a gradient from a pixel A to a pixel B can also be sampled from pixel B to pixel A with flipped sign (i.e. by reversing the gradient direction). When we now use this to extend the classical Veach Style MIS we can further reduce reconstruction artefacts.
However some artefacts remain even then, and this is due to path types that cannot benefit from MIS (since they can only be smapled by one strategy). To resolve this problem we use the same strategy as previous work.

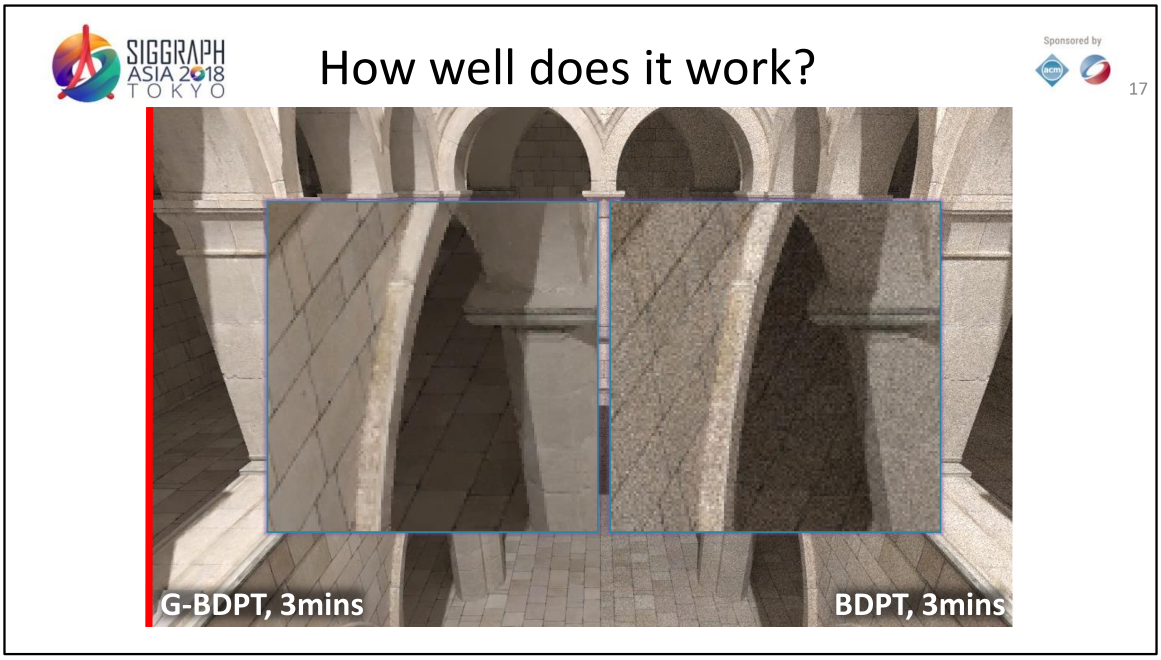


RECONSTRUCTION



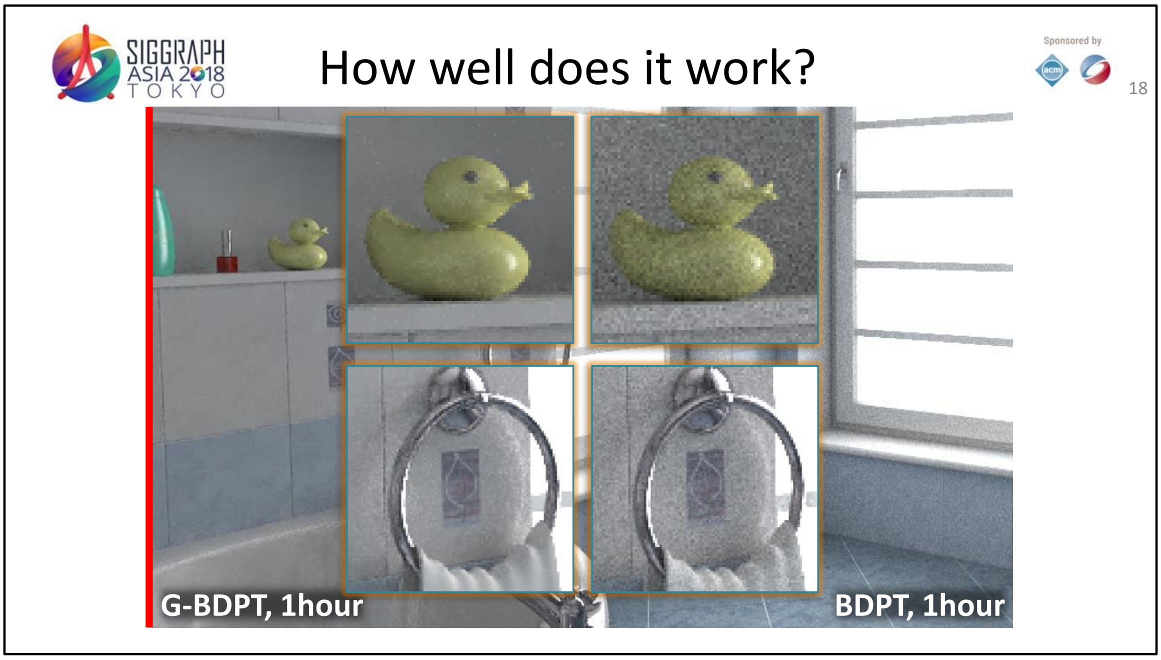


Instead of using a screened poisson reconstruciton that minimizes the L2 error, we use a iterative reweighted least square reconstruciton that minimizes the L1 error. This yields a reconstruction that is more stable but this comes at the price of bias. Despite bias, we found the L1 reconstruciton to yield more nummerically and visually pleasing results and used it to generate most our results.



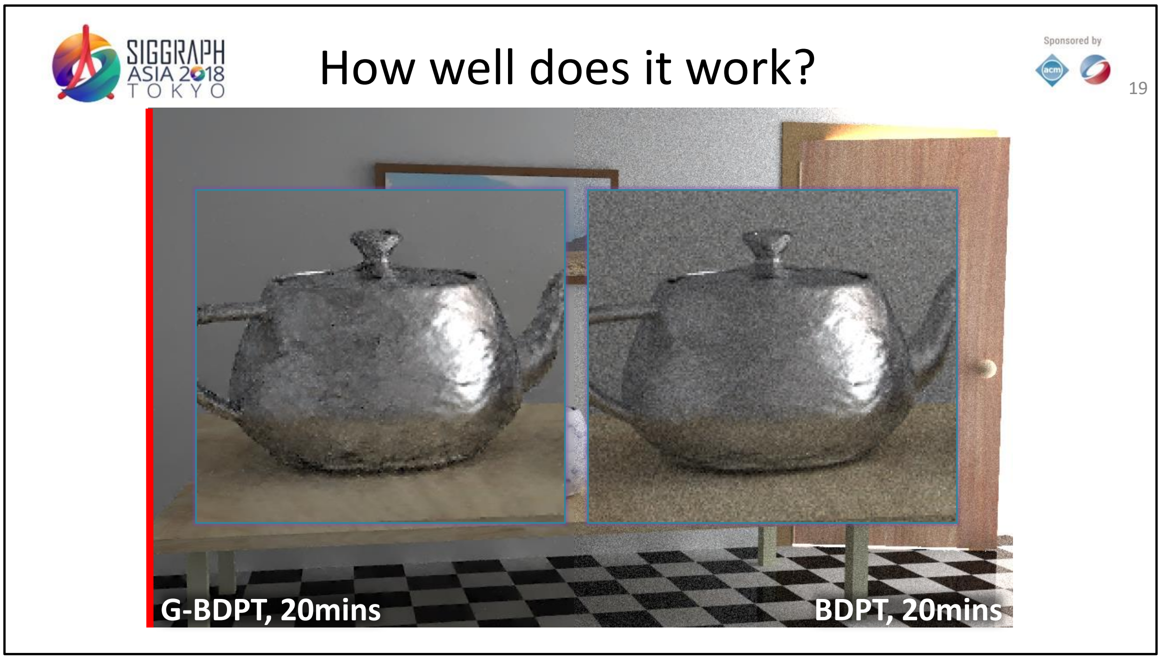
How well does it work?

So now we the final question arises: How well does our new algorithm work?
In simple scenes like the one here at equal time we get significant noise reduction comparable to the previous gradient-domain path tracing algorithm.

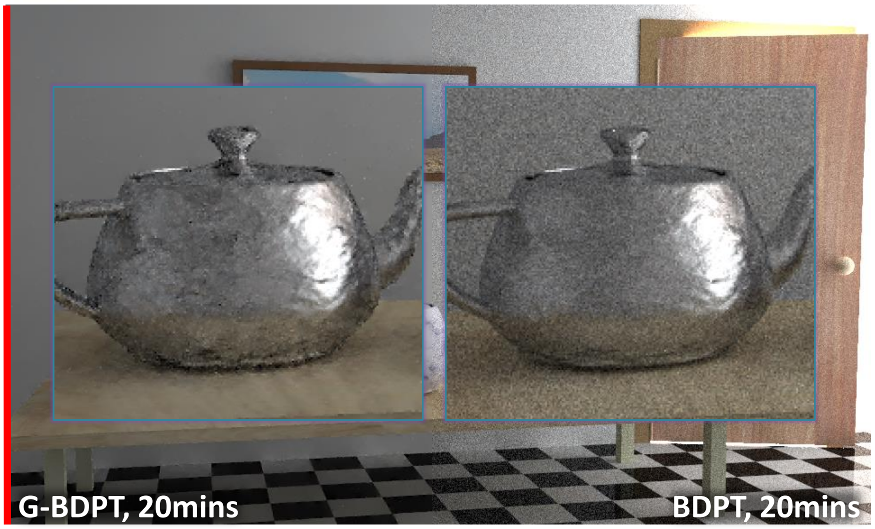


How well does it work?

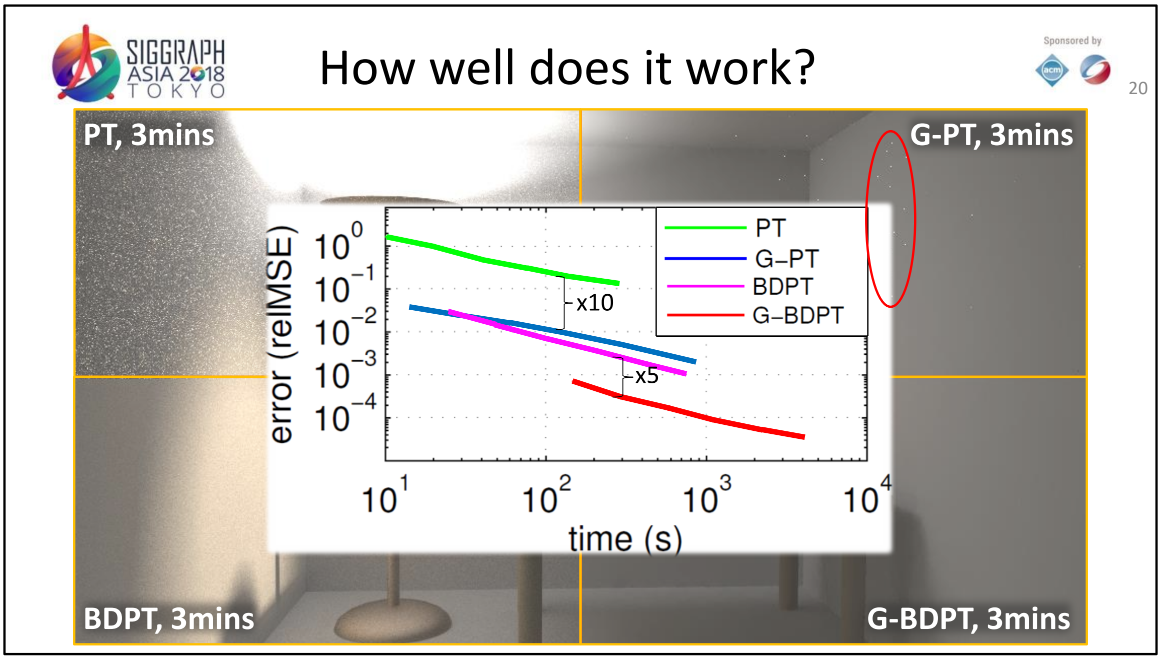
For more complex scenes like Bathroom that has heavy noise even after hours of rendering. We also get significant noise reductions. Note that at some of the caustics our algorithm isn’t able to completely remove the noise, but in more uniform regions it removes it nearly completely.



How well does it work?



The next example is the Door scene. This is a scene that is illuminated indirectly by a light source behind the door. There again we see that at equal time our algorithm very efficiently removes noise in flat regions. However we also see that in geometrically complex regions like the metal teapot, denoising doesn't work as well. This is because this surface has strong bump maps, and because of this it is not always possible to find a very similar offset path. Therefore noise cancellation in the gradients becomes less effective and also the denoising in the reconstruction becomes less effective. However in terms of relative error, G-BDPT is still beneficial compared to BDPT by an order of magnitude at equal time.



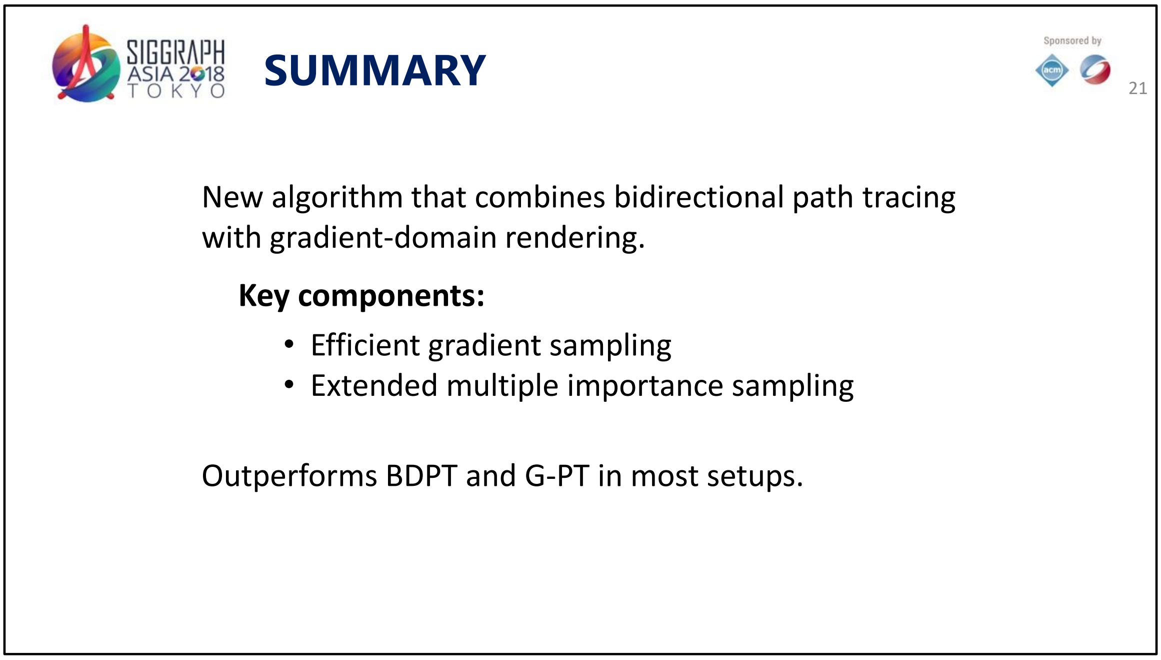
How well does it work?

Another example is to compare PT and BDPT with and without gradient-domain rendering in a simple scene with caustics and indirect illumination.
With PT the scene is resolved very badly and suffers from extensive noise because next event estimation fails most of the time.
G-PT successfully removes most noise coming from diffuse interreflections, but fails at removing noise from caustics, which are seen as bright outlies in the image. BDPT is much better for rendering this scene because it can connect more often to the light source and resolve caustics better, but still not as good in denoising diffuse interreflections as G-PT.

By contrast, G-BDPT has similar denoising functionality as G-PT but also succeeds at denoising caustics.

Looking at convergence plots for this scene with compared relMSE over time, we see that at equal time G-PT outperforms PT by a factor of 10, but is similar in terms of quality as BDPT. G-BDPT now improves upon BDPT again by a factor of 5.

In general G-BDPT directly outperforms G-PT in scenes where the underlying BDPT sampler is better than PT.



SUMMARY

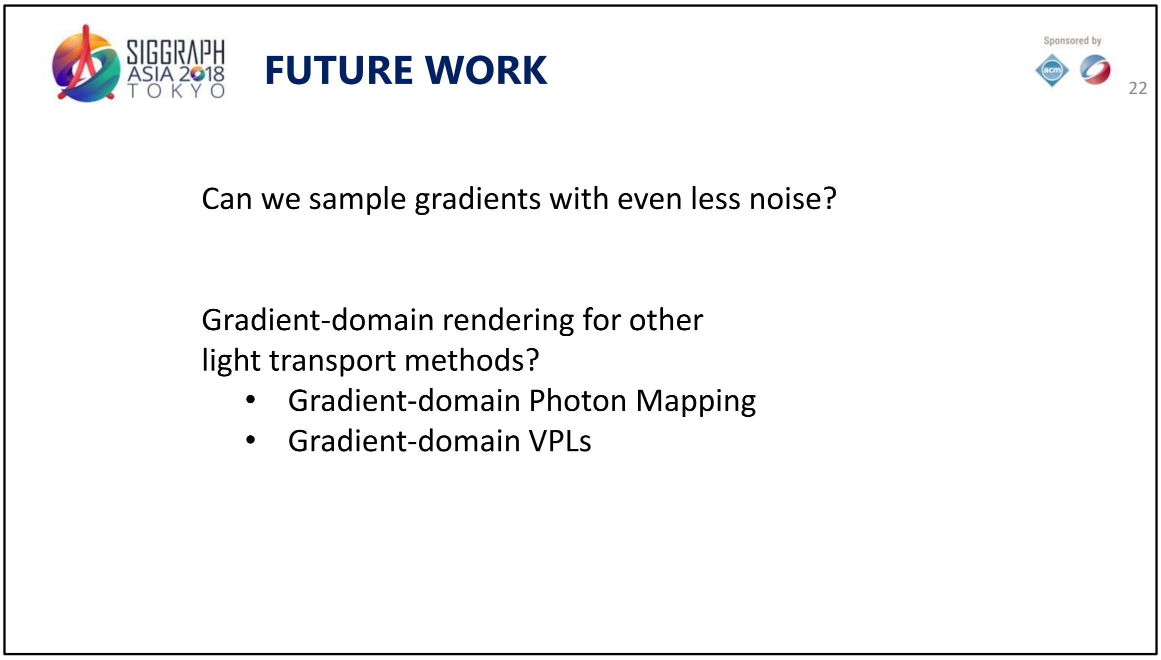
New algorithm that combines bidirectional path tracing with gradient-domain rendering.

Key components:

- Efficient gradient sampling
- Extended multiple importance sampling

Outperforms BDPT and G-PT in most setups.

In conclusion: We presented a new algorithm that combines BDPT with gradient domain rendering. The algorithm combines the robustness of BDPT with the noise-reduction of gradient-domain rendering. We showed two key components of the implementation: How to efficiently sample gradients and how to extend MIS to benefit from all gradient sampling strategies. We showed that our algorithm outperforms BDPT and G-PT in most setups.


FUTURE WORK

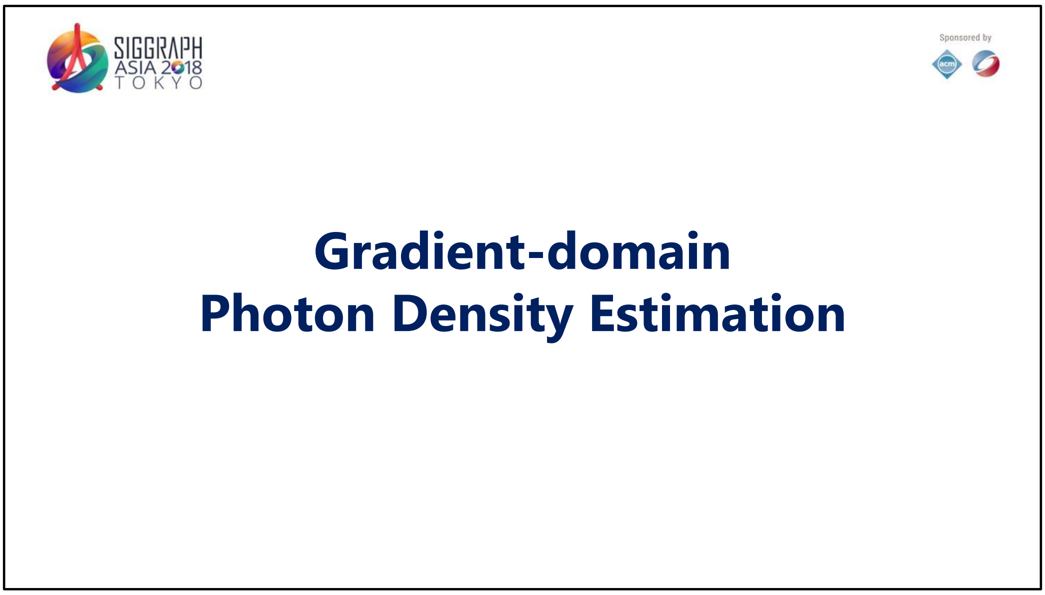
Can we sample gradients with even less noise?

Gradient-domain rendering for other light transport methods?

- Gradient-domain Photon Mapping
- Gradient-domain VPLs

As future work I want to emphasize that the gradient sampling that we use still suffers from noise in more complex setups and we believe that there is room for improvement there.

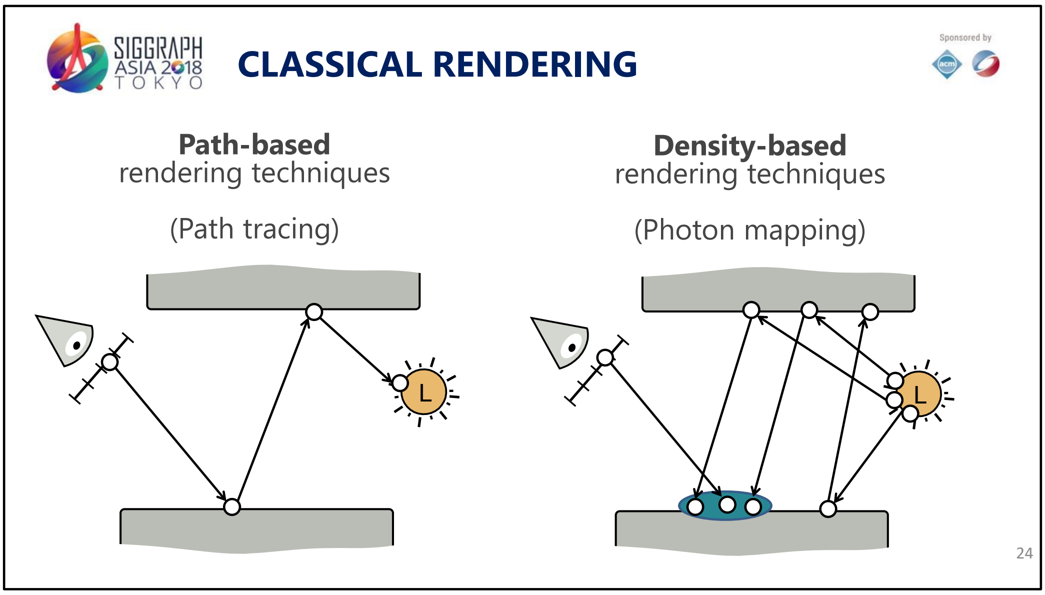
Further we think that our work implies that gradient-domain rendering can be applied on a multitude of light transport algorithms besides the one we already tried out; for example what about gradient-domain photon mapping or gradient domain VPLs? We believe this could further increase the usefulness of gradient-domain rendering.



SIGGRAPH ASIA 2018 TOKYO

Sponsored by

Gradient-domain Photon Density Estimation



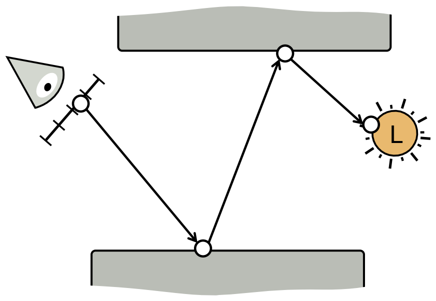
CLASSICAL RENDERING

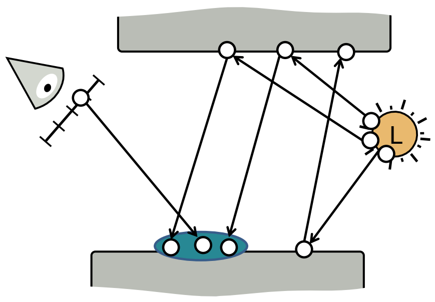
Path-based rendering techniques

(Path tracing)

Density-based rendering techniques

(Photon mapping)



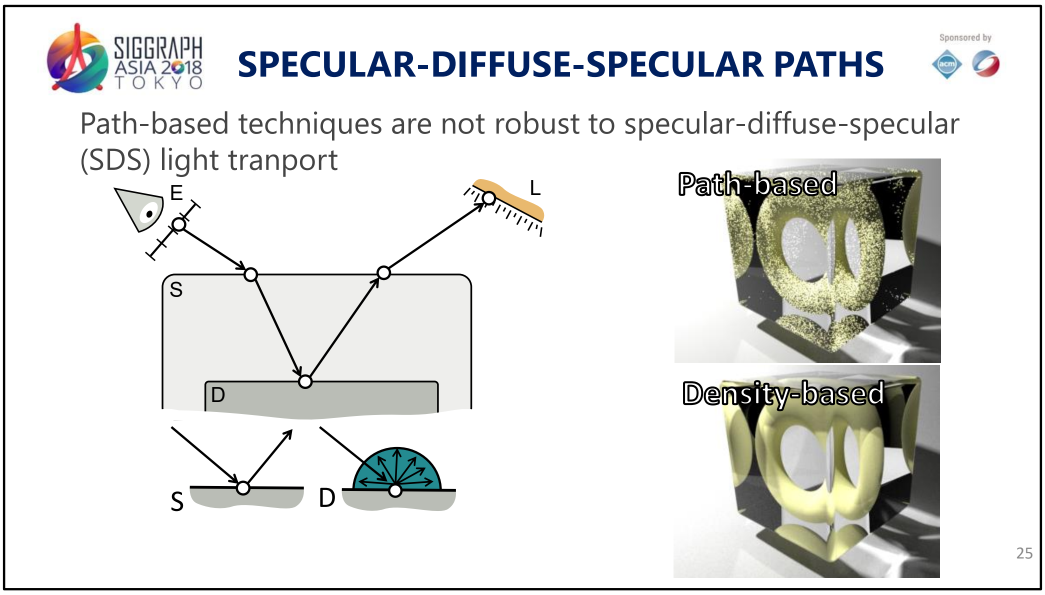


24

In the classical rendering, there is two main types of techniques.

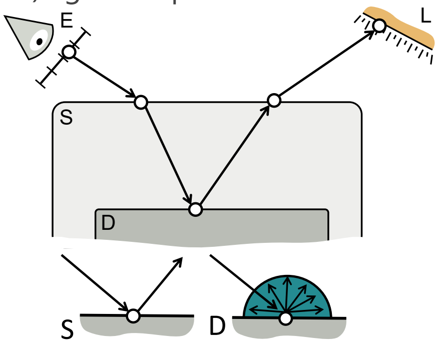
The first one are path-based rendering approach that construct the path incrementally from the camera or the light by bouncing over the surfaces.

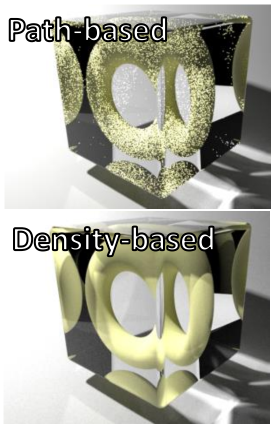
The second type of rendering technique is density-based rendering techniques which have two main steps:
First, several light path are generated from the light source and all their vertices are stored inside a spatial data structure.
Second, a path is generated from the camera and at some point there is a density estimation. The density estimation gathers all neighboring light paths and merge them to create complete light paths that connect from camera to light.



SPECULAR-DIFFUSE-SPECULAR PATHS

Path-based techniques are not robust to specular-diffuse-specular (SDS) light tranport



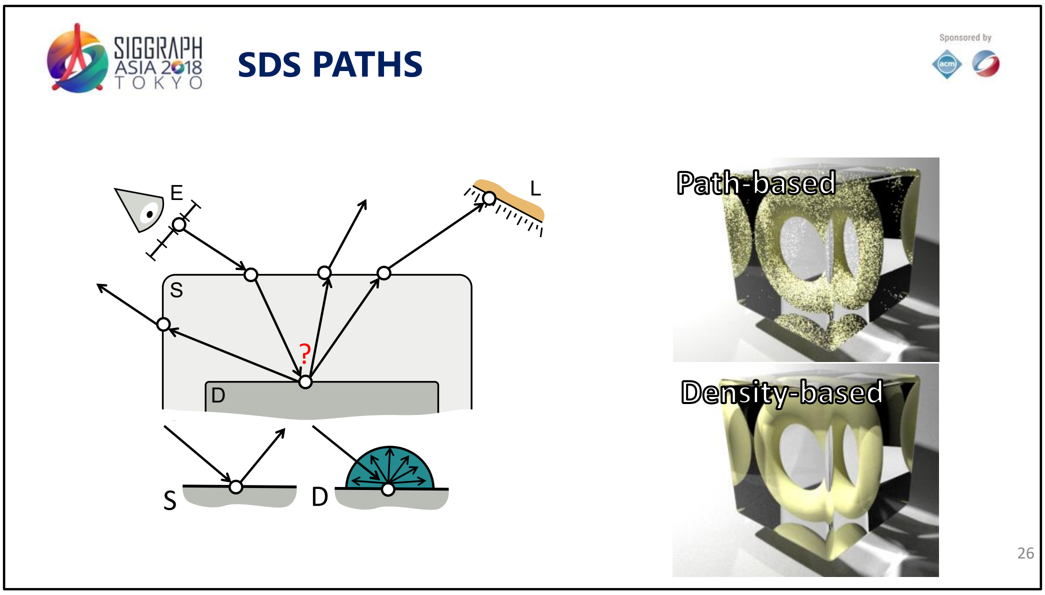


25

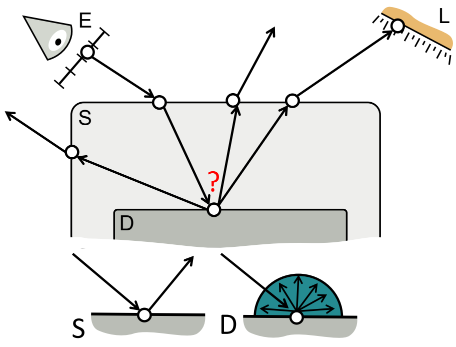
It is known that path-based rendering approach have some problem with “specular diffuse specular” light transport.

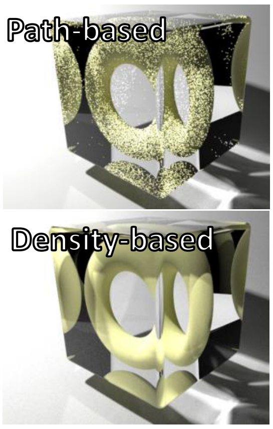
This difficulty is due to the fact that specular material like mirror, restrict the bouncing direction. It have some consequence that path-based approach are not good to handle this constraint and have a lot of noise in this light transport.

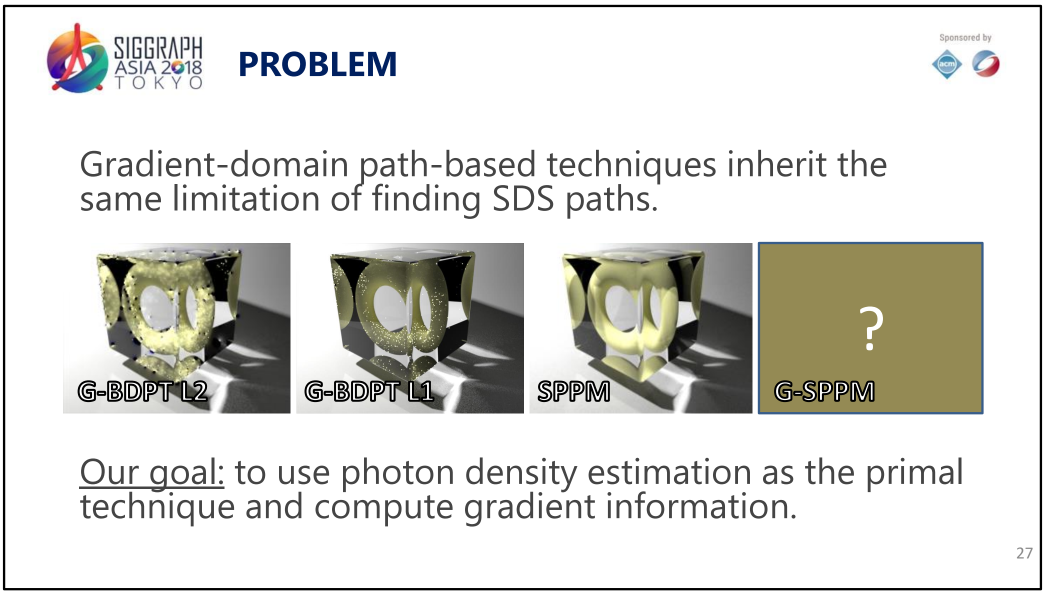
As it is shown in this image, there is a lot of noise on the yellow torus.
In comparison, density-based technique are more robust to this light transport and produce image with much less noise.



SDS PATHS

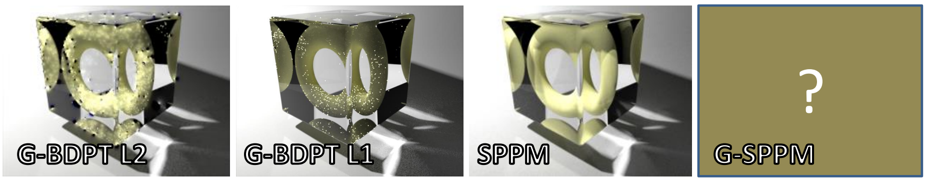






PROBLEM

Gradient-domain path-based techniques inherit the same limitation of finding SDS paths.



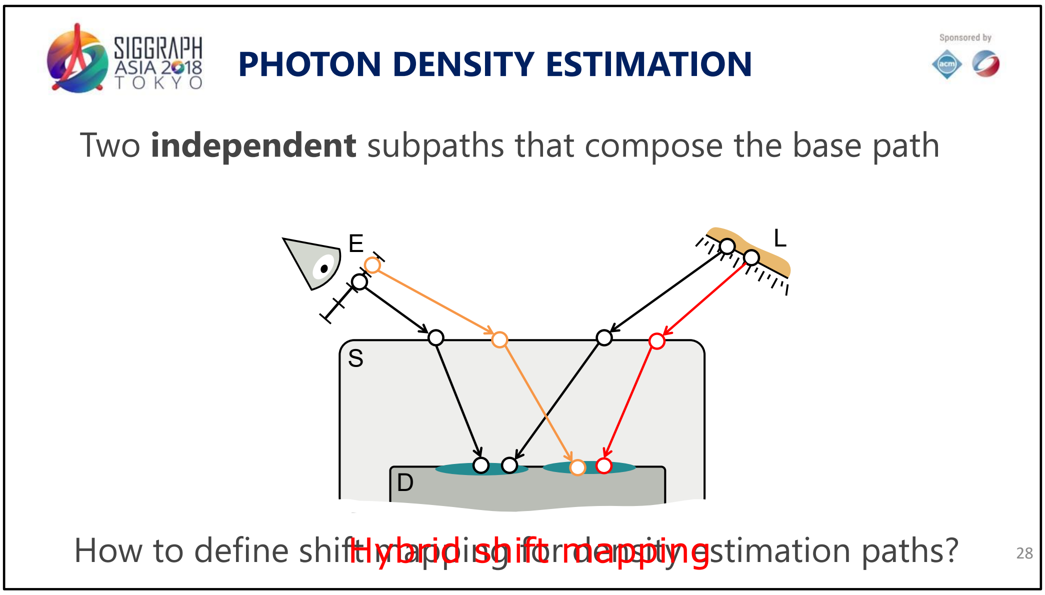
Our goal: to use photon density estimation as the primal technique and compute gradient information.

27

The current problem is that all gradient-domain rendering is path-based approach and share the same issue on the Specular Diffuse Specular light transport.

Let us look at the example here rendered with bidirectional path tracing (BDPT) with the original and gradient-domain version. We can see that the torus is not properly rendered. So using different ways of image reconstruction does not really help here: L2 have a lot of dipole artifact and L1 have a huge loss of energy.

Inspiring by the fact that photon density estimation is robust to such light transport: Stochastic Progressive Photon Mapping (SPPM) [Hachisuka et al., 2009] works really well here, our goal in this section is to improve the robustness of gradient-domain techniques using photon density estimation as the primal technique.



PHOTON DENSITY ESTIMATION

Two **independent** subpaths that compose the base path

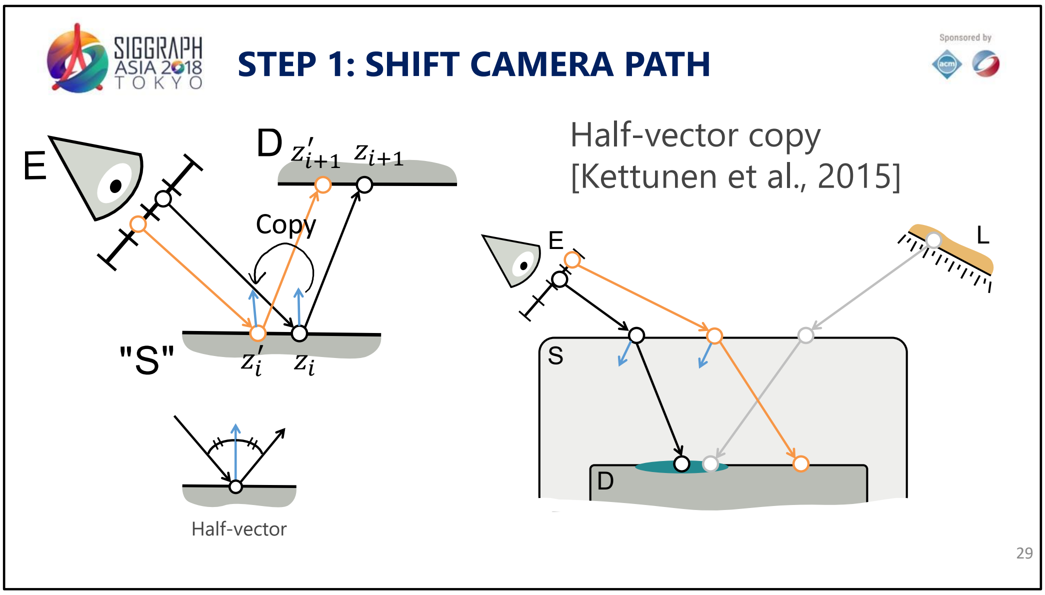
How to define shift mapping for density estimation paths?

Hybrid shift mapping

Let’s recap how photon density estimation creates a complete path. We first trace two independent subpaths, one from the camera and one from the light. A density estimation kernel is centered at the last vertex on the camera subpath, and gather energy from the light subpath. The kernel transfers energy from y_s to z_t.

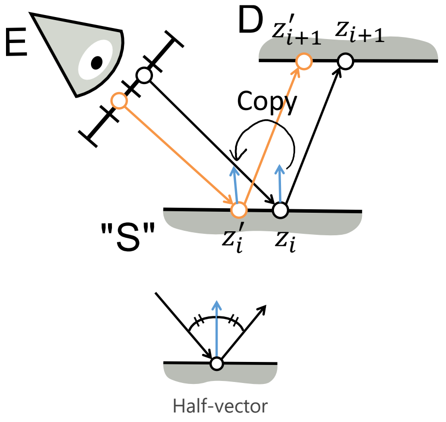
For gradient-domain rendering, we haven’t seen shift mapping for this type of path before. We need a mechanism to shift them.

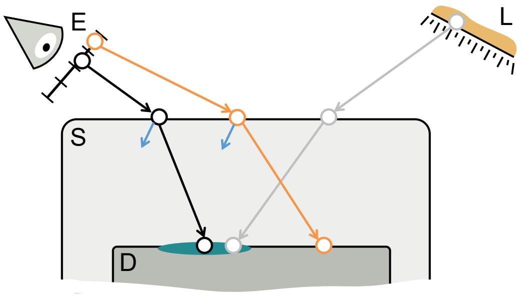
Here we solve it using a hybrid shift mapping that has two steps.



STEP 1: SHIFT CAMERA PATH

Half-vector copy
[Kettunen et al., 2015]





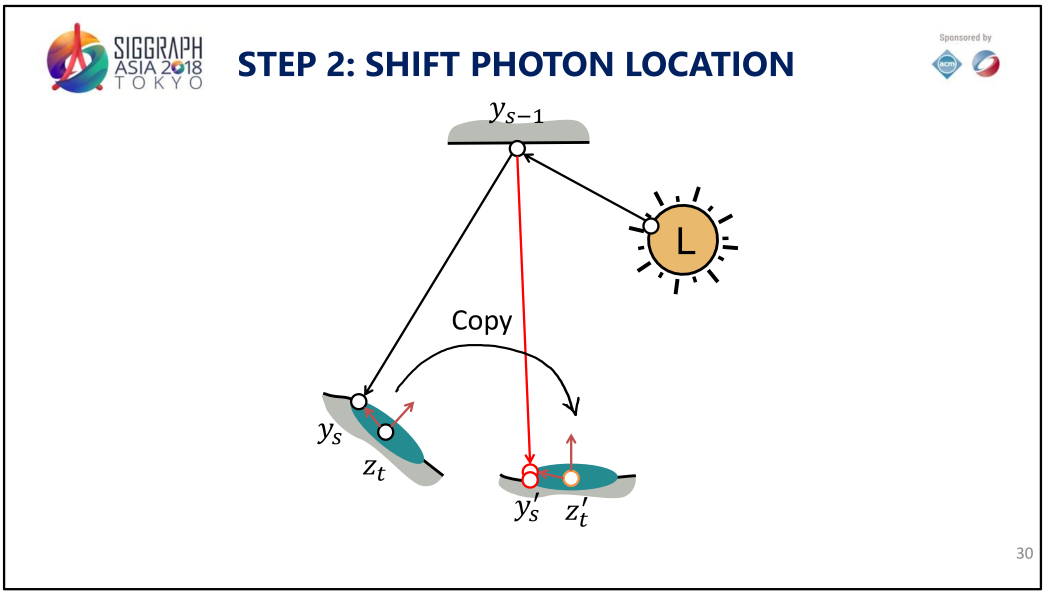
29

The first step of our shift mapping is to shift the camera path. For clarity, we denote vertices starting from the camera on the camera subpath as from z_0 to z_t, and vertices starting from the light on the light subpath as from y_0 to y_s.

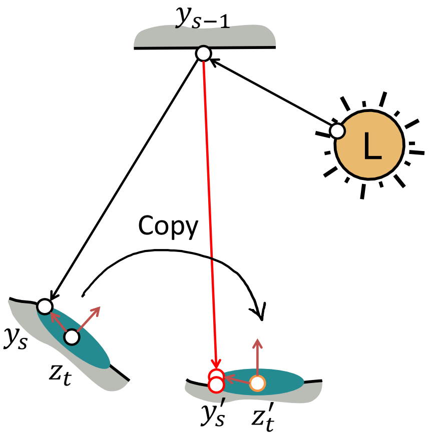
Note that similar to the final gathering step in photon mapping, when we generate the base camera path, if we encounter not enough smooth surface, we bounce over it until reach a diffuse surface.

Then generating the offset camera path is greatly similar to gradient-domain path tracing. The difference here is that we only bounce over not enough smooth surface until we hit a diffuse surface instead of a light. To determine the outgoing direction, we copy the local half vector from the base path. This scheme is simple and able to create coherent offset camera path.

After this first step, we have the offset camera path now. Let’s proceed to the next step.



STEP 2: SHIFT PHOTON LOCATION

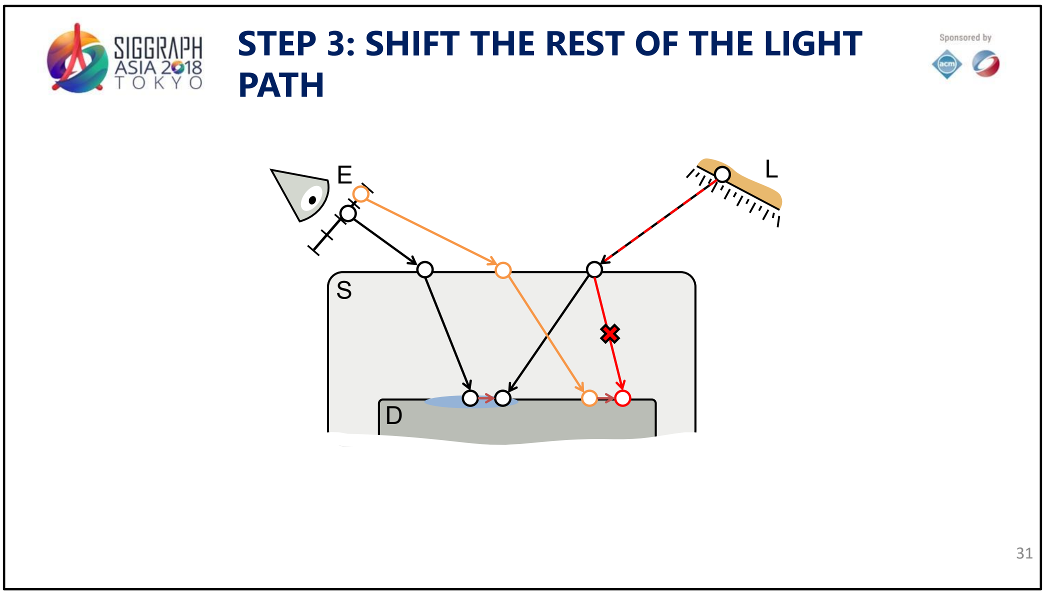


30

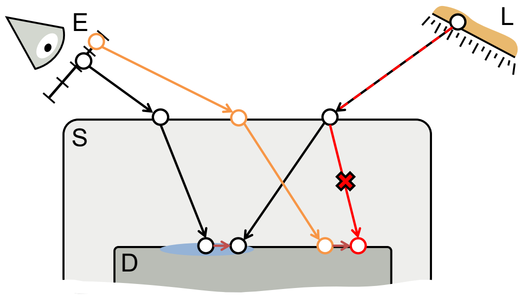
During the second step, we want to ensure the shift photon to lie inside photon density kernel.

To ensure that we maintain the relative position of the photon to the last vertex on the camera path in the shift. This determines the offset photon position.

However, as this procedure cannot ensure that the offset photon to lie on a surface, we trace an additional ray from the parent photon to project it to the nearest surface.



STEP 3: SHIFT THE REST OF THE LIGHT PATH

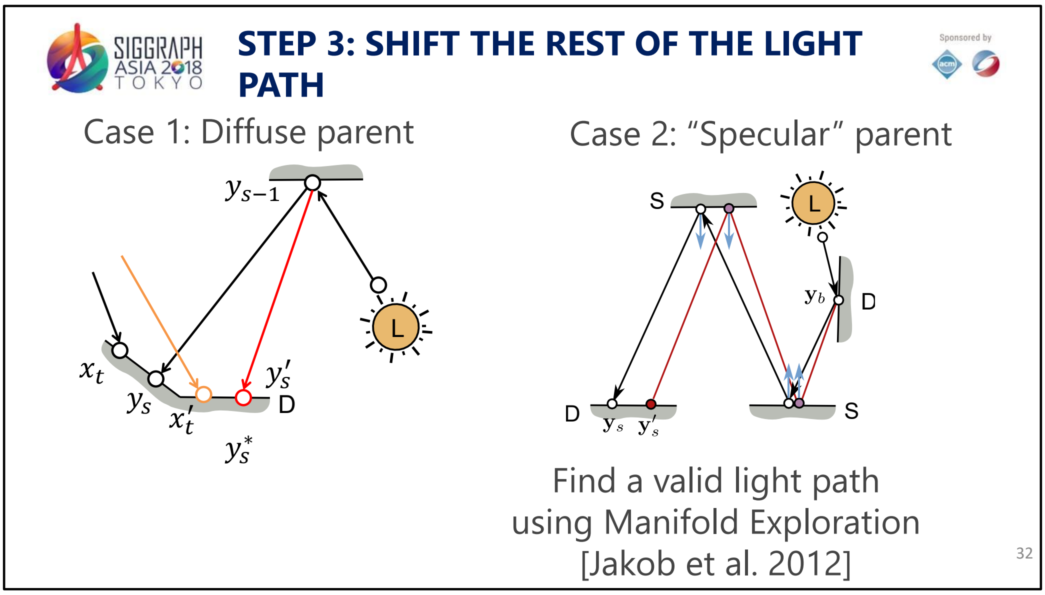


31

After we determine the offset photon, we can try to reuse the light path as much as we can by reconnecting the offset photon to the parent photon.

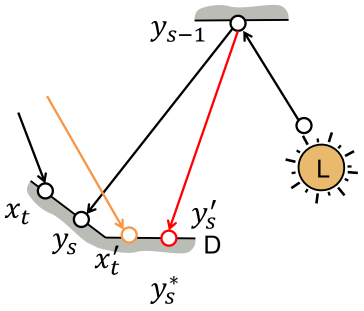
However, in some cases, we have specular materials that make this direct connection not possible.

So we need a third and last step to shift the rest of the light path.

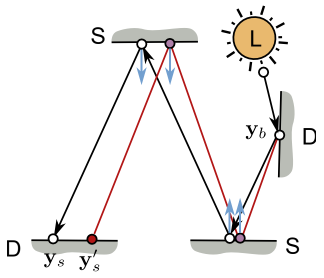


STEP 3: SHIFT THE REST OF THE LIGHT PATH

Case 1: Diffuse parent



Case 2: "Specular" parent



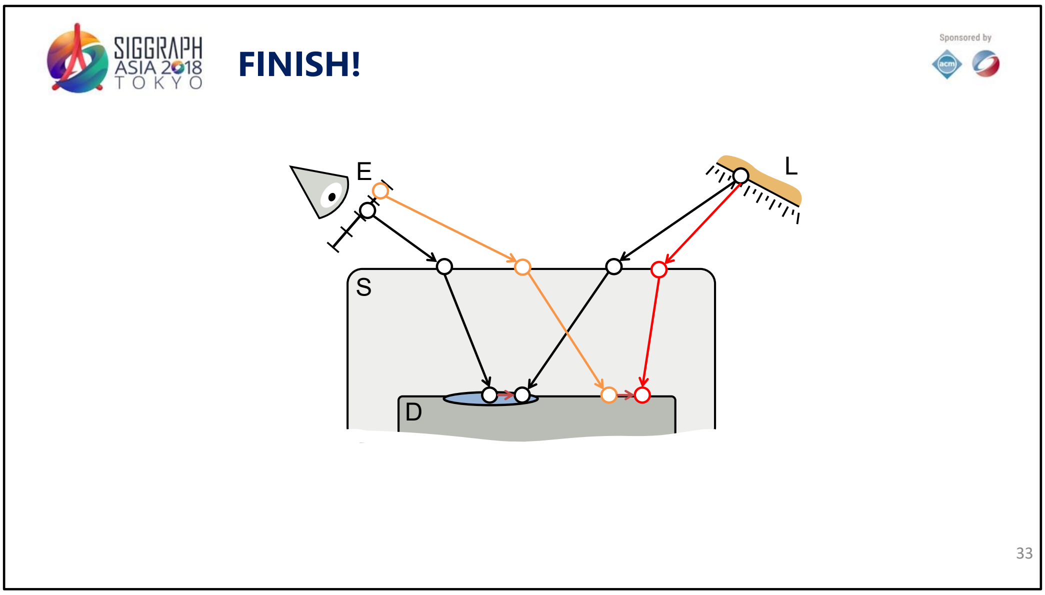
Find a valid light path using Manifold Exploration [Jakob et al. 2012]

32

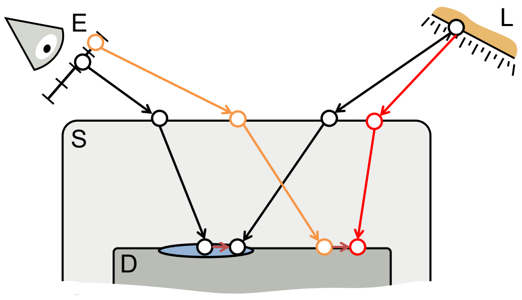
In this step, we have two cases.

The first case is when the parent photon is lying on a diffuse or mostly diffuse surface. In case, we can directly do a reconnection.

The second case is when the parent photon is on a specular or nearly specular surface. In this case, we have to construct a path between the offset photon location and the first diffuse parent, here denoted as y_b. These two points are our constraints and we can use manifold exploration [Jakob et al., 2012] to order to generate the vertices in between.

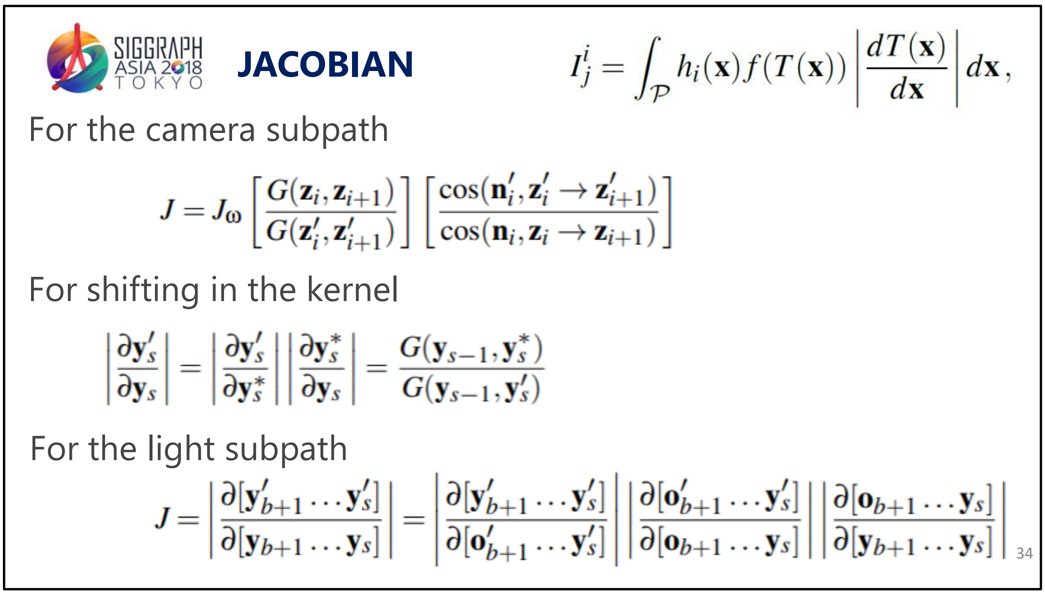


FINISH!



33

After these three steps, we have generated a pair of base and offset path for estimating the image-space gradients.



JACOBIAN

$$I_j^i = \int_{\mathcal{P}} h_i(\mathbf{x}) f(T(\mathbf{x})) \left| \frac{dT(\mathbf{x})}{d\mathbf{x}} \right| d\mathbf{x},$$

For the camera subpath

$$J = J_\omega \left[\frac{G(\mathbf{z}_i, \mathbf{z}_{i+1})}{G(\mathbf{z}'_i, \mathbf{z}'_{i+1})} \right] \left[\frac{\cos(\mathbf{n}'_i, \mathbf{z}'_i \to \mathbf{z}'_{i+1})}{\cos(\mathbf{n}_i, \mathbf{z}_i \to \mathbf{z}_{i+1})} \right]$$

For shifting in the kernel

$$\left| \frac{\partial \mathbf{y}'_s}{\partial \mathbf{y}_s} \right| = \left| \frac{\partial \mathbf{y}'_s}{\partial \mathbf{y}^*_s} \right| \left| \frac{\partial \mathbf{y}^*_s}{\partial \mathbf{y}_s} \right| = \frac{G(\mathbf{y}_{s-1}, \mathbf{y}^*_s)}{G(\mathbf{y}_{s-1}, \mathbf{y}'_s)}$$

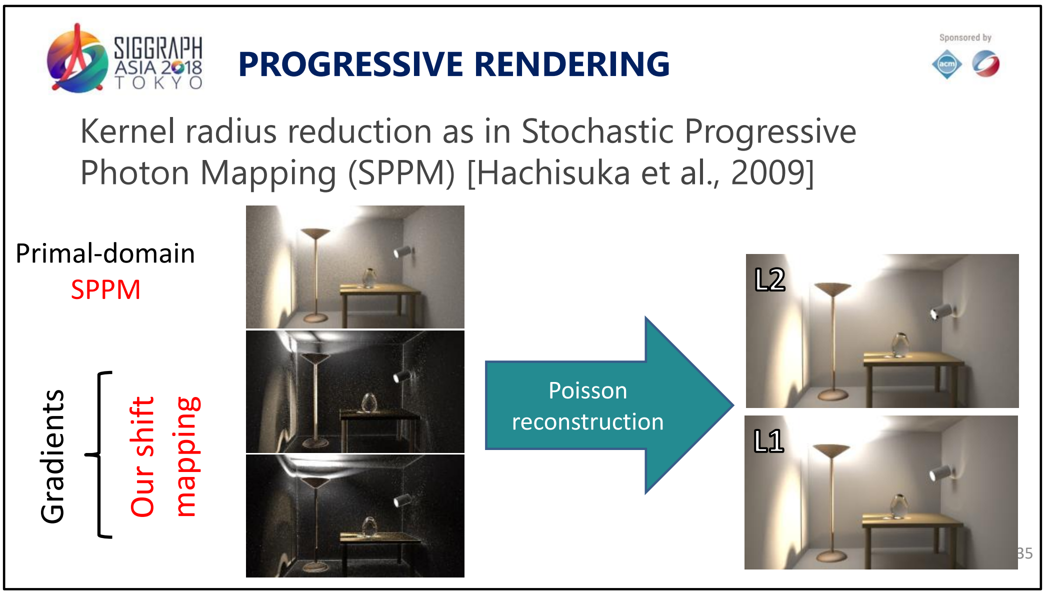
For the light subpath

$$J = \left| \frac{\partial[\mathbf{y}'_{b+1} \cdots \mathbf{y}'_s]}{\partial[\mathbf{y}_{b+1} \ldots \mathbf{y}_s]} \right| = \left| \frac{\partial[\mathbf{y}'_{b+1} \cdots \mathbf{y}'_s]}{\partial[\mathbf{o}'_{b+1} \cdots \mathbf{y}'_s]} \right| \left| \frac{\partial[\mathbf{o}'_{b+1} \cdots \mathbf{y}'_s]}{\partial[\mathbf{o}_{b+1} \ldots \mathbf{y}_s]} \right| \left| \frac{\partial[\mathbf{o}_{b+1} \cdots \mathbf{y}_s]}{\partial[\mathbf{y}_{b+1} \cdots \mathbf{y}_s]} \right|$$

34

One thing we haven't mentioned so far is the Jacobian. To recall, the Jacobian is the mathematical formulation of the change of measure due to the shift mapping. Having a correct Jacobian is important to get correct gradient values.

The details of the formula could be founded in the corresponding paper [Hua et al., 2017]. The basic idea is that the Jacobian is the product of the Jacobian for the camera subpath, the kernel, and the light subpath, so we will find the Jacobian of each part independently. The Jacobian for the camera subpath is mostly similar to the Jacobian used in gradient-domain path tracing [Kettunen et al., 2015] which is mainly based on half-vector copy. The Jacobian for the light subpath is based on manifold exploration [Jakob et al., 2012]. The Jacobian for the kernel shift is the ratio of two geometry terms due to the projection of the photon to the surface.

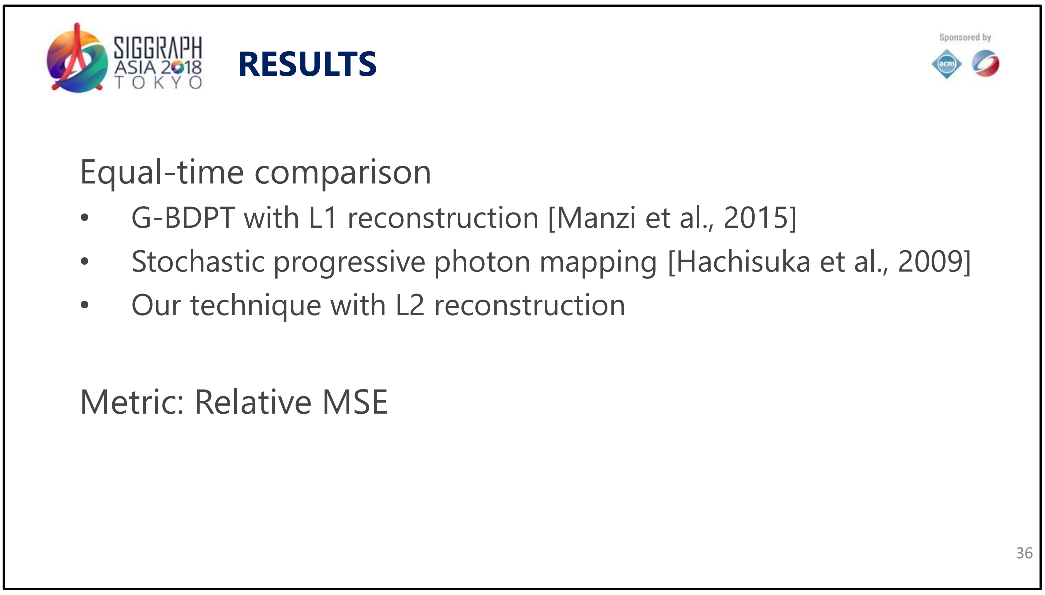


PROGRESSIVE RENDERING

Kernel radius reduction as in Stochastic Progressive Photon Mapping (SPPM) [Hachisuka et al., 2009]

So now, to summarize, to estimate the gradients, we use a special hybrid shift mapping.

In order to get a consistent estimator, we can use the same reduction rate as stochastic progressive photon mapping [Hachisuka et al., 2009].



RESULTS

Equal-time comparison

- G-BDPT with L1 reconstruction [Manzi et al., 2015]
- Stochastic progressive photon mapping [Hachisuka et al., 2009]
- Our technique with L2 reconstruction

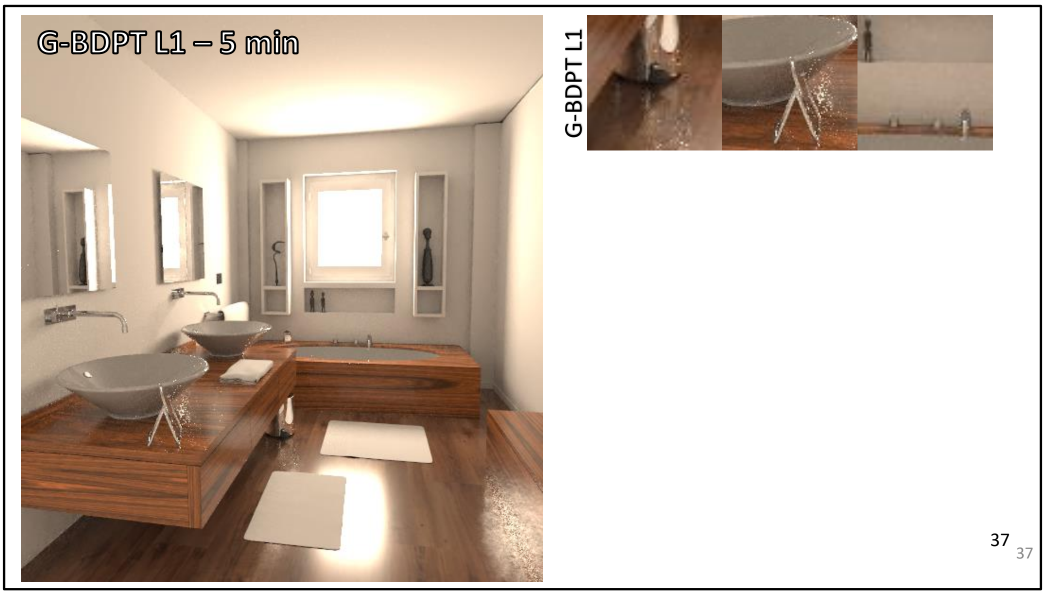
Metric: Relative MSE

36

Let us now see the equal-time comparisons. We will compare with gradient-domain bidirectional path tracing with L1 reconstruction. We choose to show L1 reconstruction as L2 reconstruction has a lot of dipole artefact, and the original stochastic progressive photon mapping.

For our method, we show the results in L2 reconstruction. In fact, we found that if the gradients are robust computed, an L2 reconstruction is enough. In this particular experiment, we only use L1 in some challenging cases.

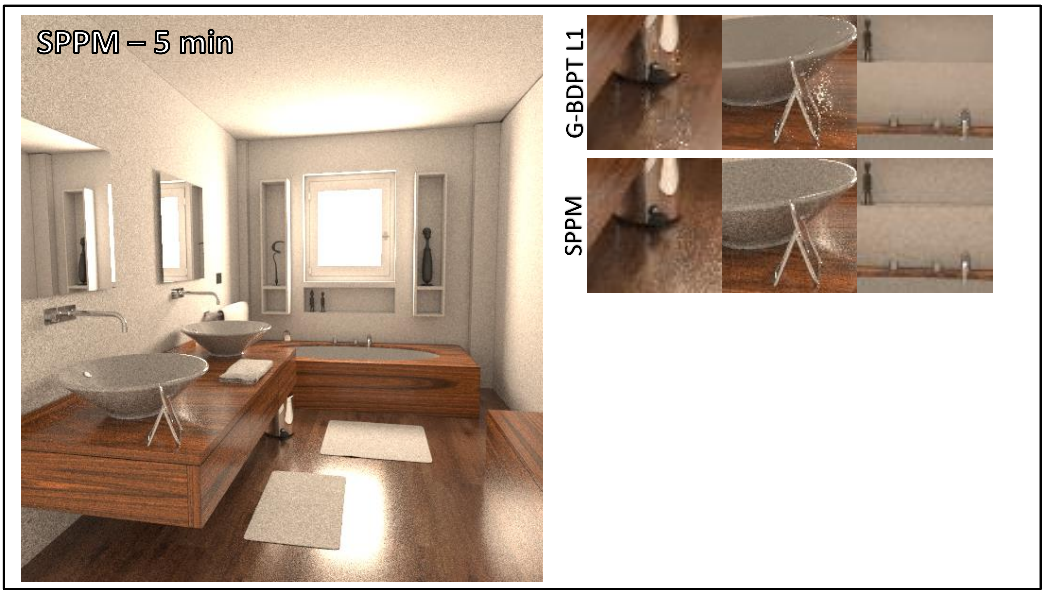
The metric that we will use is relative MSE.



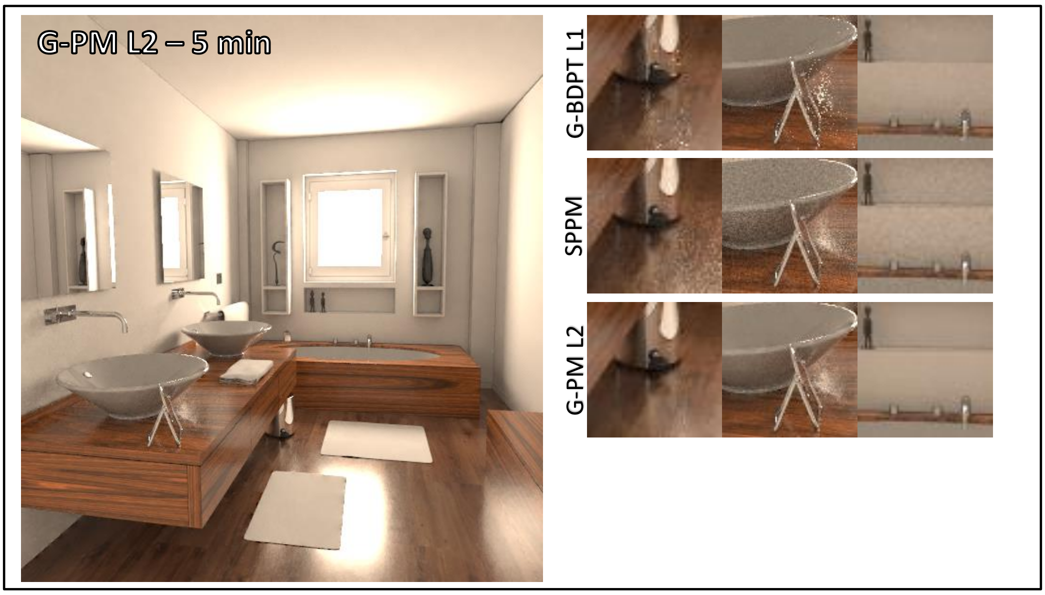
37

This is the bathroom scene with 5 minutes of rendering. We can see G-BDPT produces clean results for most parts of the image, e.g., the wall that receives a lot of diffuse light transport.

However, for complex light transport, like the specular trash bin with the glossy floor or the caustic, G-BDPT have a lot of noise (see the zoom-in patches). It is due to the fact that BDPT cannot really handle efficiently these paths.

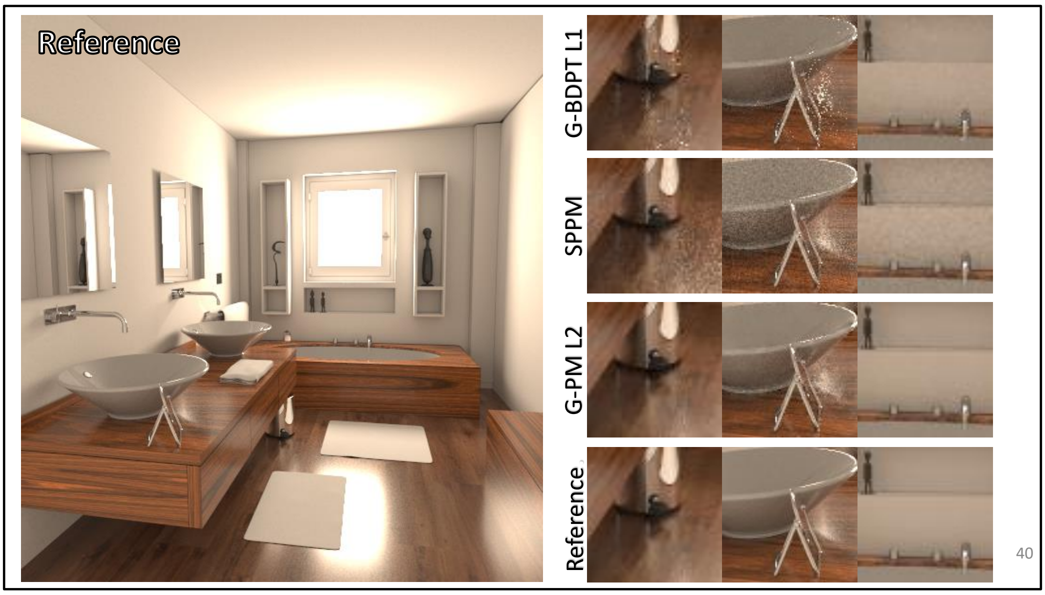


In comparison, SPPM produces more noise in all image parts. This is especially true for the diffuse/diffuse light transport where the photon are incoherent and produce a lot of noise.
By contrast, for the complex light transport, this technique doesn't have any "spike" problem compared to G-BDPT with L1 reconstruction.

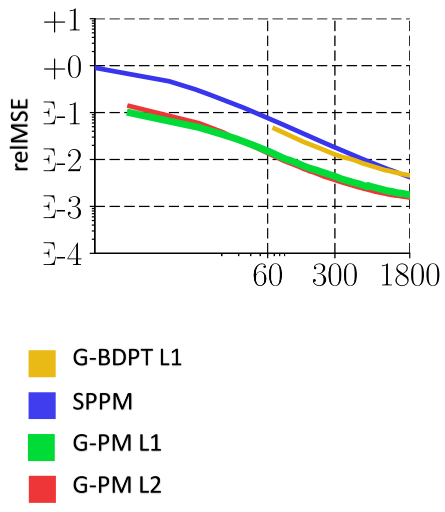


In gradient-domain photon density estimation (G-PM – just to make it shorter than G-SPPM), most of the noise is gone. It is especially visible for diffuse/diffuse light transport where we get similar results compare to G-BDPT.

This technique also works well for the specular light transport regions. With the extra denoising capability due to gradients and reconstruction, we are also better than SPPM.

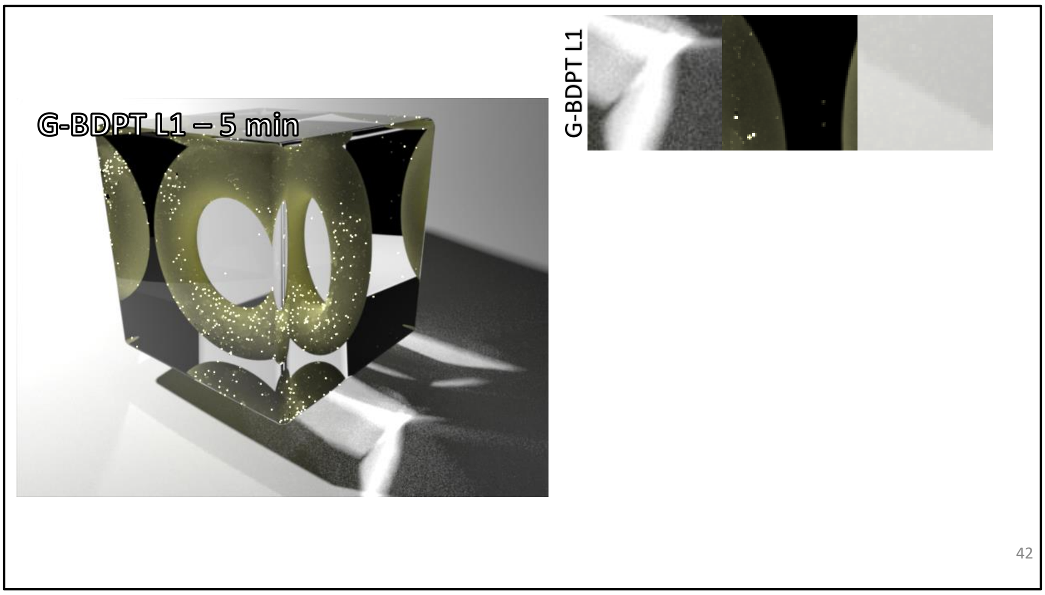


Compare to the reference image, our technique is quite similar but only requires 5 minutes of rendering.

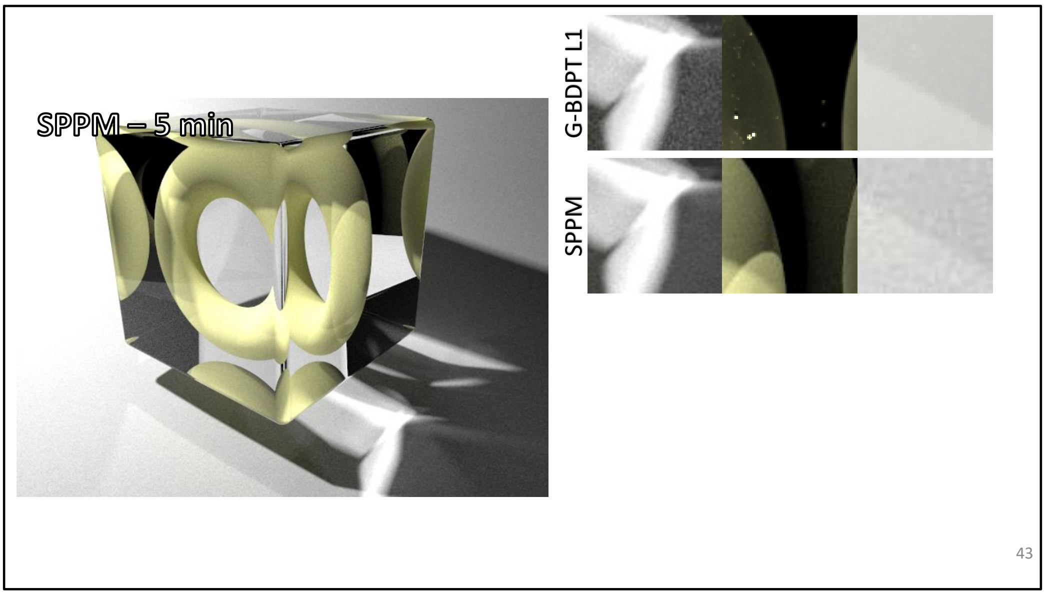


Let us see the convergence plot. SPPM and G-BDPT have the same convergence.
It is due to the fact that SPPM is better for complex light transport but G-BDPT have a better handling of diffuse/diffuse light transport.

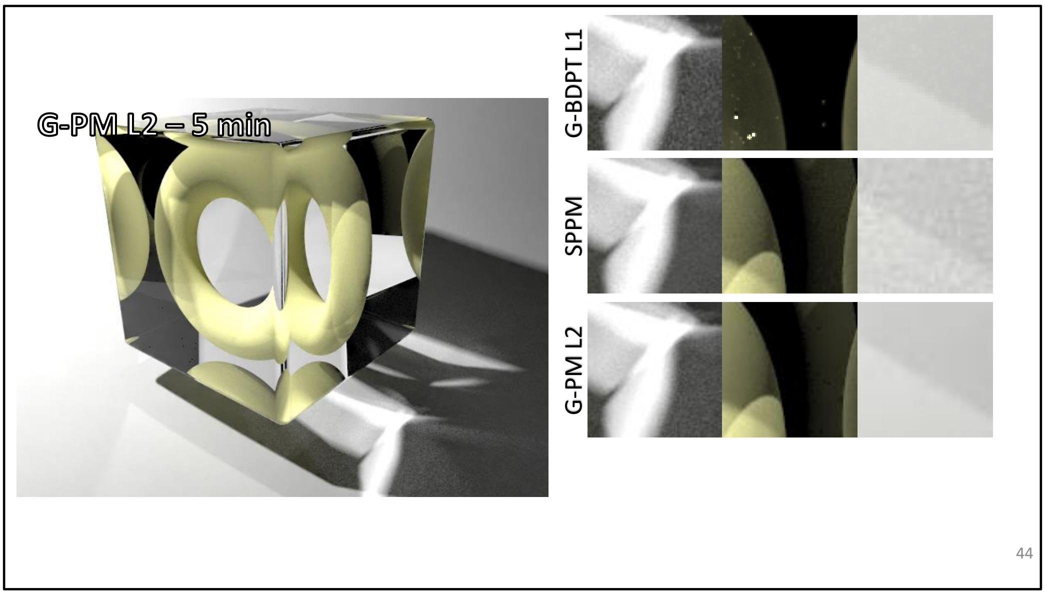
In G-PM, as it robustly handles all light transport, we get great quality improvement.



Here is another example with SDS paths dominates the light transport.
In this case, we expect that G-BDPT fails except for simple refraction or direct lighting.

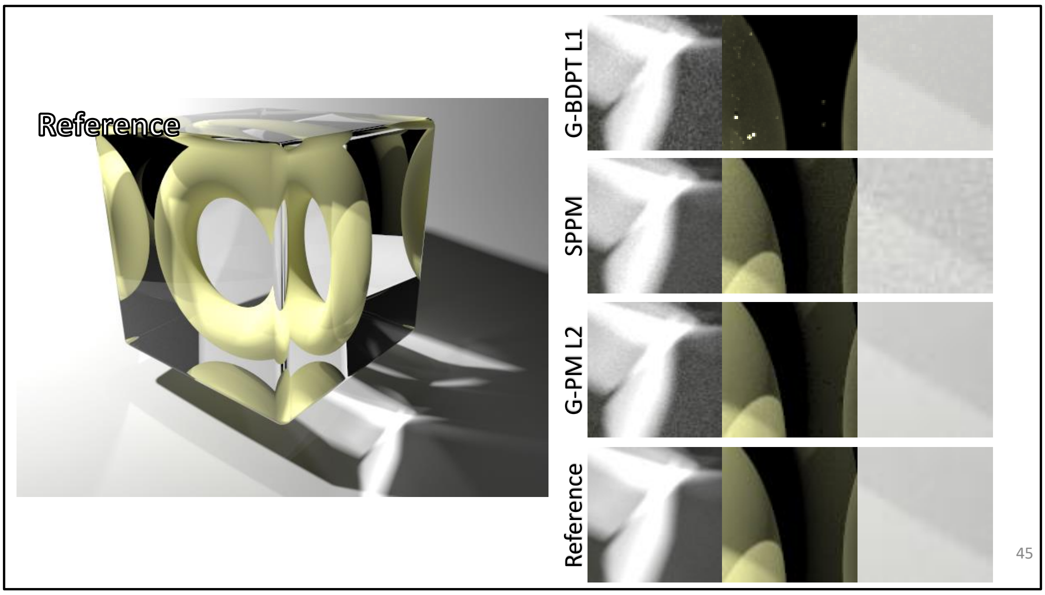


In comparison, as the scene have a lot of SDS path, SPPM handles them quite well despite some uniform noise in the image.

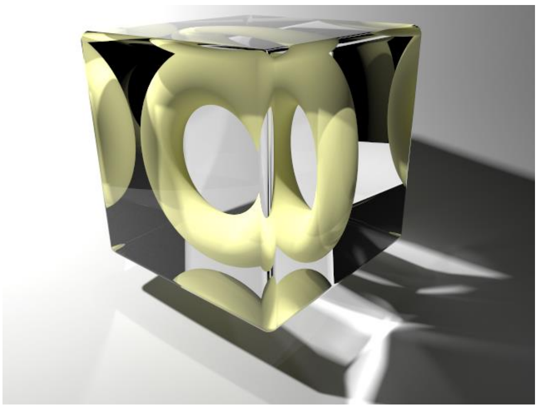


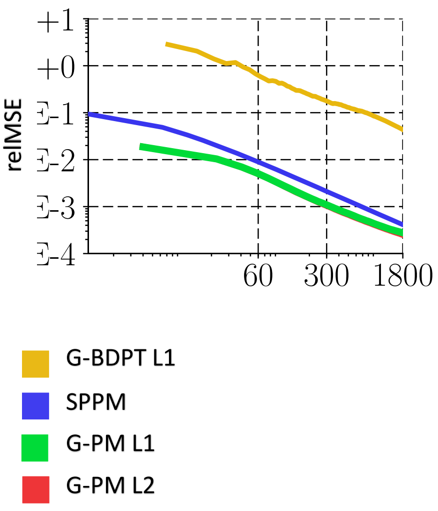
G-PM makes it possible to remove all the noise in the SDS path and direct path (second and third column).

However, for the difficult caustic (first column), the noise is slightly higher as here the light path are complicated and manifold exploration failed to generate the correct offset light path. The results is slightly worse than classical SPPM in this region.



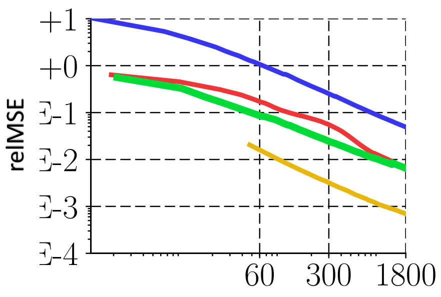
Despite that, G-PM is still the closest to the reference.





In terms of convergence, G-PM still improves over SPPM.

Sponza



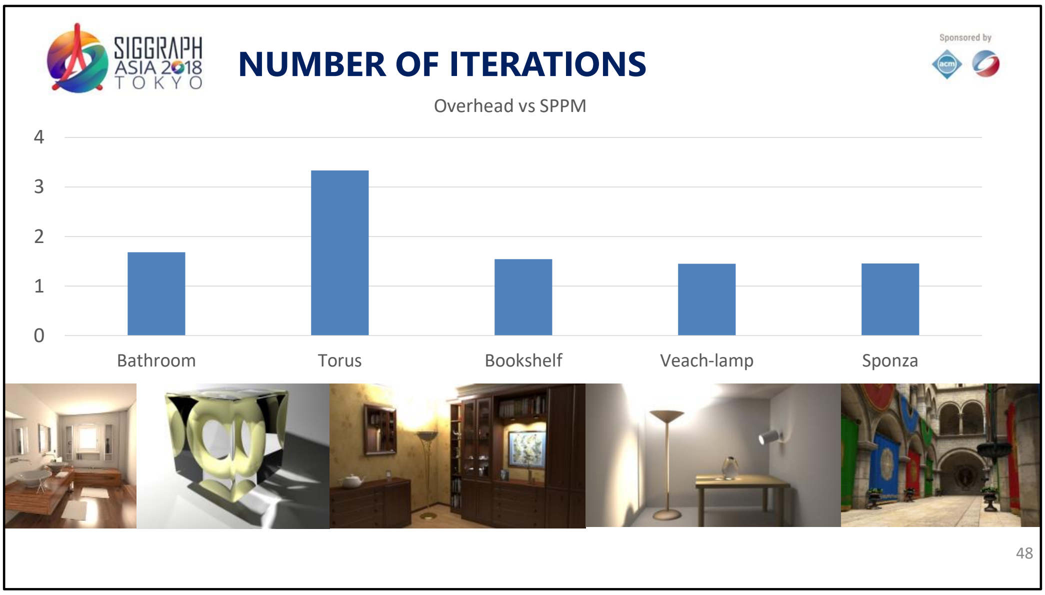
🟨 G-BDPT L1

🟦 SPPM

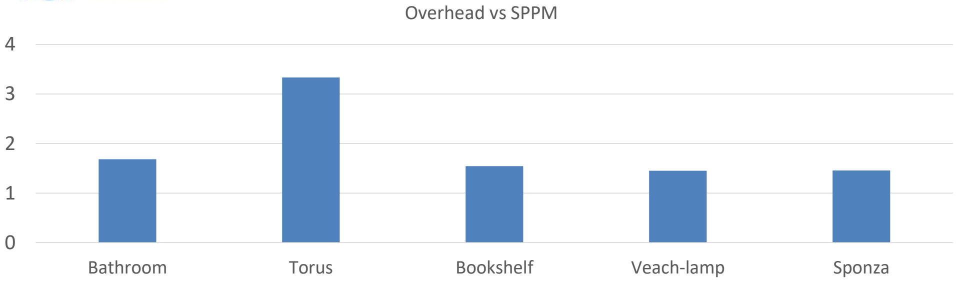
🟩 G-PM L1

🟥 G-PM L2

The Crytek Sponza scene is a diffuse scene where there is not specular material. In this scene, G-BDPT doesn't have any issue to handle diffuse/diffuse light transport and generate a good image rapidly. As SPPM doesn't ensure stratification, the noise is high as the distribution of photon is not uniform. However, our technique makes it possible to filter the noise and give a good speedup in term of relative MSE. However, G-BDPT does a better job here as it has a better stratification and sampling strategy.



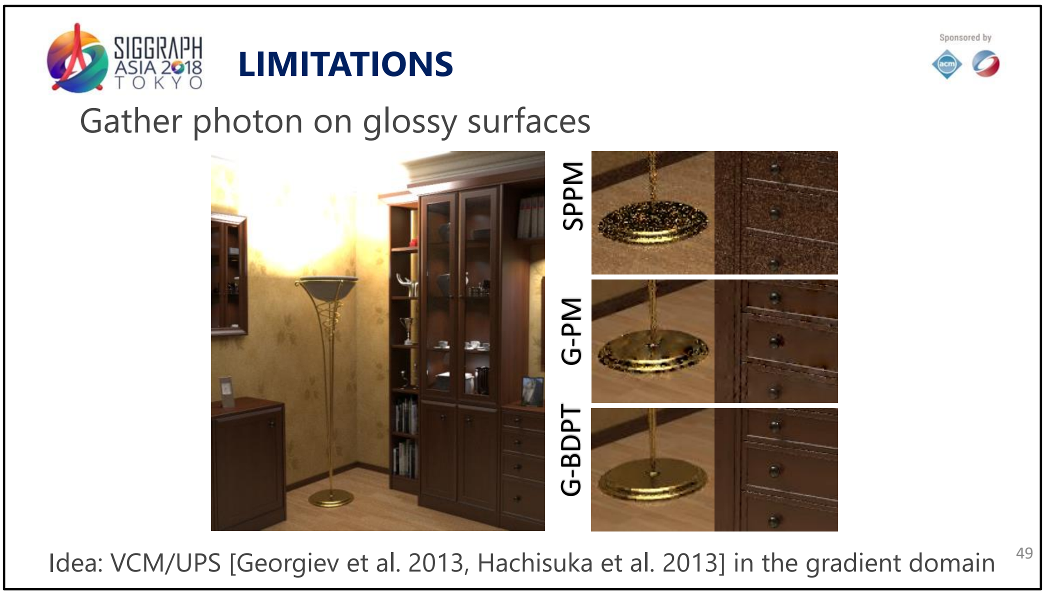
NUMBER OF ITERATIONS



48

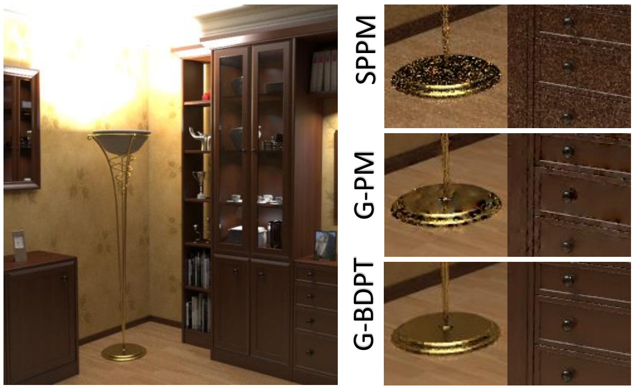
Here is a crude measurement of overhead by counting the number of photon pass in G-PM and SPPM. We see that the overhead is between 1.3 to 3.3, meaning that we pay more cost in each photon pass in G-PM for gradient computation. In return, we achieve better convergence as shown before.

In fact, a costly step in gradient computation is manifold exploration that generates the specular and glossy vertex chains. The variation of the overhead is due to the percentage of this manifold exploration in shift mapping (see Torus scene).



LIMITATIONS

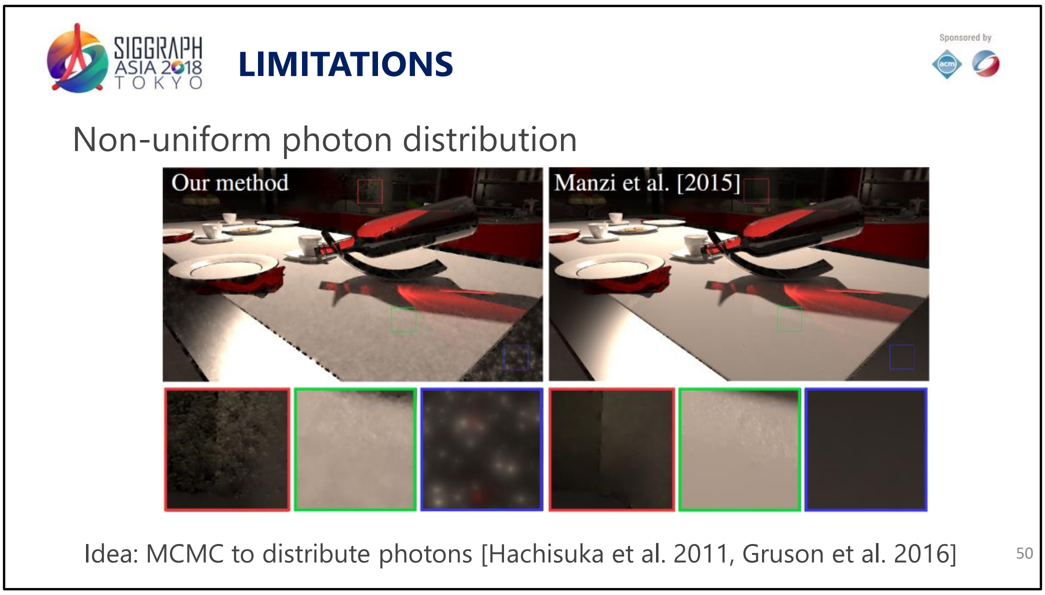
Gather photon on glossy surfaces



Idea: VCM/UPS [Georgiev et al. 2013, Hachisuka et al. 2013] in the gradient domain

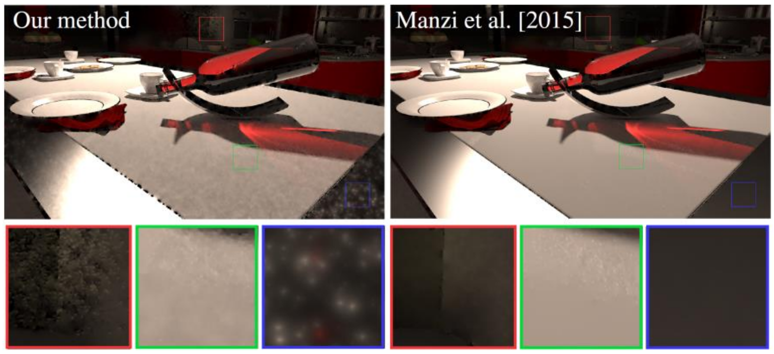
49

A limitation of G-PM is that it is not as strong as G-BDPT in handling glossy surfaces. A possible solution is to make gradient-domain vertex connection and merging to unify G-BDPT and G-PM into a single framework.



LIMITATIONS

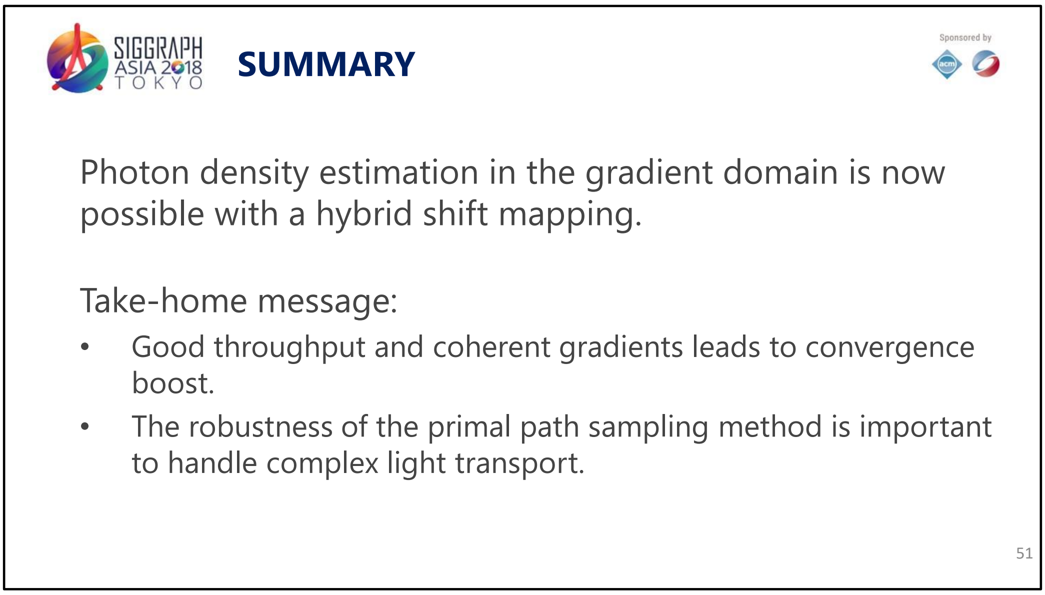
Non-uniform photon distribution



Idea: MCMC to distribute photons [Hachisuka et al. 2011, Gruson et al. 2016]

50

Another limitation is that photon distribution is random which limits energy contribution in highly occluded regions. Using Markov chain Monte Carlo techniques to distribute the photons could be a possible solution to focus photons to important areas.



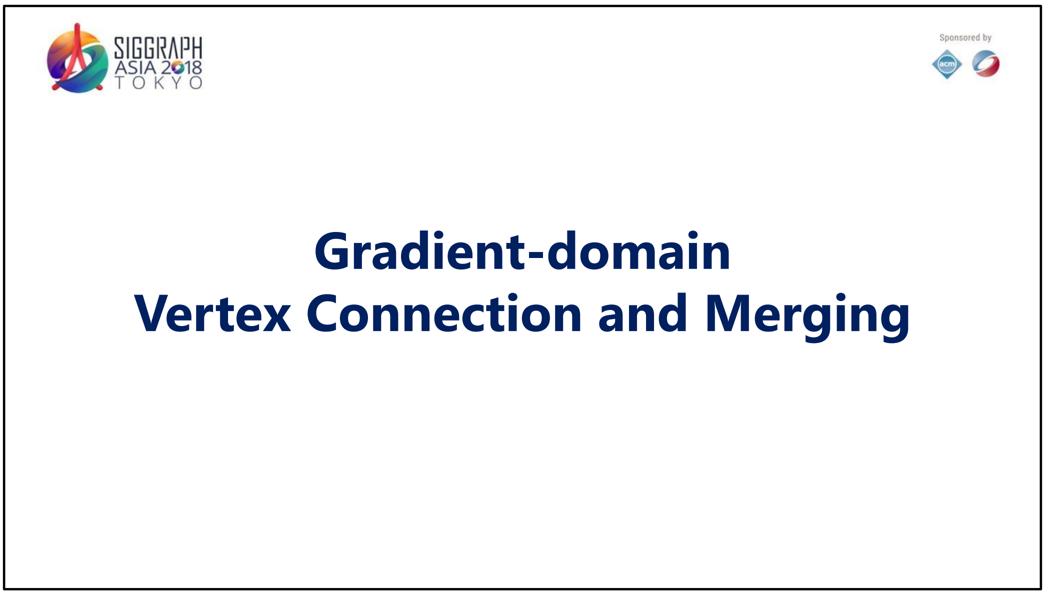
SUMMARY

Photon density estimation in the gradient domain is now possible with a hybrid shift mapping.

Take-home message:

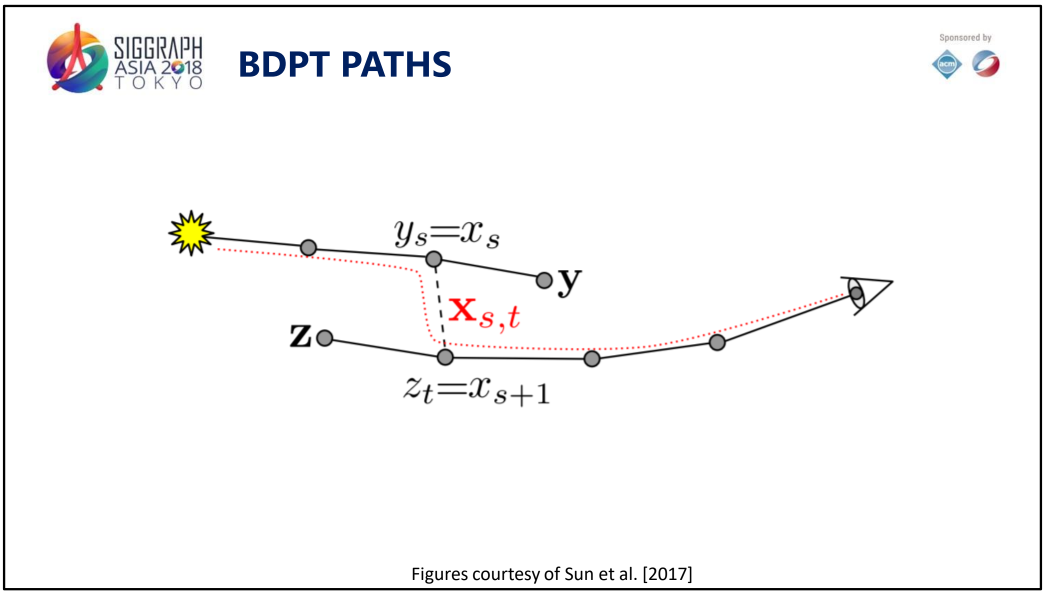
- Good throughput and coherent gradients leads to convergence boost.
- The robustness of the primal path sampling method is important to handle complex light transport.

51

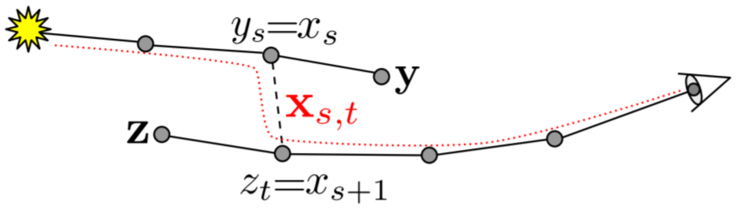


Gradient-domain Vertex Connection and Merging

Let us now do a quick look at combining gradient-domain bidirectional path tracing and photon density estimation into the same framework.

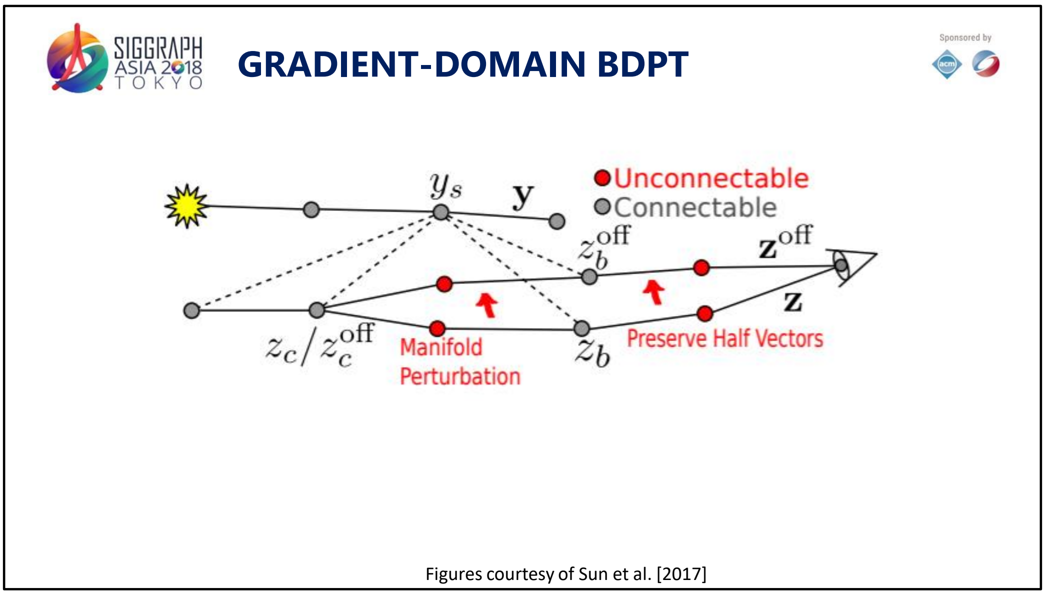


BDPT PATHS

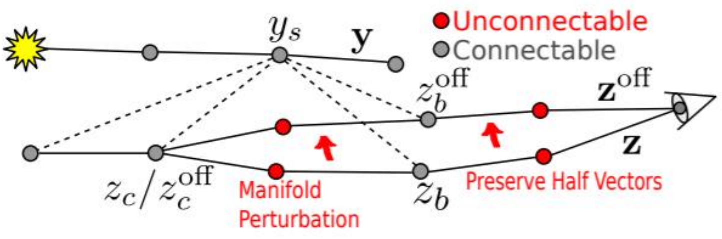


Figures courtesy of Sun et al. [2017]

BDPT performs vertex-vertex connection to establish a complete path.



GRADIENT-DOMAIN BDPT



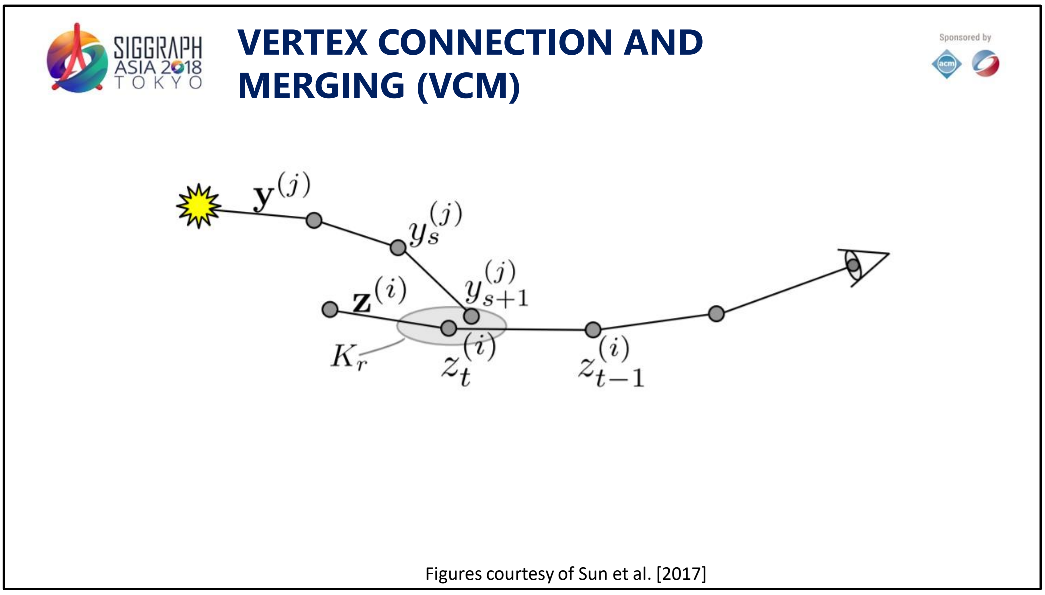
Figures courtesy of Sun et al. [2017]

G-BDPT further classifies each vertex to be connectable or unconnectable. Only connectable pair of vertices are considered in shift mapping.

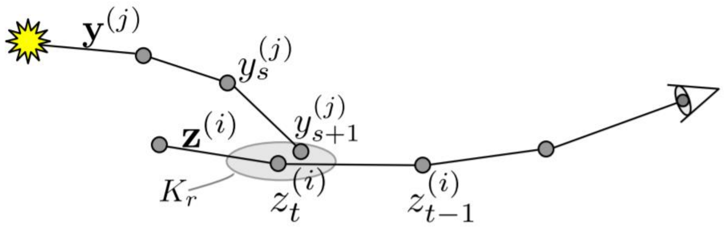
Consider an example camera subpath starting from the eye. As in the figure, the first connectable vertex is z_b and the second connectable vertex is z_c.

First, we generate the offset camera path such that the base and offset camera path shares the same connectable z_c vertex. This can be done either by performing a manifold exploration between z_b and z_c if necessary.

After that, each pair of connectable vertices between the light and camera subpath are considered to form complete paths. Shift mapping is then performed.

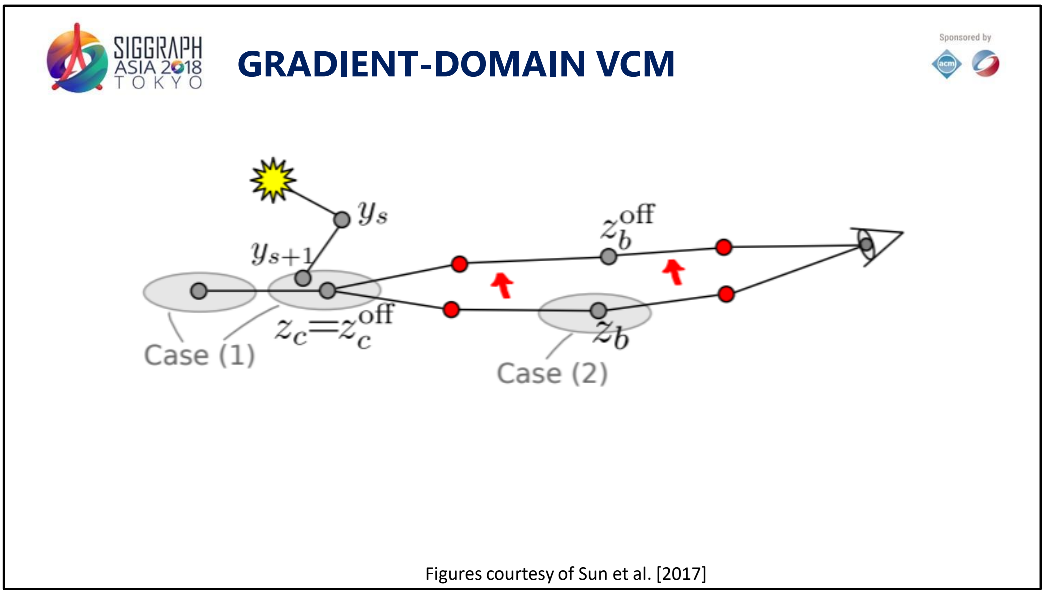


VERTEX CONNECTION AND MERGING (VCM)

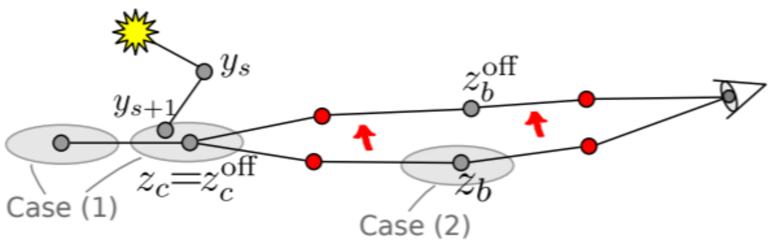


Figures courtesy of Sun et al. [2017]

Vertex connection and merging unifies Monte Carlo and density estimation paths into the same (extended) path space [Georgiev et al. and Hachisuka et al., 2012].



GRADIENT-DOMAIN VCM



Figures courtesy of Sun et al. [2017]

Let us now consider density estimation on top of G-BDPT to achieve G-VCM. Similar to G-BDPT, we will only perform density estimation at connectable vertices. (Density estimation is also known as vertex merging in VCM).

As in the figure, the first connectable vertex is z_b and the second connectable vertex is z_c. Therefore, the density estimation can occur at or after z_c (case 1), and at z_b (case 2).

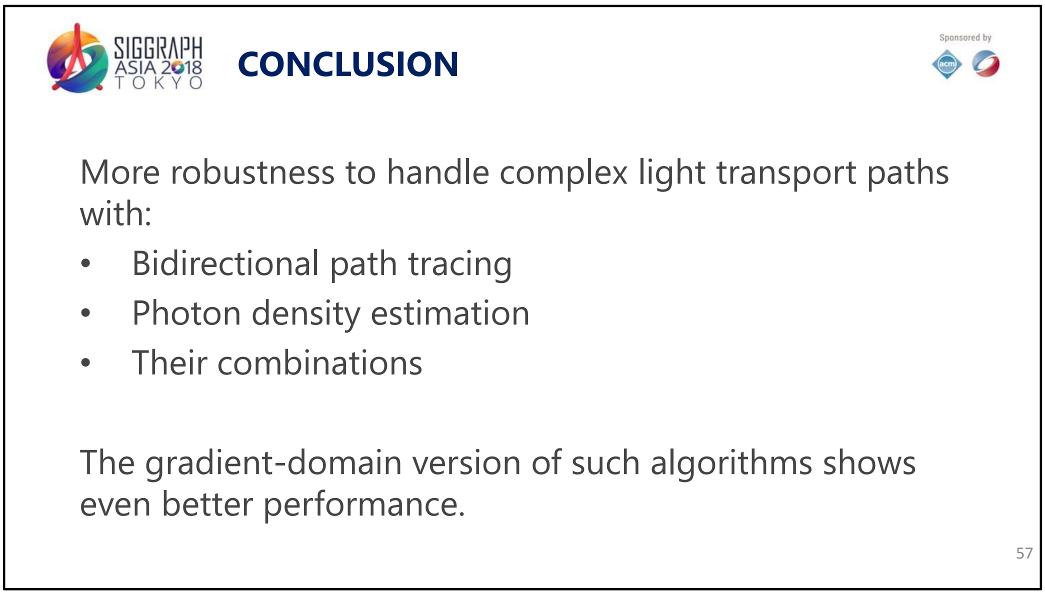
Following G-BDPT, we generate the offset camera path such that the base and offset camera path shares the same connectable z_c vertex.

Given the base and offset camera path, let us now handle the density estimation:

Case 1: if the density estimation event happens at z_c, the offset light subpath is the same as the base light subpath. This is almost similar to G-BDPT except that the event between z_c and y_{s+1} is now a vertex merging instead of vertex connection.

Case 2: if the density estimation event happens at z_b, the shift mapping is similar to gradient-domain photon density estimation.

The remaining issue in this algorithm is to support multiple importance sampling weights that consider both density-based and Monte Carlo based paths. See the papers by Georgiev et al. and Hachisuka et al. [2012] and Sun et al. [2017] for the maths.



CONCLUSION

More robustness to handle complex light transport paths with:

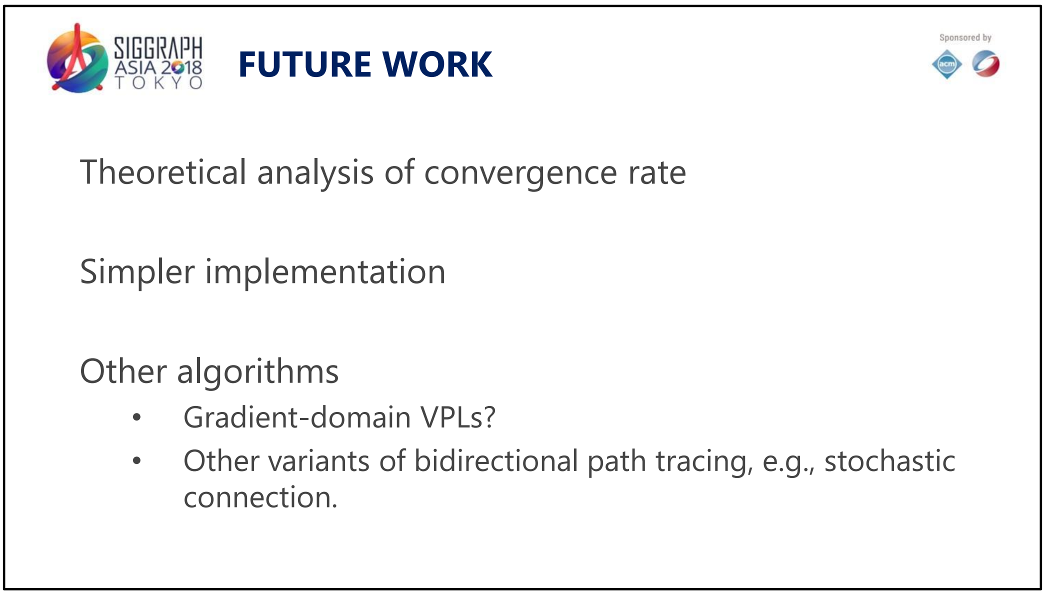
- Bidirectional path tracing
- Photon density estimation
- Their combinations

The gradient-domain version of such algorithms shows even better performance.

57

In this section, we explored two bidirectional light transport algorithms for gradient-domain rendering: bidirectional path tracing and photon density estimation. A combination of them is also possible with gradient-domain vertex connection and merging.

We show that adding gradient estimation and image reconstruction to such algorithms show a boost in convergence.



FUTURE WORK

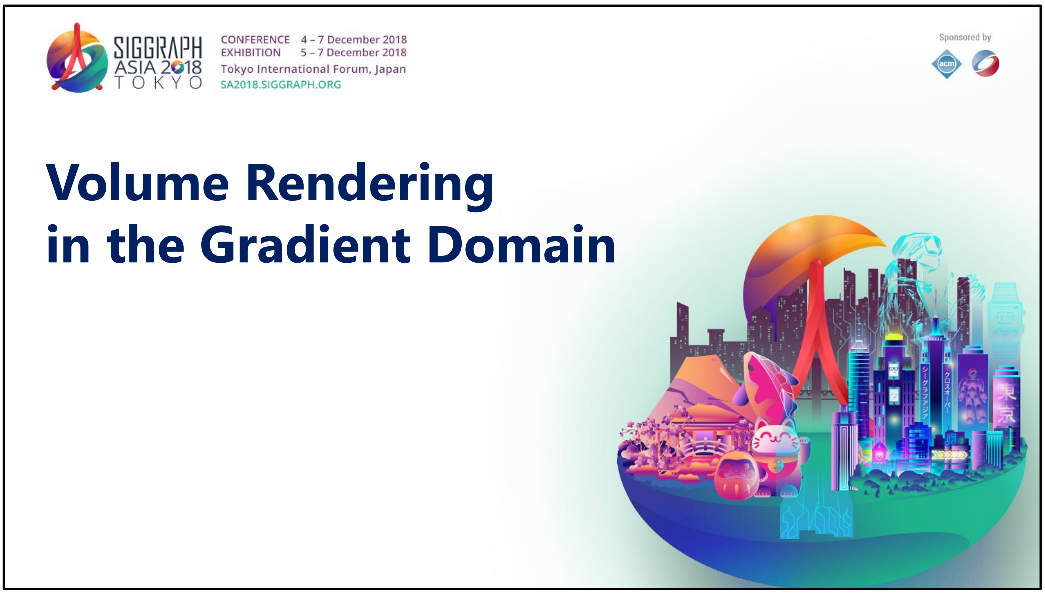
Theoretical analysis of convergence rate

Simpler implementation

Other algorithms

- Gradient-domain VPLs?
- Other variants of bidirectional path tracing, e.g., stochastic connection.

4.5 Gradient-domain volumetric rendering



SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

Sponsored by

Volume Rendering in the Gradient Domain

Motivation

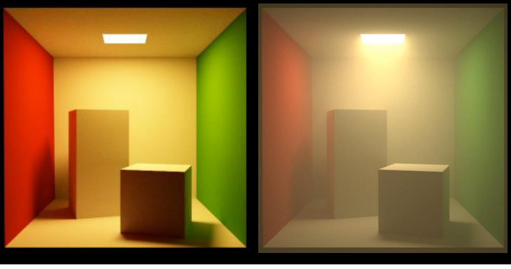
In this work, we are interested in rendering this type of scenes where we have complex light transport with the presence of participating media.

Motivation

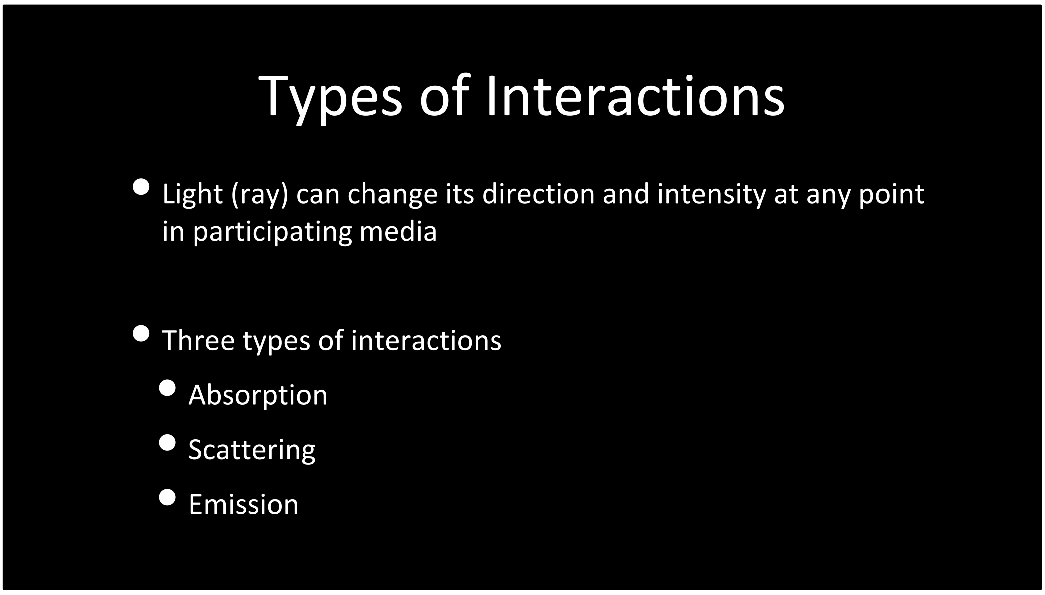
More specifically, we are interested only in improving the rendering performance of the participating media part.

Participating Media

- We assumed light interacts only with surfaces
- Relaxing this assumption = participating media
  - Light can interact with any point in the space

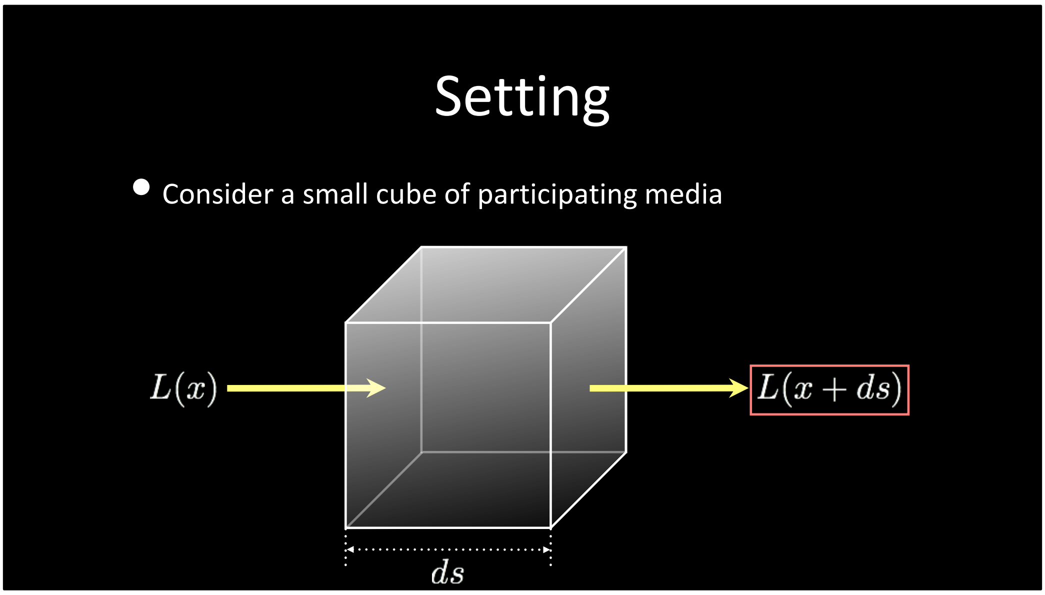


So far we only assumed light interactions with surfaces. Let’s do a recap of rendering with participating media in the primal domain before going back to gradient-domain rendering.



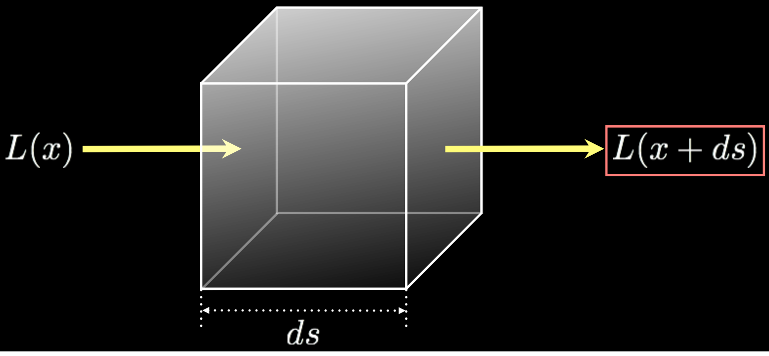
Types of Interactions

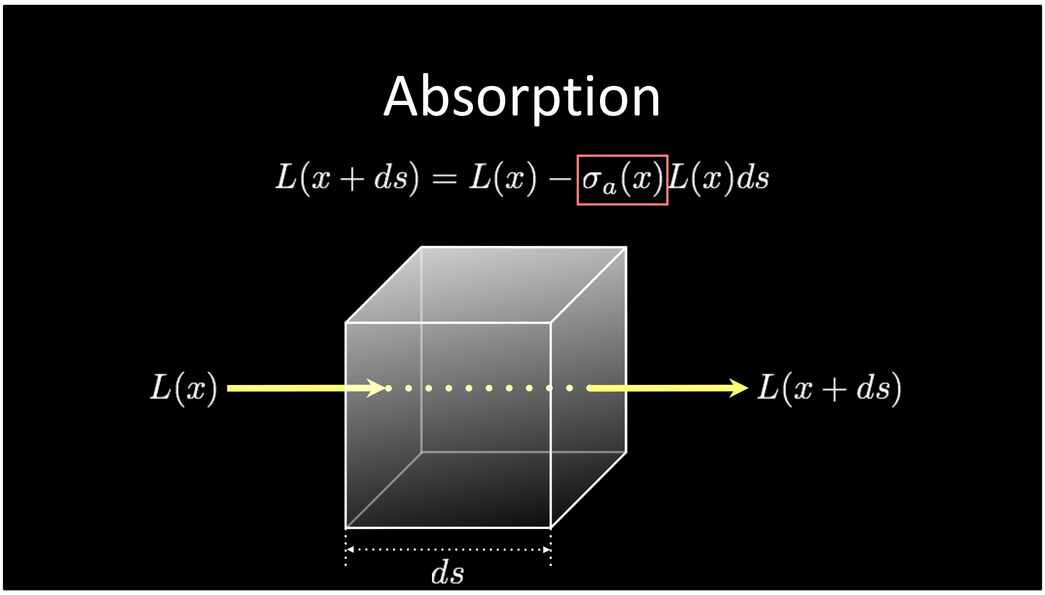
- Light (ray) can change its direction and intensity at any point in participating media
- Three types of interactions
  - Absorption
  - Scattering
  - Emission



Setting

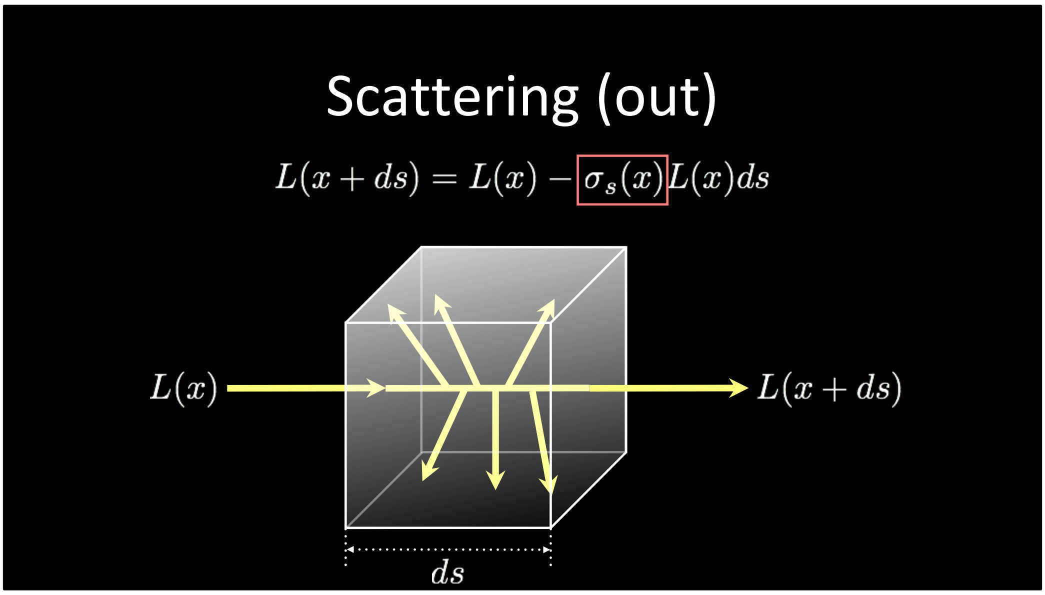
- Consider a small cube of participating media





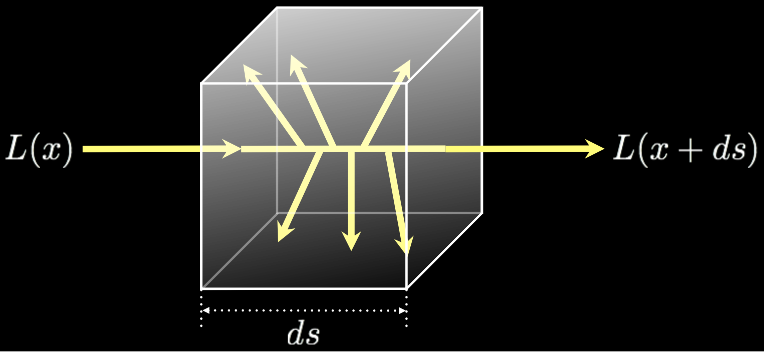
Absorption

$$L(x+ds) = L(x) - \boxed{\sigma_a(x)} L(x) ds$$



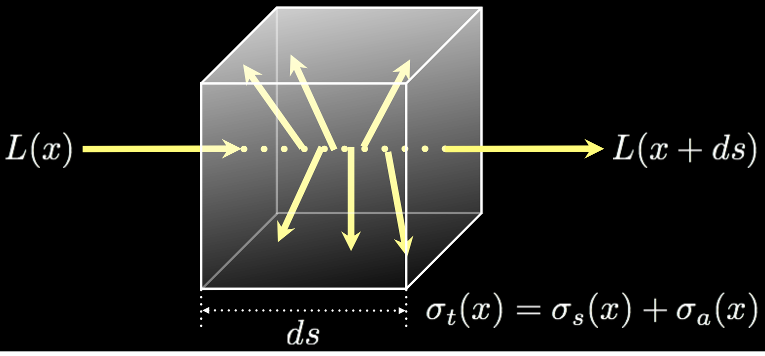
Scattering (out)

$$L(x+ds) = L(x) - \boxed{\sigma_s(x)} L(x) ds$$



Extinction

$$L(x+ds) = L(x) - \boxed{\sigma_t(x)} L(x) ds$$

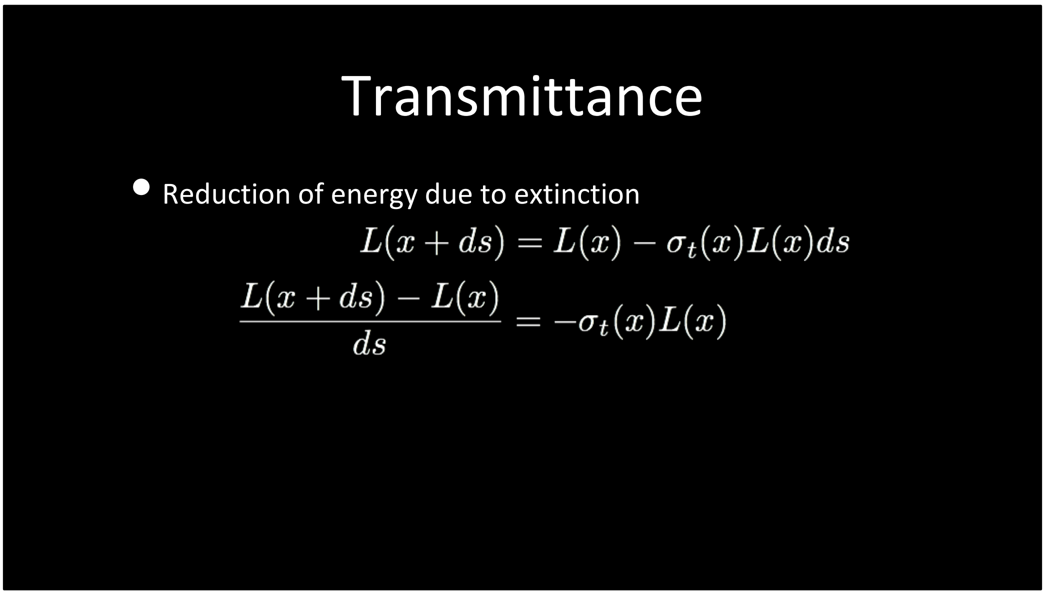


$$\sigma_t(x) = \sigma_s(x) + \sigma_a(x)$$

Transmittance

- Reduction of energy due to extinction

$$L(x+ds) = L(x) - \sigma_t(x)L(x)ds$$

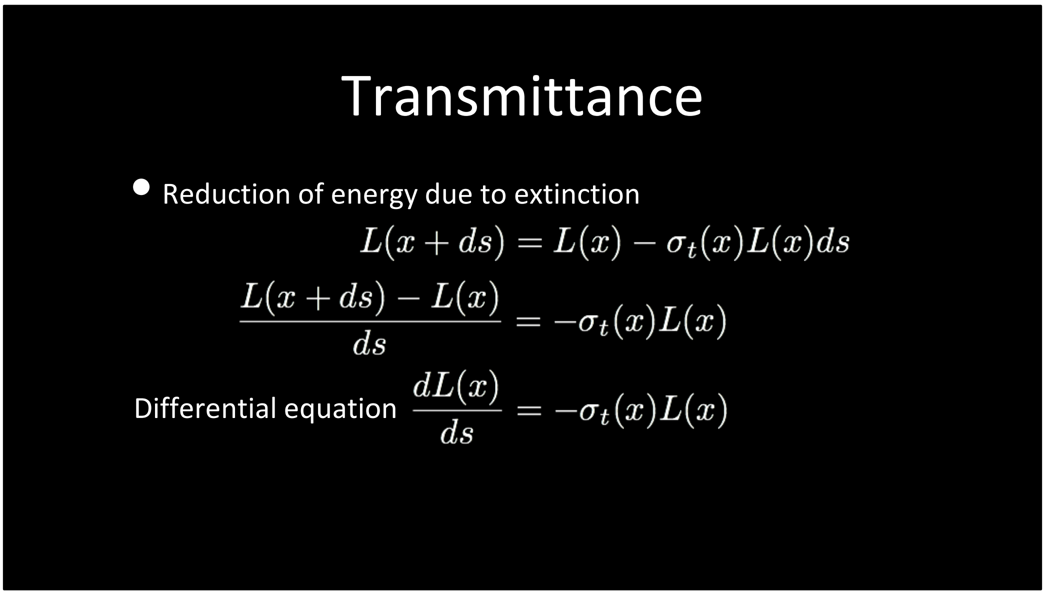


Transmittance

- Reduction of energy due to extinction

$$L(x+ds) = L(x) - \sigma_t(x)L(x)ds$$

$$\frac{L(x+ds)-L(x)}{ds} = -\sigma_t(x)L(x)$$



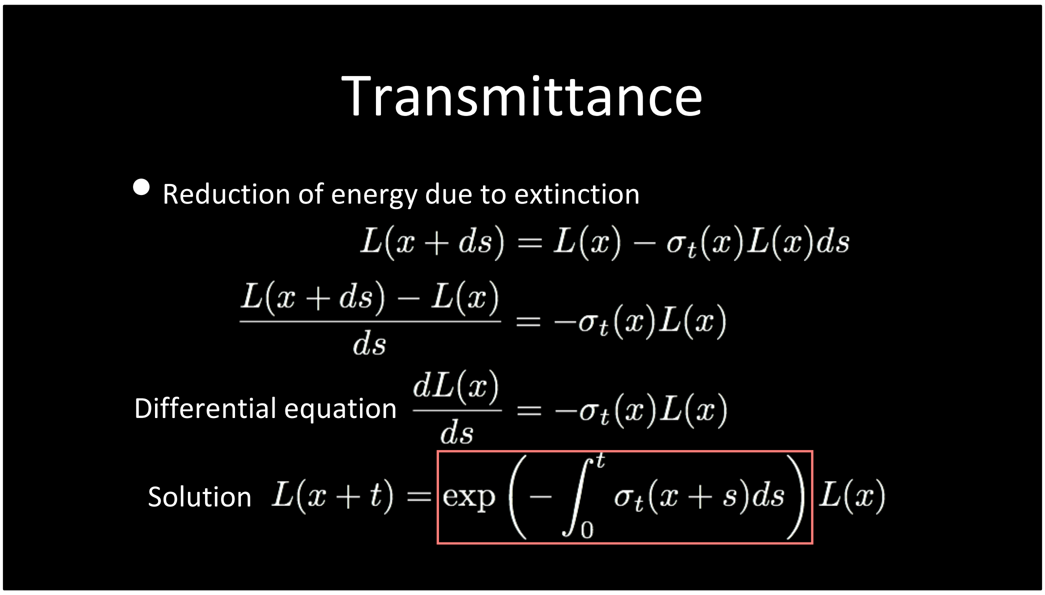
Transmittance

- Reduction of energy due to extinction

$$L(x+ds) = L(x) - \sigma_t(x)L(x)ds$$

$$\frac{L(x+ds) - L(x)}{ds} = -\sigma_t(x)L(x)$$

Differential equation $\dfrac{dL(x)}{ds} = -\sigma_t(x)L(x)$



Transmittance

- Reduction of energy due to extinction

$$L(x+ds) = L(x) - \sigma_t(x)L(x)ds$$

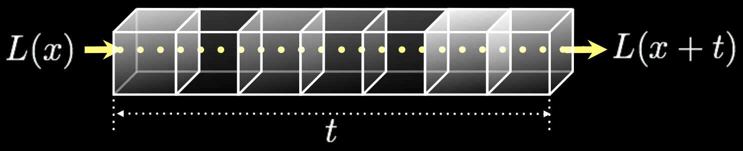
$$\frac{L(x+ds) - L(x)}{ds} = -\sigma_t(x)L(x)$$

Differential equation $\frac{dL(x)}{ds} = -\sigma_t(x)L(x)$

Solution $L(x+t) = \boxed{\exp\left(-\int_0^t \sigma_t(x+s)ds\right)} L(x)$

Transmittance

$$L(x+t) = T(x, x+t)L(x)$$
$$= \exp\left(-\int_0^t \sigma_t(x+s)ds\right) L(x)$$

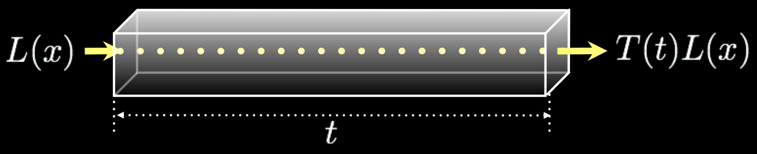


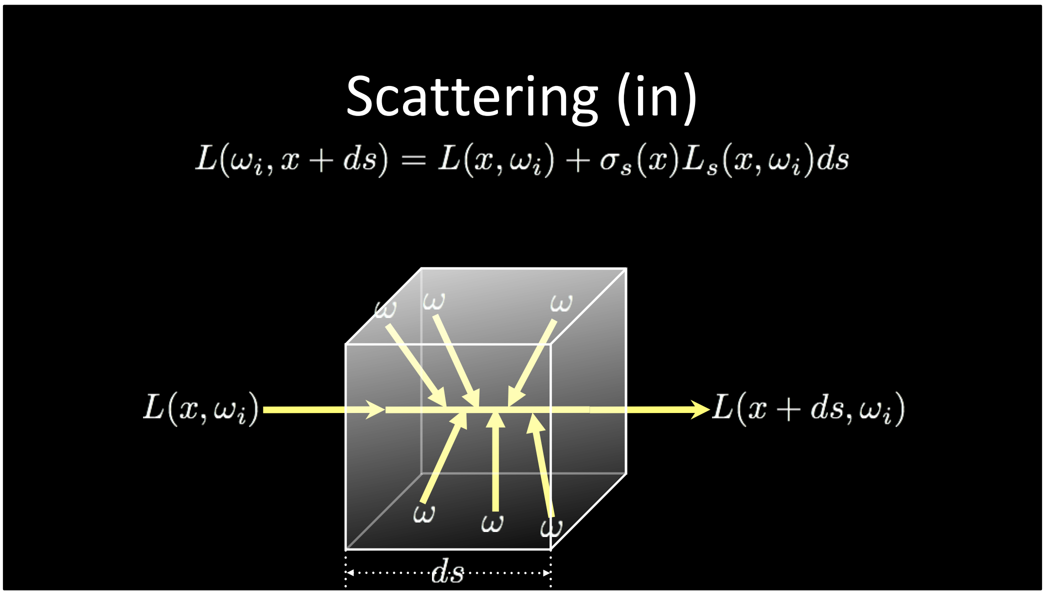
The integral can be solved by Monte Carlo integration

Beer-Lambert Law

- If the media is homogeneous
  - Spatially uniform extinction coefficient

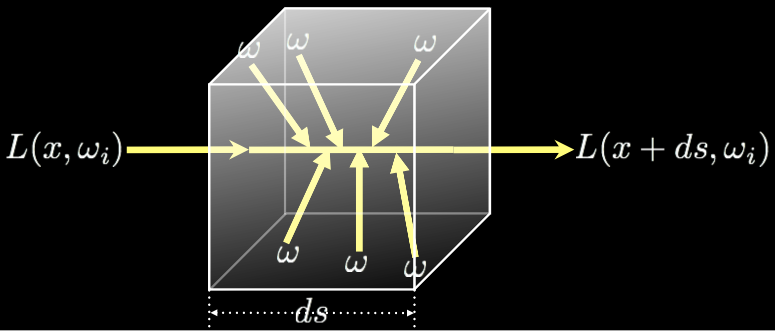
$$T(t) = \exp\left(-\int_0^t \sigma_t ds\right) = \exp\left(-t\sigma_t\right)$$

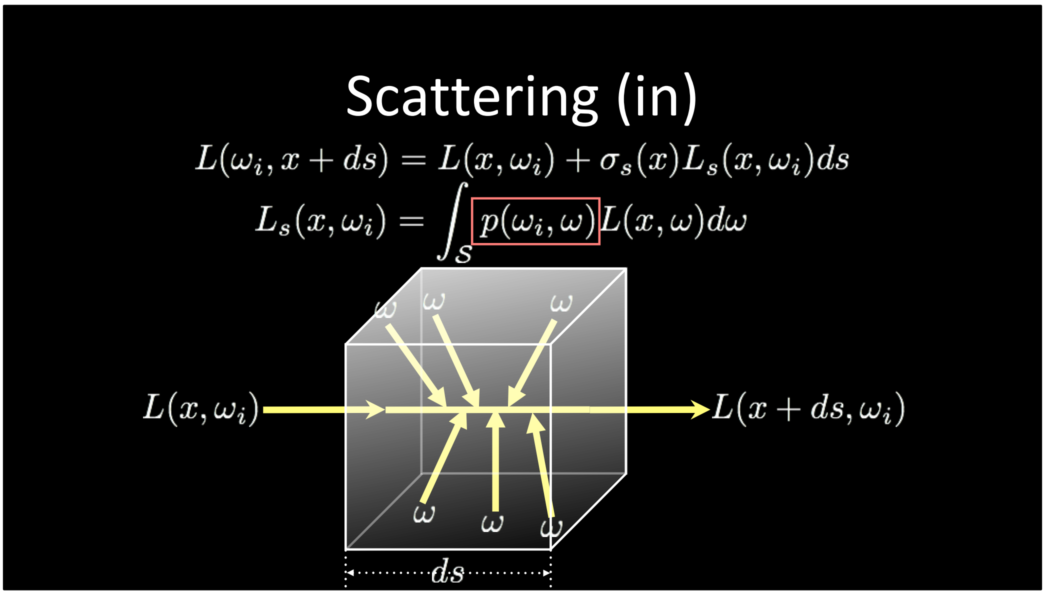




Scattering (in)

$$L(\omega_i, x + ds) = L(x, \omega_i) + \sigma_s(x) L_s(x, \omega_i) ds$$

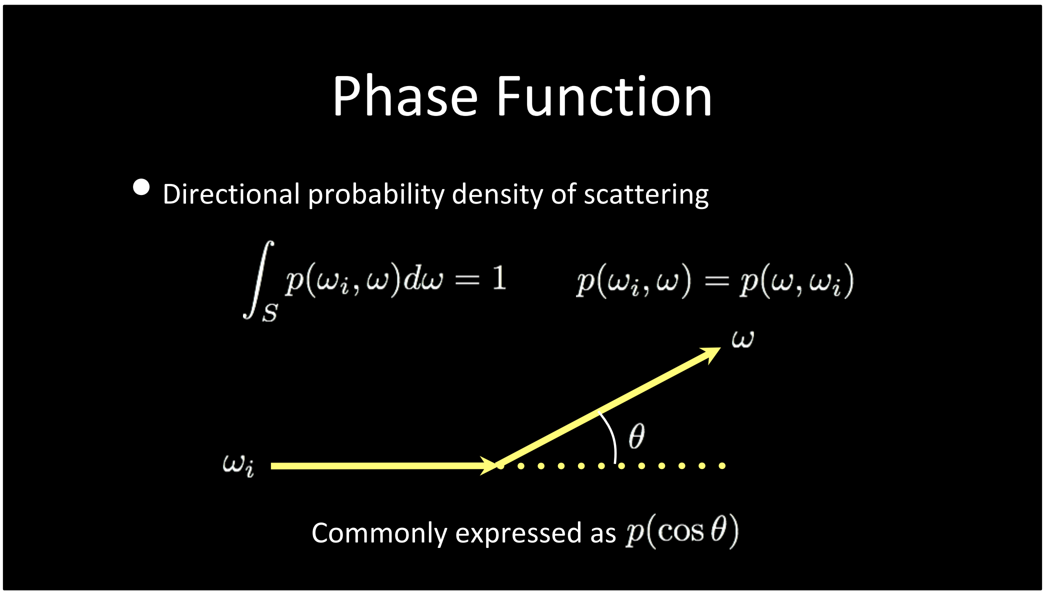




Scattering (in)

$$L(\omega_i, x + ds) = L(x, \omega_i) + \sigma_s(x) L_s(x, \omega_i) ds$$

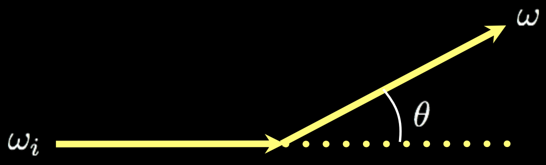
$$L_s(x, \omega_i) = \int_S \boxed{p(\omega_i, \omega)} L(x, \omega) d\omega$$



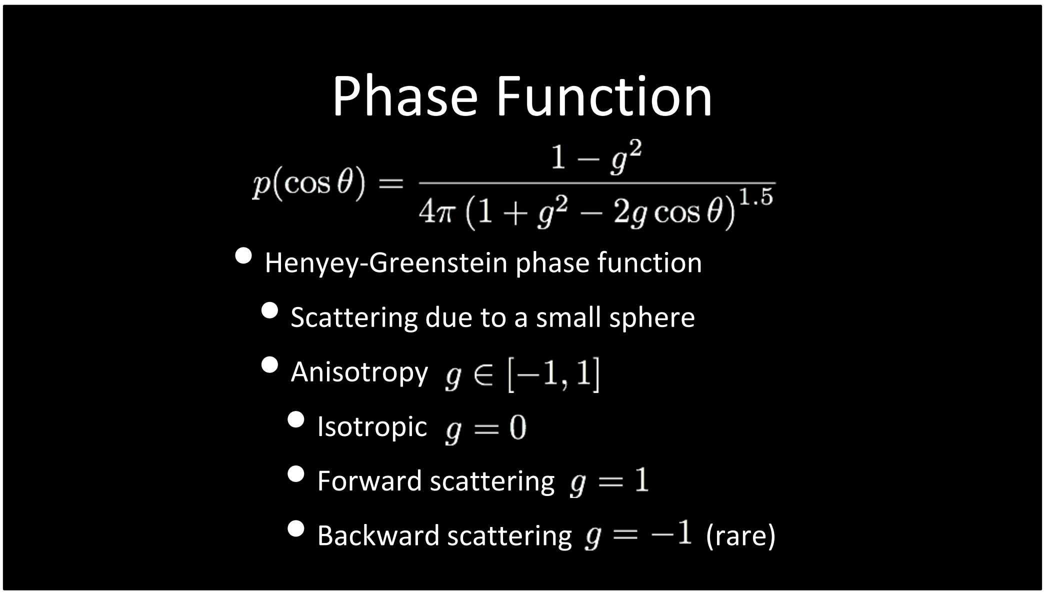
Phase Function

- Directional probability density of scattering

$$\int_S p(\omega_i, \omega) d\omega = 1 \qquad p(\omega_i, \omega) = p(\omega, \omega_i)$$



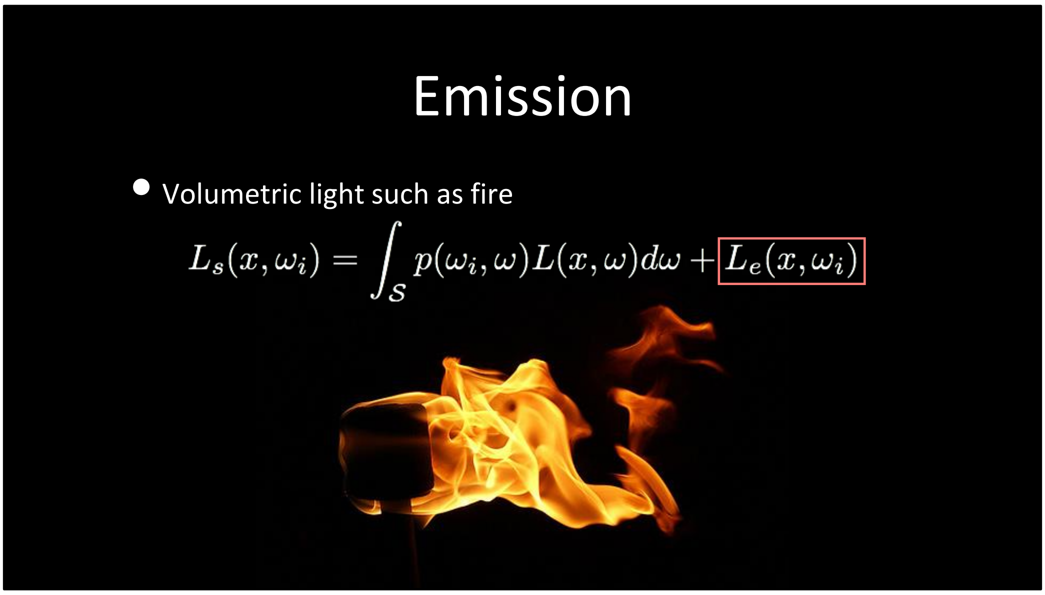
Commonly expressed as $p(\cos\theta)$



Phase Function

$$p(\cos\theta) = \frac{1-g^2}{4\pi\left(1+g^2-2g\cos\theta\right)^{1.5}}$$

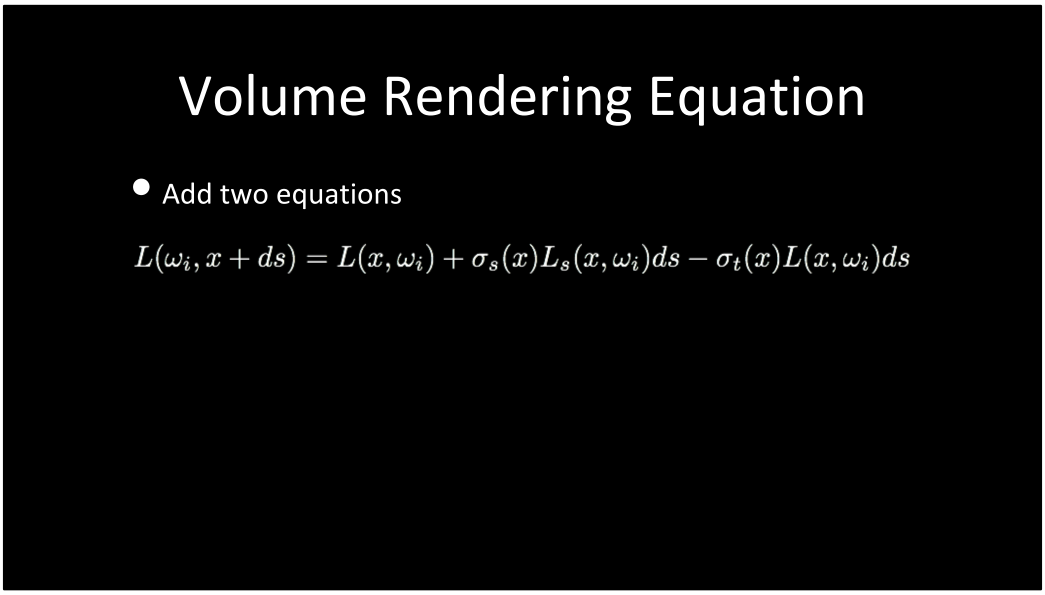
- Henyey-Greenstein phase function
  - Scattering due to a small sphere
  - Anisotropy $g \in [-1, 1]$
    - Isotropic $g = 0$
    - Forward scattering $g = 1$
    - Backward scattering $g = -1$ (rare)



Emission

- Volumetric light such as fire

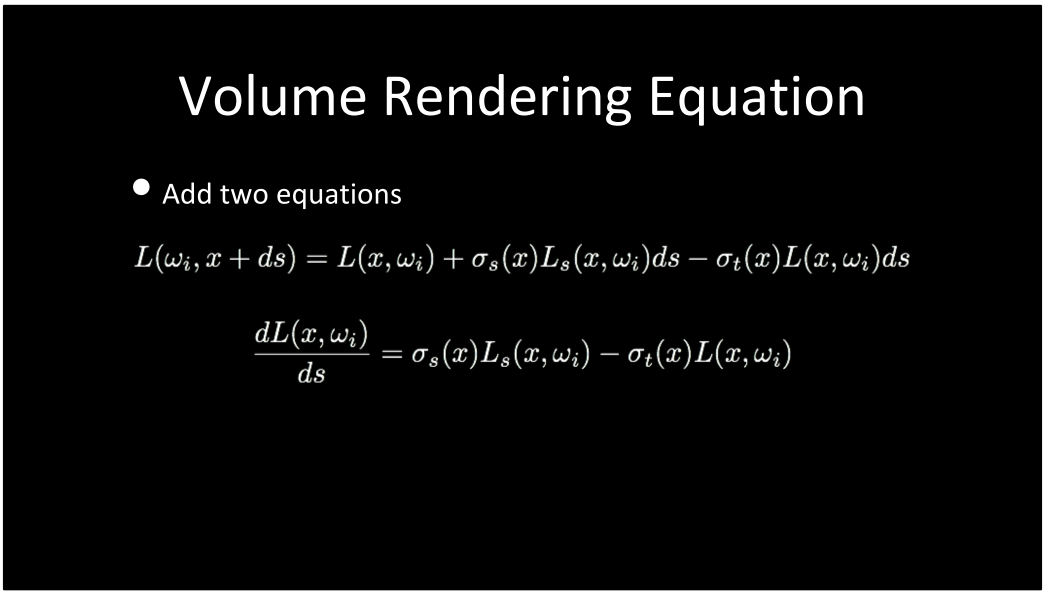
$$L_s(x,\omega_i) = \int_S p(\omega_i,\omega)L(x,\omega)d\omega + \boxed{L_e(x,\omega_i)}$$



Volume Rendering Equation

- Add two equations

$$L(\omega_i, x + ds) = L(x, \omega_i) + \sigma_s(x) L_s(x, \omega_i) ds - \sigma_t(x) L(x, \omega_i) ds$$

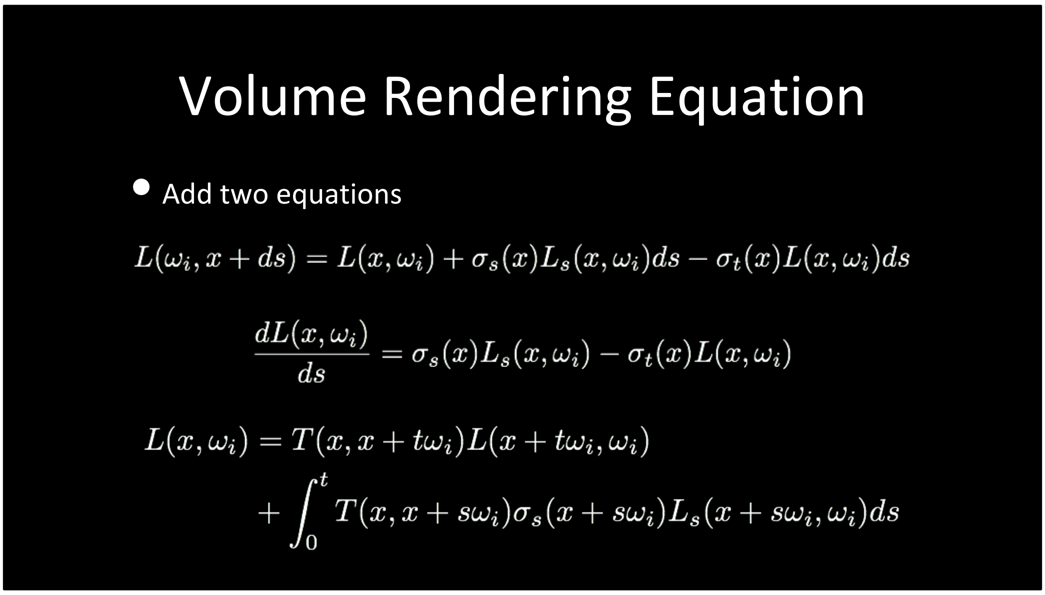


Volume Rendering Equation

- Add two equations

$$L(\omega_i, x + ds) = L(x, \omega_i) + \sigma_s(x) L_s(x, \omega_i) ds - \sigma_t(x) L(x, \omega_i) ds$$

$$\frac{dL(x, \omega_i)}{ds} = \sigma_s(x) L_s(x, \omega_i) - \sigma_t(x) L(x, \omega_i)$$



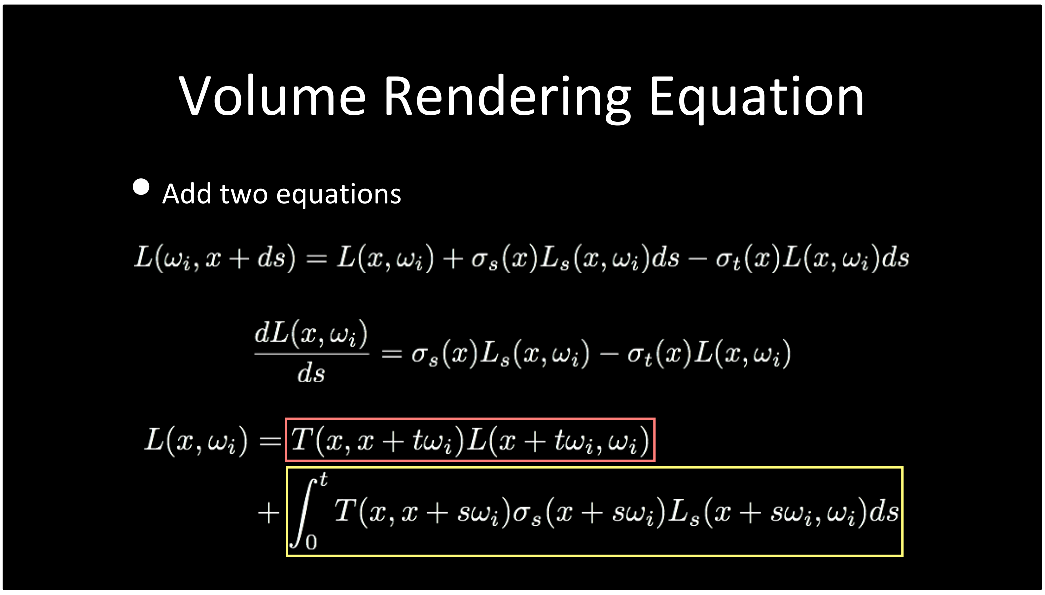
Volume Rendering Equation

- Add two equations

$$L(\omega_i, x + ds) = L(x, \omega_i) + \sigma_s(x) L_s(x, \omega_i) ds - \sigma_t(x) L(x, \omega_i) ds$$

$$\frac{dL(x, \omega_i)}{ds} = \sigma_s(x) L_s(x, \omega_i) - \sigma_t(x) L(x, \omega_i)$$

$$\begin{aligned} L(x, \omega_i) &= T(x, x + t\omega_i) L(x + t\omega_i, \omega_i) \\ &\quad + \int_0^t T(x, x + s\omega_i) \sigma_s(x + s\omega_i) L_s(x + s\omega_i, \omega_i) ds \end{aligned}$$



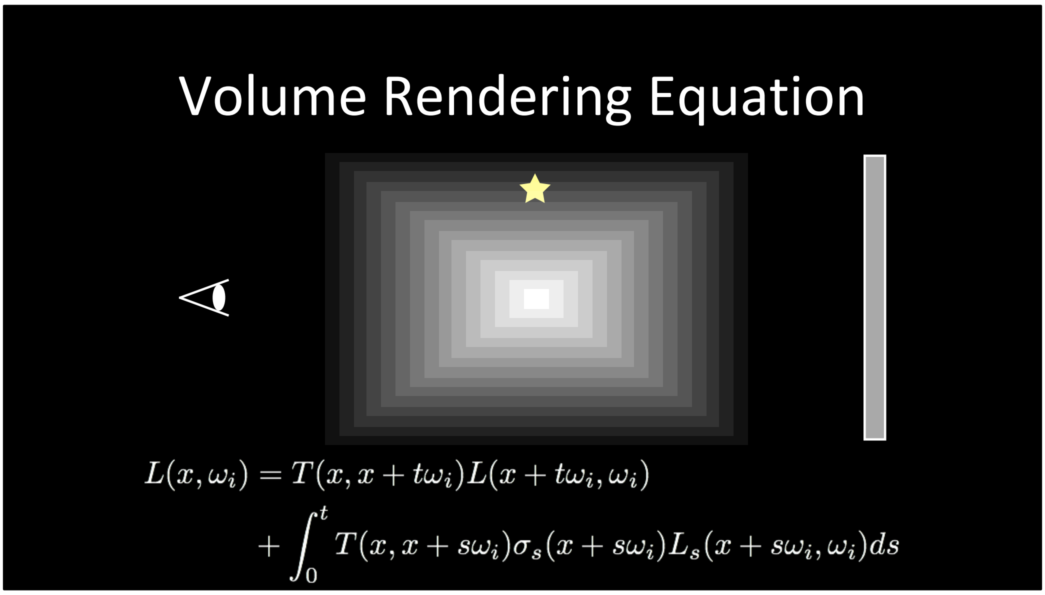
Volume Rendering Equation

- Add two equations

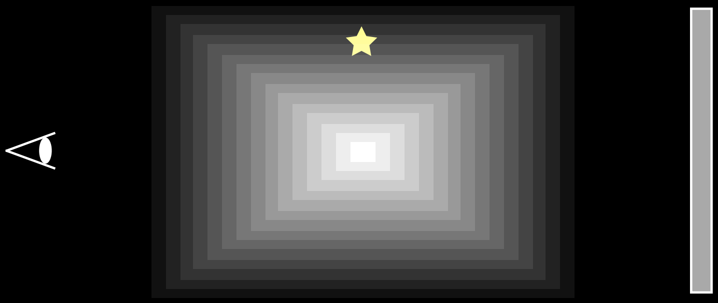
$$L(\omega_i, x + ds) = L(x, \omega_i) + \sigma_s(x) L_s(x, \omega_i) ds - \sigma_t(x) L(x, \omega_i) ds$$

$$\frac{dL(x, \omega_i)}{ds} = \sigma_s(x) L_s(x, \omega_i) - \sigma_t(x) L(x, \omega_i)$$

$$\begin{aligned} L(x, \omega_i) = & \boxed{T(x, x + t\omega_i) L(x + t\omega_i, \omega_i)} \\ & + \boxed{\int_0^t T(x, x + s\omega_i) \sigma_s(x + s\omega_i) L_s(x + s\omega_i, \omega_i) ds} \end{aligned}$$

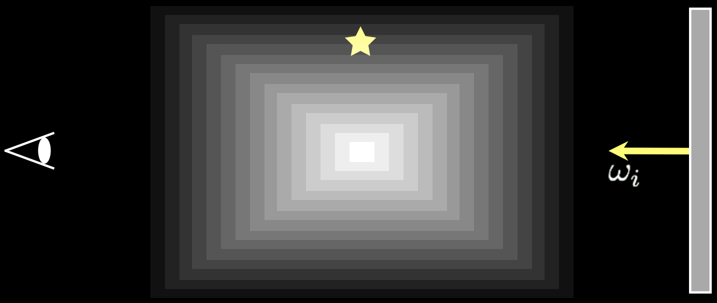


Volume Rendering Equation

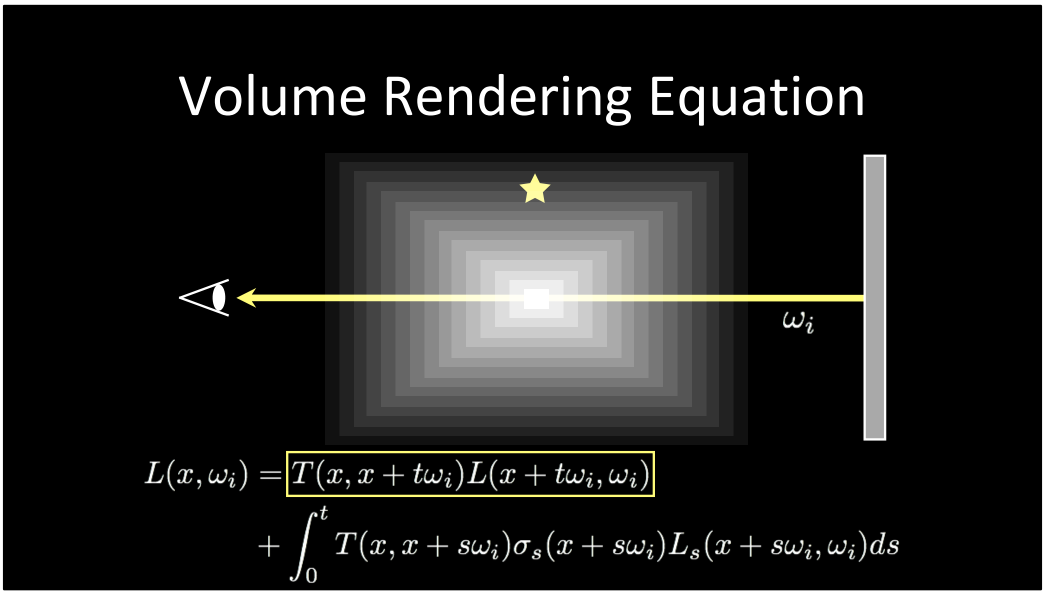


$$L(x,\omega_i) = T(x, x+t\omega_i)L(x+t\omega_i,\omega_i) + \int_0^t T(x,x+s\omega_i)\sigma_s(x+s\omega_i)L_s(x+s\omega_i,\omega_i)ds$$

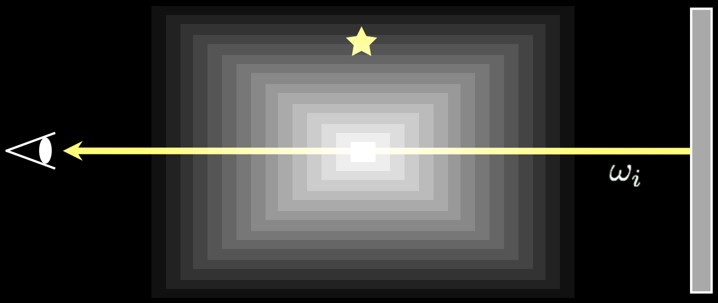
Volume Rendering Equation



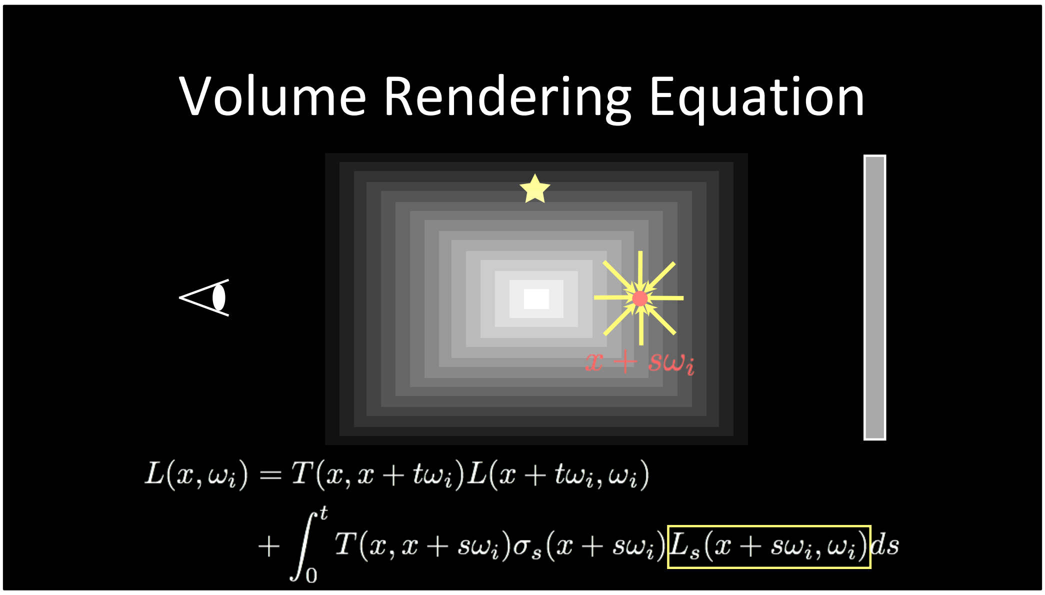
$$L(x,\omega_i) = T(x, x+t\omega_i)\boxed{L(x+t\omega_i,\omega_i)}$$
$$+ \int_0^t T(x, x+s\omega_i)\sigma_s(x+s\omega_i)L_s(x+s\omega_i,\omega_i)ds$$



Volume Rendering Equation

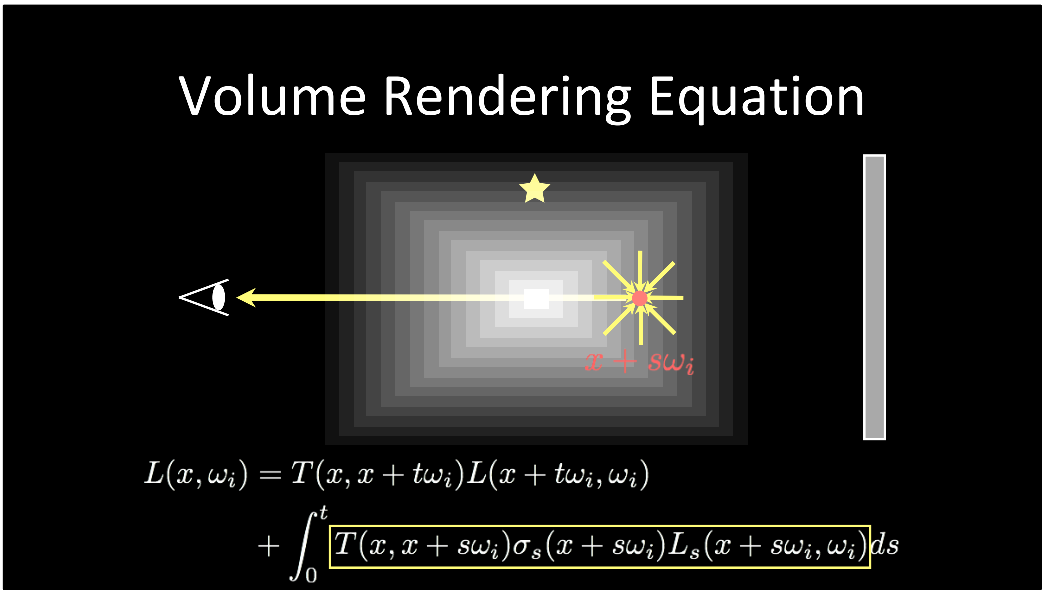


$$L(x,\omega_i) = \boxed{T(x, x+t\omega_i)L(x+t\omega_i,\omega_i)} + \int_0^t T(x, x+s\omega_i)\sigma_s(x+s\omega_i)L_s(x+s\omega_i,\omega_i)ds$$

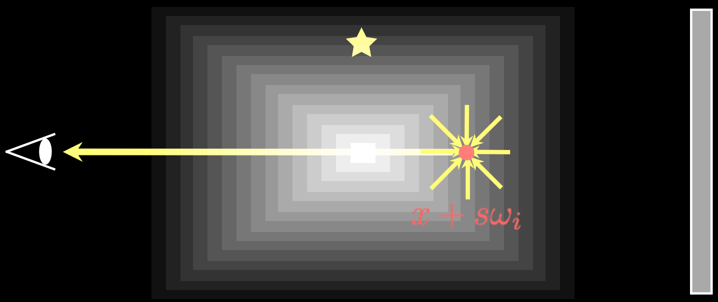


Volume Rendering Equation

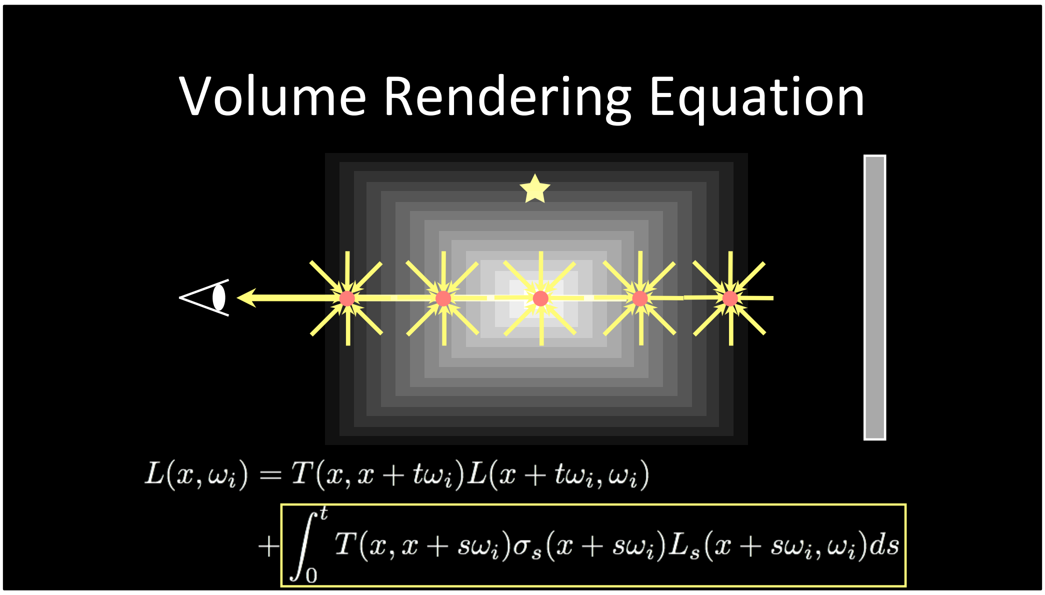
$$L(x,\omega_i) = T(x, x+t\omega_i)L(x+t\omega_i,\omega_i) + \int_0^t T(x,x+s\omega_i)\sigma_s(x+s\omega_i)\boxed{L_s(x+s\omega_i,\omega_i)}ds$$



Volume Rendering Equation

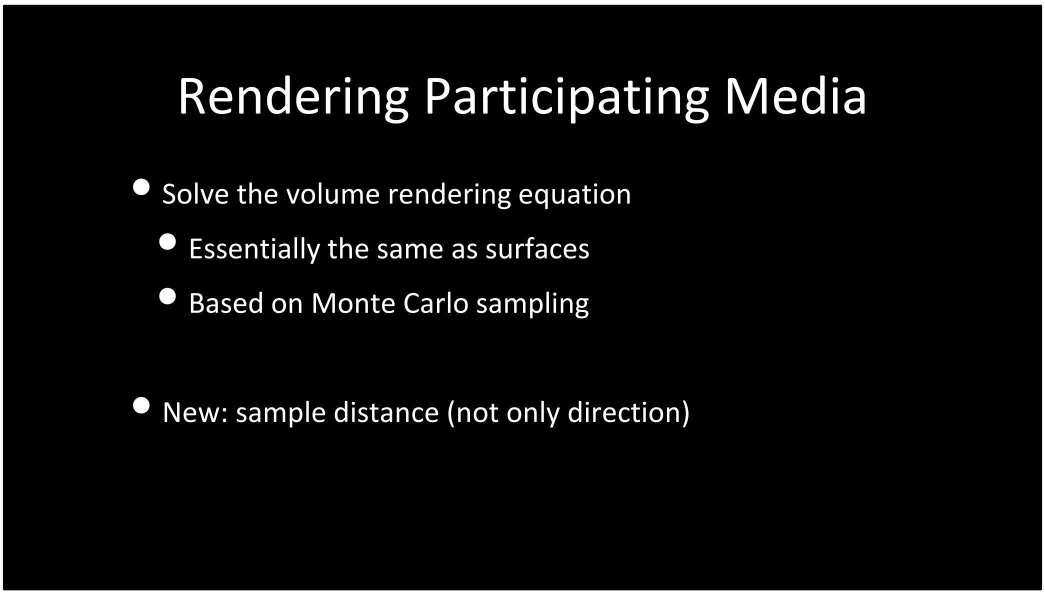


$$L(x,\omega_i) = T(x, x + t\omega_i)L(x + t\omega_i, \omega_i) + \int_0^t \boxed{T(x, x + s\omega_i)\sigma_s(x + s\omega_i)L_s(x + s\omega_i, \omega_i)}ds$$



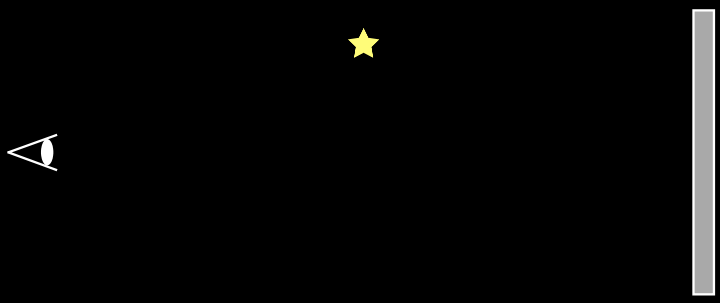
Volume Rendering Equation

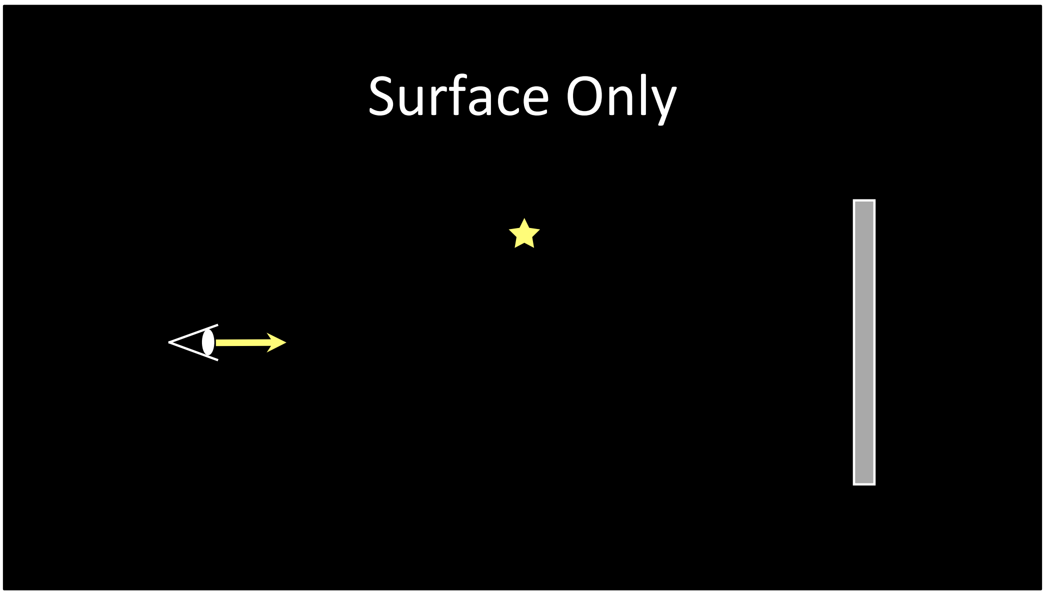
$$L(x,\omega_i) = T(x, x+t\omega_i)L(x+t\omega_i,\omega_i)$$
$$+\boxed{\int_0^t T(x,x+s\omega_i)\sigma_s(x+s\omega_i)L_s(x+s\omega_i,\omega_i)ds}$$


Rendering Participating Media

- Solve the volume rendering equation
  - Essentially the same as surfaces
  - Based on Monte Carlo sampling
- New: sample distance (not only direction)

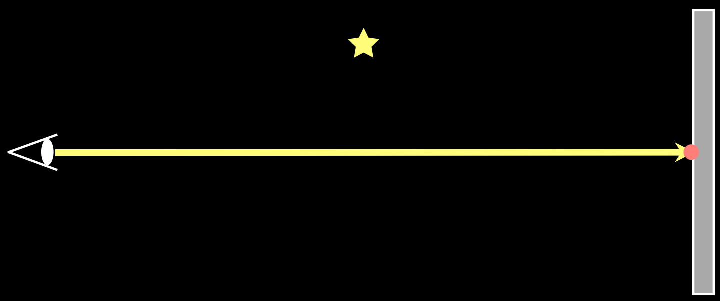
Surface Only



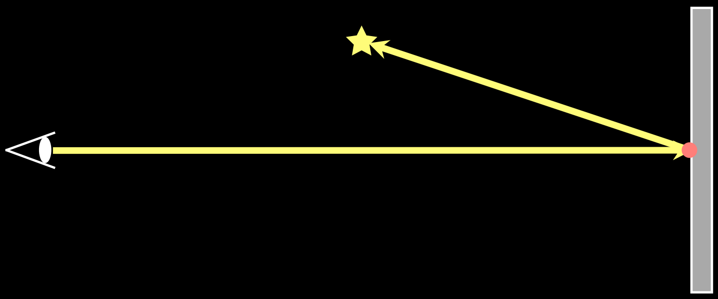


Surface Only

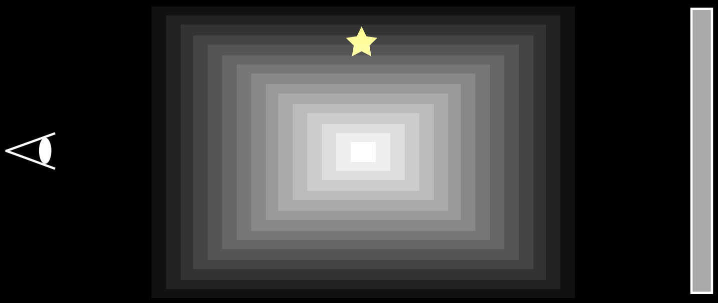
Surface Only



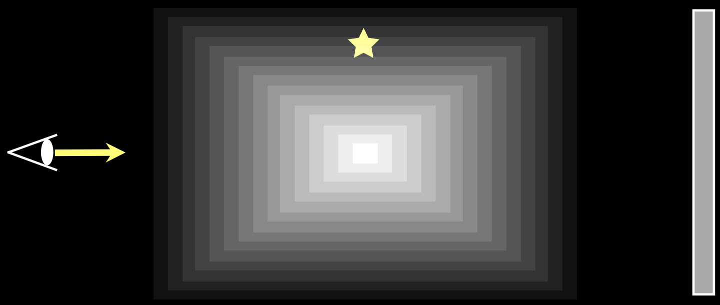
Surface Only

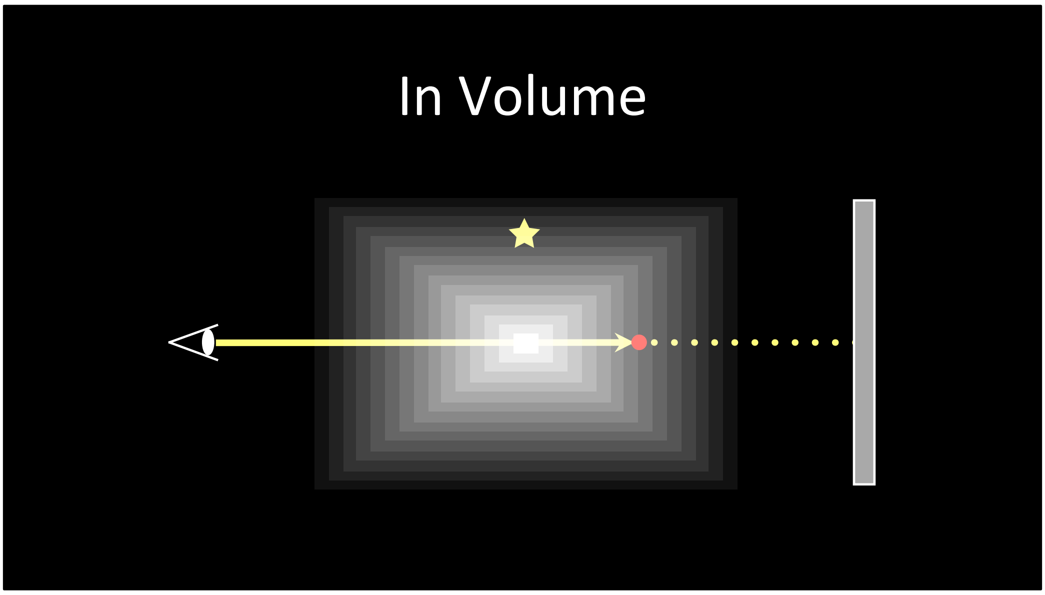


In Volume

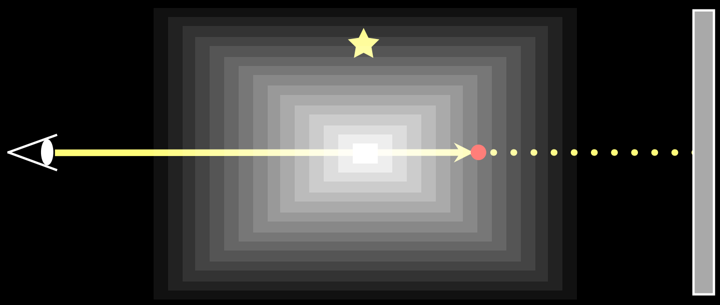


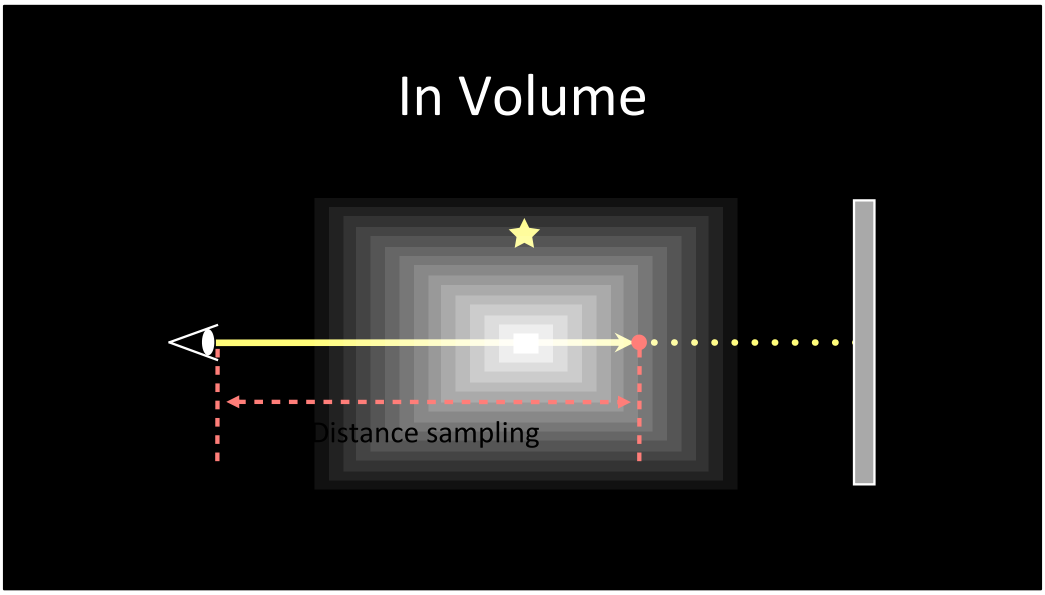
In Volume



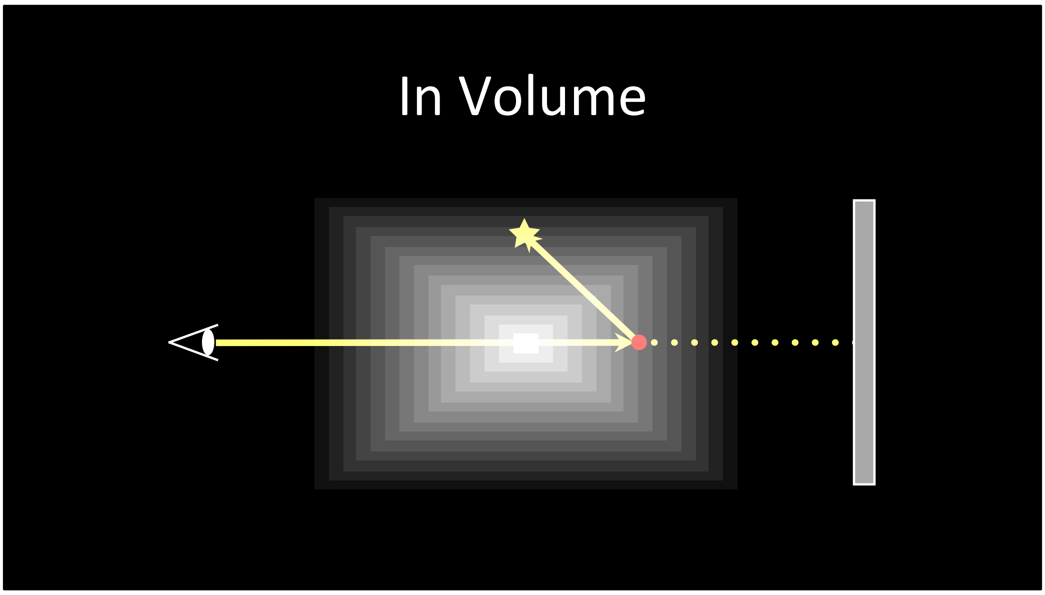


In Volume

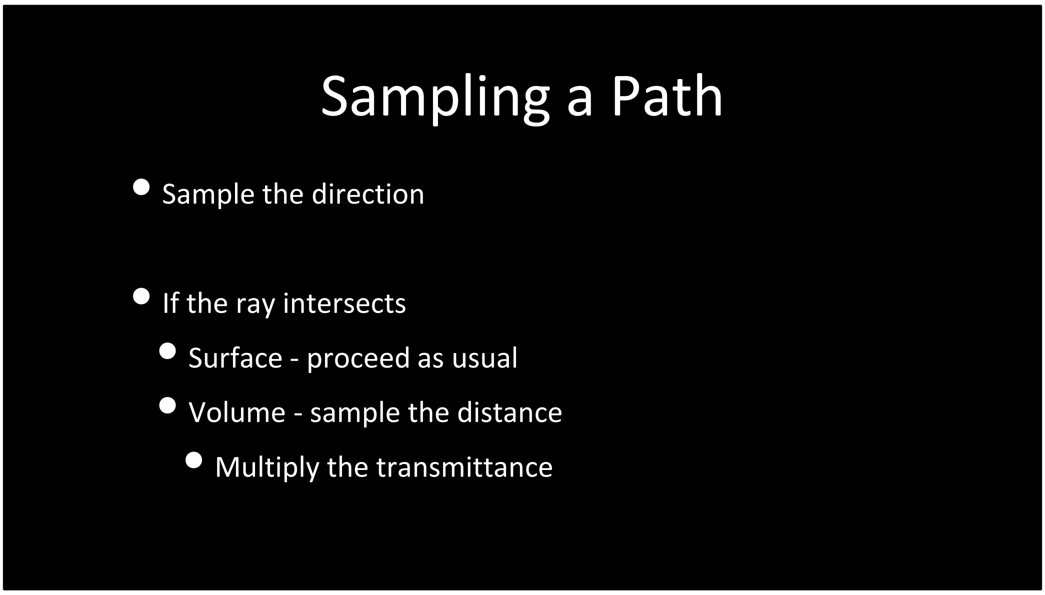




In Volume

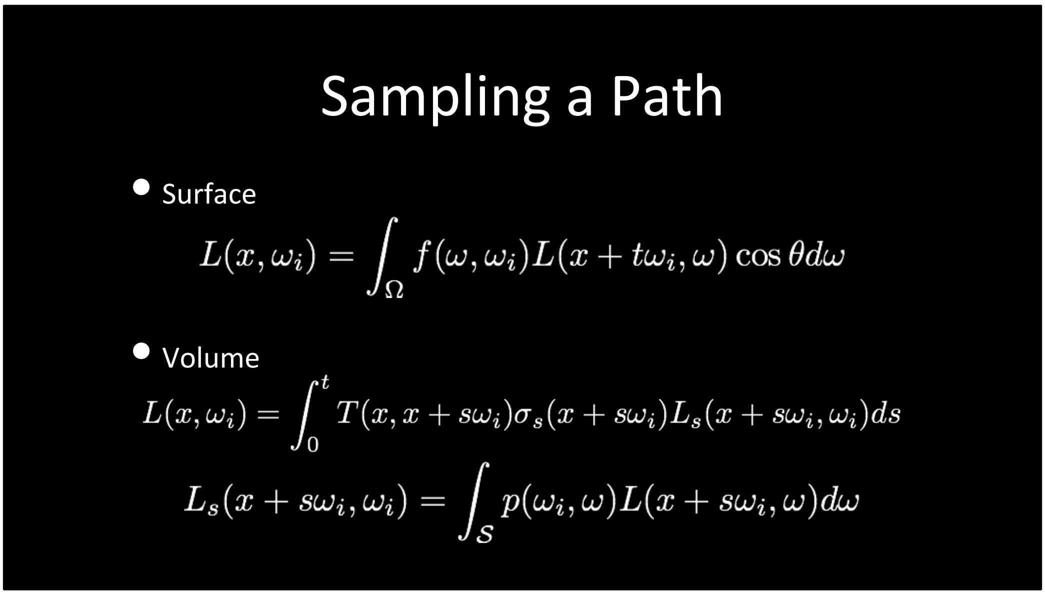


In Volume



Sampling a Path

- Sample the direction
- If the ray intersects
  - Surface - proceed as usual
  - Volume - sample the distance
    - Multiply the transmittance



Sampling a Path

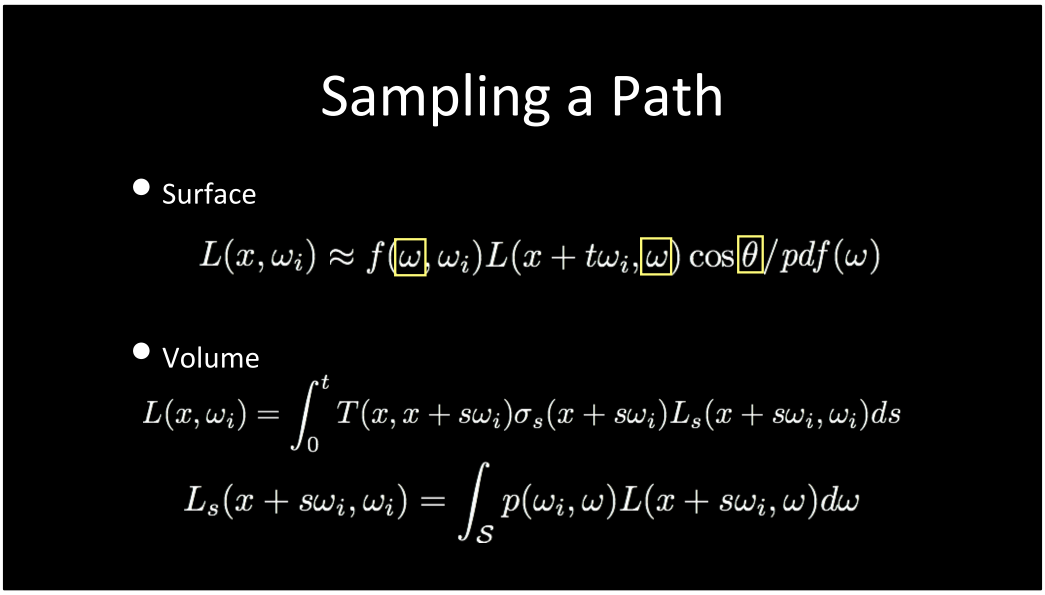
- Surface

$$L(x,\omega_i) = \int_\Omega f(\omega,\omega_i) L(x + t\omega_i, \omega) \cos\theta d\omega$$

- Volume

$$L(x,\omega_i) = \int_0^t T(x, x + s\omega_i)\sigma_s(x + s\omega_i) L_s(x + s\omega_i, \omega_i) ds$$

$$L_s(x + s\omega_i, \omega_i) = \int_S p(\omega_i, \omega) L(x + s\omega_i, \omega) d\omega$$



Sampling a Path

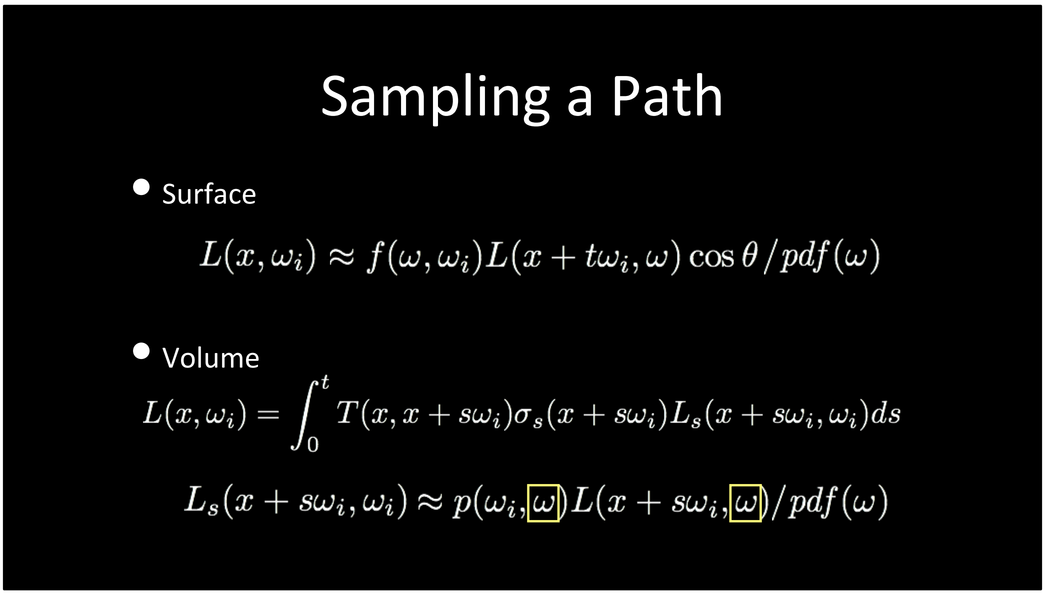
- Surface

$$L(x,\omega_i) \approx f(\boxed{\omega},\omega_i)L(x+t\omega_i,\boxed{\omega})\cos\boxed{\theta}/pdf(\omega)$$

- Volume

$$L(x,\omega_i) = \int_0^t T(x,x+s\omega_i)\sigma_s(x+s\omega_i)L_s(x+s\omega_i,\omega_i)ds$$

$$L_s(x+s\omega_i,\omega_i) = \int_S p(\omega_i,\omega)L(x+s\omega_i,\omega)d\omega$$



Sampling a Path

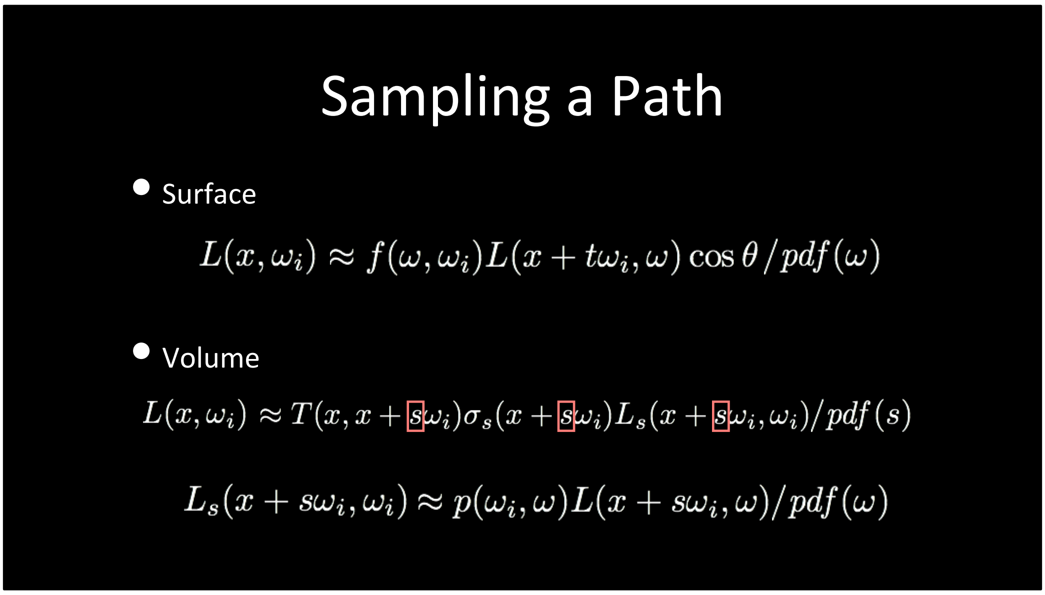
- Surface

$$L(x,\omega_i) \approx f(\omega,\omega_i)L(x+t\omega_i,\omega)\cos\theta\,/pdf(\omega)$$

- Volume

$$L(x,\omega_i) = \int_0^t T(x,x+s\omega_i)\sigma_s(x+s\omega_i)L_s(x+s\omega_i,\omega_i)ds$$

$$L_s(x+s\omega_i,\omega_i) \approx p(\omega_i,\boxed{\omega})L(x+s\omega_i,\boxed{\omega})/pdf(\omega)$$



Sampling a Path

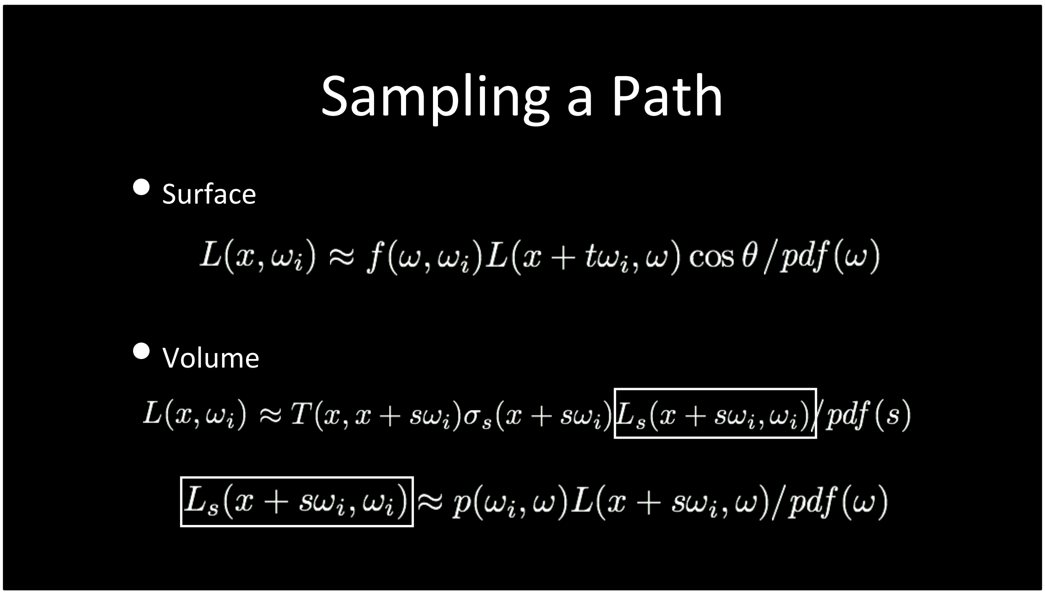
- Surface

$$L(x,\omega_i) \approx f(\omega,\omega_i)L(x+t\omega_i,\omega)\cos\theta\,/\,pdf(\omega)$$

- Volume

$$L(x,\omega_i) \approx T(x,x+\boxed{s}\omega_i)\sigma_s(x+\boxed{s}\omega_i)L_s(x+\boxed{s}\omega_i,\omega_i)/pdf(s)$$

$$L_s(x+s\omega_i,\omega_i) \approx p(\omega_i,\omega)L(x+s\omega_i,\omega)/pdf(\omega)$$



Sampling a Path

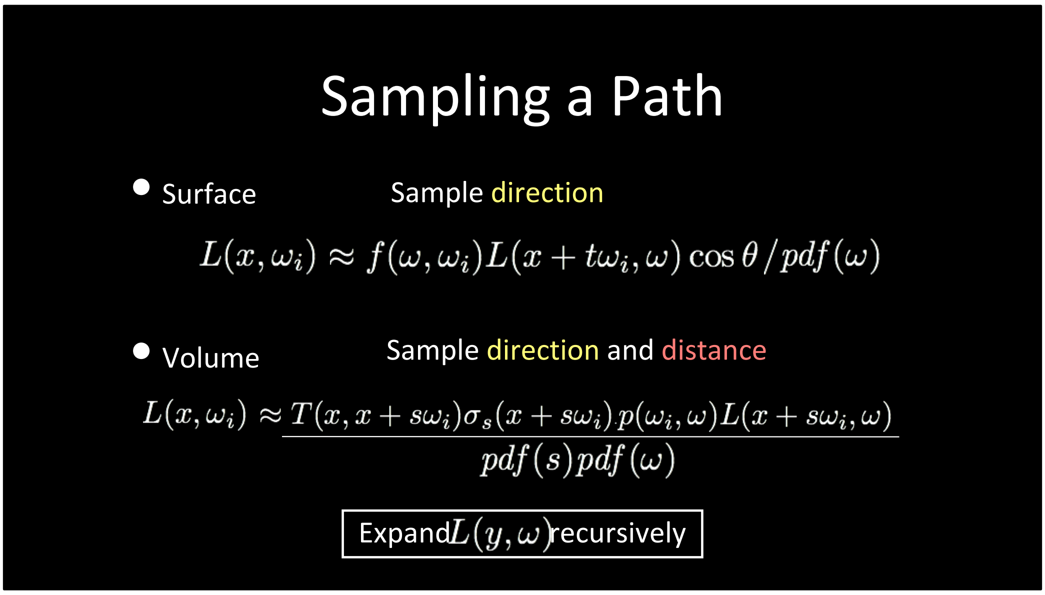
- Surface

$$L(x,\omega_i) \approx f(\omega,\omega_i)L(x+t\omega_i,\omega)\cos\theta / pdf(\omega)$$

- Volume

$$L(x,\omega_i) \approx T(x,x+s\omega_i)\sigma_s(x+s\omega_i)\boxed{L_s(x+s\omega_i,\omega_i)}/pdf(s)$$

$$\boxed{L_s(x+s\omega_i,\omega_i)} \approx p(\omega_i,\omega)L(x+s\omega_i,\omega)/pdf(\omega)$$



Sampling a Path

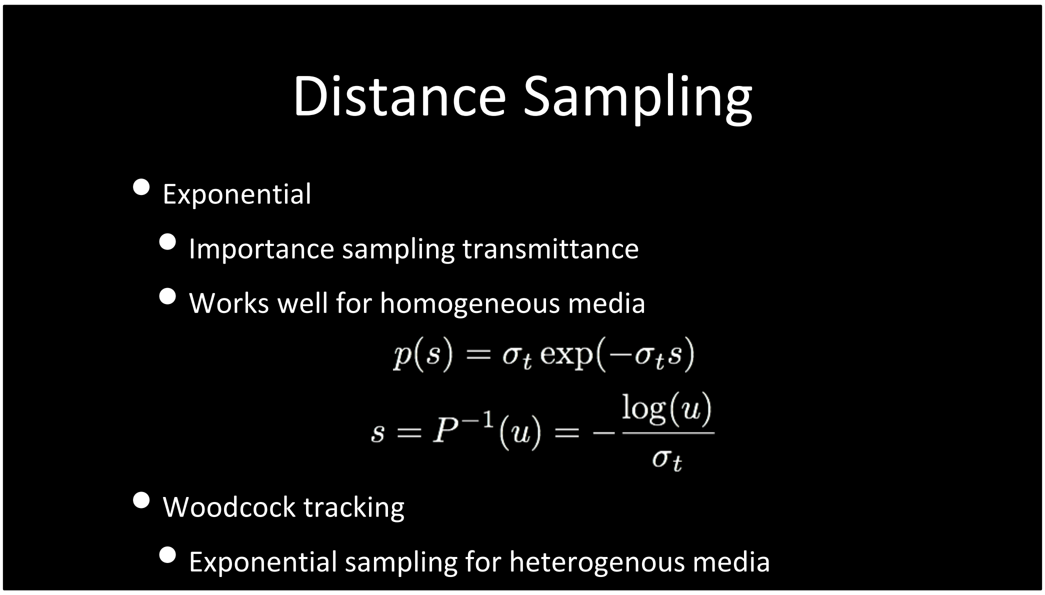
- Surface — Sample direction

$$L(x,\omega_i) \approx f(\omega,\omega_i)L(x+t\omega_i,\omega)\cos\theta\,/\,pdf(\omega)$$

- Volume — Sample direction and distance

$$L(x,\omega_i) \approx \frac{T(x,x+s\omega_i)\sigma_s(x+s\omega_i)\,p(\omega_i,\omega)L(x+s\omega_i,\omega)}{pdf(s)\,pdf(\omega)}$$

$\boxed{\text{Expand}\,L(y,\omega)\,\text{recursively}}$



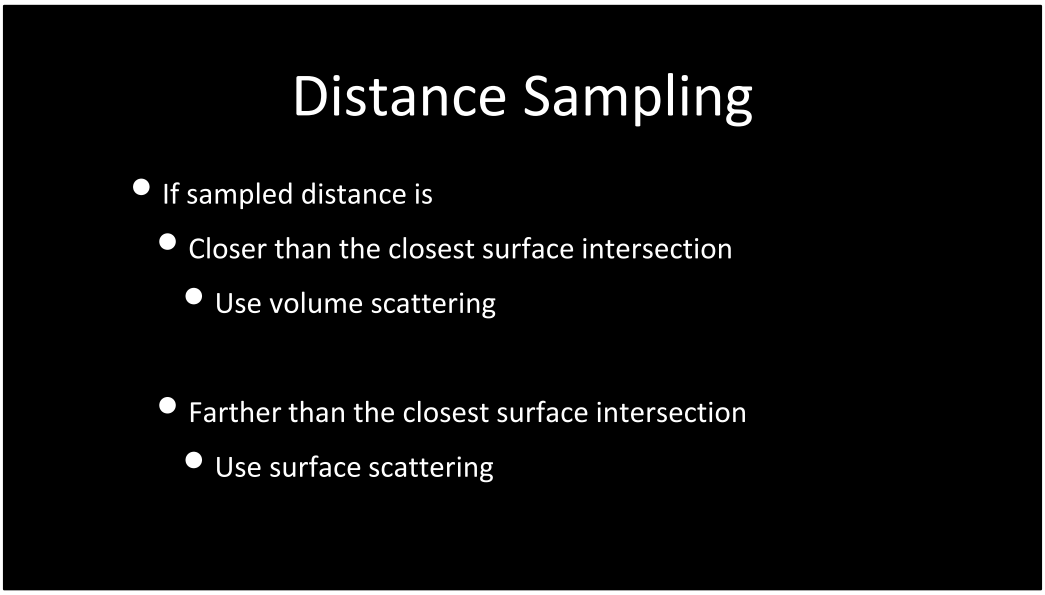
Distance Sampling

- Exponential
  - Importance sampling transmittance
  - Works well for homogeneous media

$$p(s) = \sigma_t \exp(-\sigma_t s)$$

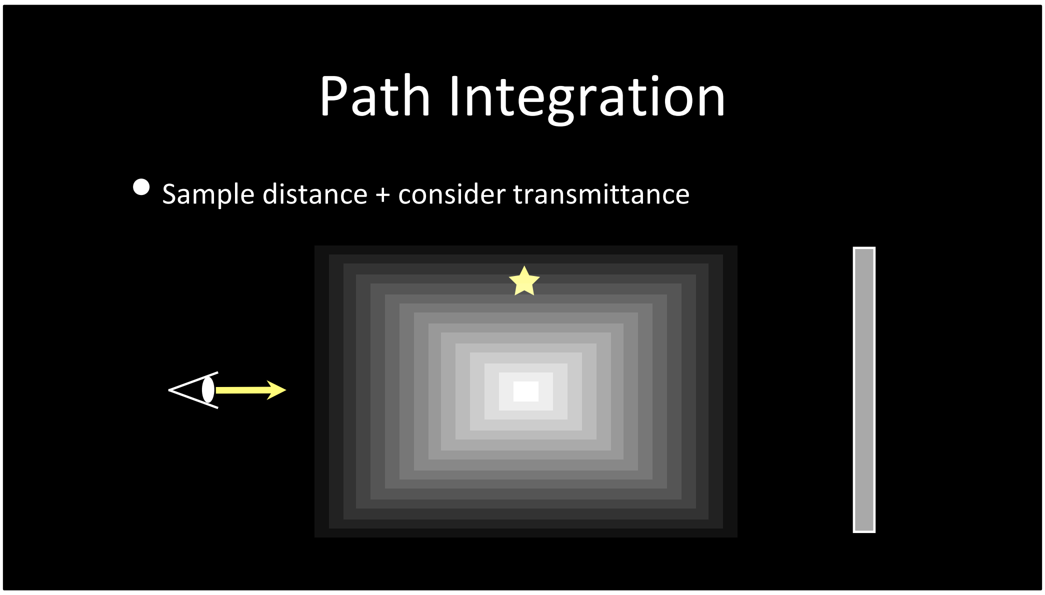
$$s = P^{-1}(u) = -\frac{\log(u)}{\sigma_t}$$

- Woodcock tracking
  - Exponential sampling for heterogenous media



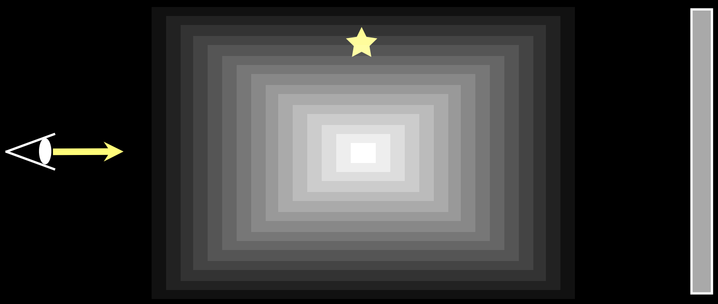
Distance Sampling

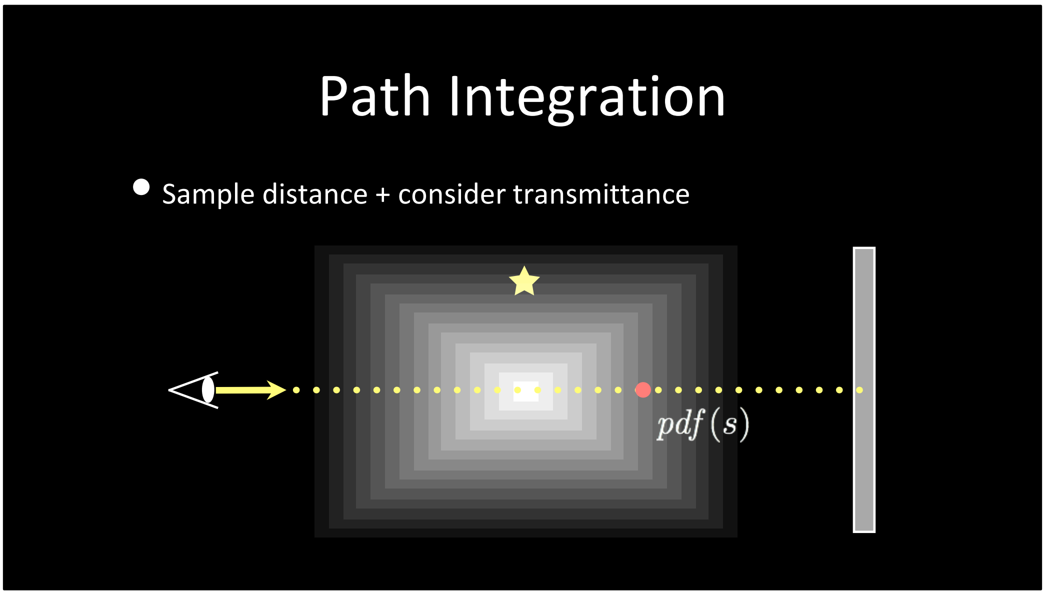
- If sampled distance is
  - Closer than the closest surface intersection
    - Use volume scattering
  - Farther than the closest surface intersection
    - Use surface scattering



Path Integration

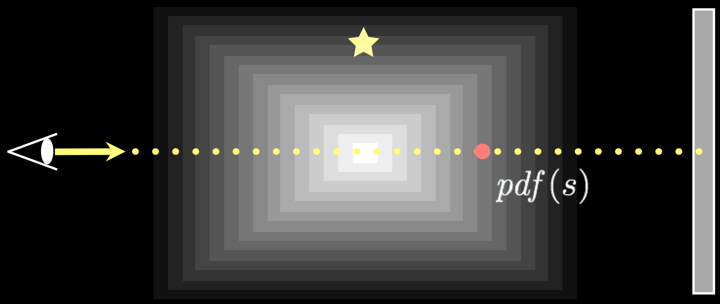
- Sample distance + consider transmittance

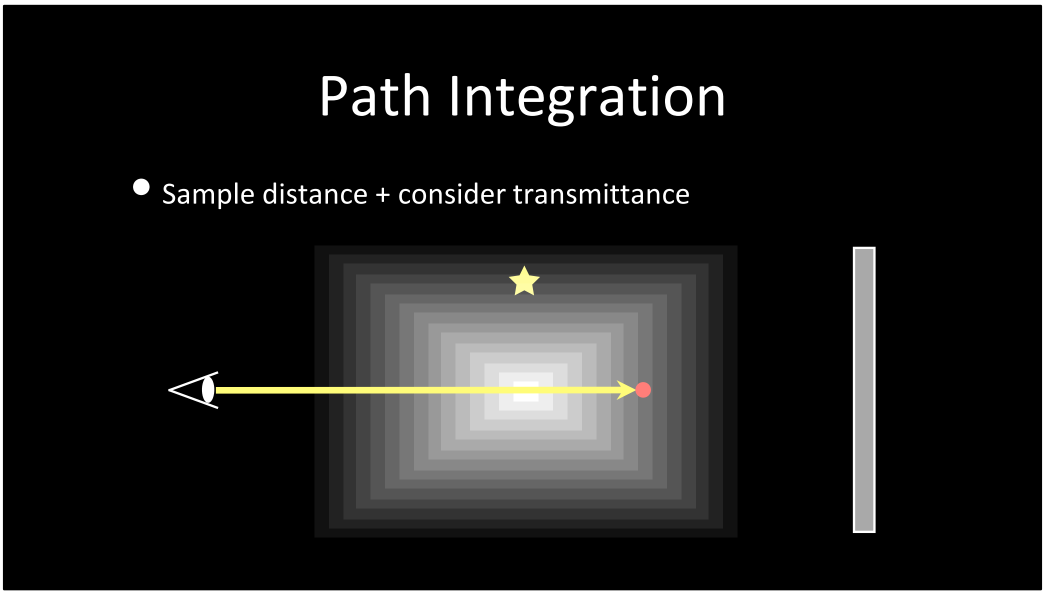




Path Integration

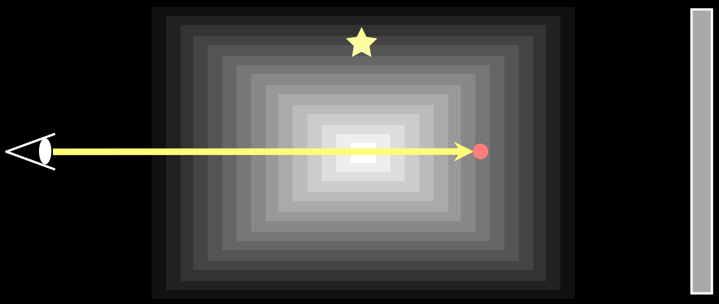
- Sample distance + consider transmittance

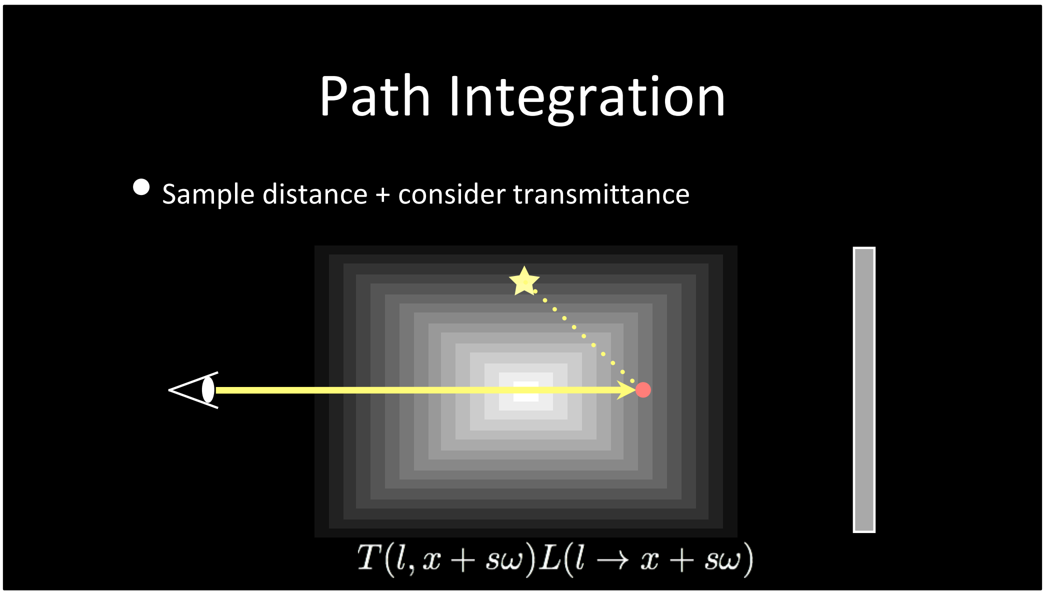




Path Integration

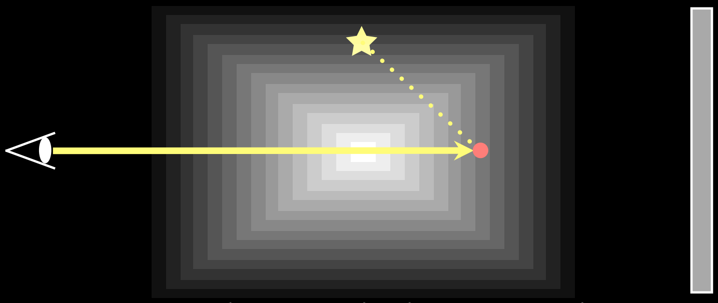
- Sample distance + consider transmittance



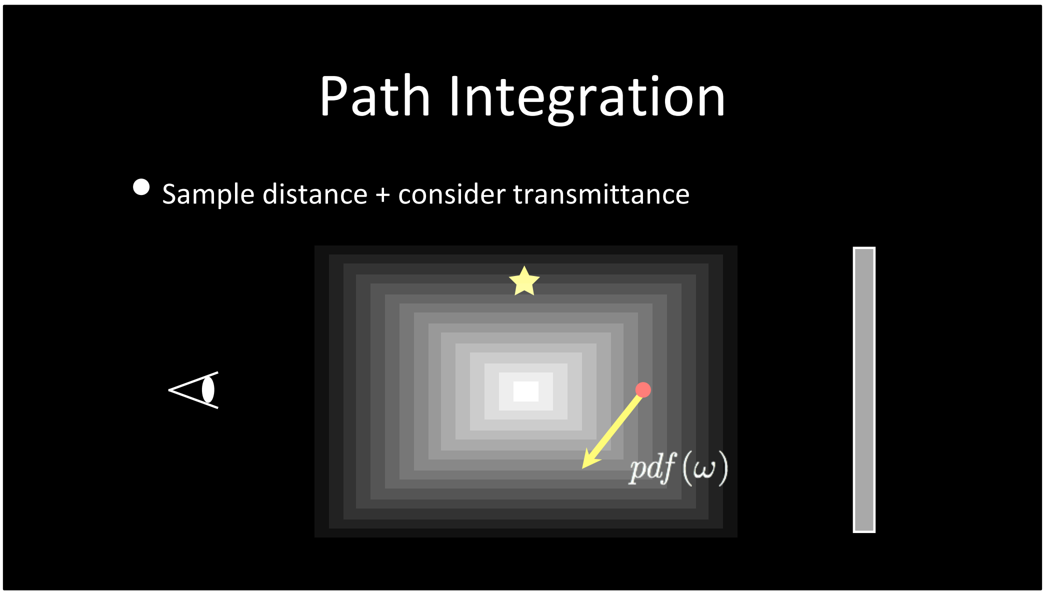


Path Integration

- Sample distance + consider transmittance

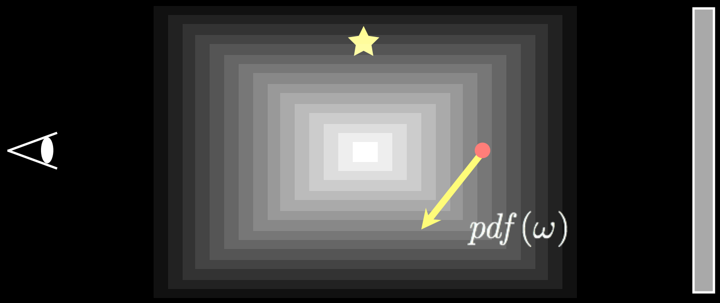


$$T(l, x + s\omega)L(l \to x + s\omega)$$



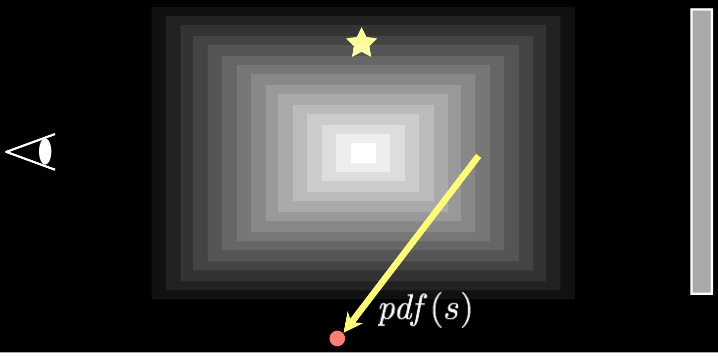
Path Integration

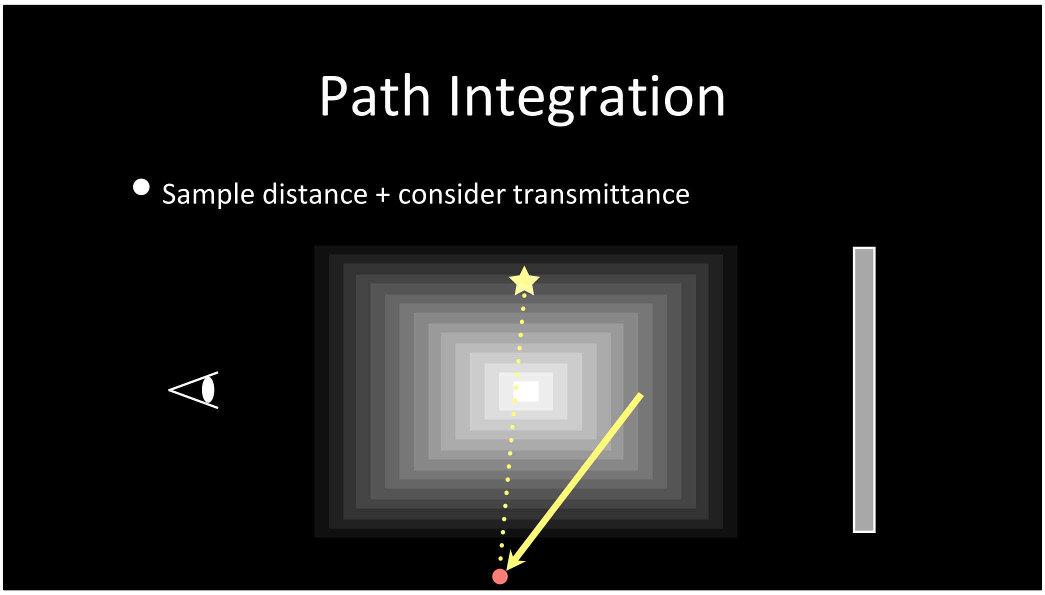
- Sample distance + consider transmittance



Path Integration

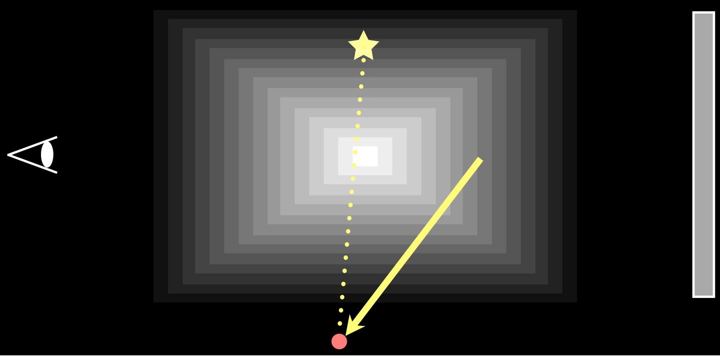
- Sample distance + consider transmittance



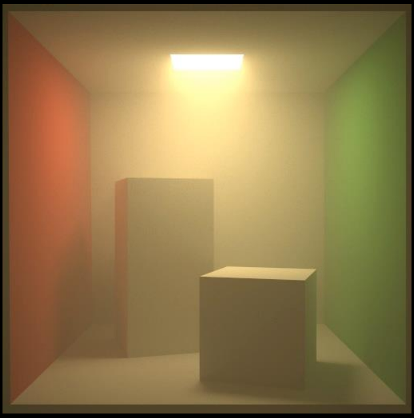


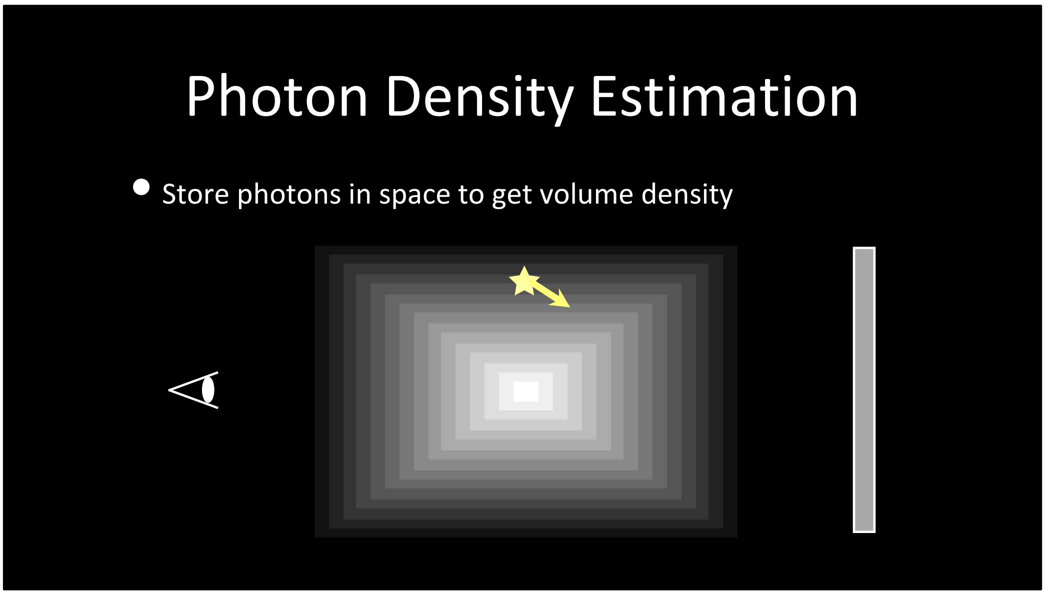
Path Integration

- Sample distance + consider transmittance



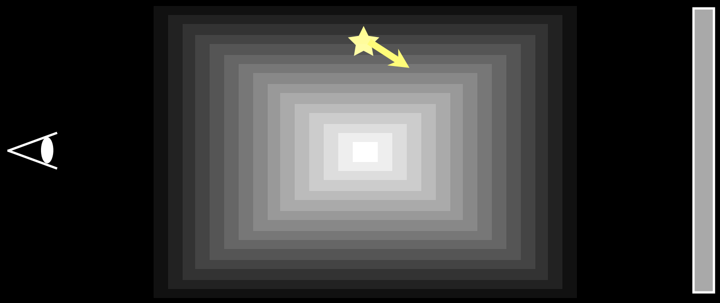
Path Integration

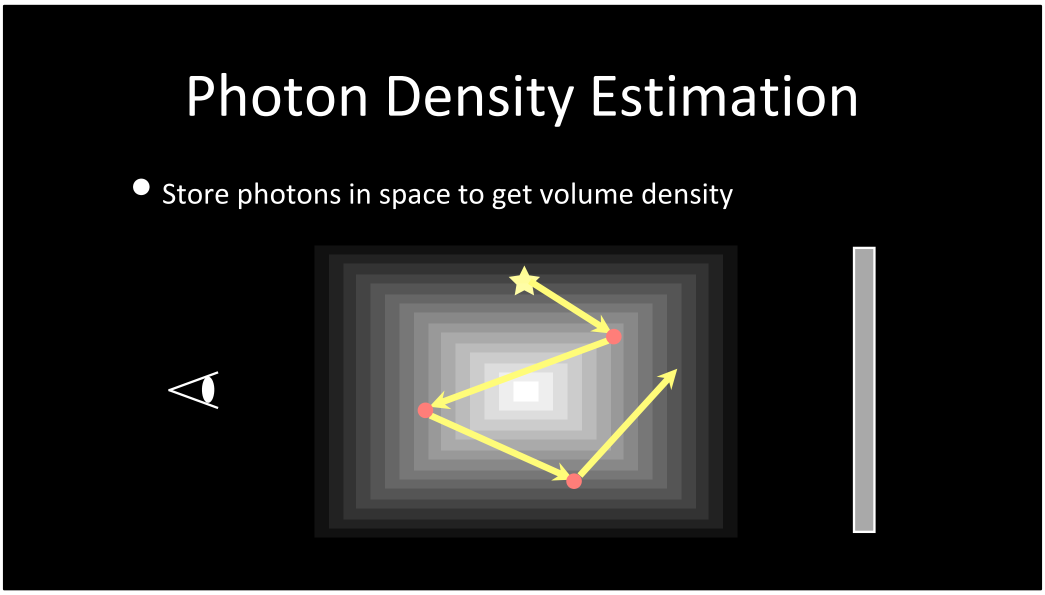




Photon Density Estimation

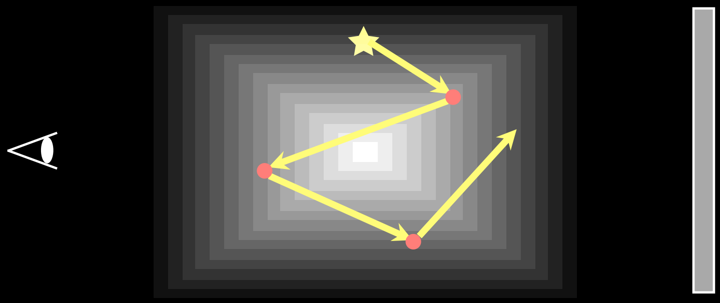
- Store photons in space to get volume density

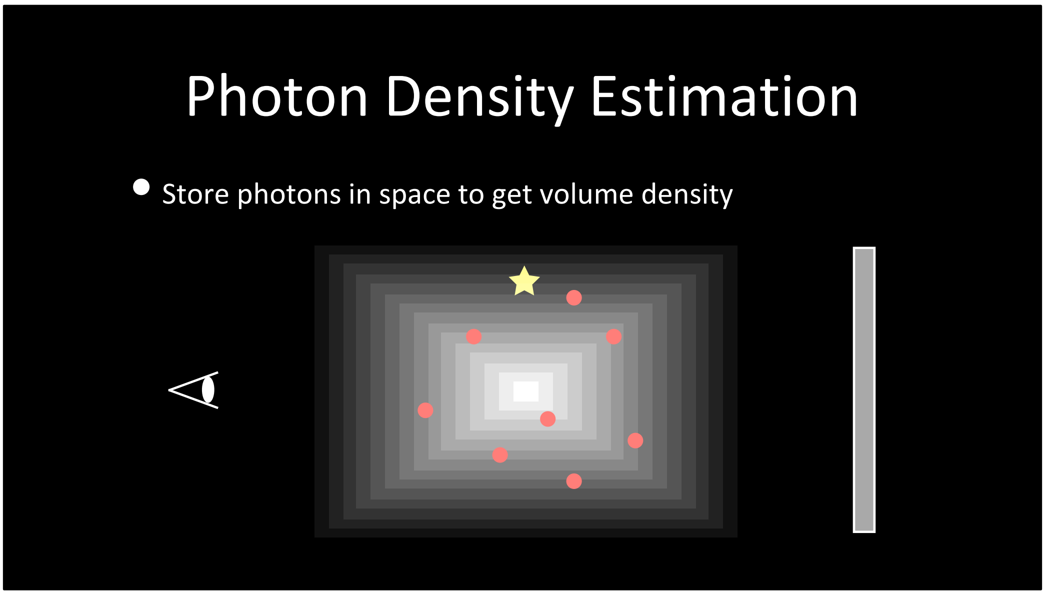




Photon Density Estimation

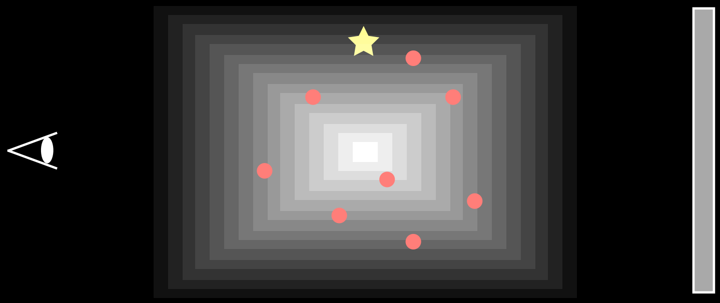
- Store photons in space to get volume density

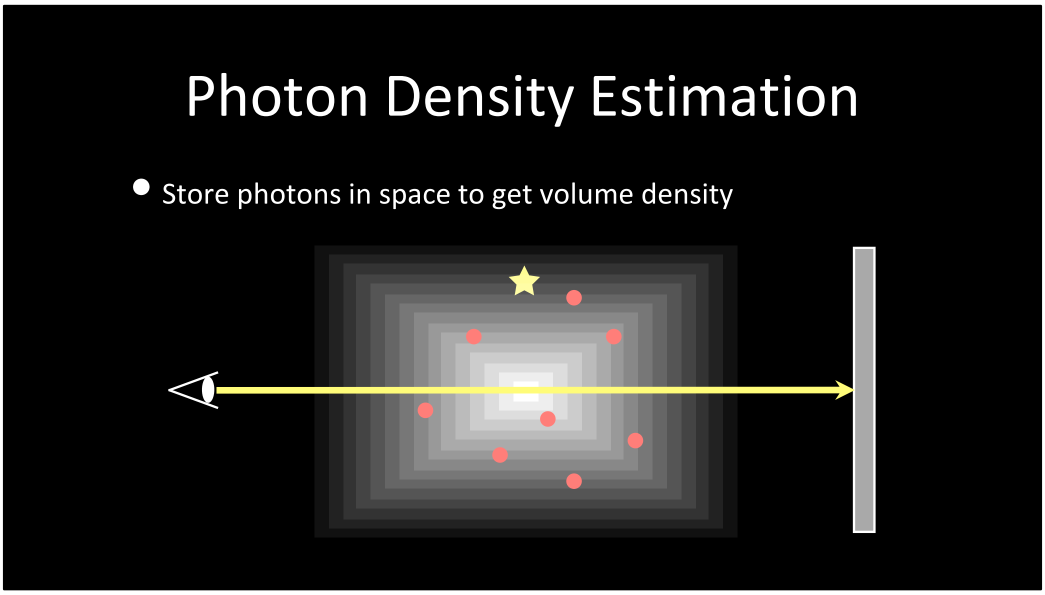




Photon Density Estimation

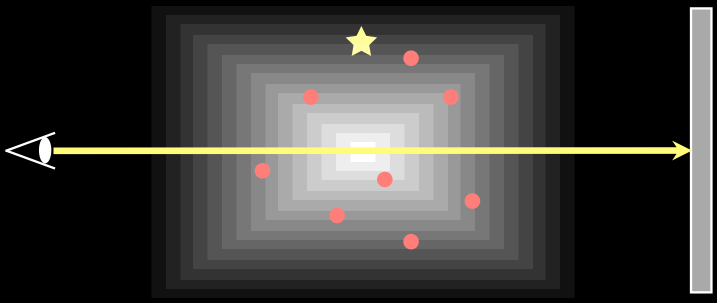
- Store photons in space to get volume density

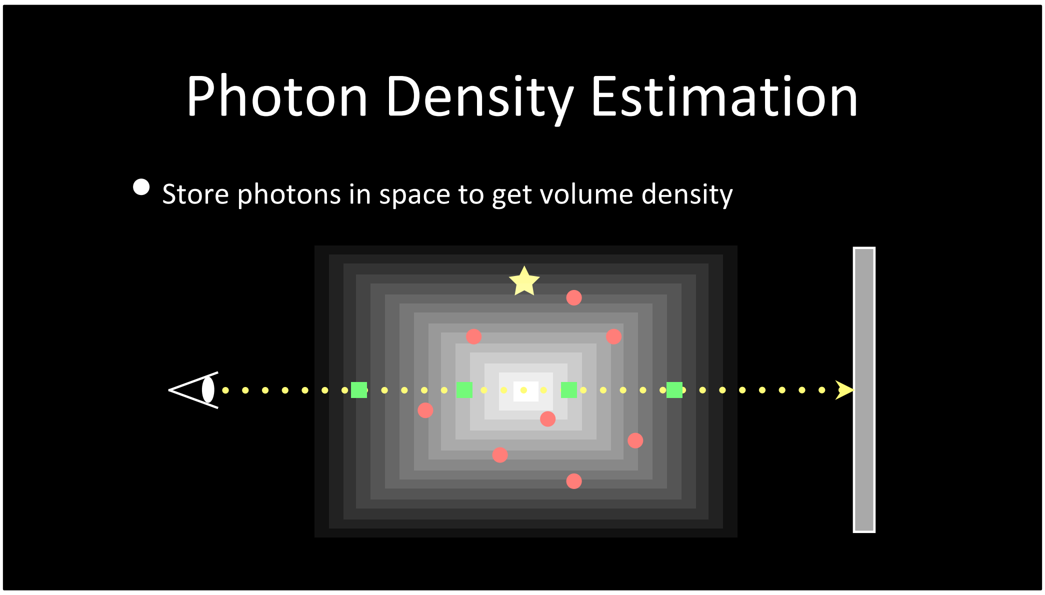




Photon Density Estimation

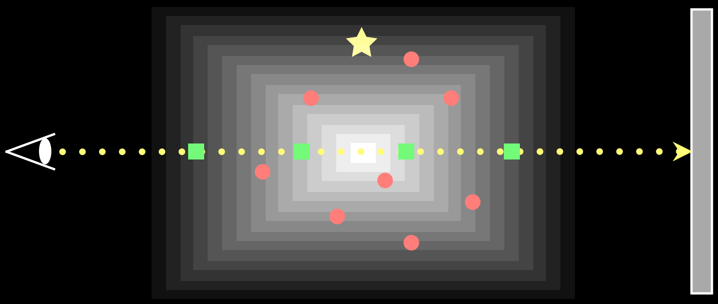
- Store photons in space to get volume density

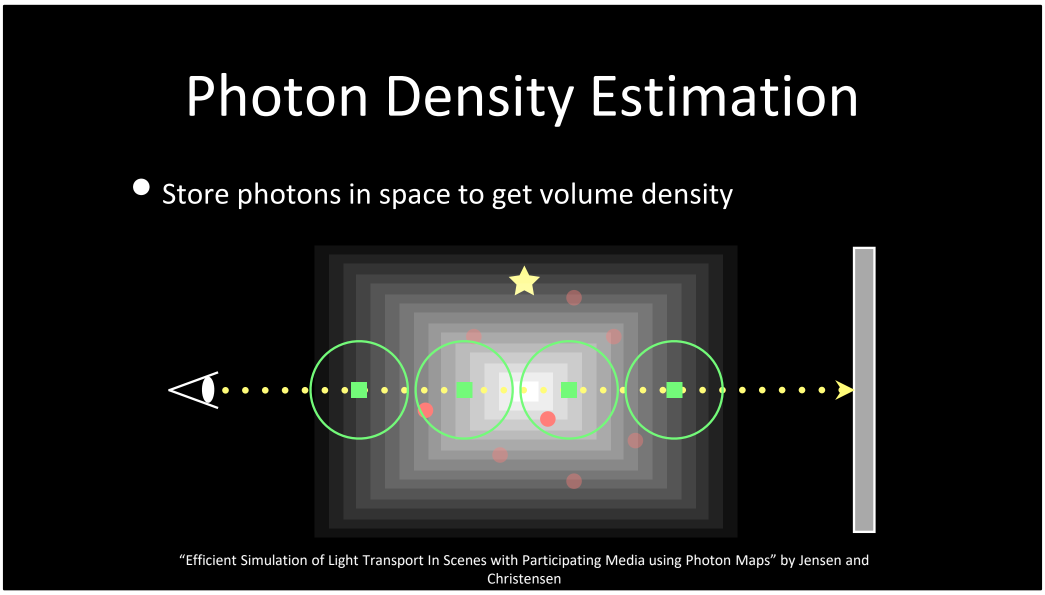




Photon Density Estimation

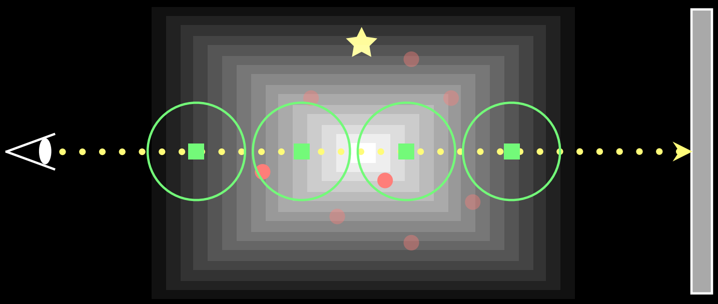
- Store photons in space to get volume density



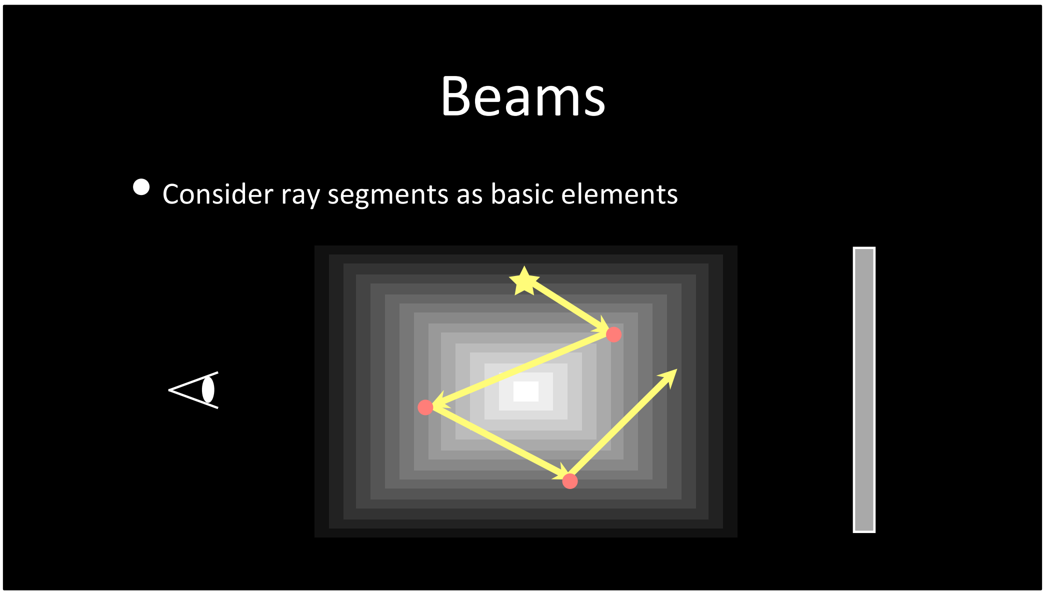


Photon Density Estimation

- Store photons in space to get volume density



"Efficient Simulation of Light Transport In Scenes with Participating Media using Photon Maps" by Jensen and Christensen

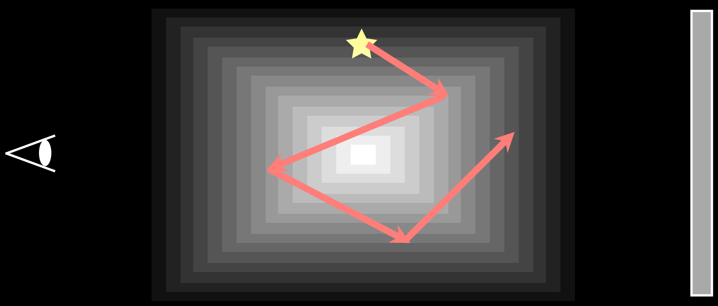


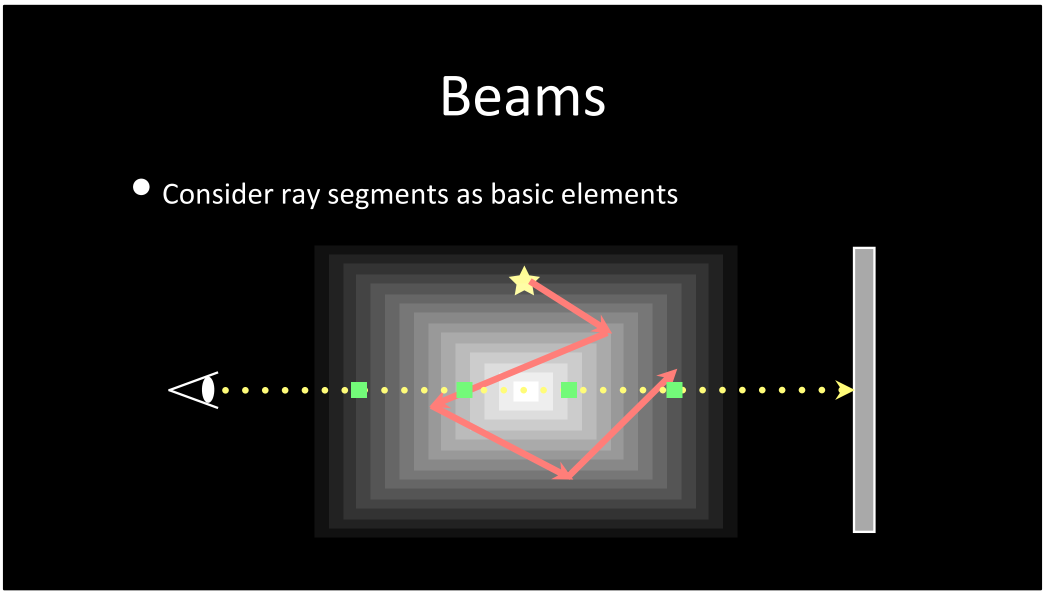
Beams

- Consider ray segments as basic elements

Beams

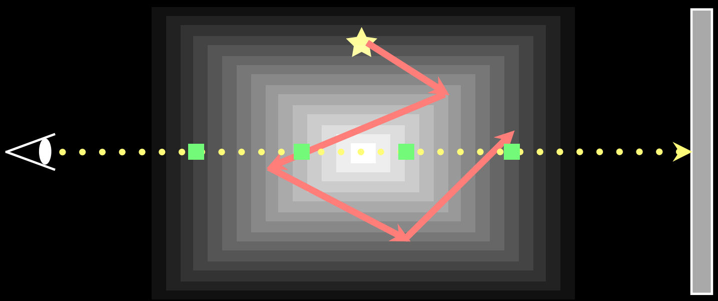
- Consider ray segments as basic elements





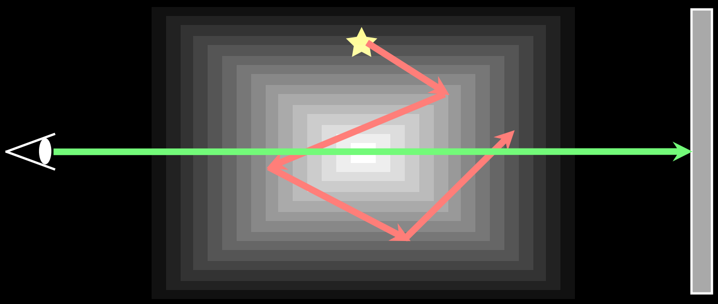
Beams

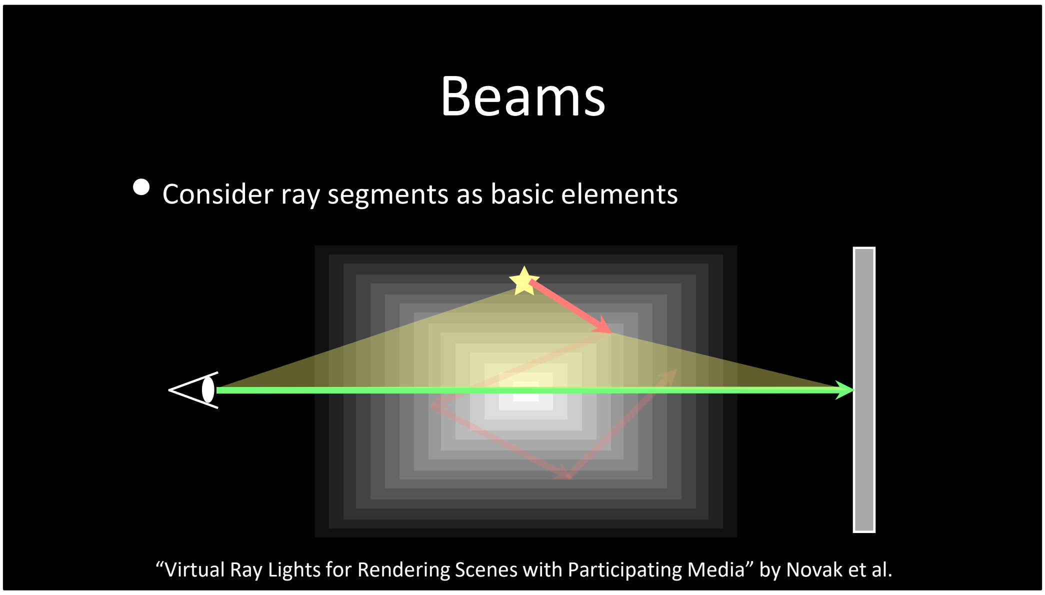
- Consider ray segments as basic elements



Beams

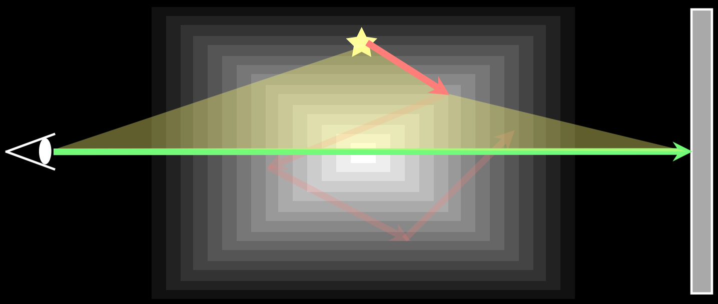
- Consider ray segments as basic elements



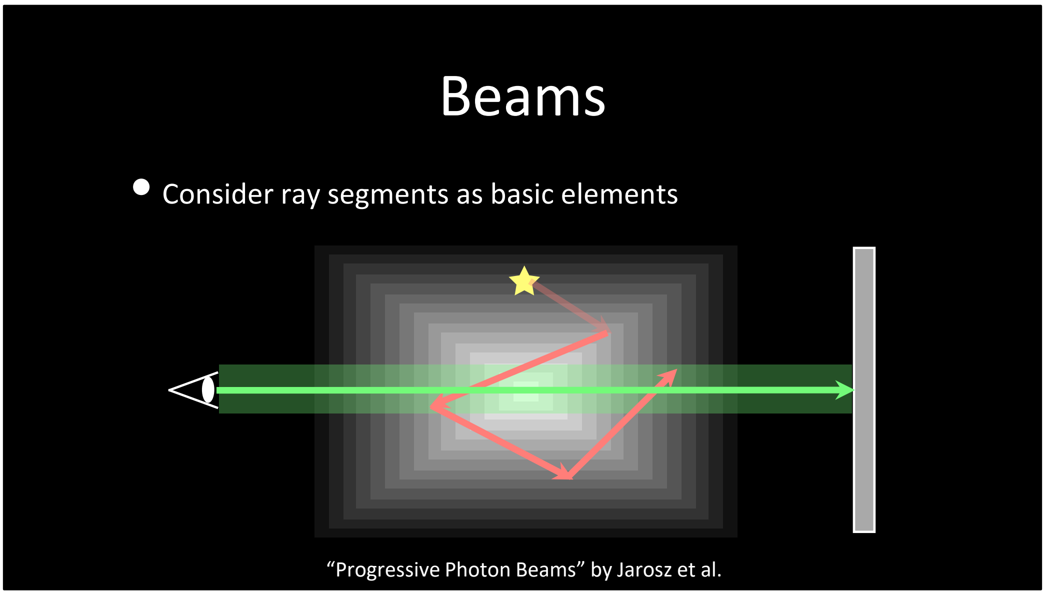


Beams

- Consider ray segments as basic elements

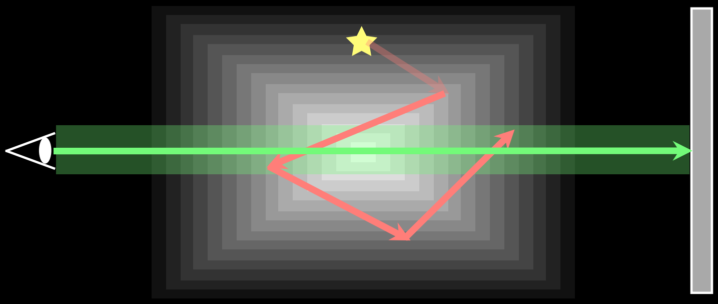


"Virtual Ray Lights for Rendering Scenes with Participating Media" by Novak et al.

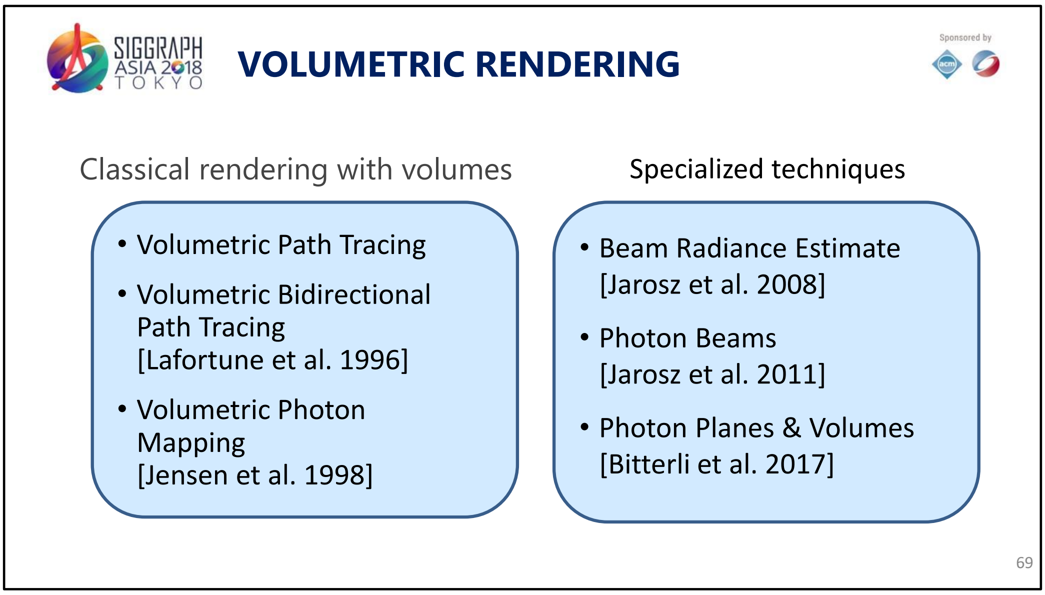


Beams

- Consider ray segments as basic elements



“Progressive Photon Beams” by Jarosz et al.



VOLUMETRIC RENDERING

Classical rendering with volumes

- Volumetric Path Tracing
- Volumetric Bidirectional Path Tracing [Lafortune et al. 1996]
- Volumetric Photon Mapping [Jensen et al. 1998]

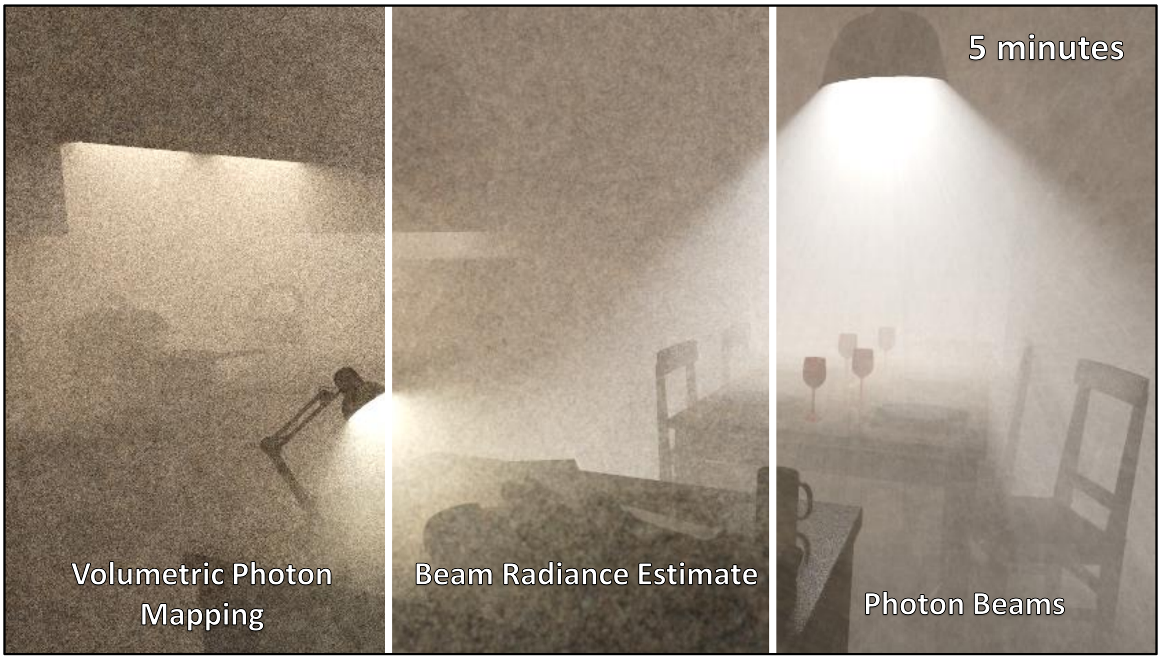
Specialized techniques

- Beam Radiance Estimate [Jarosz et al. 2008]
- Photon Beams [Jarosz et al. 2011]
- Photon Planes & Volumes [Bitterli et al. 2017]

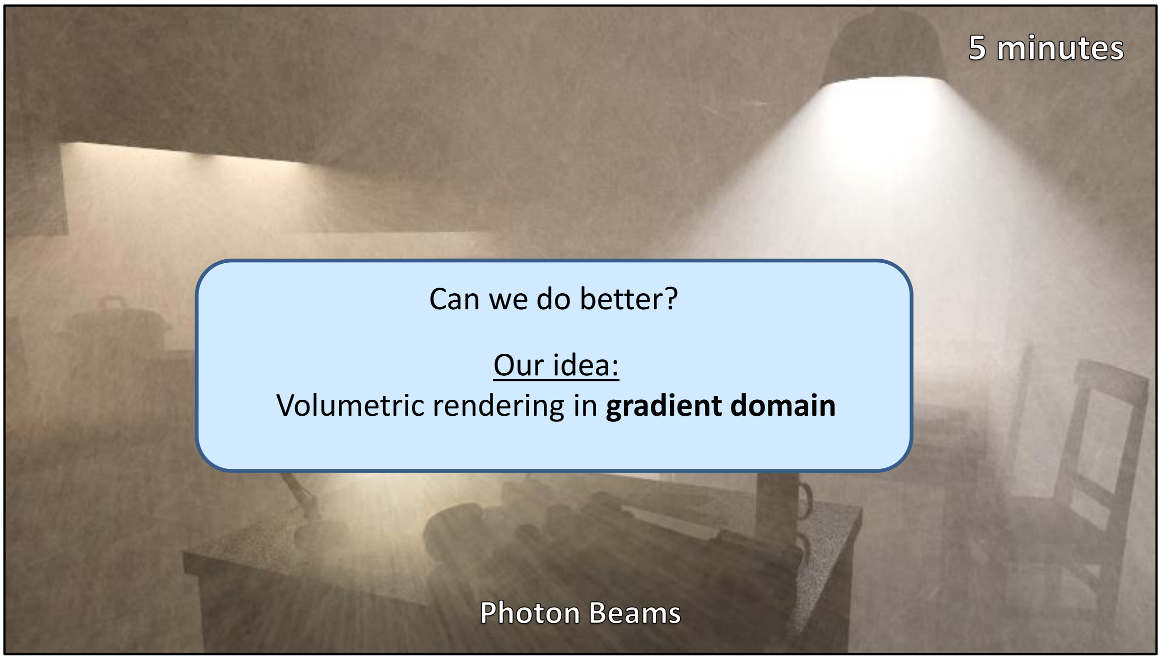
69

For computing volume interaction, there are two family of techniques. The first type includes the rendering techniques taken from surface rendering and extended to support participating media rendering like volumetric path tracing, volumetric bidirectional path tracing or volumetric photon mapping.

The second type are specialized rendering techniques that take advantage of the higher integration inside the volume like beam radiance estimate, photon beams, and photon planes.



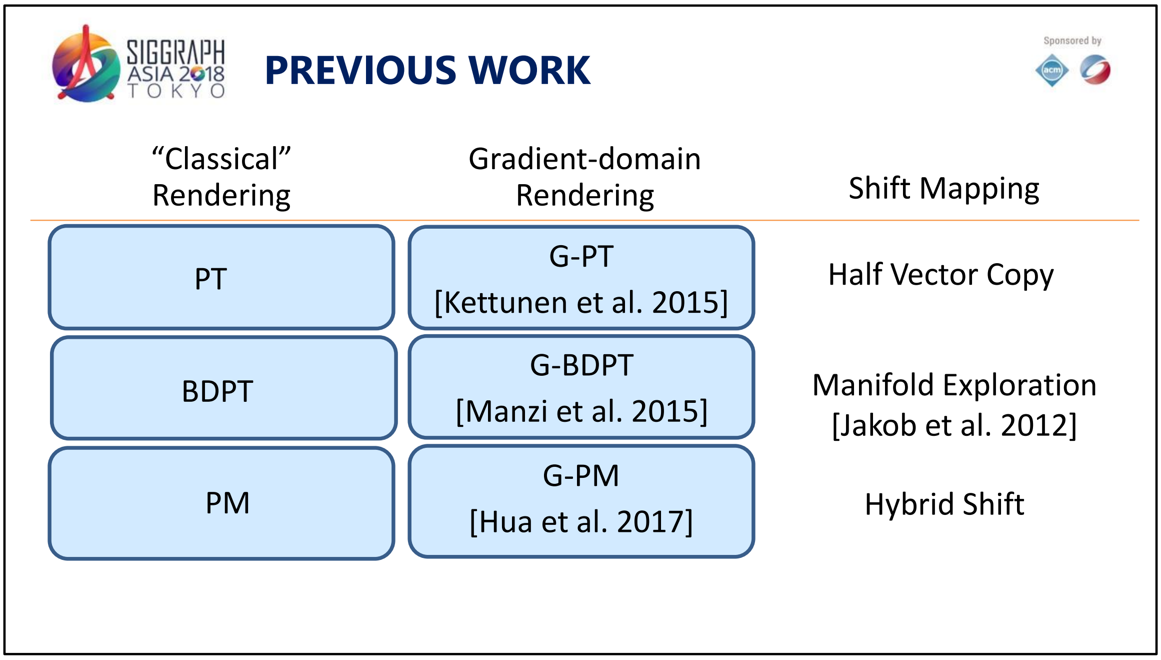
Here are the results for a smoky kitchen scene with 5 minutes of rendering. This scene have been rendered with different photon density estimators. For this particular scene, photon beams provide the best result for the same rendering time.



Can we do better?

Our idea:
Volumetric rendering in **gradient domain**

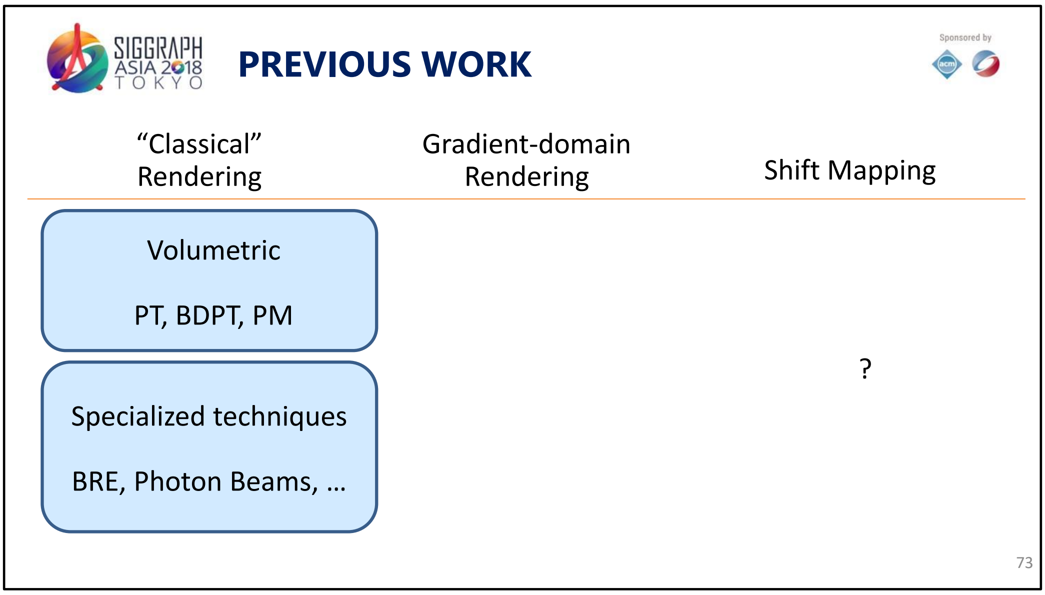
Given this best results for this particular scene, can we do better? One possibility is to exploit the smoothness in the image space. To do so our idea is to introduce volumetric rendering in the gradient domain.



PREVIOUS WORK

"Classical" Rendering	Gradient-domain Rendering	Shift Mapping
PT	G-PT [Kettunen et al. 2015]	Half Vector Copy
BDPT	G-BDPT [Manzi et al. 2015]	Manifold Exploration [Jakob et al. 2012]
PM	G-PM [Hua et al. 2017]	Hybrid Shift

So far we have explored light transport algorithms to render surfaces in the gradient domain. Depending on the rendering technique, different shift mapping operators are used. For example, for path tracing, Kettunen et al. [2015] proposed to use half vector copy as shift mapping. For bidirectional path tracing, Manzi et al. [2015] proposed to use manifold exploration. Finally, in Hua et al. [2017] have proposed a hybrid shift mapping to bring photon density estimation into gradient domain. Note that all these techniques are only for surface rendering and does not handle participating media rendering.

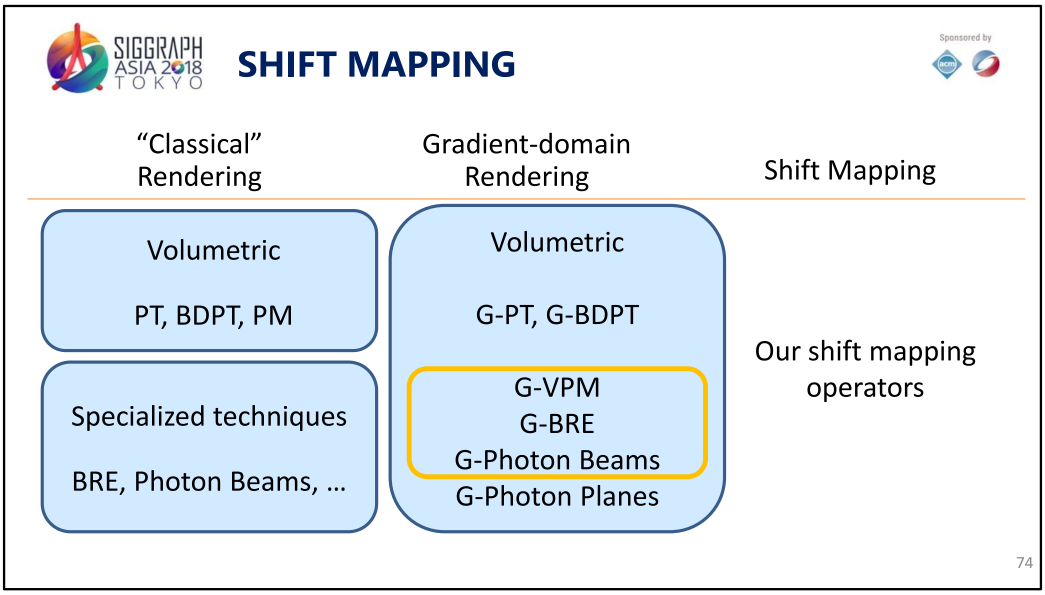


PREVIOUS WORK

"Classical" Rendering	Gradient-domain Rendering	Shift Mapping
Volumetric PT, BDPT, PM		?
Specialized techniques BRE, Photon Beams, …		

73

In this work, we are interested in bringing volume rendering into gradient-domain. For that, we need to design a shift mapping operator that take into consideration the distance sampling inside the media. We also this shift mapping operator efficient.

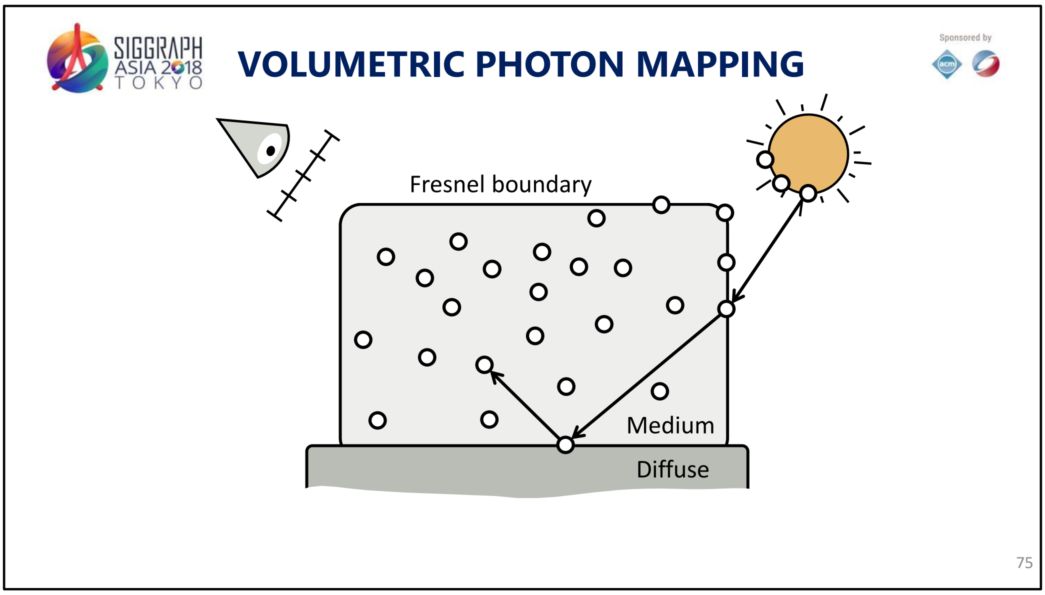


SHIFT MAPPING

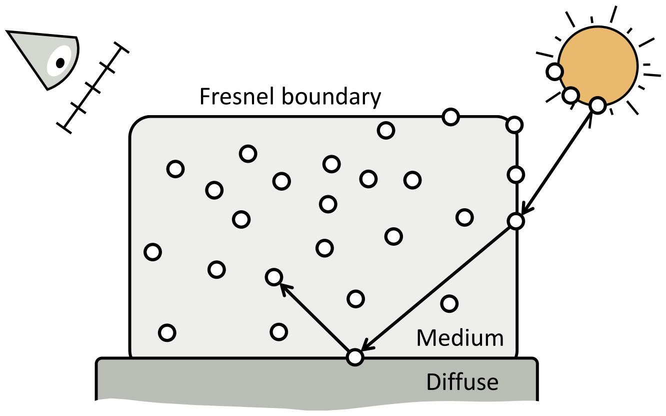
“Classical” Rendering	Gradient-domain Rendering	Shift Mapping
Volumetric PT, BDPT, PM	Volumetric G-PT, G-BDPT	Our shift mapping operators
Specialized techniques BRE, Photon Beams, …	G-VPM G-BRE G-Photon Beams G-Photon Planes	

Sponsored by
74

In this work, we have explored a wide variety of gradient-domain rendering technique for the volume. Due to time constraints, we will only focus on volumetric photon mapping, beam radiance estimate, and photon beams.

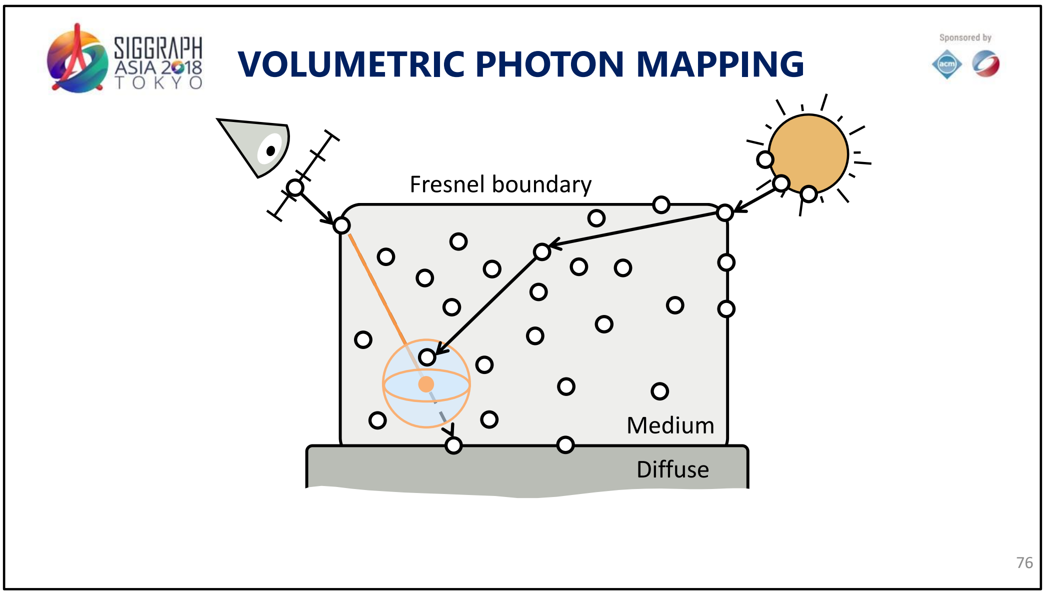


VOLUMETRIC PHOTON MAPPING

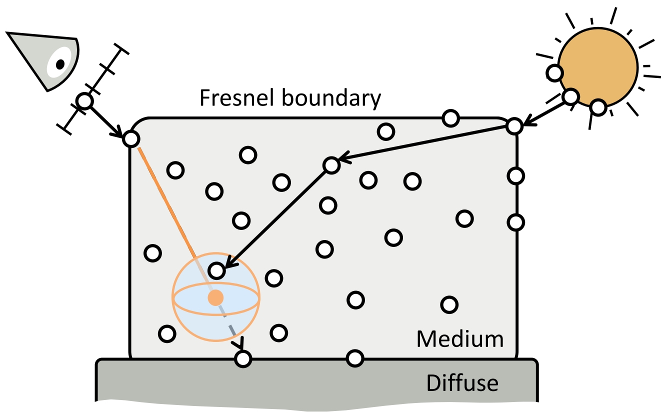


75

In the presence of participating media, it is now necessary to consider the transmittance of the path in the volume. In this case, we have the participating media coated by a specular boundary like glass. When we generate sub-paths from the light, some photon scatters inside the medium. Imaging that we have done this process several time, then we will have a number of photons inside the participating media.

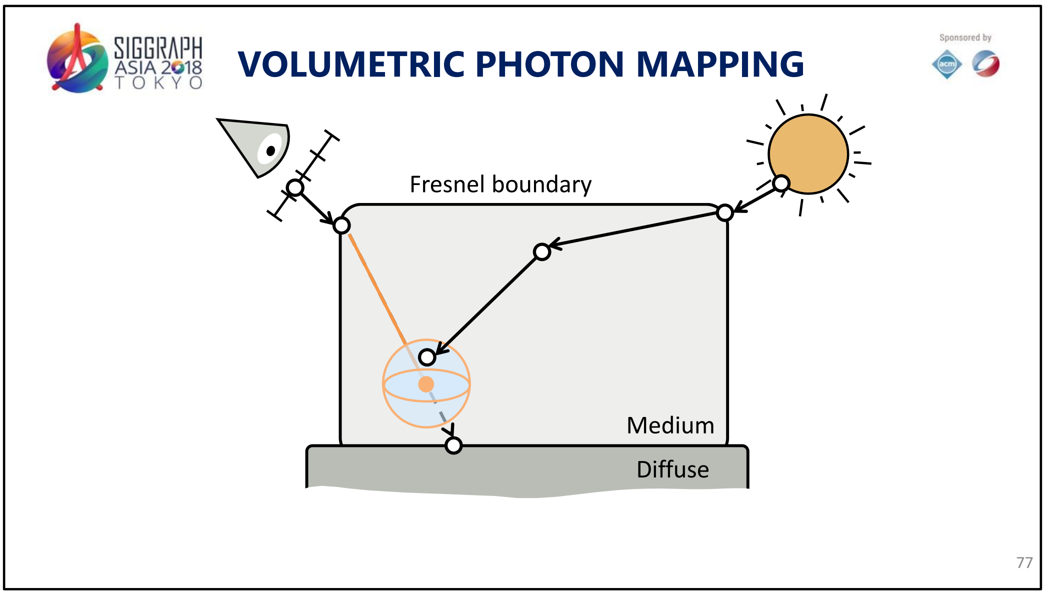


VOLUMETRIC PHOTON MAPPING

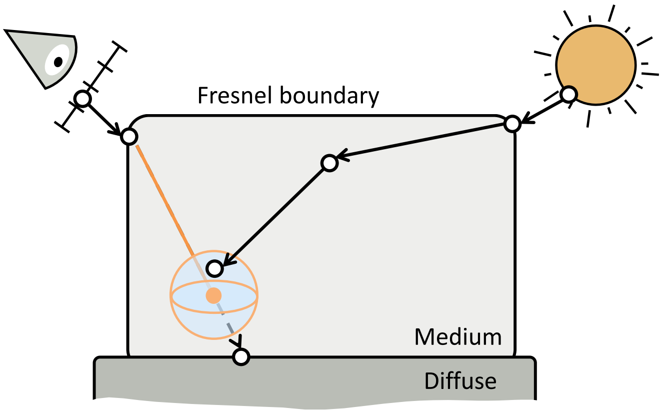


76

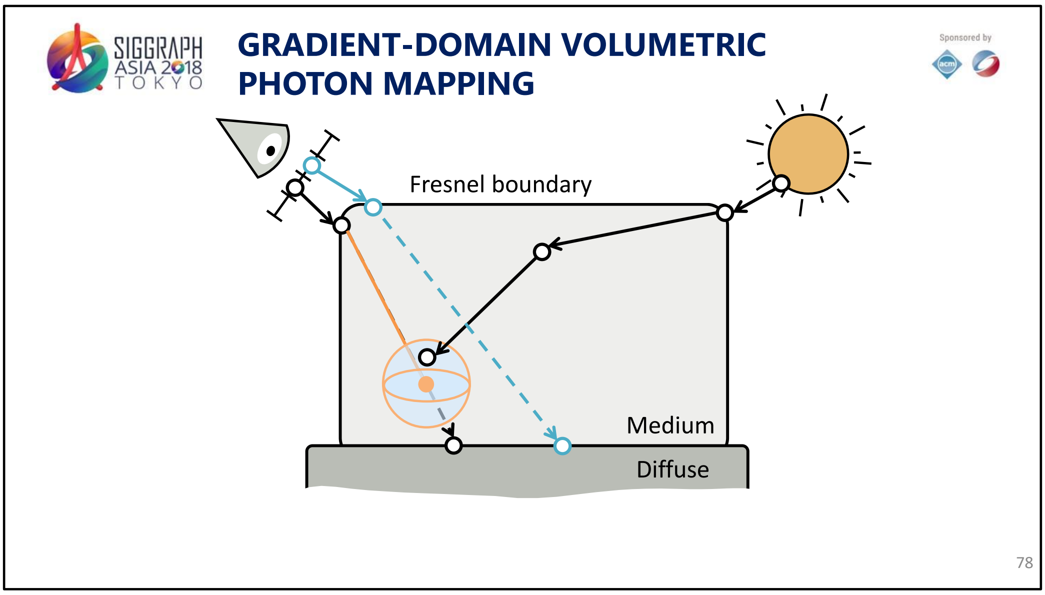
After generated enough volume photons, we starting to trace a camera subpath. During the tracing, we do not consider interactions with the volume and stop bounce over the surface after we hit a enough rough surface. After that, we pick an segment of the subpath that is inside the smoke and sample an random distance on it. This sampled distance is used to determine the location on the ray of the 3D photon density estimation kernel. In this example, the photon density estimation only gathers one light subpath.



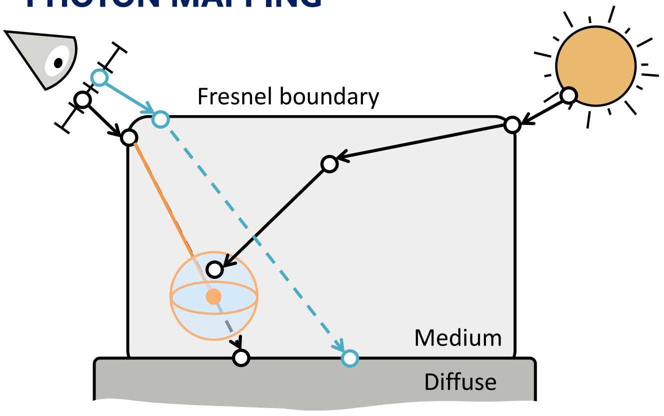
VOLUMETRIC PHOTON MAPPING



These two subpaths form an complete path due to the kernel. For simplicity, let's now focus only on how to shift this complete path.

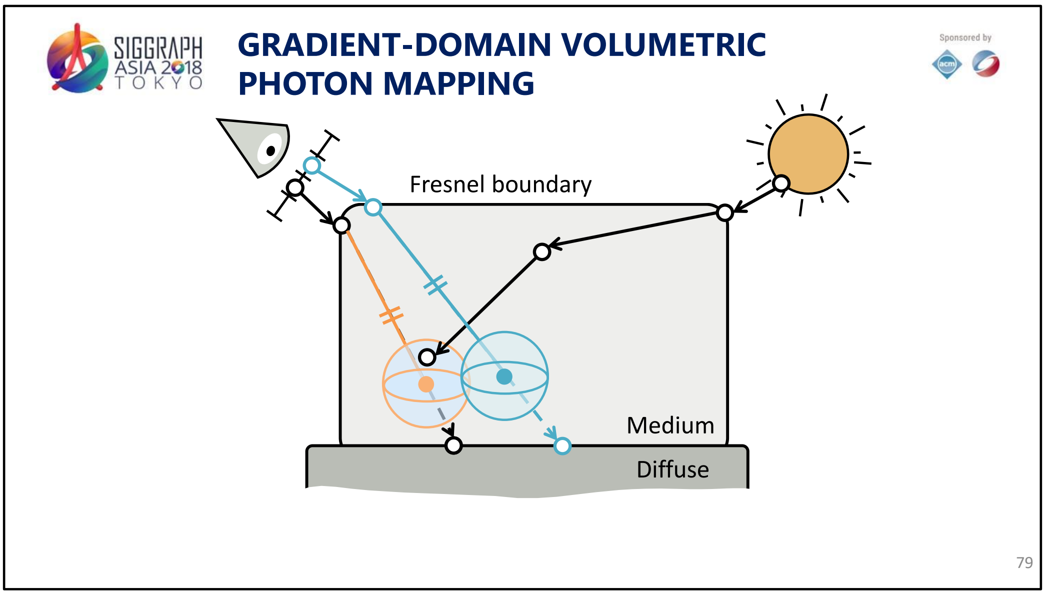


GRADIENT-DOMAIN VOLUMETRIC PHOTON MAPPING

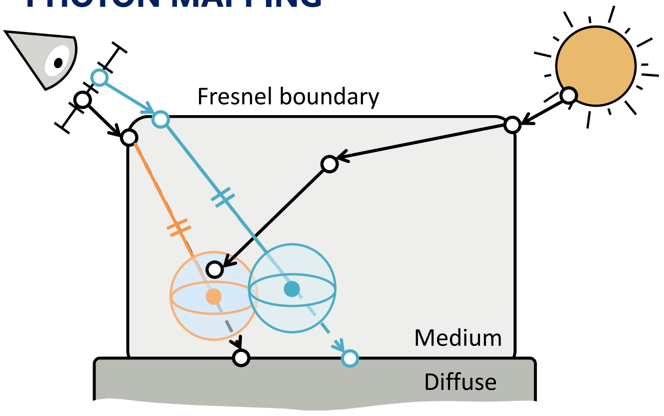


78

First, we will shift the camera subpath. To do so, an offset camera subpath is generated at the neighbor pixel with half-vector copy to make the offset path as similar as possible as the base path. Note that during this generation, we **do not** consider interactions with the volume.

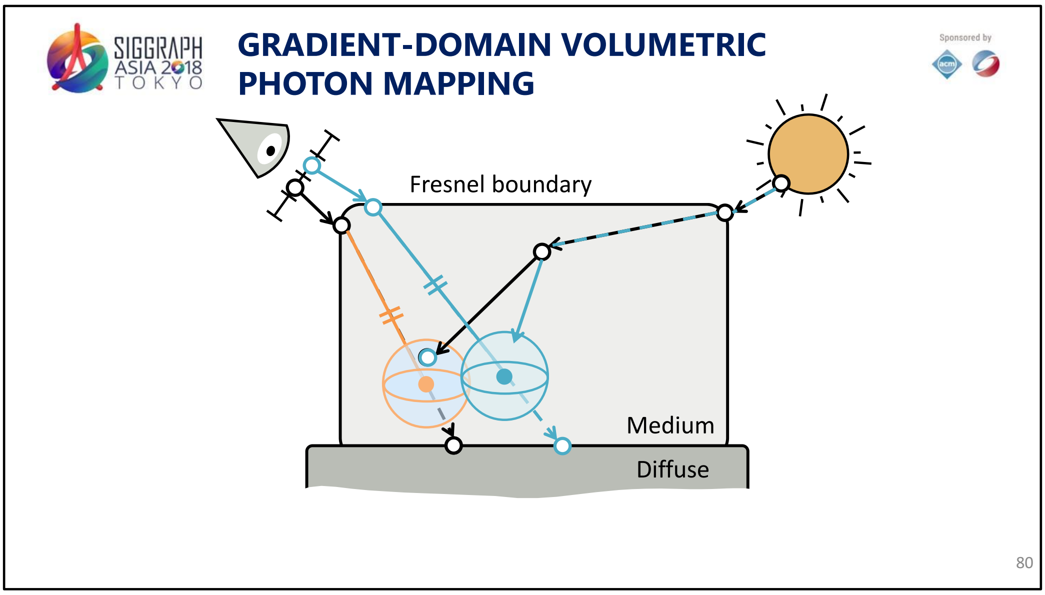


GRADIENT-DOMAIN VOLUMETRIC PHOTON MAPPING

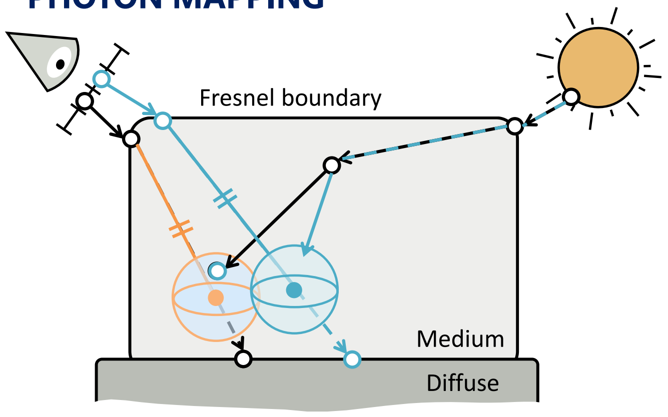


79

Then we need to decide where to perform the density estimation on the offset sensor path. We have tried several approach during this project and the simplest approach have been working the best in general: we just copy the distance sampled on the base path to the sensor path. This offset distance is used to place the offset photon density kernel. We also uses the same radius for base and offset kernel. Now we have completely shift the sensor subpath. Let’s now see how to shift the light subpath.

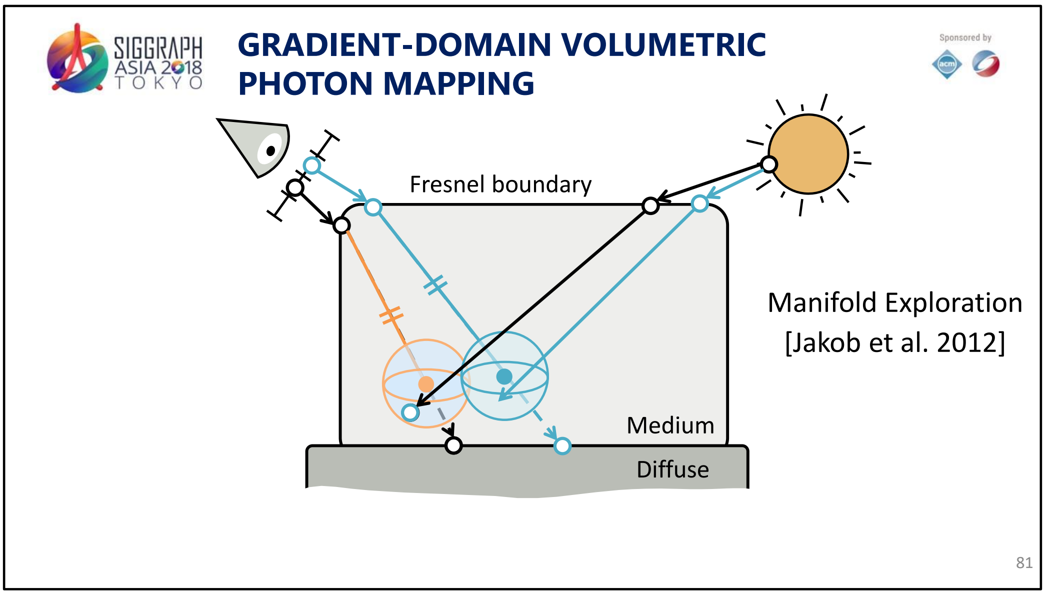


GRADIENT-DOMAIN VOLUMETRIC PHOTON MAPPING

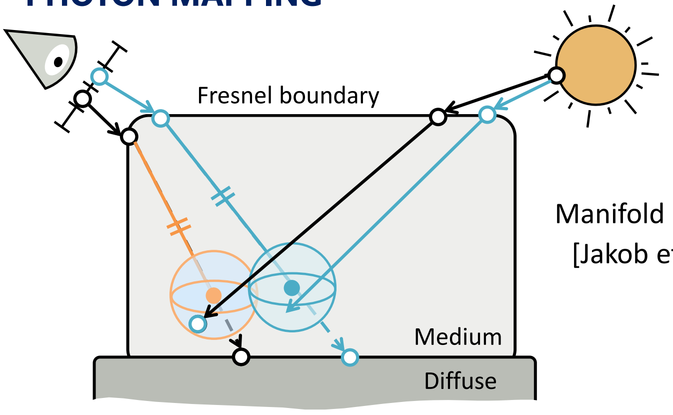


80

So, to shift the light subpath, we need first to deduce the position of the offset photon inside the offset density kernel. After finding this position, we generate the offset light subpath by reusing the base light subpath as soon as possible. For example, here we can directly reconnecting to the parent photon of the base light subpath.



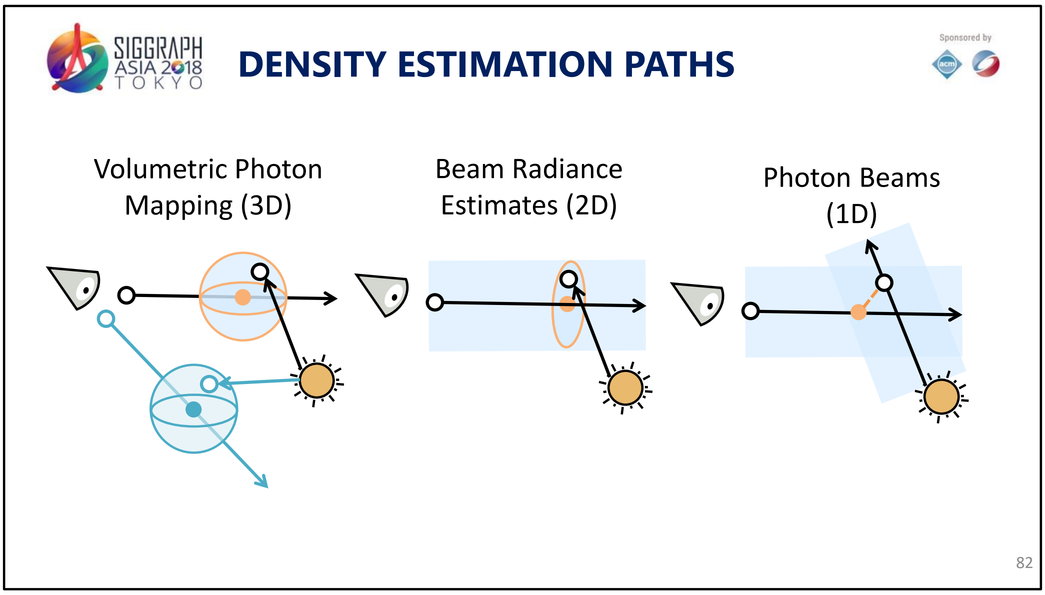
GRADIENT-DOMAIN VOLUMETRIC PHOTON MAPPING



Manifold Exploration
[Jakob et al. 2012]

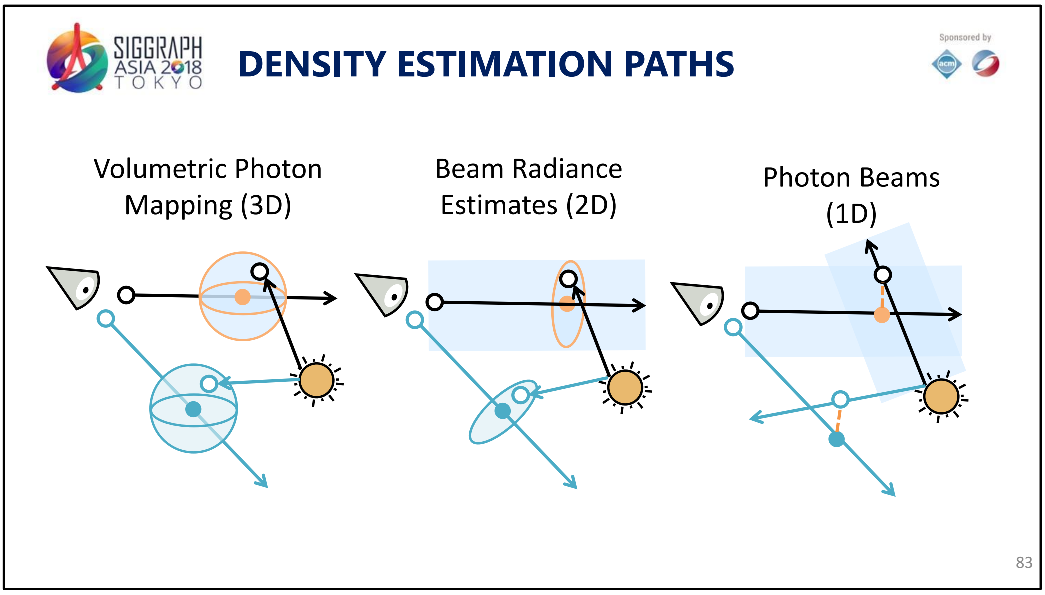
81

Sometimes the reconnection procedure to the base light subpath might be more complicated. For example, we can have another light subpath where the parent's photon is on the specular boundary. To shift this path, we need to reconnected to a deeper vertex. We achieve this by using manifold exploration [Jakob et al., 2012].



DENSITY ESTIMATION PATHS

So far we have been talking only about volumetric photon mapping and how to shift a complete path. However, what's about other rendering techniques like beam radiance estimate and photon beams.

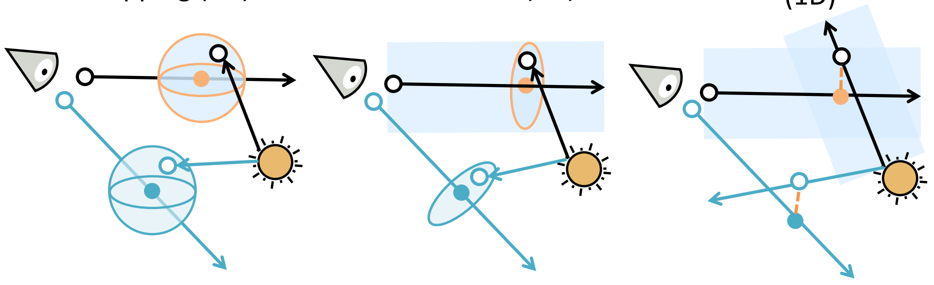


DENSITY ESTIMATION PATHS

Volumetric Photon Mapping (3D)

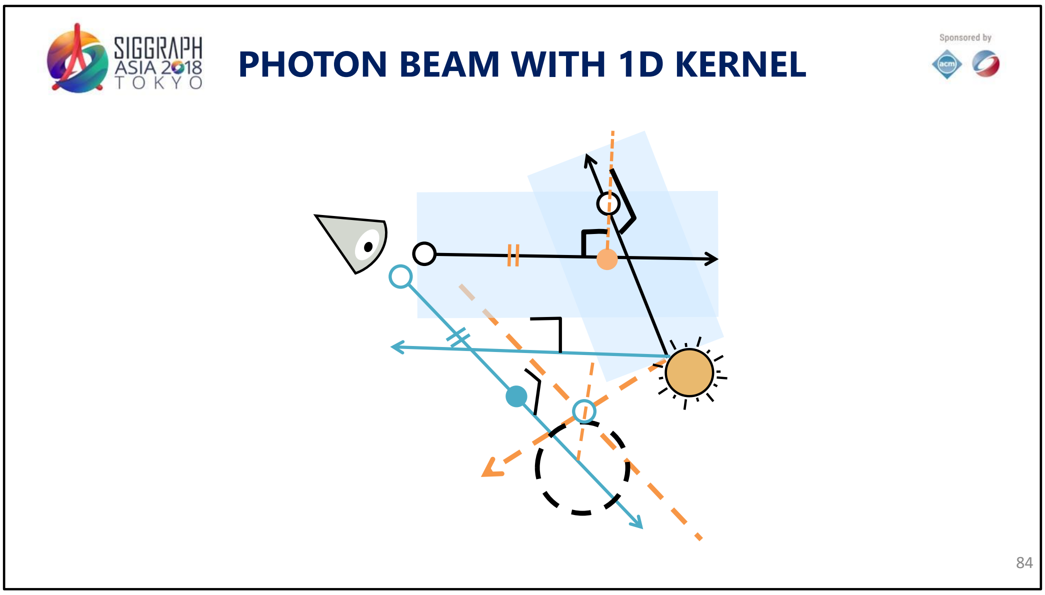
Beam Radiance Estimates (2D)

Photon Beams (1D)

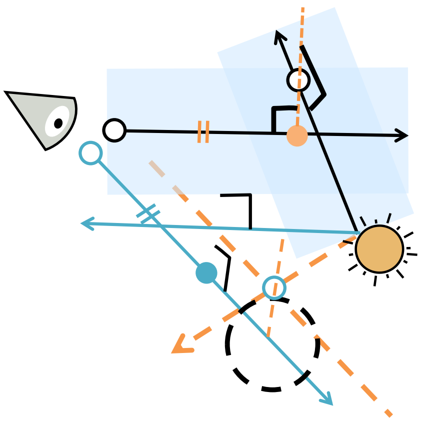


83

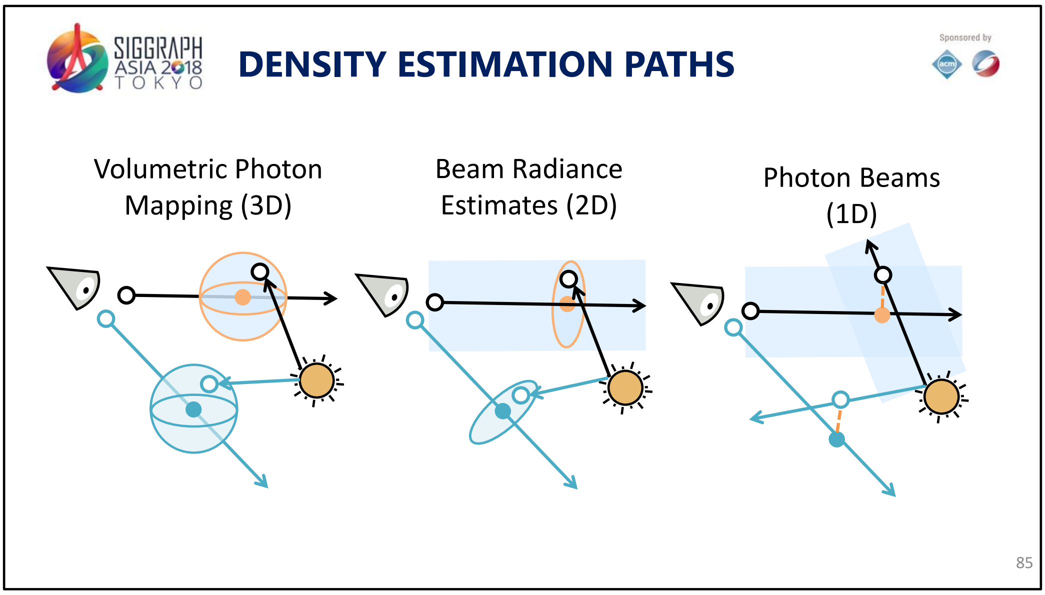
Which shift mapping operation can be used to brings these technique into gradient-domain? The answer in short: the shift mapping operator for other techniques are similar to the volumetric photon mapping case. Only the kernel constraints changes the shift mapping operator.



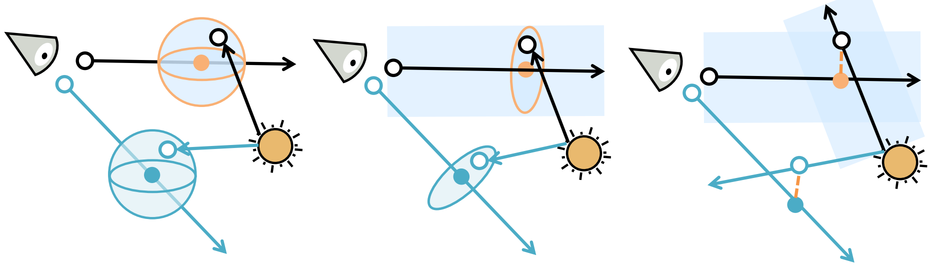
PHOTON BEAM WITH 1D KERNEL



For example the photon beam with 1D kernel is a challenging case. Using 1D kernel introduces complex geometry constrains. When shifting the photon beam, we need to maintain these constraints. We also want to keep the sampled distance and kernel distance. We proposed a geometrical shift mapping based on projection and translation which is able to satisfy all these conditions. At the end we are able to generate a shift photon beam that satisfy all these constrains. For a detail explanation, please refer to the paper [Gruson et al., 2018].

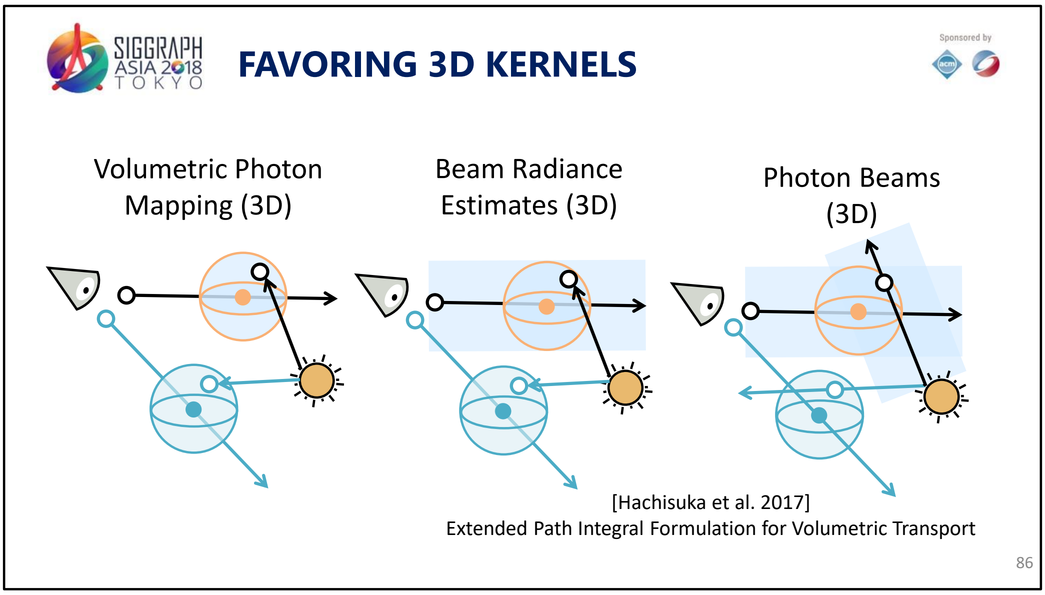


DENSITY ESTIMATION PATHS



85

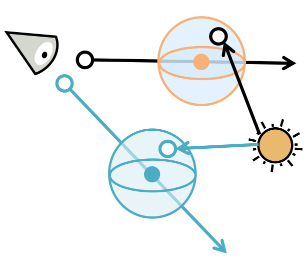
So due to the kernel dimensionality, shift mapping, especially for photon beam, is rather complicated and is difficult to generalize when we use manifold exploration. So to address these issue, we proposed to use …

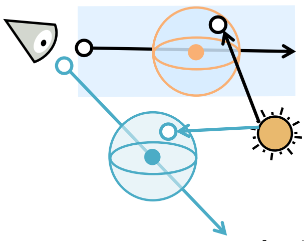


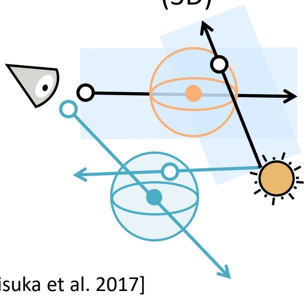
SIGGRAPH ASIA 2018 TOKYO

FAVORING 3D KERNELS

Sponsored by



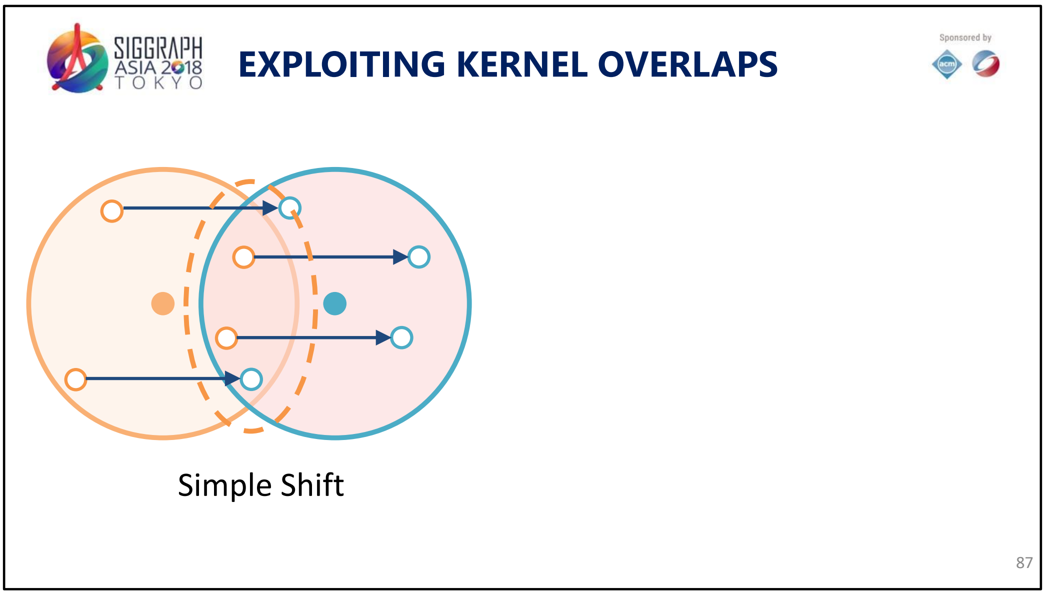




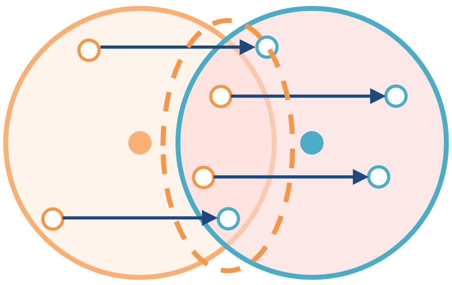
[Hachisuka et al. 2017]
Extended Path Integral Formulation for Volumetric Transport

86

… the technique by Hachisuka et al. [2017] which allows us to increase the photon density kernel dimensionality to 3D kernel for all these techniques. Using 3D kernel makes the shift mapping simpler at the cost of adding a bit of bias due to an higher dimensional kernel.



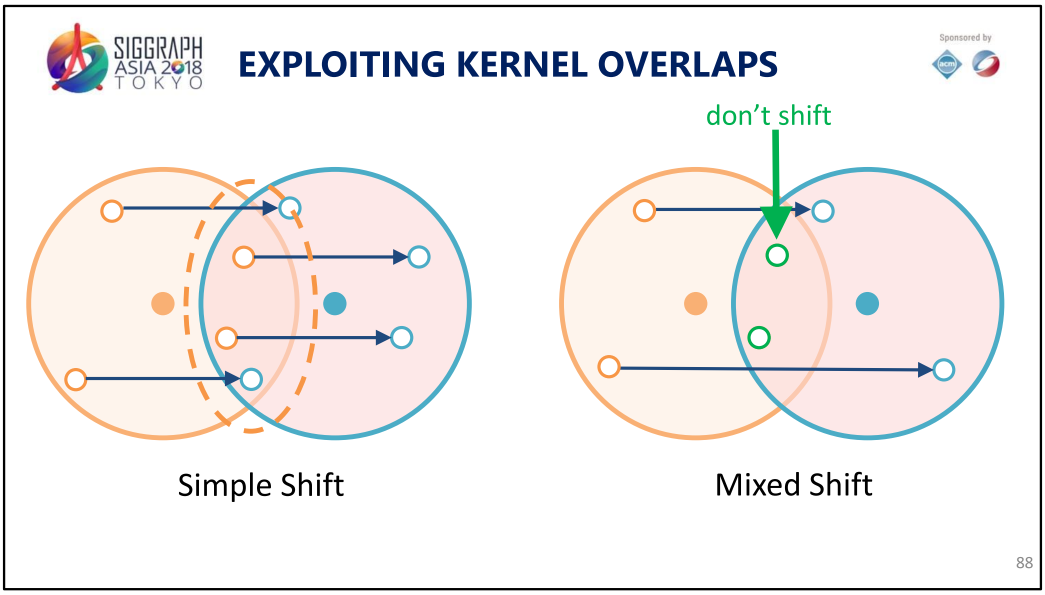
EXPLOITING KERNEL OVERLAPS



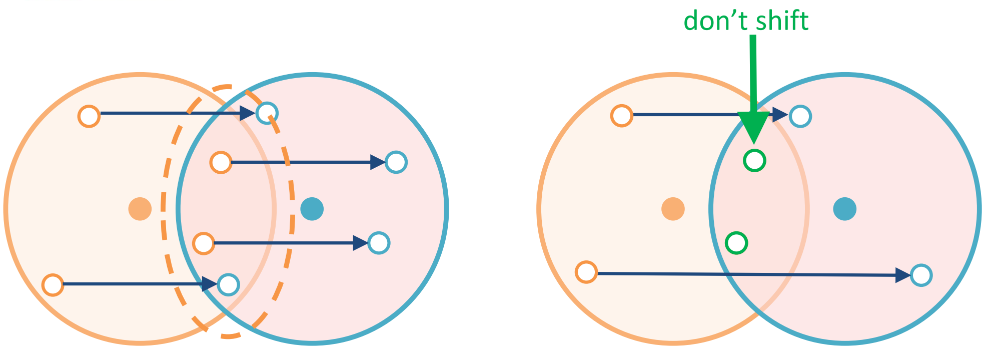
Simple Shift

87

But with this 3D kernel, we can take advantage to the kernel overlap. Indeed, when we collect photons from the base sensor path, we usually need to shift all of them to cover the space of the offset kernel. However, this strategy does not take to consideration that photon that are inside the overlapping region.



EXPLOITING KERNEL OVERLAPS



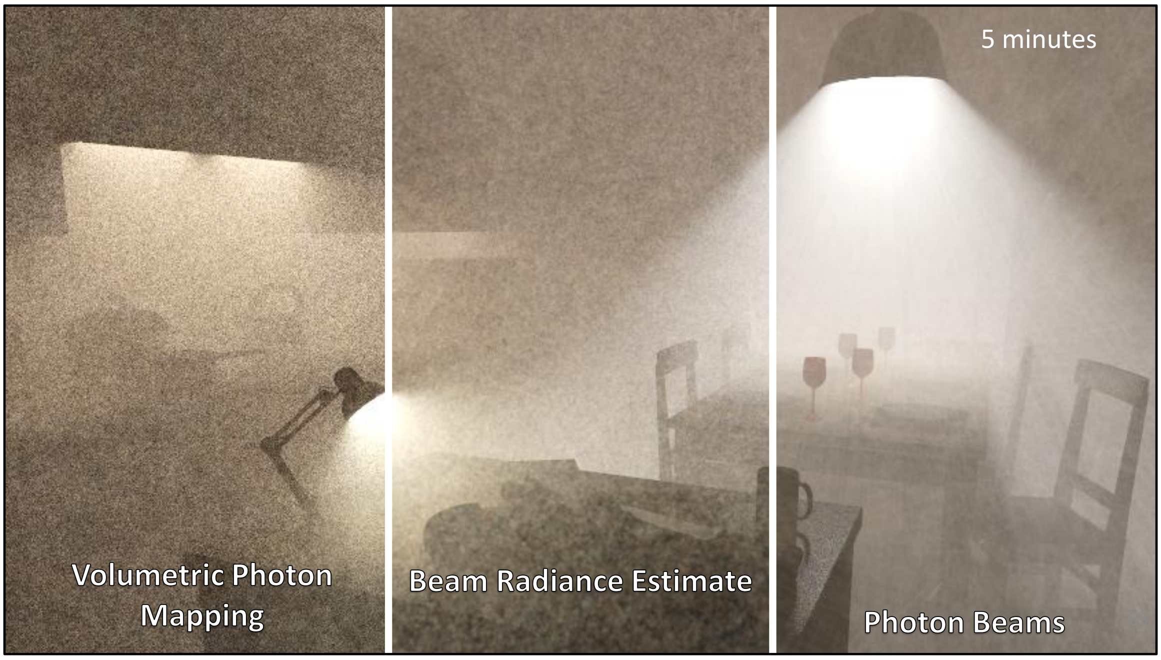
Simple Shift

Mixed Shift

88

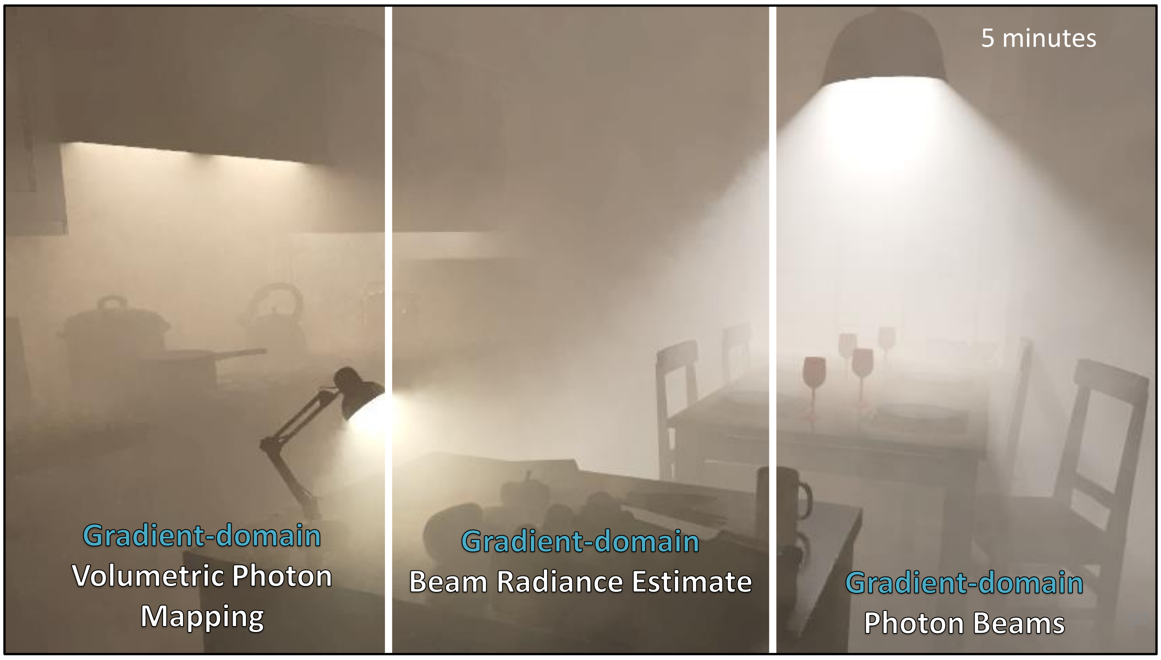
So, our idea is to reuse photons inside the overlapping region. By doing that we can save computation on shifting the light subpath.
For remaining photons we need a special way to handle them. In general, we have to shift such photons to the non-overlapped regions to ensure shift reversibility.

We call this shift mapping "mixed shift", which is a combination of simple shift and the special treatment of photons in the overlapped region.

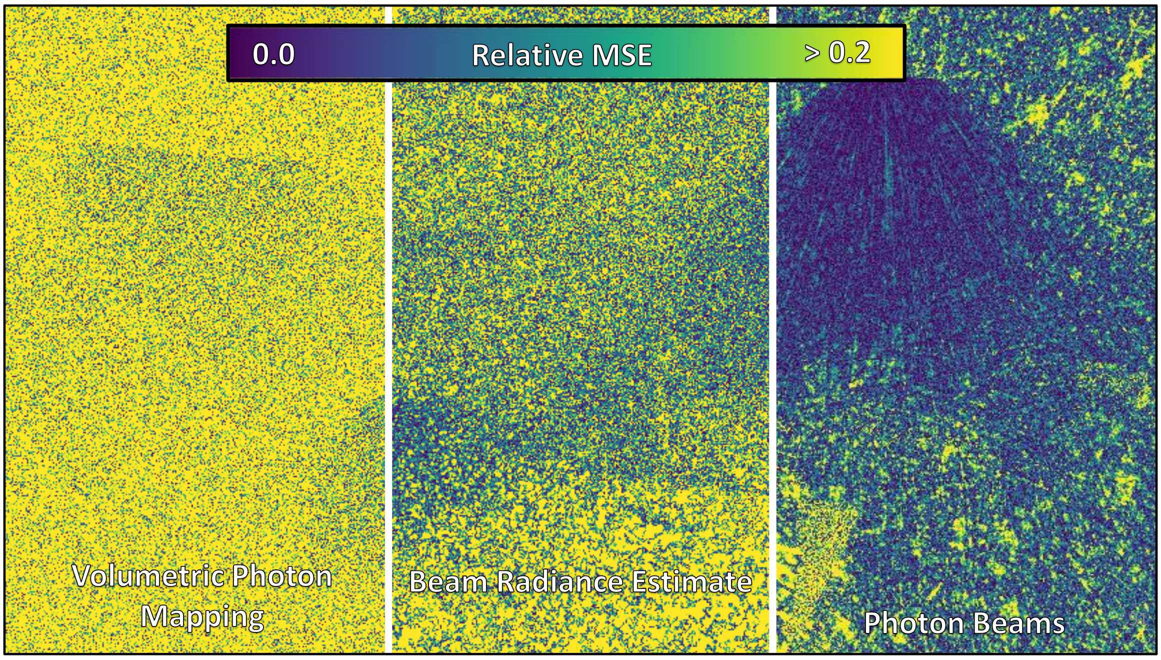


Now let’s look at some results. Here, again, the results for the smoky kitchen scene with 5 minutes of rendering. These are the results of different rendering techniques without gradient computation (primal-domain techniques).

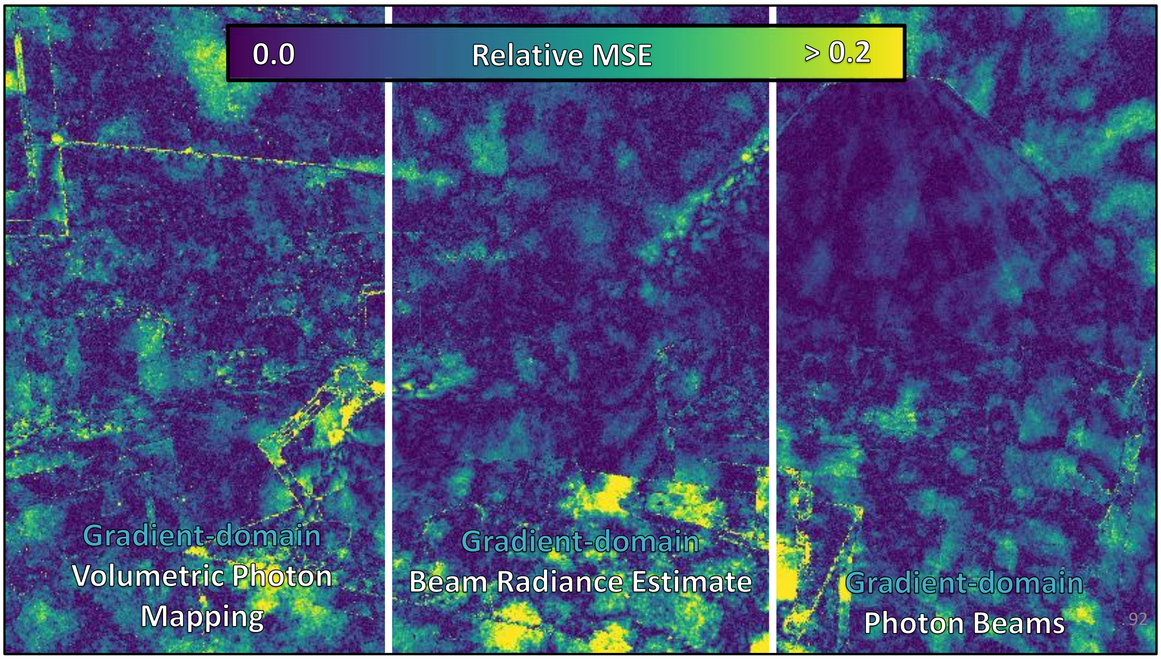
Now let’s look at gradient-domain counterpart.



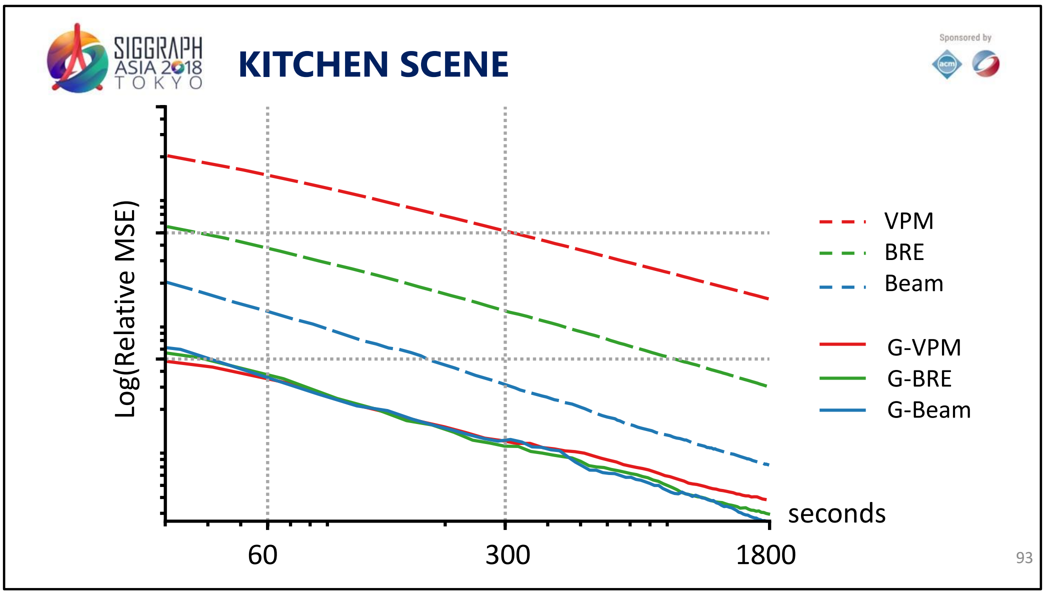
It is easy to see that gradient-domain techniques outperform the primal-domain rendering with a wide margin in the same amount of rendering time. (For best view, please flip back and forth.)



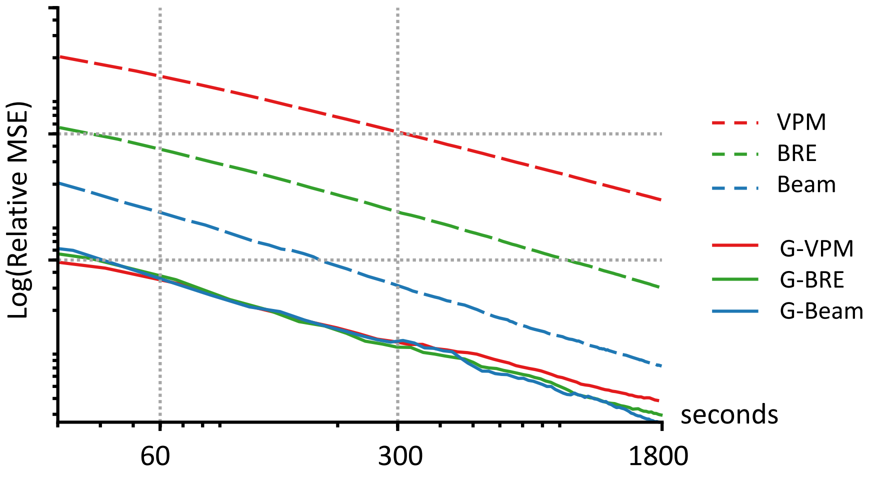
Another visualization: we visualize the relative MSE with pseudo colors. Here are the error maps of primal-domain techniques.



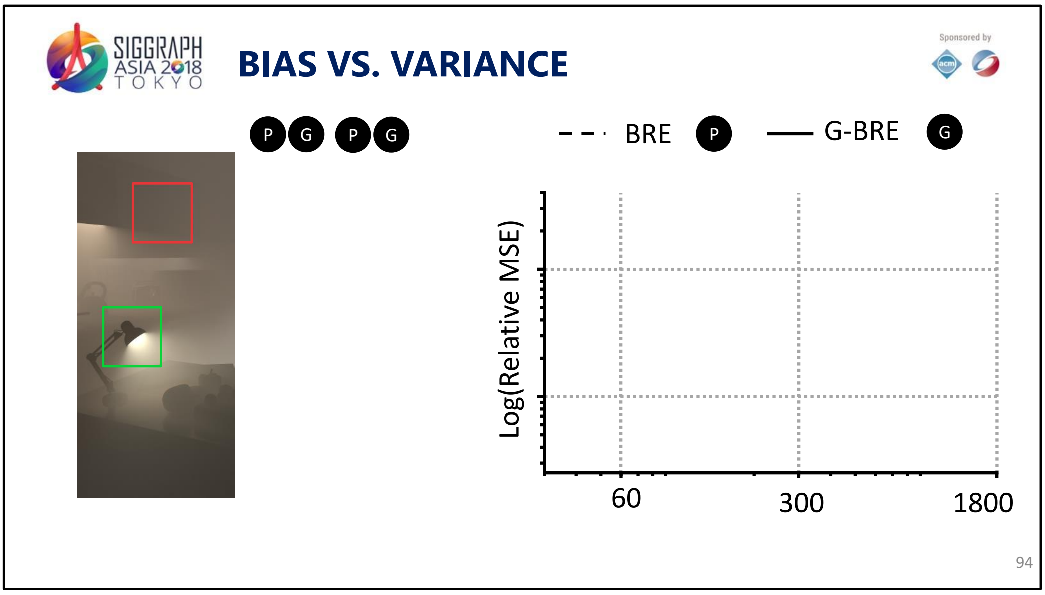
And here are the error maps of gradient-domain techniques. In this case, more blue means less errors.



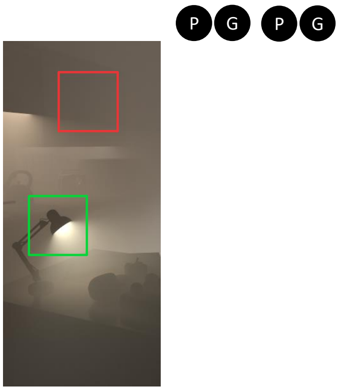
KITCHEN SCENE

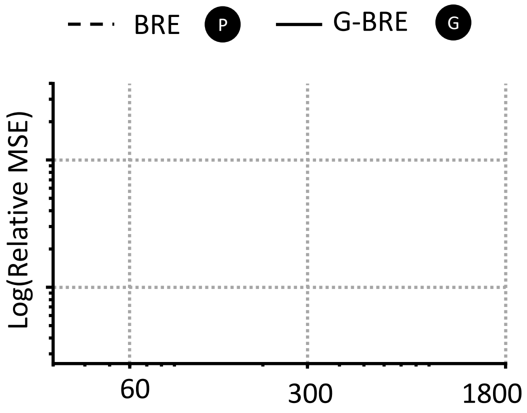


Here is the convergence of the different rendering techniques in term of relative MSE. Notation: VPM for Volumetric photon mapping, BRE for Beam Radiance estimate and Beam for photon beam. In this scene, the performance of the different gradient-domain techniques is almost the same.



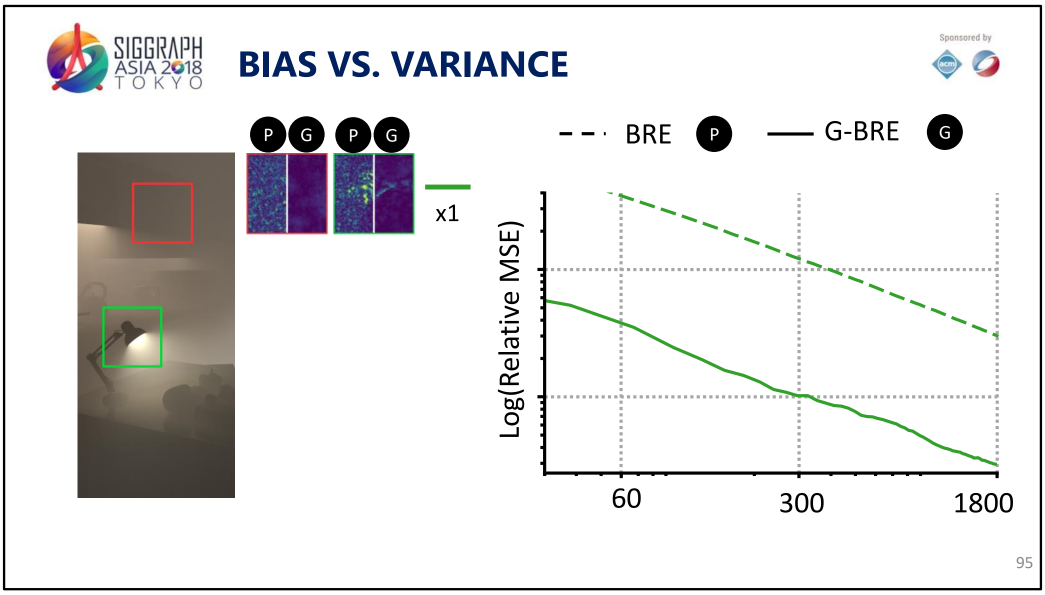
BIAS VS. VARIANCE



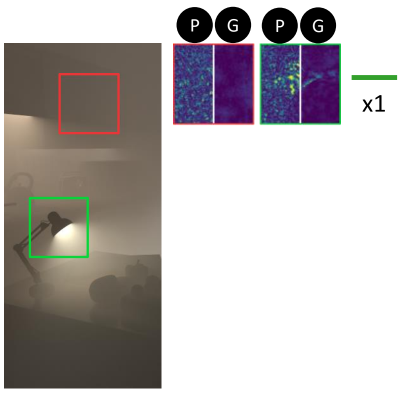


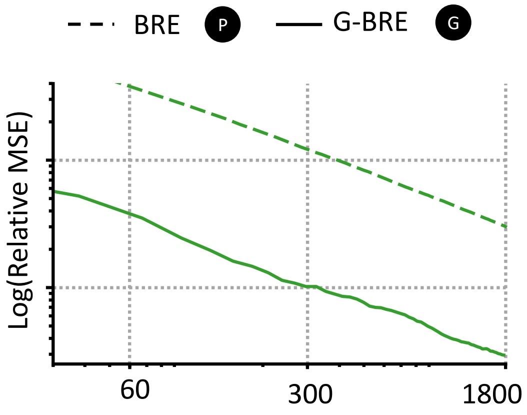
94

We can do another analysis inside this scene by considering bias and variance for primal and gradient-domain beam radiance estimate. In this slide P means primal and G means gradient. In this experiment, we visualize the error that is the sum of the bias and the variance.



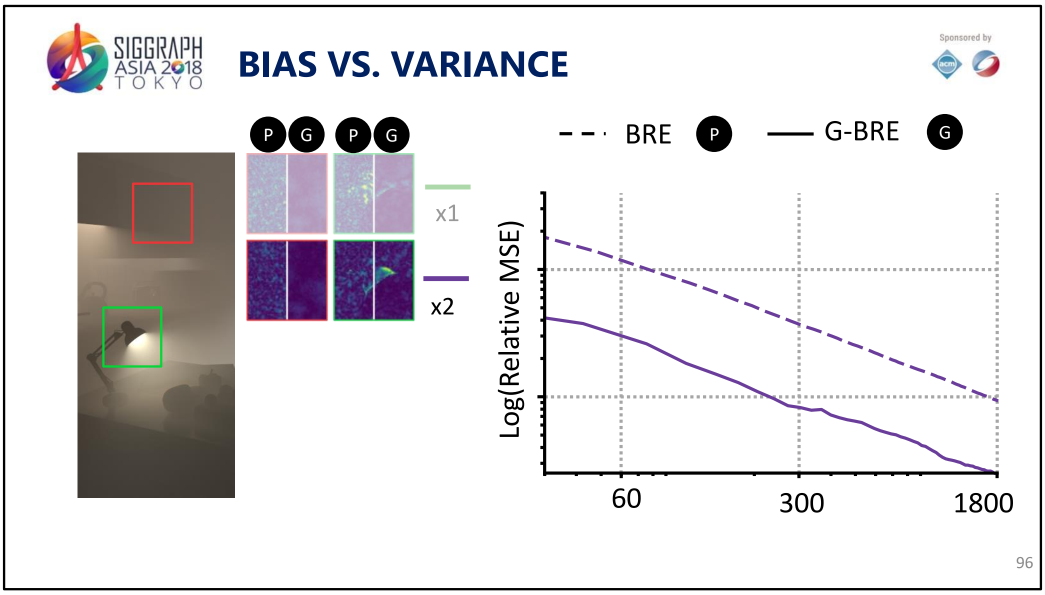
BIAS VS. VARIANCE



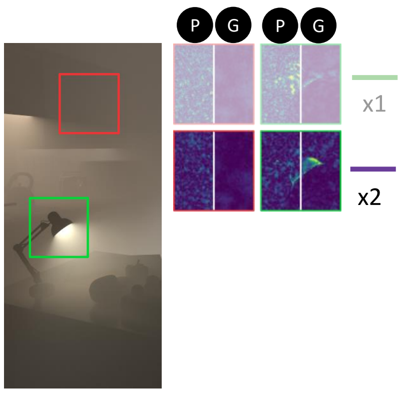


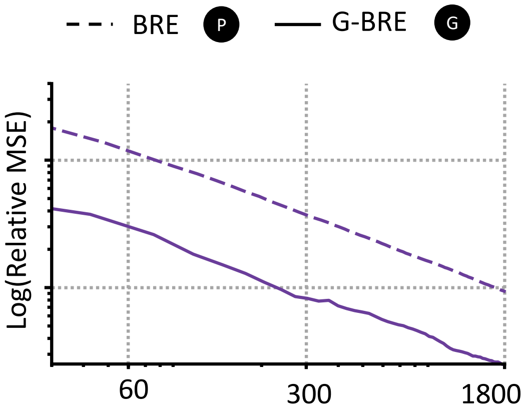
95

So, for a small enough radius, the variance dominates and gradient-domain technique reduce this variance effectively. However, …



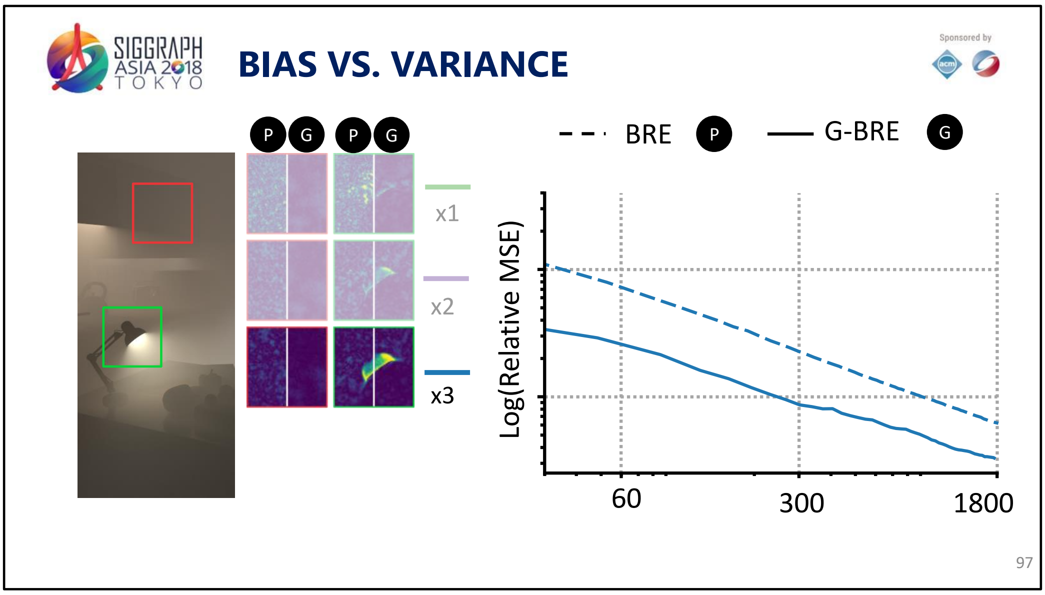
BIAS VS. VARIANCE



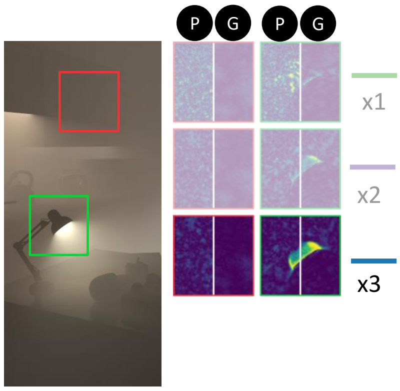


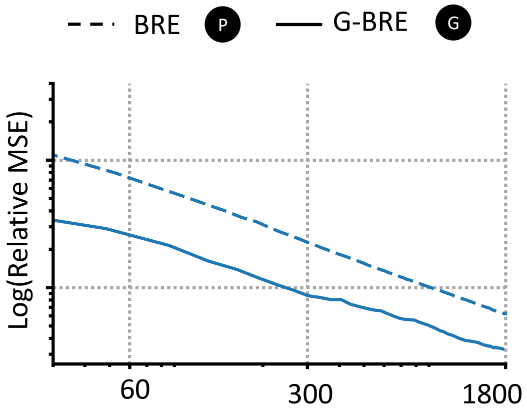
96

as we start to increase the radius, the proportion of bias increase, ….



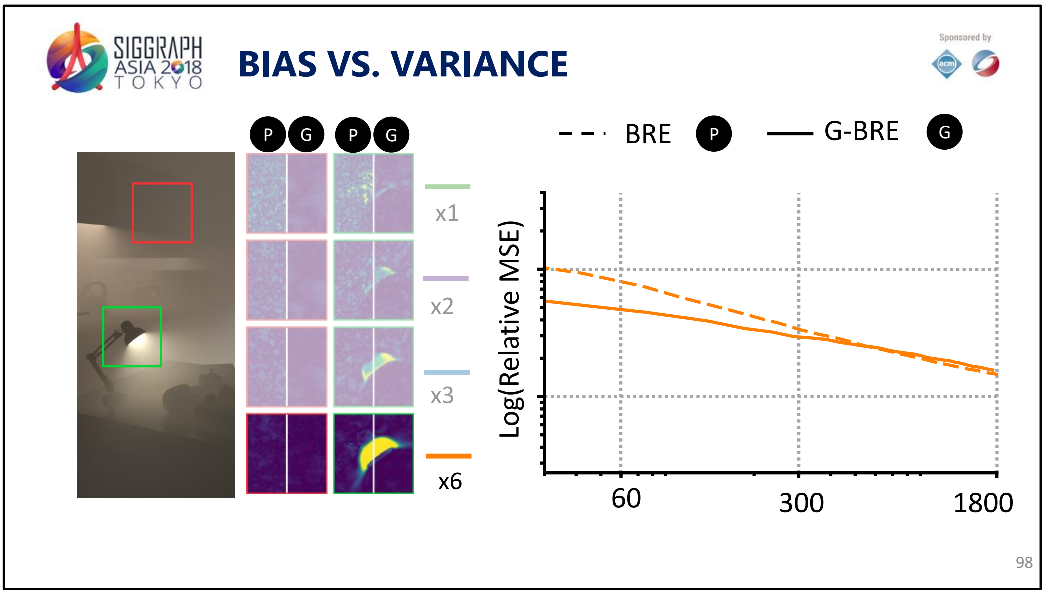
BIAS VS. VARIANCE



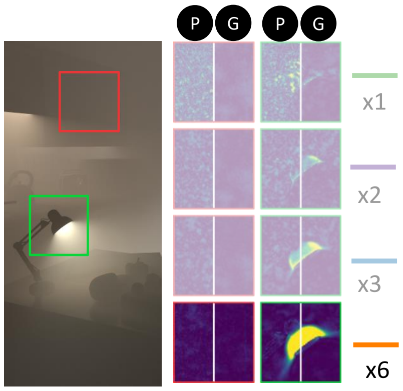


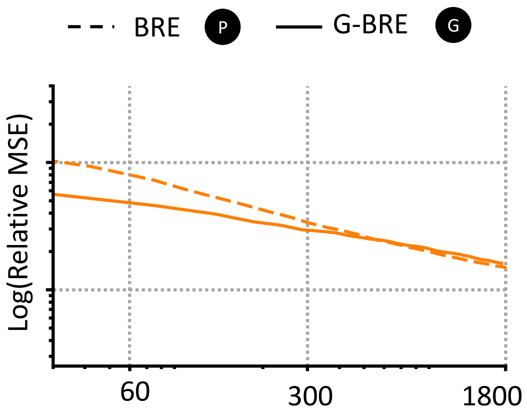
97

… as we can see inside the error map around the light source.



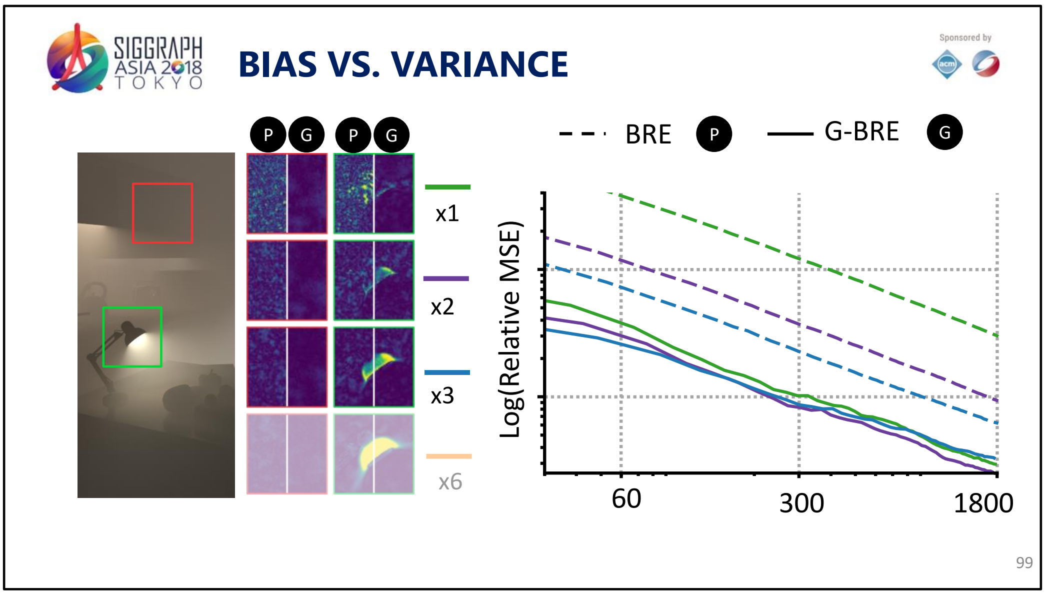
BIAS VS. VARIANCE



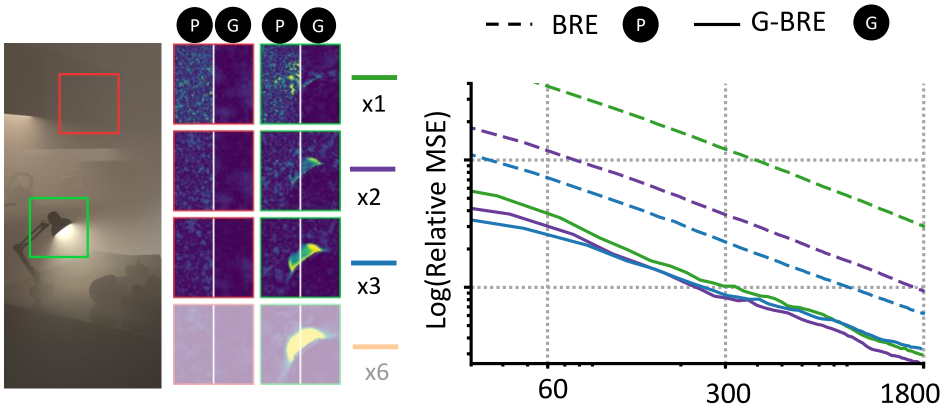


98

With an extreme big radius, the bias dominates, gradient-domain is only effective at early iterations.

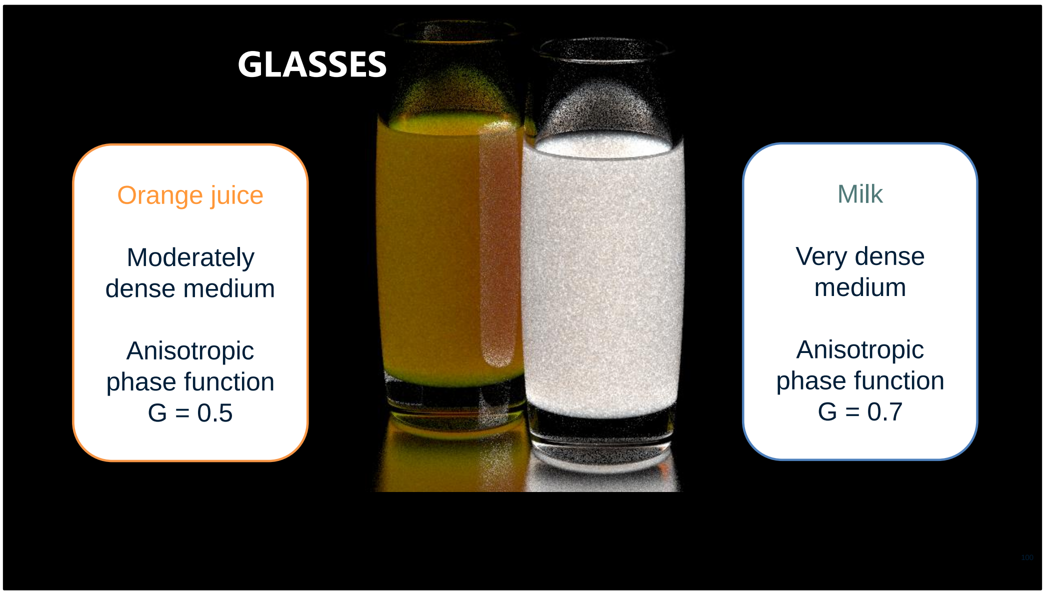


BIAS VS. VARIANCE

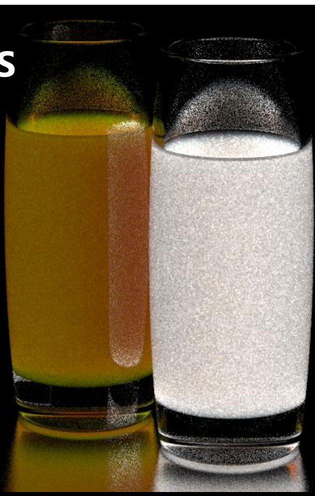


99

However, with a reasonable radius, the performance of gradient-domain technique is not affected with radius changes, which is not the case for the primal-domain technique.



GLASSES



Orange juice

Moderately dense medium

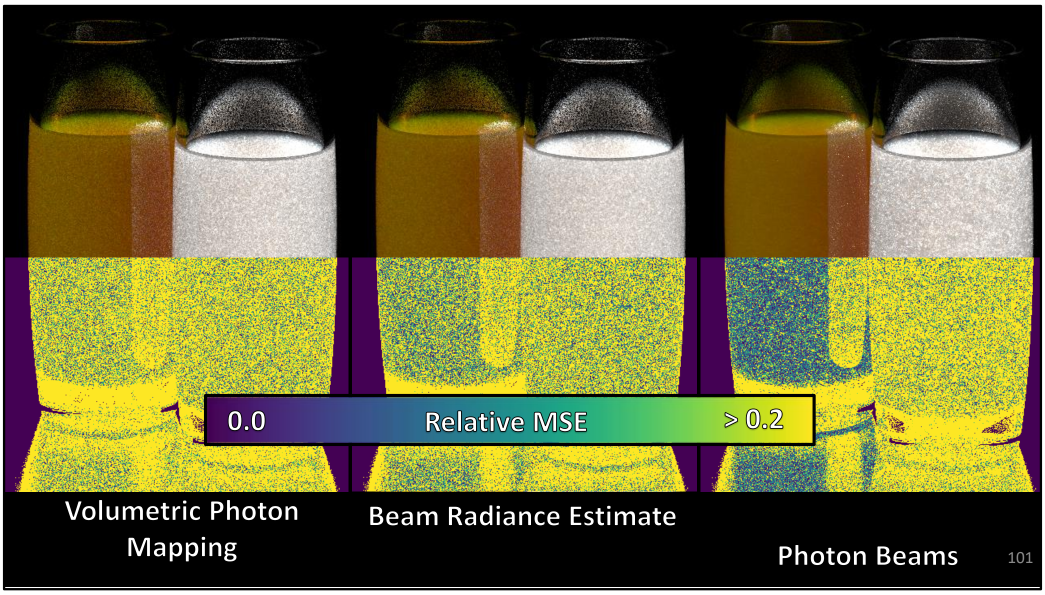
Anisotropic phase function G = 0.5

Milk

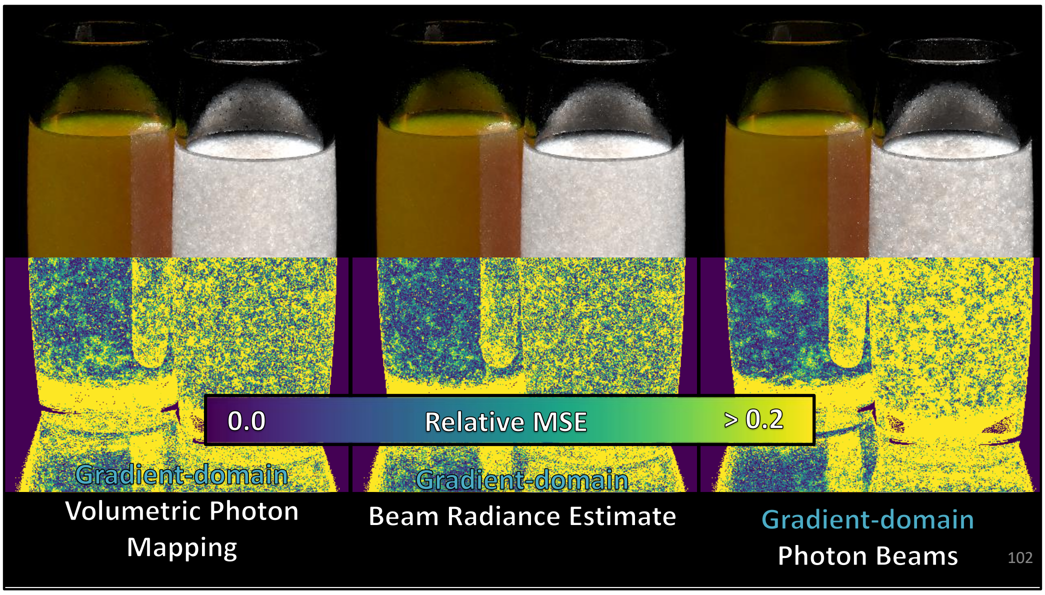
Very dense medium

Anisotropic phase function G = 0.7

Here is another challenging scene where we have two media inside glasses. The orange juice is moderately dense and have a slightly anisotropic phase function. The milk is very dense and is more anisotropic. We expect to have more noise in the milk region.

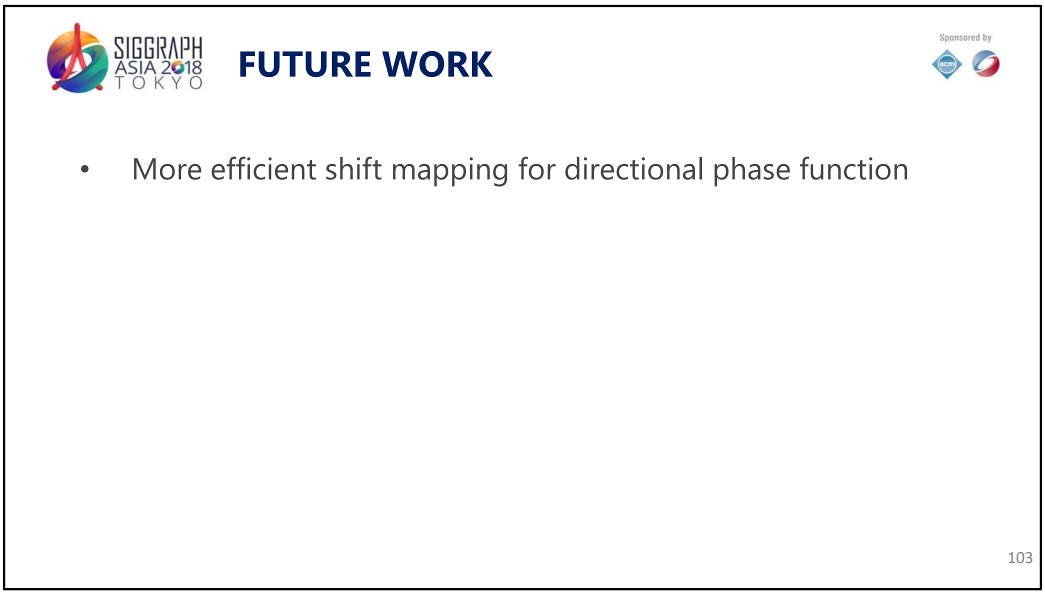


As we can see for primal-domain techniques, the noise level is pretty high. The milk regions across the different technique is more noisy than the orange juice part. Note that photon beams performs quite well on the orange juice but perform worst on the milk.



Here are the results of gradient-domain rendering.

Using gradient-domain improve significantly the orange juice part and moderately the milk part. The less improvement for the milk is due to higher anisotropic phase function and denser medium. Moreover, the gradient-domain photon beams in the milk region have even less performance due to the shifting cost and the fact that many photon beams are not relevant to shift.

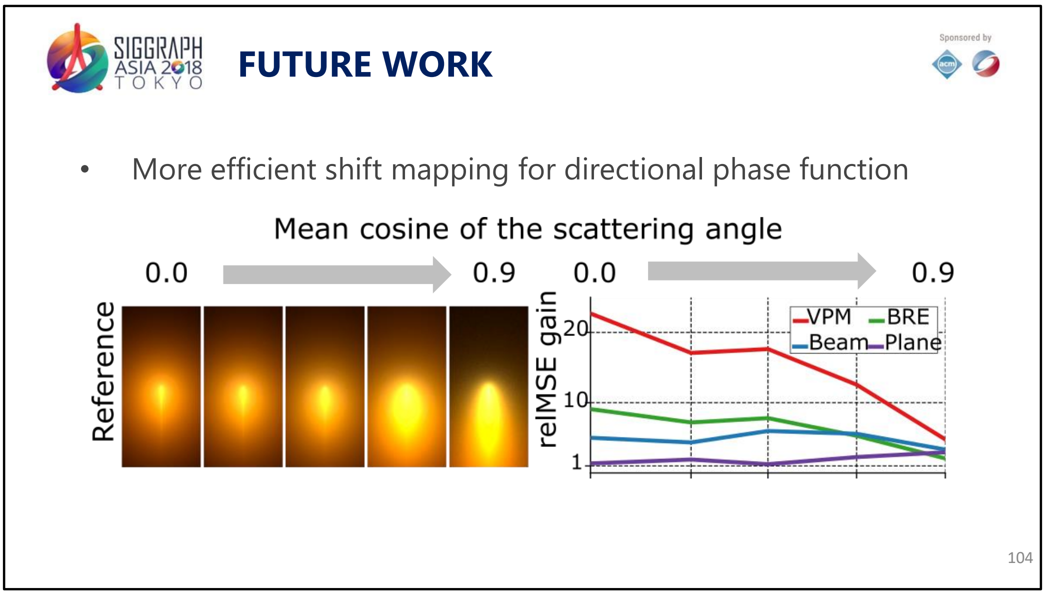


FUTURE WORK

- More efficient shift mapping for directional phase function

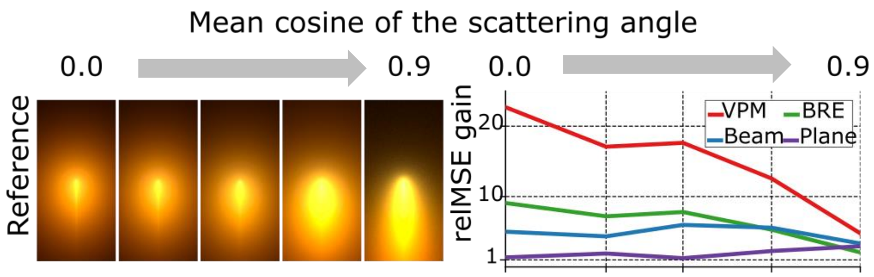
103

For future work, first, we want to have a more efficient shift mapping that handle better participating media with anisotropic phase function. To evaluate the current shift mapping, we have done an experiment on the laser scene where we have several versions of this scene and had increased the phase function directionality.



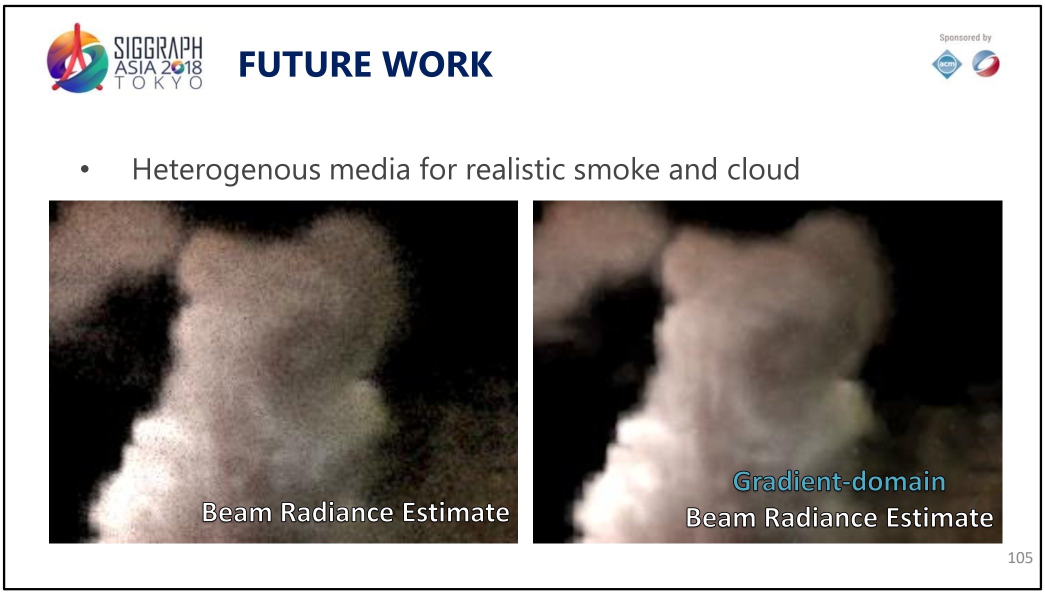
FUTURE WORK

- More efficient shift mapping for directional phase function



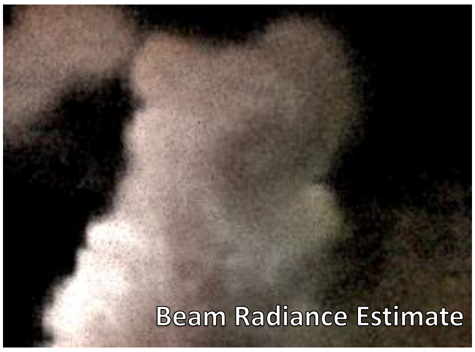
104

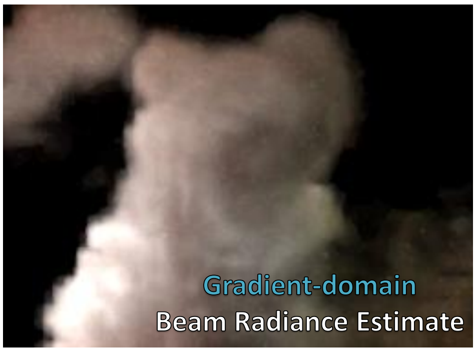
And these are the results of our different gradient-domain rendering techniques where we plot the relative gain in term of error between the primal and the gradient-domain rendering technique. For example, gradient-domain volume photon mapping have more than 20 times less error compared to primal-domain for isotropic phase function. However, having strong directionality in the phase function reduce the performance of gradient-domain. Changing the shift mapping operator by taking advantage of more freedom inside the volume may lead to more efficient shift mapping operator for this particular case.



FUTURE WORK

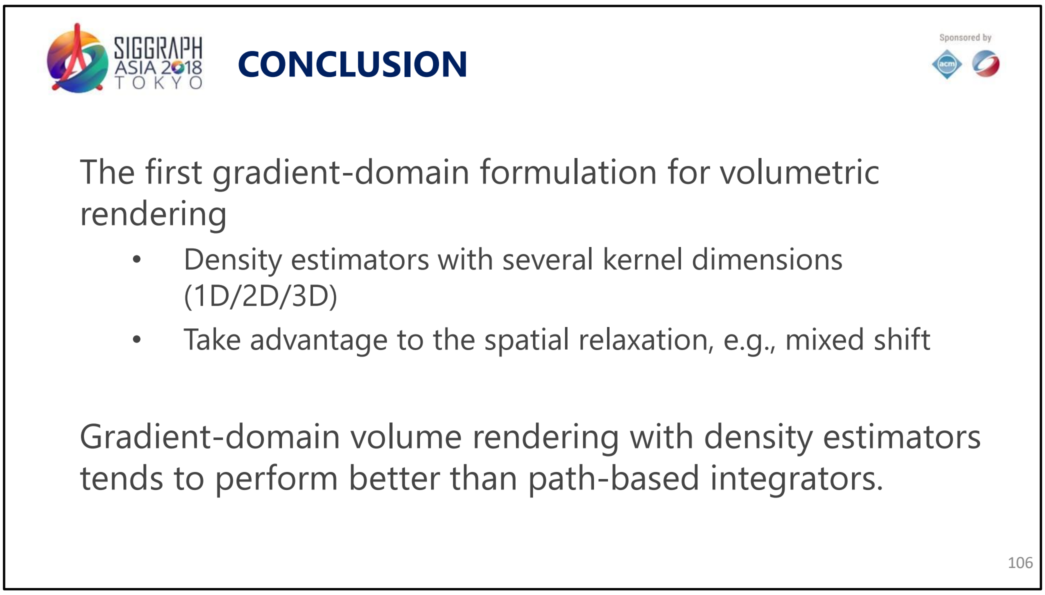
- Heterogenous media for realistic smoke and cloud





105

Another point that we want to explore is how to handle heterogenous participating media properly. You can found here an early result of gradient-domain beam radiance estimate for cloud rendering. In this result, we have used ray marching inside the volume to evaluate the transmittance and sample the distance. This transmittance evaluation is costly and biased. Making possible to uses other methods for evaluating the transmittance like delta tracking and take care of the change of density inside the shift mapping design is an interesting avenue for future work.


CONCLUSION

Sponsored by

The first gradient-domain formulation for volumetric rendering

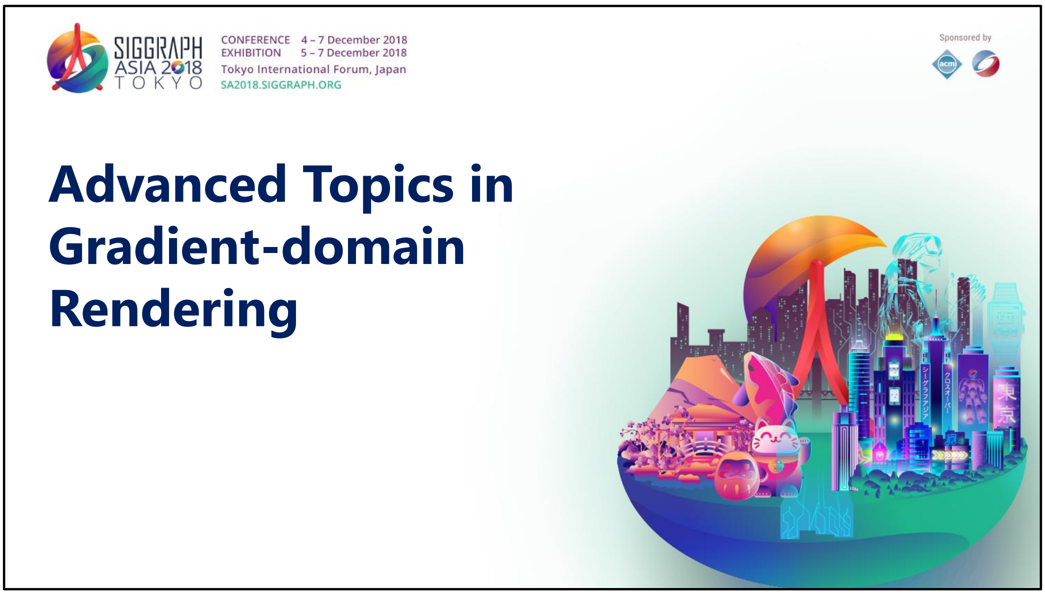
- Density estimators with several kernel dimensions (1D/2D/3D)
- Take advantage to the spatial relaxation, e.g., mixed shift

Gradient-domain volume rendering with density estimators tends to perform better than path-based integrators.

106

To conclude the talk, we presented the first gradient-domain formulation for volumetric rendering. This formulation support multiples density estimators with several kernel dimensions. Moreover, this technique, when it is possible, take advantage the spatial relaxation inside the volume to speedup the computation.

4.6 Advanced topics in gradient-domain rendering



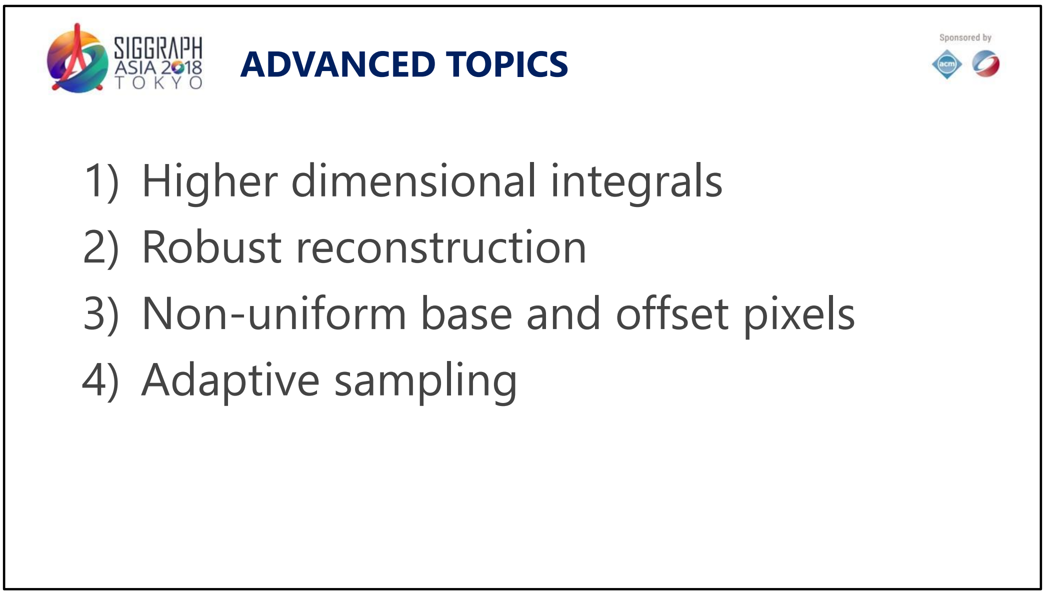
SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

Sponsored by

Advanced Topics in Gradient-domain Rendering

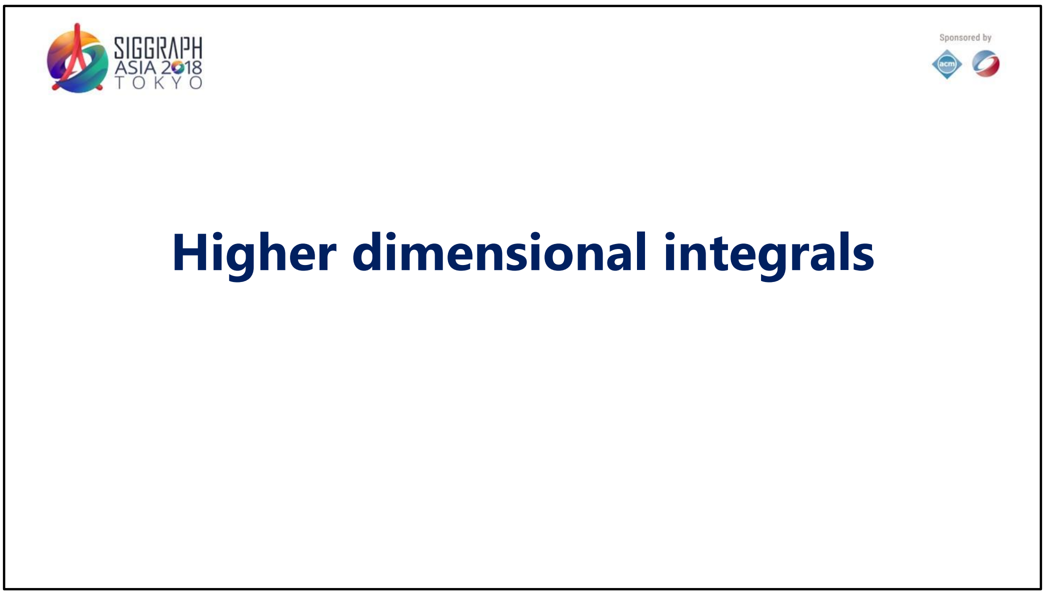
We have now gained enough background to discuss the state of the arts in gradient-domain rendering. We will explain more advanced topics that have not been covered so far. Most of these works are orthogonal and can be built on top of the gradient-domain rendering algorithms that you learn. Let’s start.



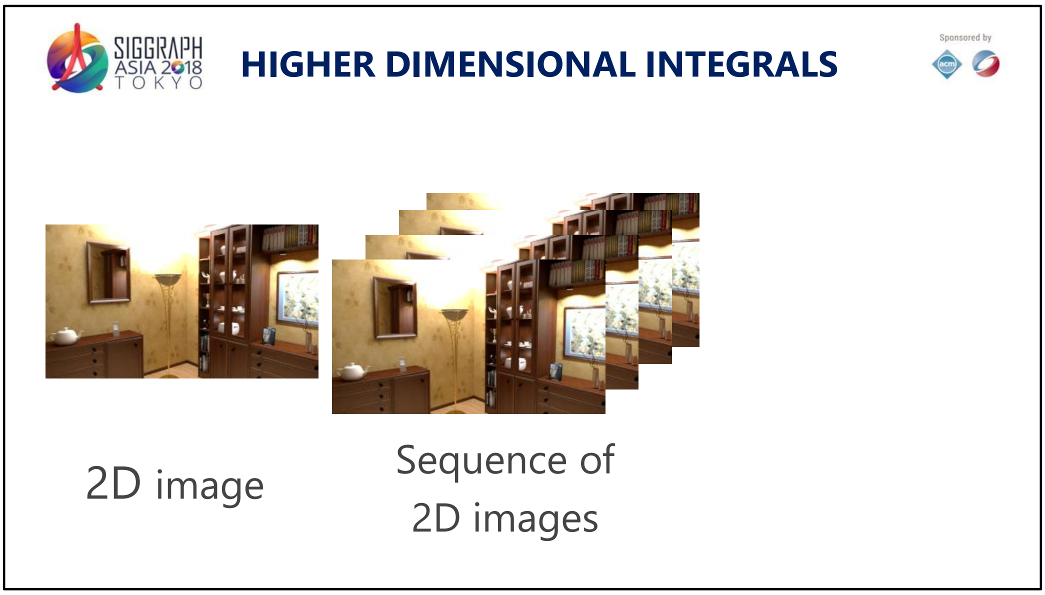
ADVANCED TOPICS

1) Higher dimensional integrals
2) Robust reconstruction
3) Non-uniform base and offset pixels
4) Adaptive sampling

We categorize the topics into 4 sections. First, we will see an extension of the gradient-domain formulation of higher dimensional integration. Then we will see more robust image reconstructions. After that, we will look at different ways to make pixel pairs to compute gradients. Finally, we will talk about adaptive sampling in the gradient domain.

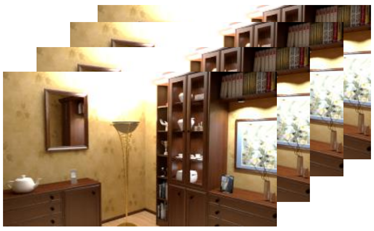


Higher dimensional integrals



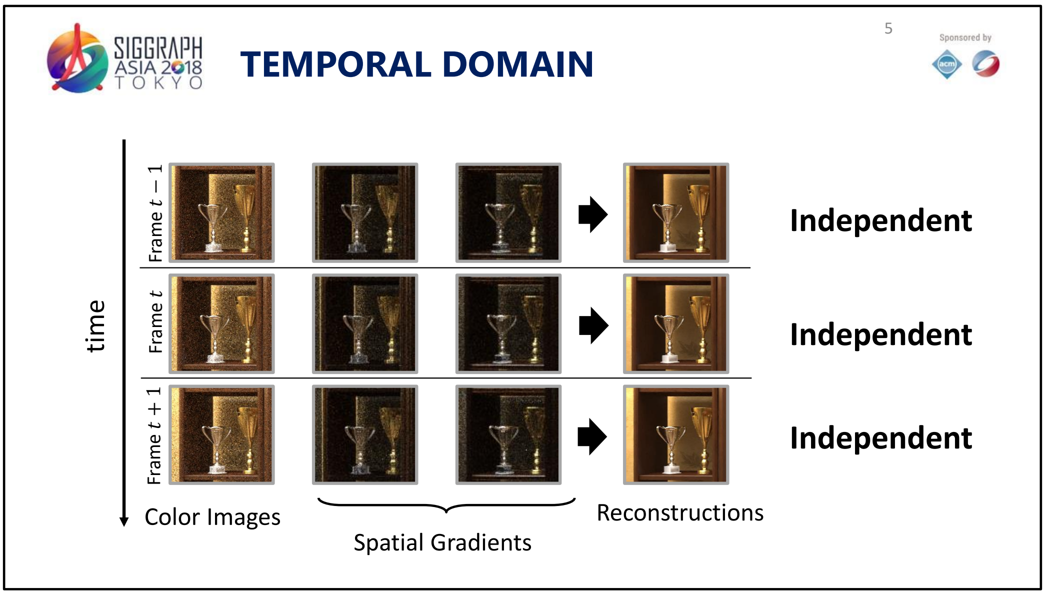
HIGHER DIMENSIONAL INTEGRALS

2D image

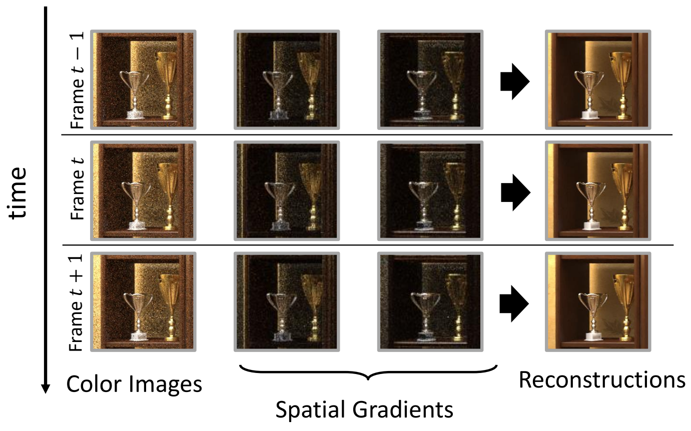


Sequence of 2D images

So, so far we have been interested in rendering independent images. However, usually, especially for movie production, we need to compute a sequence of images. The question is how to extended gradient-domain approaches to support animations and take advantage of the temporal dimension.



TEMPORAL DOMAIN



Independent

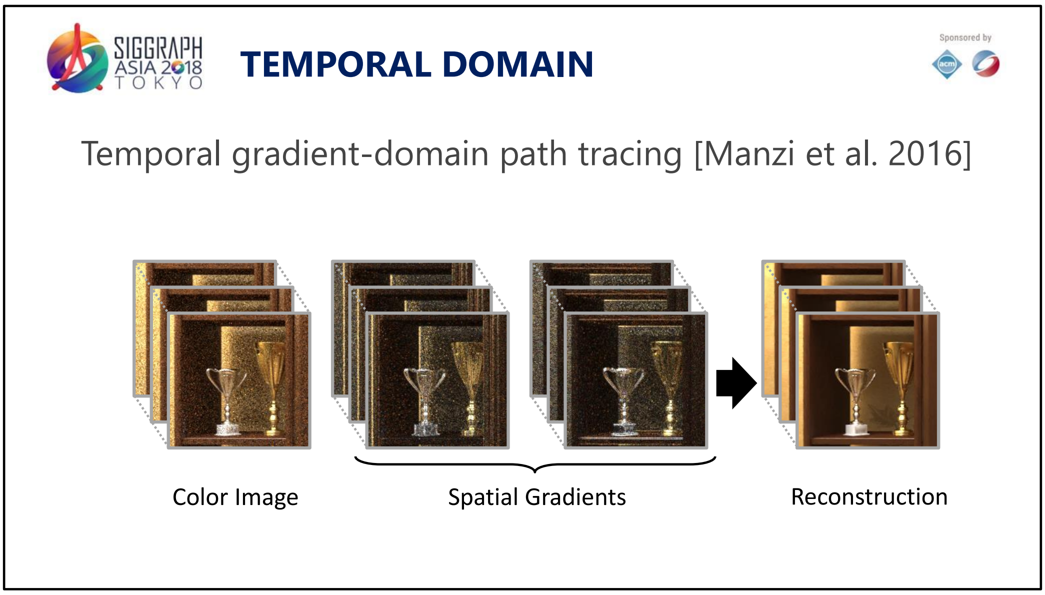
Independent

Independent

A naïve extension to support a sequence of images is to treat each image (or frame) independently. For each frame, we can compute primal and gradient values, and apply a Poisson reconstruction. This simple approach generates ...

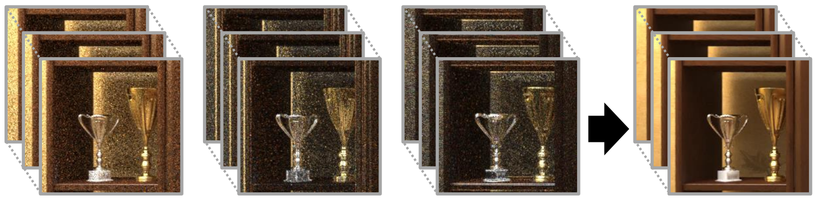
Gradient-domain path tracing
[Kettunen et al. 2015]

… this kind of results. We can see that, individually, the Poisson reconstruction generates quite smooth images for each frame. However, as the reconstruction is independent, low-frequency noise is visible when playing the animation. One way to reduce this artifact is to take into account the temporal domain and do the reconstruction across multiple frames.

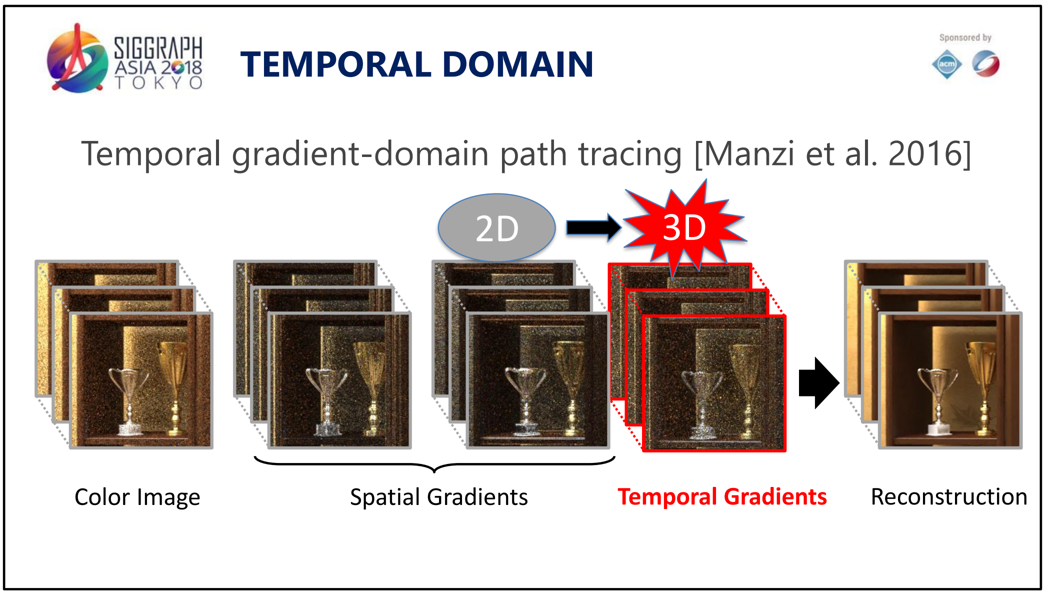


TEMPORAL DOMAIN

Temporal gradient-domain path tracing [Manzi et al. 2016]

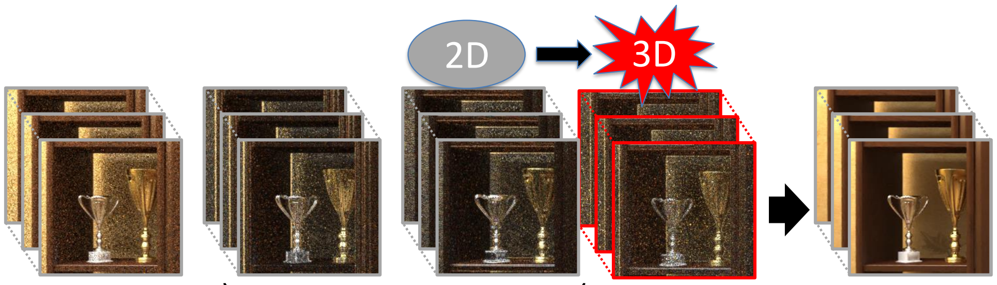


To add temporal domain to gradient-domain rendering, we need to perform the reconstruction on several frames at a time. For memory cost, we only consider a small window of consecutive frames. However, some important information is missing here.

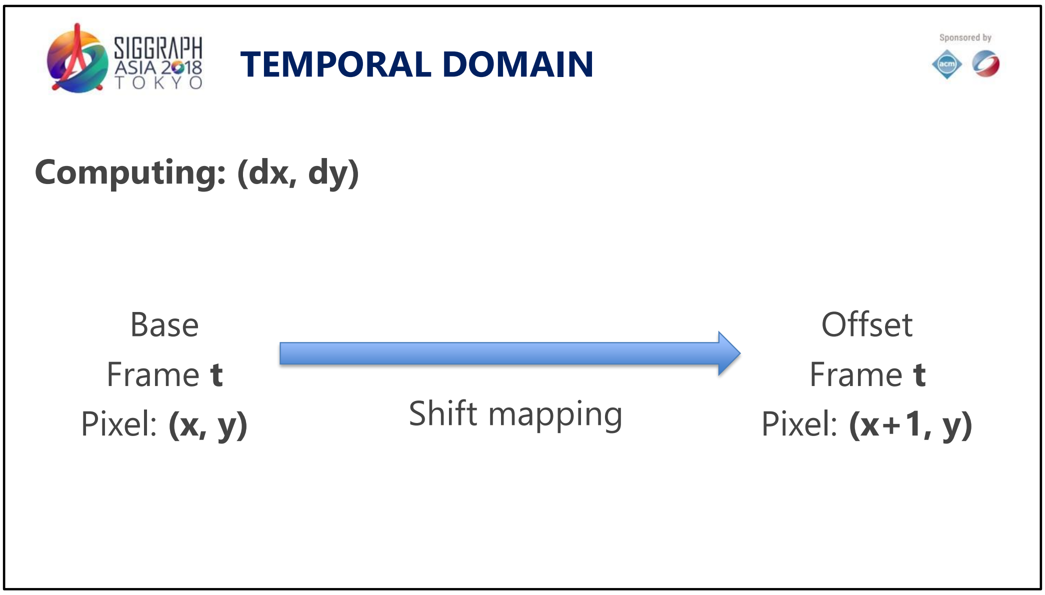


TEMPORAL DOMAIN

Temporal gradient-domain path tracing [Manzi et al. 2016]



Indeed, we need to know the temporal gradients between frames. Moreover, as we perform the reconstruction on different images at the same time, we need to do a 3D Poisson reconstruction. We will not detail the reconstruction part and only show how to compute these temporal gradients.



TEMPORAL DOMAIN

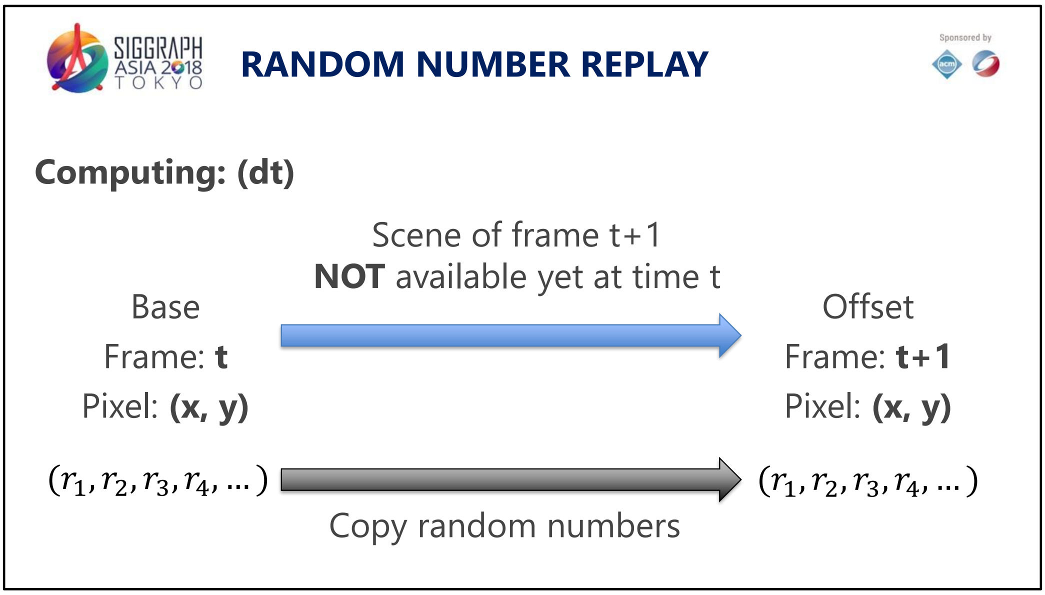
Computing: (dx, dy)

Base
Frame **t**
Pixel: **(x, y)**

Shift mapping

Offset
Frame **t**
Pixel: **(x+1, y)**

Remember, when we were computing image-space gradient, we have a base path at the frame T at the position x and y, and we want to generate an offset path inside the same frame but at the location x+1 and y. To do so, we apply shift mapping as what we have learnt so far.

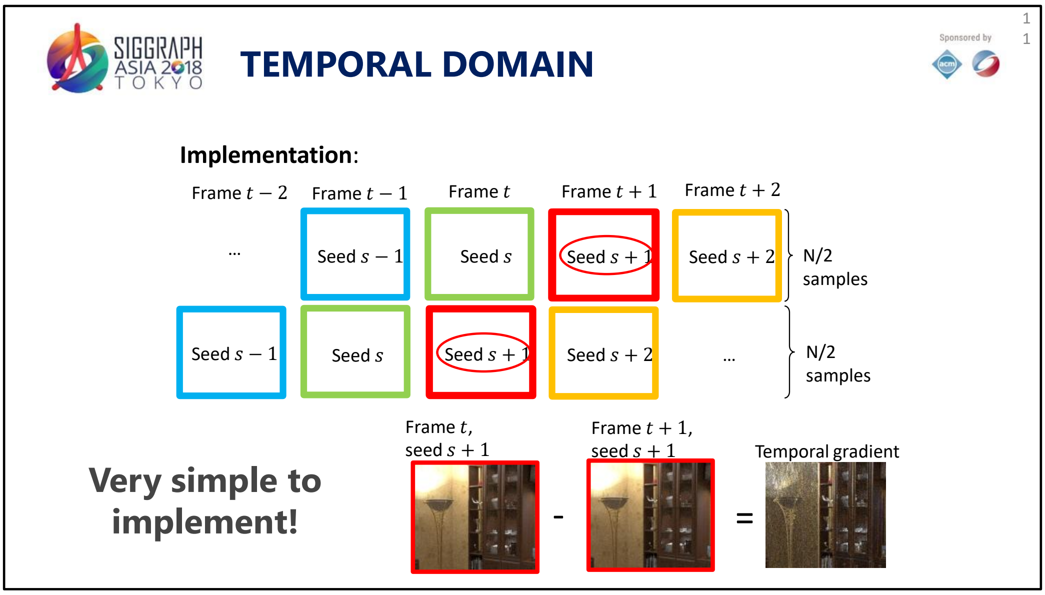


RANDOM NUMBER REPLAY

Computing: (dt)

However, when we are computing the gradient in the temporal domain, we want to shift a base path at frame t and generates an offset path at the frame t+1. The problem is that to create this offset path we need to store two scenes in the memory. Moreover, due to the object movement, it is difficult to track spatial information from frame t to frame t+1.

Our solution here is to only store one scene in the memory and rely on a simpler shift mapping. The idea is to capture the random numbers that generate the base paths in frame t, and when the scene for frame t+1 is loaded, we generate the offset paths by replaying such random numbers. This brings a certain level of path coherence, while still keeping memory usage efficient.

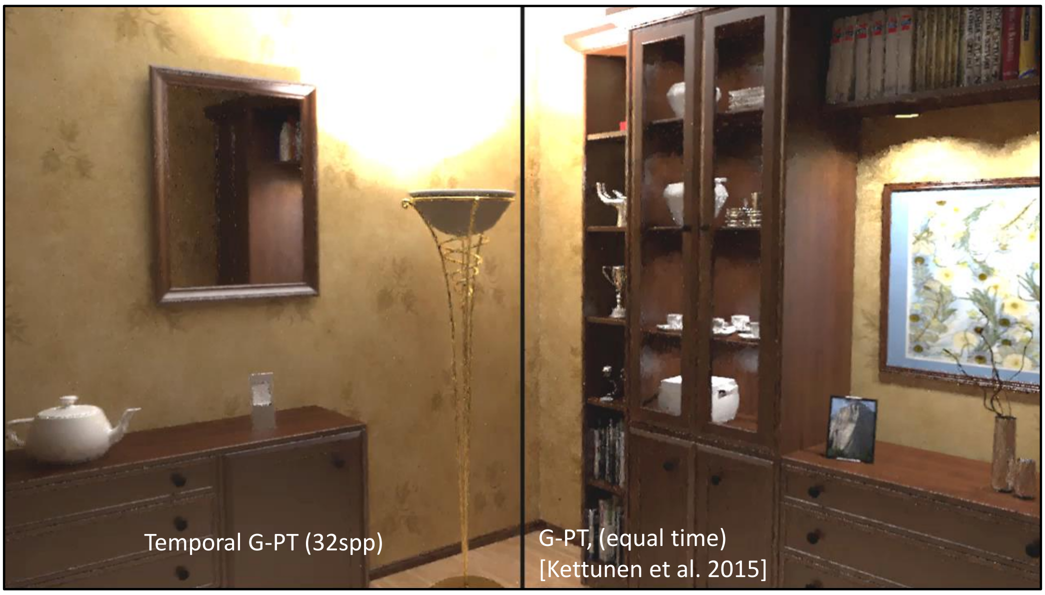


TEMPORAL DOMAIN

Implementation:

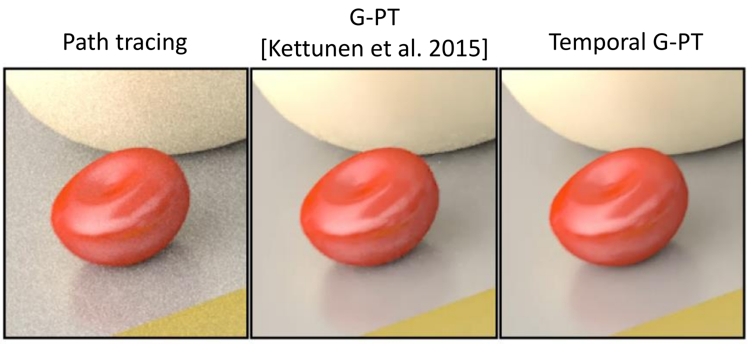
Very simple to implement!

But how to implement this random number replay? In fact it is pretty easy! For each frame we compute two images at half the sampling count with different seeds. In the next frame we again compute two images but make sure that one of them shares the seeds of one of the images in the previous frame. We continue this process for all the frames. Two images of adjacent frame with same random numbers will have strong correlations. Moreover, as no transformation is involved, we can directly take the difference between these two images to produce the temporal gradient.



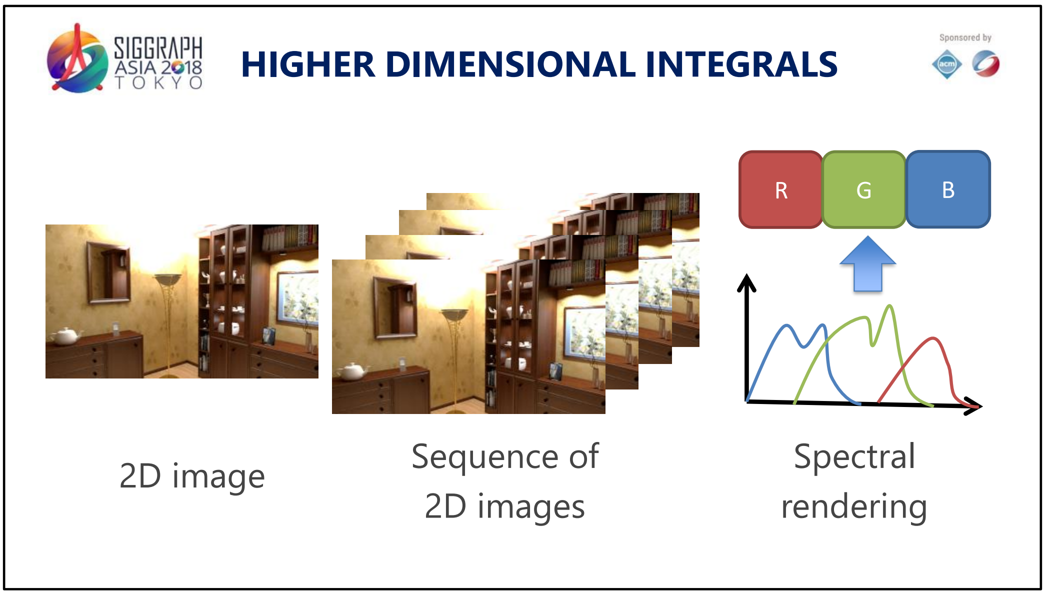
Time to look at some results. Here we can see the comparison between temporal G-PT and G-PT with independent reconstruction. We can see that leverage the information of the temporal gradients improve the results in term of reconstruction stability.

TEMPORAL DOMAIN



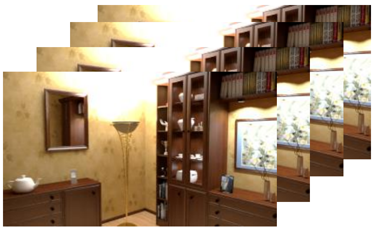
All at equal computation time!

This is another scene where we also show path tracing results. All these results are equal time. Again, temporal G-PT performs the best.

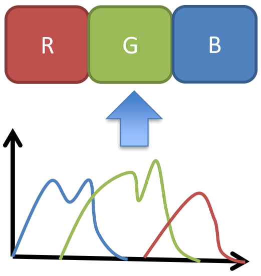


HIGHER DIMENSIONAL INTEGRALS

2D image

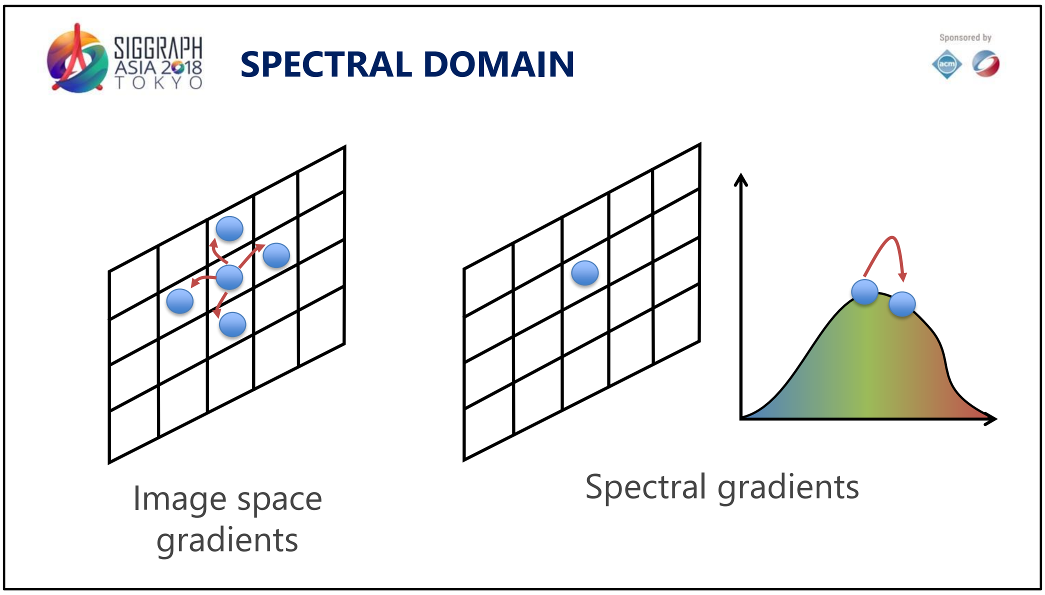


Sequence of
2D images

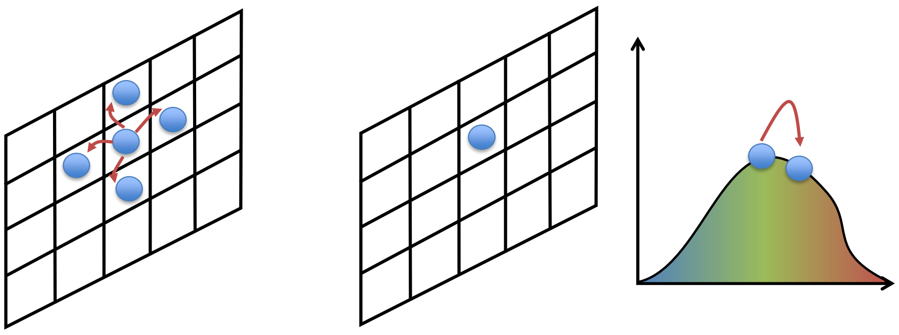


Spectral
rendering

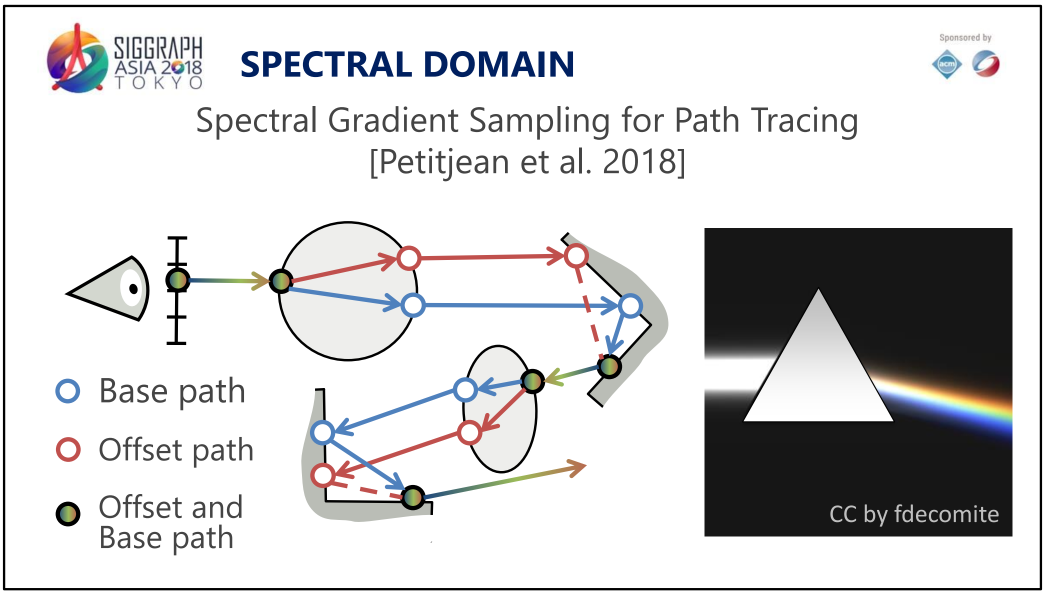
So we just saw how to extend gradient-domain to a sequence of images by taking into consideration the temporal domain. However, these images are still using RGB as their internal image representation. What about spectral rendering? Indeed, in spectral rendering, we need to estimate the distribution of energy across wavelengths for a given pixel. As you can guess, we can use a gradient-domain idea to better estimate this distribution compared to conventional rendering techniques.



SPECTRAL DOMAIN

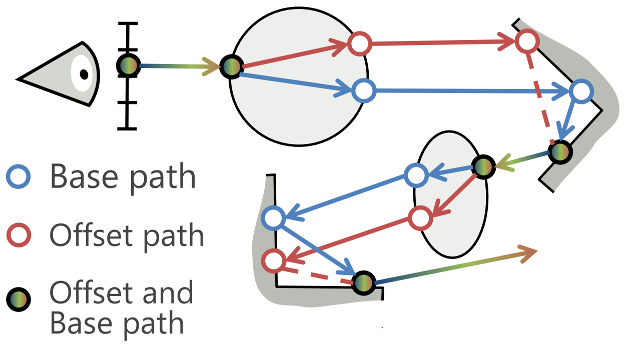


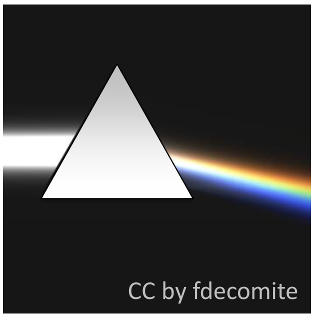
In image-space gradients, we were computing the difference of the base path generated at a given pixel location to their neighbors. The idea in spectral-domain rendering is, within the same pixel, compute the gradient inside the wavelength space. For that, we need to define a new shift mapping to generate the gradients in the wavelength dimension.



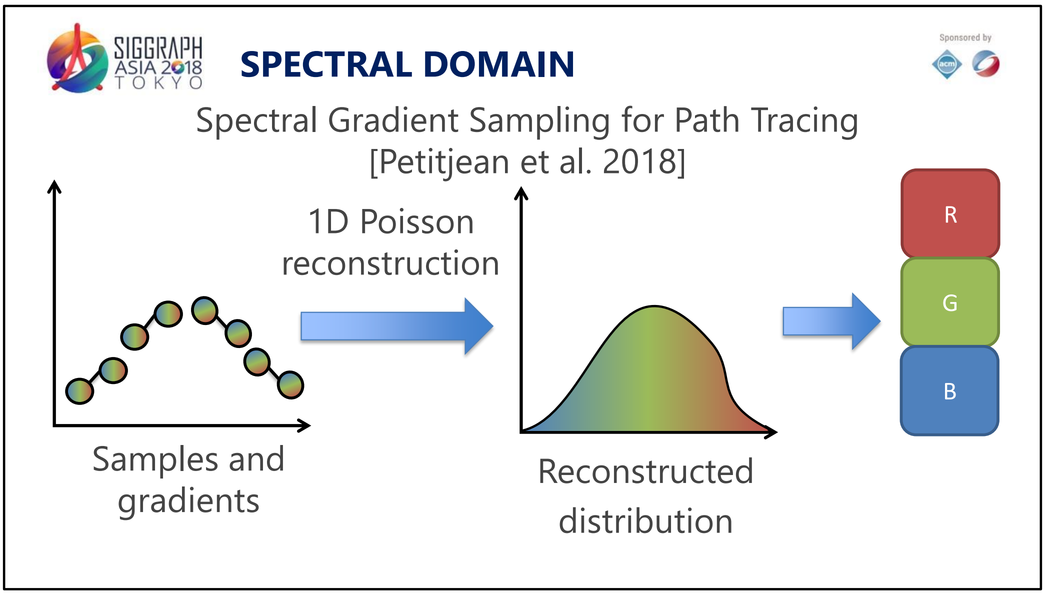
SPECTRAL DOMAIN

Spectral Gradient Sampling for Path Tracing
[Petitjean et al. 2018]



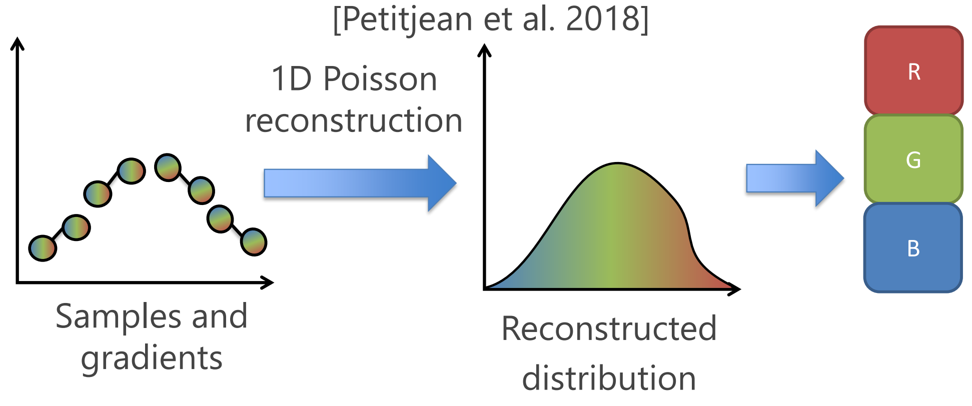


The shift mapping proposed by Petitjean and colleague is similar to that in gradient-domain path tracing. Indeed, it relies on half vector copy and diffuse reconnection. However, the difference is the condition of performing the shift mapping. Indeed, as it is within a pixel, the offset and the base path start inside the same pixel. The divergence between these two paths only occurs when we encounter a dispersive object (for example a glass object). When the paths are diverging, half-vector copy is used to keep them coherent. The offset path is merged back to the base path when diffuse reconnection is applicable. Divergence can occur multiple times on the path when several diffractive objects is encounter during tracing.

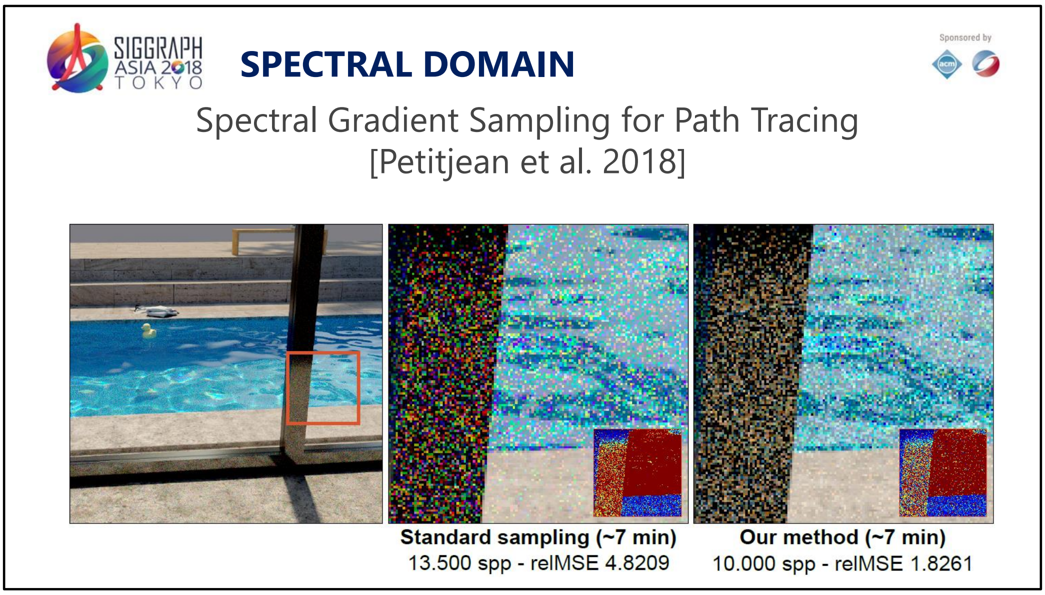


SPECTRAL DOMAIN

Spectral Gradient Sampling for Path Tracing
[Petitjean et al. 2018]

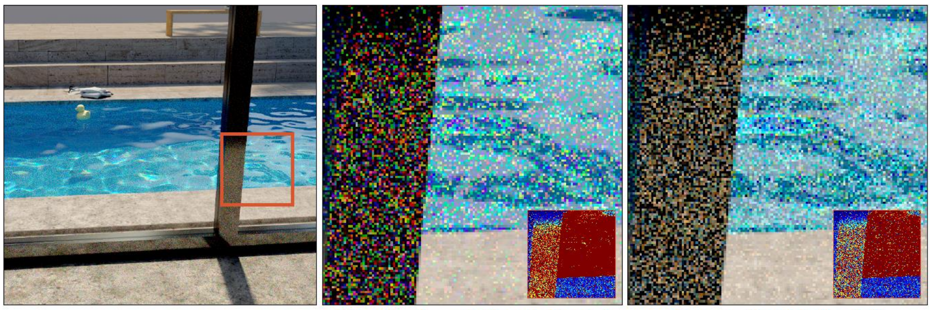


After getting the samples for a different wavelength and their gradient, a 1D Poisson reconstruction is performed to reconstruct the wavelength distribution. Then this reconstructed distribution can be converted back to RGB for viewing on a conventional display.



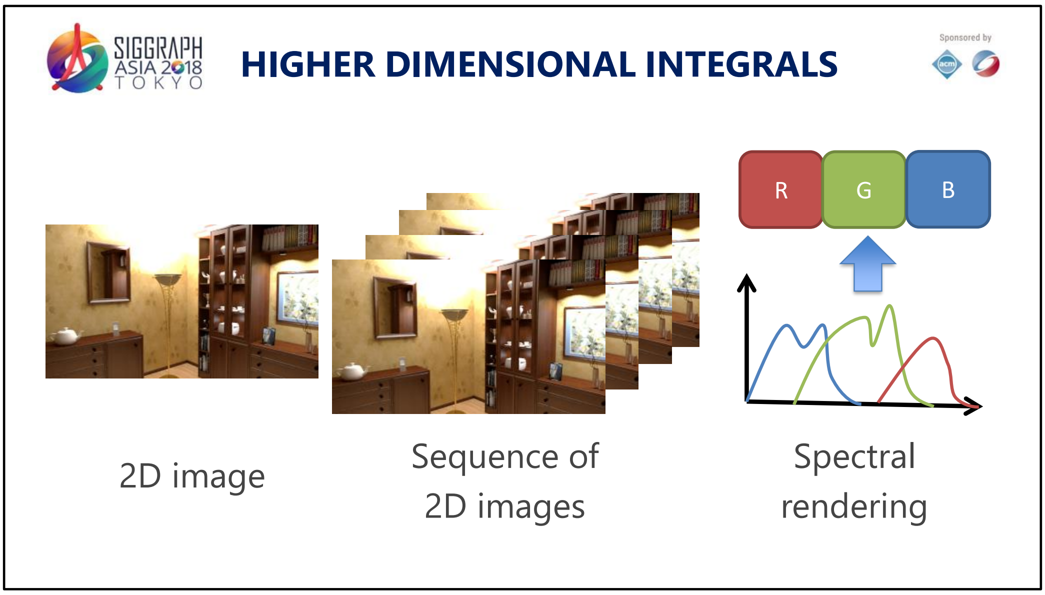
SPECTRAL DOMAIN

Spectral Gradient Sampling for Path Tracing
[Petitjean et al. 2018]



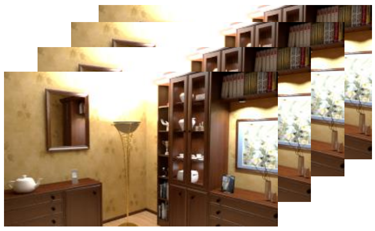
Standard sampling (~7 min)	Our method (~7 min)
13.500 spp - relMSE 4.8209	10.000 spp - relMSE 1.8261

Let’s look at some results. Note, here; we are only interested in the color noise and not the noise across pixels as we only perform a 1D reconstruction to smooth out noise in the wavelength distribution. Compared to the non-gradient technique, their method can achieve variance reduction thanks to their specialized shift mapping and 1D Poisson reconstruction. Note that if we combine spatial gradient and wavelength gradient using a 3D Poisson reconstruction, we should be able to clean out the noise across pixels. This is left as a future work.

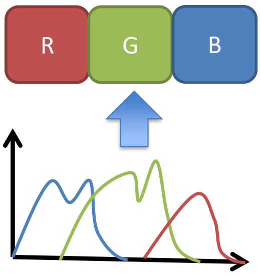


HIGHER DIMENSIONAL INTEGRALS

2D image

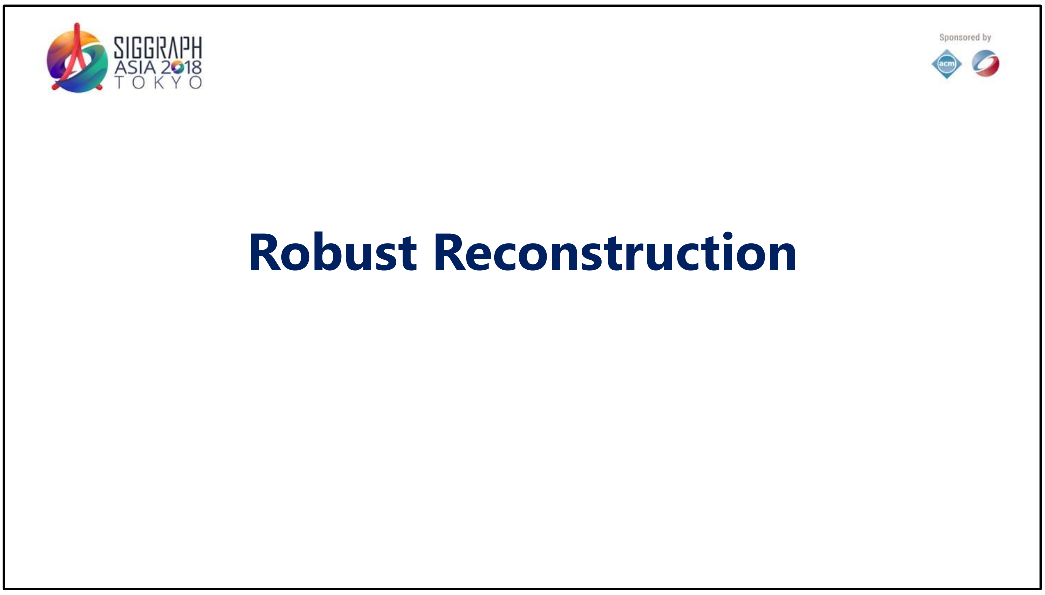


Sequence of 2D images



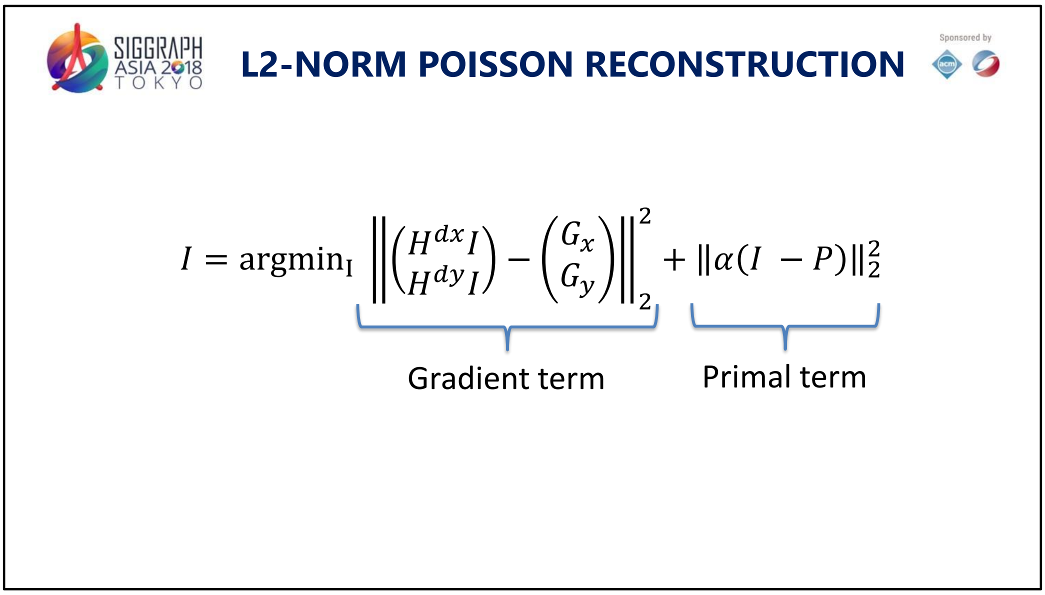
Spectral rendering

This concludes the first part about extending of gradient-domain rendering to higher dimensional integrals. We can see an emerging pattern here: a new domain, new shift mapping, new Poisson reconstruction.



Robust Reconstruction

Let’s now explore more robust reconstruction. The traditional Poisson reconstruction works well for gradient-domain rendering, but its implementation requires solving a linear equation (L2-norm reconstruction) or an iterative reweighted least square (L1-norm reconstruction). We will see that simpler reconstruction exists.

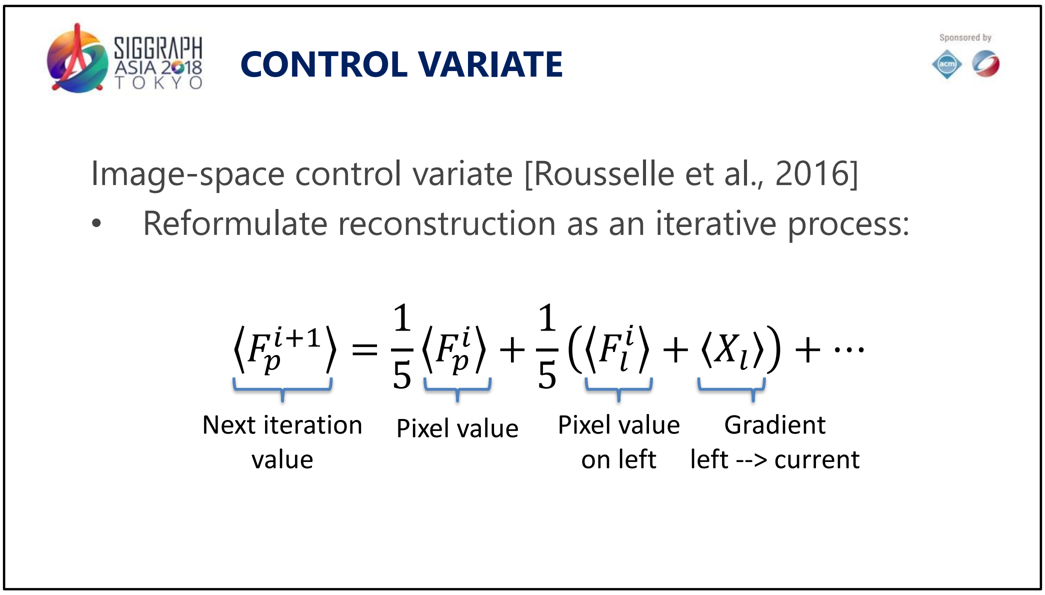


L2-NORM POISSON RECONSTRUCTION

$$I = \mathrm{argmin}_{\mathrm{I}} \underbrace{\left\| \begin{pmatrix} H^{dx} I \\ H^{dy} I \end{pmatrix} - \begin{pmatrix} G_x \\ G_y \end{pmatrix} \right\|_2^2}_{\text{Gradient term}} + \underbrace{\| \alpha (I - P) \|_2^2}_{\text{Primal term}}$$

Here is Poisson reconstruction . It takes the primal image P and the image-space gradients (Gx, Gy) to reconstruct a final image I. The L2 formula opts to minimize the difference of a gradient-based term and a primal-based term. Here H is the finite difference operator. This last term is also called a regularization term and the alpha parameter controls how strong the output should favor similarity to the primal or gradients.

This problem is a least squared minimization problem and iterative solvers like conjugate gradients can be applied directly.



CONTROL VARIATE

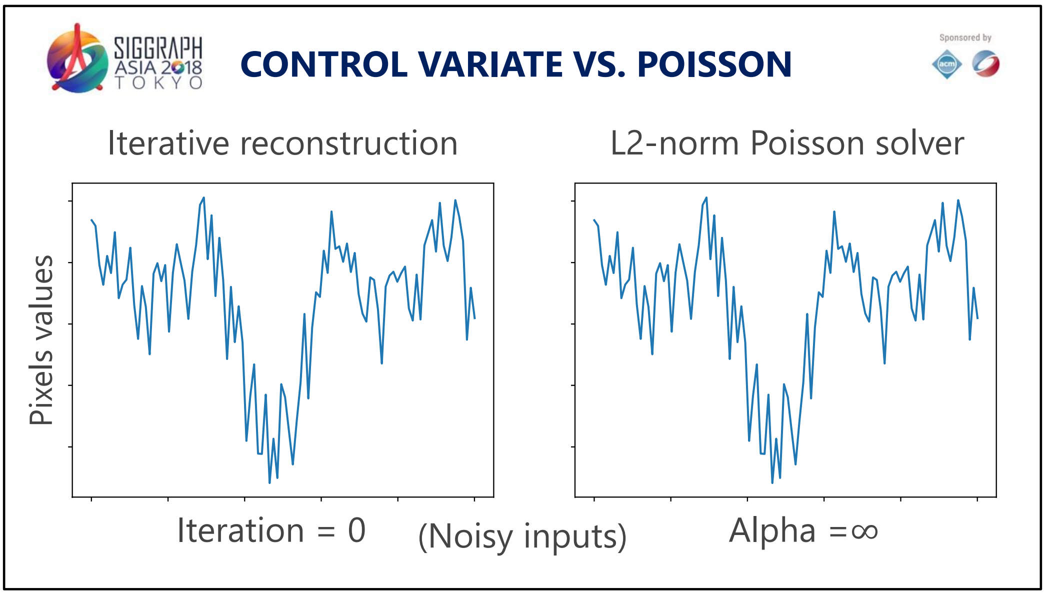
Image-space control variate [Rousselle et al., 2016]

- Reformulate reconstruction as an iterative process:

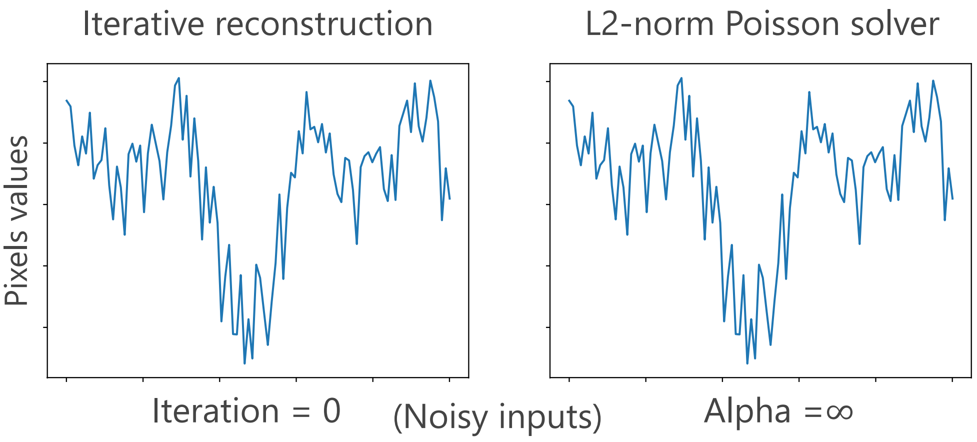
$$\underbrace{\langle F_p^{i+1}\rangle}_{\text{Next iteration value}} = \frac{1}{5}\underbrace{\langle F_p^{i}\rangle}_{\text{Pixel value}} + \frac{1}{5}\left(\underbrace{\langle F_l^{i}\rangle}_{\text{Pixel value on left}} + \underbrace{\langle X_l\rangle}_{\text{Gradient left --> current}}\right) + \cdots$$

A simpler approach has been proposed recently by Rousselle and colleague. They reformulate the reconstruction problem as an iterative reconstruction.
The key idea is that a pixel value could be well approximated by its neighbor pixel values compensated by the gradients.

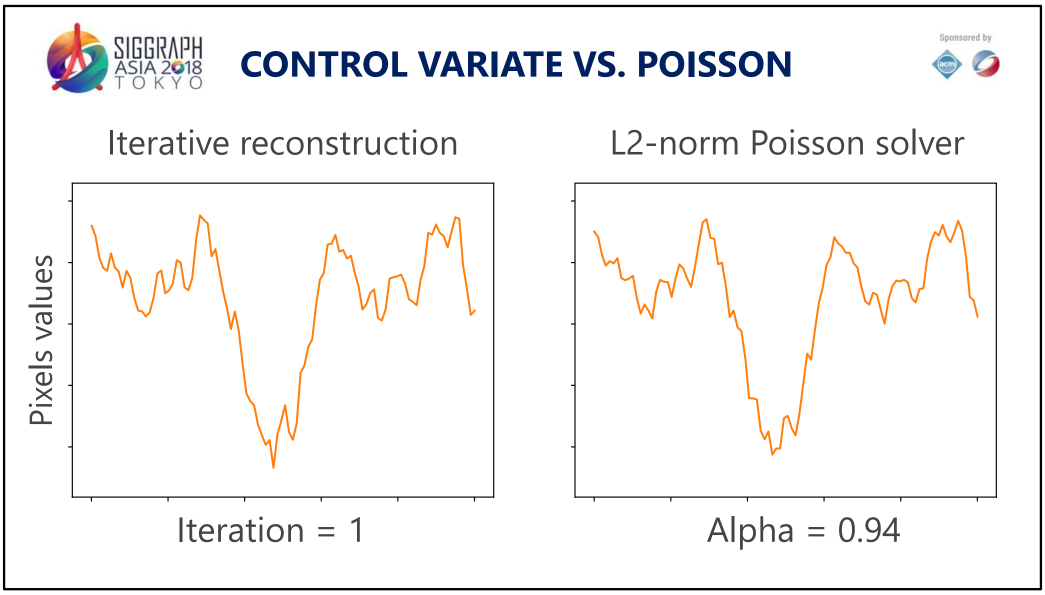
At i-th iteration, the value of pixel p (F_p) is an average of itself and its neighbors (F_l) compensated by the gradients between p and l (X_l). Averaging all these values using uniform weights gives the pixel value for the next iteration. The weight is 1/5 in this case due to 4 neighbors and 1 center pixel.



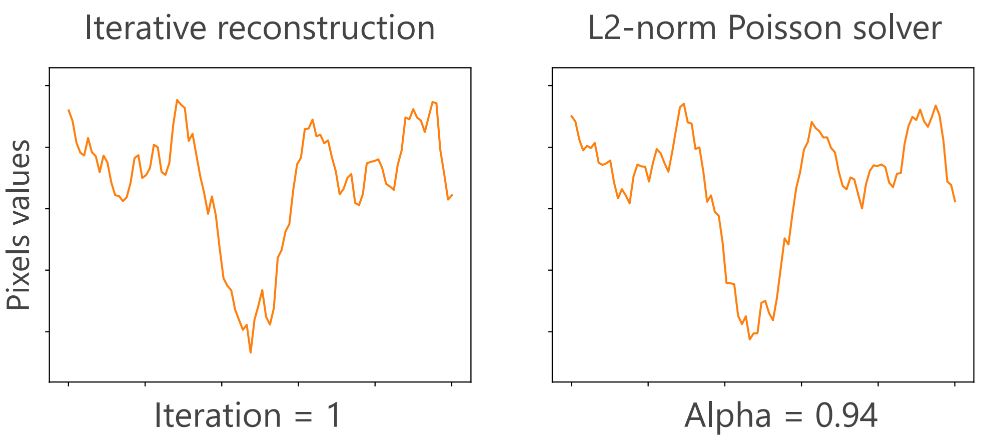
CONTROL VARIATE VS. POISSON



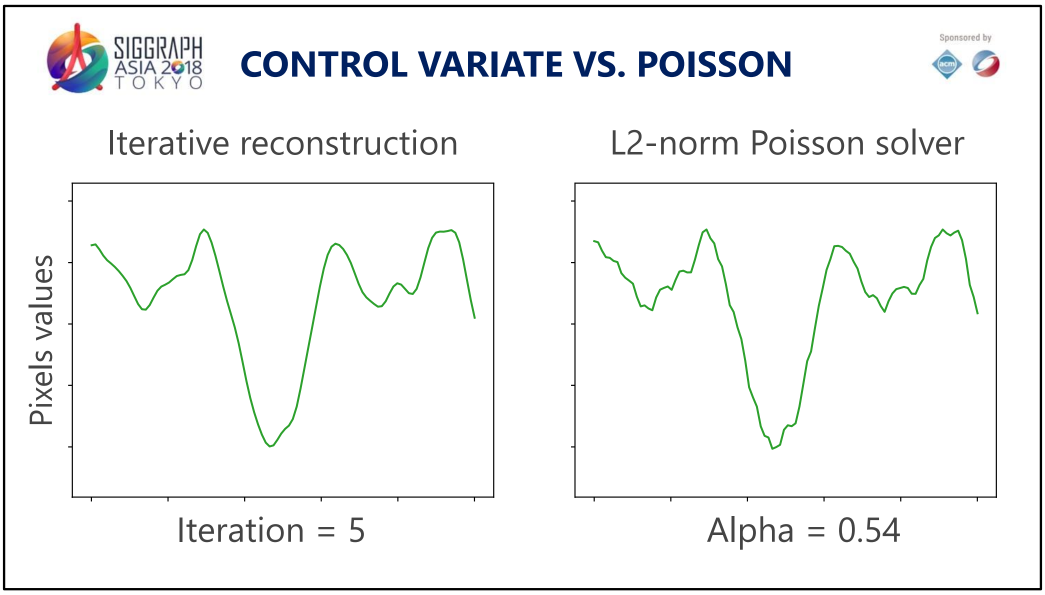
One question is about the number of iterations that we should use. In fact, the number of iteration is related to the alpha parameter used inside a L2 Poisson solver.



CONTROL VARIATE VS. POISSON

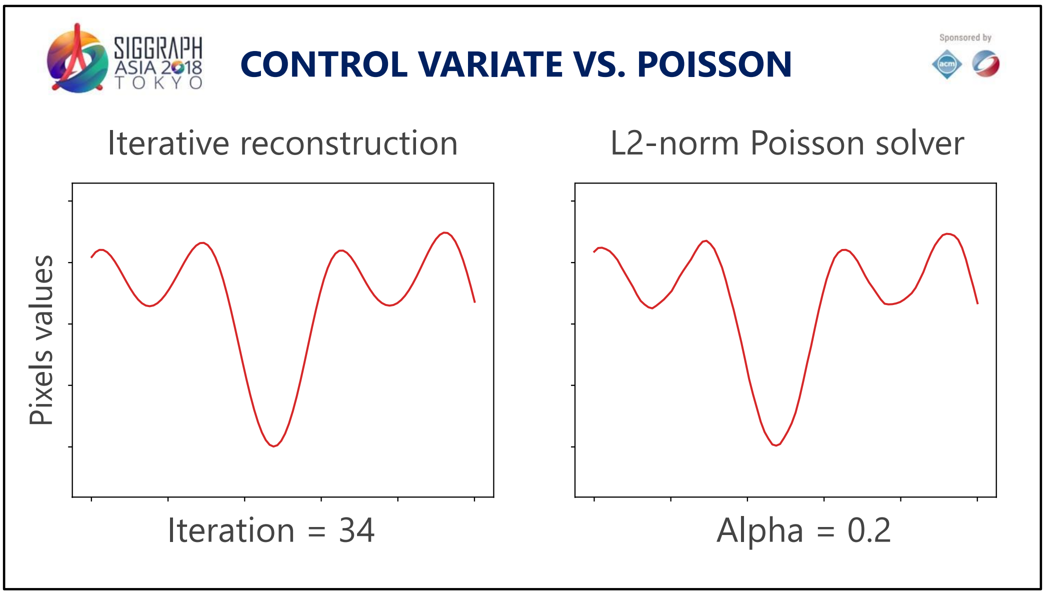


More iteration we do, lower the alpha will be.

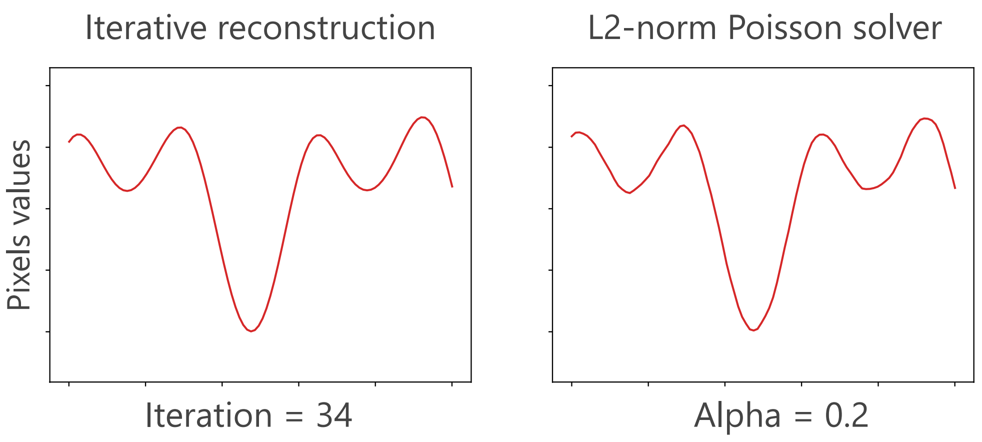


CONTROL VARIATE VS. POISSON

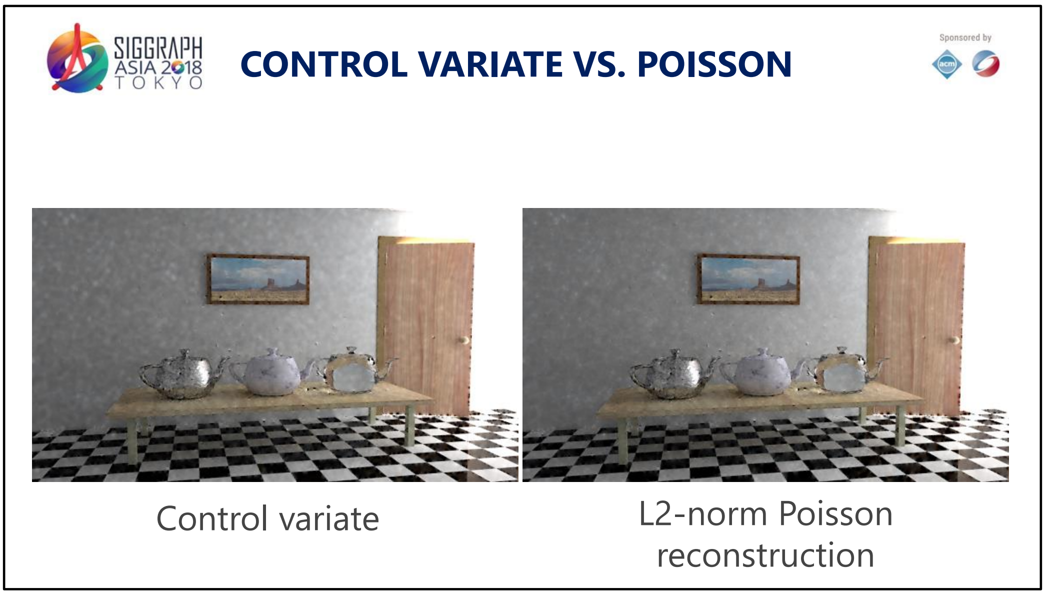
You can observe that the two solver converge to the same solution when they relies more on the gradient informations.



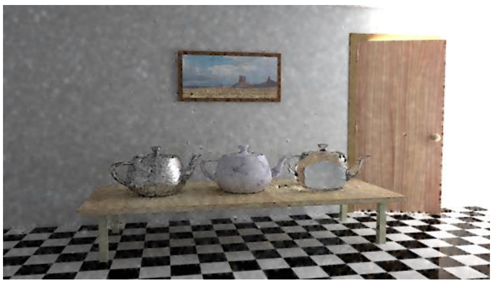
CONTROL VARIATE VS. POISSON



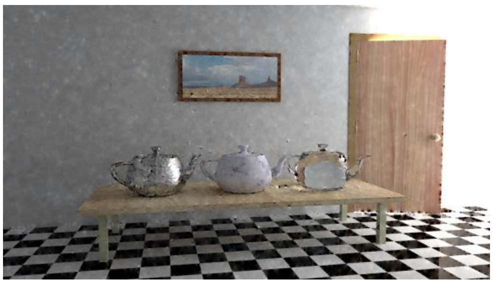
Note that this current iterative reconstruction is not a screened Poisson reconstruction. Please refer the paper to more discussion about this particular point. Despite that, this reconstruction have similar properties to a Poisson reconstruction.



CONTROL VARIATE VS. POISSON

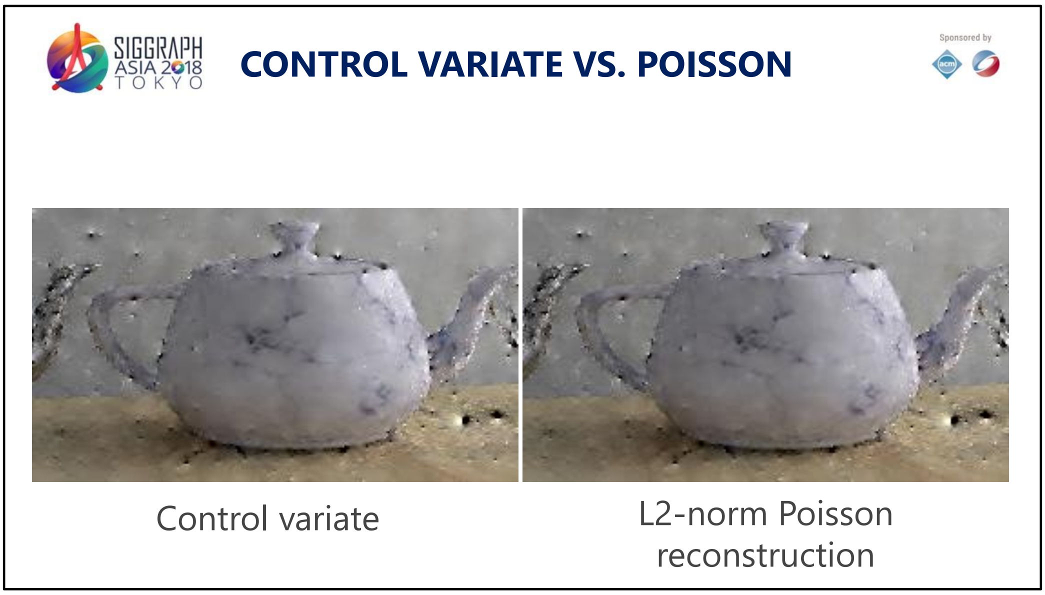


Control variate

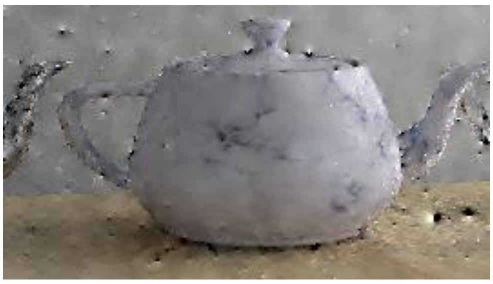


L2-norm Poisson reconstruction

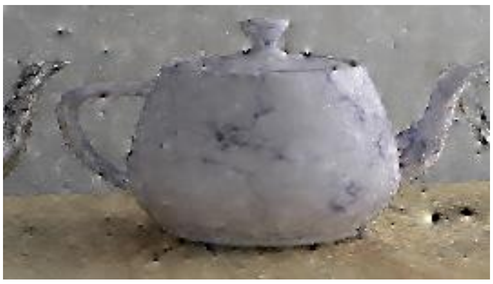
Let us look at this comparison. Uniform control variate and L2 reconstruction generates very similar results.



CONTROL VARIATE VS. POISSON

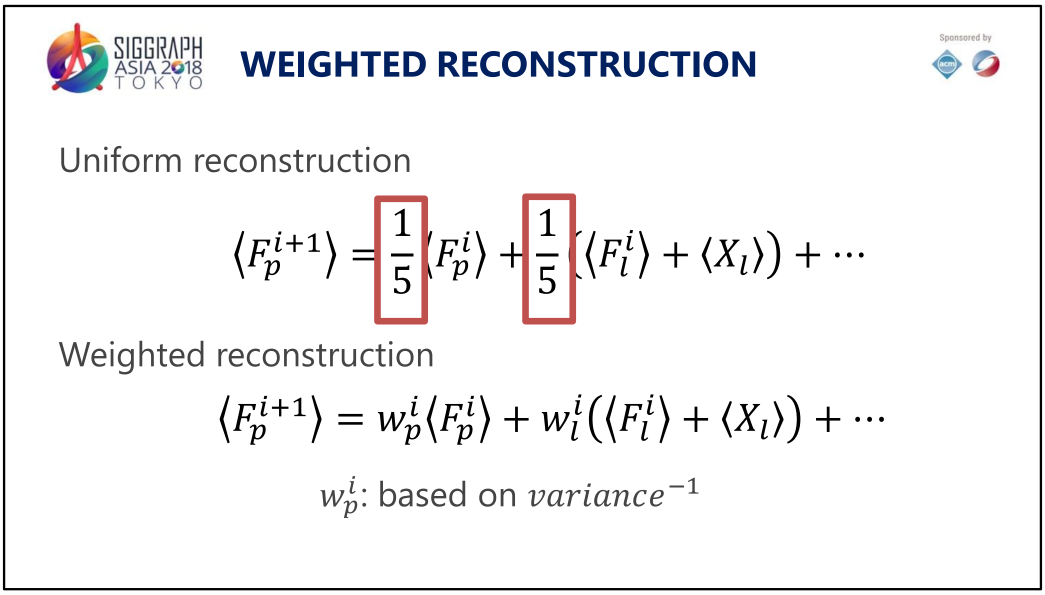


Control variate



L2-norm Poisson reconstruction

Here are the insets.



WEIGHTED RECONSTRUCTION

Uniform reconstruction

$$\langle F_p^{i+1}\rangle = \boxed{\frac{1}{5}}\langle F_p^{i}\rangle + \boxed{\frac{1}{5}}\left(\langle F_l^{i}\rangle + \langle X_l\rangle\right) + \cdots$$

Weighted reconstruction

$$\langle F_p^{i+1}\rangle = w_p^i\langle F_p^{i}\rangle + w_l^i\left(\langle F_l^{i}\rangle + \langle X_l\rangle\right) + \cdots$$

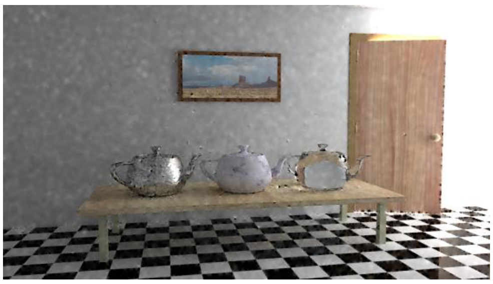
w_p^i: based on $variance^{-1}$

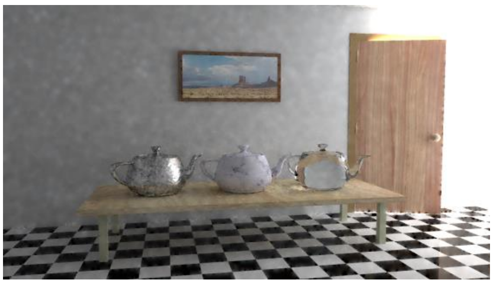
It is possible to optimize the weights for control variates. Usually, the artefacts in the L2-norm Poisson reconstruction is due to high variance primal or gradient values. Using a weight that inversely proportional to the variance can weight down problematic gradients and reduce the reconstruction artefacts.

This weighted reconstruction produces images similar (or better) than L1-norm reconstruction.

WEIGHTED RECONSTRUCTION

w_p^i: based on variance. Reduce at each iteration.


Uniform

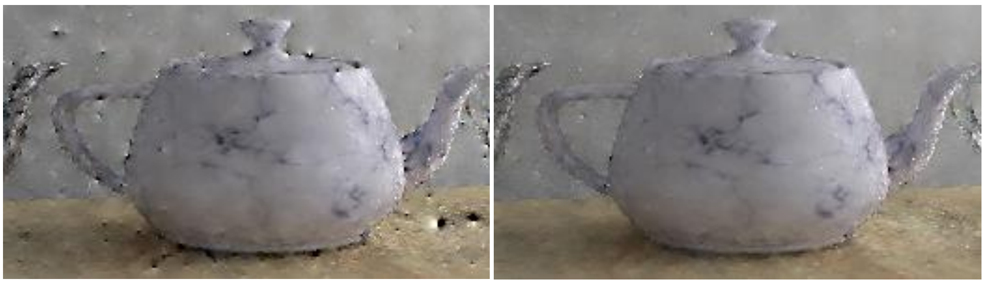

Weighted

Here is the comparison between uniform and weighted reconstruction where the weights are inversely proportional to the estimated variance of the primal pixels. We used an online variance estimation.

For the exact formula of the weights, please see the paper [Rousselle et al., 2016].

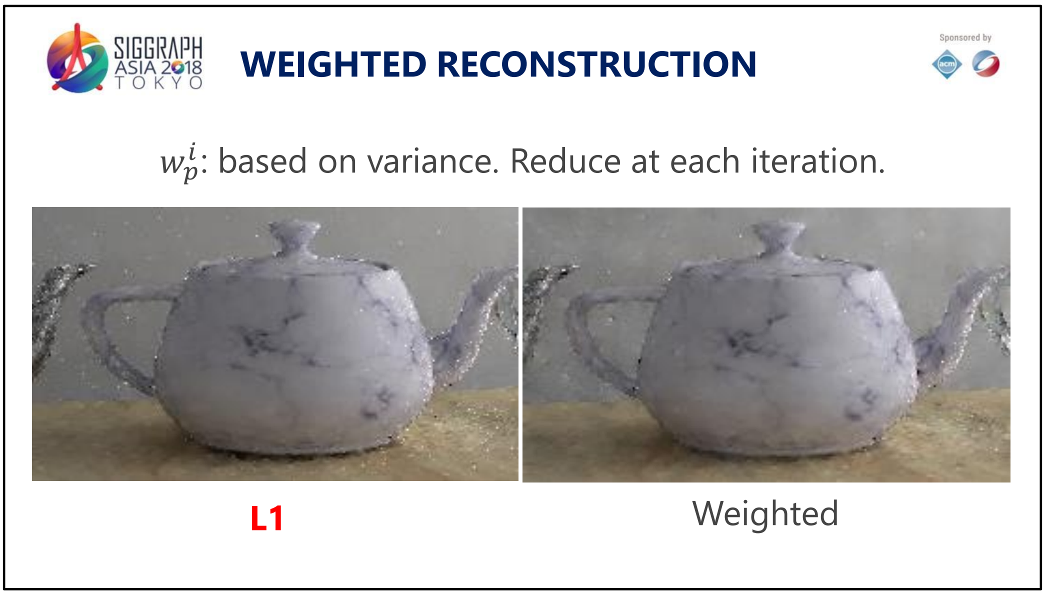
WEIGHTED RECONSTRUCTION

w_p^i: based on variance. Reduce at each iteration.



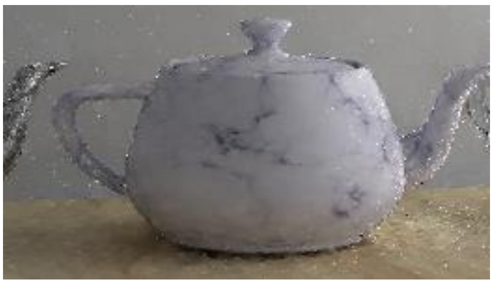
Uniform Weighted

We see that weighting down problematic gradients removes the dipole artefacts and leads to better image quality.

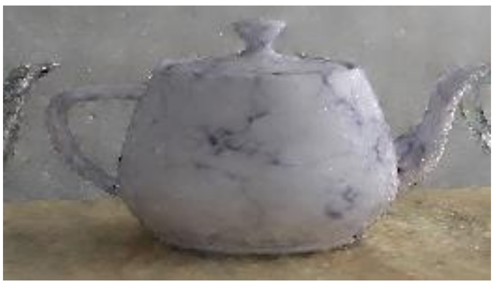


WEIGHTED RECONSTRUCTION

w_p^i: based on variance. Reduce at each iteration.

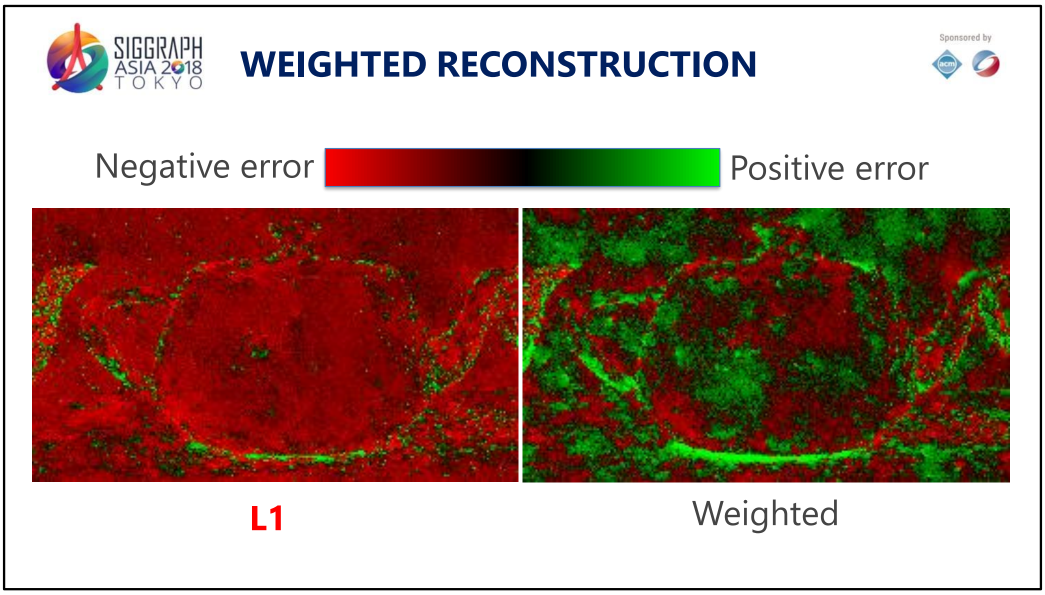


L1



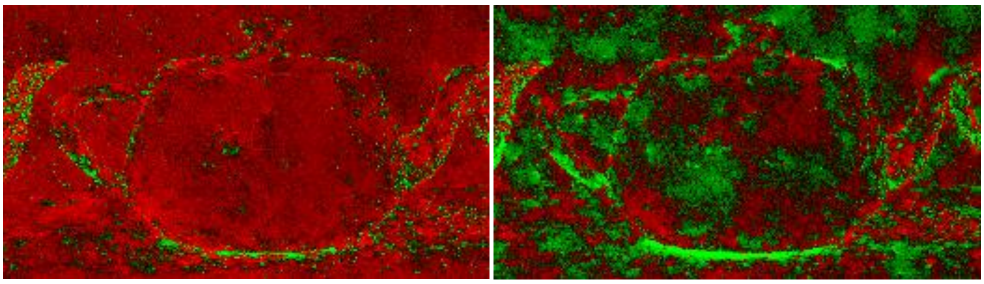
Weighted

Compared to an L1 reconstruction, a weighted reconstruction seems a bit noiser in this case. However, if we visual the errors…



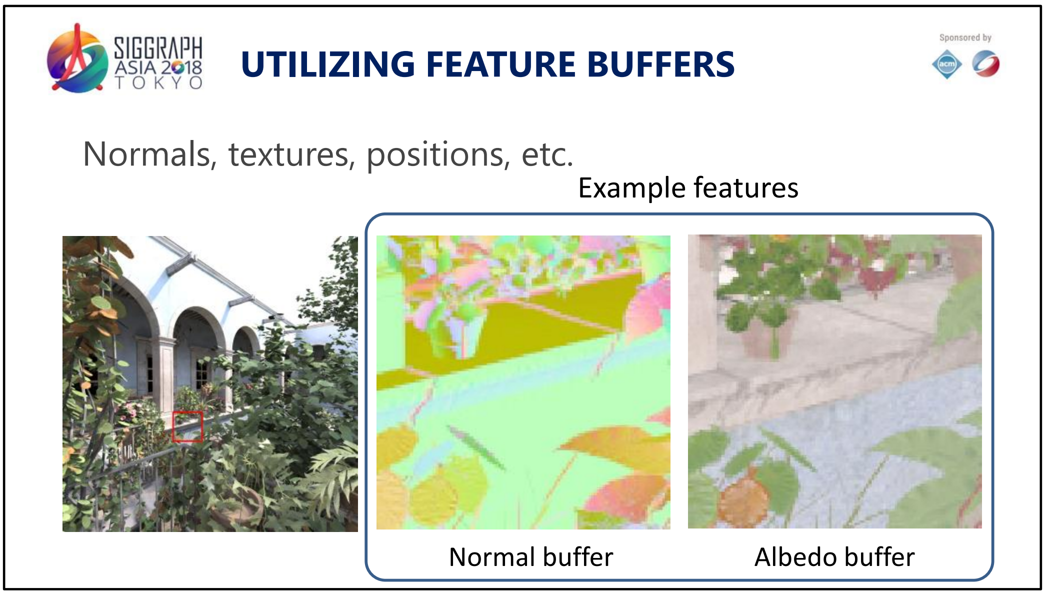
WEIGHTED RECONSTRUCTION

Negative error 🟥⬛🟩 Positive error



L1 Weighted

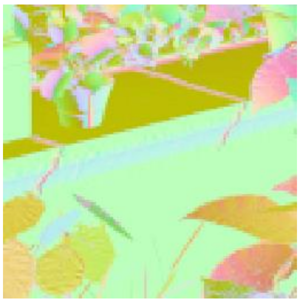
… we can see that L1 reco nstruction suffer from a large energy loss. In comparison, the weighted reconstruction has more uniform error patterns.



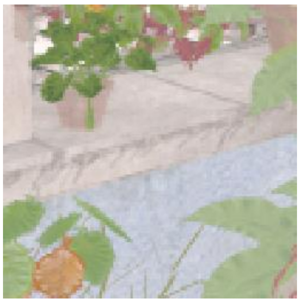
UTILIZING FEATURE BUFFERS

Normals, textures, positions, etc.

Example features



Normal buffer

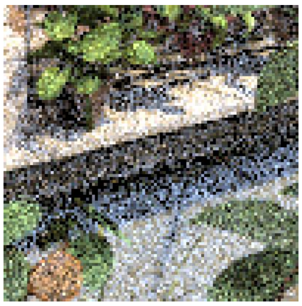


Albedo buffer

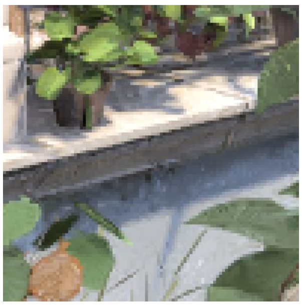
Another idea that have been explored is to use the features maps to guide the reconstruction. As features maps, we can have normal vectors, surface albedos and so on.

DENOISING WITH FEATURE BUFFERS

Robust Denoising using Feature and Color Information
[Rousselle et al., 2013]

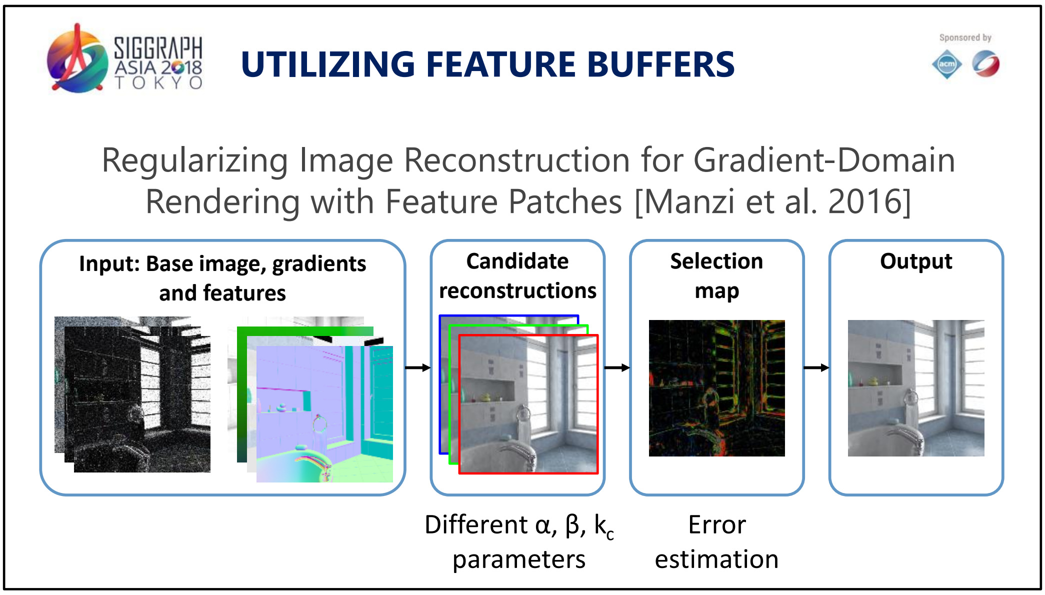


Path tracing 55s



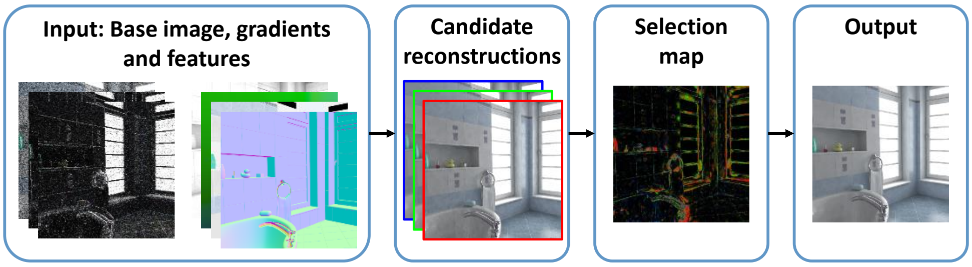
Denoising 57s

These information is usually used in post-process denoising techniques to filter noise in path tracing. These techniques are applied directly on the primal image. They do not use the estimated gradients from shift mapping that usually have less variance compared to the primal.



UTILIZING FEATURE BUFFERS

Regularizing Image Reconstruction for Gradient-Domain Rendering with Feature Patches [Manzi et al. 2016]



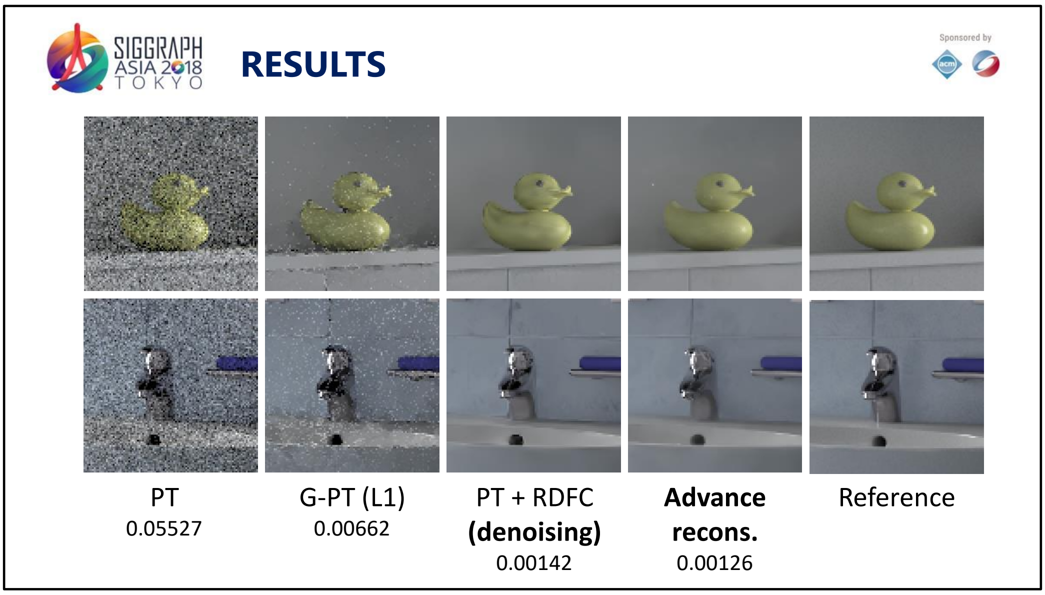
Input: Base image, gradients and features
Candidate reconstructions
Selection map
Output

Different α, β, k_c parameters

Error estimation

Manzi et al. have explored this combination of gradient-domain and denoising techniques. The authors perform a constraint Poisson reconstruction based on the feature map information.

Their algorithms have different parameters that produce potentially different images with less or more bias. It is in general quite difficult to know in advance the optimal parameters. To address this, they perform the reconstruction with multiple parameter set and combines them using a selection map. This selection map picks the best technique in term of variance/bias pixelwise.



RESULTS

PT
0.05527

G-PT (L1)
0.00662

PT + RDFC **(denoising)**
0.00142

Advance recons.
0.00126

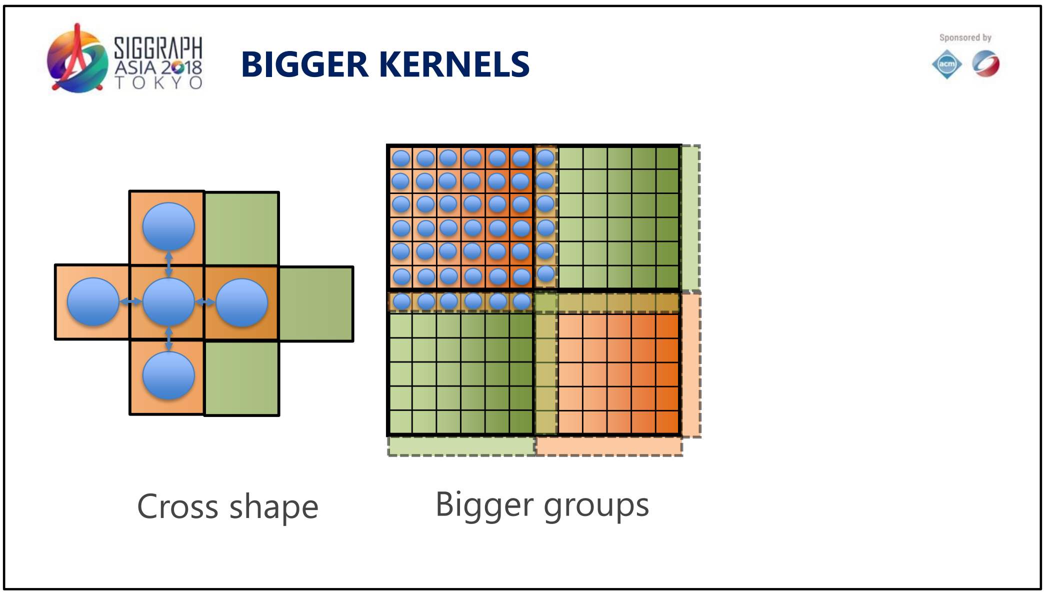
Reference

The results shows that compared to denoising technique, this new technique can achieve better reconstruction. Moreover, this technique is able to remove the problematic fireflies in the L1 reconstruction.

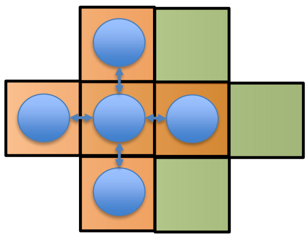
However, the optimization was quite slow. Accelerating it will make it more useful in practice.

Non-Uniform Base and Offset Pixels

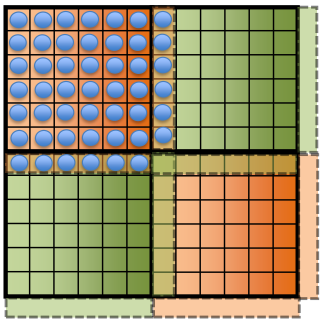
Traditional gradient-domain rendering assumes a uniform pixel grid where base and offset pixels are sampled horizontally and vertically. In fact, this is just for convenience in the implementation, and this pattern might not fit well to actual structures in the image. Let’s explore more advanced techniques that leverage non-uniform base and offset pixels.



BIGGER KERNELS



Cross shape



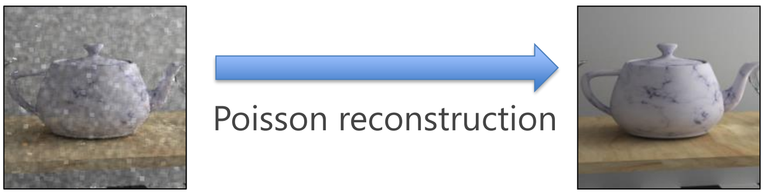
Bigger groups

In gradient-domain, we usually use cross shape finite difference where we shift the center pixel to 4 pixels neighbors. However, gradient-domain rendering can also work with other tiles shapes. For example, we can generate overlapping tiles by extending only one side and perform the shift mapping **over all the pixels** inside the same tile.

PATH REUSING

Gradient-Domain Path Reusing [Bauszat et al., 2017]

- Make the connection between gradient-domain and path reuse [Bekaert et al. 2002]
- Uses overlapping square tiles and perform a Poisson reconstruction



This is the approached proposed by gradient-domain path reusing. In fact, shifting path inside a tile have been explored in path reuse before. This paper brings two new ideas:
1) The shift mapping operator used in gradient-domain rendering to shift the path. Before this, path reuse was not efficient on specular or glossy surfaces.
2) The overlapping of the tiles and the Poisson reconstruction process can reduce the image artefacts introduced by the path correlation.

PATH REUSING

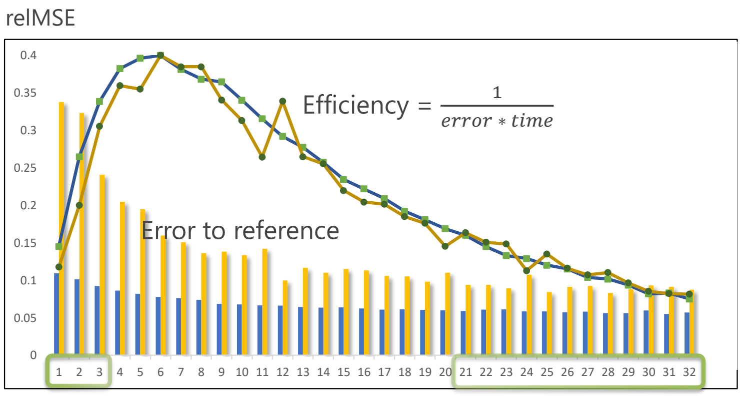


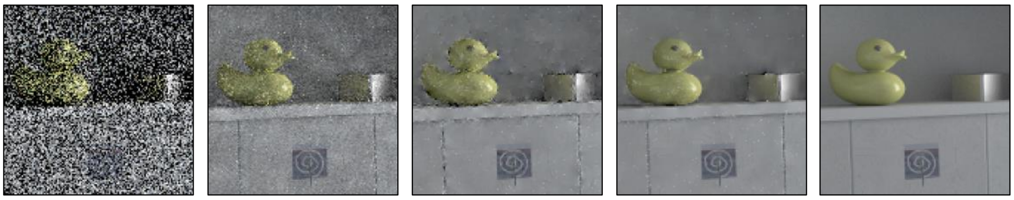
Figure courtesy of Bauszat et al. [2017]

One question is what is the tile size we should use? Inside their paper, the authors have done a small experiment on two scenes by plotting the relative error when increasing the tile size. We can see that small tile size have the largest error. However, at a certain point, too big tile did not bring any benefits.

However, this plot does not include the rendering time. If we take into account the rendering time and display the efficiency, we have another curves. This curves said that using too big tiles is not the good idea. The reason is that bigger tiles means more shift mapping be performed which is costly.

The optimal size is a tile around 6 to 7 pixels.

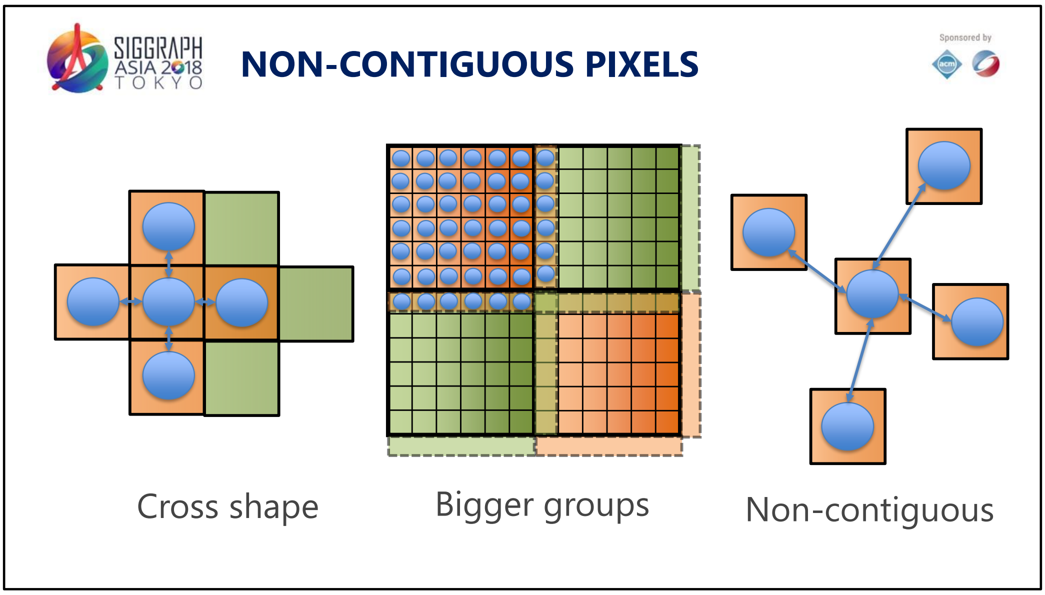
PATH REUSING



Path tracing | Path reusing | G-PT cross | G-PT tile (6) | Reference

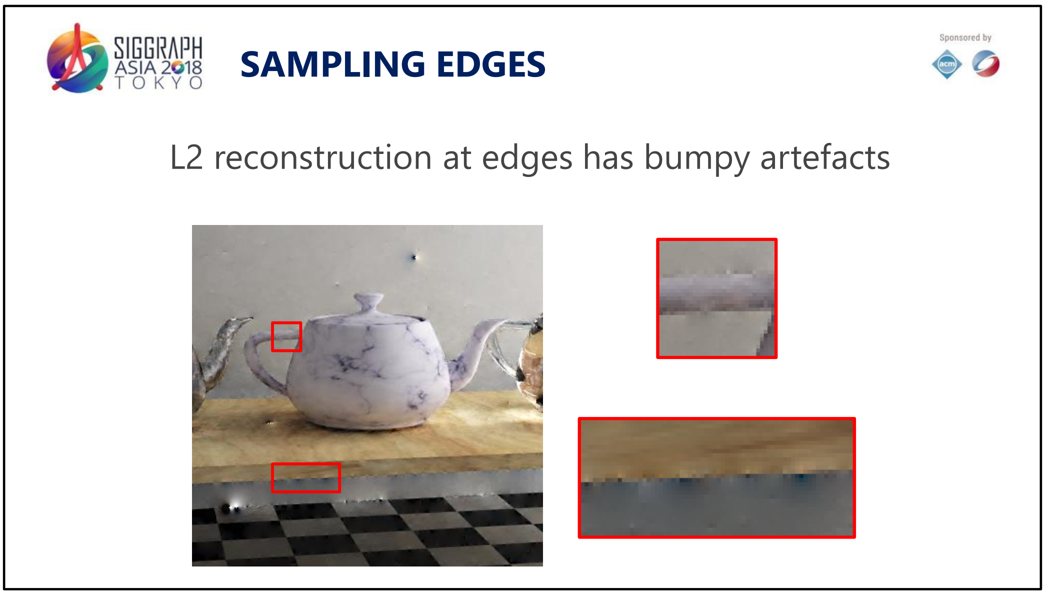
Figure courtesy of Bauszat et al. [2017]

Here are some results about this technique. We can see their methods can achieve better results compared to classical G-PT with cross shape and primal-domain path reusing. Note that inside this paper the authors use a Russian Roulette to select the path to shift. Please refer to the paper for more details.



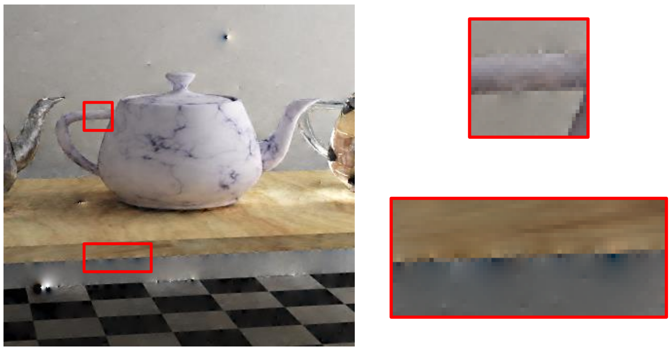
NON-CONTIGUOUS PIXELS

So far we only consider axis-aligned kernels. In general, the kernels can have arbitrary shapes.

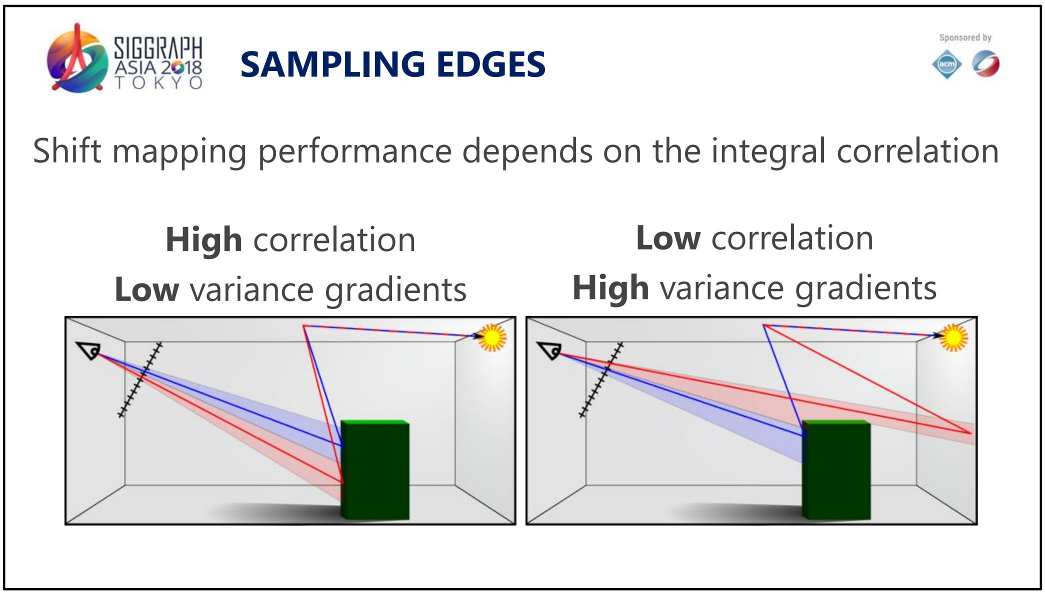


SAMPLING EDGES

L2 reconstruction at edges has bumpy artefacts



This is motivated by the fact that conventional shift mapping usually have issues with geometrical edges.

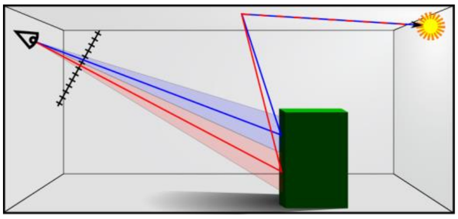


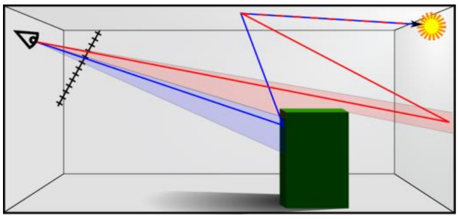
SAMPLING EDGES

Shift mapping performance depends on the integral correlation

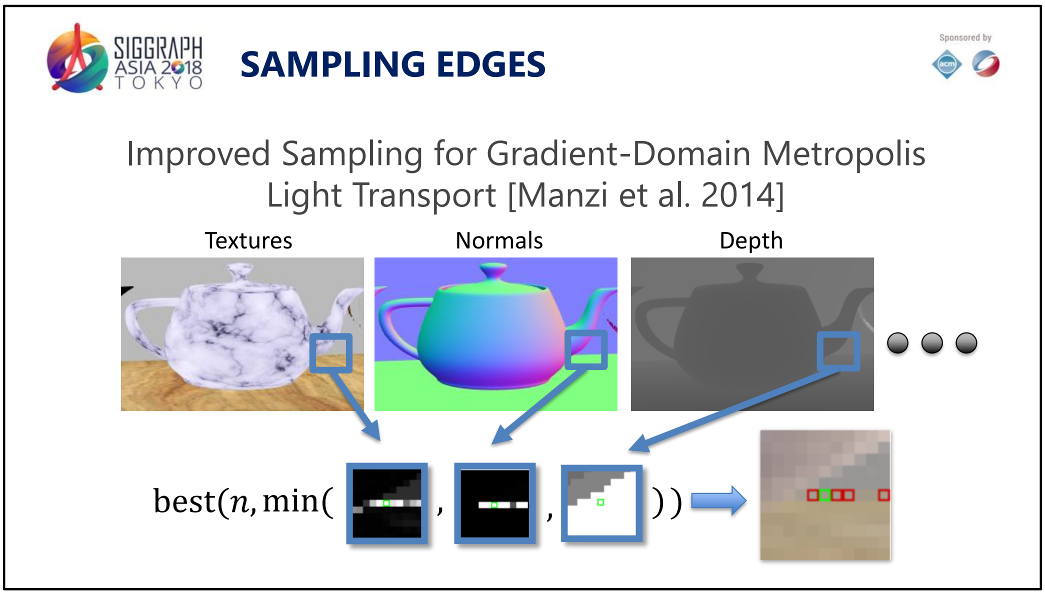
High correlation
Low variance gradients

Low correlation
High variance gradients





The reason of this problem is due to high variance inside the gradients. Indeed, when two pixels see the similar area, their integral are very similar. In this case shift mapping operator can perform low variance estimate of the gradient value. However, if now we are at the edge, the two primary points can end up to be very far away. This is problematic as their integral will be very different and correlate these two pixels will leads to high variance gradients.

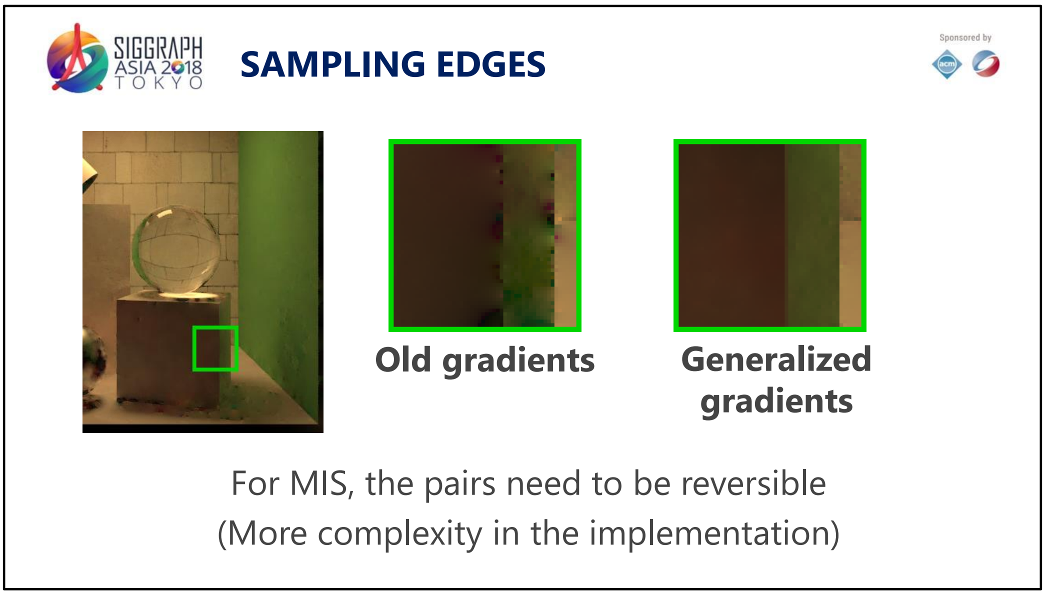


SAMPLING EDGES

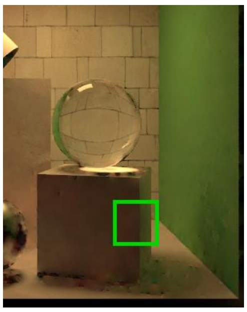
Improved Sampling for Gradient-Domain Metropolis Light Transport [Manzi et al. 2014]

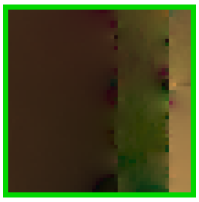
The idea of Manzi et al. [2014] is to detect similar pixel values using features maps. This is similar to what has been used in image-space denoising techniques as they assume that feature buffers in general are very similar to the final image.

To define the kernel shape, the authors proposed the following way. For each pixel we look at a window of the different feature maps. For each feature map, a weight is computed similar to denoising methods. Then the weights are concatenated and the N best pixels are extracted. These pixels will be used to determine where to shift the path.

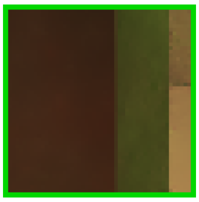


SAMPLING EDGES





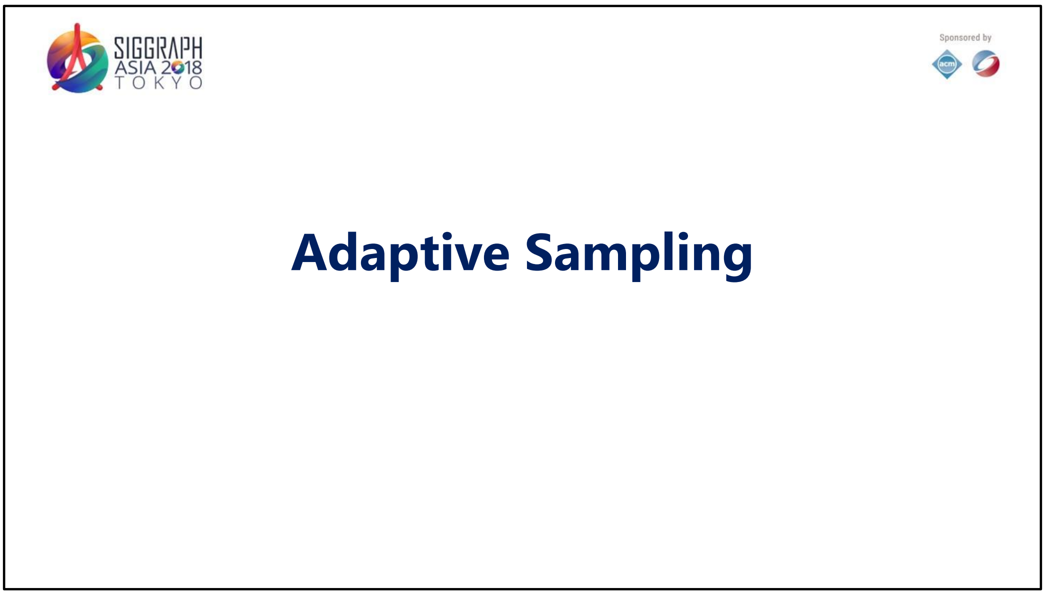
Old gradients



Generalized gradients

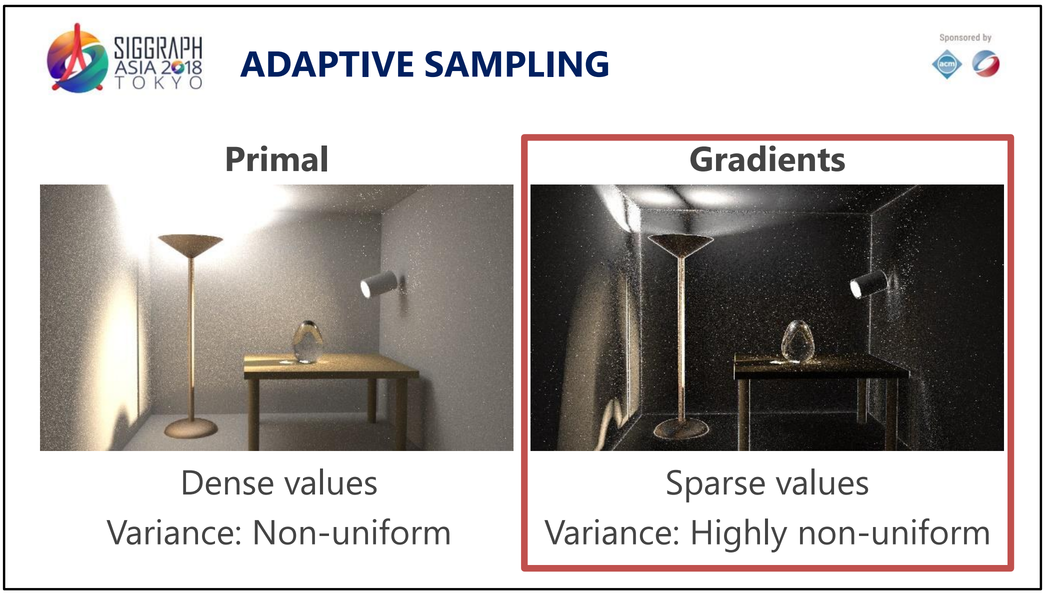
For MIS, the pairs need to be reversible
(More complexity in the implementation)

This is a comparison between old gradients and this generalized gradient formulation. We can see that this generalized gradient formulation works much better on the edges. Note that in case of MIS usage, the shift mapping need to be reversible. So extra care need to be done here. Moreover, as the gradient does not have structure anymore, the reconstruction algorithm might be a little bit more complex to implement. For details, please refer to the paper.



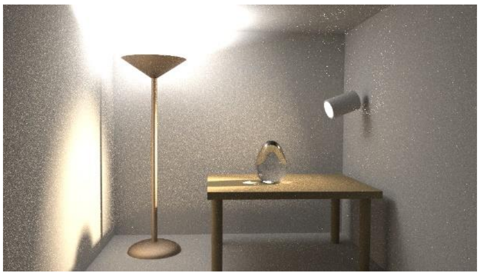
Adaptive Sampling

Now let's talk about the last point of this state of the art for gradient-domain techniques: adaptive sampling.



ADAPTIVE SAMPLING

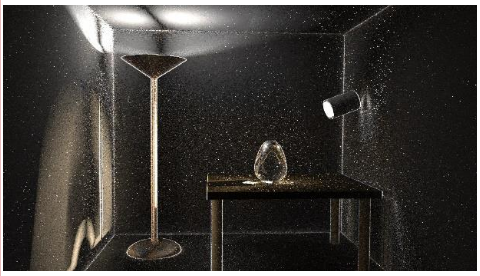
Primal



Dense values

Variance: Non-uniform

Gradients

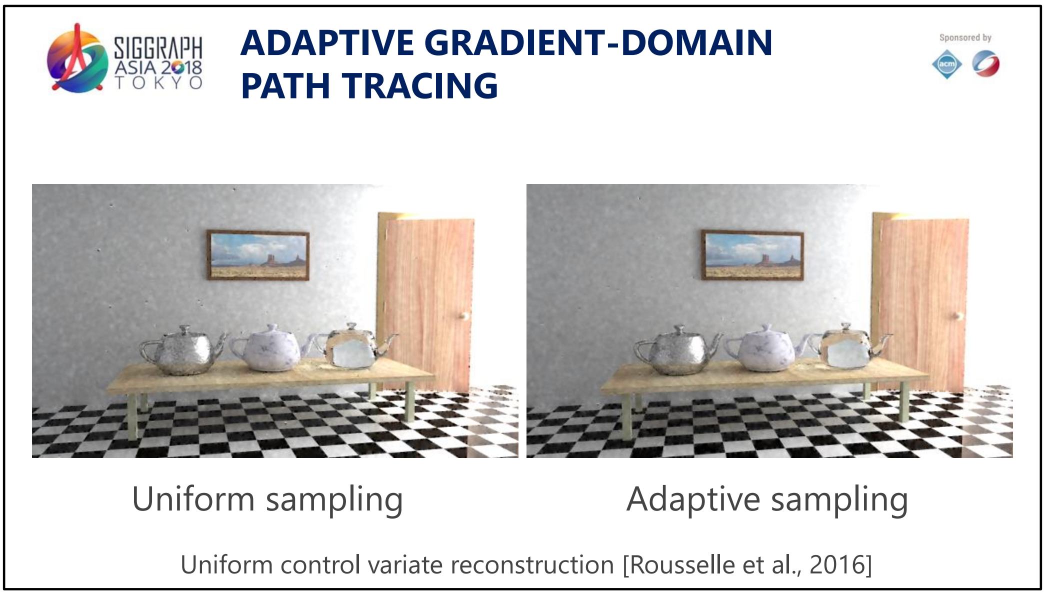


Sparse values

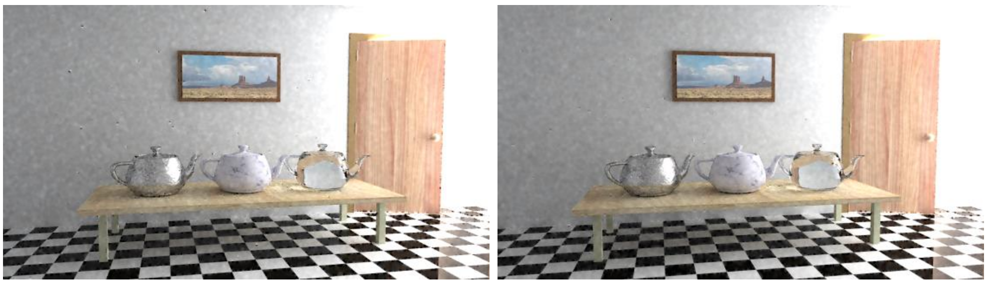
Variance: Highly non-uniform

Adaptive sampling is a classical approach for primal rendering techniques. Indeed, the energy inside the primal is usually dense but the variance is not uniform. For faster convergence, we want to concentrate samples to the area of higher error to improve the efficiency.

In gradient-domain rendering, as we also estimate the gradients, we can apply adaptive sampling. A simple approach that works well is to adaptively sample pixels that has high variance in primal-domain value estimation or gradient value estimation.



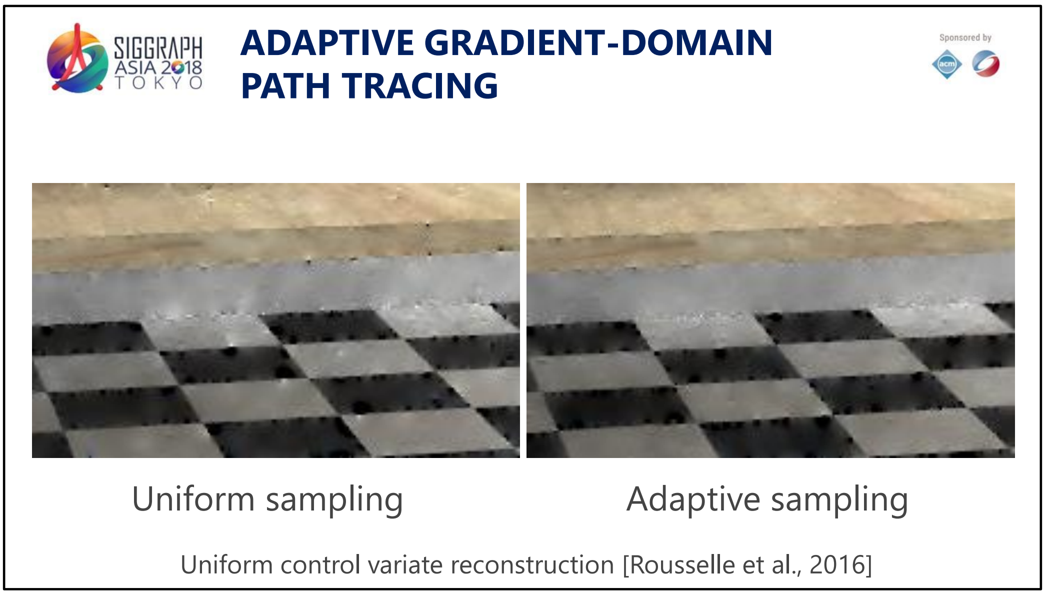
ADAPTIVE GRADIENT-DOMAIN PATH TRACING



Uniform sampling | Adaptive sampling

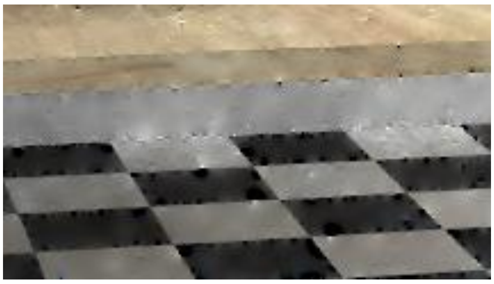
Uniform control variate reconstruction [Rousselle et al., 2016]

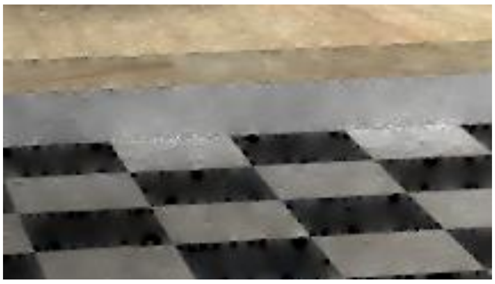
These are results of gradient-domain path tracing using uniform sampling and adaptive sampling. The reconstruction is done with uniform control variate (similar to L2-norm Poisson reconstruction). The image looks quite the same at first, however, when we zoom on edges regions …



ADAPTIVE GRADIENT-DOMAIN PATH TRACING

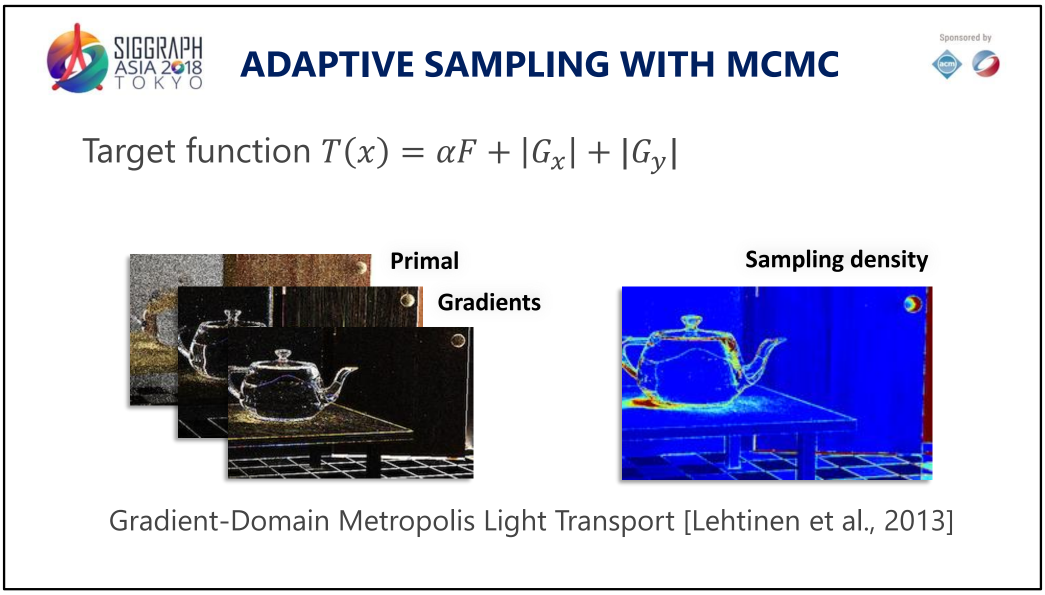
Sponsored by


Uniform sampling


Adaptive sampling

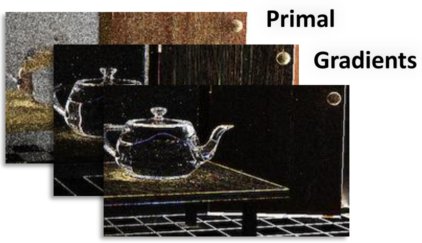
Uniform control variate reconstruction [Rousselle et al., 2016]

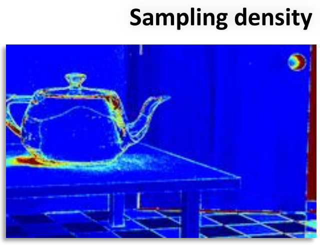
… we can see that adaptive sampling reduces the artefacts compared to uniform sampling. Note that the effect that we get is similar to non-contiguous kernel as this technique reduce the variances inside problematic gradients.



ADAPTIVE SAMPLING WITH MCMC

Target function $T(x) = \alpha F + |G_x| + |G_y|$

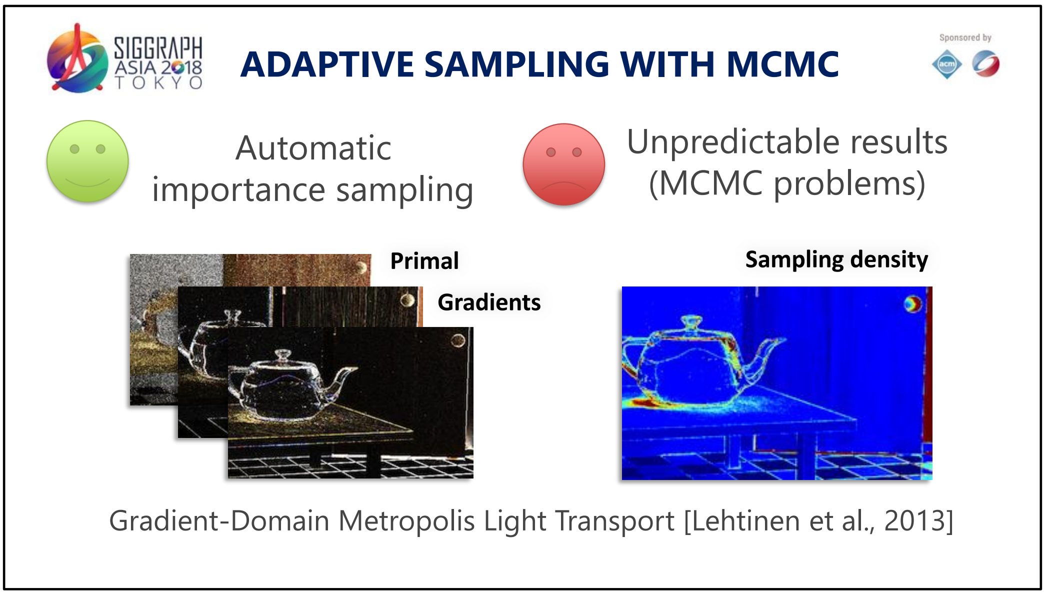




Gradient-Domain Metropolis Light Transport [Lehtinen et al., 2013]

For adaptive sampling, we can also rely on MCMC sampling. MCMC techniques are known to be able to produce samples that follow a target distribution. If this target distribution was proportional to our path throughput and the gradients, we can achieve quite good importance sampling.

Using gradient-domain rendering and MCMC has been explored in the seminal paper by Lehtinen et al. [2013]. This method focus the samples on the gradients.

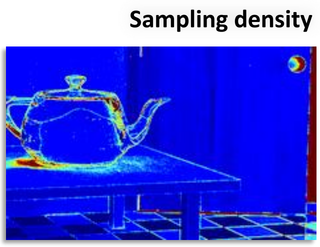


ADAPTIVE SAMPLING WITH MCMC

Automatic importance sampling

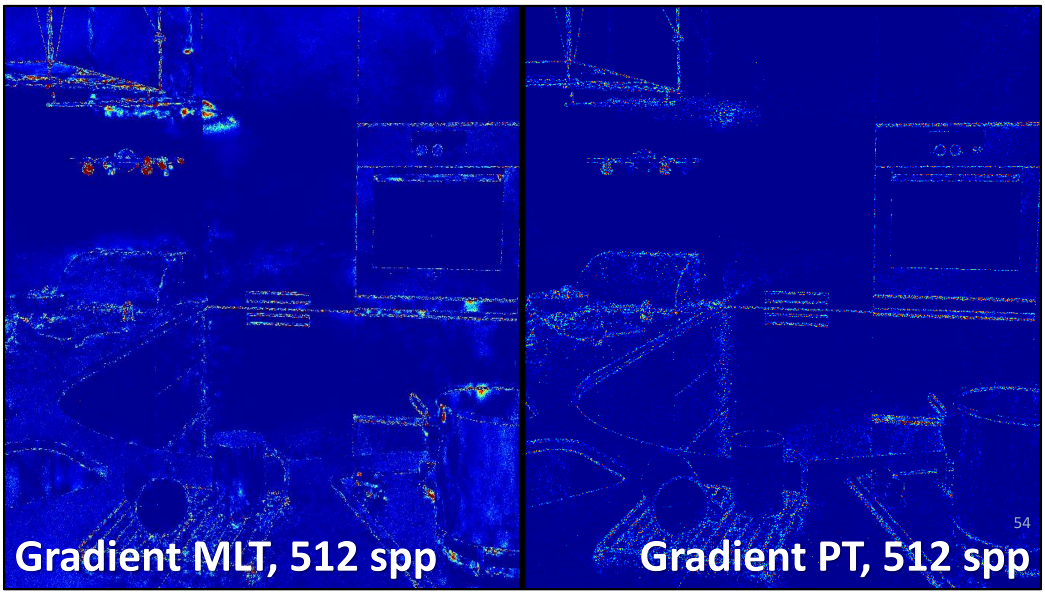
Unpredictable results (MCMC problems)

Primal
Gradients



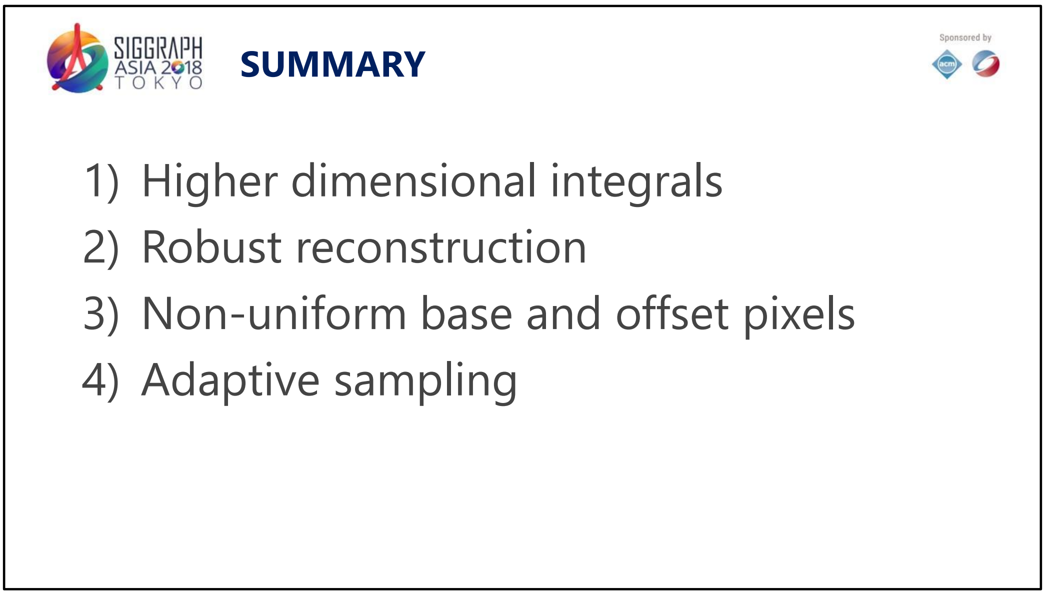
Gradient-Domain Metropolis Light Transport [Lehtinen et al., 2013]

However, implementing MCMC can be tricky to do. It also comes with major drawback such as unpredictable convergence, sudden fireflies and so on.



Let’s see an example. Standard Metropolis generally does better than standard path tracing when light transport is extremely challenging. However, its convergence is unpredictable and non-uniform as seen in the splotchiness in the error image on the left. While Gradient Metropolis generally improves the situation, it unfortunately inherits the convergence issues.

In contrast, well-implemented standard path tracing beats MLT in easy transport situations, and converges predictably.



SUMMARY

1) Higher dimensional integrals
2) Robust reconstruction
3) Non-uniform base and offset pixels
4) Adaptive sampling

Let’s conclude this section. We talked about various topics in advanced gradient-domain rendering Note that many of the techniques can be combined. It would be interesting to see more advanced techniques coming in the near future.

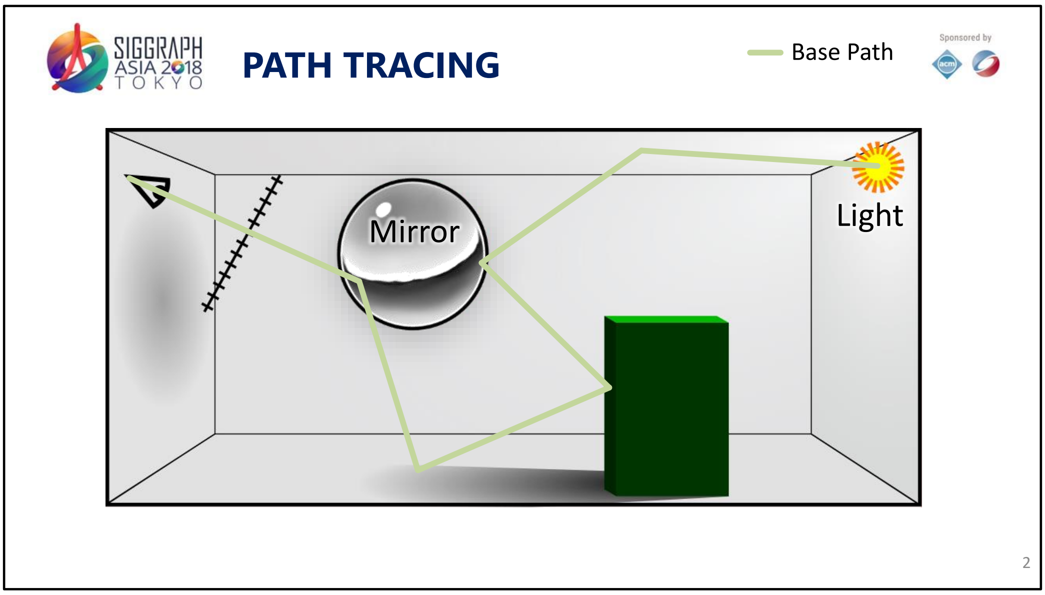
4.7 Practical tips and tricks of implementing gradient-domain rendering

SIGGRAPH ASIA 2018 TOKYO

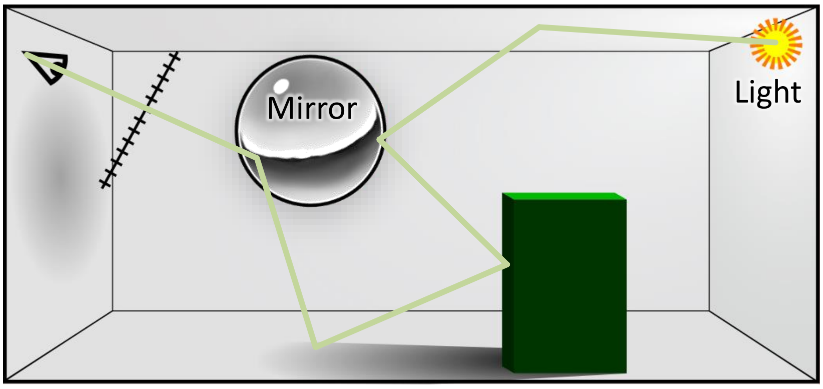
CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

Sponsored by

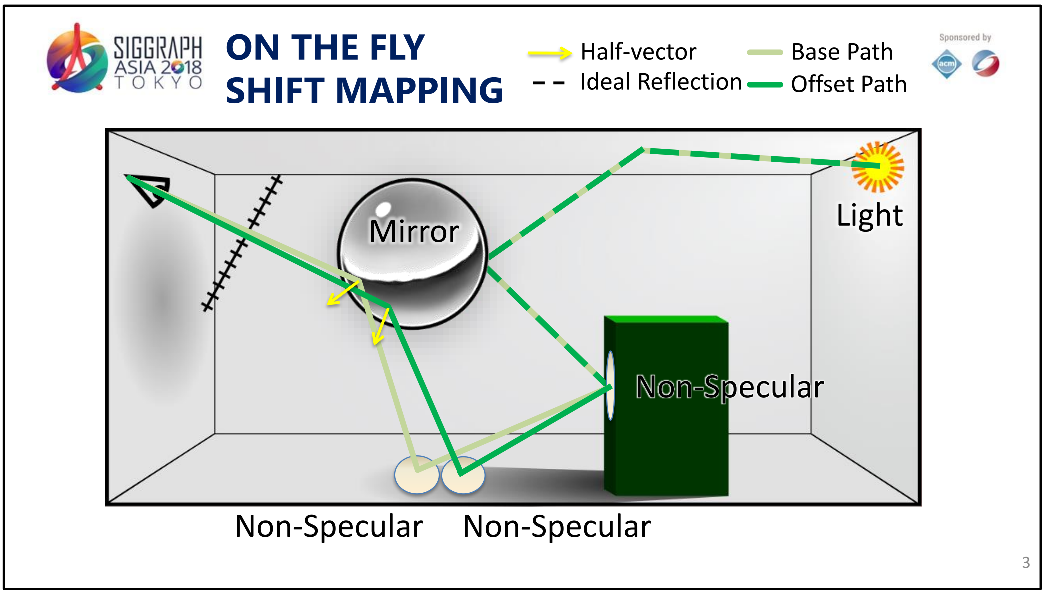
Tips and Tricks for Gradient-domain Rendering Implementation



PATH TRACING



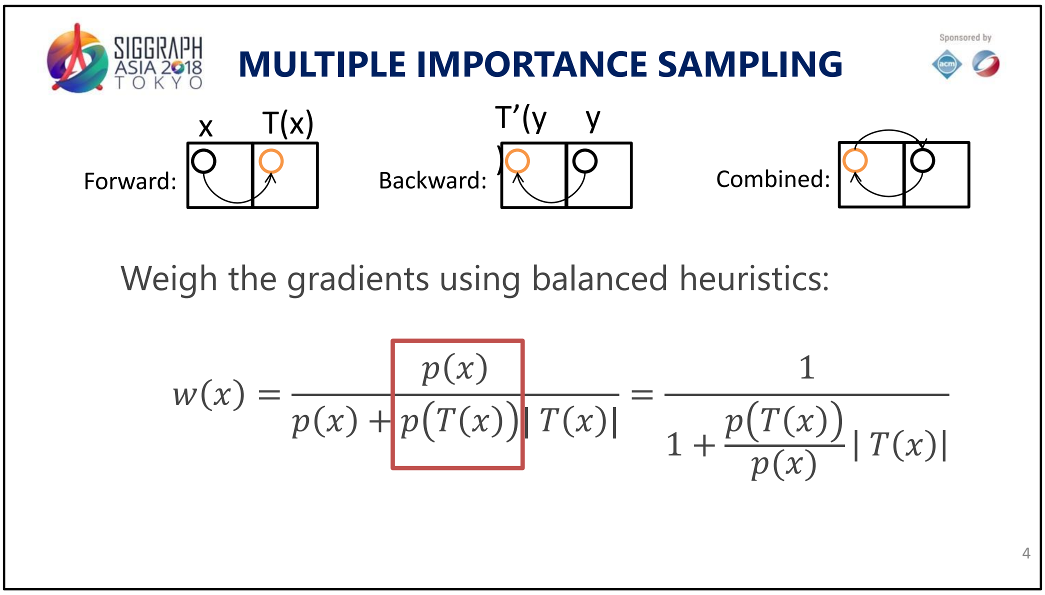
As a typical implementation, a path tracer progressively bounces a path from the camera until it hits the light.



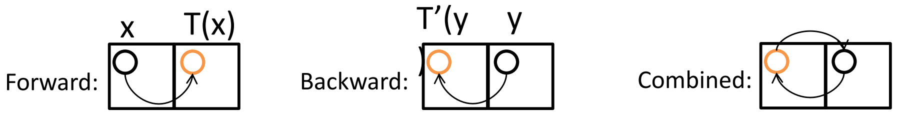
ON THE FLY SHIFT MAPPING

One of the straightforward strategy to integrate gradient sampling into existing path tracers is to perform shift mapping every time a vertex on the base path is sampled. This is the strategy used in the original gradient-domain path tracing implementation.

Some housekeeping has to be added, such as tracking path states to know whether the offset path is already reconnected to the base path. When it is reconnected, we simply continue the converged path until it hits the light.



MULTIPLE IMPORTANCE SAMPLING

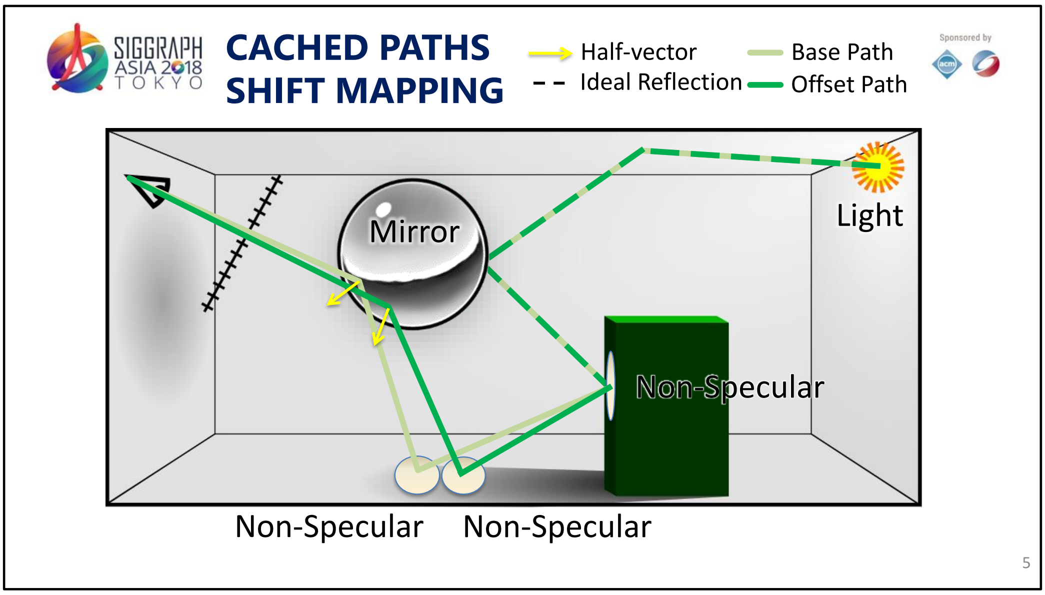


Weigh the gradients using balanced heuristics:

$$w(x) = \frac{p(x)}{p(x) + p\big(T(x)\big)|\,T(x)|} = \frac{1}{1 + \frac{p\big(T(x)\big)}{p(x)}|\,T(x)|}$$

4

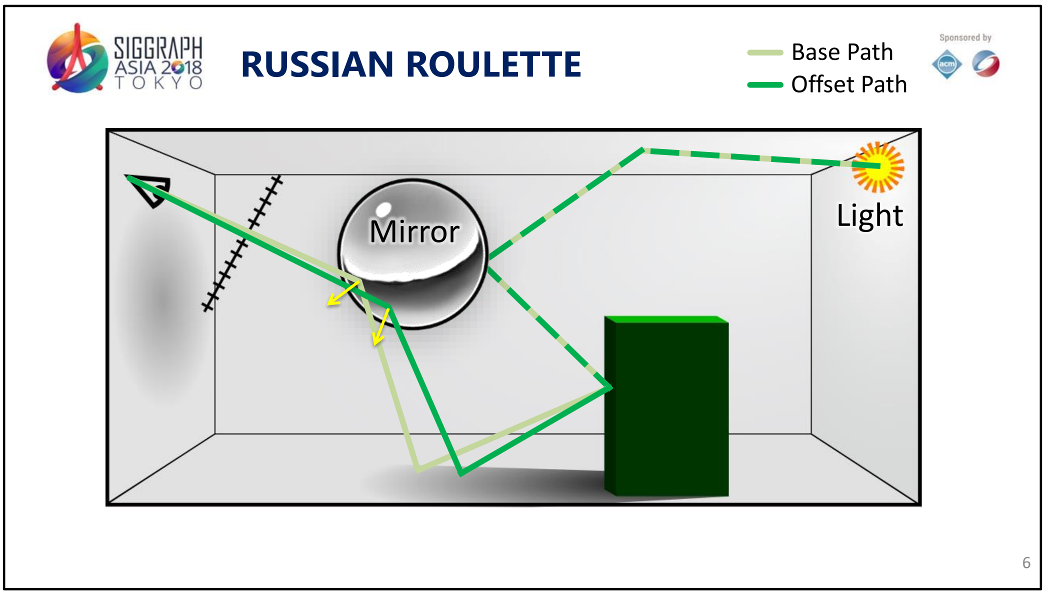
For MIS, it is also worth storing pdf ratios as in bidirectional path tracing to avoid floating point accuracy issues.



CACHED PATHS SHIFT MAPPING

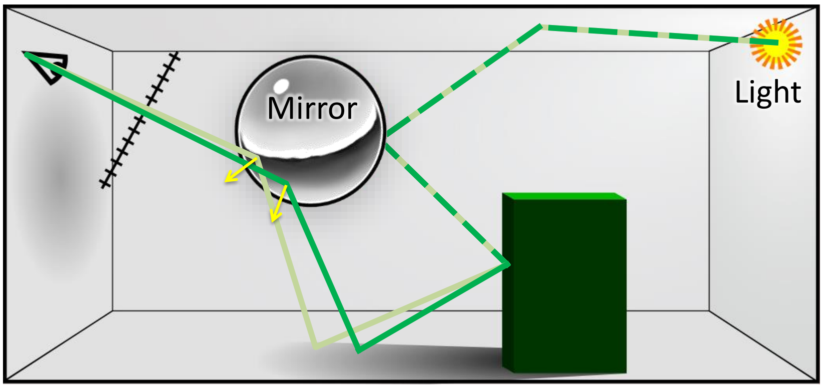
A second strategy is to cache all necessary information during base path generation, and utilize them for shift mapping after the base path is generated. This is more general and cleaner as it helps one to separate gradient-domain implementation from the actual path tracing implementation. While there are some small overheads due to extra path storage, it is more manageable when a path tracer becomes complicated, e.g., those that handles surface and volume rendering at the same time.

The current implementation of gradient-domain photon density estimation also stores all light subpaths that used to generate the photons.



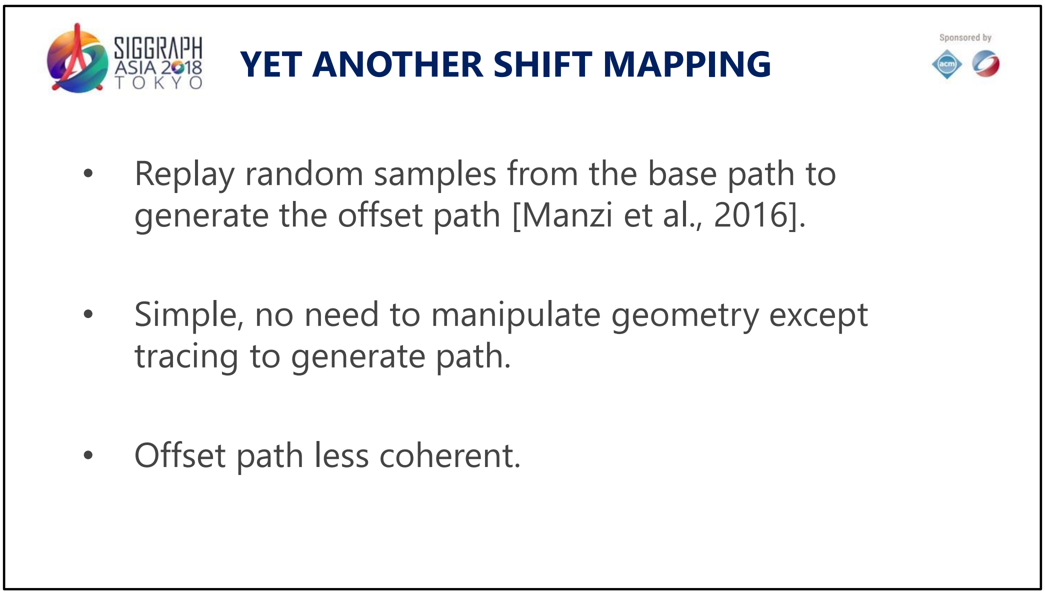
RUSSIAN ROULETTE

Base Path
Offset Path
Mirror
Light
6



A further possible optimization from storing the base paths is that we can perform an extra Russian roulette to decide whether we should perform shift mapping to generate the offset path or not. This is helpful in cases when the base path does not contribute much energy, and tracing offset paths become wasteful.

Note that Russian roulette is a random process, and it comes with a cost of extra variance in the gradients.

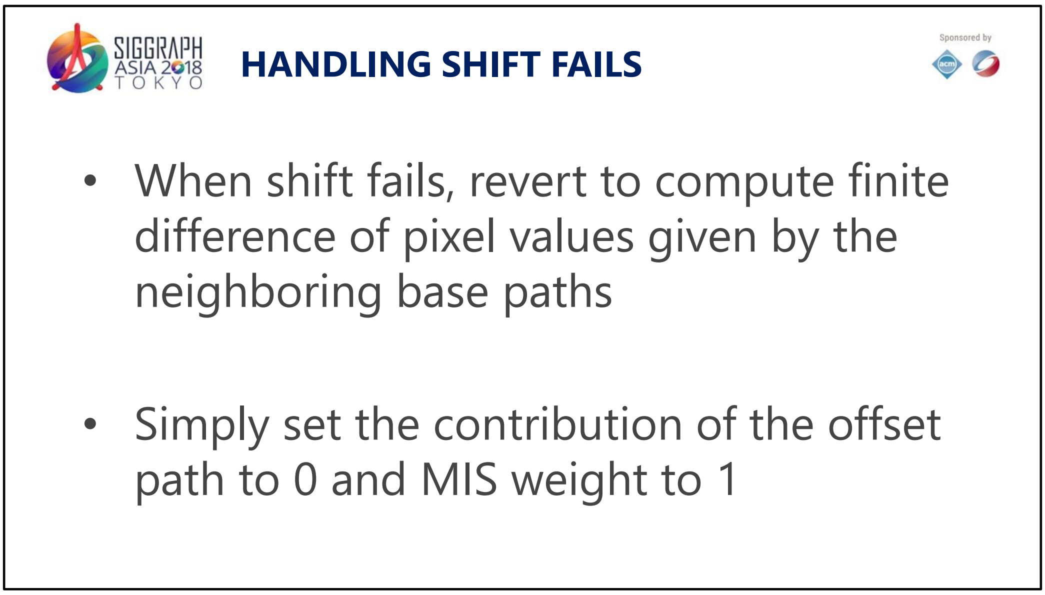


YET ANOTHER SHIFT MAPPING

- Replay random samples from the base path to generate the offset path [Manzi et al., 2016].
- Simple, no need to manipulate geometry except tracing to generate path.
- Offset path less coherent.

Another shift mapping that could be useful for practical implementation is to replay the random numbers sampled on the base path to generate the offset path. This strategy is simple to implement. However, the offset paths might not be as coherent to shift mapping that manipulates path geometry explicitly such as half-vector copy or manifold exploration.

This shift mapping is used for temporal gradient-domain path tracing [Manzi et al., 2016], and could be helpful for making the gradient-domain version of primary-sample space Metropolis light transport (PSSMLT).



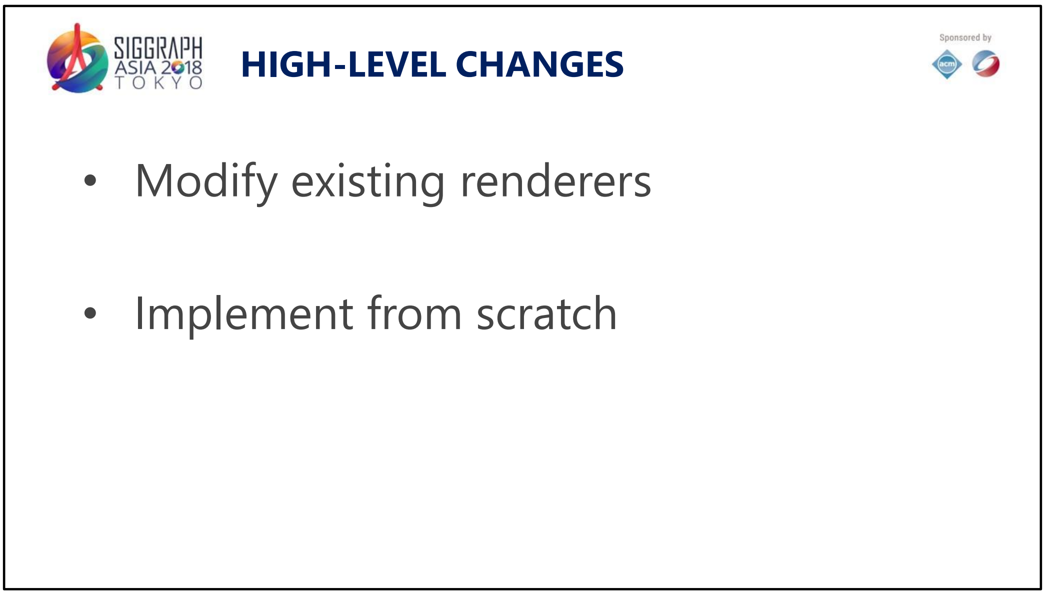
HANDLING SHIFT FAILS

- When shift fails, revert to compute finite difference of pixel values given by the neighboring base paths
- Simply set the contribution of the offset path to 0 and MIS weight to 1

In the implementation, a lot of times we have to deal with shift fail, i.e., when the offset path is invalid. This requires us to revert to computing the gradients using the finite difference of the base path and the neighbor base path.

For convenience, in the implementation we can simply return offset path contribution as zero with MIS weight as 1. When forward and backward shift mapping is guaranteed, the end result (on average) is the mentioned finite difference.

Note that in case of adaptive sampling, make sure the number of paths in the current and neighbor pixel is equal.



HIGH-LEVEL CHANGES

- Modify existing renderers
- Implement from scratch

The first approach is what we has been doing: modify Mitsuba, the open-source research oriented renderers to include new path integrators that performs shift mapping and image-space reconstruction.

This can be implemented as a direct code change or a plugin to existing renderers. For plugin, this requires the renderers to store and pass to the plugin necessary light path information for computing gradients. This approach has some overheads but is needed if one does not have access to entire path tracer implementations. The shift mapping approach that replays random numbers might be simpler to implement in this case.

There has been some attempts of the second approach, such as the 500+-line gradient-domain path tracers by Tzu-Mao Li. We also reimplement a gradient-domain path tracers from scratch in Rust, a programming language that is made to avoid segmentation faults.

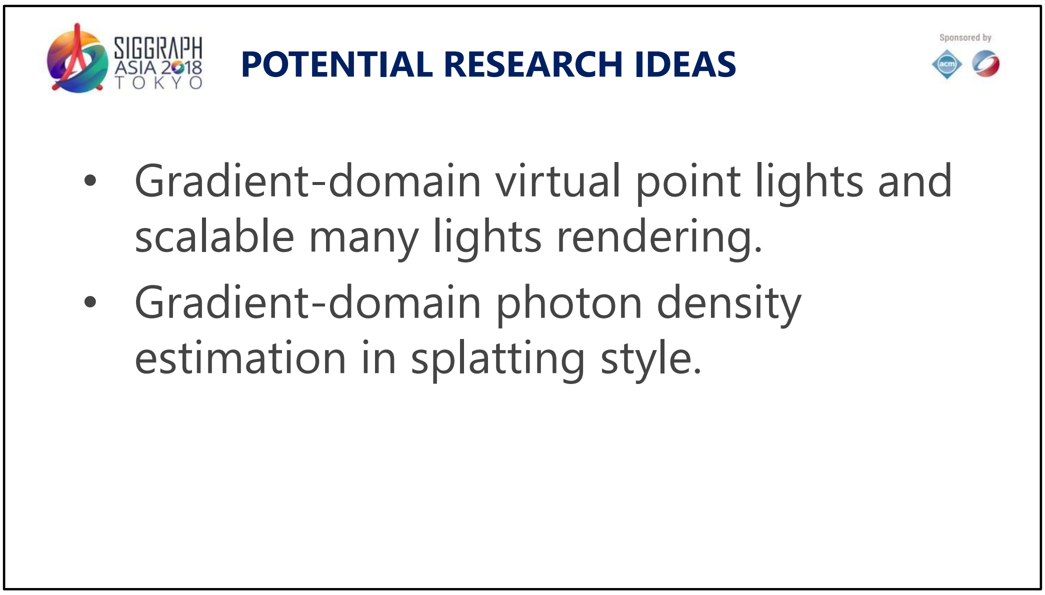
4.8 Conclusion and Q&A

SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

Sponsored by

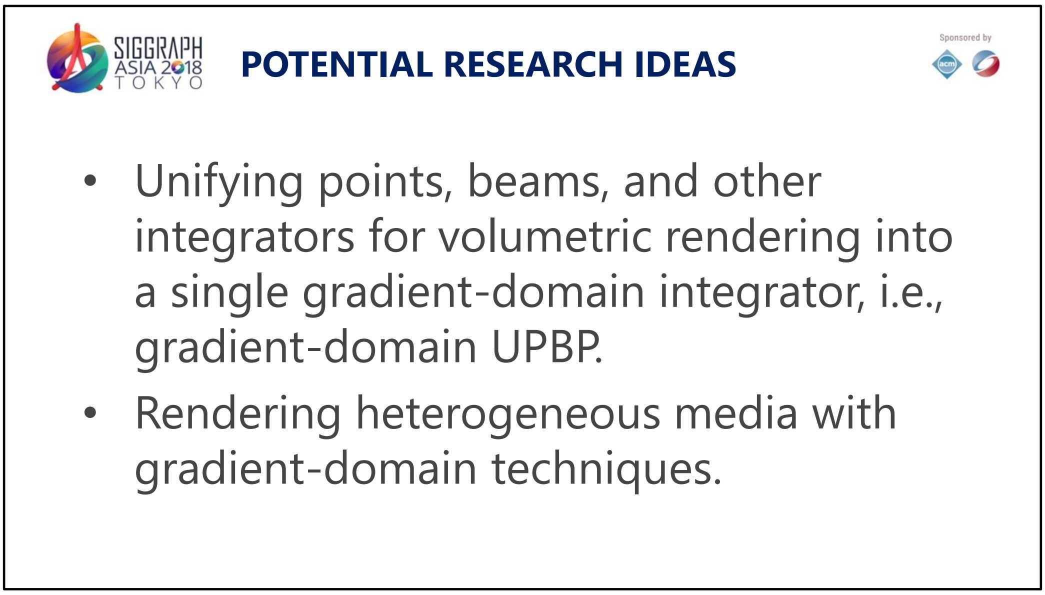
The Next Frontier



POTENTIAL RESEARCH IDEAS

- Gradient-domain virtual point lights and scalable many lights rendering.
- Gradient-domain photon density estimation in splatting style.

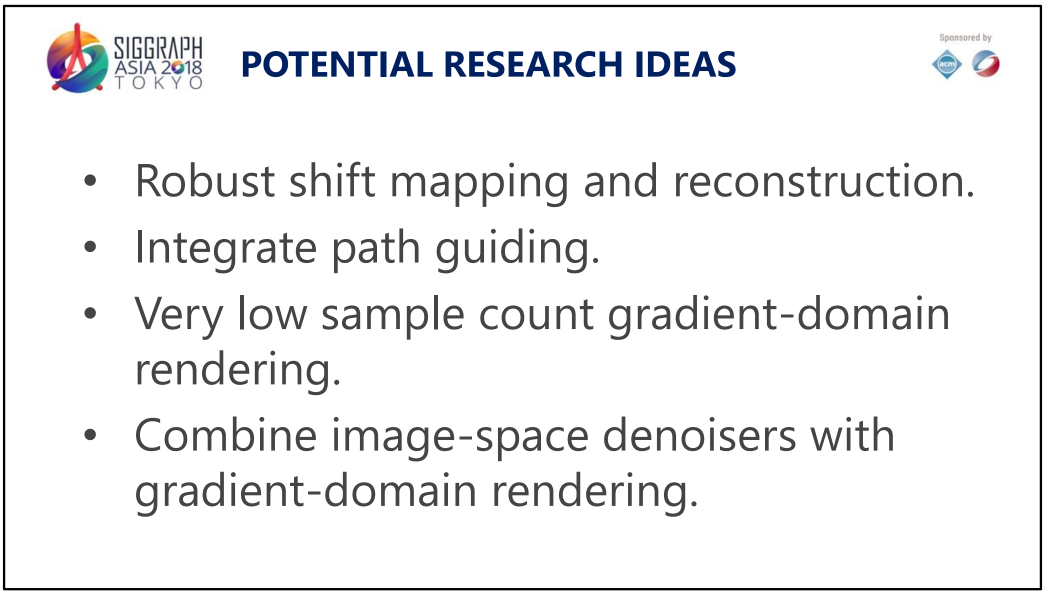
There requires further investigation to bring even more integrators to the gradient domain.



POTENTIAL RESEARCH IDEAS

- Unifying points, beams, and other integrators for volumetric rendering into a single gradient-domain integrator, i.e., gradient-domain UPBP.
- Rendering heterogeneous media with gradient-domain techniques.

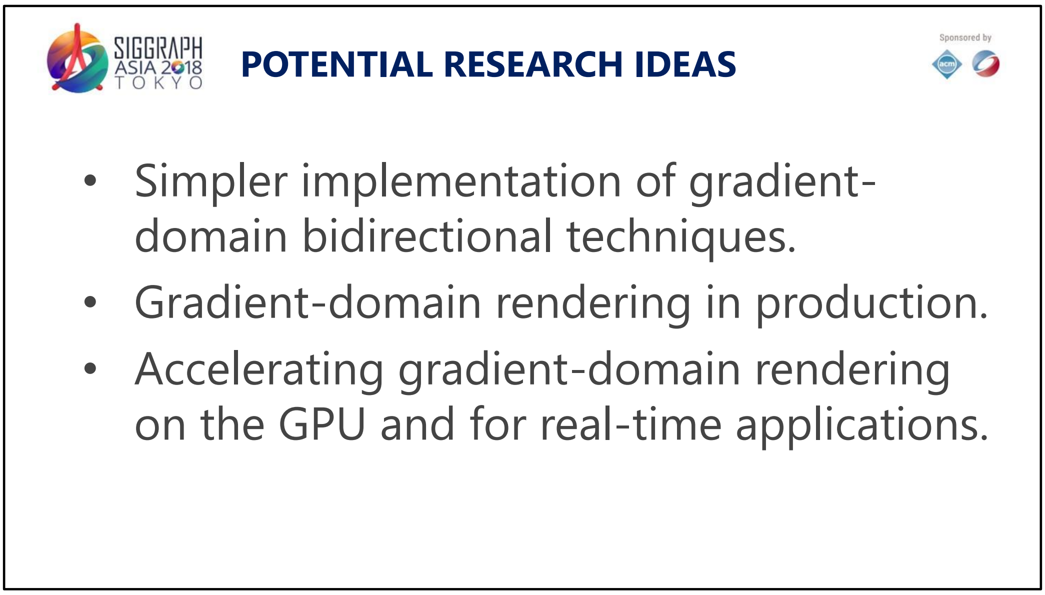
Volumetric rendering also deserves more studies, particularly for heterogeneous media such as rendering cloud, fire, cloth and fabrics.



POTENTIAL RESEARCH IDEAS

- Robust shift mapping and reconstruction.
- Integrate path guiding.
- Very low sample count gradient-domain rendering.
- Combine image-space denoisers with gradient-domain rendering.

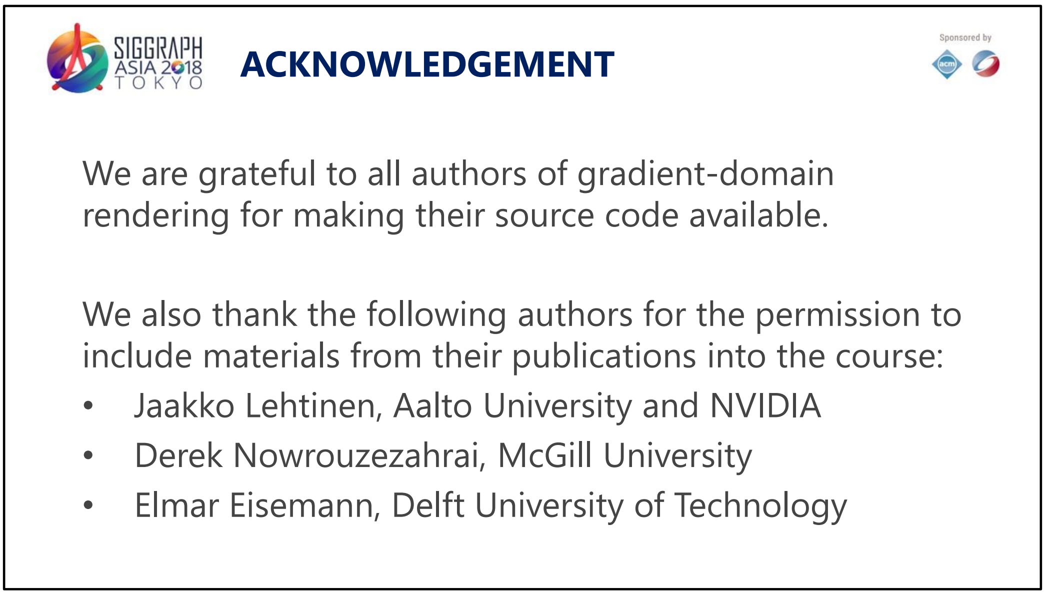
There is also a need to make gradient-domain rendering itself more robust. More evaluation of its performance and comparing to path guiding and other machine learning based denoisers that are becoming more common in the industry is an important task.



POTENTIAL RESEARCH IDEAS

- Simpler implementation of gradient-domain bidirectional techniques.
- Gradient-domain rendering in production.
- Accelerating gradient-domain rendering on the GPU and for real-time applications.

Bringing gradient-domain techniques to production requires significant changes in existing renderers. Researches of techniques to implement gradient-domain rendering as a plugin onto existing renderers will be very useful.

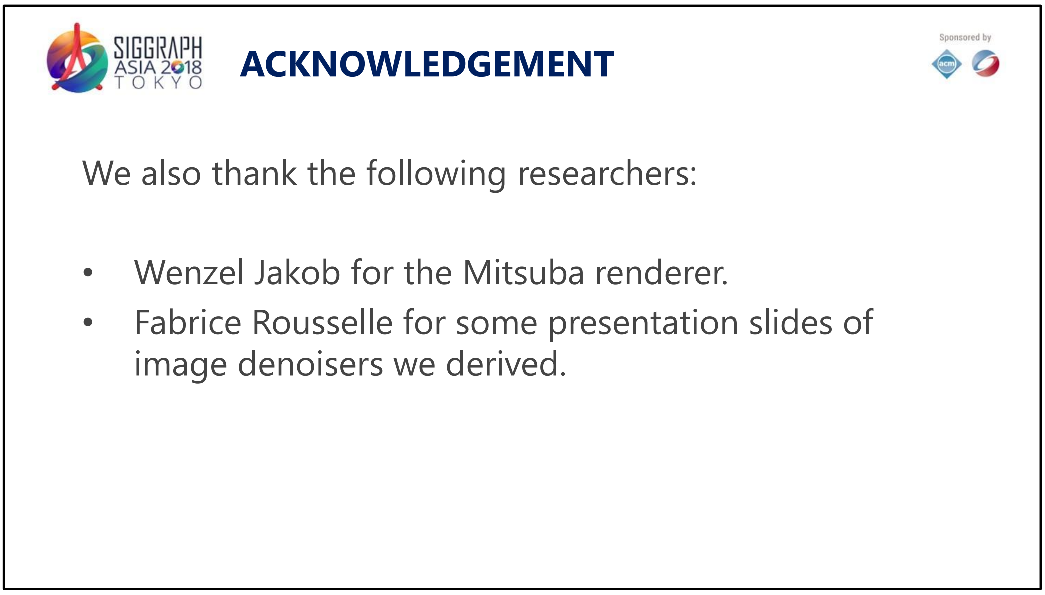


ACKNOWLEDGEMENT

We are grateful to all authors of gradient-domain rendering for making their source code available.

We also thank the following authors for the permission to include materials from their publications into the course:

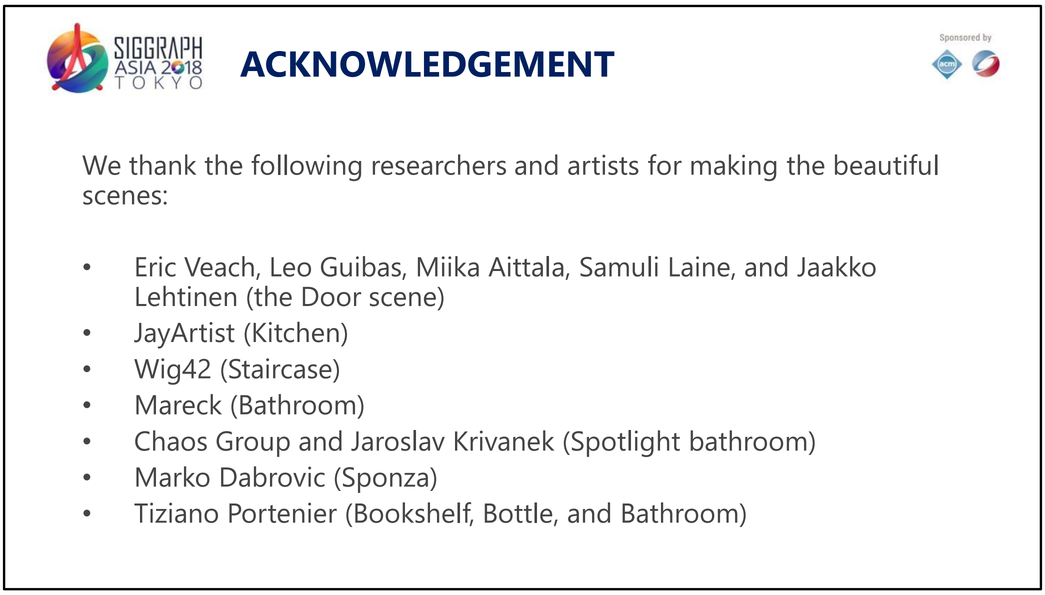
- Jaakko Lehtinen, Aalto University and NVIDIA
- Derek Nowrouzezahrai, McGill University
- Elmar Eisemann, Delft University of Technology



ACKNOWLEDGEMENT

We also thank the following researchers:

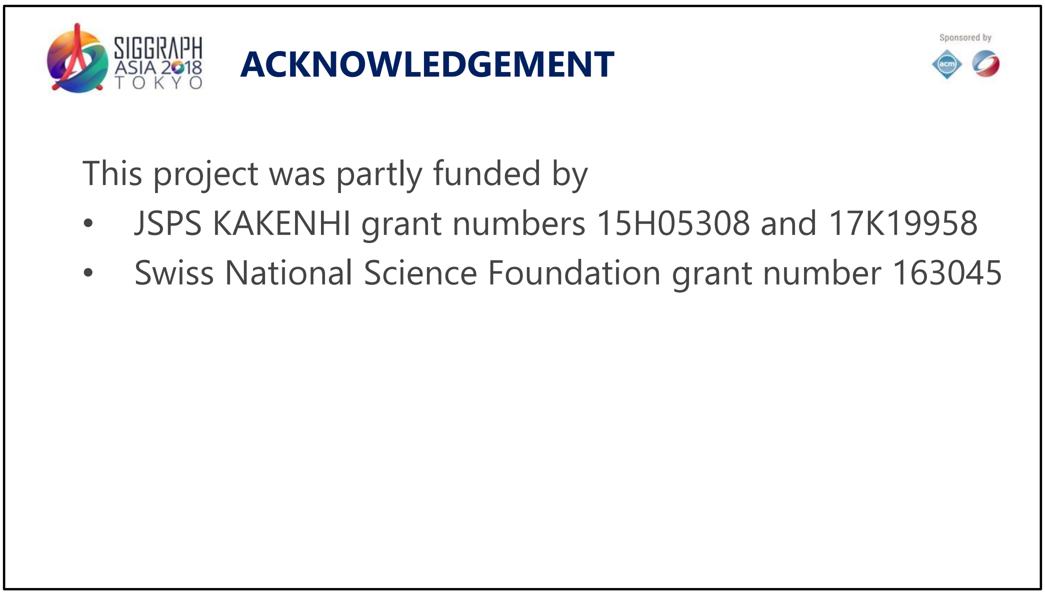
- Wenzel Jakob for the Mitsuba renderer.
- Fabrice Rousselle for some presentation slides of image denoisers we derived.



ACKNOWLEDGEMENT

We thank the following researchers and artists for making the beautiful scenes:

- Eric Veach, Leo Guibas, Miika Aittala, Samuli Laine, and Jaakko Lehtinen (the Door scene)
- JayArtist (Kitchen)
- Wig42 (Staircase)
- Mareck (Bathroom)
- Chaos Group and Jaroslav Krivanek (Spotlight bathroom)
- Marko Dabrovic (Sponza)
- Tiziano Portenier (Bookshelf, Bottle, and Bathroom)



ACKNOWLEDGEMENT

This project was partly funded by

- JSPS KAKENHI grant numbers 15H05308 and 17K19958
- Swiss National Science Foundation grant number 163045

We hope you find this course material useful. Please feel free to contact us if you have any questions or spot any errors. ☺