Path Guiding in Production and Recent Advancements

Sebastian Herholz
Intel Corporation
Karlsruhe, Germany
sebastian.herholz@gmail.com

Martin Sik
Chaos
Prague, Czech Republic
martin.sik@chaos.com

Lea Reichardt
Walt Disney Animation Studios
Vancouver, Canada
lea.reichardt@disneyanimation.com

Marco Manzi
Disney Research Studios
Zurich, Switzerland
marco.manzi@disneyresearch.com

Figure 1: Examples of different integrations of *path guiding* algorithms in production renderers, such as Blender's Cycles, Chaos's Corona, and Disney Animation's Hyperion. The images showcase different *guiding* specific use cases, such as complex multi-bounce indirect illumination and caustics. First image: scene by Jesús Sandoval. Third image: *Moana 2* ©2024 Disney.

Abstract

Over the last decade, advanced data-driven sampling algorithms, such as path guiding, have made their way from the scientific realm into production renderers [Vorba et al. 2019]. These algorithms enable the rendering of challenging lighting effects (e.g., complex indirect illumination, caustics, volumetric multi-scattering, and occluded direct illumination from multiple lights), which are crucial for generating high-fidelity images. The fact that these algorithms primarily focus on optimizing local importance sampling decisions makes it possible to integrate them into a path tracer, the de facto standard rendering algorithm used in production today ([Fascione et al. 2019],[Jakob et al. 2019]). The theory behind these algorithms has been presented and discussed on various occasions (e.g., in presentations or research papers), and their practical applications in production have been explored in the previous course on *path guiding in production*. Nevertheless, the implementation details or challenges associated with integrating them into a production render are usually unknown or not publicly discussed. This course aims to provide a deeper understanding of how specific guiding algorithms are integrated into and utilized in various production renderers, including Blender's Cycles, Chaos's VRay and Corona, SideFX's Karma, and Disney Animation's Hyperion [Burley et al. 2018]. The presented algorithms and integrations can be categorized into two main groups: the first aims to guide the entire sampling process by utilizing information about the total light transport of the scene, and the second focuses on guiding specific effects, such as caustics.

SIGGRAPH Courses '25, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1543-3/25/08
https://doi.org/10.1145/3721241.3733994

CCS Concepts

• **Computing methodologies → Rendering**; **Ray tracing**.

Keywords

rendering, Monte Carlo, light transport, path guiding

ACM Reference Format:
Sebastian Herholz, Martin Sik, Lea Reichardt, and Marco Manzi. 2025. Path Guiding in Production and Recent Advancements. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Courses (SIGGRAPH Courses '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3721241.3733994

1 Course Presenters

The course will be presented by rendering experts who not only have an academic background but also hands-on production experience.

1.1 Sebastian Herholz

Sebastian Herholz is a ray tracing engineer and light transport researcher at Intel. He studied computer science at the University of Tübingen, focusing on computer graphics, especially on Monte Carlo light transport simulation and importance sampling. His main research focus is on advanced importance sampling techniques, such as path guiding. In 2020, joined Intel to work on making state-of-the-art rendering algorithms ready for daily use in production environments. The outcome of his work is included in Intel's Open Path Guiding Library (OpenPGL), an open-source library for which he serves as the project lead.

1.2 Lea Reichardt

Lea Reichardt is a software engineer at Walt Disney Animation Studios. She is part of the rendering team developing Disney's Hyperion renderer. She studied computer science at the Swiss Federal Institute of Technology in Zurich, with a focus on computer graphics and machine learning. Her work so far at Disney Animation focuses on volume rendering and path guiding.

1.3 Marco Manzi

Marco Manzi is a research scientist at Disney Research Studios in Zurich. He obtained his Ph.D at the University of Bern under the supervision of Matthias Zwicker in 2016. His research interests include sampling and reconstruction methods for Monte Carlo rendering, light transport algorithms, and the intersection between machine learning and rendering.

1.4 Martin Šik

Martin Šik is a senior researcher and lead developer of *Corona Renderer* at *Chaos.* Martin received his Ph.D. from *Charles University in Prague* in 2019, where he studied under the supervision of Jaroslav Křivánek. His primary research interest is in realistic rendering, with a focus on Markov chain Monte Carlo and ordinary Monte Carlo methods for simulating light transport.

2 Course Format

The course format will be in a lecture style with slide presentations. Each of the three main topics will be presented by a different presenter. We will try to showcase our insights and experiences on as many production scene examples as possible during the presentation.

- Introduction (5-10min) Sebastian Herholz
  - Summary of recent advancement since the last course [Vorba et al. 2019]
- Integrating Path guiding into a production render: The Nitty Gritty Details (30min) Sebastian Herholz
  - Introducing OpenPGL
  - Presentation of multiple integration details:
    * Guided Directional Sampling
    * Path Guiding and Russian Roulette
    * Combination with Next-Event Estimation
    * Scattering Event Types and Light Path Expressions
    * Improving Robustness via Roughening
  - Additional Integration Challenges and Advice
- Path Guiding in Hyperion: A Case Study (30min) Lea Reichardt and Marco Manzi
  - Integrating path guiding into a wavefront renderer (i.e., Hyperion)
  - Challenges of recording training data for path guiding when using a wavefront renderer
  - How to debug your path guiding implementation
  - Learnings and results from deploying path guiding into a production renderer (i.e., Hyperion).
- Efficient Rendering of Caustics: Photon Guiding at Corona (20-25min) Martin Šik
  - Showcasing the limitations of traditional path guiding when rendering caustics
  - Presentation of a specialized VCM-based solution for rendering caustics that uses a guiding approach for distributing photons

3 Course Description

This course is the successor of the previous course on *Path Guiding in Production* [Vorba et al. 2019] presented at SIGGRAPH 2019. While the previous course focused on the use cases where *path guiding* is helpful in production rendering, this course will focus on sharing implementation insights and challenges faced when integrating path guiding into a production renderer. The first section (Sec. 3.2) presents a collection of implementation details collected during the integration of path guiding into various production renderers. The second section (Sec. 3.3) presents an explicit case study by Disney Animation on their challenges when experimenting with path guiding in their in-house renderer Hyperion. The third section (Sec. 3.4) of the course presents a specialized guiding approach implemented in Chaos's Corona and VRay renderers to simulate complex caustic effects efficiently.

3.1 Introduction

This course starts with a brief recap of the advancements made in the field of path guiding, research, and production-wise, since the initial *Path Guiding in Production* course [Vorba et al. 2019] held at SIGGRAPH 2019. We will also recap which path guiding techniques were adopted recently in production and which production renderers added support for path guiding since then.

3.2 Integrating Path Guiding into a Production Render: The Nitty Gritty Details

In this section, we briefly recap the concept of local path guiding algorithms and how they improve the efficiency of a renderer, followed by an introduction to Intel's Open Path Guiding Library (OpenPGL), an open-source framework for integrating local guiding algorithms into a production renderer. The following parts of this section focus on presenting various implementation details and insights gathered through multiple integrations of path guiding frameworks such as OpenPGL in production renderers such as Blender, VRay, Karma, or Disney's Hyperion (Sec. 3.3). These insights are usually not found or discussed in path guiding research literature.

3.2.1 Introducing OpenPGL. OpenPGL is an open-source path guiding library that adopts multiple techniques developed by the research community [Herholz et al. 2019; Müller et al. 2017; Rath et al. 2022; Ruppert et al. 2020; Vorba and Křivánek 2016; Xu et al. 2024] and extends and optimizes them to make them production-ready. It offers an easy way to integrate multiple local path guiding techniques into a renderer and is now integrated into several production renderers, including Blender, VRay, Karma, and Disney's Hyperion. OpenPGL provides access to guiding distributions that are proportional to the incoming radiance, as well as querying a radiance cache to get estimates of the outgoing, incoming, and in-scattered radiance. The following sections focus on how a framework like OpenPGL can be integrated robustly into a production rendering system.

3.2.2 Guided Directional Sampling. Guiding the directional sampling process is a key ingredient in local path guiding, which improves the efficiency of a renderer and enables rendering challenging light transport effects. The most commonly used approach is to combine BSDF importance sampling with a guiding distribution proportional to the incoming radiance distribution using *single-sample multiple importance sampling* (MIS) [Veach and Guibas 1995]. Unfortunately, this approach can sometimes decrease the sampling quality compared to BSDF importance sampling alone, which, in the worst case, reduces the efficiency of path guiding so significantly that it can even become inferior to standard path tracing. We therefore present a guided directional sampling approach based on *resampling importance sampling* (RIS) [Talbot et al. 2005] that also integrates defensive sampling [Hesterberg 1995]. Our resulting guided sampling strategy is robust, avoids being worse than BSDF importance sampling, and can even achieve similar sampling quality as full product sampling [Herholz et al. 2016].

3.2.3 Path Guiding and Russian Roulette. The work of Vorba et al. [2014] highlighted that using traditional Russian Roulette (RR), as presented by Arvo and Kirk [1990], with path guiding leads to early path terminations, which significantly impairs its efficiency. In this section, we analyze the reason for this behavior and present a simple but effective solution to retain the intended behavior of traditional RR when using path guiding. In addition, we will also discuss how to efficiently implement more advanced RR techniques, such as *adjoint-driven RR*, using the features of OpenPGL.

3.2.4 Combination with Next-Event Estimation. Direct light sampling, a.k.a. next-event estimation (NEE), is an essential component of any product renderer to reduce variance (i.e., noise) when rendering complex scenes with many light sources. Unfortunately, the combination and interplay of NEE with path guiding, and many path guiding research works, even disable NEE during their evaluation. Here, we will discuss the challenges of combining NEE with path guiding and present a solution that, although primarily based on intuition rather than a theoretical background, works well in practice.

3.2.5 Scattering Event Types and Light Path Expressions. The labeling of scattering event types at each path vertex and the use of light path expressions to categorize different light transport contribution types are crucial features in production rendering. We are therefore going to present our solution to robustly estimate scattering event types after sampling a scattering direction using path guiding. The resulting statistics of the scattering event types match those from traditional BSDF sampling, even for complex multi-closure BSDF models.

3.2.6 Improving Robustness via Roughening. The robustness of the training of guiding structures, especially when facing challenging light transport setups, can be improved by using path space regularization (i.e., roughening) techniques during rendering. We will demonstrate that even a naive version, which is simple to implement and yields only minimal visual differences, of *optimized path space regularization* [Weier et al. 2021] already leads to a significant improvement in quality and robustness.

3.2.7 Additional Integration Challenges and Advice. This section concludes by discussing some additional challenges we encountered when integrating path guiding into various production and research renderers. It ends with a list of advice on how to start and step-by-step progress when planning to integrate path guiding into a renderer. A receipt that has proven to be useful multiple times.

3.3 Path Guiding in Hyperion: A Case Study

In this section, we present our approach to implementing path guiding into Disney's Hyperion Renderer [Burley et al. 2018]. The first part of the section showcases the difficulty of integrating path guiding into a wavefront-based renderer, which poses a set of challenges that do not exist in a depth-first renderer (Sec. 3.3.1).

The next part of this section focuses on debugging a path guiding implementation. We demonstrate the debugging tools we used to assess the correctness of our path guiding implementation and explain why we recommend the use of visual debugging tools to anyone implementing path guiding into a renderer (Sec. 3.3.2).

In the last part, we present the learnings and results from our production tests with path guiding. In particular, we show challenges we encountered in our production tests that are unique to production settings. We will give examples of how we assess path guiding's performance in a production context, which includes considerations often overlooked in research contexts (Sec. 3.3.3).

3.3.1 Integrating Path Guiding into a Wavefront Renderer. Integrating path guiding algorithms into a breadth-first wavefront renderer presents a different set of challenges compared to the integration into a depth-first renderer. Integrating path guiding into a depth-first renderer is a well-documented process for which many reference implementations exist. Conversely, integrating path guiding into a wavefront renderer has received less attention in the research literature; however, it is an essential problem for production rendering, where wavefront architectures are more commonly found. We present our approach to integrating OpenPGL into the wavefront architecture of Disney's Hyperion Renderer [Burley et al. 2018]; our new integration supersedes our previous implementation [Müller 2019] of Practical Path Guiding [Müller et al. 2017]. In particular, we will discuss how we record radiance for training guiding caches, which requires extensive bookkeeping. Recording radiance along entire paths is difficult in wavefront path tracers, where the whole path history is not necessarily always available at each bounce. We present how we recently reworked our training data generation implementation for path guiding in Hyperion to solve this problem; our solution simplifies the recording of radiance while maintaining unbiased results.

3.3.2 Debugging Path Guiding. We show our path guiding visualizer tool that was crucial to understanding the effects path guiding had on our renders, and helped us find bugs in our implementation. Moreover, we discuss practical challenges in debugging radiance recording. A major challenge to debugging inputs to path guiding comes from the fact that incorrect inputs to path guiding can often still produce useful results. Thus, incorrect inputs can be difficult to detect only by looking at the path guiding outputs. We present an alternative debugging output that visualizes the radiance from the recorded paths directly, by splatting the recorded paths to a frame buffer. This approach is both intuitive and simple to evaluate; a correct result should simply match the beauty rendering. By combining this result with our aforementioned path guiding visualization tool, we get two independent data points that help us gain confidence in our results.

3.3.3 Deploying Path Guiding in Production. We discuss our findings from testing our path guiding implementation on production scenes from recent shows such as Disney's "Moana 2" and "Strange World". In this section, we highlight how a production rendering context differs from a research context. We show illustrative technical examples we encountered during production testing in Hyperion to showcase aspects that can affect path guiding performance in a production environment. We discuss how extremely dense volumes in a production shot motivated an optimization in our radiance recording strategy to improve recording radiance at later bounces. Moreover, we discuss the technical details necessary to employ a path guiding iteration schedule in Hyperion. We also discuss evaluation metrics for path guiding in a production rendering context, which are often overlooked in research contexts. And lastly, we will show the results of using path guiding on our production scenes.

3.4 Efficient Rendering of Caustics: Photon Guiding at Corona

In this part of the course, we will discuss an algorithm for rendering caustics used in Chaos's Corona renderer[1]. During the development of our caustics solver, we have experimented with various algorithms for rendering caustics, including path guiding. While path guiding is very effective at handling diffuse indirect illumination, we have found out that it is not as efficient at handling caustics compared to photon-based algorithms. Therefore, Corona's caustics solver uses photons to render caustics. More specifically, we utilize a lightweight version of the vertex connection and merging algorithm (VCM) [Georgiev et al. 2012] together with efficient guiding of the photons. Although part of this algorithm has already been presented [Šik and Křivánek 2019], we have made several improvements to our caustics solver, which we will discuss in this course.

[1] https://corona-renderer.com

3.4.1 Stratification over Light Sources. Our photon guiding algorithm is based on Markov chain Monte Carlo [Metropolis et al. 1953]. While this sampling method is effective at guiding photons where they are most needed, it can often over-sample or under-sample parts of the scene. This is especially true in scenes with many light sources, where caustics from one light source may converge much faster than caustics from other light sources. To mitigate this issue, we utilize multiple Markov chains in our caustics solver. This allows us to significantly improve the stratification of generated photons over all the light sources.

3.4.2 Photon Guiding. Another weak point of Markov chain Monte Carlo is that it relies on uninformed uniform sampling (so-called *large steps*) to frequently visit all the important parts of the scene. To alleviate this problem in our caustics solver, we utilize a combination of Markov chain Monte Carlo with adaptive sampling of photon emission from light sources. This allows our algorithm to frequently sample all important parts of a scene, and thus improves the speed and uniformness of caustics convergence.

3.4.3 Photon Count Automation. A difficult part of photon-based algorithms is determining the number of photons in each rendering iteration. This is especially the case in scenes that have motion blur and/or dispersion, where many photons are rejected during merging. The number of photons is usually left as a user parameter, and, therefore, it is up to the user to find a good enough value. This is not the case for our caustics solver since we have developed an automatic method for determining a sufficient number of photons generated in each rendering iteration.

3.4.4 Volumetric Caustics. The initial release of Corona's caustics solver could resolve only caustics formed on surfaces. To broaden the options available to our users, we have added the handling of volumetric caustics to our solver. We will discuss which estimator we have selected for computing the volumetric caustics and how we guide photons in volumes to important parts of the scene.

4 Learning Objectives and Outcomes

At the end of this course, attendees from the production and rendering research community gained additional knowledge about challenges and how to solve them when integrating path guiding into complex rendering systems, such as those occurring in production. We hope that this knowledge will help not only developers of production renderers but also researchers to make their integrations more robust. In addition, we hope that the challenges (e.g., efficient training data generation and storage, efficiently guiding caustics, or incorporating and considering NEE) we have pointed out will inspire researchers to develop new algorithms and solutions to further increase the usefulness and adoption of path guiding.

References

James Arvo and David Kirk. 1990. Particle Transport and Image Synthesis. *SIGGRAPH Comput. Graph.* 24, 4 (Sept. 1990), 63–66. doi:10.1145/97880.97886

Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Dan Teece. 2018. The Design and Evolution of Disney's Hyperion Renderer. *ACM Transactions on Graphics* 37, 3, Article 33 (Aug. 2018). doi:10.1145/3182159

Luca Fascione, Johannes Hanika, Daniel Heckenberg, Christopher Kulla, Marc Droske, and Jorge Schwarzhaupt. 2019. Path Tracing in Production: Part 1: Modern Path Tracing. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*.

Association for Computing Machinery, New York, NY, USA, Article 19, 113 pages. doi:10.1145/3305366.3328079

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light Transport Simulation with Vertex Connection and Merging. *ACM Trans. Graph.* 31, 6, Article 192 (Nov. 2012), 10 pages. doi:10.1145/2366145.2366211

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. 2016. Product importance sampling for light transport path guiding. *Computer Graphics Forum* 35, 4 (2016), 67–77.

Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik P. A. Lensch, and Jaroslav Křivánek. 2019. Volume Path Guiding Based on Zero-Variance Random Walk Theory. *ACM Trans. Graph.* 38, 3, Article 25 (jun 2019), 19 pages. doi:10.1145/3230635

Tim Hesterberg. 1995. Weighted Average Importance Sampling and Defensive Mixture Distributions. *Technometrics* 37, 2 (1995), 185–194. http://www.jstor.org/stable/1269620

Wenzel Jakob, Andrea Weidlich, Andrew Beddini, Rob Pieké, Hanzhi Tang, Luca Fascione, and Johannes Hanika. 2019. Path Tracing in Production: Part 2: Making Movies. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 20, 41 pages. doi:10.1145/3305366.3328085

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092.

Thomas Müller. 2019. "Practical Path Guiding" in Production. In *ACM SIGGRAPH Courses: Path Guiding in Production, Chapter 10* (Los Angeles, California). ACM, New York, NY, USA, 18:35–18:48. doi:10.1145/3305366.3328091

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical path guiding for efficient light-transport simulation. *Computer Graphics Forum* 36, 4 (2017), 91–100.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 36, 4 (June 2017), 91–100. doi:10.111/cgf.13227

Alexander Rath, Pascal Grittmann, Sebastian Herholz, Philippe Weier, and Philipp Slusallek. 2022. EARS: Efficiency-Aware Russian Roulette and Splitting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2022)* 41, 4, Article 81 (jul 2022), 14 pages. doi:10.1145/3528223.3530168

Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust Fitting of Parallax-Aware Mixtures for Path Guiding. *ACM Trans. Graph.* 39, 4, Article 147 (aug 2020), 15 pages. doi:10.1145/3386569.3392421

Martin Šik and Jaroslav Křivánek. 2019. Implementing One-Click Caustics in Corona Renderer. In *Eurographics Symposium on Rendering - DL-only and Industry Track*, Tamy Boubekeur and Pradeep Sen (Eds.). The Eurographics Association, 61–67. doi:10.2312/sr.20191221

Justin F. Talbot, David Cline, and Parris Egbert. 2005. Importance resampling for global illumination. In *Proc. EGSR.* 139–46.

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering, Vol. 29. 419–428. doi:10/d7b6n4

Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. 2019. Path Guiding in Production. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 18, 77 pages. doi:10.1145/3305366.3328091

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-Line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. Graph.* 33, 4, Article 101 (jul 2014), 11 pages. doi:10.1145/2601097.2601203

Jiří Vorba and Jaroslav Křivánek. 2016. Adjoint-driven Russian roulette and splitting in light transport simulation. *ACM Trans. Graph.* 35, 4 (2016), 42:1–42:11.

Philippe Weier, Marc Droske, Johannes Hanika, Andrea Weidlich, and Jiří Vorba. 2021. Optimised Path Space Regularisation. *Computer Graphics Forum* 40, 4 (2021), 139–151. doi:10.1111/cgf.14347 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14347

Kehan Xu, Sebastian Herholz, Marco Manzi, Marios Papas, and Markus Gross. 2024. Volume Scattering Probability Guiding. *ACM Trans. Graph.* 43, 6, Article 184 (Nov. 2024), 17 pages. doi:10.1145/3687982