Supplementary Material: Photon-Driven Neural Reconstruction for Path Guiding

SHILIN ZHU, University of California San Diego, USA
ZEXIANG XU, Adobe Research, USA
TIANCHENG SUN, University of California San Diego, USA
ALEXANDR KUZNETSOV, University of California San Diego, USA
MARK MEYER, Pixar Animation Studios, USA
HENRIK WANN JENSEN, University of California San Diego and Luxion, USA
HAO SU, University of California San Diego, USA
RAVI RAMAMOORTHI, University of California San Diego, USA

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: Global Illumination, Path Guiding, Ray Tracing, Sampling and Reconstruction, Neural Rendering

ACM Reference Format:
Shilin Zhu, Zexiang Xu, Tiancheng Sun, Alexandr Kuznetsov, Mark Meyer, Henrik Wann Jensen, Hao Su, and Ravi Ramamoorthi. 2021. Supplementary Material: Photon-Driven Neural Reconstruction for Path Guiding. *ACM Trans. Graph.* 1, 1, Article 1 (January 2021), 5 pages. https://doi.org/10.1145/3476828

1 SUPPLEMENTARY MATERIAL

In the supplementary material, we provide additional experimental results and visualizations, as well as discussions on potential extensions of the proposed framework. Although not being emphasized in the main paper, these additional studies and evaluations are also important in the design of a full-fledged path guiding supported renderer in practice.

1.1 Monte-Carlo Denoising

Monte-Carlo (MC) rendering algorithms such as path tracing are known to suffer from the slow convergence problem when producing noise-free images [Kajiya 1986; Lafortune 1996]. In recent years, MC denoising has become a successful approach to reduce pixel variance, especially those based on deep neural networks [Bako et al. 2017; Chaitanya et al. 2017; Vogels et al. 2018]. Although MC denoising is a biased operation, it significantly increases the visual quality by removing the last-mile residual pixel noise. Therefore, we apply the learning based denoising technique to the rendered results from path-guiding methods, which can be a practical way to use the method. In particular, we use the built-in denoiser from Nvidia OptiX 6.5 [Parker et al. 2010] to process the output of all the approaches, as shown in Fig. 1. In fact, although the denoiser can effectively reduce rMSE in most cases, such denoising is reasonable only when the rendered image has an acceptable level of MC noise; otherwise the results can appear blurry with missing details (e.g., the caustics area in the Egg scene) or distorted (e.g., the center region in the Hotel scene), which is unacceptable in either case for offline photo-realistic rendering.

1.2 Next-Event Estimation

In the default experimental setting, we turn off the next-event estimation (NEE) to clearly compare the effects from path guiding (similar to [Müller et al. 2017] and [Vorba et al. 2014]), though in practice NEE can be effective and should be always turned on. In particular, NEE can help reducing the variance by easing the light discovery and improving the sampling map quality. To study how NEE affects the path guiding results, we turn on the standard NEE in Fig. 2. Results show that whether NEE is useful or not depends mostly on the light setup. For the Bathroom scene, the glass bulb fixture and staggered window blinds make the direct connection difficult to succeed; for the Veach Ajar scene, NEE connection fails and succeeds from time to time depending on the local light visibility. However, we believe more advanced NEE techniques such as the light hierarchy [Walter et al. 2005] should be able to sample the direct illumination more efficiently over the standard NEE, and it is interesting to explore how different path guiding approaches interact with state-of-the-art direct light sampling in future research.

1.3 One-Sample MIS

In the main paper, we demonstrate a new heuristic pipeline for estimating the mixture coefficient α in one-sample MIS of BSDF sampling and guiding. This is useful in some cases, as shown in Fig. 3. In the Pool scene, the BSDF sampled directions from the floor surface often fail to reach the sunlight, thus contributing less energy to the final pixels. Therefore, our heuristic encourages sending more guiding samples in those regions based on the collected statistics

Authors' addresses: Shilin Zhu, University of California San Diego, USA, shz338@eng.ucsd.edu; Zexiang Xu, Adobe Research, USA, zexu@adobe.com; Tiancheng Sun, University of California San Diego, USA, tis037@eng.ucsd.edu; Alexandr Kuznetsov, University of California San Diego, USA, a1kuznet@eng.ucsd.edu; Mark Meyer, Pixar Animation Studios, USA, mmeyer@pixar.com; Henrik Wann Jensen, University of California San Diego and Luxion, USA, henrik@cs.ucsd.edu; Hao Su, University of California San Diego, USA, haosu@eng.ucsd.edu; Ravi Ramamoorthi, University of California San Diego, USA, ravir@cs.ucsd.edu.

0730-0301/2021/1-ART1 $15.00
https://doi.org/10.1145/3476828

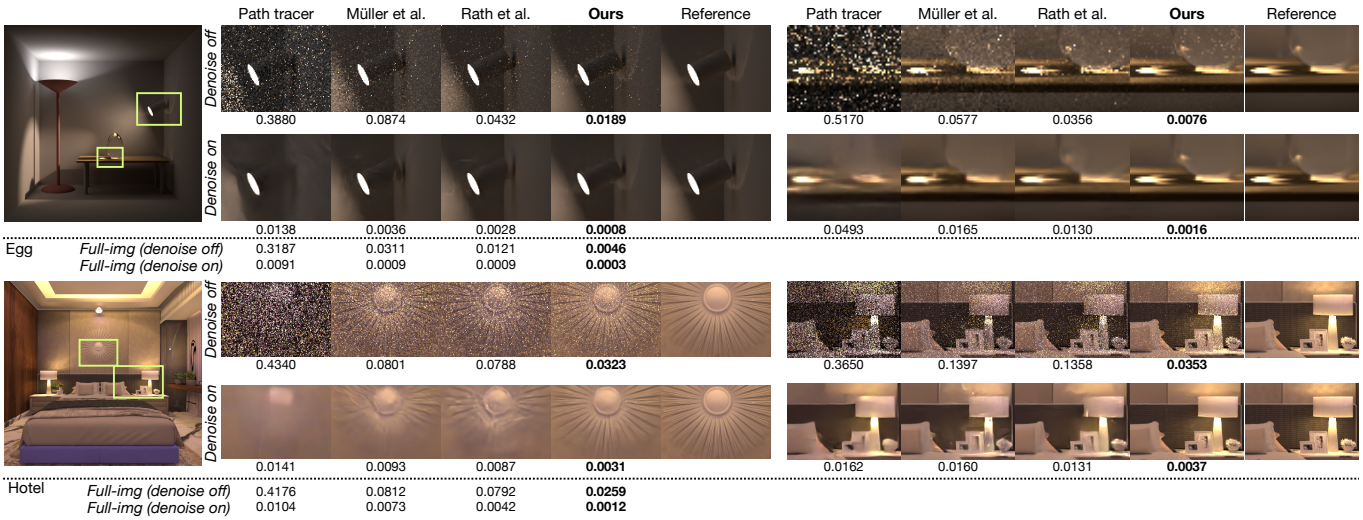

Fig. 1. Monte-Carlo denoising on the path guiding rendering results. We use the deep learning based image-space denoiser in Nvidia OptiX 6.5. In general, the denoiser in-paints the missing regions in between pixels and filters out the high-frequency MC noise. The denoised images rendered with our method are more acceptable without severe blurring or distortion.

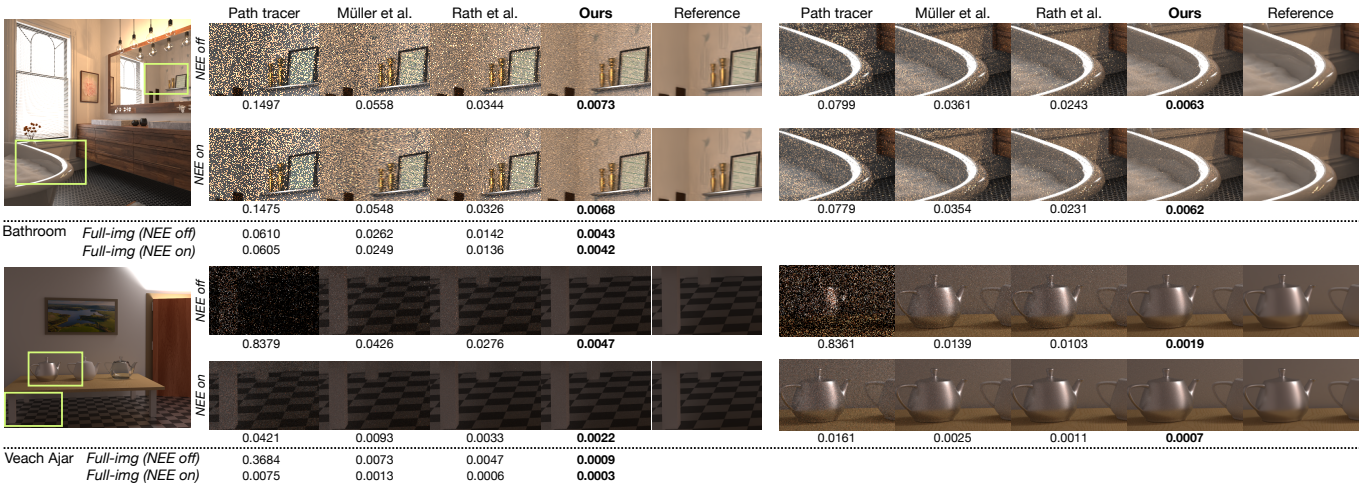

Fig. 2. The effect of the standard next-event estimation (NEE) on the path guiding rendering results. The comparison is equal-time for each row. Results show that NEE effectively improves the results in some cases while it is not very successful in some other cases, depending on the sampling map quality in path guiding and the light visibility levels at different scene locations.

of previously traced path samples. For glossy surfaces such as the metal armrest in this scene, we send more BSDF samples since many guided directions have small or zero BSDF values. Although the proposed heuristic is still sub-optimal, it is straightforward to implement and does not introduce too much overhead. In the future, we believe our heuristic can provide a good starting point to initialize other methods that try to optimize α during rendering [Müller 2019; Rath et al. 2020].

1.4 Adaptive Hierarchical Grid

We also investigate the presented spatial structure - the hierarchical grid. We show result images rendered with different initial resolutions for the adaptive grid in Fig. 4. We also show corresponding results using only an uniform grid without the adaptive partitioning inside voxels. Note that without the adaptivity, rendering quality varies drastically with different resolutions, since a low-resolution grid lacks expressibility of complex local light fields and a high-resolution grid does not cover sufficient photons in some voxels. In the extreme case, under-partitioning and over-partitioning can both reduce the performance. On the contrary, the hierarchical grid is

Scene/Method	PT	[Bako et al. 2019]	[Vorba et al. 2014]	[Müller et al. 2017]	[Rath et al. 2020]	[Ruppert et al. 2020]	**Ours**
Metric	SOTA SSIM ↑						
Caustics Egg	0.1017	0.1824	0.3472	0.4581	0.7006	0.7866	0.8242
Veach Ajar	0.0474	0.0898	0.4579	0.5455	0.6325	0.6921	0.8572
Bathroom	0.4481	0.4725	0.5472	0.5260	0.5924	0.8195	0.7427
Hotel	0.0695	0.1155	0.0914	0.2665	0.2801	0.4634	0.4378
Staircase	0.4810	0.4957	0.6513	0.7337	0.8626	0.7625	0.8951
Living Room	0.1360	0.1719	0.4734	0.2960	0.3327	0.6065	0.6576
Spaceship	0.5610	0.7476	0.8611	0.7452	0.8124	0.8922	0.8793
Classroom	0.2789	0.3037	0.5756	0.6352	0.7681	0.7505	0.8234
Wild Creek	0.3023	0.3734	0.4890	0.4852	0.5386	0.6130	0.6222
Kitchen	0.3898	0.4173	0.4655	0.6753	0.7873	0.7383	0.8168
Pool	0.2264	0.4595	0.8551	0.8598	0.9364	0.9068	0.9510

Table 1. Quantitative comparison. We compare our approach to [Bako et al. 2019; Vorba et al. 2014; Müller et al. 2017; Rath et al. 2020; Ruppert et al. 2020] with equal rendering time. We show the corresponding SSIMs of the rendered full images of the testing scenes. Red, orange, and yellow denote the best, the second-best, and the third-best method in terms of SSIM (higher is better).

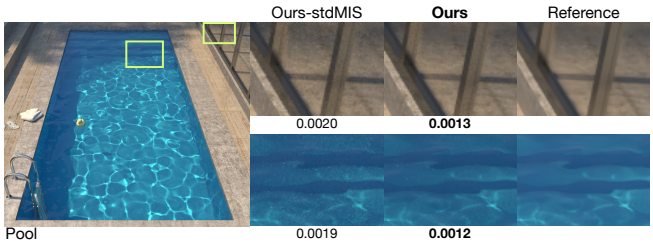

Fig. 3. Our proposed heuristic one-sample MIS scheme outperforms the default mixture coefficient $\alpha = 0.5$ especially when BSDF sampling and guiding sampling have very disparate contributions to the final image.

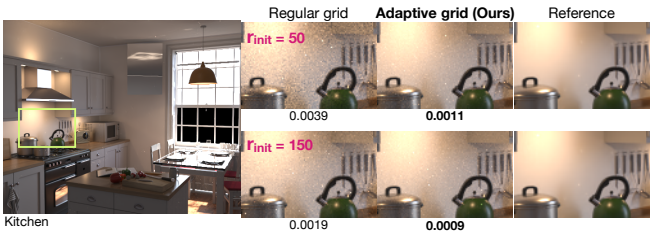

Fig. 4. The effect of initial grid resolution and our hierarchical spatial partitioning framework. Ideally, the voxel size should be small enough to reflect the locality of the incident radiance, and large enough to cover sufficient number of photons for sampling map reconstruction.

more stable and insensitive with different initial resolutions, since it can adaptively subdivide the initial grid to a desired density locally.

1.5 Additional Statistics

We include the rendering time of each scene in the experimental section of the main paper. In Table. 2, we further present the timing breakdown according to the proposed path guiding algorithm.

Component	Path (%)	Photon (%)	Neural rec (%)	Path (%)	Time (min)
Algorithm 1	LN 3∼24	LN 25∼35	LN 36∼38	LN 40	/
Device	CPU	CPU	GPU	CPU	/
Phase	iterative process (when $t < T$)			final	/
Caustics Egg	13.91	21.83	8.36	55.88	4.0
Veach Ajar	14.58	21.08	5.76	58.56	18.0
Bathroom	15.42	11.86	9.77	62.93	5.0
Hotel	15.05	18.58	5.89	60.46	20.0
Staircase	15.79	15.77	5.01	63.41	11.0
Living Room	16.42	11.73	5.89	65.94	11.0
Spaceship	16.49	9.20	8.06	66.23	3.0
Classroom	15.33	16.68	6.39	61.57	13.0
Wild Creek	17.05	8.41	6.02	68.50	10.0
Kitchen	14.35	19.06	8.94	57.63	4.0
Pool	16.78	8.22	7.57	67.40	4.0

Table 2. Running time breakdown. Percentages of running time of different components in the proposed system are shown for different testing scenes. The total rendering time for each scene is also listed in the rightmost column. The time distribution varies depending on the scene complexity and light setup.

Thanks to the effective sampling map reconstruction that is combined with iterative sample tracing and rendering, our approach can save more time for the guided path tracing over distribution learning.

For quantitative evaluation, we use the rMSE metric in the main paper. There exists other metrics that are inspired by human visual perception such as the Structural Similarity Index (SSIM [Wang et al. 2004]). The results on SSIM are presented in Table. 1, which are generally consistent with rMSE results but rankings are slightly different for some scenes. This is because SSIM focuses on multiple factors such as brightness, contrast, and structure while rMSE only

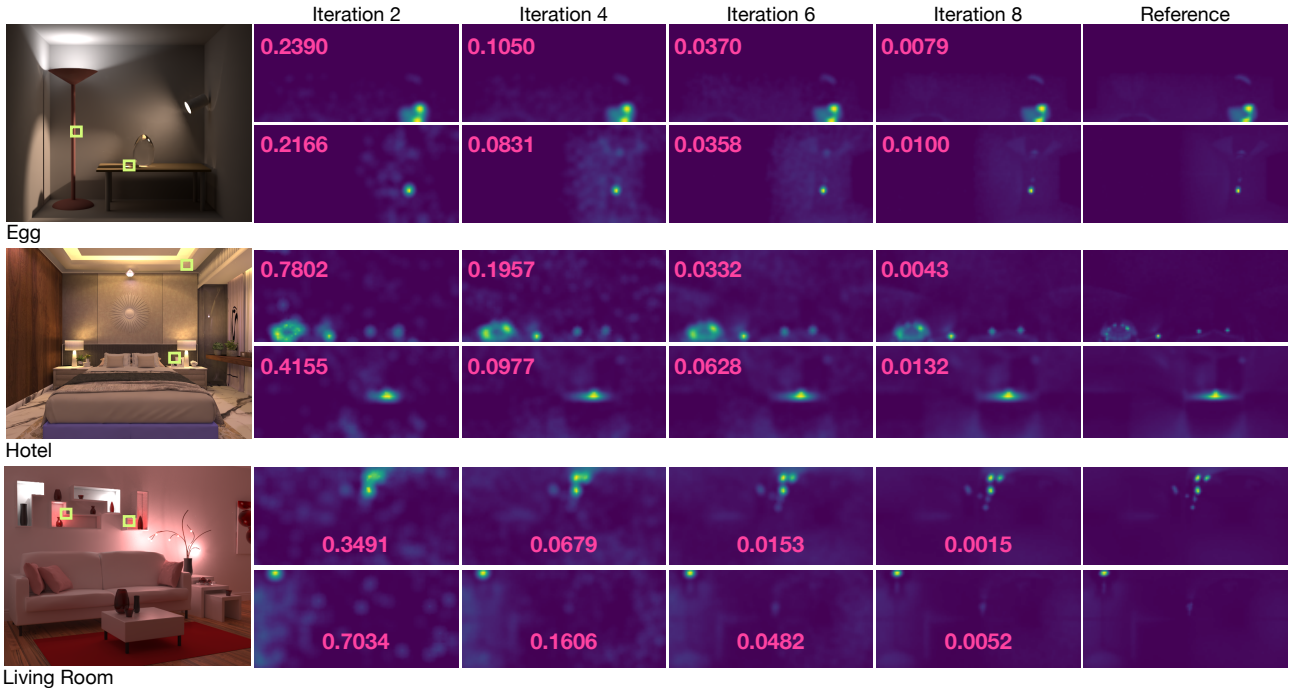

Fig. 5. Reconstructed sampling map visualization through multiple learning iterations. The reconstructed distributions lead to better path space exploration at the beginning and more accurate representations of the incident radiance in the subsequent iterations.

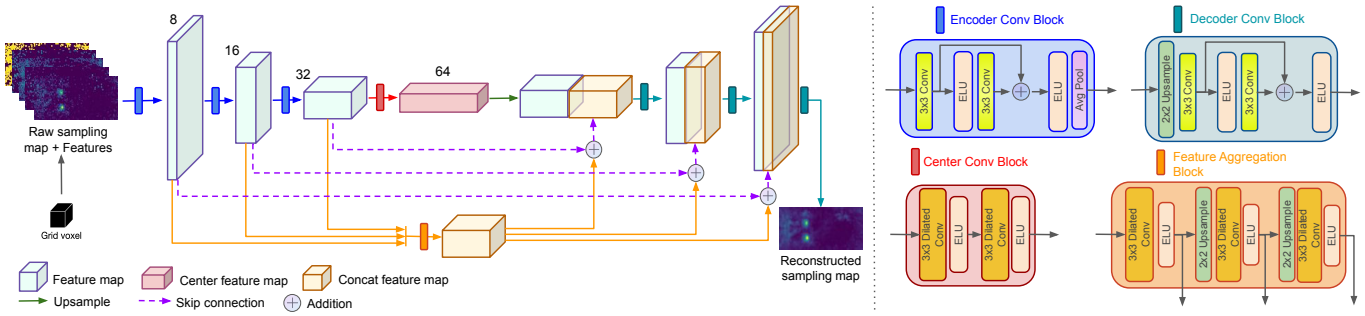

Fig. 6. The neural network architecture for sampling map reconstruction. We use a compact autoencoder with light-weight masked convolutions [Liu et al. 2018; Yi et al. 2020] and ELU [Clevert et al. 2015] activation function which can extract high-level deep features from the input power map and output a smooth and detailed sampling map. The bottleneck layers use dilated convolutions [Iizuka et al. 2017] to further expand the size of the receptive fields.

reflects the brightness difference. We leave other advanced metrics (e.g., FLIP [Andersson et al. 2020]) as future work.

1.6 Neural Sampling Map Reconstruction

Some reconstructed sampling maps are visualized in Fig. 5. After pre-training on an offline dataset, our neural networks can progressively reconstruct higher-quality sampling maps with more accumulated photon power through iterations on new scenes. Unlike previous Monte-Carlo denoising neural networks [Chaitanya et al. 2017; Bako et al. 2017; Vogels et al. 2018] which only process the input image once, our reconstruction is getting better and closer to the reference sampling maps over time. More specifically, the pre-trained neural network reconstructs more aggressively in the early iterations to encourage more exploration of the directional sampling space; in the later iterations the reconstructed maps contain more accurate details of the incident radiance distribution due to a higher level of confidence. The detailed network architecture used for such reconstruction in the main paper is presented in Fig. 6. We believe the designed deep network can be further compressed by the state-of-the-art neural compression methods, improved by more advanced neural components (e.g., reinforcement learning [Huo et al. 2020]), and extended to more compact representations (e.g., quadtrees [Müller et al. 2017; Rath et al. 2020], mixtures [Vorba et al. 2014; Ruppert et al. 2020]) in the future (e.g., [Zhu et al. 2021]).

REFERENCES

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D Fairchild. 2020. FLIP: a difference evaluator for alternating images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques (HPG 2020)* 3, 2 (2020).

Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline Deep Importance Sampling for Monte Carlo Path Tracing. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 527–542.

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.* 36, 4 (2017), 97–1.

Chakravarty R Alla Chaitanya, Anton S Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* (2015).

Yuchi Huo, Rui Wang, Ruzahng Zheng, Hualin Xu, Hujun Bao, and Sung-Eui Yoon. 2020. Adaptive Incident Radiance Field Sampling and Reconstruction Using Deep Reinforcement Learning. *ACM Transactions on Graphics (TOG)* 39, 1 (2020), 1–17.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1–14.

James T Kajiya. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 143–150.

Eric Lafortune. 1996. Mathematical models and Monte Carlo algorithms for physically based rendering. *Department of Computer Science, Faculty of Engineering, Katholieke Universiteit Leuven* 20 (1996), 74–79.

Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 85–100.

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

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical path guiding for efficient light-transport simulation. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 91–100.

Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al. 2010. OptiX: a general purpose ray tracing engine. *Acm transactions on graphics (tog)* 29, 4 (2010), 1–13.

Alexander Rath, Pascal Grittmann, Sebastian Herholz, Petr Vévoda, Philipp Slusallek, and Jaroslav Křivánek. 2020. Variance-Aware Path Guiding. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2020)* 39, 4 (July 2020), 151:1–151:12. https://doi.org/10.1145/3386569.3392441

Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust Fitting of Parallax-Aware Mixtures for Path Guiding. *ACM Transactions on Graphics (TOG)* (2020).

Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–15.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P Greenberg. 2005. Lightcuts: a scalable approach to illumination. In *ACM SIGGRAPH 2005 Papers*. 1098–1107.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. 2020. Contextual Residual Aggregation for Ultra High-Resolution Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7508–7517.

Shilin Zhu, Zexiang Xu, Tiancheng Sun, Alexandr Kuznetsov, Mark Meyer, Henrik Jensen, Hao Su, and Ravi Ramamoorthi. 2021. Hierarchical Neural Reconstruction for Path Guiding Using Hybrid Path and Photon Samples. In *ACM Transactions on Graphics*.