Supplementary Material: Hierarchical Neural Reconstruction for Path Guiding Using Hybrid Path and Photon Samples

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**.

ACM Reference Format:
Shilin Zhu, Zexiang Xu, Tiancheng Sun, Alexandr Kuznetsov, Mark Meyer, Henrik Wann Jensen, Hao Su, and Ravi Ramamoorthi. 2021. Supplementary Material: Hierarchical Neural Reconstruction for Path Guiding Using Hybrid Path and Photon Samples. *ACM Trans. Graph.* 40, 4, Article 35 (August 2021), 4 pages. https://doi.org/10.1145/3450626.3459810

1 SUPPLEMENTARY MATERIAL

In the supplementary material, we present some additional experimental results. These evaluations are still important when we design a full-fledged path guiding system for arbitrary scenes in practice.

1.1 Next Event Estimation (NEE)

Similar to Müller et al. [2017] and Vorba et al. [2014], the default setting in our main paper turns off the next event estimation (NEE) to more clearly compare the effectiveness of different path guiding algorithms. However, NEE has already been a prevalent module in modern renderers, and often it is able to ease the search of hidden or small light sources. Besides, NEE can sometimes also help in accelerating the path guiding algorithms by providing initial information on potential light locations.

Therefore, we turn on the standard NEE and see how it affects the performance of each path guiding algorithm. In Fig. 1, we test on two scenes used in our main paper with different light setups. For the Racing Car scene, because there are multiple tiny lights that are hidden behind objects with complex geometry, NEE can be useful in the beginning to quickly connect to one of the light sources. In general, NEE is more useful for a path sample based guiding approach (e.g., [Müller et al. 2017; Rath et al. 2020]) than a photon-based approach (e.g., [Zhu et al. 2020]), since photons already provided sufficient information about visible incident lights from a surface point. However, for the Bathroom scene, the glass bulb fixture and staggered window blinds reduce the chance of successful direct NEE connection to any light source. Therefore, most methods cannot benefit much from NEE in this special case. In general, whether NEE is contributing or not depends on the local light visibility as well as the specific NEE technique. More advanced direct sampling methods such as the light hierarchy [Walter et al. 2005] are able to sample the direct illumination more efficiently over the standard NEE, and it is interesting to study how path guiding interacts with those methods in future research.

In fact, although our method benefits less from NEE compared to other baselines, we can still achieve better performance (although the gain is reduced), thanks to our high-quality sampling distribution reconstruction framework which captures both direct and indirect illumination. In practice, it is often a good choice to request NEE, but the decision is also affected by the total timing or sampling budget in specific applications.

1.2 Monte-Carlo (MC) Denoising

The standard Monte-Carlo path tracing algorithm has the slow convergence problem, which requires running the algorithm for a long time to produce noise-free images [Kajiya 1986; Lafortune 1996]. In the past few years, MC denoising has been researched to reduce the last-mile residual pixel variance by filtering over the rendered image. Among all of these proposed methods, deep learning based denoising for path tracing results using a CNN has become a very successful approach [Bako et al. 2017; Chaitanya et al. 2017; Vogels et al. 2018]. This greatly increases the visual quality of the image with a small risk of bias introduced to the results.

A practical and straightforward way to combine path guiding with denoising is to apply the neural denoiser on path guiding results. More specifically, we use the built-in neural denoiser from Nvidia OptiX 6.5 [Parker et al. 2010], which runs on the GPU, on all the

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.

0730-0301/2021/8-ART35
https://doi.org/10.1145/3450626.3459810

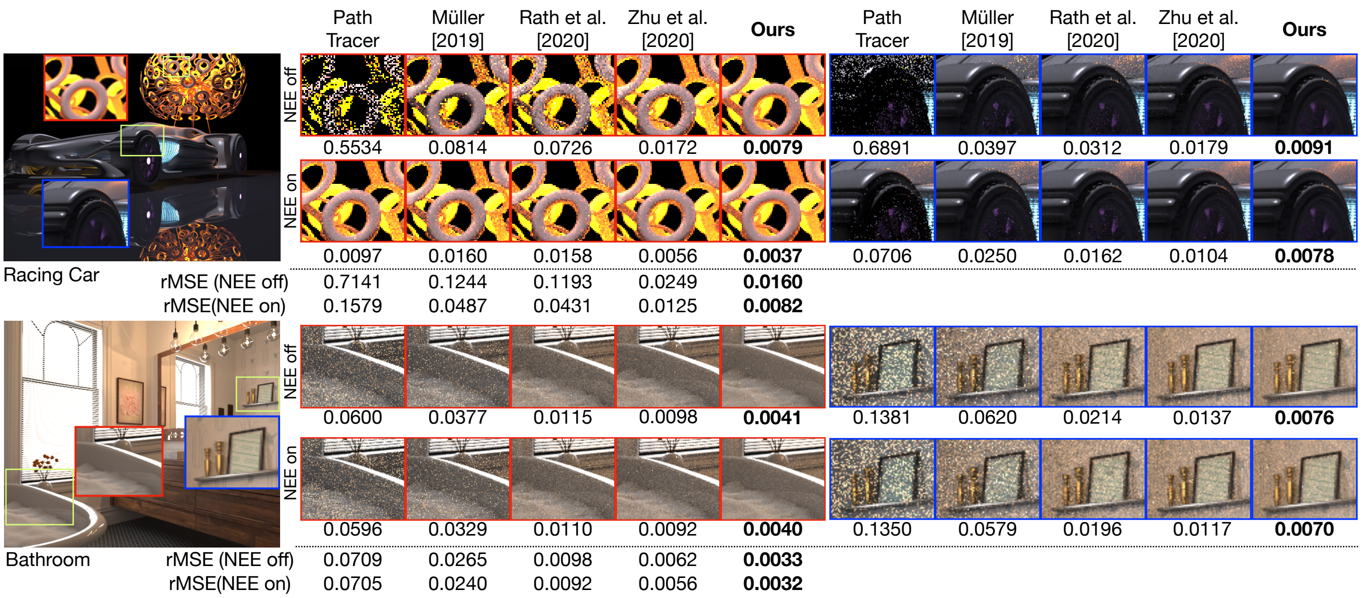

Fig. 1. The effect of the standard next event estimation (NEE) on the results rendered equal-time for each row. NEE can improve the results in some scenes when it is hard for path tracing to find lights in the beginning. However, it is not very efficient in some other light transport scenarios when most direct connections fail.

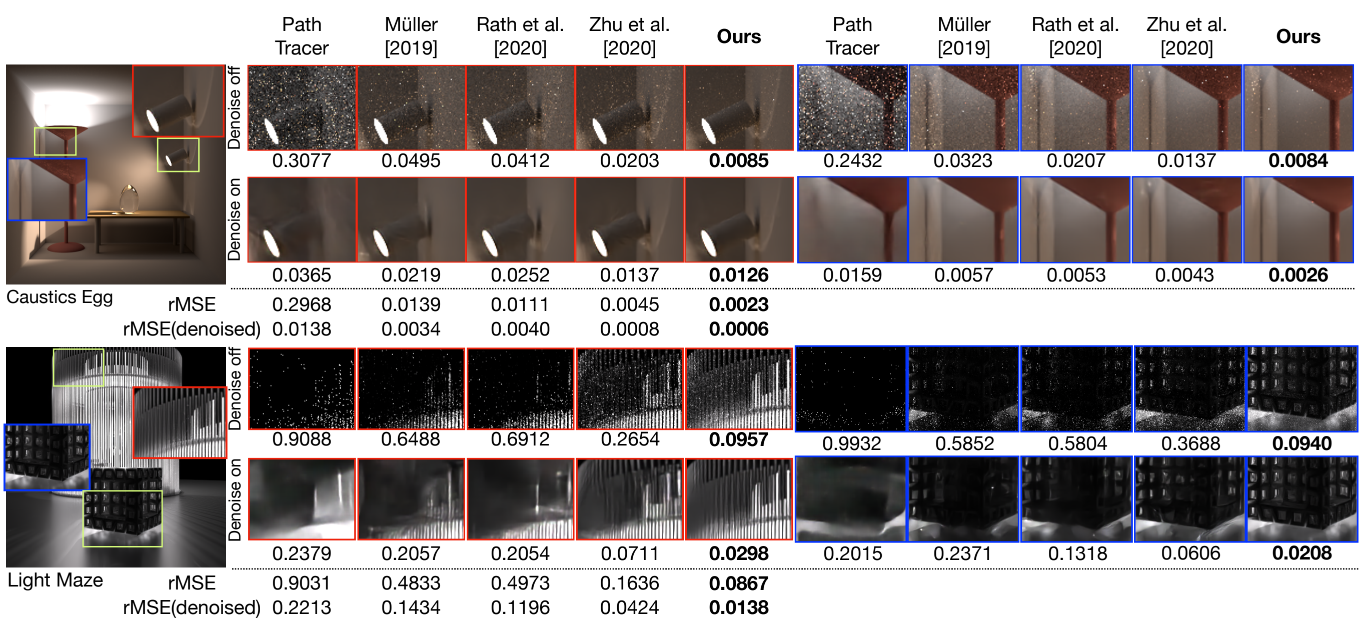

Fig. 2. The effect of Monte-Carlo denoising on path guiding results. The neural network based MC denoiser in Nvidia OptiX 6.5 is used here to process all the images. The denoiser fills the black pixels and smooths out the high-frequency noise. Thanks to our high-quality reconstructed sampling distributions, our initial rendered results contain less noise and are more acceptable after denoising without severe blur or distortion.

result images from our method and baselines. Results in Fig. 2 show that denoising can effectively smooth the results and remove most of the high-frequency artifacts. Moreover, our method still achieves better visual quality even after MC denoising. This is because the denoising performs best when the original rendered image already had only a small amount of high-frequency noise, otherwise the denoised image can be distorted or over-blurry (e.g., the wall in the Caustics Egg scene and the cube in the Light Maze scene).

1.3 Additional Ablation Results

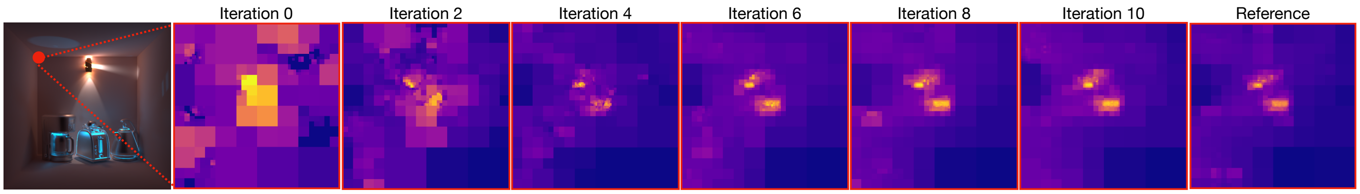

Fig. 3. Visualization of the reconstructed quadtree over iterations. In general, our reconstruction is more conservative in the beginning thus the quadtree is shallower. After more samples are collected over time, our reconstructed quadtree becomes more accurate and fine-grained, which eventually can converge to the target sampling distribution.

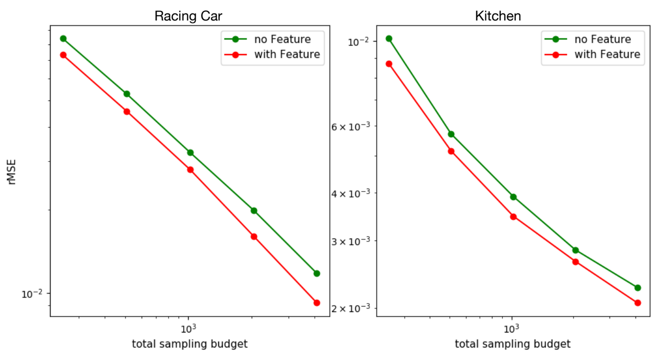

Fig. 4. The effectiveness of sample features introduced in the main paper. We show two curves with and without including the auxiliary features as the neural network input, and results show a slight improvement when features are considered.

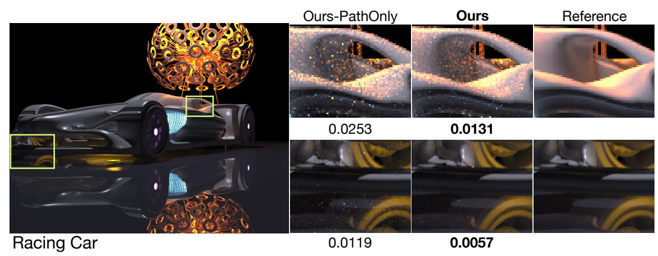

Fig. 5. The effectiveness of using hybrid samples (path samples and photons). Because of the complex visibility of the light sources, using photons is more beneficial than path samples in this scene. The presented insets clearly show the advantage of our hybrid sample framework.

We also try experimenting with more different setups in our proposed framework. To justify the use of the sample features as illustrated in the main paper, we have trained another version of the network that only takes the sample value (i.e., irradiance of path samples, or power of photons) as the input. Results in Fig. 4 show that features can offer some improvements to the final rendered image by enabling higher-quality reconstructed sampling distributions.

Since our approach takes both path samples and photons as input, it is also necessary to study how this decision affects the performance and robustness. In the main paper, we have verified the increased robustness in the extreme light transport conditions. To further justify the necessity of hybrid samples, we train another network with only path sample input (i.e., without photons). The comparison to our full model is shown in Fig. 5. Although "Ours-PathOnly" is also powered by a similar neural network, removing one type of samples from the framework can possibly cause performance to downgrade, especially when the one being neglected is more useful than the one being used.

1.4 Visualization of Quadtrees

In the main paper, we visualize and compare the sampling distribution representations of different methods. In Fig. 3, we further demonstrate how our reconstructed quadtree gets updated and evolved over multiple iterations. We can see the sampling distribution becomes more accurate and detailed with time, which gradually converges to the reference distribution. It also shows that our neural path guiding framework can process inputs of different noise and sparsity levels.

1.5 FLIP Metric

In the main paper, we use rMSE as our primary metric to quantitatively measure the quality of the rendering results. Here, we use a recently proposed metric called FLIP [Andersson et al. 2020] which can closely reflect the human perception of image differences. In addition, FLIP can also output a map that illustrates the error distribution over the scene. In Fig. 6, we show that our method can produce results that have overall less noise within the entire image.

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, 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.

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.

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.

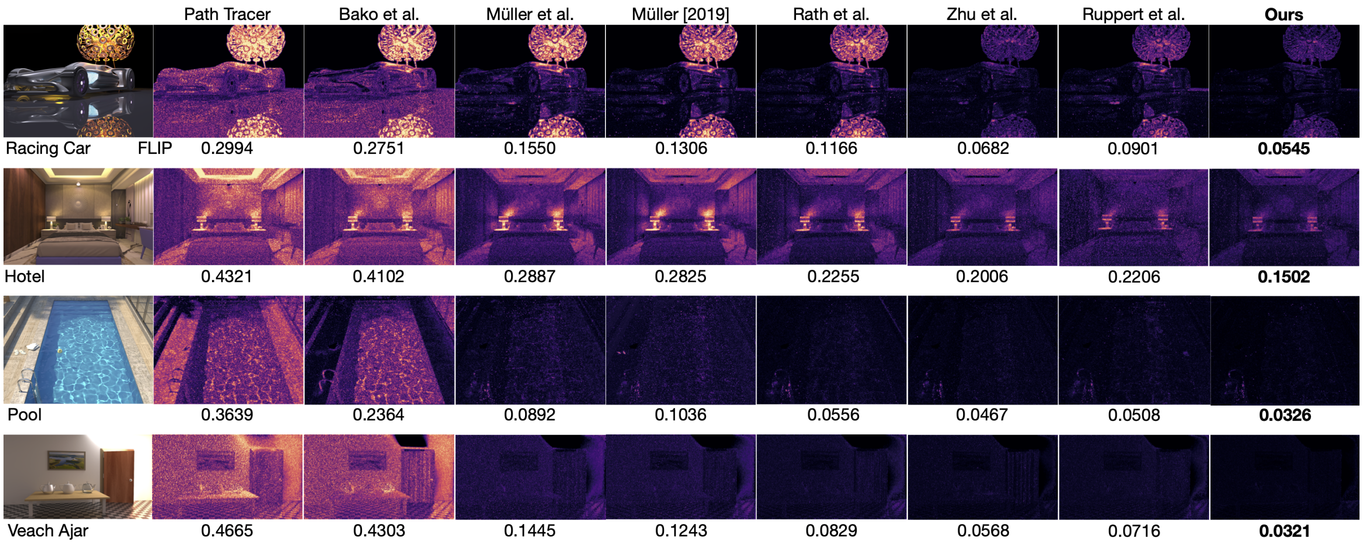

Fig. 6. FLIP error maps on example scenes from the main paper (brighter color means larger error). The number below each image represents the average error of all the pixels.

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

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.

Shilin Zhu, Zexiang Xu, Tiancheng Sun, Alexandr Kuznetsov, Mark Meyer, Henrik Wann Jensen, Hao Su, and Ravi Ramamoorthi. 2020. Photon-Driven Neural Path Guiding. *arXiv preprint arXiv:2010.01775* (2020).