

Path Guiding in Disney’s “Zootopia 2”

Yining Karl Li
Walt Disney Animation Studios
Burbank, USA
karl.li@disneyanimation.com

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

Marco Manzi
DisneyResearch|Studios
Zürich, Switzerland
marco.manzi@disneyresearch.com

Alexander Rath
DisneyResearch|Studios
Zürich, Switzerland
alexander.rath@disneyresearch.com

Brian Green
Walt Disney Animation Studios
Burbank, USA
brian.green@disneyanimation.com

Neelima Karanam
Walt Disney Animation Studios
Burbank, USA
neelima.karanam@disneyanimation.com

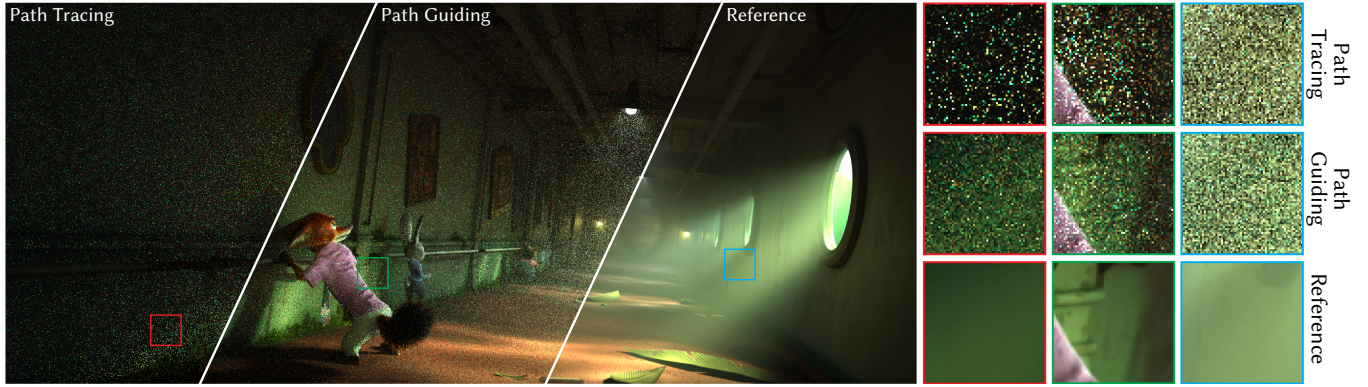


Figure 1: *CORRIDOR* scene from “Zootopia 2”: an indoor scene with complex bounce lighting and illumination from caustics. From left to right: 32 SPP with path tracing, 32 SPP with path guiding, and reference. Our efficiency metric measures that path guiding is 2.16x more efficient than path tracing for this scene. This figure is exposed up two stops from the raw renders for improved readability. © 2025 Disney

Abstract

Walt Disney Animation Studios’ “Zootopia 2” was the first film to make extensive use of Disney’s Hyperion Renderer’s [Burley et al. 2018] second-generation path guiding system, built on top of OpenPGL [Herholz and Dittbrandt 2022]. Jumping from initial implementation to at-scale production usage required us to solve additional practical problems on top of what we previously described in Reichardt et al. [2025]. We introduce two new methods: one for estimating the efficiency improvement path guiding gives over path tracing without the need for multiple long comparison renders, and one for allowing path guided results to match sources of bias common in production rendering introduced by common practical variance-reduction techniques. We also describe improvements to our radiance recorder mechanism. Finally, we will examine how our path guiding system was used on “Zootopia 2”, including artist experiences, complex and difficult scenes that path guiding enabled more efficient renders and better workflows on, and general lessons learned.

CCS Concepts

• Computing methodologies → Ray tracing.

ACM Reference Format:

Yining Karl Li, Lea Reichardt, Marco Manzi, Alexander Rath, Brian Green, and Neelima Karanam. 2026. Path Guiding in Disney’s “Zootopia 2”. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH Talks ’26)*, July 19–23, 2026, Los Angeles, CA, USA. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3799818.3812093>

1 Introduction

Path guiding is a variance reduction technique for physically based rendering, that can bring benefit in scenes with complex indirect illumination. Following our previous insights on deploying our second-generation path guiding system in Disney’s Hyperion Renderer [Reichardt et al. 2025], this talk presents our approach to roll out our new path guiding system for the production of “Zootopia 2”. To support at-scale adoption, we solved new technical challenges, and improved our workflow, while minimizing production risks.

In scenes with simple light transport, path guiding can hurt rendering performance due to its computational overhead. Since predicting path guiding efficiency prior to rendering is difficult in complex production scenes, and since A/B testing guided and non-guided renders is impractical at scale, we developed a path guiding efficiency estimate that helps artists decide when to use path guiding. While A/B testing with the guided and non-guided render is a



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
SIGGRAPH Talks ’26, Los Angeles, CA, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2541-8/2026/07
<https://doi.org/10.1145/3799818.3812093>

clear method to determine path guiding efficiency improvement, it is impractical at scale. Section 2.1 presents our approach to calculate the estimate on-the-fly in a single render.

Complex indirect light transport in non-guided renders can cause energy loss due to production renderers often applying clamping to reduce fireflies, as described in Reichardt et al. [2025]. Such energy loss is less common with path guiding, which can cause visible brightness mismatch between the two methods. The mismatch can prevent the use of path guiding on shots that already underwent creative approval. Our clamping correction method (Section 2.2) ensures consistency between guided and non-guided renders, allowing use of path guiding without disrupting the artistic look.

While path guiding is not new in our production environment, our second-generation framework uses an entirely new underlying data structure to gather radiance estimates. The Radiance Recorder [Reichardt et al. 2025] proved effective during large-scale adoption on “Zootopia 2”. We improved performance post-production, addressing key findings detailed in Section 2.3.

Lastly, Section 3 presents insights into the use of path guiding on “Zootopia 2”. Beyond the quantitative improvements, we are excited to report artist workflow improvements due to the use of path guiding, which is a key goal of our technological advancements.

2 Path Guiding Improvements

2.1 Path Guiding Efficiency Estimate

Efficiency is defined as the inverse of the product of the cost and variance over the entire image, i.e. $E = (\text{Cost} * \text{Var})^{-1}$. We use wall clock time to estimate the cost, and the mean *relative* square error (MRSE) [Rouselle et al. 2011] to estimate the image variance

$$\frac{1}{\#px} \sum_{i \in px} \frac{V_i}{\max(1e - 4, \text{avg}(\mathcal{D}_i)^2)}, \quad (1)$$

where V_i is the variance of the color-average in pixel i and $\mathcal{D}_i$ the color of the i th pixel of the denoised image. Here, the denoised image approximates the ground-truth image. We use MRSE instead of (absolute) MSE to down-weight bright regions, like directly visible light sources that would otherwise dominate the estimate.

To quantify the benefit of path guiding, we compute the efficiency improvement ratio $\Delta\text{Eff} = E_{pg+}/E_{pg-}$, where E_{pg+} assumes path guiding is enabled and E_{pg-} assumes it is disabled. Crucially, we can compute both in a single render process. We estimate E_{pg-} during the first sampling iteration, when guiding is not applied yet, and most guiding overhead is not yet present. This iteration hence provides good cost and variance estimates for a non-guided render. We estimate E_{pg+} in a later iteration once the guiding model is reasonably accurate, e.g. at 8spp. Cost and variance are estimated based on the current iteration’s samples only, since they change over time as path guiding refines its model. Because Hyperion distributes two samples for 50% of pixels per nominal sample, we can estimate variance from a single iteration. Note, however, that the denoised image $\mathcal{D}$ still uses the accumulated sample data up to when E_{pg+} can be computed.

The final metric ΔEff is intuitive for artists: If above 1, path guiding increases efficiency and should be enabled, otherwise it should be disabled. Since the estimate can be obtained from low sample count images (e.g. 8spp), artists can use it in a quick pre-pass to

make the decision. In Section 3.2 we validate our estimator against ground-truth efficiency values obtained from separate guided and unguided renders at much higher sample counts.

2.2 Clamping Compensation

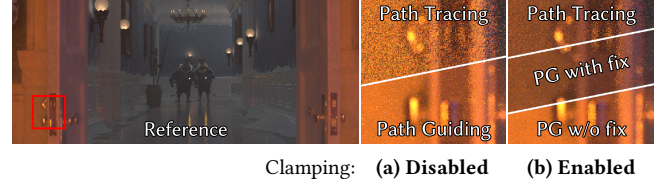


Figure 2: (a) Path tracing and path guiding are unbiased solutions of light transport that converge to the same image. (b) Clamping path contributions eliminates fireflies, but introduces bias that differs between techniques. Our solution makes the bias identical across methods. © 2025 Disney

Production renderers often clamp the contribution of individual paths to the image to reduce the impact of fireflies

$$I_{px} = \sum_i \text{clamp} \left(\frac{f_{px}(\bar{x}_i)}{p_{pt}(\bar{x}_i)} \right). \quad (2)$$

Here, each pixel I_{px} is obtained as the sum of contributions f_{px} from each path $\bar{x}_i$ that was generated with probability density $p_{pt}(\bar{x}_i)$. However, since path guiding generates paths with a different distribution $p_{pg}(\bar{x}_i)$, and the clamping operation is a non-linear function, the expected color of pixels—and thereby the look of objects—can change dramatically. Similarly, the look of objects can change across frames, as the path guiding distribution could change across frames to account for different illumination conditions. Our solution is to factor out the change in probability density out of the clamping operation, effectively performing clamping on the contribution as if it was generated by path tracing:

$$I_{px} = \sum_i \frac{p_{pt}(\bar{x}_i)}{p_{pg}(\bar{x}_i)} \text{clamp} \left(\frac{f_{px}(\bar{x}_i)}{p_{pt}(\bar{x}_i)} \right). \quad (3)$$

2.3 Radiance Recorder Improvements

Our Radiance Recorder data structure and radiance gathering technique proved effective and correct following its first production roll out. Nevertheless, we observed two performance limitations that we addressed post-production: (1) overhead could be as high as 30% when shading was extremely simple, and (2) performance degraded substantially on multi-socket NUMA machines.

To address the first issue, we introduced *per-thread* free lists. Our original implementation allocated Radiance Recorder path vertices from a global pool and pushed them onto a *global* free list when deallocated. Our optimized implementation places vertices into a per-thread free list instead. For allocation, we first check the per-thread free list, and fall back to the global list if the per-thread free list is empty. At the end of each render iteration, the per-thread free lists are freed. With very little extra memory usage, this strategy reduced thread contention and resulted in speedups in simple shading scenarios.

Relieving thread contention on the free lists helped address the second issue, but did not solve it. While our productions typically do not render on multi-socket NUMA nodes, we still wanted to address such severe performance degradations. The original Radiance Recorder implementation stored vertex data in *structure of arrays* (SOA) layout. We chose this layout to minimize memory usage by avoiding the padding requirements of *array of structures* (AOS) layout. In retrospect, this was a poor choice. When running on multi-socket NUMA there was a significant performance penalty just reading and writing the data members of single path vertices, which is the common usage pattern. Switching to AOS format fixed this issue with a barely noticeable memory increase.

3 Path Guiding in "Zootopia 2"

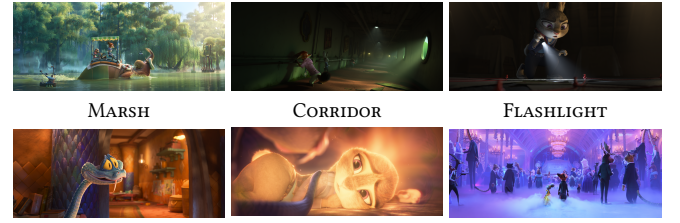
3.1 Efficiency Estimate in Production

"Zootopia 2" utilized a multishot lighting workflow where similar shots were lit together in "packets" before breaking out individual shots for further adjustments. To meet the ambitious production schedule, path guiding was implemented at the packet level. The reasoning was that shots within a packet share the same setting and lighting, meaning the impact on light transport remains consistent across the entire group. Early in the workflow, we built a "pilot render" system which allowed artists to send off automatic 8spp renders to calculate efficiency statistics. Based on these results, they decided whether to enable path guiding for the entire packet. Initially, both guided and non-guided control renders were used to validate our clamping compensation and Radiance Recorder improvements, but the control renders were dropped as confidence in the system increased. As artists built an intuitive understanding of when guiding was beneficial, they gradually moved away from the pilot render process, instead enabling path guiding directly on scenes that called for additional sampling improvements. To track how path guiding was performing on the show, we calculated the path guiding efficiency estimate statistic on all renders that used path guiding, not just for pilot renders.

3.2 Results

Table 1 presents six example "Zootopia 2" production cases where interesting light transport led us to implement path guiding. We show both the estimated path guiding efficiency generated in our pilot render process, and the ground-truth efficiency calculated after production. Our pilot render estimates proved to be strong predictors of ground-truth efficiency gains, even including the *GALA* case where guiding was ineffective. Unless otherwise noted, path guiding was applied to all passes; however, in *SNAKEVISION*, it was utilized only for volumetric passes.

We found that path guiding performed well across diverse setups, ranging from volumetric-heavy exteriors (*MARSH*) and high-bounce interiors (*DE'SNAKE HOME*) to caustics (*CORRIDOR*) and scenes lit using small bright light sources (*FLASHLIGHT*, *SNAKEVISION*). Conversely, some scenes, such as *GALA*, saw no benefit because the overhead of tracing the longer paths sampled by the system outweighed the resulting variance reduction. Expanded results are available in our supplemental document.



Scene	Estimated ΔEff	Groundtruth ΔEff
MARSH	2.36	2.37
CORRIDOR	2.03	2.16
FLASHLIGHT	1.83	2.14
DE'SNAKE HOME	2.39	2.53
SNAKEVISION (VOLUMES)	5.98	4.83
GALA	0.91	0.88

Table 1: Comparison of efficiency estimates to measured ground-truth on shots from "Zootopia 2". © 2025 Disney

4 Future Work and Conclusion

In the future, similar to systems such as Rath et al. [2022], we want to replace pilot renders with using our efficiency estimate to drive fully automatic enabling/disabling of path guiding on-the-fly during a render. In addition, we aim to guide in subsurface scattering and hair rendering, which is currently not supported in our implementation. We are also working to extend our system with VSPG [Xu et al. 2024] and neural approaches [Rath et al. 2025].

The first production use of our second-generation path guiding system on "Zootopia 2" put our new technical advancements to the test: Used by 54 lighters, on 202 shots and 11% of the film, the new method led to rendering efficiency improvements of 1.61x on average. More importantly, path guiding enabled artists to use realistic lighting in complex shots without needing to spend much additional time optimizing for render efficiency; this effect improved artist iteration time and helped maintain high lighting throughput.

References

- 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). <https://doi.org/10.1145/3182159>
- Sebastian Herholz and Addis Dittbrandt. 2022. Intel® Open Path Guiding Library. <http://www.openppl.org>.
- Alexander Rath, Pascal Grittman, Sebastian Herholz, Philippe Weier, and Philipp Slusallek. 2022. EARS: Efficiency-Aware Russian Roulette and Splitting. *ACM Transactions on Graphics (Proc. of SIGGRAPH)* 41, 4, Article 81 (July 2022). <https://doi.org/10.1145/3528223.3530168>
- Alexander Rath, Marco Manzi, Farnood Salehi, Sebastian Weiss, Tiziano Portenier, Saeed Hadadan, and Marios Papas. 2025. Neural Resampling with Optimized Candidate Allocation. In *Proc. of EGSR*. Article 20251181. <https://doi.org/10.2312/sr.20251181>
- Lea Reichardt, Brian Green, Yining Karl Li, and Marco Manzi. 2025. Path Guiding Surfaces and Volumes in Disney's Hyperion Renderer: A Case Study. *Path Guiding in Production and Recent Advancements, SIGGRAPH 2025 Course Notes*, Article 14 (Aug. 2025), 30–66 pages. <https://doi.org/10.1145/3721241.3733994>
- Fabrice Rouselle, Claude Knaus, and Matthias Zwicker. 2011. Adaptive Sampling and Reconstruction Using Greedy Error Minimization. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* 30, 6, Article 159 (Dec. 2011). <https://doi.org/10.1145/2070781.2024193>
- Kehan Xu, Sebastian Herholz, Marco Manzi, Marios Papas, and Markus Gross. 2024. Volume Scattering Probability Guiding. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* 43, 6, Article 184 (Dec. 2024). <https://doi.org/10.1145/3687982>