Texture-Space Differentiable Rendering

Max Slater (Advisors: Ioannis Gkioulekas, Adithya Pediredla)

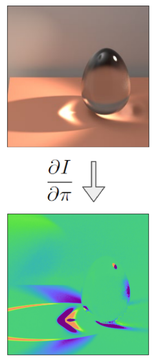
IMAGING @ CMU

Abstract

We evaluate various sampling strategies for estimating texture derivatives in a differentiable renderer. We compare texture-space sampling in forward path tracing for both camera rays and next event estimation. We propose further optimization-guided and multi-directional sampling strategies.

Differentiable Rendering

In physics-based rendering, photo-realistic images are computed by simulating how light bounces through a scene. Recent work on differentiable rendering [2,3,4] provides methods for computing image derivatives with respect to inputs such as object poses, lighting conditions, geometry, and textures. A differentiable renderer may be used to solve numerical optimization tasks such as recovering scene parameters from measurements and adding physics-based loss/regularization to machine learning pipelines.



Due to the use of Monte Carlo integration and various importance sampling strategies, many different approaches to differentiation are viable. We use an attached estimator with multiple importance sampling (MIS), implemented by automatically differentiating a MIS-based path tracer [3].

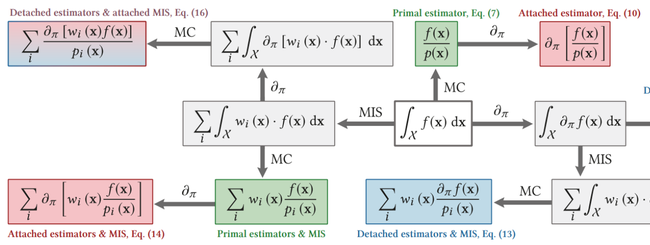


Figure 1. Differentiation Strategies [4]

Researchers are now designing sampling algorithms tailored to the differentiable setting. *Texture-space differentiable rendering* [1] maps texture-space samples into the scene via an object's inverse surface parameterization. When connected to the camera, these points become the first bounce of light paths. Compared to casting rays from the camera itself, the portion of paths that interact with differentiable parameters is increased. In this work, we apply the texture space sampling strategy to next-event estimation (NEE): at each path vertex, we estimate incoming light by sampling directions from not only the current material and emissive objects, but also via a texture-space sample on our differentiable target.

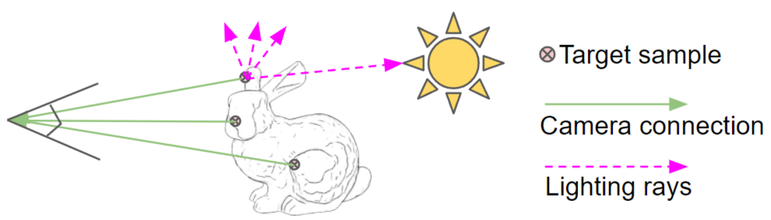
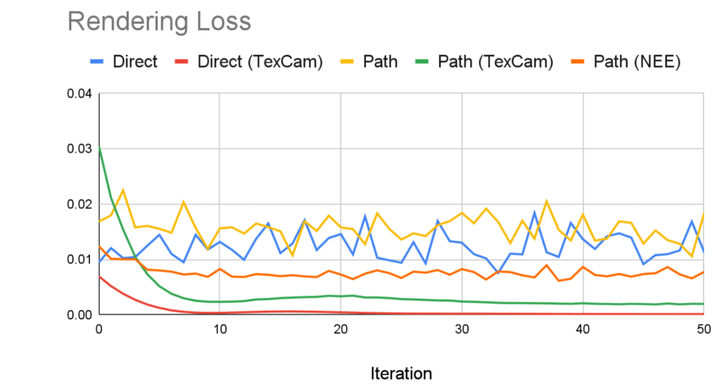
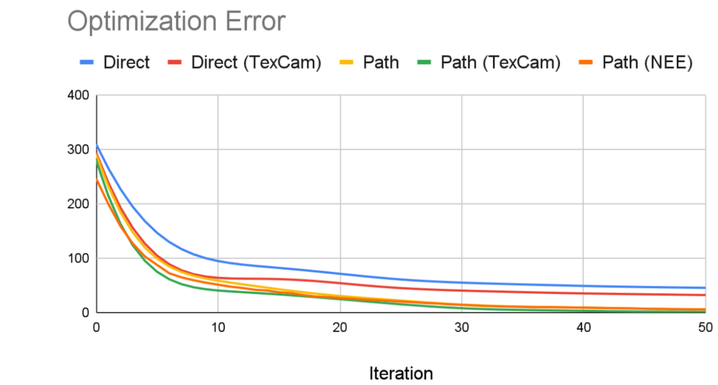


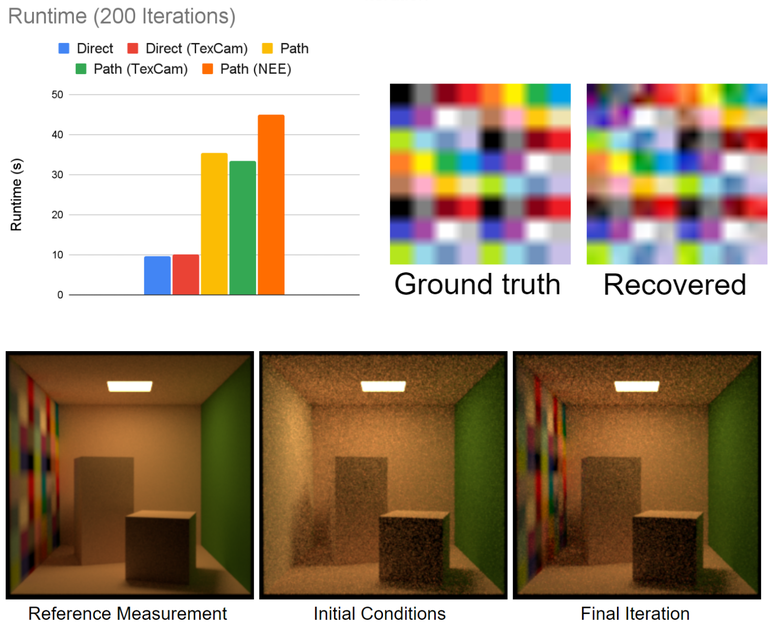
Figure 2. Texture-Sampled Camera Rays

Results

We compared optimization behavior under path tracing and direct lighting with and without texture-space camera rays, as well as path tracing with texture-space NEE. The optimization goal was to recover a 16x16 texture applied to a Cornell box wall. At each optimization step, we plot L2 distance between the reference and rendered image (loss) and distance between the recovered texture and ground truth (error).







Discussion

We determined that texture-space sampling significantly improves convergence speed, as it only generates rays that potentially intersect the target. Only rendering direct lighting greatly improves performance, but introduces bias and fails to recover features lit by indirect paths. Texture-space next event estimation was found to be only slightly more efficient than traditional path tracing, but would likely perform better in scenes more heavily reliant on indirect lighting. Further, our texture-space NEE uses single-sample MIS, increasing variance compared to the baseline path tracer. Switching to multi-sample MIS would likely tip the scales towards texture-space NEE.

Future Work

We look to develop sampling techniques optimized for faster convergence rather than variance reduction. We are exploring the use of optimization parameters to guide sample distribution both in camera and texture space:

1. Estimating per-pixel loss would allow for adaptive distribution of samples to pixels where error is high.
2. When using the ADAM optimizer, per-parameter variance estimates may be used to importance sample texture-space pixels.

In order to handle indirect specular illumination, NEE is traditionally developed into bidirectional path tracing. We may similarly extend our strategy to *tridirectional* path tracing: by generating three subpaths (camera, target, emitter), we can importance sample paths involving multi-bounce connections missed by NEE. Tridirectional tracing would improve sample efficiency in scenes where (e.g.) the target is viewed indirectly.

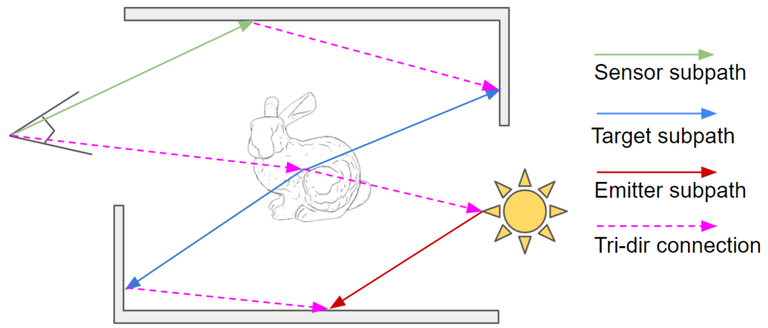


Figure 3. Tridirectional Connections

References

[1] Luke Anderson, Tzu-Mao Li, Jaakko Lehtinen, and Frédo Durand.
Aether: An embedded domain specific sampling language for monte carlo rendering.
ACM Trans. Graph., 36(4), jul 2017.

[2] Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan.
Material and Lighting Reconstruction for Complex Indoor Scenes with Texture-space Differentiable Rendering.
In Adrien Bousseau and Morgan McGuire, editors, *Eurographics Symposium on Rendering - DL-only Track*. The Eurographics Association, 2021.

[3] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob.
Mitsuba 2: A retargetable forward and inverse renderer.
Transactions on Graphics (Proceedings of SIGGRAPH Asia), 38(6), December 2019.

[4] Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob.
Monte carlo estimators for differential light transport.
Transactions on Graphics (Proceedings of SIGGRAPH), 40(4), August 2021.