Learning Light Transport the Reinforced Way

Ken Dahm
NVIDIA
kdahm@nvidia.com

Alexander Keller
NVIDIA
akeller@nvidia.com

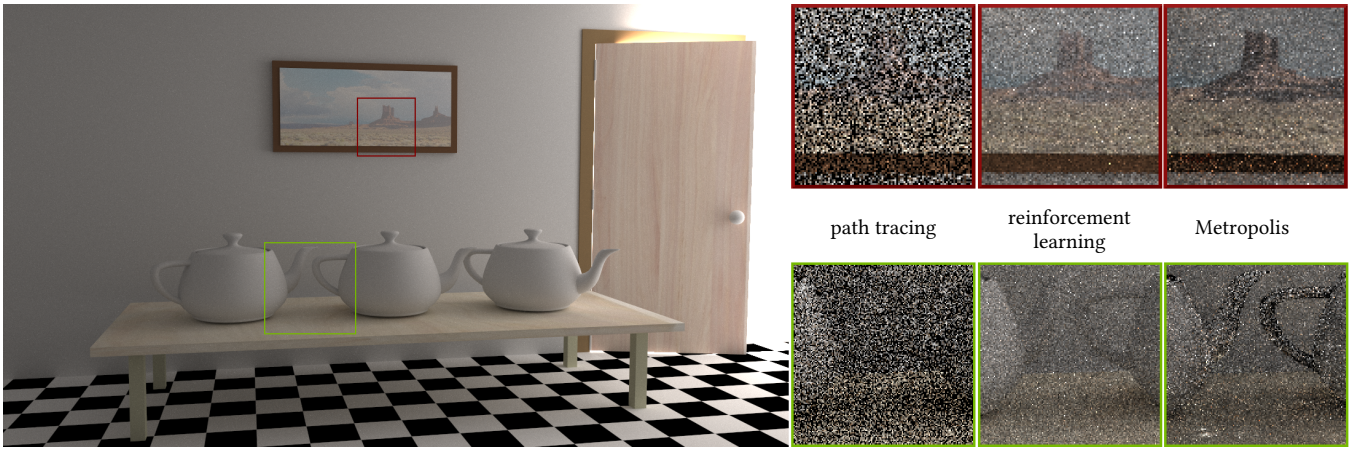

Figure 1: For the same budget of light transport paths, reinforcement learned importance sampling is much less noisy than path tracing with standard importance sampling proportional to surface properties and lacks the smeary splotches typical for Metropolis sampling inherent with its local space exploration as shown in the insets on the right.

ABSTRACT

We introduce a rendering algorithm that is as simple as a path tracer but dramatically improves light transport simulation and even outperforms the Metropolis light transport algorithm. The underlying method of importance sampling learns where radiance is coming from and in fact coincides with reinforcement learning. The cost for the improvement is a data structure similar to irradiance volumes as used in realtime games.

CCS CONCEPTS

•Theory of computation →Reinforcement learning;
•Computing methodologies →Rendering;

KEYWORDS

Rendering, path tracing, reinforcement learning, integral equations, Monte Carlo and quasi-Monte Carlo methods.

ACM Reference format:
Ken Dahm and Alexander Keller. 2017. Learning Light Transport the Reinforced Way. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085032

SIGGRAPH '17 Talks, Los Angeles, CA, USA
 978-1-4503-5008-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084363.3085032

1 INTRODUCTION

Physically based rendering [Pharr et al. 2016] in principle consists of summing up the contributions of light transport paths that connect the camera with the light sources. Due to averaging samples of path space, initial noise is intrinsic to simulating light transport this way. Importance sampling is among the first methods to reduce noise early up. Still, many popular techniques [Pharr et al. 2016; Veach 1997] are limited, as they cannot consider visibility.

Storing an approximation of the incident radiance [Greger et al. 1998; Lafortune and Willems 1995; Vorba et al. 2014], however, allows one to guide light transport paths to where the radiance is originating.

As the equations governing reinforcement learning and light transport simulation are related integral equations [Dahm and Keller 2017], the incident radiance can be learned during rendering. In fact, given a learning rate $\alpha \in (0, 1]$, updating the incident radiance

$$Q'_L(x,\omega) := (1-\alpha)\cdot Q_L(x,\omega) + \alpha \cdot \Bigg(L_e(y,-\omega) + \int_{S^2_+(y)} Q_L(y,\omega_i)\; f_s(\omega_i, y, -\omega)\cos\theta_i d\omega_i \Bigg) \tag{1}$$

is identical to the Q-learning algorithm for reinforcement learning [Watkins and Dayan 1992]. Matching notions in computer graphics and reinforcement learning, light sources L_e are reward, a location x in space corresponds to a state, shooting a ray from x into direction ω is an action resulting in the next state $y := h(x,\omega)$, which is the closest point of intersection of that ray, and the bidirectional scattering distribution function f_s amounts to what is the discount times the action policy. Besides, θ_i is the angle between the surface

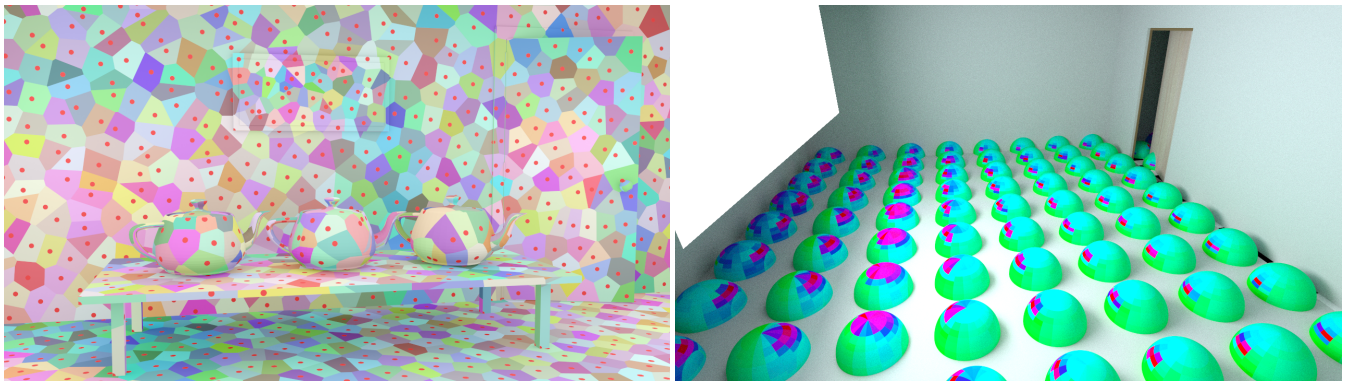

Figure 2: To learn where the radiance originates, points are uniformly distributed over the scene surface (left image), the incident radiance is stored in discretized hemispheres (right image), and updated during image synthesis.

normal and the direction ω_i on the hemisphere $\mathcal{S}^2_+$ in y. Finally, normalizing the incident radiance Q_L over the hemisphere in a point x results in a probability distribution function that corresponds to the policy to choose the next action, i.e. the direction to scatter.

2 ALGORITHM

Extending a path tracer by reinforcement learning requires a representation of the incident radiance Q_L. Among the many choices to represent incident radiance, we uniformly sample the scene surface (see the left image in Fig. 2), placing a discretized hemisphere in each sampled point (see the right image in Fig. 2), where the strata are initialized with a small positive constant. Given a query point x, the closest hemisphere with admissible deviation in normal is selected.

Paths are traced from a virtual camera through the pixels of the screen. First, upon an intersection with the scene geometry, the light transport path is continued into a direction sampled proportionally to incident radiance Q_L. Second, the value of $Q_L(x, \omega)$ is updated according to Eqn. 1, where the integral is evaluated as a weighted sum over the strata of Q_L in the hitpoint y. Note that using Q_L itself instead of path tracing is different from previous approaches and dramatically improves the local convergence of the importance function approximating Q_L. Scattering and ray tracing are repeated until a light source is hit. Finally, the contribution of this complete light transport path is added to the pixel pierced by the initial camera ray.

3 RESULTS

For the comparison in Fig. 1, the same number of paths has been used for all three competing methods. In each iteration, for path tracing with and without reinforcement learning one path has been started per pixel, while for the Metropolis variant the number of Markov chains equals the number of pixels of the image. Rendering the image at 1280x720 pixels, each iteration takes 41ms for path tracing, 49ms for Metropolis light transport [Kelemen et al. 2002; Veach and Guibas 1997], and 51ms for the algorithm with reinforcement learned importance sampling.

Guiding paths towards where the radiance originates requires to compute cumulative distribution functions for hemispheres. These computations are performed in between iterations. Discretizing the hemisphere into 8x8 patches, storing the approximation of Q_L (see Fig. 2) requires about 2MB. This cost is more than amortized by the improved image quality.

4 CONCLUSION

The new consistent algorithm [Dahm and Keller 2017] uses reinforcement learning to progressively learn where light comes from. Hence, the number of light transport paths with non-zero contribution can be dramatically increased, while the average path length is shortened, which improves efficiency. Within a fixed budget of paths, images are much less noisy, while noise remains much more uniform as compared to Metropolis sampling as shown in Fig. 1, which in addition makes the new algorithm much more amenable to noise filtering. Note, however, that the new scheme could as well be combined with Metropolis sampling.

ACKNOWLEDGEMENTS

The authors would like to thank Jaroslav Křivánek and Tero Karras for profound discussions. Scene courtesy (cc) 2013 Miika Aittala, Samuli Laine, and Jaakko Lehtinen, see https://mediatech.aalto.fi/publications/graphics/GMLT/.

REFERENCES

K. Dahm and A. Keller. 2017. Learning Light Transport the Reinforced Way. *CoRR* abs/1701.07403 (2017). http://arxiv.org/abs/1701.07403

G. Greger, P. Shirley, P. Hubbard, and D. Greenberg. 1998. The Irradiance Volume. *IEEE Computer Graphics and Applications* 18, 2 (1998), 32–43.

C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka. 2002. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Computer Graphics Forum* 21, 3 (2002), 531–540.

E. Lafortune and Y. Willems. 1995. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Rendering Techniques 1995 (Proc. 6th Eurographics Workshop on Rendering)*, P. Hanrahan and W. Purgathofer (Eds.). Springer, 11–20.

M. Pharr, W. Jacob, and G. Humphreys. 2016. *Physically Based Rendering - From Theory to Implementation.* Morgan Kaufmann, Third Edition.

E. Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation.* Ph.D. Dissertation. Stanford University.

E. Veach and L. Guibas. 1997. Metropolis Light Transport. In *Proc. SIGGRAPH 1997 (Annual Conference Series)*, Turner Whitted (Ed.). ACM SIGGRAPH, Addison Wesley, 65–76.

J. Vorba, O. Karlík, M. Šik, T. Ritschel, and J. Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)* 33, 4 (2014).

C. Watkins and P. Dayan. 1992. Q-Learning. *Machine learning* 8, 3 (1992), 279–292.