Real-time video based lighting using GPU raytracing

Joel Kronander, Johan Dahlin, Daniel Jönsson, Manon Kok, Thomas Schön and Jonas Unger

Joel Kronander, Johan Dahlin, Daniel Jönsson, Manon Kok, Thomas Schön and Jonas Unger, Real-time video based lighting using GPU raytracing, 2014, Proceedings of the 22nd European Signal Processing Conference (EUSIPCO), 2014.

samples from getting stuck in local narrow modes. The local moves are represented by a uniform random perturbations of the current samples by a few degrees. The independent moves are represented by drawing new samples from the environment map or the BRDF.

6.3. Reflected illumination estimate

Given the weighted sample set obtained from the SMC sampler, the reflected surface radiance (1) is estimated by

$$\hat{L}_r(x;\omega_o;t) = Z_t \sum_{i=1}^{N} W_t^i \frac{\tilde{L}_t(x;\omega_t^i)\,\rho(x;\omega_o;\omega_t^i)V(\omega_t^i)}{\tilde{L}_{Y;t}(x;\omega_t^i)\,\rho_Y(x;\omega_o;\omega_t^i)}. \quad (7)$$

One advantage of the SMC rendering algorithm is that the normalization constant Z_t can be incrementally estimated via the relation

$$Z_t \approx Z_{t-1} \sum_{i=1}^{N} w_t^i. \quad (8)$$

For details on the derivation of this expression see [9]. The initial Z_1 can be estimated from the samples obtained via bidirectional importance sampling in the first frame.

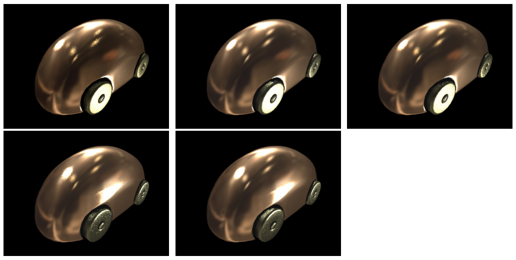
The SMC algorithm presented here works well under the assumption that $\pi_t(\omega) \approx \pi_{t-1}(\omega)$. As the resampling operation does not affect the computation of the normalization constant Z_t, this estimate is likely to be poor when the target distribution changes rapidly. This is typically the case for real HDR video environment maps. To smoothen the transition, a set of intermediate distributions can be used [9] to guide the samples smoothly between the targets. As recommended in [9] we use one MCMC move for each intermediate distribution to adapt the samples gradually.

A big drawback of the original SMC rendering algorithm from [9] is that as time progresses, the variance of the estimated normalizing constant tends to increase. This can lead to visually disturbing artifacts. To counteract this, we propose a simple but effective approach where we monitor the change in $\frac{Z_t}{Z_{t-1}}$ and when a large increase occurs we reinitialize the samples and the normalizing constant estimate using bidirectional importance sampling.

7. IMPLEMENTATION

To enable real-time raytracing with video environment maps we have implemented the three rendering algorithms described above in the CUDA based OptiX 3.0 framework of NVIDIA [1], running on the GPU. We use a regular sampling of the image plane to spawn a set of rays into the scene. Our implementation currently only considers direct illumination, however it is trivial to extend it to path tracing as well, by spawning a new ray from the shading point. To handle several samples per pixel we utilize two OptiX kernels. The first kernel spawns R rays per pixel and updates the associated sample buffer. The second kernel filters the reflected radiance of the sampled ray locations using an reconstruction filter and tonemaps the image for display. In the examples presented here, we used a box filter for reconstruction and a gamma mapping for tonemapping. To sample from the environment map we use the inversion method with tabulated row and column CDFs [19]. The environment maps at frame t and $t-1$ are accessed through two texture samplers, enabling efficient lookups using the texture hardware on the GPU. To draw samples from the environment map we precompute tabulated column and row CDF on the CPU and upload this to a read-only global GPU buffer before rendering the frame. For the SMC rendering algorithm we for each queried shading point we read, compute and store the sampled directions and weights $\{\omega_t^i, w_t^i\}_{i=1}^{N}$ in a 3D floating point buffer residing in global GPU memory indexed using the ray origin.

Fig. 2: Real-time rendered helicopter model (25 fps using SMC) composited into the backplate video sequence



(a) Env.Map. (b) MIS (c) BIS (d) SMC

Fig. 3: Equal rendering time comparison using a video light probe with several moving direct light sources. The media is best viewed in the pdf file. Top row: frame 5, Bottom row: frame 90. a) Light probe used to illuminate frame b) Results rendered using MIS with $N_\rho = 15$, $N_L = 15$ c) Results rendered using BIS with $M = 400$ and $N = 20$, d) SMC with $N = 8$ particles and 3 intermediate distributions.

8. RESULTS AND COMPARISIONS

All the results presented in here were computed using a desktop PC with a NVIDIA GTX 770 graphics card. Using the described OptiX implementation we can render virtual objects in temporally varying illumination environments in real time. In figure 2, one frame of a video sequence with a rendered helicopter model composited into a high resolution blackplate video sequence is shown. The helicopter model was rendered with a video light probe, enabling moving reflections, in 26 fps using the SMC rendering algorithm.

Comparing the three different rendering algorithms, we found that somewhat surprisingly the MIS and BIS algorithms perform better for a fixed computational cost. This is in contrast to previous work using CPU renderers which showed a increased efficiency of the SMC rendering algorithm compared to BIS [9]. A reason for the decreased performance on GPUs is likely due to the global memory access needed to read the result from the previous frame. We have also found that the SMC algorithm is highly sensitive to the parameters, such as the number of intermediate distributions and the mixture of MH proposals. This limits the applicability of theses methods. Furthermore, MIS and BIS have a strong advantage in practice in that they can handle freely moving geometry and cameras with little extra implementation effort while SMC requires back projecting shading point to the image plane in the next frame [9].

The choice between MIS and BIS is dependent on the type of materials and geometry present in the scene. For BIS we found that first sampling from the BRDF leads to a much more efficient sampling algorithm as sampling from the environment is more costly than sampling from the Blinn-Phong material we used in the renderings. This choice implies that BIS is more efficient in rendering glossy surfaces, see for example the comparison in figure 3, similar results hold for other glossy objects in other scenes. However, MIS using half the samples to sample the BRDF and half the samples to sample form the environment map is more robust to different illumination environments and material types than our BIS implementation. In future work we would like to consider approaches inspired by the SMC samplers for improving the MIS and BIS estimators, for example using ratio estimators of the normalizing constants, however without storing all the explicit samples between frames. We would also like to investigate the use of control variated to limit the amount of computation necessary for each new frame.

REFERENCES

[1] Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al., "OptiX: a general purpose ray tracing engine," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 66, 2010.

[2] Paul Debevec, "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1998, SIGGRAPH '98, pp. 189–198, ACM.

[3] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers, *Advanced high dynamic range imaging: theory and practice*, CRC Press, 2011.

[4] Vlastimil Havran, Miloslaw Smyk, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel, "Interactive system for dynamic scene lighting using captured video environment maps," in *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, Aire-la-Ville, Switzerland, Switzerland, 2005, EGSR'05, pp. 31–42, Eurographics Association.

[5] Jonas Unger, Joel Kronander, Per Larsson, Stefan Gustavson, and Anders Ynnerman, "Temporally and spatially varying image based lighting using hdr-video," in *Proceedings of EUSIPCO'13: Special Session on HDR-video*, 2013.

[6] Saghi Hajisharif, Joel Kronander, Ehsan Miandji, and Jonas Unger, "Real-time image based lighting with streaming HDR-light probe sequences," in *Proc. of SIGRAD '12*, 2012.

[7] Eric Veach and Leonidas J Guibas, "Optimally combining sampling techniques for Monte Carlo rendering," in *Proceedings of SIGGRAPH*, New York, NY, USA, 1995, ACM, pp. 419–428.

[8] David Burke, Abhijeet Ghosh, and Wolfgang Heidrich, "Bidirectional importance sampling for direct illumination.," in *proc. of EGSR 2005*. 2005, vol. 5, pp. 147–156, Citeseer.

[9] Abhijeet Ghosh, Arnaud Doucet, and Wolfgang Heidrich, "Sequential sampling for dynamic environment map illumination," in *Proceedings of the 17th Eurographics conference on Rendering Techniques*. Eurographics Association, 2006, pp. 115–126.

[10] Karol Myszkowski, *High Dynamic Range Video*, Morgan and Claypool Publishers, 1 edition, 2008.

[11] Joel Kronander, Stefan Gustavson, Gerhard Bonnet, and Jonas Unger, "Unified HDR reconstruction from raw CFA data," in *IEEE International Conference on Computational Photography (ICCP), 2013*, April 2013, pp. 1–9.

[12] Joel Kronander, Stefan Gustavson, Gerhard Bonnet, Anders Ynnerman, and Jonas Unger, "A unified framework for multi-sensor HDR video reconstruction," *Signal Processing: Image Communication*, vol. 2, no. 19, pp. 203–215, 2014.

[13] Christian Bloch, *The HDRI Handbook 2.0: High Dynamic Range Imaging for Photographers and CG Artists*, Rocky Nook, 2012.

[14] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi, "Efficient brdf importance sampling using a factored representation," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 496–505, 2004.

[15] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen, "Structured importance sampling of environment maps," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 605–612, 2003.

[16] Liang Wan, Shue-Kwan Mak, Tien-Tsin Wong, and Chi-Sing Leung, "Spatiotemporal sampling of dynamic environment sequences," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 10, pp. 1499–1509, 2011.

[17] Arnaud Doucet and Adam M Johansen, "A tutorial on particle filtering and smoothing: 15 years later," in *Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.

[18] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra, "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, 2006.

[19] Matt Pharr and Greg Humphreys, *Physically based rendering: From theory to implementation*, Morgan Kaufmann, 2010.