Radiance Cache Splatting: A GPU-Friendly Global Illumination Algorithm

Pascal Gautron
pgautron@irisa.fr
IRISA *- UCF †

Jaroslav Křivánek
jarda@slimak.cz
IRISA * - UCF † - CTU ‡

Kadi Bouatouch
kadi@irisa.fr
IRISA *

Sumanta Pattanaik
sumant@cs.ucf.edu
UCF †

(a) Sibenik Cathedral (80K poly) (b) Sponza Atrium (66K poly) (c) Castle (58K poly)

Figure 1: First bounce diffuse and glossy global illumination rendered at resolution 1000×1000 in less than 20s by our GPU-based renderer.

Introduction

Fast global illumination computation is a challenge in several fields such as lighting simulation and computer-generated visual effects for movies. To this end the irradiance caching algorithm is commonly used since it provides high-quality renderings in a reasonable time. However this algorithm relies on a spatial data structure in which nearest-neighbors queries and data insertions are performed alternately within a single rendering step. Due to this central and permanently modified data structure, irradiance caching algorithm cannot be easily implemented on graphics hardware. We propose a novel approach to first bounce glossy global illumination using irradiance and radiance cache: the *radiance cache splatting*. This method directly meets the processing constraints of graphics hardware since it avoids the need of complex data structure traversal.

Radiance Cache Splatting

Our work is based on the irradiance and radiance (or *(ir)radiance*) caching algorithms [Ward et al. 1988; Křivánek et al. 2005b]. These approaches rely on sparse sampling and interpolation of indirect incoming radiance. For each sample location, an (ir)radiance record represents the sampled incoming radiance. Given a record, its value can be used to estimate the indirect incoming radiance at nearby points. Records are stored in the (ir)radiance cache.

In order to generate an image, the classical (ir)radiance caching algorithm queries the spatial data structure of the (ir)radiance cache for each visible point to determine its indirect incoming radiance. We propose an image-plane method which uses the opposite approach: for a given record, our algorithm determines which visible points it contributes to. This set of visible points is determined using the weighting function and interpolation scheme proposed in [Ward et al. 1988]: a given record k located at $\mathbf{p}_k$ can contribute to the incoming radiance at point $\mathbf{p}$ only if $\|\mathbf{p}-\mathbf{p}_k\| \leq aR_k$. a represents a user-defined accuracy value and R_k is the harmonic mean distance to the objects visible from $\mathbf{p}_k$. Therefore, record k can only contribute to the indirect lighting of points within a sphere of influence I_k centered at $\mathbf{p}_k$ with radius aR_k. For each record k, our algorithm splats I_k onto the image plane. For each pixel within the splat we accumulate the contribution of the record into the *radiance splat buffer*. When each record has been processed, the radiance splat buffer contains the indirect lighting of the visible points. Mixed with direct lighting, this method generates a fast, high quality global illumination solution. Note that radiance cache splatting, unlike original radiance caching, does not rely on any spatial structure.

This approach has been efficiently implemented on graphics hardware. For a given record k, our algorithm splats I_k on the image plane by drawing a quadrilateral tightly bounding the projection of I_k on the image plane. Then, for each fragment within the quadrilateral, the gradient-based methods defined in [Ward et al. 1988; Křivánek et al. 2005a] are used to determine the contribution of the record to the incoming radiance of visible points. The accumulation is performed on the GPU using native floating-point blending capabilities. Since the record computation represents an important part of the rendering time, we also use the GPU to compute (ir)radiance records.

Results and Conclusion

In Figure 1, we show the results from an implementation of our radiance cache spatting algorithm. The comparison with the Radiance software shows that our GPU-based renderer drastically reduces the rendering time: the *Sponza Atrium* scene is rendered in 10min45s with Radiance, while our renderer generates an image with comparable quality in 13.7s, yielding a speedup of 47.1.

Future work will focus on extending this work to multiple light bounces and animated scenes. We would also like to accelerate the rendering process to obtain high quality global illumination in real-time.

References

KŘIVÁNEK, J., GAUTRON, P., BOUATOUCH, K., AND PATTANAIK, S. 2005. Improved Radiance Gradients Computation. *To appear in Proceedings of SCCG*.

KŘIVÁNEK, J., GAUTRON, P., PATTANAIK, S., AND BOUATOUCH, K. 2005. Radiance Caching for Efficient Global Illumination Computation. *To appear in IEEE TVCG*.

WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In *Proceedings of SIGGRAPH*, 85–92.

*IRISA/INRIA, Rennes, France
†University of Central Florida, Orlando, FL, USA
‡Czech Technical University, Prague, Czech Republic