The Overview of Many Light Rendering

Genyuan Zhang[1,a], Jun Huang[2,b], Huabing Zhang[1, c], Xiaohua Li[3,d]
[1]*The Zhejiang University of Media and Communications, Hangzhou, P.R. China*
[2]*Shanghai Advanced Research Institute Chinese Academy of Sciences, China*
[3]*The information center, Zhejiang University, China*
[a]*zgenyuan@163.com,* [b]*huangj@sari.ac.cn,* [c]*huabingzhang@163.com,* [d]*xiaohua@zju.edu.cn*

Abstract

Rendering complex scenes with indirect illumination, high dynamic range environment lighting, and many direct light sources remains a challenging problem. Prior work has shown that all these effects can be approximated by many point lights. For hundreds of thousands of lights, a brute force solution that computes all columns of the many-lights problem is prohibitively expensive. Many methods have been proposed to reduce the computation complexity of the many-lights problem. In this paper, We introduce three different many light algorithms which exploit matrix structure from different aspects. The advantage and disadvantage are analyzed according to different aspects of three algorithms.

Keywords: many light rendering; lightslice; row-collumn sampling;

Introduction

Fast computation of global illumination in complex scenes with many glossy surfaces is a long-standing unsolved problem of rendering research; indeed, no satisfactory solution exists. Pure

Monte Carlo methods like path tracing, bidirectional path tracing or Metropolis [4] take very long to converge. Photon mapping with final gathering [11] is generally much faster than pure Monte Carlo, but not so much in highly glossy scenes. Irradiance caching [13] or radiosity approaches [15] are not applicable to non-diffuse materials, and their directional extensions [14] can only handle moderate gloss. Many-light methods [10] provide good performance, but are incorrect for two reasons. First, the generated virtual point lights (VPLs) are diffuse-only, so light bouncing off glossy surfaces is ignored. Second, the VPL contributions are clamped to prevent illumination spikes. These measures produce visually pleasing results in diffuse scenes, but they remove much of the interesting illumination in scenes with a substantial proportion of glossy materials. Lightcuts [16] hierarchically clusters the lights into a light tree using geometric proximity as the cluster metric. It then renders the final image by choosing a set of representative clusters differently for each pixel. Matrix Row-Column Sampling (mrcs in short) [10] clusters entire matrix columns together and renders one representative column for the entire clusters. This is motivated by the observation that the transport matrix is close to low rank. For large environments and complex lighting, neither lightcuts nor mrcs optimally exploits the structure found in these matrices. The former works well for local lighting and mostly-visible global lights, but oversamples shadowed global lights (corresponding to bright columns with large black segments or entirely black). The latter works well for global lights, quickly determining the global visibility behavior, but is inefficient for local lighting (corresponding to low intensity columns that are mostly black) in that it samples them for all pixels.

The main observation of our work is that, if we cluster similar pixels together, the slice of the matrix corresponding to these pixels has significantly lower rank than the original matrix. Intuitively this is true since for each slice, local lighting and shadowed regions can all be approximated together with a low intensity representative. However each of these approaches are significantly different from ours since they aim to approximate the whole matrix.

Overview of Many Light Rendering

lightslice

LightSlice[2], an algorithm that efficiently solves the many-lights problems by sampling matrix slices. LightSlice[2] first determine matrix slices by clustering similar image pixels based on their geometric proximity. For each of these slices a representative and roughly cluster all columns based on all row values is rendered. This initial clustering effectively captures the global structure of the matrix[2]. For each slice, then such global clusters into per-slice clusters based on representative rows of the given slice and its neighboring slices are refined[2]. This effectively captures the local structure of the matrix, including its shadowing behavior[2]. Each slice is rendered by choosing representative columns and only rendering the column elements corresponding to the slice rows[2]. LightSlice combines the advantages of both lightcuts and mrcs by effectively capturing the local structure of the matrix, including its shadowing, while adapting to the local changes for each slice[2]. LightSlice is consistently faster than other algorithms, with between three and six times speedup[2]. More importantly, each of these prior algorithms works well for some scenes, but becomes inefficient for others[2]. This is due to the fact that each of them is optimal for some matrix structure but inefficient for others[2]. LightSlice is instead consistently efficient in all our scenes since it can adapt to the typical matrix structures found in complex lighting scenarios[2].

Rrow-column sampling

[3] presented an algorithm to compute fast and high-quality solutions to the many-light problem, which is treated as the problem of approximating the sum of all columns of an unknown matrix. [3] explores the matrix structure by sampling a small set of rows, and reconstruct the image by rendering a small set of representative columns. [3] explicitly takes advantage of GPU acceleration and requires no precomputation. Since complex and arbitrary object appearance can be expressed in the context of the many-light problem, [3] could have compelling applications in cinematic and architectural lighting design. One of the drawbacks of [3] is that shadow mapping artifacts may be present. In particular shadow bias is an issue, since there might not exist a single bias setting that works for all 100 thousand automatically generated lights. Moreover, the conversion of indirect illumination to point

lights requires clamping, common to all similar approaches [Keller 1997; Walter et al. 2005]. This leads to slight darkening of indirect lighting, especially in corners[3]. [3] does not specifically address this limitation of the many-light formulation, treating indirect lights like any other programmable shader. Furthermore, while our algorithm is mostly designed for previewing single frames, [3] would like to explore rendering animations. Currently, slight temporal artifacts might be seen due to the Monte Carlo nature of the algorithm, and can be remedied by increasing the number of samples.

Virtual Spherical Rendering

[1]have introduced the *virtual spherical light*, which addresses fundamental limitations in many-light rendering of glossy scenes: the loss of energy due to clamping and the use of diffuse VPLs. [1] have shown that rendering with VSLs produces high-quality images in minutes, even in relatively difficult lighting scenarios where current approaches are either incorrect, or converge very slowly (taking hours). [1] takes an important step towards solving the difficult problem of glossy inter-reflections, and will stimulate new developments in the field of many-light rendering. goal is to eliminate these limitations of many-light methods, thereby developing the first algorithm to render scenes with high amounts of glossy reflectance significantly faster than pure Monte Carlo techniques[1]. Our key contribution is a new light type, the *virtual spherical light*, with the following desirable properties: The

point-wise evaluations of traditional VPLs are replaced by integration over a non-zero solid angle, eliminating the spikes caused by narrow glossy BRDF lobes and the geometry term[1]. Clamping is no longer needed, preserving illumination energy that would otherwise be lost. Since the VSL acts as a point light in visibility computations, fast GPU shadow mapping can be exploited. The VSL contribution depends only on values local to the surface point and light location, so its estimation becomes a purely numerical kernel ideally suited to evaluation in a GPU shader[1].

Conclusion

In this paper, We introduce three different many light algorithms which exploit matrix structure from different aspects. The advantage and disadvantage are analyzed according to different aspects of three algorithms. This work is partially supported by Zhejiang Provincial projects (2014C31075), the National Nature Science Foundation of China (61201446) ,and the National Key Technology R&D Program projects (2012BAH43F03,2013BAH27F01, 2013BAH27F02).

References

[1] Miloˇs Haˇsan, Jaroslav Kˇriv´ anek, Bruce Walter, Kavita Bala. Virtual Spherical Lights for Many-Light Rendering of Glossy Scenes.Siggraph Asia 2009.

[2] Kevin Egan, Fredo Durand, Ravi Ramamoorthi, Matthias Zwicker. LightSlice: Matrix Slice Sampling for the Many-Lights Problem. Siggraph Asia 2011.

[3] Miloˇs Haˇsan, Fabio Pellacini, Kavita Bala. Matrix Row-Column Sampling for the Many-Light Problem. Siggraph 2007.

[4] VEACH, E. 1997. Robust Monte Carlo Methods for Light Transport Simulation. PhD thesis, Stanford University.

[5] PHARR, M., AND HUMPHREYS, G. 2010. Physically Based Rendering, Second Edition: From Theory To Implementation, 2nd ed. Morgan Kaufmann Publishers Inc.

[6] KELLER, A. 1997. Instant radiosity. In Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series, 49–56.

[7] WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K.,DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: a scalable approach to illumination. ACM Transactions on Graphics 24, 3 (Aug.), 1098–1107.

[8] RITSCHEL, T., ENGELHARDT, T., GROSCH, T., SEIDEL, H.-P.,KAUTZ, J., AND DACHSBACHER, C. 2009. Micro-rendering forscalable, parallel final gathering. ACM Transactions on Graphics 28, 5 (Dec.), 132:1–132:8.

[9] CHRISTENSEN, P. H. 2008. Point-based approximate color bleeding. Tech. Rep. 08-01, Pixar Animation Studios.

[10] HAˇSAN, M., PELLACINI, F., AND BALA, K. 2007. Matrix rowcolumn sampling for the many-light problem. ACM Transactions on Graphics 26, 3 (July), 26:1–26:10.

[11] JENSEN, H. W. 2001. Realistic image synthesis using photon mapping. A. K. Peters, Ltd., Natick, MA, USA.

[12] KELLER, A. 1997. Instant radiosity. In Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series,49–56.

[13] WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In Proceedings of ACM SIGGRAPH 88, 85–92.

[14] KˇRIV ´ANEK, J., GAUTRON, P., PATTANAIK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination. In IEEE Transactions on Visualization and Computer Graphics, 550–561.

[15] COHEN, M. F., AND WALLACE, J. R. 1993. Radiosity and Realistic Image Synthesis. Morgan Kaufmann, San Francisco, CA.

[16] WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K.,DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: a scalable approach to illumination. ACM Transactions on Graphics 24, 3 (Aug.), 1098–1107.