Global Illumination using Precomputed Light Paths for Interactive Light Condition Manipulation

Yonghao Yue*  
Univ. Tokyo  
Kei Iwasaki  
Wakayama Univ.  
Yoshinori Dobashi  
Hokkaido Univ.  
Tomoyuki Nishita  
Univ. Tokyo

Figure 1: (a), (b), and (c) show the rendered results using our method. The scenes contain polygons up to 142k, and the frame rates are 1–9 fps on a desktop PC with PentiumD 3.0 GHz and GeForce 7800 GTX. (d) shows the rendered result using a pure Monte Carlo method.

1 Introduction

In this paper, we propose an interactive rendering method for taking into account global illumination using light path precomputation and a hierarchical data structure for fast final gathering. Our method can render static scenes interactively allowing the user to move the viewpoint and the positions of local light sources. Moreover, our method can change the materials (including the textures, diffuse and low-frequency glossy BRDFs) of objects at run-time. In recent years, many methods have been proposed to achieve interactive rendering by precomputing light transfer, for example, [Haşan et al. 2006]. These methods, however, do not allow the user to modify the materials of objects interactively.

We propose to record each light path rather than the transfer matrix. To calculate global illumination, we use particle tracing (a variant of photon mapping) and final gathering as [Haşan et al. 2006] did. The differences lie in the processes of precomputation and final gathering. We also use radiance caching to compute the color of each pixel. The rendering process can be implemented fully on the GPU, and interactive rendering rates are achieved.

2 Precomputation and Rendering

To precompute light paths for particle tracing, we first sample entry points of light paths uniformly in the entire scene (Figure 2(a)). Then we sample each light path from each entry point (b). We consider light paths with length up to 3 (b) shows light paths with length 2). To precompute final gathering rays, we have to determine where to perform final gathering. We use a rejection sampling technique to do this. That is, we densely sample candidate points uniformly in the entire scene, and check for each point whether the computation can be interpolated from other points (c).

There are two main tasks during the rendering process. The first one is to recalculate the energy of each precomputed light path. Since we have sampled the entry points uniformly, the differential solid angle of an entry point viewed from the sampling point of a light source can be obtained easily. Therefore, we can recompute the energy of the entry point of each light path, and the energies of the remaining vertices in the path can be obtained by multiplying the precomputed reflectance (d).

The other main task is to perform final gathering as fast as possible from the radiance distribution obtained from the recomputation of the energy of light paths. We extend the Irradiance Atlas method proposed by [Christensen and Batali 2004] for faster computation and handling of glossy BRDFs. To do so, we construct a hierarchical volumetric data structure for the entire scene ((e), (f) and (g)), which accumulates the radiance exitant from each volumetric grid and records the radiance toward each discretized direction in the grid. To gather radiance via the data structure, the projected solid angle of each gathering ray is accounted for when deciding the level of the hierarchy (h). The computation of the final gathering does not need any nearest neighbor searching, and can be performed deterministically. The entire process during rendering can be performed on the GPU.

3 Results and Conclusion

We show the rendered results in Figure 1. The memory usage is between 165 MB and 197 MB, which is nearly equal to the memory requirement for a photon map solution. The time for precomputation is about 50 minutes. Using our method, we were able to interactively render scenes with global illumination, moving the viewpoint and light sources, and/or changing the characteristics of materials. From the results described above, we conclude that we have developed an efficient rendering method for interactive global illumination.

References
