

Multiresolution Radiosity Caching for Global Illumination in Movies

Per H. Christensen

George Harker

Jonathan Shade

Brenden Schubert

Dana Batali

Pixar Animation Studios



Figure 1: Global illumination images from ‘Monsters U’. These images render more than $30\times$ faster with radiosity caching. © Disney/Pixar.

1 Introduction

We describe a multiresolution radiosity caching method that enables efficient computation of global illumination (GI) in a single pass in complex CG movie production scenes.

For ray-traced GI in scenes with complex geometry and shaders, the bottleneck is not the “raw” ray tracing (acceleration structure traversal and ray hit tests), but the evaluation of displacement, light source, and surface shaders at the ray hit points. We reduce this time by dividing the shader calculations into view-dependent and view-independent components, and caching the view-independent part (radiosity) needed for GI. During distribution ray tracing these radiosities are computed on demand and reused many times. The cache contains multiple resolutions of the radiosity on the surface patches in the scene; the resolution for each cache lookup is selected using ray differentials. Our GI method is implemented in PRMan and used for interactive material/lighting design and final rendering at several studios.

2 Global Illumination in Movies

The most widely used GI methods in CG movies and special effects are distribution ray tracing, path tracing, and point-based GI [Křivánek et al. 2010].

Tabellion and Lamorlette [2004] compute direct illumination and store it as texture maps, and then use distribution ray tracing to compute 1-bounce GI. Path tracing [Fajardo 2010] is a brute-force single-pass method that provides fast feedback during interactive lighting design. Its main disadvantages are slow convergence to a noise-free final image and unsuitability for caching of shading results (each shading result is very noisy so cannot be reused without introducing significant bias). Point-based GI [Christensen 2008] is fast and noise-free and requires few shader evaluations, but due to its multipass nature, it is not ideal for interactive design.

Our caching method combines the best traits: it is single-pass (suitable for interactive design), needs few shader evaluations due to caching, handles multiple bounces, converges fast, and has fixed cache size and no file I/O.

3 The Multiresolution Radiosity Cache

The radiosity cache contains surface patch radiosities at three different resolutions. The fine resolution contains radiosities for all the micropolygon vertices of a patch; the medium and coarse resolutions contain a subset of those. The implementation is similar to our multiresolution tessellation cache [Christensen et al. 2006].

We use ray differentials to select the cache resolution for a ray hit. Lookups in the fine cache are coherent; lookups in the medium and coarse caches are less coherent, but the capacity of those caches is higher since each entry is smaller. Hence, high cache hit rates are obtained for all resolutions. If the cache does not contain an entry for the requested patch/side/resolution/diffusedepth, the view-independent part of the surface shader is run (along with the displacement and light source shaders) at the patch vertices, and the resulting radiosities are stored. Then the radiosity values near the hit point are interpolated.

4 Global Illumination Examples

Figure 1 shows GI in three movie scenes with motion blur. The images are 1920×1080 pixels and were rendered using 4 threads on a 2.9 GHz machine with 12 GB memory. The direct illumination consists of 2 lights (sun and dome) for the exterior scene and 20 lights for the interior scenes. There are 13, 64, and 15 million shading points, respectively. The indirect illumination is rendered with 256–1024 diffuse rays from each shading point. The radiosity cache size is set to 256 MB; peak total memory is 16, 11 and 6 GB. The render times are 1.5, 4.0, and 1.1 hours. *Without radiosity caching the render times are 33, 41, and 94 times longer.*

5 Other Caching Applications

Multiresolution caches are also used to store shader opacity results for faster ray-traced shadows, ambient occlusion, and volume extinction, and to store irradiance for ray-traced subsurface scattering.

More details at: graphics.pixar.com/library/RadiosityCaching. Two videos illustrate interactive changes of viewpoint, illumination, and shader parameters while rendering GI and subsurface scattering with progressive ray tracing.

References

- CHRISTENSEN, P., FONG, J., LAUR, D., AND BATALI, D. 2006. Ray tracing for the movie ‘Cars’. In *Proc. IEEE Symposium on Interactive Ray Tracing 2006*, 1–6.
- CHRISTENSEN, P. 2008. Point-based approximate color bleeding. Tech. Rep. 08-01, Pixar Animation Studios.
- FAJARDO, M., 2010. Ray tracing solution for film production rendering. In [Křivánek et al. 2010].
- KŘIVÁNEK, J., ET AL. 2010. Global illumination across industries. *SIGGRAPH 2010 Course Notes*.
- TABELLION, E., AND LAMORLETTE, A. 2004. An approximate global illumination system for computer generated films. In *Proc. SIGGRAPH 2004*, 469–476.