Direct Illumination from Dynamic Area Lights With Visibility

Greg Nichols[*] University of Iowa
Rajeev Penmatsa[†] University of Iowa
Chris Wyman[‡] University of Iowa

Figure 1: *A simple scene illuminated solely by a large area light source displaying a video; (left) with illumination; (right) visibility only. Illumination varies in real-time as the video plays, and the complex soft shadows change convincingly as the objects are manipulated.*

1 Introduction

Although area light sources are common in the real world, their use in interactive rendering is challenging. Each surface point on such a light source can contribute illumination to every point in the scene, and non-binary visibility gives rise to complex shadows that cannot be easily approximated using point light shadow techniques. We recognize that this problem is similar in nature to the rendering of single-bounce indirect illumination, and can be addressed with similar techniques.

One previous method of rendering indirect illumination [Dachsbacher and Stamminger 2006] computes a reflective shadow map (RSM), chooses VPLs from the RSM, and splats each VPL's contribution onto the scene. This technique extends naturally to the problem at hand: we simply gather the VPLs from the surface of our area light source. In contrast to these techniques, we add a visibility approximation: we compute occlusion by marching rays towards each VPL through a coarse voxel buffer. Combining these two approaches allows interactive scenes with dynamic area light sources (such as video screens) that cast realistic shadows.

2 Our Approach

2.1 Visibility using Real Time Voxelization

We take advantage of recent work in real-time solid voxelization [Eisemann and Décoret 2008] that produces a coarse set of voxels stored in an integer framebuffer; in a given texel, each bit represents a quantized depth slice. A bit is turned on if geometry exists in that slice, and turned off otherwise. This voxel buffer serves as an efficient representation of scene geometry for our purposes.

We compute visibility along a ray by marching through the voxel buffer, checking at discrete steps to see the nearest voxel bit is set (and hence, the ray is occluded).

[*]e-mail: gbnichol@cs.uiowa.edu
[†]e-mail: rpenmats@cs.uiowa.edu
[‡]e-mail: cwyman@cs.uiowa.edu

I3D 2010, Washington, DC, February 19 – 21, 2010.
ISBN 978-1-60558-938-1/10/0002

2.2 Gathering from VPLs

We slightly reorganize the original RSM approach, avoiding splatting in favor of gathering illumination from the set of VPLs. When rendering a fragment, we first compute the visibility of each VPL as described above; if a VPL is unoccluded, we add its illumination.

This technique benefits from interleaved sampling, in which we compute visibility and illumination with different ray offsets and different sets of VPLs for neighboring pixels. This allows each fragment to do less computation, bringing a corresponding speed increase. We also find that we can boost performance by using a relatively low resolution voxel buffer, and precomputing coarse visibility once per frame, allowing us to avoid many finer voxel buffer traversals.

3 Results and Future Work

Our method renders the scene in Figure 1 at 25fps, with full-motion video on the light source. In the future, we plan to adapt recent multiresolution techniques [Nichols et al. 2009] to this problem, further improving performance. These techniques also smooth illumination samples during upsampling, which will filter out some of the high-frequency artifacts that can be seen in Figure 1.

Acknowledgements: The video still in Figure 1 is from "Big Buck Bunny", copyright © Blender Foundation (www.blender.org).

References

DACHSBACHER, C., AND STAMMINGER, M. 2006. Splatting indirect illumination. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, 93–100.

EISEMANN, E., AND DÉCORET, X. 2008. Single-pass gpu solid voxelization for real-time applications. In *Proceedings of Graphics Interface*, 73–80.

NICHOLS, G., SHOPF, J., AND WYMAN, C. 2009. Hierarchical image-space radiosity for interactive global illumination. *Computer Graphics Forum*, 1141–1149.