CC Shadow Volumes

Brandon Lloyd Jeremy Wendt Naga Govindaraju Dinesh Manocha
University of North Carolina at Chapel Hill
http://gamma.cs.unc.edu/ccsv

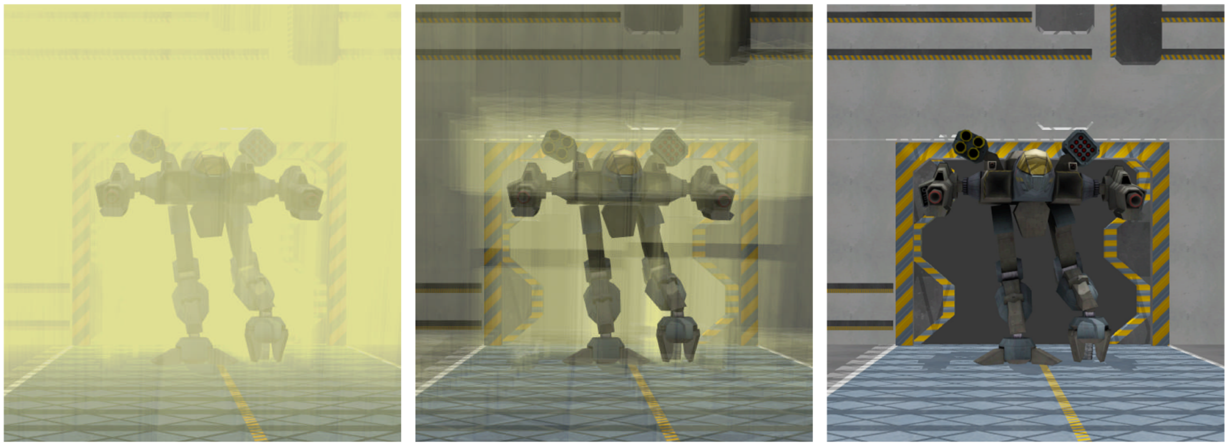

Figure 1. *Fill reduction using CC shadow volumes: On the left, standard shadow volumes are rendered in transparent yellow. The middle image shows CC shadow volumes. The final rendered scene is shown on the right. In this environment, CC shadow volumes reduced the fill requirements by up to 7 times over standard shadow volumes.*

1 Introduction

Shadow volumes are a popular technique for generating shadows [Crow 1977]. They are being used more and more in games such as Doom 3. The technique involves rendering shadow volumes to the stencil buffer. Since the size and number of shadow volumes are often quite large, the rendering can become fill-bound. We present an acceleration technique that reduces the size and number of shadow volumes when possible, thereby reducing rasterization time.

The key to our technique is to eliminate parts of the shadow volumes that do not make a contribution to the final image. In the first stage we *cull* the redundant shadow volumes arising from occluders that are completely in shadow. The remaining shadow volumes often contain large regions empty space within which no shadows will be generated. Our second stage *clamps* the shadow volumes to the regions that actually contain shadow receivers, which reduces rasterization costs. Culled and clamped (CC) shadow volumes can lead to significant savings in both fill and rendering time.

2 Algorithms

We present two algorithms for accelerating of shadow volume rendering:

1. Shadow Volume Culling: Using a variation of the shadow culling algorithm presented in Govindaraju et al. [2003], we eliminate the shadow volumes that are completely in shadow. The algorithm uses occlusion queries on the GPU to test the visibility of each occluder in the light view. If an occluder is not visible, its shadow volume must be in shadow and can be removed.

2. Shadow Volume Clamping: We use two different techniques to determine the occupied regions of the shadow volumes: continuous and discrete clamping.

Continuous Clamping. In the light's view we determine which objects lie behind each occluder. These objects are shadow receivers. To compute the occupied intervals along a shadow volume we merge the depth bounds of the receivers lying within the volume. We utilize temporal coherence to accelerate the calculation of the set of shadow receivers that lies within each shadow volume.

Discrete Clamping. In discrete clamping we partition the view frustum into slices using a fan of planes that pass through the eye. The planes partition each shadow volume into discrete regions. We check whether each region contains objects using the GPU. First, we render a slice of the scene using a pair of clipping planes. Using and occlusion query, we test each region formed by the intersection of a shadow volume with the slice to determine if the region contains objects. After testing all the slices we know which regions of the shadow volume we should render.

The two algorithms have complimentary advantages and disadvantages. The continuous algorithm clamps precisely to the bounds of the contained shadow receivers, but can overestimate the size of a shadow volume when only a small part of the receiver lies inside it. The discrete algorithm clamps only to regions that are truly occupied, but the bounds on the region are only as accurate as the region discretization.

3 Results

We have tested our algorithms on a 2.8 GHz PC with an NVIDIA GeForce 5950FX graphics card. In a dynamic environment composed of 100K triangles and a moving light source, we have observed up to a 7 times reduction in fill and a 4 times speed-up in shadow volume rendering time by using CC shadow volumes over standard shadow volumes. By carefully arranging CPU and GPU computations we can compute shadow volumes for the next frame while the GPU is rendering the current scene, effectively hiding the overhead associated with continuous clamping. For our test scenes we have observed a 2-2.5 speed-up in overall frame rate.

This work was supported in part by ARO Contract DAAD19-02-1-0390, NSF awards ACI 9876914, NSF ACR-0118743, ONR Contracts N00014-01-1-0067, Intel, and an NSF Fellowship.

References

Crow, F. C. 1977. Shadow algorithms for computer graphics. *ACM Computer Graphics 11*, 3, 242–248.

Govindaraju, N., Lloyd, B., Yoon, S., Sud, A., and Manocha, D. 2003. Interactive shadow generation in complex environments. *Proc. of ACM SIGGRAPH/ACM Trans. on Graphics 22*, 3, 501–510.