Voxel-Space Ambient Occlusion

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

Figure 1: *Image of a complex scene rendered using voxel-space ambient occlusion*

1 Introduction

Ambient occlusion adds realism to a scene, as it approximates the darkening in cavities and contact shadows. This occlusion information is frequently used as a coarse global illumination approximation. In this poster, we introduce a new screen space technique to approximate ambient occlusion using screen space scene voxelization.

Prior work computes ambient occlusion in screen space [Mittring 2007] using depth buffer information, but this lacks spatial location of world geometry. Reinbothe et al. [2009] used a voxel based scene, but used ray marching to approximate ambient occlusion.

We present a GPU-based ambient occlusion technique that uses voxels to represent spatial location of geometry and compute occlusion in voxel-space. No precomputation is required as we generate voxels in real-time using the single pass voxelization technique by Eisemann and Décoret [2008].

2 Our Technique

We use a two pass technique to approximate ambient occlusion by filtering in voxel-space.

In the first pass, we compute voxels in screen space. The voxels are stored in a 32-bit RGBA integer buffer with each bit representing a single depth bin, resulting in 128 depth samples per pixel in screen space. Currently, the space between the near and far plane is divided into 128 equal divisions in the z-direction. Presence of geometry in a spatial location sets a voxel bit representing it.

In the second pass, we sample the surrounding area of each pixel in voxel space to approximate ambient occlusion. Ambient occlusion at a given location is defined by the following equation:

$$AO = \frac{1}{\pi}\int_{\Omega} V(x,\omega)[\omega.n]d\omega$$

Here $V(x,\omega)$ is the visibility function which has value zero if there is no geometry in direction ω. n is the normal at the given location and $\int_{\Omega}$ represents integration over the hemisphere about the location. $[\omega.n]$ represents a dot product between ω and n.

We try to approximate this equation by reducing it to a sum over voxel bits surrounding the pixel. We use voxel bits to approximate the visibility function, and an unset voxel bit represents zero visibility in a given location. For a given pixel, we initially determine the voxel it represents, and sample the voxel buffer in a cubic area around it.

The ratio of set bits with respect to the total number of bits sampled is determined to approximate the occlusion for the respective pixel. Like screen space ambient occlusion, unoccluded areas are a medium grey in color and edges are a lighter grey. We plan to correct this by sampling over a hemisphere in the direction of the normal.

3 Results and Future Work

Our early prototype runs the scene in Figure 1 at approximately 5fps, using 1616 samples per pixel (sampling a 7x7x33 area around it). The low frame rate is due to added additional texture lookups for determining a good sampling volume.

Lower number of samples results in banding artifacts, which we plan to address by sampling only in an area in the direction of the normal associated with the pixel. To make our technique more robust, we plan to use voxel ambient occlusion for nearby geometry, but revert to screen space ambient occlusion for distant geometry. We also plan to extend voxel-space ambient occlusion to use screen space multiresolution techniques.

References

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

MITTRING, M. 2007. Finding next gen: Cryengine 2. In *ACM SIGGRAPH course notes, course 28*, 97–121.

REINBOTHE, C., BOUBEKEUR, T., AND ALEXA, M. 2009. Hybrid ambient occlusion. *EUROGRAPHICS Areas Papers*.

*e-mail: rpenmats@cs.uiowa.edu
†e-mail: gbnichol@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