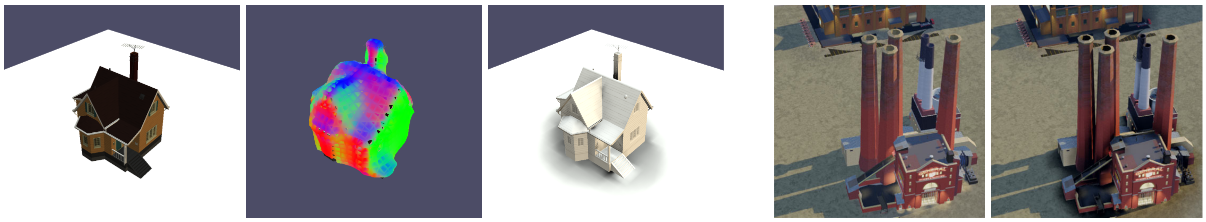
Fast Generation of Directional Occlusion Volumes

Andrew Willmott*



Left: Original model, resulting directional occlusion field, evaluated AO term. Right: scene without and with AO term.

1 Introduction

Ambient occlusion is an important visual cue in video games. (In fact it has been shown to be more important to 3D shape discrimination than direct lighting.) For static scenes, the standard practice is to 'bake' occlusion generated by an offline tool into per-vertex attributes or textures. For scenes with moving or animated models, approaches range from the use of simple polygonal cards for contact shadows to the more recent screen space ambient occlusion (SSAO) algorithm, which runs as a post-process on depth and/or normal buffers. Because pre-baked occlusion is coupled with texture or mesh resolution, storage can be significant, making it best suited to static large-scale occlusion. SSAO on the other hand is best suited to dynamic small-scale occlusion.

Ambient occlusion fields, proposed by Kontkanen and Laine [2005], and extended to volumes by Malmer et al. [2005], provide an intermediate solution between pre-baked AO and SSAO. Occlusion for a particular model is precomputed over a surrounding volume, and then evaluated at render time over any currently contained surfaces. The volumes can be relatively low resolution, are fast to evaluate, and retain SSAO's independence from scene and occluder complexity. Their use in games has previously been described by Hill [2010] and Reed [2012].

Typically AO volumes are generated offline, by casting a significant number of rays for each cell against the model, to find an average occlusion term. This provides high quality results, but is slow, sometimes taking several minutes depending on the parameters chosen. This leads to friction in art iteration, and makes the approach unfeasible when geometry is generated at runtime.

2 Fast AO Generation

Evaluating occlusion at a cell is expensive because it can depend on any other cell in the volume. The related problem of calculating the distance field over a volume has the same issue. A fast solution to the 2D version of this problem was originally proposed by Danielsson [1980], and involves making four diagonal passes through the image, each one traversing cells in order by x and then y, with the direction dependent on the diagonal chosen. A key observation is that for each new cell processed, the immediate neighbours in the corresponding quadrant will already have been visited. Thus the minimum distance to the nearest occupied voxel in that quadrant can be propagated in such a pass, and the final distance field is then the minimum of the four partial results. The algorithm is trivially extended to 3D, with eight passes needed instead of four.

Our algorithm uses the same set of diagonal passes, but propagates occlusion in the given direction instead of distance. The occlusion for each cell traversed is approximated as a function of its three direct neighbours, using an approach similar to the probabilistic composition of spherical caps. Although the intermediate results can then be combined into a scalar occlusion term, we find it advantageous to store instead a directed solid angle. Using this helps avoid self-occlusion of model surfaces, and allows occlusion to vary properly according to the occluded surface normal. We use a fast approximation to the full formula for clipping and projecting the solid angle onto the local surface plane.

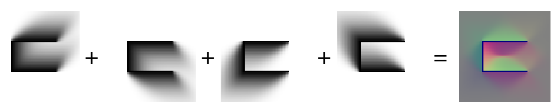


Figure 1: *Four diagonal sweeps combine to produce directional AO.*

The algorithm does require that the model is first rasterised into the volume, e.g., by half-plane scan conversion. Rather than binary occlusion, we store fractional occlusion in each of the directions.

Our approach turns out to be similar to that described by Singh et al. [2009], though their simpler occlusion function leads to over-darkening along diagonals, and the evaluation approach is different.

3 Results

We find limiting volume resolutions to 32^3 cells (128K uncompressed) provides satisfactory results, and that at these resolutions AO volume generation takes 1ms or less, fast enough that it can be applied as part of the load process for a given model, or once player modification has been completed. If desirable, distance fields can be generated as part of the same process (e.g., for collision purposes), and the occlusion term can also be repurposed to provide an initial bounce of the light incident on models.

References

DANIELSSON, P. E. 1980. Euclidean distance mapping. In *Computer Graphics and Image Processing*, 227–248.

HILL, S., 2010. Rendering with Conviction. GDC 2010.

KONTKANEN, J., AND LAINE, S. 2005. Ambient occlusion fields. In *Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games*, ACM Press, 41–48.

MALMER, M., MALMER, F., ASSARSSON, U., AND HOLZSCHUCH, N. 2005. Fast Precomputed Ambient Occlusion for Proximity Shadows. Rapport de recherche RR-5779, INRIA, Dec.

REED, N., 2012. Ambient occlusion fields and decals in Infamous 2. GDC 2012.

SINGH, M., YUKSEL, C., AND HOUSE, D. 2009. Fast occlusion sweeping. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I*, Springer-Verlag, Berlin, Heidelberg, ISVC '09, 167–178.

*Maxis/Electronic Arts.

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008