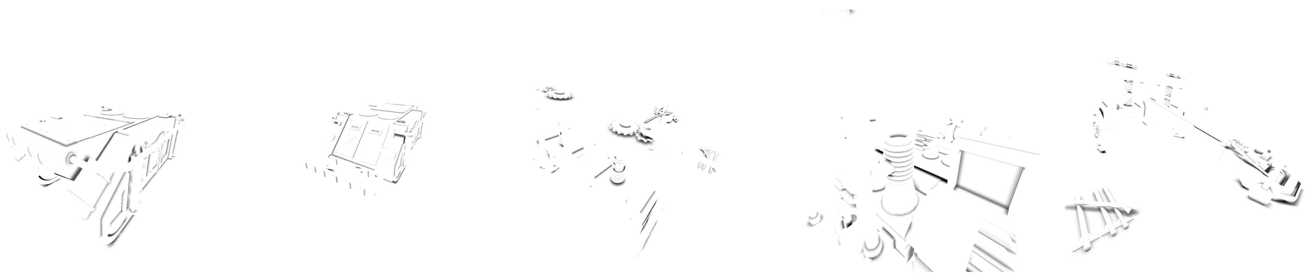
Dynamic Ambient Occlusion from Volumetric Proxies

Andrew Cox[*] and Jan Kautz[†]
University College London, United Kingdom



Abstract

Real-time applications require simple techniques that enable predictable high performance rendering, but illumination methods for GPUs tend either to be complex and fragile on the one hand or very limited and offering poor visual quality on the other. The addition of a plausible global look to an application's lighting impacts users' perceptions of its realism, and on the spectrum of existing approaches, screen space ambient occlusion (SSAO) lies at the simplest and most predictable end. We have extended screen space ambient occlusion by replacing its depth buffer comparisons with a sampling of a volumetric discretization of the scene, while still evaluating it in an image-space post-process and thus retaining its predictable performance. We show improvements in quality over depth buffer based alternatives at a reasonable additional cost.

CR Categories: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

Keywords: Slicemap, ambient occlusion, global illumination

1 Introduction

Ambient Occlusion (AO) is an approximation to global illumination which is not physically-based but gives a look to renderings which is anecdotally well-liked at a far lower cost than a true global illumination solution.

While not a true global method, AO does require information about a local region of the scene around each shading point. That makes it a challenge to map to the independent, triangle-at-a-time and fragment-at-a-time model that GPUs are built to accelerate. In its classic presentation AO relies on the primitive of tracing rays through a full geometric representation of the scene. Modern variants that run on GPUs have captured a per-frame implicit surface as a heightmap by rendering it from the current point of view and writing nearest-z to a buffer. They then perform texture lookups in this buffer in the region of a shade point, using a binary in/out test on the implicit surface or the distance to intersection with it in place of a scene ray intersection. This division into separate capture and lookup stages plays to the forward rendering strengths of GPUs but it is presently limited by the use of a single-valued heightmap as scene proxy, with the result that both false positives and false negatives are reported for occlusions. We will present a method that shares this separation but which captures a richer volumetric proxy for the scene and thus allows a closer approximation to the effects that the ray-traced progenitors provide.

Our main contributions include the following. Firstly, we provide a method for employing a volumetric discretization in dynamic ambient occlusion that is both faster and higher-quality than previous attempts. Secondly, we show both that the additional information in our volumetric representation allows us to display effects that SSAO cannot and that we can avoid its characteristic artifacts. Thirdly, we demonstrate that that information can be captured at a lower cost than previous methods based on depth peeling or rendering from multiple viewpoints. Finally, we show that a strong recent contribution to the theoretical basis of volumetric AO can be unified with a practical data structure that is a truer volumetric proxy for a scene.

2 Ambient Occlusion from a Discrete Volumetric Proxy

We use a discrete volumetric data structure known as a *Slicemap* that was originated by Eisemann and Décoret [Eisemann and Décoret 2006], and which stores a single bit per voxel. Its chief strength is that it can be built in a single geometry pass simply by rasterizing on the GPU. We employ a solid Slicemap as a proxy for scene geometry in an AO pass, thus allowing the substitution of complex ray/geometry traversals with simple texture fetches to evaluate point in/out with respect to a model. For watertight models we generate a solid Slicemap where object interiors are filled. A solid voxelization has the benefit of avoiding the gaps between voxels that afflict rasterization-based surface voxelizations.

Sampling directly from the Slicemap has bandwidth implications since with 128 bit texels, each texel load will bring in large amounts of data that is irrelevant to the shading of a particular point. Running at 60Hz in a 1024^2 window would require between 48 GiB and 96 GiB (50 GiB expected) per second of bandwidth just to evaluate AO. We considered a number of options for reducing this cost and went with the simplest: we generate a linear format with 8 bit texels. The shader to achieve this simply loads a texel, masks out

* andrew.cox@cs.ucl.ac.uk
† j.kautz@cs.ucl.ac.uk

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007

bytes from it, and then routes them to eight render targets.

We adopt a dense sampling strategy in our AO pass, loading all voxels within $R^{\tan}$ of a shading point's tangent sphere center. To make this viable, for each scene we choose a Slicemap resolution that results in $R^{\tan}$ equaling four Slicemap units. That gives us 280 voxels, which is enough to allow smooth gradients and requires either 52 or 104 texture loads per sample point. This compares favorably to Reinbothe et al. [2009] who must do up to 256 texture loads for their preferred parameter set of 8 ray steps in 32 directions.

References

EISEMANN, E., AND DÉCORET, X. 2006. Fast scene voxelization and applications. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM SIGGRAPH, 71–78.

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