

Volumetric Obscurrence

Bradford James Loos*
University of Utah

Peter-Pike Sloan†
Disney Interactive Studios

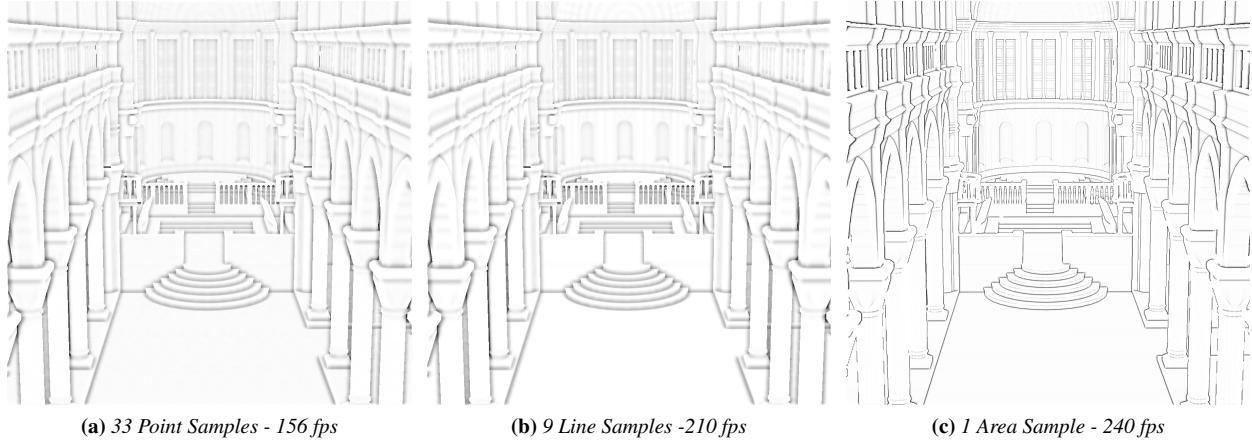


Figure 1: Demonstration of the similarities between Screen Space Ambient Occlusion and Volumetric Obscurrence

Abstract

Obscurrence and Ambient Occlusion (AO) are popular techniques in both film and games that model how ambient light is shadowed. While it is largely a solved problem for static scenes, for dynamic scenes it is still difficult to compute at interactive rates. Recent attempts to compute AO in screen space for dynamic scenes either have poor performance or suffer from under-sampling problems. We formulate the problem as a 3D volumetric integral, which maps more naturally to graphics hardware. This integral can be solved using line samples to improve the under-sampling problems that plague other techniques. Following the idea of line integrals to its logical conclusion, we show results using area samples that use a simple statistical model of the depth buffer that allows us to use a single sample. We also discuss strategies for generating point, line, and area sample patterns along with ways to incorporate the surface normal into the volume obscurrence calculation.

CR Categories: I.3.3 [Picture/Image Generation]

Keywords: Ambient Occlusion, Shadows, GPU, Interactive

1 Introduction

Generating realistic images is difficult, doing so in real-time even more so. There are methods that attempt to simulate a more complex model for ambient light such as Obscurrence [Zhukov et al.

1998] and Ambient Occlusion (AO) [Landis 2002]. These techniques give a softer and more realistic look while providing important contact cues. While it is straightforward to implement these techniques for off-line rendering or static objects and scenes, doing so for dynamic objects has proven to be difficult. Recent games [Mittering 2007; Filion and McNaughton 2008] use screen-space techniques, but they suffer from performance and under sampling problems. This paper builds on those techniques, presenting a method that uses line samples that suffer less from under-sampling than point samples along with an area sampling technique that can generate plausible results without under-sampling issues.

Volumetric Obscurrence (VO) at a point P is defined as the integral of an occupancy function around P times a compact kernel. The occupancy function is zero for points inside an object and one otherwise. While this does not correspond to any physical process, or special case of global illumination like AO, it generates related imagery. Previous work is effectively computing this integral by point sampling the volume[Mittering 2007; Filion and McNaughton 2008]. VO can be computed more efficiently by using line samples or by querying a simple statistical model of the scene's depth buffer. This effectively samples areas of the screen and integrates them against a volumetric piece of the integral instead of using point samples to estimate the integral.

2 Related Work

Obscurrence [Zhukov et al. 1998], and Ambient Occlusion (AO) [Landis 2002] both model a visibility term for constant illumination. Accessibility [Miller 1994] is a related technique, where the scene is colored based on the radius of the largest sphere that can touch a given point. While AO only accurately models the shadowing of ambient light, Precomputed Radiance Transfer (PRT) [Sloan et al. 2002; Lehtinen 2007] can extend this to more general lighting environments, but requires a precomputation and is only practical for very smooth lighting represented using spherical harmonics. Several recent papers [Ren et al. 2006; Sloan et al. 2007] have shown techniques that enable soft shadows from distant low fre-

*e-mail: loos@cs.utah.edu

†e-mail: Peter-Pike.Sloan@disney.com

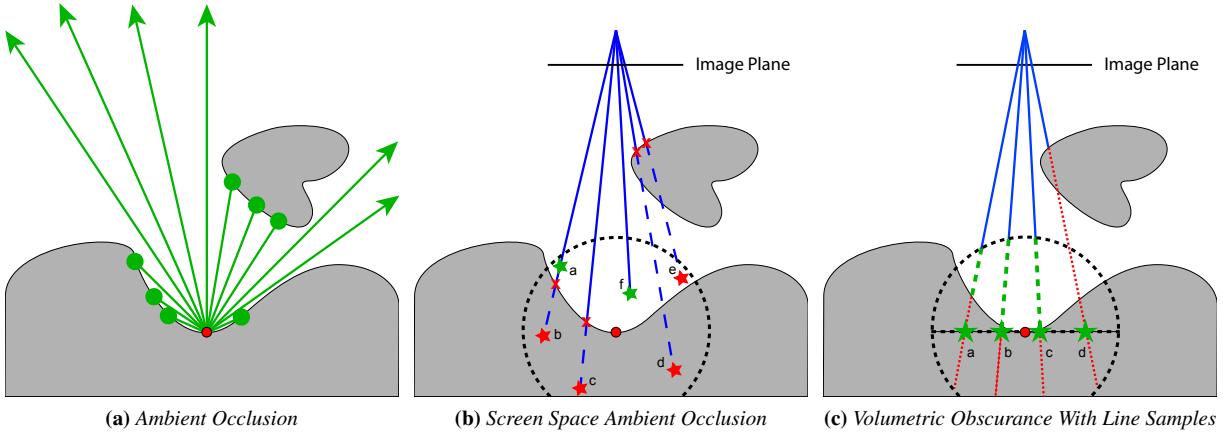


Figure 2: An illustration of the three ambient shadowing models. Ambient Occlusion (a) is defined as the ratio of rays emanating from a point on the surface that are able to escape the scene. Volumetric Obscurrence, originally computed using point samples (b), attempts to replicate this by generating a number of samples around a point on the surface and using a ratio of points that fall behind the depth buffer. We compute this volume integral using line samples (c), the visible portions of each line (shown in green) are used to compute the integral.

frequency lights for dynamic scenes by approximating objects in the scene using a moderate number of spheres. However, this model of visibility could not scale to model the fine scale effects as seen in Figure 1. We focus on just ambient visibility, but use a richer model of the scene based on the depth buffer.

For dynamic scenes, several approaches have been taken to generate AO. Bunnell [2005] approximated faces as discs and used techniques from hierarchical radiosity to solve for an AO-like quantity. Multiple iterations of Bunnell’s technique simulate indirect lighting, but may have problems scaling to complex scenes. Another approach is to define the Ambient Occlusion from an object to any point in space efficiently [Kontkanen and Laine 2005; Shanmugam and Arikan 2007], this can be problematic for small objects and makes it difficult to reason about combining independent objects. Concurrent with our work McGuire [2009] analytically computed the AO contribution for polygons in the scene. An approach tailored for characters is to precompute a model of the AO on the character and ground as a function of the joint angles [Kontkanen and Aila 2006; Kirk and Arikan 2007] but this does not address how to combine the AO from disjoint characters or objects.

The approaches most closely related to ours work in screen space and use the depth buffer to compute AO [Mittring 2007; Shanmugam and Arikan 2007; Filion and McNaughton 2008; Bavoil et al. 2008]. Concurrently to our work a similar technique was proposed [Szirmay-Kalos et al. 2009] that includes a novel method to incorporate the affects of the normal. One benefit of these approaches is that execution is independent of scene complexity, depending solely on display resolution. Two techniques [Shanmugam and Arikan 2007; Bavoil et al. 2008] are close to the traditional formulation, scanning the frame buffer in the neighborhood of a point. Other techniques [Mittring 2007; Filion and McNaughton 2008] have a more volumetric feel, but no formal mathematical model, these are the closest to our technique. All of these approaches suffer from under-sampling the scene, resorting to working at a reduced resolution, randomizing the samples and then blurring the scene [Mittring 2007; Filion and McNaughton 2008; Bavoil et al. 2008] to maintain performance. Our line sampling approach computes the integral more analytically at a set of 2D points but still sometimes requires an edge aware blur. A related technique [Smedberg and Wright 2009] reprojects previous frames AO estimates [Nehab et al. 2007] and blends them with the current frame.

Another approach that gives similar results is to use unsharp masking on the depth buffer [Luft et al. 2006], to enhance some property of the image. Our approach is closer to ambient occlusion because the radius of the effect is fixed in object space. To mimic this behavior using filtering of the depth buffer would require a more complex spatially varying blur function. Our technique can be used to create more stylized imagery as well, for example higher weight lines at boundaries of objects as used in technical illustration [Gooch et al. 1999]. But the primary focus is closer to how ambient occlusion is used in film and games.

Variance Shadow maps [Donnelly and Lauritzen 2006; Lauritzen and McCool 2008] use a statistical model of depth, but for the purpose of computing soft shadows. Our area sampling technique for volumetric obscurrence uses a similar statistical model, but the query is much more involved. A similar method was described in [Pesce 2009].

A recent paper [Ritschel et al. 2009] uses SSAO-like techniques to approximate indirect lighting along with a directional model of visibility. While the results are visually pleasing, the technique appears to be far too costly for current game consoles and is still based on point queries of the depth buffer. It would be interesting to try and extend this work using line or area sampling.

3 Obscurrence and Ambient Occlusion

Obscurrence is defined as:

$$A(P) = \frac{1}{\pi} \int_{\Omega} \rho(d(P, \omega)) \cos \theta d\omega$$

Where Ω is the hemisphere, ρ is a fall-off function, d is the distance to the first intersection, θ is the angle between the normal at P and the direction ω . The fall-off function should start at zero and become one at some fixed distance, this enables rays to be traced with a limited extent. Ambient Occlusion is a special case of obscurrence where the fall-off function is zero for any value besides ∞ (See Figure 2a). Both these techniques model ambient illumination. AO is the transfer coefficient that maps direct lighting to outgoing radiance for a diffuse surface[Sloan et al. 2002]. Sometimes the surface normal is not included, which makes AO simply

the DC projection of the visibility function, which can be used with a triple product to light the surface.¹

The primary difficulty in mapping these techniques to the GPU is that the queries are over ray directions, which does not interact well with the traditional rasterization framework. These techniques also have the property that they are not separable, which makes combining the AO from multiple objects more difficult.

4 Volumetric Obscurrence

Volumetric Obscurrence (VO) is a related quantity defined as:

$$V(P) = \int_X \rho(d(P, x)) O(x) dx$$

Where X is a 3D neighborhood around P , and O is an occupancy function, zero if there is matter at x and one otherwise. The fall-off function ρ will be defined to be one at P and possibly falling off to zero at a fixed distance. We have experimented with a constant function and a quadratic function that falls off to zero at a finite distance but the differences are fairly subtle and did not seem to warrant the extra cost. For the remainder of this paper we assume ρ is a constant function.

VO is a volumetric generalization of obscurrence, and will have the same results when any ray originating from P intersects a single solid surface. The attenuation function used in obscurrence can be thought of as the integral of the function ρ used here from t to the extent of the kernel. There are two primary benefits to this formulation: the contribution of solid non-interpenetrating objects is separable, which makes it easier to combine dynamic parts of the scene with static parts that could pre-compute VO and computing this integral is more amenable to GPUs. Both the Crytek [Mittring 2007] and Blizzard [Filion and McNaughton 2008] techniques effectively use this formulation.

4.1 Line Sampling

Instead of using point sampling to numerically compute the integral [Mittring 2007], we use analytic computations in depth and numeric in the other two dimensions. Given a sphere of constant size in object space centered at P we compute the analytic integral of the occupancy function times the depth extent of the sphere at each sample point on a disk.

Given the depth d at a 2D point the occupancy function $f(z)$ is:

$$f(z) = \begin{cases} 1 & : z \leq d \\ 0 & : z > d \end{cases}$$

$f(z)$ is integrated against the function ρ which is defined to be constant over the unit sphere centered at point P . Because ρ is constant this integral is simply:

$$\int_{-z_s}^{z_s} f(z) dz = \max(\min(z_s, d_r) + z_s, 0)$$

Where $z_s = \sqrt{1 - x^2 - y^2}$, d_r is the depth of pixel (x, y) mapped into the unit sphere's coordinate system, and x and y are carefully sampled points on a unit disk. This is visually depicted in Figure 3, where the integral of the step-function against the z interval is shown. The samples on the disk allow an estimation of the full volumetric obscurrence integral. Line sampling is more efficient due

¹The triple product is extremely simple if one of the terms is constant, since that term can be factored out of the integral.

to the fact that all 2D samples will project to distinct points in the z-buffer whereas two point samples may project to the same location (see points a and b in Figure 2b).

A differential change of the camera will cause a differential change of the VO as long as the depth function is continuous in the neighborhood of the sample. This is not true when using point sampling techniques.² This is particularly evident even when blurring the results using a small number of point samples. While line samples perform better than point samples, unless the radius is fairly small it still needs an edge aware blur pass.

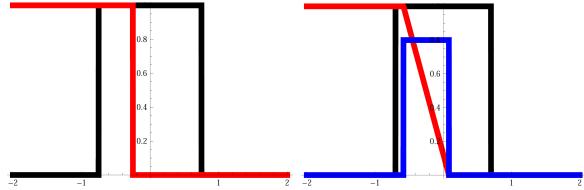


Figure 3: Depth from P is along the horizontal axis. Sphere extents (z_s) are in black and occupancy function is in red. For line sampling (left) the occupancy function is a step function (see Section 4.1) but falls off smoothly for area sampling (right). The statistical depth model is shown in blue.

4.2 Area Sampling

Instead of point or line sampling the depth buffer, a statistical representation over the area associated with each sample can be built. This method generates adequate images using a single sample if darkening creases and corners is all that is desired. Unfortunately, our area sampling method is not competitive performance wise when using multiple samples.

The simplest statistical model is the mean and variance of the depth values over a given area. As in Variance Shadow Maps [Donnelly and Lauritzen 2006] this can be computed from the first two moments of the depth buffer. The average depth, $M_1 = E[z]$, and average squared depth, $M_2 = E[z^2]$, can be filtered because the expectation operator E is linear. The mean μ is simply M_1 and the variance σ^2 is $E[z^2] - E[z]^2$, where σ is the standard deviation.

If a large radius or small number of area samples are going to be chosen, reconstruction artifacts from tri-linear interpolation can occur. This can be alleviated by computing a Gaussian blur on the top level of the mip-map, and carefully placing the taps when down-sampling to compensate.

The depth buffer is sampled at each pixel to determine the screen space extent, but all quantities are mapped to a unit sphere centered at the origin before the integral is computed. When computing with multiple samples only a single \log_2 has to be computed, the logs for the fractional area covered by a sample are precomputed and added to the log for the entire sphere to determine the LOD each time the depth model needs to be sampled.

This mip-map is sampled based on the area of a given sample to generate a distribution of depths over that area. Given this distribution we compute a simple integral using the x, y coordinates of the disk at the center of the area sample. In Figure 3 the depth distribution is modeled as a box function³, so the visibility function now

²Filion [2008] uses a fall off function that is only used when a sample is occluded, this is closer in spirit to our thickness model.

³The convolution of the distribution with the visibility function results in the complimentary error function for a Gaussian which has no closed form solution. We approximate the Gaussian with a box function with a

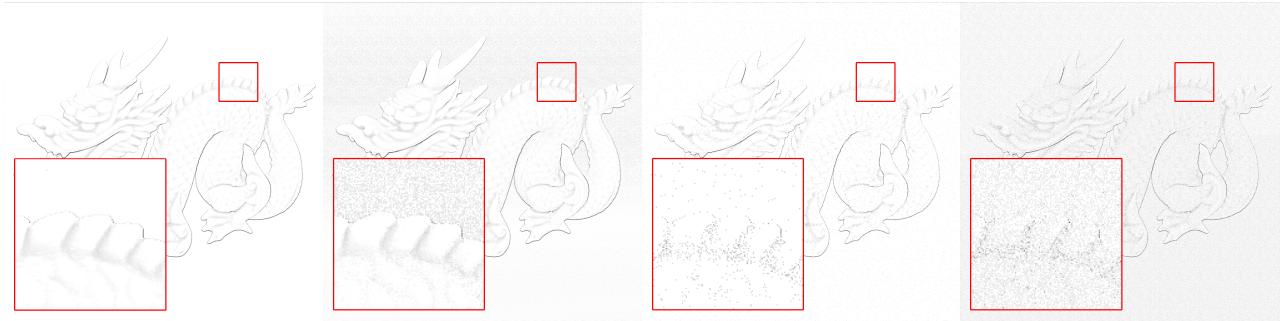


Figure 4: Some point selections turn out to be much better than others. On the left, you can see that nine line samples create much less noise than ten line samples. On the right you can see 12 point samples create much less noise than 13 point samples.

has two regions that can contribute to the integral, the leading constant part and linear part. Below are the equations for computing the integral with a constant attenuation function. z_{min} and z_{max} are the entry and exit points of the sphere at given point, z_0 and z_1 are the extents of the linear part of the visibility function intersected with the sphere extents and $a = \frac{-1}{2\sigma}$ and $b = -a(\mu + \sigma)$ are the coefficients of the line from the integrated visibility function:

$$V_c(z_{min}, z_{max}) = z_{max} - z_{min}$$

$$V_c(z_0, z_1, a, b) = a(z_1^2 - z_0^2)/2 + b(z_1 - z_0)$$

4.3 Thickness Model

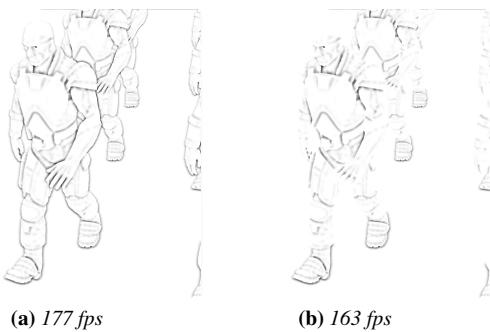


Figure 5: On the left thickness is not used, on the right it is. Notice how the silhouette edges are no longer shadowed when the thickness model is used.

One of the limitations of all screen space algorithms is that there is no information about what happens behind the depth buffer. One simple approximation is to assume that all surfaces have a fixed depth. This can be done by replacing the step-function in the convolution above with some model of occlusion behind the front most surface. Figure 5 shows a comparison with and without the thickness model. To compute VO using the thickness model all that is required is to compute the integral twice, once with the usual visibility function, and again with $1 - \text{visibility function shifted back by the fixed thickness}$. The results are simply summed.

4.4 Incorporating the Surface Normal

If a surface normal is available it can be used to restrict the sampling to a hemisphere instead of a sphere. For point samples [Filion 2008 and McNaughton 2008] this has been done by reflecting the points that are underneath the hemisphere defined by the surface normal. Line and area sampling can be modified to compute the depth of the plane defined by the surface normal⁴ at the x, y coordinates for the given sample, simply clamp the evaluation of the integral with this depth. Figure 6 shows a comparison between using and not using the normal. Using the normal allows the capture of finer scale details, even with a moderately large radius. An intriguing alternative was concurrently proposed in [Szirmay-Kalos et al. 2009], where the sphere sampled is the largest sphere contained by the hemisphere centered at the point. This also incorporates the cosine weighted fall-off commonly used in AO techniques.

and McNaughton 2008] this has been done by reflecting the points that are underneath the hemisphere defined by the surface normal. Line and area sampling can be modified to compute the depth of the plane defined by the surface normal⁴ at the x, y coordinates for the given sample, simply clamp the evaluation of the integral with this depth. Figure 6 shows a comparison between using and not using the normal. Using the normal allows the capture of finer scale details, even with a moderately large radius. An intriguing alternative was concurrently proposed in [Szirmay-Kalos et al. 2009], where the sphere sampled is the largest sphere contained by the hemisphere centered at the point. This also incorporates the cosine weighted fall-off commonly used in AO techniques.



Figure 6: Left: using the normal. Right: no normal is used.

4.5 Sample Generation

Considerable time and effort was spent on generating good sample patterns for line, area, and point sampling. Initially, we tried using Lloyd relaxation techniques [Lloyd 1982], Quasi Monte Carlo methods, and generating Poisson distributions using dart throwing. The technique outlined below performed significantly better, particularly at low sample counts.

Solving an electrostatics problem generates good results over the surface of the sphere [Bulatov 1996], we extend this technique to samples inside a sphere and a disk. Since we are computing integrals, the goal is to have the integral over the Voronoi region of every sample to be equal. While the electrostatics solution works well on a surface, in a disk or sphere the particles need to be coerced to stay inside the boundary. Simply adding a charge to the boundary does not work, Gauss's law implies that the net force from that charge will always be zero. Instead we use a $\frac{1}{d^4}$ repulsion force and optimize the boundary charge to push points toward

⁴Since all the computations are done in the space of the unit sphere, the plane equation is just the eye-space surface normal

the weighted centroid of their Voronoi regions periodically. This process is initialized with tens of starting point sets generated using Poisson sampling. The net force is reduced until the error decreases (not applying the force until there is a reduction) and increased otherwise. The Voronoi regions are computed using a discrete Voronoi diagram with 128^3 cells for the volume and 512^2 for the disk. Each disk sample has a weight proportional to the Z extent on the sphere.⁵

As in [Mittring 2007] we randomize our sample sets at every pixel using a random texture. Experiments with various sizes of random textures seem to imply that the algorithm is not sensitive to the size of the texture nor does random texture size seem to have an impact on performance. For line samples we compute a reflection about a random direction in 2D instead of 3D, otherwise randomization of point samples and line samples is treated in the same manner.

It is interesting to note that certain generated sample sets produce much less noise than those around them (see Figure 4). In 3D 6, 12, and 33 samples have extremely low error compared to sets generated with other numbers of samples. In 2D 5, 7, 9, 13, and 16 samples have low error when the center is one of the samples, whereas 12, 13, and 15 samples seem to work optimally if the center is not included. We imagine that these sample sets work better due to symmetries. For example, 6 and 12 samples in 3D closely match the vertices of platonic solids.

5 Results

5.1 Performance

In Table 1 we compare the performance of Crytek’s point sampling method, our line sampling method, and the Horizon Split Ambient Occlusion (HSAO) technique [Bavoil et al. 2008]. These tests were run on a desktop with an NVIDIA QuadroFX 4800 GPU at a resolution of 1024x1024 using full resolution AO buffers.

Geometry	Ambient Occlusion				
	Lines		Points		HSAO
	5	9	12	33	
Dragon	3.39	0.71	1.01	1.18	3.11
Dragon*	3.39	1.06	1.85	2.44	7.36
Cornell	0.45	0.71	1.03	1.31	3.55
Soldiers	2.37	0.54	0.77	1.14	2.32
Knight	0.27	0.59	0.82	0.97	2.52
Average	0.64	0.91	1.15	2.87	39.56

Table 1: Timing Breakdown per pass in milliseconds. The bilateral blur used for each scene required a median time of 2.23 ms. *Using double radii, not included in the average

The difference in visual quality between Crytek’s point sampling using 12 samples and line sampling using 5 samples is negligible. In animation tests, small numbers of samples caused the Crytek method to exhibit temporal aliasing whereas even with as few as 5 line samples the temporal aliasing was much less pronounced. Table 1 also shows that similar visual quality is obtainable much more quickly using line samples.

⁵The disk is an orthographic projection of a sphere, the integral being computed. Each sample will scale its line integral by the integral of all the discrete samples in its Voronoi region.

6 Conclusion and Future Work

Volumetric Obscurrence is an alternative to screen-space Ambient Occlusion that can be efficiently evaluated on GPU’s. Using carefully created sample patterns as few as 5 line samples can generate convincing results. The area sampling technique generates images that are adequate if you just want to highlight creases⁶, but is not competitive performance wise in the general case. However, using a statistical model of the depth buffer alleviates the under-sampling problems that plague prior work and enables an interesting result with only a single sample. Qualities of AO, such as darkening at concave creases, darkening near contact and in fine-scale features are preserved. However some properties of AO-like the size of the ambient shadow at contact, and the effect of thin objects are not preserved. We also show a simple representation of object thickness, but a more complete treatment of thickness is left for future work.

Using a small radius allows you to accentuate fine features, but has more subtle contact and depth discontinuity shading. Using a larger radius makes some of the effects more pronounced, but loses the fine scale features. Combining two radii is simple (see Figure 8) - using either a product of the two results (or the minimum) produces pleasing images.

Acknowledgements

Anis Ahmad, Pete Ivey, Neil Hutchinson, John Ownby and Peter Shirley for helpful discussion. Data sets are from the NVIDIA and DirectX SDK’s.

References

- BAVOIL, L., SAINZ, M., AND DIMITROV, R. 2008. Image-space horizon-based ambient occlusion. In *SIGGRAPH ’08: ACM SIGGRAPH 2008 talks*.
- BULATOV, V., 1996. Point repulsion newsgroup posting.
- BUNNELL, M. 2005. *GPU Gems 2*. Addison Wesley, ch. Dynamic Ambient Occlusion and Indirect Lighting.
- DONNELLY, W., AND LAURITZEN, A. 2006. Variance shadow maps. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*.
- FILION, D., AND MCNAUGHTON, R. 2008. Effects & techniques. In *SIGGRAPH ’08: ACM SIGGRAPH 2008 classes*, 133–164.
- GOOCH, B., SLOAN, P.-P. J., GOOCH, A., SHIRLEY, P., AND RIESENFELD, R. 1999. Interactive technical illustration. In *Proceedings of the 1999 symposium on Interactive 3D graphics*.
- KIRK, A. G., AND ARIKAN, O. 2007. Real-time ambient occlusion for dynamic character skins. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*.
- KONTKANEN, J., AND AILA, T. 2006. Ambient occlusion for animated characters. In *Rendering Techniques 2006 (Eurographics Symposium on Rendering)*.
- KONTKANEN, J., AND LAINE, S. 2005. Ambient occlusion fields. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*.
- LANDIS, H., 2002. Global illumination in production. ACM SIGGRAPH 2002 Course #16 Notes, July.

⁶Using a single sample generates interesting results, but looks more like some type of NPR shader than AO.

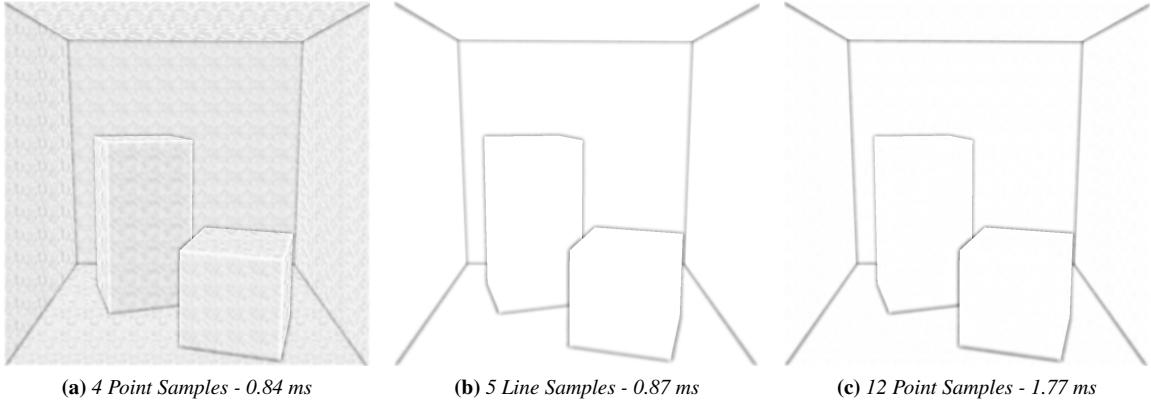


Figure 7: Line samples produce better quality results in less time

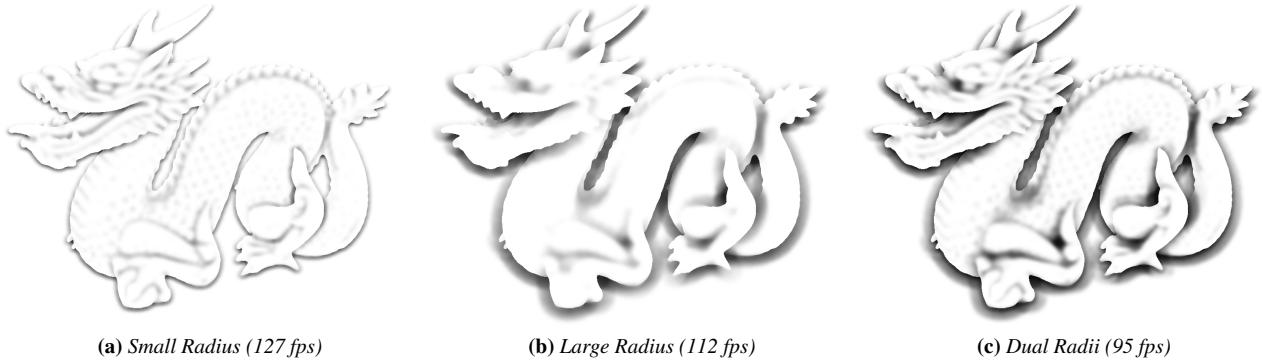


Figure 8: When small radii are used, fine features can be captured but contact shadows and other large scale features are lost. By using dual radii both sets of features can be captured for more pleasing results.

- LAURITZEN, A., AND MCCOOL, M. 2008. Layered variance shadow maps. In *Proceedings of graphics interface 2008*.
- LEHTINEN, J. 2007. A framework for precomputed and captured light transport. *ACM Transactions on Graphics* 26, 4 (Oct.).
- LLOYD, S. P. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* IT-28, 2 (Mar.), 129–137.
- LUFT, T., COLDITZ, C., AND DEUSSEN, O. 2006. Image enhancement by unsharp masking the depth buffer. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*.
- MCGUIRE, M. 2009. Ambient occlusion volumes. Tech. rep., Williams College.
- MILLER, G. 1994. Efficient algorithms for local and global accessibility shading. In *Proceedings of SIGGRAPH '94*.
- MITTRING, M. 2007. Finding next gen: Cryengine 2. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, ACM, 97–121.
- NEHAB, D., SANDER, P. V., LAWRENCE, J., TATARCHUK, N., AND ISIDORO, J. R. 2007. Accelerating real-time shading with reverse reprojection caching. In *Graphics Hardware*.
- PESCE, A. 2009. *ShaderX7*. ch. Variance methods for Screen-Space Ambient Occlusion.
- REN, Z., WANG, R., SNYDER, J., ZHOU, K., LIU, X., SUN, B., SLOAN, P.-P., BAO, H., PENG, Q., AND GUO, B. 2006. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Transactions on Graphics* 25, 3.
- RITSCHEL, T., GROSCH, T., AND SEIDEL, H.-P. 2009. Approximating dynamic global illumination in image space. In *Proc. ACM Symposium on Interactive 3D Graphics and Games 2009*.
- SHANMUGAM, P., AND ARIKAN, O. 2007. Hardware accelerated ambient occlusion techniques on gpus. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*.
- SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics* 21, 3.
- SLOAN, P.-P., GOVINDARAJU, N. K., NOWROUZEZHRAI, D., AND SNYDER, J. 2007. Image-based proxy accumulation for real-time soft global illumination. In *Proceedings of the Fifteenth Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, 97–105.
- SMEDBERG, N., AND WRIGHT, D. 2009. Rendering techniques in gears of war 2. In *Game Developers Conference*.
- SZIRIMAY-KALOS, L., UMHENHOFFER, T., TOTH, B., SZECSEI, L., AND SBERT, M. 2009. Volumetric ambient occlusion. *IEEE Computer Graphics and Applications*, Preprint – to appear.
- ZHUKOV, S., INOES, A., AND KRONIN, G. 1998. An ambient light illumination model. In *Rendering Techniques '98*.