# Image-Space Horizon-Based Ambient Occlusion

Louis Bavoil Miguel Sainz Rouslan Dimitrov

NVIDIA Corporation

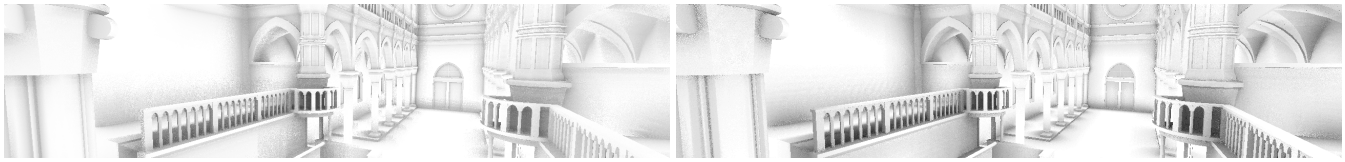

Figure 1: Ambient occlusion without any shading. (left) 67 MPixels/s with our horizon-based algorithm. (right) 15 MPixels/s with ray marching and 4 rays per direction. Images rendered with $N_d = 8$ and $N_s = 8$ on GeForce 8800 GTX Ultra.

## 1 Introduction

Ambient occlusion is a lighting model that approximates the amount of light reaching a point on a diffuse surface based on its directly visible occluders. It gives perceptual clues of curvature and spatial proximity. Like [Mittring 2007] and [Shanmugam and Arikan 2007], we propose a real-time ambient occlusion computation as a postprocessing pass mainly based on a depth image from the eye's point of view. This approach requires no scene-dependent precomputations and is applicable to dynamic scenes. Our proposed method does not have the overocclusion issue from [Shanmugam and Arikan 2007] and samples inside the radius of influence, unlike [Mittring 2007].

We use the following form of the ambient occlusion illumination $A$ at a given surface point $\mathbf{P}$

$$A = 1 - \frac{1}{2\pi}\int_{\Omega} V(\vec{\omega})W(\vec{\omega})d\omega \tag{1}$$

where $V$ is the visibility function over the normal-oriented unit hemisphere $\Omega$, which returns 1 if a ray starting from $\mathbf{P}$ in direction $\vec{\omega}$ intersects an occluder and 0 otherwise, and $W$ is a linear attenuation function.

## 2 Horizon-Based Ambient Occlusion

We use a spherical coordinate system with the zenith axis aligned in the view direction $\vec{V}$, azimuth angle $\theta$ and elevation angle $\alpha$ (see Figure 2a). Similarly to horizon mapping [Max 1986], we split the unit sphere by a horizon line defined by the signed horizon angle $h(\theta)$ (see Figure 2b). Assuming that the neighborhood of $\mathbf{P}$ is a continuous heightfield, rays that would normally be traced below the horizon are known to intersect an occluder so the intersection test for these rays can be omitted. Under the continuous heightfield assumption, Equation 1 can be rewritten as:

$$A = 1 - \frac{1}{2\pi}\int_{\theta=-\pi}^{\pi}\int_{\alpha=t(\theta)}^{h(\theta)} W(\vec{\omega})cos(\alpha)d\alpha d\theta \tag{2}$$

We use the linear attenuation function $W(\theta) = max(0, 1 - r(\theta)/R)$ where $r(\theta)$ is the distance between $\mathbf{P}$ and the horizon point in direction $\vec{\omega}$ and $R$ is the radius of influence. In this case,

$$A = 1 - \frac{1}{2\pi}\int_{\theta=-\pi}^{\pi} (sin(h(\theta)) - sin(t(\theta)))W(\theta)d\theta \tag{3}$$

## 3 Image-Space Integration

Our algorithm takes as input per-pixel linear depths and eye-space normals. For every pixel, we compute its eye-space position $\mathbf{P}$ and we integrate Equation 3 using a Monte Carlo approach by sampling

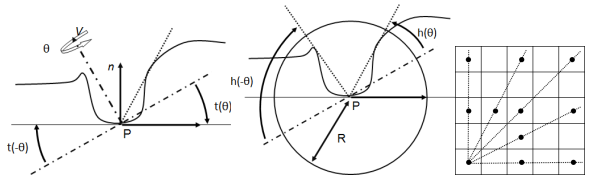


Figure 2: (a) The azimuthal angle $\theta$ around the view vector $\vec{V}$. The tangent angle $t(\theta)$ is the signed elevation angle of the surface tangent vector. (b) The horizon angle $h(\theta)$ is the maximum elevation $\alpha >= t(\theta)$ for which $\vec{\omega}$ is occluded for all $\alpha < h(\theta)$ (c) Example of sample locations for 4 directions and a step size of 2 texels.

directions in image space and stepping on the heightfield stored in the depth image. We pick $N_d$ directions $\theta$ in image space around the current pixel which correspond to directions around the Z axis in eye space ($\vec{V}$ on Figure 2a). For each angle $\theta$, we compute the horizon angle $h(\theta)$ by sampling the depth image along a line segment in image space. The eye-space radius of influence $R$ is projected on the image plane and is subdivided into $N_s$ steps of equal lengths.

For finding the horizon angle in direction $\theta$, we start by computing the tangent angle $t(\theta)$ and intersect a view ray with the tangent plane defined by $\mathbf{P}$ and the surface normal $\vec{n}$. We then step in the depth image in direction $\theta$ and compute $D = S_i - P$, where $S_i$ is the reconstructed eye-space position of a given sample $S_i$. Based on the elevation angles $\alpha(S_i) = atan(-D.z/||D.xy||)$, the horizon angle $h(\theta)$ is $max(t(\theta), \alpha(S_i), i = 1..N_s)$ where $N_s$ is the number of steps per direction. We ignore samples for which $||S_i - P|| > R$. Because $D.z$ must be the exact depth associated with the offset $D.xy$, we make sure to always sample at texel centers. To do so, we snap the texture coordinates of the samples along each direction to the nearest texel centers (see Figure 2c).

To trade banding artifacts for noise, we randomly jitter the step size per pixel and randomly rotate the $N_d$ uniform directions per pixel. Although using a single depth layer typically produces plausible results (see Figure 1), it could be extended by using multiple layers such as the front and back faces.

## References

MAX, N. L. 1986. Horizon mapping: Shadows for bump-mapped surfaces. In *Proceedings of Computer Graphics Tokyo '86 on Advanced Computer Graphics*.

MITTRING, M. 2007. Finding next gen: Cryengine 2. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*.

SHANMUGAM, P., AND ARIKAN, O. 2007. Hardware accelerated ambient occlusion techniques on gpus. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*.