Practical Realtime Strategies for Accurate Indirect Occlusion

Jorge Jimenez[1] Xian-Chun Wu[1] Angelo Pesce[1] Adrian Jarabo[2]
Technical Memo ATVI-TR-16-01
[1]Activision Blizzard [2]Universidad de Zaragoza

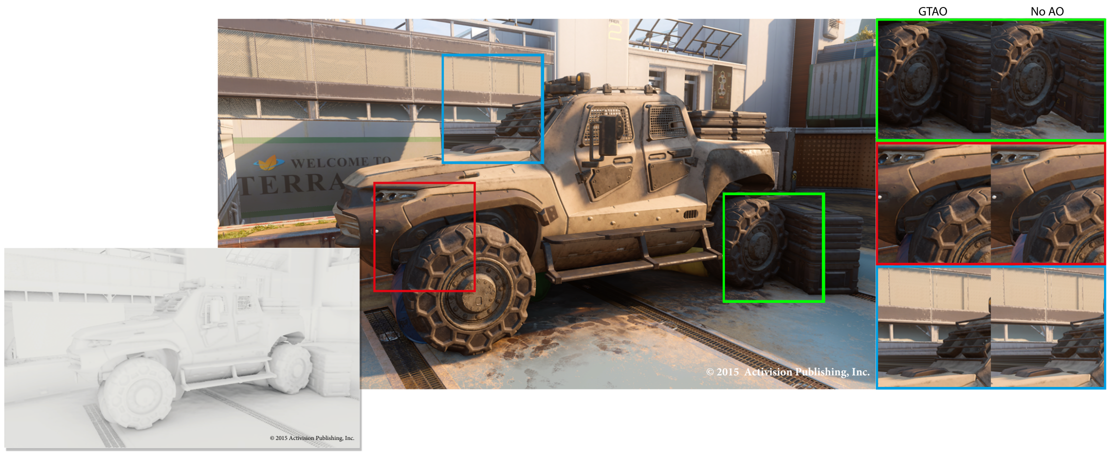
GTAO
No AO
WELCOME TO
TERRA
© 2015 Activision Publishing, Inc.
© 2015 Activision Publishing, Inc.

Figure 1: *Example frame rendered using our* Ground Truth Ambient Occlusion (GTAO). *The insets on the right, show comparison of rendering using GTAO and the input radiance, while the inset on the right shows the ambient occlusion layer. Our technique achieves high-quality ambient occlusion matching the ray-traced ground truth, in just 0.5 ms on a PS4 at 1080p.*

Abstract

Ambient occlusion is ubiquitous in games and other real-time applications to approximate global illumination effects. However there is no analytic solution to ambient occlusion integral for arbitrary scenes, and using general numerical integration algorithms is too slow, so approximations used in practice often are empirically made to look pleasing even if they don't accurately solve the AO integral. In this work we introduce a new formulation of ambient occlusion, GTAO, which is able to match a ground truth reference in half a millisecond on current console hardware. This is done by using an alternative formulation of the ambient occlusion equation, and an efficient implementation which distributes computation using spatio-temporal filtering. We then extend GTAO with a novel technique that takes into account near-field global illumination, which is lost when using ambient occlusion alone. Finally, we introduce a technique for specular occlusion, GTSO, symmetric to ambient occlusion which allows to compute realistic specular reflections from probe-based illumination. Our techniques are efficient, give results close to the ray-traced ground truth, and have been integrated in recent AAA console titles.

1 Introduction

Global illumination is an important visual feature, fundamental in photo-realistic rendering as a large part of perceived scene illumination comes from indirect reflection. Unfortunately, it is in general very expensive to compute, and cannot currently be included in real-time applications without severe simplifications. From these approximations, *ambient occlusion (AO)* is one of the most popular, since it improves the perception of objects' shapes (contrast), and in captures some of the most important effects in global illumination, in particular soft shadows due to close-by occluders. Ambient occlusion is also useful in conjunction with other global illumination algorithms and even when using precomputed (baked) irradiance, as often these effects need to be computed (or stored) at relatively low spatial resolution, thus computing ambient occlusion per pixel can enhance the overall appearance of indirect illumination. Unfortunately, solving the ambient occlusion integral is still expensive in certain scenarios (e.g. 1080p rendering at 60 fps), so approximations have been developed in the past to achieve fast enough performance.

We introduce a new screen-space technique for ambient occlusion, that we call *ground-truth ambient occlusion (GTAO)*. The main goal of this technique is to match ground truth ambient occlusion , while being fast enough to be included in highly-demanding applications such as modern console games. Our technique bases on the horizon-based approach, but using an alternative formulation of the problem. This formulation allows us to reduce significantly the cost of the effect and can still be used to exactly solve the ambient occlusion integral under the assumption that our scene is represented as an height-field (depth buffer). We implement our technique efficiently by using temporal reprojection and spatial filtering to compute a noise-free ambient occlusion solution in just 0.5 ms per frame (on a Sony Playstation 4, for a game running at 1080p).

Based on this formulation, we extend our ambient occlusion solution to model a set of illumination effects generally ignored when using ambient occlusion alone. On one hand, we introduce an approximate technique that computes a very fast correction factor to account for near-field global illumination. This technique is based on the observation that these is a relationship between the local surface albedo and ambient occlusion term, and the multiple-bounces near-field illumination. Following this observation, we develop an efficient, simple and local technique to account for the local illumination that is lost when computing ambient occlusion alone.

Finally, we present a new technique, symmetric to AO, but generalized for arbitrary specular materials, that we call *ground-truth specular occlusion (GTSO)*.We develop its formulation, and present an efficient technique for computing it, based on approximating the visibility as a function of the bent normal and the ambient occlusion at the point. GTSO allows to efficiently computing specular reflection from probe-based illumination, taking into account the occlusion at the surface.

2 Background & Related Work

The reflected radiance $L_r(x, \omega_\mathbf{o})$ from a point x with normal $\mathbf{n_x}$ towards a direction $\omega_\mathbf{o}$ can be modeled as:

$$L_r(x, \omega_\mathbf{o}) = \int_{\mathcal{H}^2} L_i(x, \omega_\mathbf{i}) f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o}) \langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i}, \qquad (1)$$

where $\mathcal{H}^2$ is the hemisphere centered in x and having $\mathbf{n_x}$ as its axis, $L_i(x, \omega_\mathbf{i})$ is the incoming radiance at x from direction $\omega_\mathbf{i}$, $f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o})$ is the BRDF at x, and $\langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+$ This is a recursive operator, that depends on the reflected (and emitted) radiance in all the scene. While many works have focused on solving this problem, it is still too expensive to be solved in highly demanding scenarios such as games. Here we focus on *ambient occlusion* techniques, and refer to the survey by Ritschel et al. [RDGK12] for a wider overview on the field.

Ambient occlusion [ZIK98] approximates Equation (1), by introducing a set of assumptions: *i)* all light comes from an infinite uniform environment light, which might be occluded by the geometry around x; *ii)* all surfaces around x are purely absorbing (i.e. do not reflect any light), and *iii)* the surface at x is diffuse. This transforms Equation (1) into

$$\begin{aligned} L_r(x, \omega_\mathbf{o}) &= L_i \frac{\rho(x)}{\pi} \int_{\mathcal{H}^2} V(x, \omega_\mathbf{i}) \langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i} \\ &= L_i \frac{\rho(x)}{\pi} \mathcal{A}(x), \end{aligned} \qquad (2)$$

where $\mathcal{A}(x)$ is the *ambient occlusion* term at point x, $\frac{\rho(x)}{\pi}$ is the diffuse BRDF with albedo $\rho(x)$, and $V(x, \omega_\mathbf{i})$ is the visibility term at x in direction $\omega_\mathbf{i}$, which returns 0 if there is an occluder in direction $\omega_\mathbf{i}$ closer than a given distance r and 1 elsewhere. Note that previous works [ZIK98,Mit07,BSD08] have modeled this visibility term $V(x, \omega_\mathbf{i})$ as an attenuation function with respect to the distance to the occluder, referring to $\mathcal{A}(x)$ as *obscurance*. This attenuation function was used to create an ad-hoc solution to avoid the typical AO overdarkening produced by ignoring near-field interreflections; we instead introduce a novel formulation for adding this lost light (Section 5) while keeping a radiometricaly correct ambient occlusion term. It is worth to note that there is an alternate definition of ambient occlusion where the foreshortening is ingored: while during the rest of the paper we follow the radiometrically-correct cosine-weighted formulation, in Appendix A we describe our technique under this alternative form.

Screen-Space Ambient Occlusion The ambient occlusion term $\mathcal{A}(x)$ is affected by all the geometry in the scene, and is in general computed via ray-tracing [ZIK98], although point-based approaches more suitable for interactive rendering exist [Bun05, REG$^+$09]. However, this is still too expensive for real-time applications. To avoid the costly three-dimensional full-scene visibility computations, Mittring [Mit07] proposed to move all computations to screen-space, assuming that only the geometry visible form the camera acts as occluder. This is achieved by sampling the GPU's depth map of the scene in a sphere of points around x, and evaluating whether a point is occluded (behind) geometry in the depth map. Since then, several improvements on the idea of screen-space sampling have been made, improving the sampling strategy [LS10, SKUT$^+$10, HSEE15] and reducing noise by filtering [MML12].

Horizon-Based Ambient Occlusion Bavoil et al. [BSD08] proposed to compute the non-occluded region based on the maximum horizon angle at which light can get the light. They transform the integration domain into a set of directions parametrized by ϕ tangent to the surface, and on each of them they computed the total non-occluded solid angle, transforming Equation (2) into:

$$\mathcal{A}(x) \approx \hat{\mathcal{A}}(x) = \frac{1}{\pi} \int_0^\pi \int_{-\pi/2}^{\pi/2} V(\phi, \theta) |\sin(\theta)| \mathrm{d}\theta \mathrm{d}\phi, \qquad (3)$$

where the $1/\pi$ term is for normalization to one (i.e. $\mathcal{A}(x) \in [0, 1]$). Note that here we differentiate between the actual ambient occlusion $\mathcal{A}(x)$ and the approximated screen-space term $\hat{\mathcal{A}}(x)$.

Alchemy Ambient Obscurance [MOBH11,MML12] later improved robustness of the screen-space approach and increased the efficiency of the sampling procedure used. While HBAO is relatively efficient, it is still costly since many samples from the depth map needs to be gathered per pixel when finding the maximum horizon. Timonen [Tim13a] improves over this by performing line sweeps along all the image, which allows him to find the maximum horizon angle for a given direction in constant time by amortizing the sampling along many pixels in the image. Closely related to our work, the same author [Tim13b] proposed an new estimator for ambient occlusion, which is able to match a ground truth solution at small cost, by line-scanning and filtering the depth map, which allows to compute ambient occlusion even for very large gathering radii, covering the entire screen.

Our work improves these works by proposing an efficient formulation of ambient occlusion, without the need of ad-hoc attenuation functions, which saves computation time by allowing very efficient analytical integration. Core to avoid ad-hoc attenuation function is our efficient approximation for including the indirect illumination from the near-field occluders. In addition, all these works assume diffuse surfaces: instead, we generalize the concept of ambient occlusion to non-Lambertian surfaces introducing a technique for specular occlusion.

3 Overview

In this work we have two main goals: On one hand, we aim to have an ambient occlusion technique that matches ground truth results, while being efficient enough to be used in demanding real-time applications. On the other hand, we want to extend the amount of global illumination effects that can be efficiently approximated. The first goal imposes severe limitations in terms of input data, number of passes, and number of instructions. Bounded by these limitations, we develop a technique that works in screen space, taking as inputs only the depth buffer and surface normals (which can be derived from it by differentiation or can be supplied separately), and that can coexist and enhance other sources of global illumination (specifically baked irradiance). In order to achieve the second goal, we relax some of the assumptions done for traditional ambient occlusion. In particular, while we keep the assumption of a white (or monochrome) dome illumination, we relax the assumption of purely Lambertian surfaces, and include diffuse interreflections of near-field occluders.

Removing previous limitations, allows us to transform Equation (2) into:

$$L_r(x, \omega_\mathbf{o}) = (1 - F(\omega_\mathbf{o})) L_i \frac{\rho(x)}{\pi} \mathcal{G}(\mathcal{A}(x)) + F(\omega_\mathbf{o}) \mathcal{L}(x, \omega_\mathbf{o}) \mathcal{S}(x), \qquad (4)$$

where F is the Fresnel reflectance term, $\mathcal{A}(x)$ is our ambient occlusion term (Section 4), that matches the results of the ground truth and that we call *ground-truth ambient occlusion (GTAO)*, $\mathcal{G}(x)$ is the function that, based on the ambient occlusion term, introduces

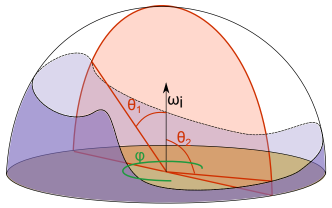

Figure 2: *Diagram of our reference frame when computing horizon-based ambient occlusion.*

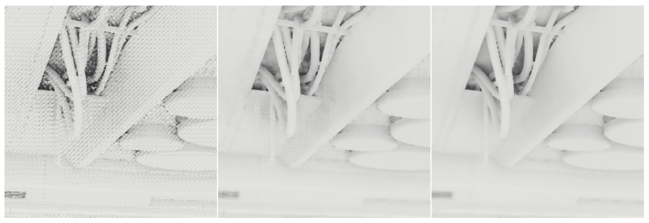

Figure 3: *Comparison between the samples computed on a single pixel (left), adding the spatial occlusion gathering using a bilateral reconstruction filter (middle), and adding the temporal reprojection using an exponential accumulation buffer (right). In each image we use 1, 16 and 96 effective sample directions per pixel respectively.*

the diffuse near-field indirect illumination (Section 5), and $\mathcal{S}(x)$ is the *specular occlusion* term (Section 6), which is multiplied by the preconvolved with the BRDF $\mathcal{L}$. In the following we describe our formulation for each of these terms.

4 GTAO: Ground-Truth Ambient Occlusion

Our formulation of ambient occlusion follows the horizon-based approach of Bavoil et al. [BSD08], but presents a set of key differences that allow efficient computation without sacrificing quality. First of all, we reformulate the reference frame on which the horizons are computed, and therefore the integration domain: we follow Timonen [Tim13a], and compute the horizon angles with respect to the view vector $\omega_\mathbf{o}$ (see Figure 2). This means that the horizons are searched in the full sphere around x, and that the spherical integration axis is set to $\omega_\mathbf{o}$. In practice, this allows us to simplify the formulation, and as we will see later to reduce the number of transcendental functions needed.

The second main difference is that, as opposed to Bavoil's work, our visibility term $V(\phi, \theta)$ is just a binary function, instead of a continuous attenuation as a function of occluder distance (ambient obscurance). Formulating AO this way allows us to compute the inner integral of Equation (3) simply as the integral of the arc between the two maximum horizon angles $\theta_1(\phi)$ and $\theta_2(\phi)$ for direction ϕ. Formulating the integral around $\omega_\mathbf{o}$, and using a binary visibility term transforms Equation (3) into:

$$\hat{\mathcal{A}}(x) = \frac{1}{\pi}\int_0^\pi \underbrace{\int_{\theta_1(\phi)}^{\theta_2(\phi)} \cos\left(\theta - \gamma\right)^+ \left|\sin\left(\theta\right)\right| \mathrm{d}\theta}_{\hat{a}}\, \mathrm{d}\phi, \qquad (5)$$

where γ is the angle between the normal $\mathbf{n_x}$ and the view vector $\omega_\mathbf{o}$, and $\cos\left(\theta\right)^+ = \max(\cos\left(\theta\right), 0)$. This formulation is in fact very important, since allows computing the inner integral $\hat{a}$ analytically while, matching the ground truth ambient occlusion. This means that only the outermost integral needs to be computed numerically, by means of Monte Carlo integration with random ϕ. In the following, we detail how we compute the horizon angles and the inner integral $\hat{a}$.

Computing maximum horizon angles Core to the solution of Equation (5) is to find the maximum horizon angles $\theta_1(\phi)$ and $\theta_2(\phi)$ for a direction in the image plane $\hat{\mathbf{t}}(\phi)$, parametrized by the rotation angle ϕ. To do this, we search in the $n \times n$ neighborhood in pixels of pixel $\hat{x}$ (the projected pixel of point x) in screen-space directions $\hat{\mathbf{t}}(\phi)$ and $-\hat{\mathbf{t}}(\phi)$ to get each angle, and get the maximum horizon angle with respect to the view vector $\omega_\mathbf{o}$ as:

$$\theta_1(\phi) = \arccos\left(\max_{s<n/2} \langle \omega_\mathbf{s}, \omega_\mathbf{o}\rangle^+\right) \qquad (6)$$

where $\omega_\mathbf{s} = \frac{s-x}{\|s-x\|}$ and s the projection on world space of the pixel in the image plane $\hat{s} = \hat{x} + \hat{\mathbf{t}}(\phi)\cdot s,$. Angle θ_2 is computed analogously with $\hat{s} = \hat{x} - \hat{\mathbf{t}}(\phi)\cdot s$. Note that the size of the neighborhood n is scaled depending on the distance from the camera: this is necessary to make $\hat{\mathcal{A}}(x)$ view-independent, and is clamped to a maximum radius in pixels to avoid too large gathering radiuses on objects very close to the near plane, which would needlessly trash the GPU caches.

Given that we are interested only in the radiometric solid angle, we only need to keep track on the maximum angle, and not on other quantities (e.g. max distance) as in previous work. This allows, on AMD GCN hardware [AMD12] (our target platform) to compute the search loop with only one quarter speed instruction (`rsqrt`). Using this formulation, the shader becomes completely memory bound.

Solving the inner integral Timonen [Tim13a] solved this integral (including an attenuation function) by precomputing the result in a look-up table accessed in runtime. However, a key property of our formulation is that, given our computed horizon angles θ_1 and θ_2 we can solve analytically the inner integral $\hat{a}$ in Equation (5) as

$$\begin{aligned}\hat{a} = {} & \frac{1}{4}\left(-\cos(2\theta_1 - \gamma) + \cos(\gamma) + 2\theta_1\sin(\gamma)\right) \\ & + \frac{1}{4}\left(-\cos(2\theta_2 - \gamma) + \cos(\gamma) + 2\theta_2\sin(\gamma)\right). \qquad (7)\end{aligned}$$

It is important to note that this formulation requires that the normal $\mathbf{n_x}$ lays in the plane P defined by $\hat{\mathbf{t}}(\phi)$ and $\omega_\mathbf{o}$, which in general does not hold. Following Timonen [Tim13a], we compute the angle γ as the angle between the normalized *projected* normal $\frac{\overline{\mathbf{n_x}}}{\|\overline{\mathbf{n_x}}\|} \in P$ and $\omega_\mathbf{o}$ as $\gamma = \arccos(\langle \frac{\overline{\mathbf{n_x}}}{\|\overline{\mathbf{n_x}}\|}, \omega_\mathbf{o}\rangle)$. Then, we correct the change on the dot product by multiplying by the norm of $\overline{\mathbf{n_x}}$, which leaves Equation (5) as:

$$\hat{\mathcal{A}}(x) = \frac{1}{\pi}\int_0^\pi \|\overline{\mathbf{n_x}}\|\, \hat{a}(\phi)\, \mathrm{d}\phi. \qquad (8)$$

We found that our analytic solution is very fast, specially using fast `acos` and `sqrt` instructions [Dro14]. In terms of trascendental functions, after optimization we get a code with just 2 `cos` and 1 `sin`, plus three additional `acos` functions for setting up the integration domain. This makes our shader memory bounded, so the ALU operations required make almost no difference in terms of performance.

4.1 Implementation Details

Our technique is memory-bound, so the number of accesses to memory determine the final performance of our target platform. Given that our performance target is to integrate the technique in

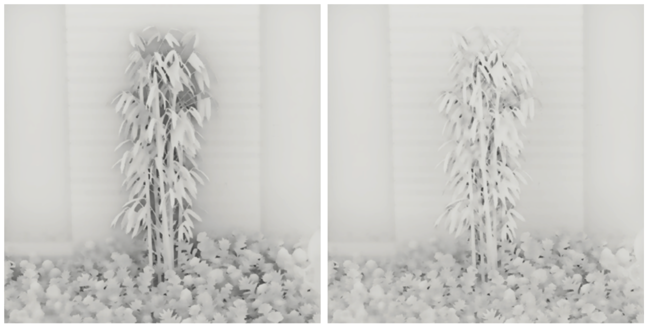

Figure 4: *Effect of using our thickness heuristic (right) in comparison to not using it (left). In screen-space methods, thin occluders such as leaves or branches tend to cast an unrealistic amount of occlusion: assuming that their thickness is similar to their width in screen, and correcting the maximum horizon angle we correct this effect.*

games running at 60 frames per second, we only have around half a millisecond to do our computations, which makes implementing optimizations mandatory. On one hand, we compute our ambient occlusion on half-resolution, which is later upsampled to full resolution. Moreover, in order to compute as much samples as possible without damaging the performance, we distribute the occlusion integral over both space and time: we sample the horizon in only one direction per pixel, but use the information gathered on a neighborhood of 4×4 using a bilateral filter for reconstruction. In addition, we make aggressive use of temporal coherency by alternating between 6 different rotations and reprojecting the results, using an exponential accumulation buffer. All this gives a total of $4 \times 4 \times 6 = 96$ effective sampled directions per pixel. Figure 3 shows the effect of the spatial and temporal gathering on the final reconstruction.

As opposed to ambient obscurance techniques, in our formulation we do not consider any attenuation function, which can result in abrupt discontinuities in the computed occlusion, especially as our gather radius does not cover the entire screen. In order to minimize artifacts we employ a conservative attenuation strategy. The idea is to ensure ground truth near-field occlusion, while for far-field occlusion attenuate it to zero, since in general far-field occlusion is baked together with the indirect lighting in our use cases. Our attenuation function is a linear blending from 1 to 0 from a given, large enough distance, to the maximum search radius.

Finally, since we cannot infer thickness from a depth buffer, thin features tend to cast too much occlusion to be realistic. While this could be solved with e.g. depth peeling, it is impractical in our case. Instead, we introduce a conservative heuristic derived from the assumption that the thickness of an objects is similar to their screen space size. This heuristic introduces a modification on the horizon search (Equation (6)), so that for each iteration $s \in [1, n/2]$ of the search we update the horizon θ as:

$$\theta = \begin{cases} \max(\theta_s, \theta) & \text{if } \cos(\theta_s) \geq \cos(\theta_{s-1}) \\ \mathrm{blend}(\theta_{s-1}, \theta_s) & \text{if } \cos(\theta_s) < \cos(\theta_{s-1}) \end{cases} \quad (9)$$

where blend is the blend operator based on the exponential moving average, and $\theta_0 = 0$. This heuristic also has the property of not biasing the occlusion results for simple corners (e.g. walls), which are a common occurrence in our application. Figure 4 shows the effect of this heuristic.

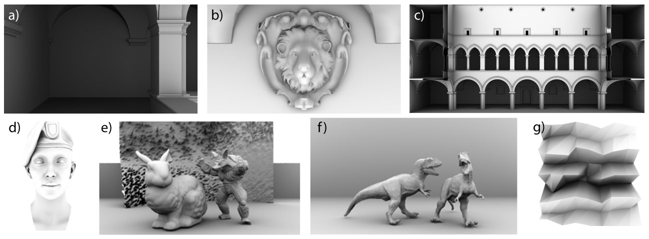

Figure 5: *Input scenes used for computing the mapping between the ambient occlusion and the near-field global illumination, rendered using only ambient occlusion.*

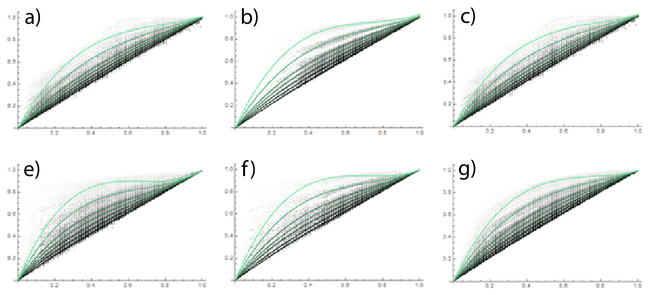

Figure 6: *Mapping between the ambient occlusion (x-axis) and the global illumination (y-axis) for the scenes in Figure 5 and different albedos. We can see how a cubic polynomial fits the data very well.*

5 Approximated Occlusion-Based Indirect Illumination

One of the main assumptions of ambient occlusion is that the only light reaching the shading point x comes directly from the uniform lighting environment. This means that the light incoming due multiple surface reflections is completely lost. This energy loss translates into overdarkening of edges and corners: these areas are where ambient occlusion affects most, but these are also where near-field interreflections are more dominant. Previous works on ambient obscurance(e.g. [MML12, Tim13b]) use ad-hoc attenuation functions to get some of this light back. However, these are ad-hoc solutions without an underlying physical meaning.

In order to address this issue, while remaining efficient and avoiding the costly computations of a full global illumination solution, we make the key observation that the near-field indirect illumination of a point in a region of constant albedo exhibits a strong relationship with its ambient occlusion value. This is not true in general for varying albedos, but we care most about nearby occluders that reflect close-range indirect illumination into x which makes the assumptions of similar albedos more likely to happen. Furthermore, the nearly-constant albedo assumption in the ambient occlusion neighborhood is also imposed by the fact we didn't want to sample albedos at each occluder surface, to remain in the limits of our runtime performance requirements.

Based on this key observation, and assuming that the albedo $\rho(s)$ at all points s around x is $\rho(s) = \rho(x)$, we want to design a mapping between the albedo and ambient occlusion at x and the reflected global illumination at x. To build this function $\mathcal{G}(\mathcal{A}(x), \rho(x))$ we compute seven simulations with different albedos ($\rho = [0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9]$) in a set of scenes scene showing a variety of different types occlusion conditions (see Figure 5). We compute both the ambient occlusion and multibounce

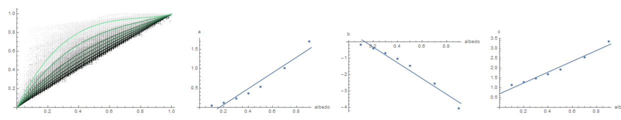

Figure 7: *Final cubic fit for our mapping between the ambient occlusion and the three-bounce global illumination for different albedos (left). We observed that a linear fit between the coefficients of the polynomial wrt the albedo gives a good continuous fit, as shown in the three rightmost figures. The combination of these fits give form to our model (Equation* (10)*).*

indirect illumination (in our case, up to three bounces). Figure 6 shows the mapping between $\mathcal{A}(x)$ and $\mathcal{G}(x)$ for each albedo, and for each scene in Figure 5. By taking the combination of all points, we fit this mapping using a cubic polynomial for each albedo (Figure 7 (left)), generating a set of polynomial coefficient for each scene albedo. We then observed that said coefficients were well approximated by a linear fit as a function of the input albedo (Figure 7). This last observation allows us to build a bidimensional mapping between the albedo ρ and ambient occlusion $\mathcal{A}$:

$$\begin{aligned}\mathcal{G}(\mathcal{A},\rho) &= a(\rho)\,\mathcal{A}^3 - b(\rho)\,\mathcal{A}^2 + c(\rho)\,\mathcal{A},\\ a(\rho) &= 2.0404\,\rho - 0.3324,\\ b(\rho) &= 4.7951\,\rho - 0.6417,\\ c(\rho) &= 2.7552\,\rho + 0.6903. \end{aligned}\tag{10}$$

This fitting-based approximation fulfills some of the requirements: on one hand, it can be integrated seamlessly in any ambient occlusion framework so that it re-incorporates the missing energy due to global illumination. On the other hand, it is extremely efficient, since it bases on already computed information, and without the need of expensive light transport simulations, while giving visually plausible results. This makes it very suitable for our target real-time applications.

6 GTSO: Specular Occlusion

Here we introduce our solution for *specular occlusion*, the glossy counterpart of the Lambertian-based ambient occlusion. As such, we would like to develop an illumination model where the near-field occlusion modulates the probe-based lighting while supporting arbitrary BRDFs. Moreover, for the specific cases of constant probe illumination, we would like a model delivering ground truth results, similar to AO.

As in Section 4, lets assume that all light comes from an infinitely far lighting environment (light probe) to express Equation (1) as:

$$L_r(x,\omega_\mathbf{o}) = \int_{\mathcal{H}^2} V(x,\omega_\mathbf{i}) L_i(x,\omega_\mathbf{i}) f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}. \tag{11}$$

Computing this integral by numerical integration is too expensive for real-time applications, and the current state-of-the-art, when using Cook-Torrance [CT82] microfacet BRDFs, is to adopt an formulation that assumes constant perfect visibility ($\forall\omega_\mathbf{i}|V(x,\omega_\mathbf{i}) = 1$) and uses a split-integral approximation [Laz13, Kar13] as:

$$\begin{aligned} &L_r(x,\omega_\mathbf{o}) \approx \mathcal{L}(x)\cdot\mathcal{F}(x,\omega_\mathbf{o}),\\ &\mathcal{L}(x) = \frac{1}{C_L}\int_{\mathcal{H}^2} \overbrace{V(x,\omega_\mathbf{i})}^{=1} L_i(x,\omega_\mathbf{i}) D(x,\omega_\mathbf{h}) \langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i},\\ &\mathcal{F}(x,\omega_\mathbf{o}) = \int_{\mathcal{H}^2} f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}, \end{aligned}\tag{12}$$

where $C_L = \int_{\mathcal{H}^2} D(x,\omega_\mathbf{h})\langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}$ is the normalization factor needed in the first integral to guarantee it is always in the range $[0,1]$ when $L_i(x,\omega_\mathbf{i}) = 1$, $D(x,\omega_\mathbf{h})$ is the normal distribution function of the surface [TS67], and $\omega_\mathbf{h}$ is the half vector. Intuitively, the second integral is the full microfacet BRDF at the pixel under an uniform white lighting enviroment, and can be stored in a pre-computed lookup table, while the first integral is the convolution of the actual lighting environment $L_i(x,\omega_\mathbf{i})$ with a circularly symmetric lobe that approximates the distribution function in the Cook-Torrance BRDF. When we represent the lighting environment as a irradiance cubemap, this first integral can be computed by preconvolving the cubemap with lobes from different surfaces roughness, which makes it very efficient for rendering glossy materials, although most approximations ignore occlusion or approximate it with heuristics.

In order to compute specular occluded ligthing, we opt for an approach similar to the split-integral approximation, and separate the visibility term form the first integral as a constant. In essence, the idea is computing an occlusion term that, in the spirit of ambient occlusion, modulates the amount of illumination reaching x. This allows us transforming Equation (12) into a product of three integrals:

$$L_r(x,\omega_\mathbf{o}) \approx \mathcal{S}(x,\omega_\mathbf{o})\cdot\mathcal{L}(x)\cdot\mathcal{F}(x,\omega_\mathbf{o}), \tag{13}$$

where the term modeling visibility $\mathcal{S}$ is our *specular occlusion* term computed as:

$$\mathcal{S}(x,\omega_\mathbf{o}) = \frac{1}{C_V}\int_{\mathcal{H}^2} V(x,\omega_\mathbf{i}) f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}, \tag{14}$$

with the normalization term $C_V = \int_{\mathcal{H}^2} f_r(x,\omega_\mathbf{i},\omega_\mathbf{o})\langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}$ ensuring that the specular occlusion $\mathcal{S}$ ranges into $[0,1]$. As we can see, our definition of specular occlusion is weighted by the BRDF, and thus is directionally dependent. In the following subsection, we detail the computations of specular occlusion $\mathcal{S}$.

Interestingly, the normalization factor C_V is the same as the latter integral $\mathcal{F}$, and thus it cancels out in Equation (13), leaving Equation (13) as:

$$\begin{aligned} L_r(x,\omega_\mathbf{o}) \approx &\int_{\mathcal{H}^2} V(x,\omega_\mathbf{i}) f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}\\ &\cdot \frac{1}{C_L}\int_{\mathcal{H}^2} L_i(x,\omega_\mathbf{i}) D(x,\omega_\mathbf{h}) \langle \mathbf{n_x},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}. \qquad (15) \end{aligned}\tag{16}$$

This final form has the property that for a constant probe illumination it matches exactly the ground truth. Moreover, if we compare it with the original split-integral approximation (Equation (12)), we can see that the main difference is that the visibility term has been moved to the BRDF integral, and is no longer assumed constant.

6.1 Computing Specular Occlusion

Our key idea to compute specular occlusions $\mathcal{S}(x,\omega_\mathbf{o})$ efficiently is to model an approximation for both the visibility and the BRDF lobes, and then compute the intersection between these two as the specular occlusion. With that in mind, the problem reduces to the question on how representing both the visibility and the BRDF compactly, and on how to compute the intersection between both.

For the **visibility**, we assume that it can be approximated as a cone, computed from a bent normal [Lan02] and an ambient occlusion term. These two can be computed on the fly (see Section 4), or just be precomputed (e.g. stored as texture or vertex data). We chose

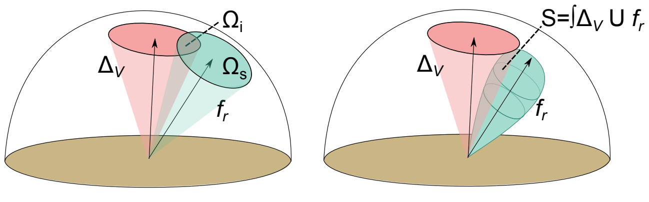

Figure 8: *Geometry of our specular occlusion, assuming that both the visibility and the specular are modeled as cones (left), and with accurate specular lobe (right).*

this representation because it allows us to reuse the data from high-quality screen-space ambient occlusion computed in Section 4. The bent normal $\mathbf{b}$ acts as the direction of the cone. To compute the cone amplitude α_v, we rely on the ambient occlusion: Assuming that the visibility is homogeneous around the bent normal, then α_v is the maximum horizon angle $\theta_1(\phi) = \theta_2(\phi) = \alpha_v$ for all directions ϕ (see Equation (5)). With these assumptions, the ambient occlusion $\hat{\mathcal{A}}(x)$ can be expressed analytically as:

$$\hat{\mathcal{A}}(x) = 1 - \cos(\alpha_v(x))^2, \tag{17}$$

which we can invert to get the cone angle α_v as a function of $\hat{\mathcal{A}}(x)$ as:

$$\cos(\alpha_v(x)) = \sqrt{1 - \hat{\mathcal{A}}(x)}. \tag{18}$$

Similar to the visibility, we can model the **specular** lobe as a cone centered on the reflection direction $\omega_\mathbf{r}$. This imposes several assumptions, including constraining the BRDF's lobe to be rotationally symmetric on $\omega_\mathbf{r}$ (which is not true with microfacet BRDFs, but it's the same approximation done with cubemap pre-convolution), and approximates the actual BRDF as a single constant value. However, this allows to compute the specular occlusion as

$$\mathcal{S}(x, \omega_\mathbf{o}) = \frac{\Omega_i(x, \omega_\mathbf{o})}{\Omega_s(x, \omega_\mathbf{o})}, \tag{19}$$

the ratio between the solid angle of the intersection of solid angles of both the visibility and specular cones Ω_i and the specular cone Ω_s (see Figure 8). This ratio can be compute analytically, and given an gives good results, despite being a coarse approximation of the underlying specular reflectance. Figure 8 (left) shows an example when approximating both the visibility and the BRDF using cones (we refer to Appendix B for details).

However, in a real-time application as our target, these computations might be still expensive, and end up baked in a pre-computed three-dimensional look up table, parametrized by the angle between the bent normal and the reflection vector $\beta = \arccos(\langle \mathbf{b}, \omega_\mathbf{r} \rangle)$, and the amplitude of both cones α_v and α_s respectively. With that in mind, we opt for a more accurate, precomputation-based approximation, where we compute the specular occlusion $\mathcal{S}$ as the product of the visibility cone Δ_V and the actual BRDF F (Figure 8, right):

$$\mathcal{S}(x, \omega_\mathbf{o}) \approx \frac{1}{C_V} \int_{\mathcal{H}^2} \Delta_V(\alpha_v(x), \beta(\mathbf{b}(x), \omega_\mathbf{i})) \\ f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o}) \langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i}, \tag{20}$$

with $\Delta_V(\alpha_v, \beta)$ a binary function returning 1 if $\beta \leq \alpha_v$ and 0 elsewhere. Assuming an isotropic microfacet-based BRDF with a GGX NDF [WMLT07] parametrized by a roughness value r, we model the reflected direction $\omega_\mathbf{r}$ as the angle $\theta_o = \arccos(\langle \mathbf{n_x}, \omega_\mathbf{r} \rangle)$ with respect to the normal $\mathbf{n_x}$. With these assumptions, and omitting the spatial dependence for clarity, we can express $\mathcal{S}$ as a four dimensional function:

$$\mathcal{S}(\alpha_v, \beta, r, \theta_o) \approx \frac{1}{C_V} \int_{\mathcal{H}^2} \Delta_V(\alpha_v, \beta) f_r(\omega_\mathbf{i}, \theta_o, r) \langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i}. \tag{21}$$

This function can be compactly baked as a four-dimensional table. Moreover, by assuming the normal $\mathbf{n_x}$ is the bent normal $\mathbf{b}$ then $\theta_o = \beta$, which would reduce the dimensionality of the table to three dimensions, at the price of introducing a little error. Given that the function is relatively smooth, we can encode it to a four-dimensional 32^4 (or 32^3 for the 3D approximation) BC4 8-bit look up table, which can be efficiently accessed in runtime.

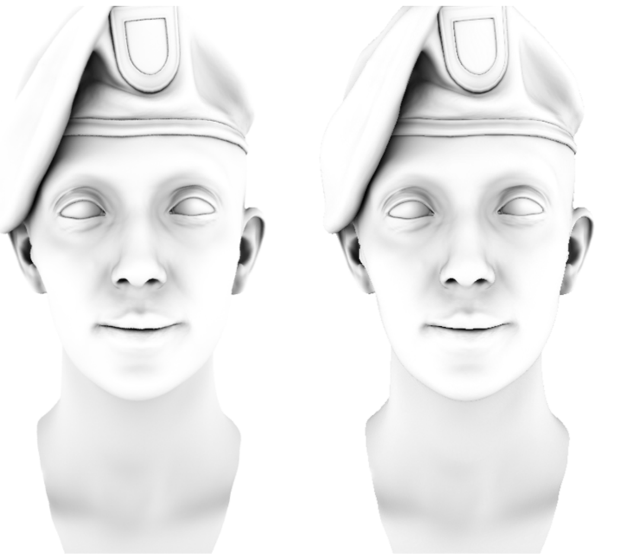

Figure 9: *Comparison of ambient occlusion between our Monta Carlo rendered ground truth (left) and our technique.*

7 Results

Here we show the results obtained with our techniques, and compare it against a ground truth computations. These are done using explicit ray-traced ambient occlusion, multiple bounces path tracing for our global illumination approximation, and BRDF ray-traced sampling for the specular occlusions.

We implemented our techniques in both an stand-alone application, and within a full-featured commercial game engine. Figure 9 compares our results against a Monte Carlo ground truth: for the white probe assumption of ambient occlusion, our technique is able to faithfully match the ground truth, while being practical for games at HD resolution and 60 fps. Figure 10 show in-game examples of our GTAO, with physically-based shading and complex geometry. Our technique computes screen-space ambient occlusion in just 0.5 ms in PS4 at 1080p, by taking advantage of both our AO formulation and the spatio-temporal sample integration.

Similarly, we compare our approximation to near-field global illumination against a path traced ground truth. Figure 11 shows the *Lauren* model rendered with ambient occlusion only (GTAO), and then including global illumination both with gray and colored albedo, while Figure 12 shows the same comparison with different values of gray albedo.

Figure 13 compares our specular occlusion technique against a

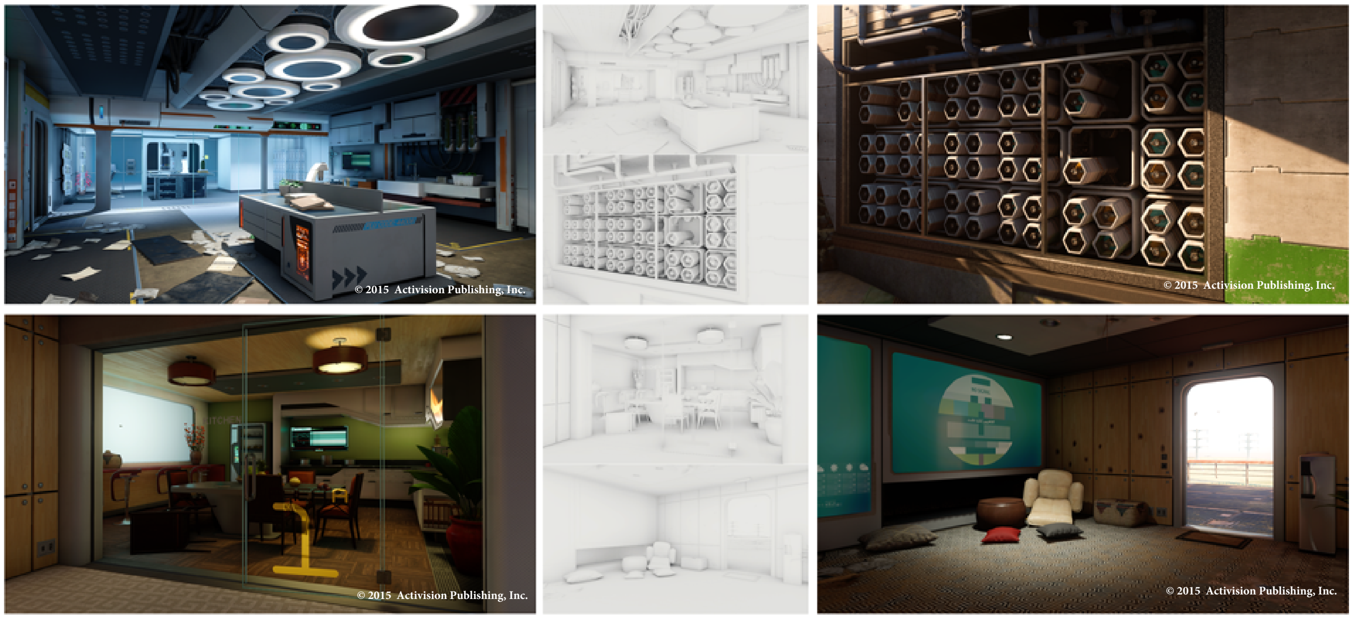
© 2015 Activision Publishing, Inc.
© 2015 Activision Publishing, Inc.
© 2015 Activision Publishing, Inc.
© 2015 Activision Publishing, Inc.

Figure 10: *Screenshots of our GTAO being used in-game for accurate and efficient ambient occlusion, in scenes with high-quality physically-based shading and high geometric complexity. Our GTAO computes the ambient occlusion layer (in the insets) in just 0.5 ms for PS4.*

Monte Carlo ground truth: while there are some small differences specially at grazing angles, our technique is able to match most of the specular appearance of the model while taking into account occlusions, even for non-constant illumination. Finally, Figure 14 compares the use of the three-dimensional and four-dimension look-up tables for computing the specular occlusion, compared against the ground truth.

8 Conclusions

In this work we have presented several contribution to screen-space real-time ambient occlusion. In the first place, we have presented GTAO: an efficient formulation of ambient occlusion that matches the Monte Carlo ground truth within a very tight budget. Our formulation goes together with an efficient implementation that aggressively makes use of both spatial and temporal coherence to effectively integrate almost 100 samples per pixel while computing only one each frame. GTAO goes together with a simple but effective technique that simulates near-field diffuse inter-reflections based on the ambient occlusion at the shading point. The technique bases on the observation that these inter-reflections can be modeled, from data, as a function of the local albedo and the ambient occlusion. This allows to avoid the typical over-darkening resulting from ambient occlusion.

Finally, we have introduced an approximation of specular occlusion with our *Ground-Truth Specular Occlusion*, which generalizes the ambient occlusion operator to deal with specular surfaces, and introduced an efficient technique based on a precomputed look-up table to efficiently compute the specular reflection from constant and non-constant probe-based illumination.

Acknowledgements

We would like to thank Stephen Hill, Stephen McAuley, Christer Ericson, Dimitar Lazarov, Eran Rich, Jennifer Velazquez, Josh Blommestein, Josiah Manson, Manny Ko, Michal Iwanicki, Danny Chan, Michal Drobot and Peter-Pike Sloan.

References

AMD. AMD graphics cores next (GCN) arquitecture. Technical report, AMD, 2012.

Louis Bavoil, Miguel Sainz, and Rouslan Dimitrov. Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 Talks*, 2008.

Michael Bunnell. Dynamic ambient occlusion and indirect lighting. In *Gpu gems*, volume 2, pages 223–233. CRC Press, 2005.

Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG)*, 1(1):7–24, 1982.

Michal Drobot. Low level optimization for gcn. In *Digital Dragons 2014*, 2014.

Quintjin Hendrickx, Leonardo Scandolo, Martin Eisemann, and Elmar Eisemann. Adaptively layered statistical volumetric obscurance. In *Proceedings of the 7th Conference on High-Performance Graphics*, pages 77–84. ACM, 2015.

Brian Karis. Real shading in unreal engine 4. In *ACM SIGGRAPH 2013 Courses*, 2013.

Hayden Landis. Production-ready global illumination. In *ACM SIGGRAPH 2002 Courses*, 2002.

Dimitar Lazarov. Getting more physical in call of duty: Black ops ii. In *ACM SIGGRAPH 2013 Courses*, 2013.

Bradford James Loos and Peter-Pike Sloan. Volumetric obscurance. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 151–156. ACM, 2010.

Oleg Mazonka. Solid angle of conical surfaces, polyhedral cones, and intersecting spherical caps. *arXiv preprint arXiv:1205.1396*, 2012.

Martin Mittring. Finding next gen: Cryengine 2. In *ACM SIGGRAPH 2007 Courses*, 2007.

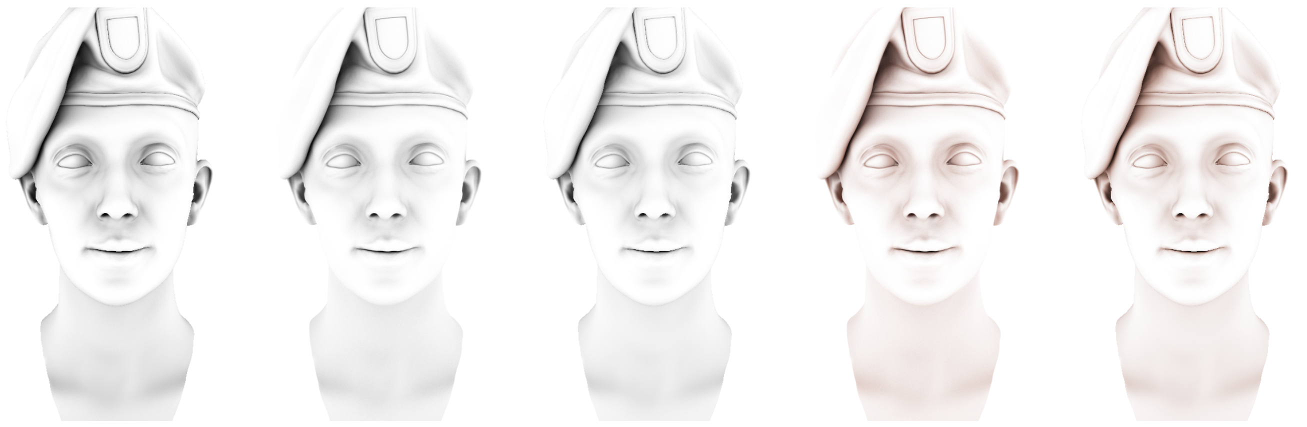

Figure 11: *Adding near-field global illumination to ambient occlusion: From left to right, ambient occlusion only (GTAO), ground truth three bounces global illumination and our technique for approximating global illumination with gray albedo, both for gray and colored albedo. Our approximation model for diffuse interreflections based on the ambient occlusion matches very closely the ground truth, being able to recover the light lost due to considering only one-bounce illumination.*

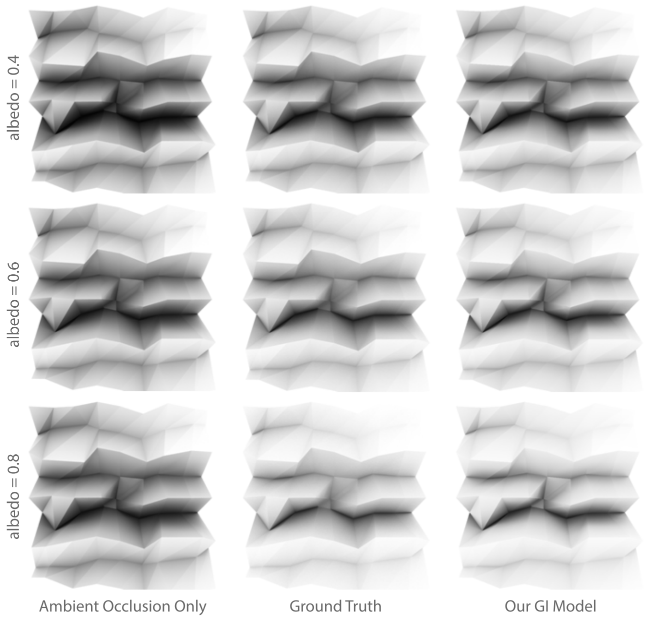

Figure 12: *Effect of albedo in our ambient occlusion-based global illumination approximation, for the* groove *scene. From left to right: GTAO only, Monte Carlo ground truth, and our approximation based on GTAO, for albedos 0.4, 0.6 and 0.8.*

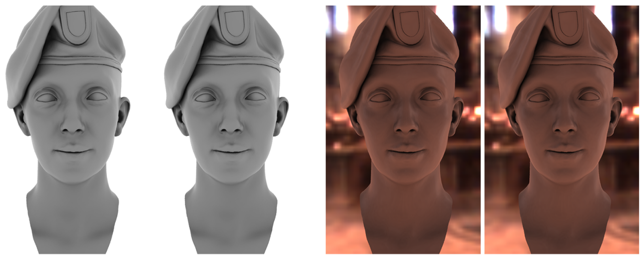

Figure 13: *Comparison between ground truth specular illumination (left) and our specular occlusion model (right), under two different illumination setups. While the setup of the left follows the white probe assumption underlying the theoretic model of specular occlusion, the second uses a non-constant illumination probe.*

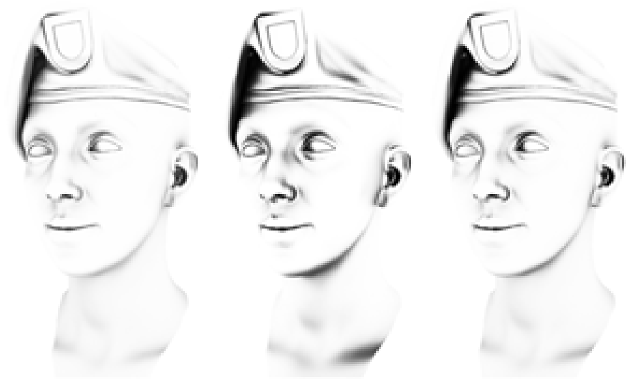

Figure 14: *Comparison between using a four-dimensional look-up table (right) for storing the specular occlusion based on a visibility cone, using the reduced three-dimensional look-up (left), and the Monte Carlo ground truth (middle), for a white light probe.*

Morgan McGuire, Michael Mara, and David Luebke. Scalable ambient obscurance. In *Proceedings of the Fourth ACM SIGGRAPH/Eurographics conference on High-Performance Graphics*, pages 97–103. Eurographics Association, 2012.

Morgan McGuire, Brian Osman, Michael Bukowski, and Padraic Hennessy. The alchemy screen-space ambient obscurance algorithm. In *Proc. of the ACM SIGGRAPH Symposium on High Performance Graphics*, pages 25–32. ACM, 2011.

Christopher Oat and Pedro V Sander. Ambient aperture lighting. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 61–64. ACM, 2007.

Tobias Ritschel, Carsten Dachsbacher, Thorsten Grosch, and Jan Kautz. The state of the art in interactive global illumination. *Computer Graphics Forum*, 31(1):160–188, 2012.

Tobias Ritschel, Thomas Engelhardt, Thorsten Grosch, H-P Seidel, Jan Kautz, and Carsten Dachsbacher. Micro-rendering for scalable, parallel final gathering. *ACM Transactions on Graphics (TOG)*, 28(5):132, 2009.

László Szirmay-Kalos, Tamás Umenhoffer, Balázs Tóth, László Szécsi, and Mateu Sbert. Volumetric ambient occlusion for realtime rendering and games. *IEEE Computer Graphics and Applications*, 30(1):70–79, 2010.

Ville Timonen. Line-sweep ambient obscurance. *Computer Graphics Forum*, 32(4):97–105, 2013.

Ville Timonen. Screen-space far-field ambient obscurance. In *Proceedings of the 5th High-Performance Graphics Conference*, pages 33–43. ACM, 2013.

Kenneth E Torrance and Ephraim M Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1112, 1967.

Yasin Uludag. Hi-z screen-space cone-traced reflections. In *GPU Pro 5: Advanced Rendering Techniques*, page 149. CRC Press, 2014.

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proc. of EGSR '07*, 2007.

Sergey Zhukov, Andrei Iones, and Grigorij Kronin. An ambient light illumination model. In *Rendering Techniques 98*, pages 45–55. Springer, 1998.

A Uniformly-weighted GTAO

Although not radiometrically correct, some works [BSD08] use a formulation slightly different to Equation (2), removing the foreshortening term, and therefore weighting occlusion uniformly in the hemisphere, instead of cosine-weighted. In this scenario, our formulation for GTAO would also be valid, with the modification of Equation (5), which becomes:

$$\hat{\mathcal{A}}(x) = \frac{1}{2\,\pi}\int_0^{\pi} \underbrace{\int_{\theta_1(\phi)}^{\theta_2(\phi)} |\sin(\theta)|\,\mathrm{d}\theta}_{\hat{a}}\,\mathrm{d}\phi. \tag{22}$$

Removing the cosine-term not only changes the normalization factor, from $1/\pi$ to $1/(2\pi)$, but also changes the full inner integral $\hat{a}$. Fortunately, in this form this integral has also an analytic solution as:

$$\hat{a} = 2 - \cos(\theta_1) - \cos(\theta_2). \tag{23}$$

Note that since here there is not cosine term with respect to the normal, the projection needed for cosine-weighted GTAO (Equation (8)) is not needed.

This formulation of GTAO does not support the near-field global illumination approximation shown in Equation (10), since the cubic polynomial fit has been done for the radiometrically-correct cosine-weighted GTAO. However, occlusion computed using Equation (22) can be used to determine the aperture $\alpha_v(x)$ of the visibility cone in Section 6 as:

$$\cos(\alpha_v(x)) = 1 - \hat{\mathcal{A}}(x). \tag{24}$$

B Cone-to-Cone GTSO

In order to compute our GTSO for specular occlusion based on Equation (19), we need to compute the visibility and specular cones, defined by a direction and an aperture, and their intersection solid angle Ω_i. The visibility cone is explained in Section 6 (Equation (18)). In the case of the specular cone, its direction is defined by the reflection vector $\omega_{\mathbf{r}}$. Its aperture α_s, on the other hand, it is defined by the roughness r (or specular power p in the case of a Phong BRDF). Since there are no exact solution for this, we opt of an approach similar to the one by Uludag [Ulu14], which uses the Phong importance sampling routine by Walter et al. [WMLT07] to relate the aperture with the Phong power p:

$$\alpha_s = \arccos\left(u^{\frac{1}{p+2}}\right), \tag{25}$$

where u is a constant. As opposed to Uludag, we do not obtain u by fitting the cone to lobes ($u = 0.244$), but minimize differences between resulting GTSO and Monte Carlo ground truth references, getting $u = 0.01$. Then, by mapping roughness Phong specular power p to r by using $r = (2/(p+2))^{0.5}$ for faster evaluation we get a final aperture cone:

$$\cos(\alpha_s) = 0.01^{\frac{1}{p+2}} = 0.01^{0.5\,r^2} = e^{-2.30259\,r^2} = 2^{-3.32193\,r^2}. \tag{26}$$

Once we have both cones, the only thing left is computing the intersection solid angle Ω_i from these cones. This intersection has analytical solution [OS07, Maz12], as a function of the cone apertures and the angle between their respective directions, the bent normal $\mathbf{b}$ and the reflection direction $\omega_{\mathbf{r}}$.