Practical Real-Time Strategies for Accurate Indirect Occlusion

Technical Memo ATVI-TR-19-01

Jorge Jimenez[1] Xian-Chun Wu[1] Angelo Pesce[1] Adrian Jarabo[2]

[1]Activision Blizzard [2]Universidad de Zaragoza, I3A

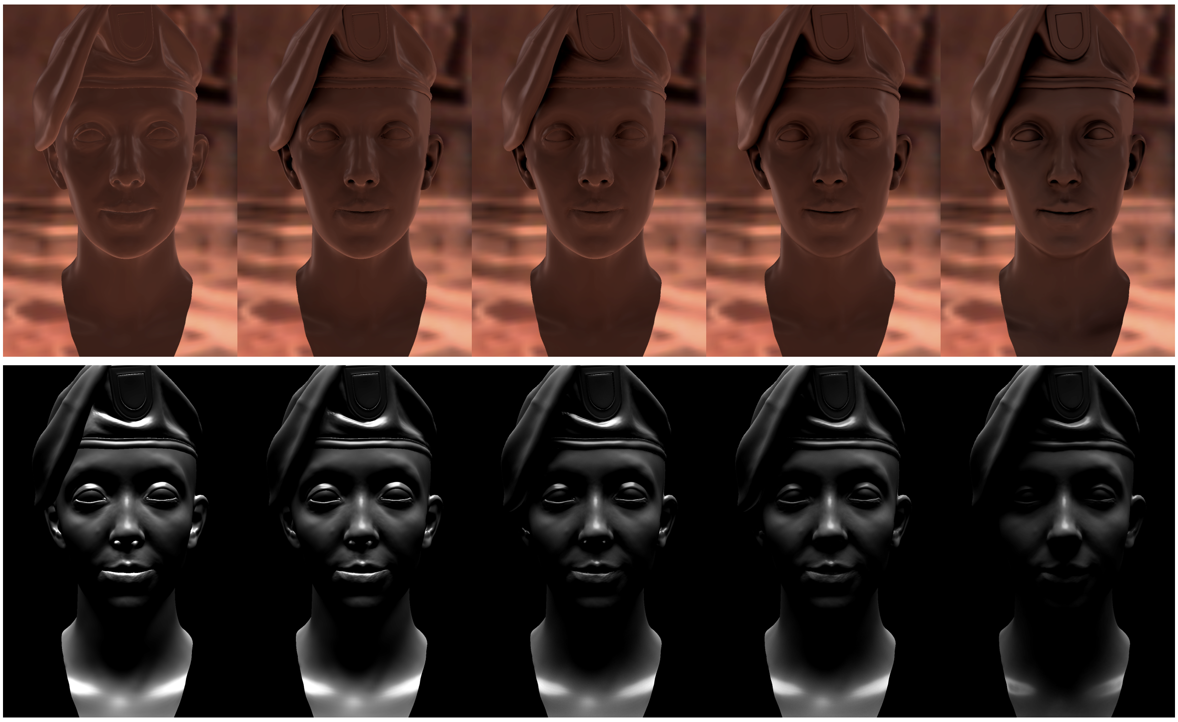

Figure 1: *Example renders with our practical occlusion techniques under illumination from the* Grace *light probe (top) and a high-frequency binary probe (bottom). From left to right: no occlusion, our GTAO with multiple-bounces, with spherical harmonics directional occlusion, with our GTSO modeling specular occlusion, and a ray-traced ground truth. Our techniques render high-quality occlusion matching the ground truth, with the baseline GTAO + GI rendering in just 0.5 ms on a PS4 at 1080p (for a standard halfres occlusion buffer).*

Abstract

In this work we introduce a set of techniques for real-time ambient occlusion targeted to very tight budgets. We propose GTAO, a new formulation of screen-space ambient occlusion that allows the composited occluded illumination to match the ground truth reference in half a millisecond on current console hardware. This is done by using a radiometrically-correct formulation of the ambient occlusion equation, and an efficient implementation that distributes computation using spatio-temporal sampling. As opposed to previous methods, our technique incorporates the energy lost by missing interreflections by using an efficient, accurate physically-based parametric form, avoiding the use of ad-hoc approximations of indirect illumination. Then, we extend GTAO to account for directionally-resolved illumination, by fastly projecting coupled visibility and foreshorting factors into spherical harmonics, and thoroughly analyze with previous work. Finally, we introduce a novel model for specular occlusion formulation that accounts for the coupling between visibility and BRDF, closely matching the ground truth specular illumination from probe-based lighting, and propose GTSO, an efficient implementation of this concept based on tabulation. Our techniques are practical real-time, give results close to the ray-traced ground truth, and have been integrated in recent AAA console titles.

1. Introduction

Ambient occlusion (AO) is an approximation of global illumination, that models the diffuse shadows produced by close, potentially small occluders in a tight budget. It allows to preserve high-frequency details and contrast in low-frequency precomputed indirect illumination via pre-baked illumination or light probes. Unfortunately, solving the ambient occlusion integral is still too expensive to be practical in certain scenarios (e.g. 1080p or 4K rendering at 60 fps), so approximations have been developed in the past to achieve the target performance budget.

We introduce a new set of screen-space occlusion techniques, that target *practical* real-time performance while matching ray-traced ground truth solutions. We propose a novel technique for ambient occlusion, that we call *ground truth-based ambient occlusion (GTAO)*, that decouples ambient occlusion from the near-range indirect illumination. This allows us to solve efficiently the ambient occlusion integral by avoiding piecewise integration (as required when using obscurance estimators), while recovering the lost multiple scattered diffuse lighting by using an efficient physically-based functional approximation. This allows to match not only ground truth *occlusion*, but also *illumination* references. Then, we extend our ambient occlusion model to directionally-resolved illumination from distant probes, that uses our accurate ambient occlusion term and our from-horizons bent normal calculations to derive an efficient expansion in spherical harmonics, that can be used to efficiently integrate ambient illumination. Finally, we generalize ambient occlusion for arbitrary specular materials and formulate it by using a novel split-integral formulation that couples the BRDF with the visibility. We propose an efficient implementation of this formulation, that we call *ground truth-based specular occlusion (GTSO)*, to compute it in runtime by accessing a small precomputed table.

In particular, our contributions are:

- **GTAO:** An efficient ambient occlusion technique that matches a radiometrically-correct ambient occlusion integral, and incoporates the lost energy due to close-range indirect illumination using a simple closed-form analytical expression.
- **Directional GTAO:** an extension that accounts for directionally-resolved distant illumination, which includes a ground truth derivation of horizon-based bent normals.
- **Specular occlusion (SO):** A generalization of the standard ambient occlusion formulation for arbitrary specular BRDFs that couples visibility and reflectance for efficiently computing specular reflection from distant probes. This formulation matches ground truth references under the same assumptions as ambient occlusion (uniform dome and a single bounce), and is one of the principal results of our work.
- **GTSO:** An efficient implementation of this specular formulation for microfacets-based BRDFs.

Figure 1 shows the effect of these techniques, and how their combination match the Monte Carlo raytraced ground truth. We implement them efficiently, leveraging temporal reprojection and spatial filtering to compute our baseline ambient occlusion in just 0.5 ms per frame on a Sony Playstation 4, for a game running at 1080p (using a standard halfres occlusion buffer). Our results highlight that for today hardware standards, performing ad-hoc occlusion calculations are no longer necessary for performance reasons anymore.

2. Related Work

Given the large amount of previous work on global illumination in general, and ambient occlusion in particular, here we focus on the most related works with ours. For a wider overview on the field we refer to the surveys by Ritschel et al. [RDGK12], and Aalund and Bærentzen [AB12].

Screen-Space Ambient Occlusion Ambient occlusion [ZIK98] integrates the visibility from a point in the scene, to modulate the ambient illumination term. It requires to perform expensive visibility queries from the shaded point. In order to alleviate the overdarkening resulting from ignoring interreflections the visibility is commonly modulated by an ad-hoc fall-off function; in these cases, it is common to term AO as *ambient obscurance*. In his seminal work, Mittring [Mit07] proposed to move the visibility queries to screen-space, assuming that only the geometry visible from the camera acts as occluder. He approximated ambient occlusion by sampling the depth map of the scene, and evaluated whether a point is occluded (behind) geometry in the depth map, effectively calculating volumetric occlusion using point samples. Several works have improved the sampling strategy [LS10, SKUT*10, HSEE15], by integrating using line samples rather than points. While they obtain high quality results, those methods simplify the integral function resulting into radiometrically-incorrect ambient occlusion[†]. Bavoil et al. [BSD08] proposed to perform line integrals based on the horizon angles of the geometry around x using screen space ray tracing. They termed their technique *horizon based AO* (HBAO). This work is similar in spirit to volumetric line sampling approaches in that it realizes that any ray under the horizon will be occluded if the horizon was already occluded. McGuire and colleages [MOBH11, MML12] later simplified the ray tracing process by assuming that x and any near-field position on the positive hemisphere are mutually visible. While HBAO and its improvements are efficient, they are not radiometrically correct, and does not account for multiple scattering in the near field. Timonen [Tim13a, ST15] improves HBAO by performing line sweeps along all the image, finding the maximum horizon angle for a given direction in constant time by amortizing samples over many pixels. Closely related to our GTAO, the same author [Tim13b] proposed a radiometrically-correct estimator for ambient obscurance by line-scanning and filtering the depth map, which is able to match the raytraced *obscurance* ground truth at small cost even for very large gathering radii. While the technique yields impressive results, the use of a obscurance estimator prevents matching ground truth *illumination* (rather matching obscurance), and obligates the usage of a piecewise inner integral for the occlusion computations, implemented with a look up table. Our work efficiently computes radiometrically-correct ambient occlusion based on visibility horizons. It does not require ad-hoc fall-off functions to avoid over-darkening, since indirect illumination is accounted by a physically-based parametric formula. This allows to reduce AO computations

[†] By "radiometrically-correct" we mean that foreshortening is taken into account in the ambient occlusion integral.

to its bare bones by solving the inner integral analyticaly. In addition, we generalize GTAO to directional and non-Lambertian occlusion.

Directional Occlusion While AO has received significant attention, only a few works have focused on introducing the directional dependence of ambient illumination encoded in e.g. probes. Rather than relying on a pre-filtered probe, Ritschel et al. [REG*09] approximate directional diffuse lighting by evaluating the render equation on the fly. Despite of using an approximate visibility test, it is too slow for practical real-time environments. Landis [Lan02] proposed to use bent normals to fetch from the ambient probe in the most visible direction, in order to increase the directional fidelity of ambient occlusion. Since then, bent normals have observed a widespread usage both for off-line and real-time rendering. Klehm et al. [KRES11] extended SSAO [Mit07] to handle bent normals, averaging the directions to visible samples. They also propose a variant for HBAO using a similar rationale, although averaging horizon directions does not match 3d ray-traced bent normals. Oat and Sander [OS07] precalculated visibility by means of ambient occlusion and bent normals, then using this information during real-time rendering by calculating the spherical cap intersection with the light source aperture, effectively applying visibility to diffuse lighting. Ramamoorthi and Hanrahan [RH01] proposed to encode light probes into spherical harmonics, allowing to efficiently convolve light and the foreshortening factor in real-time. Green [Gre03] convolved with the visibility as well by means of the triple SH product. We build on these ideas, and propose an efficient projection into SH of the coupled visibility and foreshortening factor in run-time that avoids the more expensive and less accurate triple SH product.

Specular Occlusion While ambient occlusion as been largely studied over the last two decades, specular occlusion has not received similar attention, despite of possibly being as important, specially with the adoption of physically-based shading models. Gotanda [Got12] derived empirically specular occlusion from ambient occlusion. He noted that ambient occlusion does not consider the BRDF lobe shape, resulting in a mismatching occlusion scale. Lagarde [Ld14] adopted a similar empirical approach, adapted to GGX-based microfacets using the roughness of the surface to shape the resulting specular occlusion. Jimenez and von der Pahlen. [Jv13] highlighted the importance of specular occlusion for rendering photorealistic characters, even when using ad-hoc approaches. In contrast, we formally derive a specular occlusion term analogous to ambient occlusion, that couples visibility and specular BRDF. In addition, we propose an efficient model for rendering with this specular occlusion term.

3. Background & Overview

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

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

where $\mathcal{H}^2$ is the hemisphere centered in x and having $\mathbf{n}$ as its axis, $L(x,\omega_\mathbf{i})$ 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 $\langle \mathbf{n},\omega_\mathbf{i}\rangle^+$ models foreshortening. Ambient occlusion [ZIK98] approximates Equation (1), by introducing a set of assumptions: *i)* all surfaces around x are purely absorbing (i.e. do not bounce light), *ii)* all light comes from an infinite uniformly white environment light (or generalizing, of any uniform color), which might be occluded by the geometry around x; and *iii)* the surface at x is a Lambertian surface. This transforms Equation (1) into

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

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 $\omega_\mathbf{i}$. 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 as an ad-hoc solution to avoid the typical over-darkening in AO produced by ignoring near-field interreflections.

Ambient occlusion is only exact for uniform illumination. However, it is ofter used in practice for *any* illumination stored in a light probe. In these cases, the illumination is approximated as

$$\begin{aligned} L_r(x,\omega_\mathbf{o}) &\approx \frac{\rho(x)}{\pi} \mathcal{A}(x) \int_{\mathcal{H}^2} L(x,\omega_\mathbf{i}) f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i} \\ &= \frac{\rho(x)}{\pi} \mathcal{A}(x) \mathcal{L}(x,\omega_\mathbf{h}), \end{aligned} \qquad (3)$$

where $\mathcal{L}(x,\omega_\mathbf{h})$ is the light probe pre-convolved with the BRDF, and $\omega_\mathbf{h}$ is the query direction at the probe. Several works base on the bent normals $\omega_\mathbf{h} = \mathbf{b}$ [Lan02] to fetch the probe, which is later attenuated by the ambient occlusion term $\mathcal{A}(x)$. While this incorporates some degree of directionality in the incoming radiance, the visibility V and lighting L terms remain decoupled in Equation (3).

An alternatively common approach is to encode the light probe and visibility as a n^{th} order spherical harmonics (SH) expansion [RH01]. This allows to compute $L_r(x,\omega_\mathbf{o})$ efficiently, as a SH double product

$$L_r(x,\omega_\mathbf{o}) \approx \frac{\rho(x)}{\pi} \sum_{j=0}^{n} \widehat{L}_j \widehat{V}_j, \qquad (4)$$

where $\widehat{L}_j$ and $\widehat{V}_j$ are the j^{th} SH coefficient for L and V respectively. Unfortunately, introducing the foreshortening or adding a BRDF requires an expensive triple SH product, which can limit the applicability of this approach.

Objectives In this work we have two main goals: On one hand, we propose a technique that matches the radiometrically-correct ambient occlusion definition, 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, to not only match ground truth *occlusion* but rather ground truth *illumination*, for an extended set of material BRDFs and input lighting configurations including non-uniform dome illuminations.

The first goal imposes severe limitations in terms of input data, number of passes, and number of instructions. Bounded by these

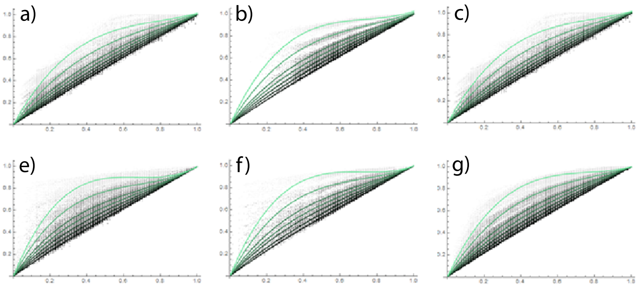

Figure 2: *Mapping between the ambient occlusion (x-axis) and the global illumination (y-axis) for the scenes in Figure 17 and different albedos. A cubic polynomial (drawn in green) fits the data very well, suggesting a functional relationship between AO and GI. We develop such relationship in Section 4.2.*

limitations, we describe in Section 4 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 will relax all of the assumptions done for traditional ambient occlusion. In particular: a) we include diffuse interreflections of near-field occluders, for addressing assumption *i* (Section 4.2); b) we introduce a fast and accurate Lambertian directional occlusion approach, for the uniform dome assumption *ii* (Section 5); and c) we propose a formal directional and specular occlusion formulation, for relaxing the purely Lambertian surface assumption *iii* (Section 6).

4. GTAO: Ground Truth-based Ambient Occlusion

To develop an efficient model that accounts for near-field indirect illumination, we make the key observation that there is a functional relationship between the total ambient occlusion, the surface's albedo, and the indirect illumination reflected from x, as shown in Figure 2. This allows us to build a GI-aware ambient occlusion technique in two parts: First we compute the radiometrically-correct ambient occlusion at x assuming binary visibility (Section 4.1), and then we reintroduce the lost indirect illumination based on the computed ambient occlusion (Section 4.2). This has two main benefits: 1) We model interreflections based on a physically-plausible approximation, instead of an heuristic obscurance term; and 2) eliminating the empirical obscurance term allows us to reduce complexity by removing the piecewise inner integration, and consequently, for computations to be performed once per direction, rather than once per sample.

4.1. Computing ambient occlusion

Our formulation of ambient occlusion follows the horizon-based approach of Bavoil et al. [BSD08], which under the height field assumption computes Equation (2) as an integral along an azimuthal angle ϕ as

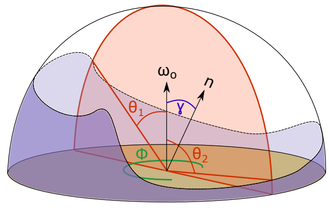

Figure 3: *Diagram of our reference frame when computing horizon-based ambient occlusion. Horizons angles θ_1 and θ_2 are drawn in red, slice angle ϕ is drawn in green, view direction $\omega_\mathbf{o}$ and normal $\mathbf{n}$ in black, and γ the angle between $\omega_\mathbf{o}$ and $\mathbf{n}$ in blue.*

$$\mathcal{A}(x) = \frac{1}{\pi}\int_0^{\pi}\int_{-\pi/2}^{\pi/2} V(\phi,\theta)\cos(\theta-\gamma)^+\,|\sin(\theta)|\,d\theta\,d\phi, \tag{5}$$

where θ is the polar angle along the view vector $\omega_\mathbf{o}$, γ is the angle between the normal $\mathbf{n}$ and the view vector $\omega_\mathbf{o}$ [TW10], $\cos(\theta)^+ = \max(\cos(\theta),0)$, and $V(\phi,\theta)$ is the visibility attenuation function. Note that unlike [BSD08], this integral is written here on its radiometrically-correct form, and hence accounting for the foreshortening factor. The coordinate system has also been changed by defining (ϕ,θ) with respect to the view vector $\omega_\mathbf{o}$ instead of the tangent vector, which requires introducing *abs* values to account for the *sin* term of the differential solid angle. Assuming a binary visibility function $V(\phi,\theta)$ that returns 1 when θ is above the horizon angles $\theta_1(\phi)$ and $\theta_2(\phi)$, and 0 below them (see Figure 3 for the reference system), and consequently not having per-sample attenuation, Equation (5) can be transformed as

$$\mathcal{A}(x) = \frac{1}{\pi}\int_0^{\pi}\underbrace{\int_{\theta_1(\phi)}^{\theta_2(\phi)}\cos(\theta-\gamma)^+\,|\sin(\theta)|\,d\theta}_{\hat{a}}\,d\phi. \tag{6}$$

Given the horizon angles θ_1 and θ_2 we can solve analytically the inner integral $\hat{a}$ in Equation (6) as

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

It is important to note that this formulation requires the normal $\mathbf{n}$ to lay in the plane P defined by the horizon vectors, which does not hold in general. However, it can be show that the following identity holds [Tim13b]:

$$\int_{-\pi/2}^{\pi/2}\langle\mathbf{n},\omega_\mathbf{i}\rangle^+\,|\sin(\theta)|\,d\theta = \|\overline{\mathbf{n}}\|\int_{-\pi/2}^{\pi/2}\langle\frac{\overline{\mathbf{n}}}{\|\overline{\mathbf{n}}\|},\omega_\mathbf{i}\rangle^+\,|\sin(\theta)|\,d\theta, \tag{8}$$

where $\overline{\mathbf{n}}$ is the *projected* normal in P. Combining with Equation (6) we obtain

$$\mathcal{A}(x) = \frac{1}{\pi}\int_0^{\pi}\|\overline{\mathbf{n}}\|\,\hat{a}(\theta_1(\phi),\theta_2(\phi),\gamma')\,d\phi, \tag{9}$$

where $\gamma' = \arccos(\langle\frac{\overline{\mathbf{n}}}{\|\overline{\mathbf{n}}\|},\omega_\mathbf{o}\rangle)$.

This analytic integral can be efficiently executed only once per direction. Additionally, after optimization only two `cos` and one `sin` are needed per sample, plus three additional `acos` functions per direction for setting up the integration domain, which can be efficiently approximated [Dro14].

Computing maximum horizon angles Core to the solution of Equation (9) is to find the maximum horizon angles $\theta_1(\phi)$ and $\theta_2(\phi)$ for a direction $\hat{\mathbf{t}}(\phi)$ in the image plane, where ϕ is the uniformly distributed azimuthal angle. We compute $\theta_1(\phi)$ by ray-tracing in screen-space from the projected pixel $\hat{x}$ of point x using $\hat{s}(r) = \hat{x} + \hat{\mathbf{t}}(\phi) \cdot r$, with $r \in [0,1]$ the parametrization of the ray. For each camera space point $s(r)$ we compute $\omega_\mathbf{s}(r) = \frac{s(r)-x}{\|s(r)-x\|}$. The maximum horizon angle with respect to the view vector $\omega_\mathbf{o}$ is then

$$\theta_1(\phi) = \arccos\left(\max_{r\in[0,1]} \left(\langle \omega_\mathbf{s}(r), \omega_\mathbf{o} \rangle^+\right)\right). \quad (10)$$

We compute a fixed number of discrete samples per direction. Angle $\theta_2(\phi)$ is computed analogously with $\hat{s}(r) = \hat{x} - \hat{\mathbf{t}}(\phi) \cdot r$. The maximum screen-space ray tracing distance r is scaled depending on the distance from the camera; this is necessary to make $\mathcal{A}(x)$ view-independent. We clamp the maximum r to avoid large gathering areas in objects close to the camera, which would trash the GPU cache. Algorithm 1 details GTAO computations.

4.2. Adding indirect illumination

Equation (9) matches the ground truth if we assume that the neighborhood of x only occludes light, and therefore no interrelections are present. This results into an energy lost, visible as an overdarkening at e.g. corners. In other words, Equation (9) computes Equation (2), but not the physically-accurate Equation (1), for which no analytical closed-form solution exists. However, as shown by Nayar et al. [NIK91], if we assume that the neighborhood $S(x)$ of x has constant albedo $\rho(x)$ and diffuse reflectance, we can express Equation (1) as a Neumann series as

$$L_r(x, \omega_\mathbf{o}) = L_i \frac{\rho(x)}{\pi} \mathcal{A}(x) + \sum_{m=1}^{\infty} \rho^m \int_{S(x)} K_m(x \leftarrow x') L_i(x') \mathrm{d}x', \quad (11)$$

where $K_m(x \leftarrow x')$ is the transfer function between x' and x. This equation relates the light incoming at the points in $S(x)$, the geometric relationship $K_m(x \leftarrow x')$ between x and the points $x' \in S(x)$, and the ground truth total light reflected at x. Introducing the assumption of uniform illumination L_i at $S(x)$, and following Stewart and Langer [SL96], we can find a closed-form solution for Equation (11) as

$$\begin{aligned} L_r(x, \omega_\mathbf{o}) &= L_i \frac{\rho}{\pi} \frac{\pi^{-1} \int_{\mathcal{H}^2} V(x, \omega_\mathbf{i}) \langle \mathbf{n}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i}}{1 - \rho\left(1 - \pi^{-1} \int_{\mathcal{H}^2} V(x, \omega_\mathbf{i}) \langle \mathbf{n}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i}\right)} \\ &= L_i \frac{\rho}{\pi} \frac{\mathcal{A}(x)}{1 - \rho(1 - \mathcal{A}(x))}. \end{aligned} \quad (12)$$

Equation (12) accurately computes the indirect illumination as a function of the ambient occlusion at x, with just a few algebraic operations. This allows to eliminate the need of ad-hoc obscurance operators when computing $\mathcal{A}(x)$.

Accuracy analysis We analyze the accuracy of Equation (12) by comparing it to Monte Carlo-based measurements of global illumination with respect to ambient occlusion (Figure 2). To analyze the average reflected radiance on that scenario and to reduce the effect of simulation variance, we fit a polynomial relating the albedo $\rho(x)$, the ambient occlusion term $\mathcal{A}(x)$, and the total reflected radiance illuminated by a uniform dome. Details can be found in Appendix B. As we can observe in Figure 4, Equation (12) provides a very accurate approximation of global illumination based on the surface's albedo and ambient occlusion for a reasonable range of surface albedos.

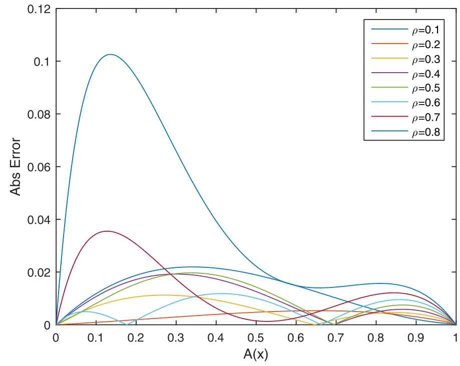

Figure 4: *Error comparison between the analytic approximation for indirect illumination in Equation (12) and the polynomial data-driven fitting derived from a set of simulations, with respect to the amount of ambient occlusion $\mathcal{A}(x)$, for different albedos ρ. Even for high albedo values in highly occluded areas (low $\mathcal{A}(x)$), where indirect illumination dominataes, the introduced error is below 10%.*

Figure 5: *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.*

4.3. Implementation details

For a game running at 60 frames per second, around half a millisecond is a reasonable screen-space ambient occlusion budget, which makes optimization mandatory. Similarly, working in screen space imposes some limitations.

Spatio-temporal sampling approach We compute our ambient occlusion on half-resolution, which is later bilaterally upsampled to full resolution. Moreover, in order to compute as many samples as possible without harming the performance, we distribute the occlusion integral over both space and time: We sample the horizon

Figure 6: *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 cast an unrealistic amount of occlusion, which is not temporaly consistent. Our simple heuristic allows for significantly disminissing the effect of such thin occluders.*

Figure 7: *Comparison between the ground truth-based ambient occlusion computed with Monte Carlo ray-tracing (left) and our method (right) without multiple scattering (Equation (9)). Our method closely matches the ground truth, while being significantly faster to compute.*

in only one direction per pixel (including both sides of a direction, with 12 steps in total) but use the information gathered on a neighborhood of 4×4 using a bilateral filter for reconstruction, using uniform convolution weights. To generate per-pixel directions we use a tileable spatial uniform noise of 4×4. 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 5 shows the effect of the spatial and temporal gathering on the final reconstruction. We opted for a regular sampling approach rather than using line sweeps [Tim13a] because it fitted better our tight budget and target quality (single direction of 12 steps per pixel). Line-sweep ambient occlusion achieves very high quality results, but unfortunately requires a high scan direction count to avoid banding, given the impossibility of randomizing directions per-pixel. Silvennoinen et al. [ST15] reported a cost of 1.6ms for a 1280×720 image on the Xbox One, representing a different tradeoff than our approach.

Bounding the sampling area As opposed to ambient obscurance techniques, in our formulation we do not use an attenuation function. However, we only want to calculate local ambient occlusion, as larger-range low-frequency occlusion can be computed using baked irradiance or occlusion. We compute near-field occlusion using our formulation, and combine it with baked far-field occlusion by calculating their minimum. In order to minimize artifacts we employ a conservative attenuation strategy. We linearly interpolate the current sample horizon angle cosine ($\cos(\theta)$) towards -1 when we exceed the near-field occlusion area, meaning that any sample outside of the near-field will be progressively attenuated, and with all the other samples remaining unmodified.

Height-field assumption considerations Screen-space techniques assume that the depth map is a heigh-field, which generally does not hold. As a result, thin features at depth discontinuities cast too much occlusion. 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 object is similar to its size in screen space. We introduce this heuristic by modifying the horizon search (Equation (10)): For each iteration i, the cosine of the maximum horizon angle $\bar{\theta}^i(\phi) = \cos(\theta^i(\phi))$ is updated using the sample at distance r_i as

$$\bar{\theta}^i(\phi) = \begin{cases} \langle \omega_{\mathbf{s}}(r_i), \omega_{\mathbf{o}} \rangle^+ & \text{if } \langle \omega_{\mathbf{s}}(r_i), \omega_{\mathbf{o}} \rangle^+ \geq \bar{\theta}^{i-1}(\phi) \\ \bar{\theta}^{i-1}(\phi) - \beta & \text{if } \langle \omega_{\mathbf{s}}(r_i), \omega_{\mathbf{o}} \rangle^+ < \bar{\theta}^{i-1}(\phi) \end{cases} \tag{13}$$

where β is a correction constant, and $\bar{\theta}^0 = -1$. Note that the superscript notation is used here to distinguish the iteration process from horizon angles θ_1 and θ_2 notation, with the overline being used in this equation to indicate cosine of the horizon angle. A single sample that is behind the horizon will not significantly decrease the computed horizon, but many of them (in e.g. a thin feature) will considerably attenuate it. This allows to progressively attenuate the occlusion on convex features by reducing the horizon angle, while leaving concavities unmodified in e.g. simple corners in indoor settings. For the correction to not affect small convex objects, such as facial features in a human, it is only applied when the sample distance to current maximum horizon is sufficient, and when it is not too far away from the sampling hemisphere base. Figure 6 shows the effect of this heuristic.

4.4. Results

We implement our GTAO it in an DirectX stand-alone application. In all cases, we compare against a Monte Carlo ray-traced reference. Figure 7 compares our GTAO without global illumination (ambient occlusion only) against the ground truth: Our technique matches the ray-traced results, while being practical for games at 1080p and 60 fps. Similarly, we compare our approximation to near-field global illumination against a path traced ground truth. Figure 8 shows a scene rendered with ambient occlusion only,

Figure 8: *Adding near-field global illumination to ambient occlusion: From left to right, HBAO [BSD08], our GTAO with ambient occlusion only, GTAO with our global illumination approximation for gray albedo, our GTAO with colored indirect illumination, and path traced Monte Carlo ground truth in a surface with colored albedo. Our approximation model for diffuse interreflections based on ambient occlusion matches very closely the ground truth, and it is able to recover the energy lost by assuming one-bounce illumination only.*

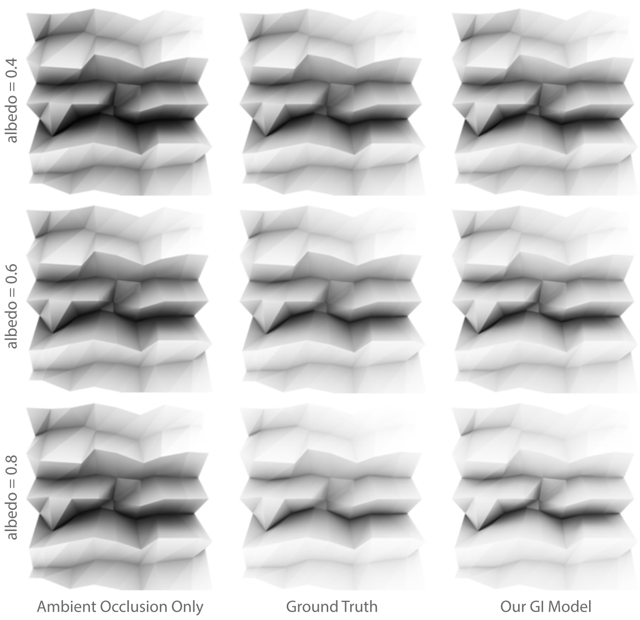

Figure 9: *Effect of albedo in our ambient occlusion-based global illumination approximation, for the* groove *scene. From left to right: GTAO only, Monte Carlo ground truth, and our approximation based on GTAO, for albedos 0.4, 0.6 and 0.8.*

and then including global illumination with the analytical model proposed in Section 4.2, both for gray and colored albedos. Figure 9 shows the same comparison in an abstract groove-like shape, with increasing values of gray albedo. In both cases, for a uniform distant illumination our technique delivers similar results to the ground truth, while rendering it in a tight practical real-time budget.

5. Directional GTAO

So far we have assumed a uniform infinite light source (i.e. a colored probe). Unfortunately, this approach is too simplifying in practical conditions, specially with the widespread use of light probes for ambient illumination. To account for that, we need to recover the directional component of the light in Equation (1), while still being able to retain real-time performance. Let us approximate Equation (1) for diffuse reflectance and distant illumination as

$$L_r(x,\omega_\mathbf{o}) \approx \frac{\rho(x)}{\pi}\int_{\mathcal{H}^2} V(x,\omega_\mathbf{i})L(\omega_\mathbf{i})\langle\mathbf{n},\omega_\mathbf{i}\rangle^{+}\mathrm{d}\omega_\mathbf{i}, \tag{14}$$

with $L(\omega_\mathbf{i})$ the light incoming from an infinitely far lighting environment (light probe), and $V(x,\omega_\mathbf{i})$ its visibility. To solve Equation (14), we project the terms of the integral as their spherical harmonics expansion [RH01] as

$$L_r(x,\omega_\mathbf{o}) \approx \frac{\rho(x)}{\pi}\int_{\mathcal{H}^2}\left(\sum_j \widehat{L}_j\, y_j(\omega_\mathbf{i})\right)\left(\sum_j \widehat{V}'_j\, y_j(\omega_\mathbf{i})\right)\mathrm{d}\omega_\mathbf{i} \tag{15}$$

$$= \sum_j \widehat{L}_j \widehat{V}'_j, \tag{16}$$

where $\widehat{L}_j$ and $\widehat{V}'_j$ are the j-th term of the SH expansion of $L(\omega_\mathbf{i})$ and $V'(x,\omega_\mathbf{i})$ respectively, with $V'(x,\omega_\mathbf{i}) = V(x,\omega_\mathbf{i})\langle\mathbf{n},\omega_\mathbf{i}\rangle^{+}$, and y_j is the j-th spherical harmonics basis function. Assuming that the visibility $V(x,\omega_\mathbf{i})$ can be approximated by a cone centered at the *bent normal* $\mathbf{b}$ [Lan02] with aperture angle α_v defined as a function of the AO term $\mathcal{A}$, we can project both the visibility and the dot product in *zonal harmonics* [Slo08], which can be computed efficiently in runtime, and from their expansion compute $\hat{V}'$. This allows to compute Equation (14) as a simple dot product between the expansion of $L(\omega_\mathbf{i})$ and $V'(x,\omega_\mathbf{i})$.

Zonal Harmonics Zonal harmonics [Slo08] are the projection on spherical harmonics for functions that have rotational symmetry around an axis. They only contain non-zero information for the central coefficients of the expansion (i.e. for $m=0$ in y_l^m). The key advantage is that they can be efficiently rotated to a new direction $\omega_\mathbf{i}$ at runtime following:

$$f_l^m = \sqrt{\frac{4\pi}{2l+1}} z_l\, y_l^m(\omega_\mathbf{i}), \tag{17}$$

where f_l^m is the rotated spherical harmonics coefficient, z_l is the zonal harmonic coefficient of level l, and $y_l^m(\omega_\mathbf{i})$ is the spherical harmonics basis for direction $\omega_\mathbf{i}$.

Computing the bent normal b We compute $\mathbf{b}$ using a radiometric formulation weighted by the cosine as

$$\mathbf{b} = \int_{\mathcal{H}^2} \omega_\mathbf{i}\, V(x,\omega_\mathbf{i}) \langle \mathbf{n}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i}. \tag{18}$$

We compute Equation (18) using a similar approach to Equation (6), following the horizon-based approximation as

$$\mathbf{b} = \int_0^\pi \int_{\theta_1(\phi)}^{\theta_2(\phi)} \omega_\mathbf{i}(\theta,\phi) \cos(\theta-\gamma)^+ |\sin(\theta)| \mathrm{d}\theta \mathrm{d}\phi, \tag{19}$$

with $\omega_\mathbf{i}(\theta,\phi)$ the direction defined by the polar coordinates (θ,ϕ). We integrate the vertical slices defined by the rotation angle ϕ, while the inner integral over θ can be solved analytically for each component of $\mathbf{b}$ as

$$\begin{aligned}
\mathbf{b}_x &= \cos(\phi) \int_{\theta_1(\phi)}^{\theta_2(\phi)} \sin(\theta)\cos(\theta-\gamma)^+ |\sin(\theta)| \mathrm{d}\theta = \cos(\phi)\, \hat{v}_{xy}(\phi),\\
\mathbf{b}_y &= \sin(\phi) \int_{\theta_1(\phi)}^{\theta_2(\phi)} \sin(\theta)\cos(\theta-\gamma)^+ |\sin(\theta)| \mathrm{d}\theta = \sin(\phi)\, \hat{v}_{xy}(\phi),\\
\mathbf{b}_z &= \int_{\theta_1(\phi)}^{\theta_2(\phi)} \cos(\theta)\cos(\theta-\gamma)^+ |\sin(\theta)| \mathrm{d}\theta\\
&= \frac{1}{12}\big(-\cos(3\theta_1(\phi)-\gamma) - \cos(3\theta_2(\phi)-\gamma) + 8\cos(\gamma)\\
&\quad - 3\left(\cos(\theta_1(\phi)+\gamma) + \cos(\theta_2(\phi)+\gamma)\right)\big),
\end{aligned} \tag{20}$$

where $\hat{v}_{xy}(\phi)$ can be analytically solved as

$$\begin{aligned}
\hat{v}_{xy}(\phi) = \frac{1}{12}\big(&6\sin(\theta_1(\phi)-\gamma) - \sin(3\theta_1(\phi)-\gamma)\\
&+ 6\sin(\theta_2(\phi)-\gamma) - \sin(3\theta_2(\phi)-\gamma) + 16\sin(\gamma)\\
&- 3\left(\sin(\theta_1(\phi)+\gamma) + \sin(\theta_2(\phi)+\gamma)\right)\big).
\end{aligned} \tag{21}$$

Note that $\mathbf{b}$ needs to be normalized after it is computed. We calculate Equation (20) at the same time as the ambient occlusion term $\mathcal{A}$, following the approach and implementation described in Section 4, and detailed in Algorithm 2.

Computing $V'(x,\omega_\mathbf{i})$ In order to compute the spherical harmonics expansion of $V'(x,\omega_\mathbf{i})$ efficiently we leverage the speed of zonal harmonics. We approximate visibility as a visibility cone; thus, both $V(x,\omega_\mathbf{i})$ and $\langle \mathbf{n}, \omega_\mathbf{i}\rangle^+$ are radially symmetric with respect a particular axis. In particular, the visibility cone approximating $V(x,\omega_\mathbf{i})$ is symmetric with respect the bent normal [Lan02], and the dot product is symmetric with respect the normal at x.

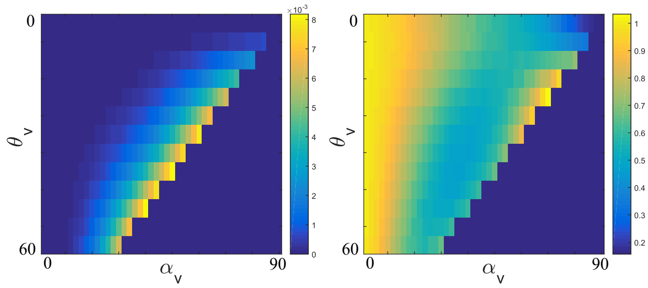

Figure 10: *Left: Error introduced by computing the cosine term in $V'(x,\omega_\mathbf{i})$ with respect to the bent normal in zonal harmonics, for a 3 levels SH expansion, for the visibility cone aperture α_v and the angle between the bent normal and normal of θ_v. Right: Ratio of the error introduced by our technique, with respect to the error introduced by the triple product approximation, for a 3 levels SH expansion, where 1 is equal performance, and below 1 means that our technique has less error. Our technique introuces less error than the triple product for most cases, while being more efficient.*

We compute the visibility cone in run-time, by computing the bent normal, and an ambient occlusion term. These two terms allow us to compute the visibility cone centered at the bent normal, with an aperture angle derived from the visibility as (see Appendix A for details)

$$\alpha_v(x) = \arccos\left(\sqrt{1-\mathcal{A}(x)}\right). \tag{22}$$

Then, by assuming for efficiency that the dot product is computed with respect to the bent normal instead of the geometric normal, we can compute the zonal harmonics expansion of $V'(x,\omega_\mathbf{i})$ as

$$\begin{aligned}
z_0 &= \frac{\sqrt{\pi}}{2}\sin(\alpha_v(x))^2,\\
z_1 &= \frac{\sqrt{3\pi}}{3}\left(1-\cos(\alpha_v(x))^3\right),\\
z_2 &= \frac{\sqrt{5\pi}}{16}\sin(\alpha_v(x))^2\left(2+6\cos(\alpha_v(x))^2\right),
\end{aligned} \tag{23}$$

where only the first three coefficients are shown. This formulation introduces error, since we use the bent normal to compute the cosine term, instead of the normal at x. However, we observed that the divergence between them is not very large (see Figure 10, left). Moreover, computing $V'(x,\omega_\mathbf{i})$ instead of multiplying the SH projections of $V(x,\omega_\mathbf{i})$ and the cosine term, gives significant more accurate approximation for a practical low-order SH expansion: The cosine term smooths the step function $V(x,\omega_\mathbf{i})$, that is better approximated using SH (see Figure 10, right).

5.1. Results

We compare our method for directional occlusion against the standard non-directional AO approximation, the bent normal approx-

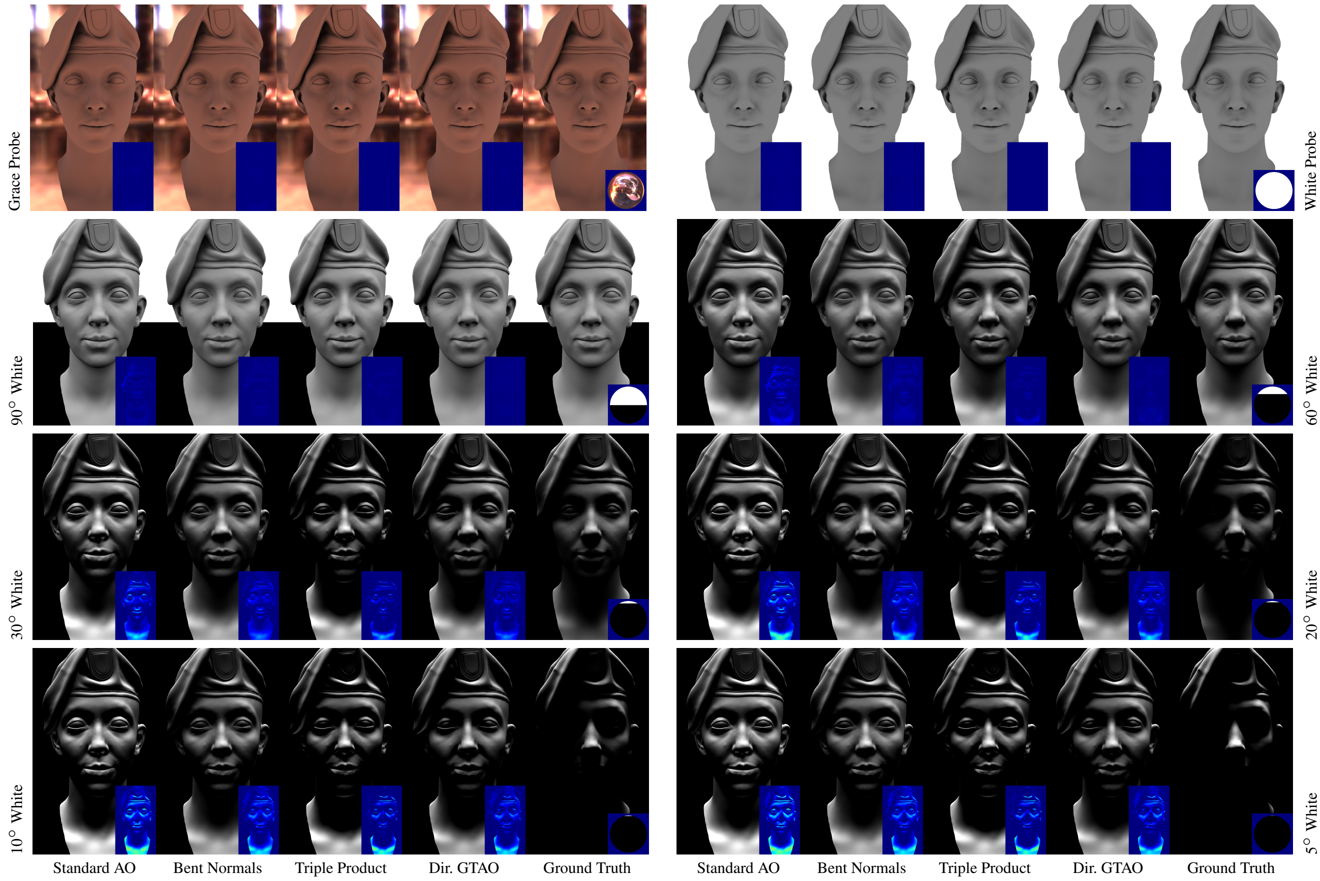

Figure 11: *Error analysis for directional ambient occlusion under illumination with varying frequency, for standard AO (Section 4), standard AO using the bent normal to fetch the environment map [Lan02], the triple product approximation [Sny06], and our work (Dir. GTAO). The insets show the per-pixel error (MSRE for each method is shown in Figure 12).*

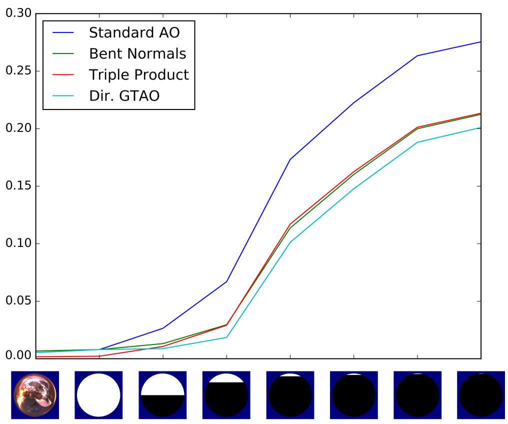

Figure 12: *MRSE for the results in Figure 11*

imation [Lan02], and the triple product approximation [Sny06]. The former two are computed by fetching a pre-filtered environment map, while the triple product and our directional GTAO use a three-levels SH expansion (SH9) of the probe. Figure 11 shows the results of such comparison for different probes, with increasing frequency. The standard AO approximation quickly fails to capture the directional behaviour of light, while the un-expensive bent normals approximation performs similarly as the triple product; our directional GTAO performs the best in all scenarios both qualitatively and quantitatively (MRSE, see Figure 12).

In terms of cost, we evaluate the performance on a GCN platform, by measuring the final ISA instructions for a dedicated pixel shader running only the occlusion code. The bent normal is given as a pixel shader input, and could come in practice from either our GTAO screen-space approach, or baked offline. Table 1 shows the results. Not surprisingly, the non-directional AO and the bent normal approximation on a prefiltered probe are the cheapest options. Modeling the probe using a SH expansion almost doubles their cost; the cost of our technique is comparable with these simpler techniques, while introducing significantly less error. Note that the actual cost of those techniques is not directly proportional to the

	#ISA	**VMEM**	**Cycles**	**VGPR**
Standard AO (Probe)	39	1	357	8
Bent Normal (Probe)	39	1	357	8
Standard AO (SH9)	74	0	642	12
Bent Normal (SH9)	74	0	642	12
Triple Product (SH9)	452	0	2053	44
Dir. GTAO (SH9)	90	0	722	12

Table 1: *Performance comparison for different techniques approximating directional AO on a GCN platform on a dedicated pixel shader. We measure the number of ISA instructions, use of VMEM, total cycles, and register pressure (VGPR). Our directional GTAO is comparable to simpler ones working with SH.*

cycles they consume on isolation and will depend on which shader (and where) they are located, as often cost can be hidden by other operations. It is interesting to observe that the bent normal approximation results in low error; this suggests that when using a pre-filtered environment map it could be the technique of choice. However, in cases when using SH9 to encode the light probe, our technique introduces minimal overhead over simpler techniques, while reducing significantly the error.

6. Specular Occlusion

Here we generalize classic Lambertian-based ambient occlusion, by proposing its specular counterpart. We develop an illumination model where the near-field occlusion modulates the distant lighting while supporting arbitrary BRDF models (e.g. microfacets). Moreover, for the specific cases of uniform dome illumination, our model delivers ground truth results.

Let us assume that all light incides from an infinitely far lighting environment (light probe) to express Equation (1) as

$$L_r(x,\omega_\mathbf{o}) = \int_{\mathcal{H}^2} V(x,\omega_\mathbf{i}) L(x,\omega_\mathbf{i}) f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}. \quad (24)$$

Computing this integral by numerical integration is too expensive for real-time applications. Equation (24) is the generalization of Equation (14) to arbitrary BRDFs, and could be computed following a similar procedure as in Section 5; however, in order to support all-frequency BRDFs a large number of coefficients in the SH expansion would be required, reducing significantly the performance. The current state-of-the-art assumes uniform perfect visibility ($\forall \omega_\mathbf{i} | V(x,\omega_\mathbf{i}) = 1$) and uses a split-integral approximation [Laz13, Kar13] as

$$\begin{aligned} L_r(x,\omega_\mathbf{o}) &\approx \mathcal{L}(x)\cdot\mathcal{F}(x,\omega_\mathbf{o}),\\ \mathcal{L}(x) &= \frac{1}{C_L}\int_{\mathcal{H}^2} \overbrace{V(x,\omega_\mathbf{i})}^{=1} L(x,\omega_\mathbf{i}) D(x,\omega_\mathbf{h}) \langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i},\\ \mathcal{F}(x,\omega_\mathbf{o}) &= \int_{\mathcal{H}^2} f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}, \end{aligned} \quad (25)$$

where $D(x,\omega_\mathbf{h})$ is the normal distribution function of the surface [TS67], $\omega_\mathbf{h}$ is the half vector, and $C_L = \int_{\mathcal{H}^2} D(x,\omega_\mathbf{h})\langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}$ is the normalization factor needed in the first integral to guarantee it is always in the range $[0,1]$ when $L(x,\omega_\mathbf{i}) = 1$. Intuitively, the second line of Equation (25) is the full microfacet BRDF at the pixel under uniform light, that can be stored in a pre-computed lookup table (typically referred to as *environment lut*). The first integral, on the other hand, is the convolution of the distant environment light $L(x,\omega_\mathbf{i})$ with a circularly symmetric kernel that approximates the NDF of the microfacets. This first integral can be precomputed by convolving the distant illumination (e.g. a cubemap) with lobes from different surfaces roughness, making it very efficient for rendering glossy materials. However, most approximations ignore occlusion or approximate it with heuristics.

In order to account for occlusion in specular lighting, we opt for an approach similar to the split-integral approximation in Equation (25). We separate the visibility term from the first integral as a constant, to modulate the amount of illumination reaching x. This allows us to transform Equation (25) into a product of three integrals, or our *triple-split-integral* approximation:

$$L_r(x,\omega_\mathbf{o}) \approx \mathcal{S}(x,\omega_\mathbf{o})\cdot\mathcal{L}(x)\cdot\mathcal{F}(x,\omega_\mathbf{o}), \quad (26)$$

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

$$\mathcal{S}(x,\omega_\mathbf{o}) = \frac{1}{C_V}\int_{\mathcal{H}^2} V(x,\omega_\mathbf{i}) f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}, \quad (27)$$

with the normalization term $C_V = \int_{\mathcal{H}^2} f_r(x,\omega_\mathbf{i},\omega_\mathbf{o})\langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}$ ensuring that the specular occlusion $\mathcal{S}$ ranges into $[0,1]$. Our definition of specular occlusion is weighted by the BRDF, and thus is directionally dependent. This weight was carefully chosen for Equation (26) to match the ground truth for uniform illumination. The normalization factor C_V is the same as the latter integral $\mathcal{F}$, and thus it cancels out if substituting $\mathcal{S}$ into Equation (26) reducing $L_r(x,\omega_\mathbf{o})$ to

$$L_r(x,\omega_\mathbf{o}) \approx \mathcal{L}(x)\cdot\int_{\mathcal{H}^2} V(x,\omega_\mathbf{i}) f_r(x,\omega_\mathbf{i},\omega_\mathbf{o}) \langle \mathbf{n},\omega_\mathbf{i}\rangle^+ \mathrm{d}\omega_\mathbf{i}. \quad (28)$$

In this form, we can observe that for a uniform distant illumination ($\mathcal{L}(x) = 1$) it matches exactly the ground truth expressed by Equation (24).

Figure 13 shows the differences between our approximation in Equation (26) and the raytraced ground truth: For a constant probe, our formulation of specular occlusion models Equation (1) exactly, while for environment probes, it results into a faithful approximation of the rendering equation, specially for specular materials. In the following section, we describe a technique for solving Equation (27) practically for highly demanding applications.

7. GTSO: Ground Truth-based 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 follow the same procedure as in Section 5, and build a visibility cone centered in the bent normal $\mathbf{b}$ and with amplitude angle derived from the ambient occlusion term $\mathcal{A}(x)$ using Equation (22). Similarly, we can model the **specular** lobe as a

Figure 13: *Comparison between ground truth specular illumination and our specular occlusion model under two different illumination setups, for increasing roughness of the GGX microfacet BRDF. From top to bottom: Our specular occlusion (Equation (26)) under environment lighting, ground truth result under the same enviroment light, our specular occlusion with constant illumination, and ground the rendering equation under the same white probe. For constant illumination, our specular occlusion model exactly models the rendering equation.*

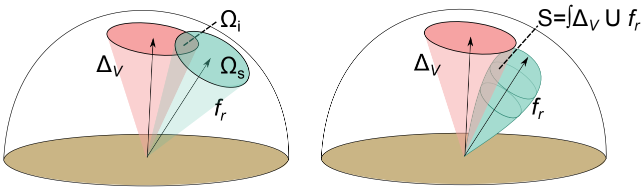

Figure 14: *Geometry of our specular occlusion, assuming that both the visibility and the specular are modeled as cones (left), and with accurate specular lobe (right).*

cone centered on the reflection direction $\omega_\mathbf{r}$, and compute $\mathcal{S}(x,\omega_\mathbf{o})$ as the intersection of the visibility and BRDF cones (see Figure 14, left, and Appendix C for more details on this approach).

Unfortunately, in a tight-bounded real-time application, these computations are still expensive. Furthermore, we have found specular lobes to be poorly represented by cones. To improve on both qualities, we opt for a more accurate approximation by precomputing the specular occlusion $\mathcal{S}$ as the product of the visibility cone Δ_V and the actual BRDF F (Figure 14, right):

$$\mathcal{S}(x,\omega_\mathbf{o}) \approx \frac{1}{C_V}\int_{\mathcal{H}^2} \Delta_V(\alpha_v(x),\beta(\mathbf{b}(x),\omega_\mathbf{r})) \, f_r(x,\omega_\mathbf{i},\omega_\mathbf{o})\langle\mathbf{n},\omega_\mathbf{i}\rangle^+ d\omega_\mathbf{i}, \quad (29)$$

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

$$\mathcal{S}(\alpha_v,\beta,r,\theta_o) \approx \frac{1}{C_V}\int_{\mathcal{H}^2} \Delta_V(\alpha_v,\beta) f_r(\omega_\mathbf{i},\theta_o,r)\langle\mathbf{n},\omega_\mathbf{i}\rangle^+ d\omega_\mathbf{i}. \quad (30)$$

This function can be compactly baked as a four-dimensional table if assuming a reflectance at normal incidence of 0.04. Given that the function is relatively smooth, we can encode it to a four-dimensional 32^4 BC4 8-bit look up table, which can be efficiently accessed in runtime. While not explored in this work, this lookup table for $\mathcal{S}(x,\omega_\mathbf{o})$ could be merged with the lookup table often used for $\mathcal{F}(x,\omega_\mathbf{o})$.

7.1. Results

Figure 15 compares the results of our GTSO implementation using a 4D look-up table for computing the specular occlusion, compared against the ground truth and the empirically-based technique described in Lagarde [Ld14]. For any roughness parameter of the microfacet BRDF, the introduced error is minimal.

8. Conclusions

In this work we have presented several contribution to 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. We implement our technique efficiently, by aggressively making 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 as a function of the local albedo and the ambient occlusion. Then, we have generalized our GTAO to

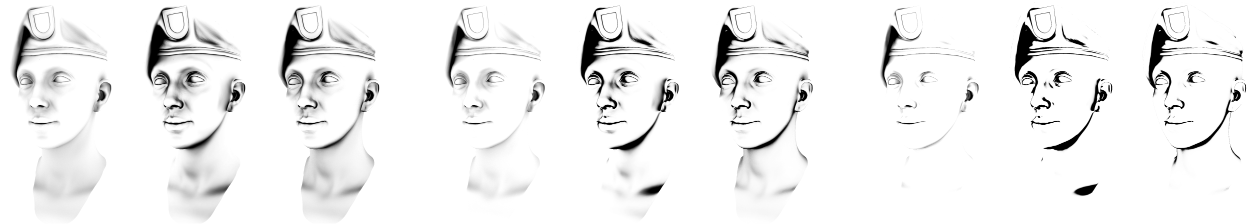

Figure 15: *Comparison of Lagarde specular occlusion [Ld14], the Monte Carlo ground truth and our GTSO using a four-dimensional look-up table, for a GGX microfacet BRDF with roughness $r = 0.4, 0.13$ and 0.02. The ground truth case shows $\mathcal{S}(x, \omega_\mathbf{o})$ for traced visibility $V(x, \omega_\mathbf{i})$, whereas GTSO shows it for our cone-based, lookup table approximation.*

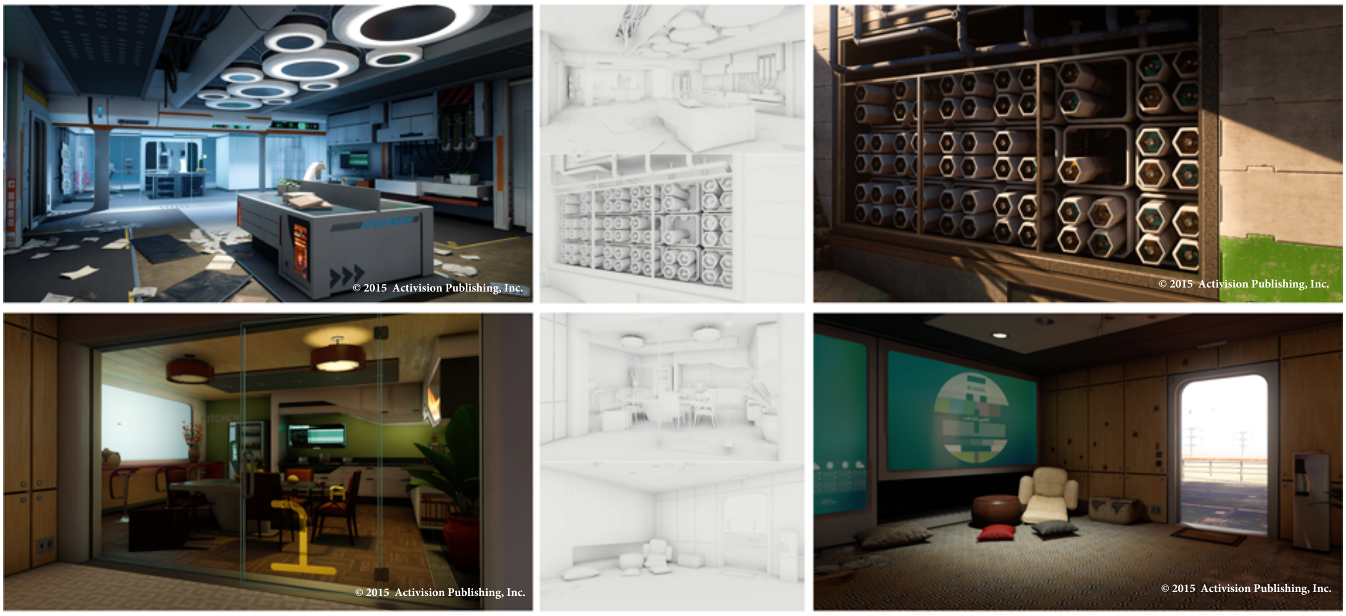

Figure 16: *Screenshots of our GTAO being used in-game for accurate and efficient ambient occlusion, in scenes with high-quality physically-based shading and high geometric complexity. Our GTAO computes the ambient occlusion layer (in the insets) in just 0.5 ms for PS4.*

a directional spherical harmonics-based generalization, that leverages zonal harmonics and efficient on-line computation of SH-based Lambertian 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 uniform and non-uniform probe-based illumination.

As shown in Figure 1 combining all our techniques results into a complete solution for efficient probe-based illumination, allowing to match the raytraced ground truth. The near-field indirect illumination, directional GTAO, and GTSO base on the results obtained using our efficient implementation of ambient occlusion (GTAO), resulting in very optimized techniques targeting very tight time budgets, like videogames, even for current console platforms (Figure 16).

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; Infinity Realities, in particular Lee Perry-Smith, for the head model. A. Jarabo is funded by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation programme (project CHAMELEON, grant No 682080), DARPA (project REVEAL), and a Leonardo Grant from the BBVA Foundation.

References

[AB12] Aalund F. P., Bærentzen J. A.: *A Comparative Study of Screen-Space Ambient Occlusion Methods*. Tech. rep., DTU, 2012. 2

[BSD08] Bavoil L., Sainz M., Dimitrov R.: Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 Talks* (2008). 2, 3, 4, 7

[Dro14] DROBOT M.: Low level optimization for gcn. In *Digital Dragons 2014* (2014). 5

[Got12] GOTANDA Y.: Practical physically based rendering in real-time. In *GDC* (2012). 3

[Gre03] GREEN R.: Spherical harmonic lighting: The gritty details. In *Archives of the Game Developers Conference* (2003), vol. 56, p. 4. 3

[HSEE15] HENDRICKX Q., SCANDOLO L., EISEMANN M., EISEMANN E.: Adaptively layered statistical volumetric obscurance. In *Proceedings of the 7th Conference on High-Performance Graphics* (2015), ACM, pp. 77–84. 2

[Jv13] JIMENEZ J., VON DER PAHLEN J.: Next generation character rendering. In *GDC* (2013). 3

[Kar13] KARIS B.: Real shading in Unreal Engine 4. In *ACM SIGGRAPH 2013 Courses* (2013). 10

[KRES11] KLEHM O., RITSCHEL T., EISEMANN E., SEIDEL H.-P.: Bent normals and cones in screen-space. In *VMV* (2011), Citeseer, pp. 177–182. 3

[Lan02] LANDIS H.: Production-ready global illumination. In *ACM SIGGRAPH 2002 Courses* (2002). 3, 8, 9

[Laz13] LAZAROV D.: Getting more physical in call of duty: Black ops ii. In *ACM SIGGRAPH 2013 Courses* (2013). 10

[Ld14] LAGARDE S., DE ROUSIERS C.: Moving Frosbite to physically-based rendering. In *ACM SIGGRAPH 2014 Courses* (2014). 3, 11, 12

[LS10] LOOS B. J., SLOAN P.-P.: Volumetric obscurance. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), ACM, pp. 151–156. 2

[Maz12] MAZONKA O.: Solid angle of conical surfaces, polyhedral cones, and intersecting spherical caps. *arXiv preprint arXiv:1205.1396* (2012). 14

[MH99] MÖLLER T., HUGHES J. F.: Efficiently building a matrix to rotate one vector to another. *Journal of Graphics Tools 4*, 4 (1999), 1–4. 15

[Mit07] MITTRING M.: Finding next gen: Cryengine 2. In *ACM SIGGRAPH 2007 Courses* (2007). 2, 3

[MML12] MCGUIRE M., MARA M., LUEBKE D.: Scalable ambient obscurance. In *Proceedings of the Fourth ACM SIGGRAPH/Eurographics conference on High-Performance Graphics* (2012), Eurographics Association, pp. 97–103. 2

[MOBH11] MCGUIRE M., OSMAN B., BUKOWSKI M., HENNESSY P.: The alchemy screen-space ambient obscurance algorithm. In *Proc. of the ACM SIGGRAPH Symposium on High Performance Graphics* (2011), ACM, pp. 25–32. 2

[NIK91] NAYAR S. K., IKEUCHI K., KANADE T.: Shape from interreflections. *International Journal of Computer Vision 6*, 3 (1991), 173–195. 5

[OS07] OAT C., SANDER P. V.: Ambient aperture lighting. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games* (2007), ACM, pp. 61–64. 3, 14

[RDGK12] RITSCHEL T., DACHSBACHER C., GROSCH T., KAUTZ J.: The state of the art in interactive global illumination. *Computer Graphics Forum 31*, 1 (2012), 160–188. 2

[REG*09] RITSCHEL T., ENGELHARDT T., GROSCH T., SEIDEL H.-P., KAUTZ J., DACHSBACHER C.: Micro-rendering for scalable, parallel final gathering. *ACM Transactions on Graphics (TOG) 28*, 5 (2009), 132. 3

[RH01] RAMAMOORTHI R., HANRAHAN P.: An efficient representation for irradiance environment maps. In *SIGGRAPH '01* (2001). 3, 7

[SKUT*10] SZIRMAY-KALOS L., UMENHOFFER T., TÓTH B., SZÉCSI L., SBERT M.: Volumetric ambient occlusion for real-time rendering and games. *IEEE Computer Graphics and Applications 30*, 1 (2010), 70–79. 2

[SL96] STEWART A. J., LANGER M. S.: Towards accurate recovery of shape from shading under diffuse lighting. In *Proceedings of CVPR'96* (1996), IEEE, pp. 411–418. 5

[Slo08] SLOAN P.-P.: Stupid spherical harmonics (SH) tricks. In *Game developers conference* (2008), vol. 9, Citeseer. 8

[Sny06] SNYDER J.: *Code Generation and Factoring for Fast Evaluation of Low-order Spherical Harmonic Products and Squares*. Tech. Rep. MSR-TR-2006-53, Microsoft Research, 2006. 9

[ST15] SILVENNOINEN A., TIMONEN V.: Multi-scale global illumination in quantum break. In *ACM SIGGRAPH 2015 Courses* (2015). 2, 6

[Tim13a] TIMONEN V.: Line-sweep ambient obscurance. *Computer Graphics Forum 32*, 4 (2013), 97–105. 2, 6

[Tim13b] TIMONEN V.: Screen-space far-field ambient obscurance. In *Proceedings of the 5th High-Performance Graphics Conference* (2013), ACM, pp. 33–43. 2, 4

[TS67] TORRANCE K. E., SPARROW E. M.: Theory for off-specular reflection from roughened surfaces. *JOSA 57*, 9 (1967), 1105–1112. 10

[TW10] TIMONEN V., WESTERHOLM J.: Scalable height field self-shadowing. *Computer Graphics Forum 29*, 2 (2010), 723–731. 4

[Ulu14] ULUDAG Y.: Hi-z screen-space cone-traced reflections. In *GPU Pro 5: Advanced Rendering Techniques*. CRC Press, 2014, p. 149. 14

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *Proc. of EGSR '07* (2007). 11, 14

[ZIK98] ZHUKOV S., IONES A., KRONIN G.: An ambient light illumination model. In *Rendering Techniques' 98* (1998), Springer, pp. 45–55. 2, 3

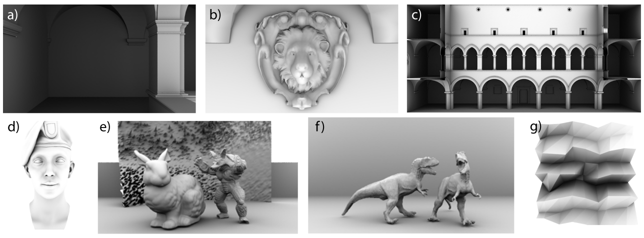

Figure 17: *Input scenes used for computing the mapping between the ambient occlusion and the near-field global illumination, rendered using only ambient occlusion.*

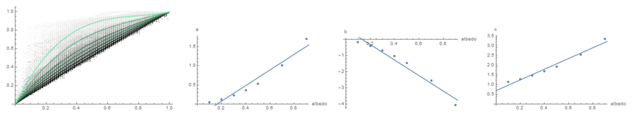

Figure 18: *Cubic fit for our mapping between the ambient occlusion and the three-bounce global illumination for different albedos (left). We observed that a linear fit between the coefficients of the polynomial wrt the albedo gives a good continuous fit, as shown in the three rightmost figures. The combination of these fits give form to our model (Equation (33)).*

Appendix A: Aperture Calculation

The visibility equation for a cone can be calculated as follows:

$$\mathcal{A}(x) = \frac{1}{\pi}\int_0^{2\pi}\left(\int_0^{\alpha_v(x)} \cos(\theta)\sin(\theta)\,d\theta\right) d\phi = 1-\cos(\alpha_v(x))^2 \quad (31)$$

where solving for $\alpha_v(x)$ yields the equation to convert from occlusion to aperture angles:

$$\alpha_v(x) = \arccos(\sqrt{1-\mathcal{A}(x)}) \quad (32)$$

Appendix B: Polynomial Fitting of Global Illumination

Based on the observation that there is a relationship between ambient occlusion and global illumination exists (Figure 2), 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 showing a variety of different types of occlusion conditions (see Figure 17). We compute both the ambient occlusion and multi-bounce indirect illumination (in our case, up to three bounces). By taking the combination of all points, we fit this mapping using a cubic polynomial for each albedo (Figure 18 (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 18). This last observation allows us to build a bidimensional mapping between the albedo ρ and ambient occlusion $\mathcal{A}$:

$$\begin{aligned}\mathcal{G}(\mathcal{A},\rho) &= a(\rho)\,\mathcal{A}^3 - b(\rho)\,\mathcal{A}^2 + c(\rho)\,\mathcal{A},\\ a(\rho) &= 2.0404\,\rho - 0.3324,\\ b(\rho) &= 4.7951\,\rho - 0.6417,\\ c(\rho) &= 2.7552\,\rho + 0.6903. \end{aligned} \quad (33)$$

Appendix C: Analytical Cone-to-Cone Specular Occlusion

Specular occlusion can be computed as the ratio between the intersection of the visibility and specular cones Ω_i, and the specular cone Ω_s (see Figure 14, left):

$$\mathcal{S}(x,\omega_o) = \frac{\Omega_i(x,\omega_o)}{\Omega_s(x,\omega_o)}, \quad (34)$$

We then need to compute the visibility and specular cones, defined by a direction and an aperture, and their intersection solid angle Ω_i. To leverage previous work on mappings from specular lobes to cones, Phong is used instead of GGX on the experiments described in this section.

The visibility cone is explained in Section 6 (Equation (22)). 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, 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:

$$\alpha_s = \arccos\left(u^{\frac{1}{p+2}}\right), \quad (35)$$

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 $u = 0.01$.

Once we have both cones, we can compute their intersection solid angle Ω_i. 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 $\omega_\mathbf{r}$.

Algorithm 1 Computes the ambient occlusion term $\mathcal{A}(x)$.

```
cPosV ← VIEWSPACEPOSFROMDEPTHBUFFER(cTexCoord)                          ▷ We will abbreviate center with c
viewV ← NORMALIZE(-cPosV)
visibility ← 0
for slice ∈ [0, sliceCount) do
    ϕ ← (π/sliceCount) ∗ slice
    ω ← {cos ϕ, sin ϕ}

    directionV ← {ω[0], ω[1], 0}
    orthoDirectionV ← directionV − DOT(directionV, viewV) ∗ viewV
    axisV ← CROSS(directionV, viewV)
    projNormalV ← normalV − axisV ∗ DOT(normalV, axisV)

    sgnN ← SIGN(DOT(orthoDirectionV, projNormalV))
    cosN ← SATURATE(DOT(projNormalV, viewV)/LEN(projNormalV))
    n ← sgnN ∗ arccos(cosN)

    for side ∈ [0, 1] do                                                 ▷ Equation (13)
        cHorizonCos ← −1
        for sample ∈ [0, directionSampleCount) do
            s ← sample/directionSampleCount
            sTexCoord ← cTexCoord + (−1 + 2 ∗ side) ∗ s ∗ scaling ∗ {ω[0], −ω[1]}   ▷ Flip y due to texture coordinate system
            sPosV ← VIEWSPACEPOSFROMDEPTHBUFFER(sTexCoord)
            sHorizonV ← NORMALIZE(sPosV − cPosV)
            cHorizonCos ← MAX(cHorizonCos, DOT(sHorizonV, viewV))
        end for

        h[side] ← n + CLAMP((−1 + 2 ∗ side) ∗ arccos(cHorizonCos) − n, −π/2, π/2)   ▷ Horizon angle θ_i
        visibility ← visibility + LEN(projNormalV) ∗ (cosN + 2 ∗ h[side] ∗ sin(n) − cos(2 ∗ h[side] − n))/4   ▷ Equation (7)
    end for
end for
visibility ← visibility/sliceCount
```

Algorithm 2 Extension that computes bent normals **b**. Repeated code from ambient occlusion algorithm is omitted.

```
...
for slice ∈ [0, sliceCount) do
    ...                                                                  ▷ Equations (20) and (21)
    t[0] ← (6∗sin(h[0]−n) − sin(3∗h[0]−n) + 6∗sin(h[1]−n) − sin(3∗h[1]−n) + 16∗sin(n) − 3∗(sin(h[0]+n) + sin(h[1]+n)))/12
    t[1] ← (−cos(3∗h[0]−n) − cos(3∗h[1]−n) + 8∗cos(n) − 3∗(cos(h[0]+n) + cos(h[1]+n)))/12
    bentNormalL ← {ω[0]∗t[0], ω[1]∗t[0], −t[1]}                           ▷ Flip z due to change of handedness
    bentNormalV ← bentNormalV + MULT(bentNormalL, ROTFROMTOMATRIX({0, 0, −1}, viewV)) ∗ LEN(projNormalV)   ▷ [MH99]
end for
bentNormalV ← NORMALIZE(bentNormalV)
```