Separable Approximation of Ambient Occlusion

Jing Huang[1] Tamy Boubekeur[1] Tobias Ritschel[1,2] Matthias Holländer[1] Elmar Eisemann[1]

[1]Telecom ParisTech – CNRS / LTCI [2]Intel Visual Computing Institute

Abstract

Ambient occlusion (AO) provides an effective approximation to global illumination that enjoys widespread use amongst practitioners. In this paper, we present a fast easy-to-implement separable approximation to screen space ambient occlusion. Computing occlusion first along a single direction and then transporting this occlusion into a second pass that is stochastically evaluating the final shading based on the AO estimates proves extremely efficient. Combined with interleaved sampling and geometry-aware blur, visually convincing results close to a non-separable occlusion can be obtained at much higher performance.

1. Introduction

Ambient occlusion (AO) is an effective approximation of indirect lighting that can be computed in real-time for dynamic scenes. As demonstrated in many recent algorithms, it improves the perception of volumes, concavities and contact areas of 3D objects.

AO is a real positive value, defined at every point of a surface as the proportion of occluders present in the vicinity of the point. This value is then used to replace the standard constant ambient term of popular direct shading models. Formally, the AO at a surface element **p** with normal **n** is defined as the integral over the surface-aligned hemisphere Ω:

$$AO(\mathbf{p},\mathbf{n}) = \frac{1}{\pi}\int_{\Omega} V(\mathbf{p},\omega)\,\mathbf{n}\cdot\omega d\omega,$$

with $V(\mathbf{p},\omega)$ being the *visibility term* from **p** in direction ω. Often V is coupled to a falloff function that ensures that distant occluders are ignored.

The AO integral can be evaluated using Monte Carlo integration [**?**]. Although AO is less expensive than a complete indirect lighting evaluation, it remains too expensive for interactive computation. Therefore, approximations are commonly used for real time applications. In this paper, we focus on screen space AO (SSAO) [**?**] which uses the depth buffer as an approximation of the scene, simplifying the formulation to a 2.5D filter.

In this context, fast computations for the visibility function and for the sampling pattern are two extensively studied optimizations. Observing that ambient occlusion is a form of local filter in screen space, we suggest a stochastically inspired separable computation of the 2D visibility that results in images that are similar to the non-separable version, but significantly more efficient.

2. Previous Work

Bülthoff and Langer [**?**] were the first to analyze the perception of geometry lit on a cloudy day, which was later introduced as AO to the rendering community by Landis [**?**]. As well, several approaches [**?**,**?**,**?**] successfully used the principle to depict small surface variations. The strength of the resulting shape cues lead to a high interest in AO and its efficient computation. A considerable number of methods were developed along two main branches: *geometry-based* AO and *screen space* AO. Their names do **n**ot refer to **w**here AO is computed (vertex or pixel), but to the information considered as occluders – full 3D geometry or view-dependent 2.5D Z-buffers sampling.

Geometry-based AO Bunnell [**?**] presents an AO technique to treat vertices dynamically as receivers and emitters. In a preprocess, a distance-based vertex hierarchy is created that is traversed during runtime. Each vertex gathers shadow values from all emitters in an efficient manner, but the method's performance depends on the geometric complexity.

Kontkanen and Laine [**?**] precompute AO functions in a cubemap surrounding each object. At run-time, they evaluate these AO functions to produce cast shadows between objects. Their system achieves real-time performance, but requires a long precomputation and large amounts of memory, and is also limited to scenes featuring rigid motions only. Instead of cubemaps, one can use full 3D textures [**?**].

Reinbothe et al. [**?**] address fully dynamic scenes by combining object- and image-space computations. They sample occlusion against an on-the-fly voxelization of the scene before applying a feature-sensitive filtering on the AO.

McGuire [**?**] renders occlusion volumes around polygon primitives that each initiate a fragment shader to compute the occlusion contribution of the entity. The quality is high, but performance is geometry and AO falloff dependent because it depends on the bounding-box sizes.

Screen space AO The idea of SSAO – as introduced in the game Crysis [**?**] – is that the depth buffer values around every pixel, as seen from the camera, can serve as a scene approximation in the vicinity of the pixel's world location.

© The Eurographics Association 2011.

AO is evaluated by randomly sampling 3D points around the current pixel. These are projected into the depth buffer and compared to the stored value using a binary (or linear) depth comparison to approximate a volumetric obscurance.

Another such method was presented earlier by Loos and Sloan [?] who replaced point sampling by line or area sampling to better approximate the volumetric occupation in a sphere around the current pixel. Based on an observed relation to the cosine falloff, Ruiz et al. [?] decided to integrate the free space of a tangent sphere shifted in the normal direction above the current pixel.

Shanmugam et al. [?] present two methods for high- and low-frequency occlusion (for nearby and distant surfaces respectively). They define a sphere for each surface pixel in screen space and sample from a Normal-Depth buffer around the current pixel location. These samples are used to define occluders as spherical caps that are projected in direction on the current pixel's unit hemisphere.

Horizon-based ambient occlusion (HBAO) [?] uses the surface's angle of elevation to approximate AO. They compute this angle by summing up the tangent and the horizon angle in view space for a predefined set of screen directions and choose the most representative one.

Finally, Ritschel et al. [?] extend SSAO by considering directional occlusion to simulate bounced light and propose depth-peeling to better handle occluders.

SSAO is usually faster than geometric solutions, but its performance depends on the 2D domain used to evaluate occlusion around a point. E.g., for a disc, a linear growth of the radius implies a quadratic increase in the number of samples. Large regions can be costly, independently of the applied visibility method.

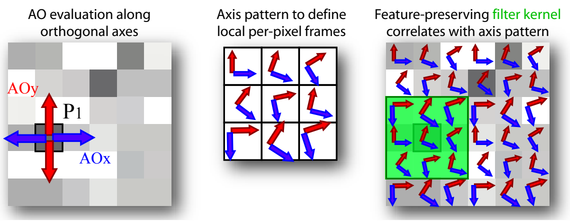

Figure 1: ***Principle:*** *our separable approximation combines two 1D evaluations in orthogonal screen directions (e.g. x,y axis, left). By changing coordinate frames per pixel, we can derive a stochastically valid approximation of the 2D occlusion by combining the result of neighboring samples.*

3. Separable Approximation of Ambient Occlusion

Consider computing AO at a pixel $\{i,j\}$ at world location $\mathbf{p}_{i,j} \in \mathbb{R}^3$ with normal $\mathbf{n}_{i,j} \in \mathcal{S}^2$. We observe that, when formulated in screen space, AO can be understood as a form of local filtering of the 2D screen buffers (depth, normals):

$$AO(i,j) = \frac{1}{k^2} \sum_{x=i-k/2}^{i+k/2} \sum_{y=j-k/2}^{j+k/2} V(\mathbf{p}_{i,j}, \omega_{x,y}) \mathbf{n}_{i,j}.\omega_{x,y},$$

where ω is the point above the pixel on the unit sphere.

As shown by Pham [?] for bilateral image filtering, even when not formally separable, local filters can be efficiently approximated in a separable fashion. In our method, we approximate AO by an x-term and a y-term. Both are evaluated in 1D only (i.e. sampling a half circle of directions).

In a first pass, we compute the occlusion $AO_{\text{X}}(i,j)$ in direction X at every pixel by fixing y in the sum to 0. In a second pass, pixels are occluded by neighbor pixels along the y-axis $AO_{\text{Y}}(i,j)$, this time fixing x to 0. Averaging these two partial AO evaluations provides a basic form of separable approximation, but ignores most of the neighboring occluders, i.e. all elements located in diagonal directions (cf. Fig 1). Such potential occlusions are crucial to take into consideration. Our goal is to exploit the visibility computation from neighboring points to derive an approximation of the actual occlusion in all directions.

Randomization Although a feature-preserving smoothing usually helps in softening SSAO artifacts [?, ?], the decision to rely on a global frame leads to visible artifacts even after filtering over several adjacent samples (cf. Fig 2). We solve this problem by replacing the global {x,y} frame by a local, randomized one: local axes are built by choosing a random direction for each pixel. In order to favor the subsequent filtering process, we use interleaved sampling [?] to create noise patterns. The size of the noise pattern are chosen to match the filter support size to ensure that artifacts are mostly removed because it implicitly correlates both processes. Relying on two 1D evaluations only, turns the original $O(k^2)$ SSAO complexity into $O(k)$.

Size	Samples / Separable	Crytek [?]	Vol. Obs. [?]	HBAO [?]
5	5×5 / no	3.2 ms	3.5 ms	3.6 ms
	5×2 / yes	3.4 ms	3.6 ms	3.5 ms
11	11×11 / no	13.9 ms	14.8 ms	15.0 ms
	11×2 / yes	5.8 ms	5.9 ms	6.0 ms
21	21×21 / no	49.7 ms	51.9 ms	51.9 ms
	21×2 / yes	9.9 ms	9.9 ms	10.2 ms

Table 1: *Performance comparison between exact and separable evaluation for various methods and filter sizes.*

4. Implementation and results

We have implemented our algorithm in OpenGL/GLSL on an NVIDIA GTX480 graphics card. To demonstrate the independence of our approach to other kinds of optimization, we used three different SSAO techniques. For each technique,

© The Eurographics Association 2011.

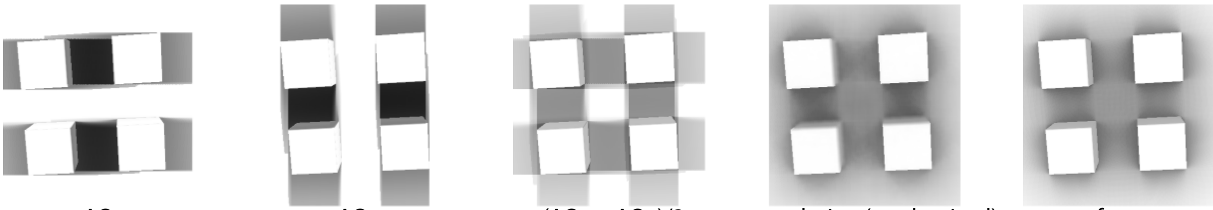

Figure 2: *Randomizing the notion of of "horizontal" and "vertical" axis on a per-pixel basis leads to a good result when compared to reference AO, while a simple combination of both axis AOs leads to visible artifacts.*

the original result is compared to its separable version. Performances are reported in Table 1 for a 1024x768 screen resolution on scenes of 50k triangles. In each case, we obtained a similar visual quality using either an exact evaluation or our separable approximation. The timings clearly show the significant acceleration for various SSAO methods, most particularly when using a large AO support radius. We also analyzed the error introduced by our method and measured perceptual differences in the Lab color space [**?**] (Fig. 4, third row) between reference AO buffers (Fig. 4, first row) and our approximated ones (Fig. 4, second row). We can see that they remain low in practice and are even hidden when other illumination effects are added to the final rendering (Fig. 4, last row). Choosing only a single direction instead of two axis of a local frame proved insufficient. The corresponding artifacts did not justify the speedup (Fig. 3).

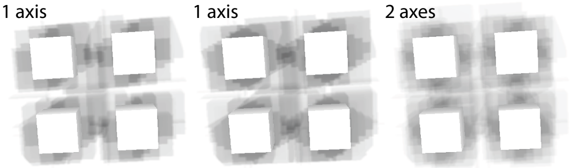

Figure 3: *One axis does not provide enough information (left), even when using the same number of samples which leads to the same cost (middle), the result is inferior to a two-axis separation (right).*

5. Conclusion

We have introduced a new approximation method for screen-space ambient occlusion by decomposing the AO evaluation into a separable transport. Our approximation shows good quality and reduces significantly the AO computation time. It can be easily integrated into existing AO methods as demonstrated by the use of three different techniques. Furthermore, it inherits all SSAO properties such as the natural view-dependent adaptivity and its ability to deal with arbitrary dynamic scenes. We believe that similar separable approximations could be useful for other screen-space or general rendering techniques and plan to investigate this possibility in the future.

© The Eurographics Association 2011.

6. Acknowlegements

This work has been partially funded by the French National Research Agency (ANR)(for the projects : CeCil and MEDI-AGPU) and the 3DLife European Network of Excellence.

References

[BG07] Bruckner S., Gröller E.: Enhancing depth-perception with flexible volumetric halos. *IEEE Trans. on Vis. and Comp. Graph. 13* (November 2007), 1344–1351.

[BL99] Bulthoff H. H., Langer M. S.: Perception of shape from shading on a cloudy day, 1999.

[BSD08] Bavoil L., Sainz M., Dimitrov R.: Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 Talks* (2008), pp. 22:1–22:1.

[Bun05] Bunnell M.: Dynamic ambient occlusion and indirect lighting. *GPU Gems 2* (2005), 223–233.

[KH01] Keller A., Heidrich W.: Interleaved sampling. In *Proc. EGWR '01* (2001), pp. 269–276.

[KL05] Kontkanen J., Laine S.: Ambient occlusion fields. In *Proc. ACM I3D '05* (2005), pp. 41–48.

[Lan02] Landis H.: Production-ready global illumination, 2002.

[LCD06] Luft T., Colditz C., Deussen O.: Image enhancement by unsharp masking the depth buffer. In *ACM SIGGRAPH* (2006), pp. 1206–1213.

[LS10] Loos B. J., Sloan P.-P.: Volumetric obscurance. In *Proc. ACM I3D '10* (2010), pp. 151–156.

[McG10] McGuire M.: Ambient occlusion volumes. In *Proc. ACM HPG '10* (June 2010).

[Mil94] Miller G.: Efficient algorithms for local and global accessibility shading. In *SIGGRAPH* (1994), pp. 319–326.

[Mit07] Mittring M.: Finding next gen: CryEngine 2. In *ACM SIGGRAPH '07 Courses* (2007), pp. 97–121.

[MMAH07] Malmer M., Malmer F., Assarsson U., Holzschuch N.: Fast precomputed ambient occlusion for proximity shadows. *Journal of Graphics Tools 12*, 2 (2007), 59–71.

[PV05] Pham T. Q., Vliet L. J.: Separable bilateral filtering for fast video preprocessing. In *Proc. IEEE Int. Conf. on Multimedia and Expo* (2005), pp. 1–4.

[RBA09] Reinbothe C., Boubekeur T., Alexa M.: Hybrid ambient occlusion. *EUROGRAPHICS '09 Areas Papers* (2009).

[RGS09] Ritschel T., Grosch T., Seidel H.-P.: Approximating dynamic global illumination in image space. In *Proc. ACM I3D '09* (2009), pp. 75–82.

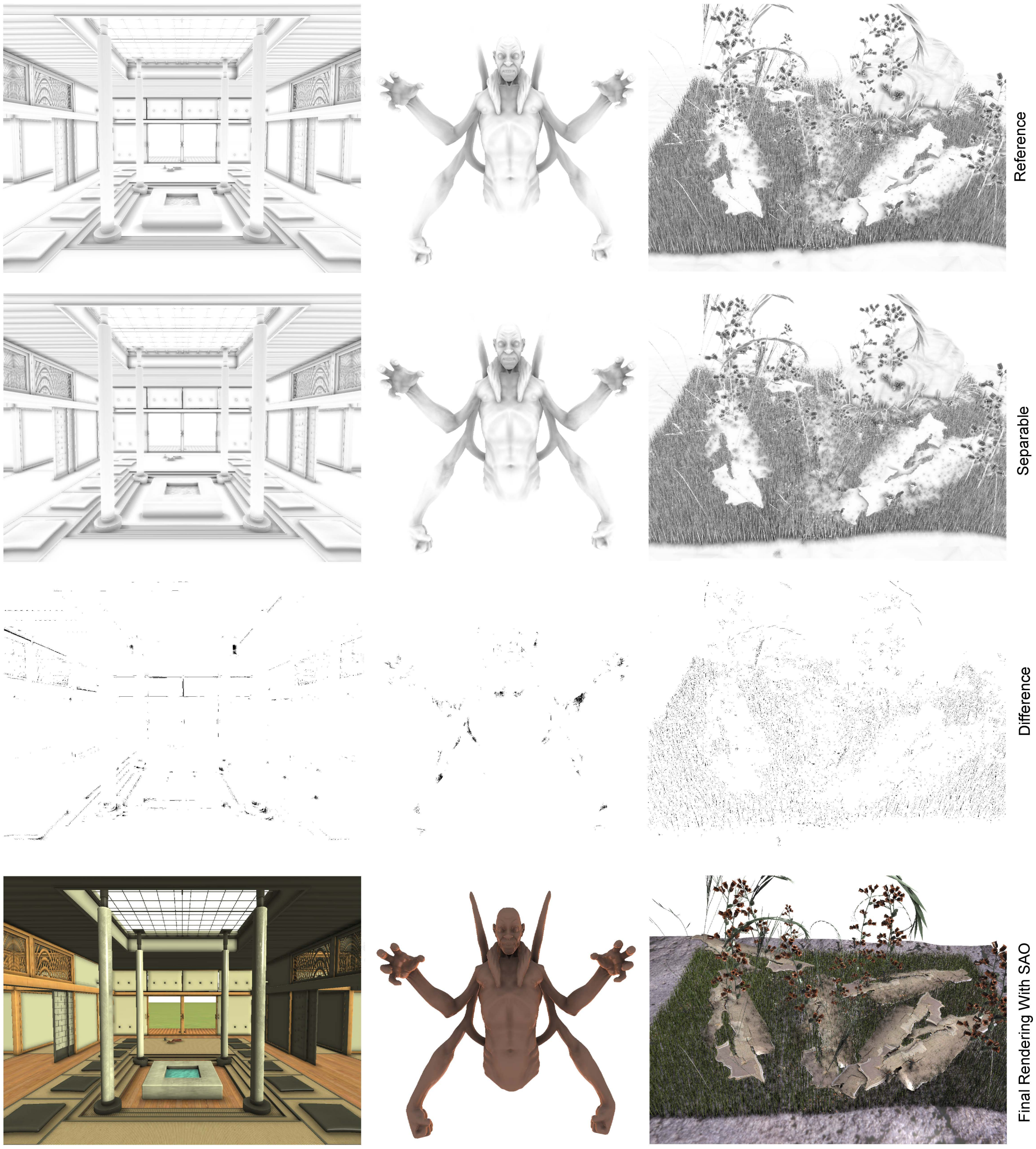

Figure 4: *Comparison between reference (first row) and separable AO evaluation (second row): for the perceptual difference images (third row), white values stand for no difference while black values indicate visible shifts.*

[RSKU*10] RUIZ M., SZIRMAY-KALOS L., UMENHOFFER T., BOADA I., FEIXAS M., SBERT M.: Volumetric ambient occlusion for volumetric models. *Vis. Comput. 26* (June 2010), 687–695.

[SA07] SHANMUGAM P., ARIKAN O.: Hardware accelerated ambient occlusion techniques on GPUs. In *Proc. ACM I3D '07* (2007), pp. 73–80.

[Yee04] YEE H.: A perceptual metric for production testing. *Journal of Graphics Tools 9*, 4 (2004).

© The Eurographics Association 2011.