

Fast Image-Based Ambient Occlusion IBAO



Robert Sajko and Zeljka Mihajlovic

University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Electronics, Microelectronics, Computer and Intelligent Systems

Abstract— The quality of computer rendering and perception of realism greatly depend on the shading method used to implement the interaction of light with the surfaces of objects in a scene. Ambient occlusion (AO) enhances the realistic impression of rendered objects and scenes. Properties that make Screen Space Ambient Occlusion (SSAO) interesting for real-time graphics are scene complexity independence, and support for fully dynamic scenes. However, there are also important issues with current approaches: poor texture cache use, introduction of noise, and performance swings.

In this paper, a straightforward solution is presented. Instead of a traditional, geometry-based sampling method, a novel, image-based sampling method is developed, coupled with a revised heuristic function for computing occlusion. Proposed algorithm harnessing GPU power improves texture cache use and reduces aliasing artifacts. Two implementations are developed, traditional and novel, and their comparison reveals improved performance and quality of the proposed algorithm.

Index Terms— Shading, Ambient occlusion, Screen space ambient occlusion, GPU accelerated techniques

I. INTRODUCTION

To simulate interaction of light with physical surfaces, all aspects of light behavior should be considered including aspects of geometrical, physical and quantum optics. Real-world surfaces are transparent or semi-transparent, and anisotropically reflective. Including of all these aspects in one, single model that would simulate interaction of light with surfaces of objects for purposes of computer graphics is enormously demanding. On the other hand, renouncing some aspects of the nature of light leads to a lack of realism in rendered scenes. Hence, we are aware that some compromises have to be made, and that the most influencing components on the perception of realism of displayed environments must be determined.

Full global illumination models certainly give the most visual advantages but are time consuming and inappropriate for real-time applications. On the other side, local models are fast but omit many effects and produce poor results. The compromise is in the inclusion of at least some influence of global effects in the illumination model at an acceptable price.

Ambient lighting is an approximation of the light reflected from other objects in the scene. Its presence reveals the spatial relationship between objects, their shape, depth and surface complexity details. Ambient lighting can be locally occluded by nearby object or a fold in the surface. Ambient occlusion produces only subtle visual cues, however, they are very important in natural perception and thus also in a convincing, realistic lighting model.

Modern consumer GPUs offer impressive computational power which has allowed the use of various techniques and algorithms in real-time computer graphics that were previously possible in offline rendering only. One such technique is ambient occlusion, which approximates soft shadows due to indirect (ambient) lighting. Therefore, it can be used as part of a larger global illumination solution, or as a stand-alone effect. The form of the technique most viable for use in real-time graphics is based in screen space, which means it is essentially an image-based effect, allowing for a simple and efficient implementation in a pixel shader. The main benefit of this approach is independence from scene geometry, allowing for complex and dynamic scenes with no precomputation required. This paper proposes a new approach to Screen Space Ambient Occlusion (SSAO). We named the proposed approach Image-based ambient occlusion (IBAO). In the proposed approach significant quality and performance improvements are achieved through improved texture cache utilization and the removal of noise generation, thus also removing the need for additional noise filtering.

This paper is structured as follows: first, in Section II we discuss related work. In Section III we describe overview of AO concept. Section IV explains traditional approach to SSAO, and in Section V novel approach to SSAO is presented. We present our results in Section IV, and Section VII contains the concluding remarks.

II. PREVIOUS STUDIES

The idea for ambient occlusion was devised in a context where virtual objects were combined into a real scene proposed by Landis [4]. The surrounding environment of a real scene was captured using a chrome sphere and unwrapped into a spherical map. The resulting environment map was also used as a representation of incoming light in the location it was taken at. Nevertheless, such light maps are static and do not support dynamic scenes with variable lighting conditions. An early but

interesting approach to the dynamic ambient occlusion and indirect lighting is given in Bunnell et al. [1], but it was still geometry based and thus dependent on scene complexity.

The original idea of implementing dynamic ambient occlusion in screen space was presented by Mitrting [6], followed by proposed ideas such as Fillion and McNaughton [3] and Loos and Sloan [5]. The basic, common idea presented in these papers is based on the observation that the depth buffer can be used to approximate the visibility function needed to compute the ambient occlusion term. In a common rendering pipeline, a depth buffer will usually be available after the initial geometry-rendering pass. The idea is that the visibility function, which requires computing intersections of rays with scene geometry, is replaced by a heuristic function, which samples the conveniently available depth buffer. As a result, the algorithm now operates in screen space, and is essentially an image filter, no longer dependent on actual scene geometry.

The obvious advantages of decoupling ambient occlusion computation from geometry processing are the simplicity of implementation, as a post-processing effect, and support for fully dynamic scenes with arbitrary geometric complexity. However, there are some drawbacks as well. Geometry that is currently not inside the viewport is ignored, even though it may still affect the visible scene. Additionally, back-facing polygons are inaccessible to the algorithm as well.

A solution that makes compromise between benefits of decoupling ambient occlusion computation from geometry processing and drawbacks in that approach is described by research group Reinbothe, Boubekeur and Alexa [7], in the form of a hybrid approach that resolves these issues by performing a separate rasterization of a 3D region of the scene. Additionally, Ritschel, Grosch and Seidel [8] include indirect bounce of light to improve perception of nearby colored surfaces and name it directional occlusion (DO). They also propose depth peeling and additional cameras to improve rendering of contact shadows or influence of surfaces that are not visible for some camera angle in screen space to visible surfaces. Even so, a simple, purely image-based approach still has the best performance potential and unmatched implementation simplicity. Even with the mentioned restrictions, image-based approach is able to provide a sufficient approximation of ambient occlusion for real-time use.

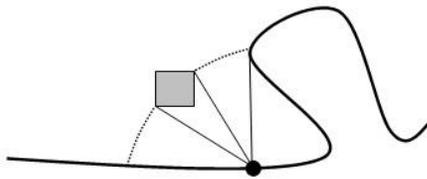


Fig. 1. Illustration of ambient occlusion. Ambient lighting at point P can be locally occluded by another nearby object or fold in the surface.

III. OVERVIEW OF AMBIENT OCCLUSION CONCEPT

The concept of ambient occlusion is simple to understand, but the technique can become complex to implement. Indirect

diffuse lighting is propagated from every point on the surface of an object, equally in every direction, thus forming a hemisphere above the surface. Therefore, the ambient occlusion for any given point on the surface of an object is simply the percentage of the hemisphere covered by another object, as shown in Fig. 1

More formally, ambient occlusion A_p can be defined as the integral of the visibility function $V_{p,\omega}$ over the hemisphere Ω :

$$A_p = \frac{1}{\pi} \int_{\Omega} V_{p,\omega} (\mathbf{N} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega}, \quad (1)$$

where \mathbf{P} is any given point on the surface, \mathbf{N} is the surface normal at point \mathbf{P} , and $\boldsymbol{\omega}$ is the angle of integration. The visibility function can be determined by constructing a ray that originates at point \mathbf{P} at angle $\boldsymbol{\omega}$. If the ray intersects any other object, visibility function equals zero; otherwise, it equals one. Beside binary classification of visibility, visibility function could also be used as continuously defined function. In that case distance to nearby surface is used to smoothly and monotonically change visibility function over the hemisphere Ω for each nearby object.

Expression (1) can be approximated by replacing the integration with a summation:

$$A_p \approx \frac{1}{C} \sum_{\Omega} V_{p,\omega} (\mathbf{N} \cdot \Delta\boldsymbol{\omega}) \quad (2)$$

This step allows sampling of the visibility function in some appropriate fashion, while adding the contribution of each sample of differential solid angle $\Delta\omega$ to the final solution. A direct, elegant solution of this problem is ray-tracing, which is often used to produce 'ground-truth' ambient occlusion. Nevertheless, an efficient GPU implementation of ambient occlusion requires a different approach.

IV. TRADITIONAL APPROACH TO SCREEN SPACE AMBIENT OCCLUSION

Existing approaches to the Screen space ambient occlusion differ in the actual occlusion function used to evaluate depth buffer samples, and in the fashion in which these samples are gathered. A generic SSAO algorithm can be defined as follows:

- Reconstruct the 3D position of the current pixel (in camera space).
- Choose a number of samples in 3D space, within the pixel's hemisphere.
- Project the samples back into screen space.
- Determine the depth of each sample, by reading the depth buffer.
- Evaluate the occlusion function for each sample and accumulate the individual contributions.

The occlusion function can be a simple, arbitrary 1D function that takes into account the difference between the current pixel's depth and the sampled depth. An important issue with current implementations is strong aliasing artifacts that occur due to the undersampling, as shown in Fig. 3. (top image).

This problem can be solved by randomized sampling and a good approach is proposed by Fillion and McNaughton [3]. Unfortunately, this approach introduces high-frequency noise into the resulting image. Therefore, an additional noise-reduction pass is required. Noise-reduction pass can be achieved by a bilateral filter, or custom smart blur filters, which do not blur across object edges as presented in Fig. 3. (bottom image).

Another issue is performance swings that can occur as the camera moves through the scene. The problem lies in the fact that the samples are selected in 3D (camera) space, in the vicinity of the reconstructed surface point. Therefore, this is essentially geometry-based sampling, albeit performed from rasterized data. Even though the sampling area that is actually the pixel's hemisphere is of constant size in camera space, its projection into screen space (the actual depth buffer sampling region) is of variable size, depending on the depth of the current pixel. Variations of the projection size cause performance swings which is very unpleasant effect.

We can notice that, for example, Bavoil, Sainiz and Dimitrov [1] project the eye space radius R of influence on the image plane. Faraway pixels thereby result in a tiny sampling area, while close ones require a much larger one (Fig. 2). Random sampling further exacerbates the problem, and as a result, the texture cache is often very poorly utilized, especially for close-up objects, leading to degraded performance.

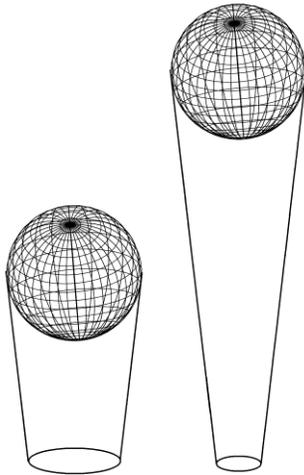


Fig. 2. Projected sampling region.

V. NOVEL APPROACH TO SSAO

In this paper we propose novel approach to the Screen space ambient occlusion that we named Image based ambient occlusion. It can be observed, that if the samples were selected directly from a two dimensional sampling region rather than in three dimensional space, we would have far greater control over the size and shape of the sampling region. The pixel neighborhood in 3D space is not cache coherent for texture access, but sampling the neighborhood in 2D space is cache coherent. The region's radius may remain depth-dependent but

with an easily set upper bound, to ensure an acceptable level of texture cache misses.

However, by selecting samples in a 2D area rather than 3D space, the implicit information about the samples' 3D position is lost, and the simple 1D occlusion function that takes into account depth only is no longer a sufficient heuristic function. Instead, it is necessary to define the occlusion function as a three dimensional function that takes into account the actual (reconstructed) 3D positions of the current pixel and a selected sample.

In evaluation of 3D occlusion function for each vertex, surface elements (surfels) are precomputed. Surface elements are oriented disks with a position, normal, and area. Occlusion of 3D positions of the current pixel is calculated using disk-based form factors as it is proposed in Bunnell [2].

In essence, we abandon the traditional geometry-based sampling approach used in SSAO, and develop a purely image-based sampling method, that is better suited to modern GPU architecture. Therefore, the novel algorithm may be referred to as Image Based Ambient Occlusion (IBAO). The algorithm can now be defined as follows:

Sample the depth buffer in a 2D region around the current pixel.

Reconstruct the 3D positions of the current pixel and gathered samples in camera space.

Evaluate the occlusion function for each sample and accumulate the individual contributions.

Random sampling may still be applied to alleviate aliasing artifacts, nevertheless, this is no longer necessary. Due to the change in nature of the sampling process using a 2D sampling area, undersampling now produces weak banding artifacts, as shown in Fig. 4. Constraining the sample space to a narrow 2D region changes the look of the effect into a more local ambient occlusion, causing a loss of the soft shading across surfaces. However, this also considerably improves performance due to better texture cache use and limits artifacts, while still providing an aesthetically pleasing and convincing effect.

Furthermore, due to using a constrained 2D sampling region, performance swings are no longer an issue. As an additional benefit, aliasing artifacts are weak enough not to require randomized sampling, ensuring good utilization of the texture cache. This also results in the absence of noise, which removes the need for an additional noise-filtering step, improving performance even further.

VI. RESULTS

Performance comparison between the traditional SSAO and novel IBAO algorithm is given in Fig. 5. The chart displays average processing time per frame in milliseconds, depending on the number of samples used. Measures are provided on ATI Radeon HD3870, in 1280x720 resolution. From the presented chart we can notice that IBAO is approximately twice as fast as SSAO. With all other parameters equal, our approach displays a substantial performance improvement in the test scene, and reduced aliasing artifacts are obvious from the presented result.

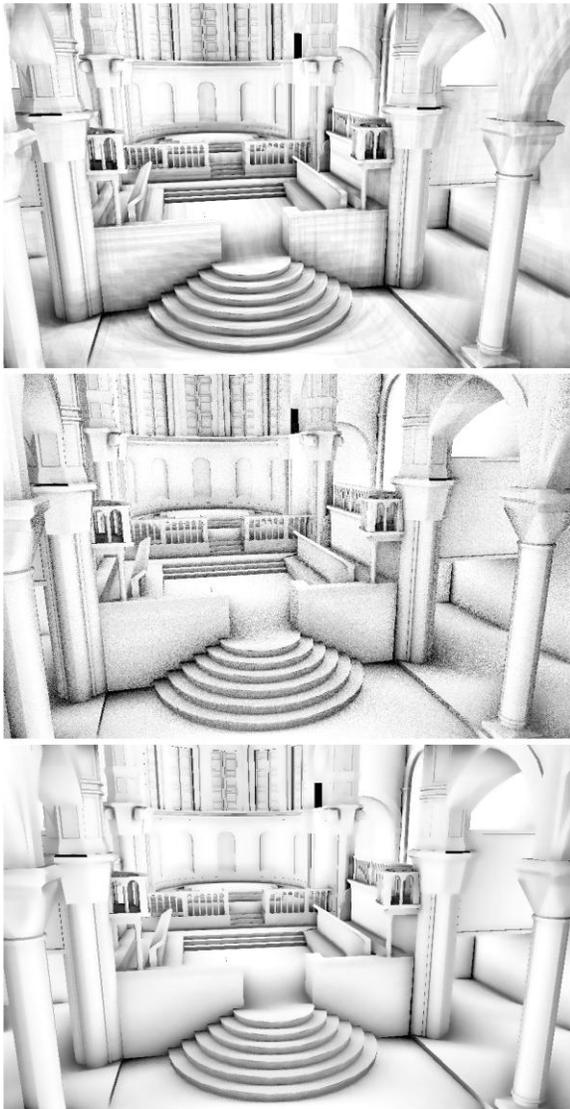


Fig. 3. The top image shows an example of aliasing artifacts appearing in sequential sampling. The middle image shows the result of randomized sampling. The bottom image is the final result after applying bilateral filtering.

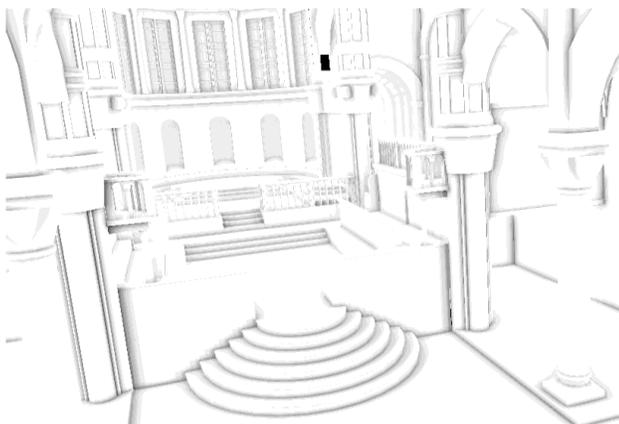


Fig. 4. Results of the proposed IBAO algorithm

Performance of both implementations can be improved. For instance, improvement could be achieved by using a down-sampled depth buffer or other means to improve texture

cache use, as proposed by Fillion and McNaughton [3], or by using a variable (depth-dependent) number of samples. However, these reference implementations are using full-resolution buffers, with no additional optimizations.

Perceived locality of the final occlusion effect may be regarded as an issue. Nevertheless, a dual sampling approach, such as proposed by Shanmugam and Arikan [9] is still applicable to the novel algorithm. In short, two different sampling regions may be defined: a narrow one, and a wide one. The algorithm is executed once for each region, and results are combined to form the final solution. In effect, both near and distant occluders are taken into account. Nevertheless, if a simple, fast and convincing approximation of ambient occlusion is sufficient, wide sampling may not be cost-effective.

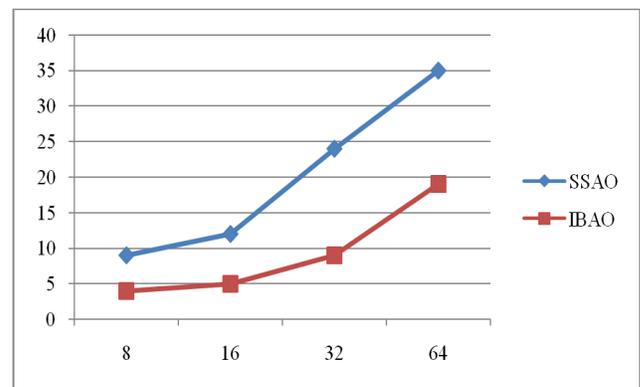


Fig.5. Performance comparison between the traditional SSAO and novel IBAO algorithm. Average processing time per frame in milliseconds depending on the number of used samples is presented.

VII. CONCLUSION

We have discussed existing SSAO implementations and constructed a generic algorithm based on the commonly used approach. Properties that make SSAO interesting for real-time graphics are scene complexity independence and support for fully dynamic scenes. But, there are also important issues with current implementations such as poor texture cache use, introduction of noise and performance swings. The observation is that these issues are caused by a reconstructive geometry-based sampling method used in traditional SSAO approaches.

A simple and straightforward solution is presented in the form of a novel, purely image-based sampling method. The main improvements of IBAO are in switching from 3D to 2D sampling space and using 3D instead of a 1D occlusion function. The main advantages are a better control of sampling region's size and therefore better usage of texture caches and reduced aliasing artifacts with stable performances. This alternative sampling pattern improves cache efficiency since sample points are locally clustered in image-space. In addition, banding artifact is less visually distracting since projection from image-space to object-space naturally breaks banding boundaries.

Reduced aliasing effects make additional noise reduction passes and efforts for randomized sampling unnecessary. While retaining the same basic idea, all the good properties of the original algorithm are preserved. While all the main issues are resolved, offering improvements in both quality and performance is achieved. Proposed IBAO is appropriate for complex, fully dynamic scenes that simultaneously achieve stable, greatly improved performances and reduced aliasing artifacts.

Computer and Intelligent Systems, of the University of Zagreb, Croatia. Her current research interests include computer graphics and visualization algorithms, reconstruction techniques as well as e-learning technologies.

ACKNOWLEDGEMENT

The test scene used is Sibenik Cathedral, courtesy of Marko Dabrovic. This work has been carried out within the project 036-0362980-1921 Computing Environments for Ubiquitous Distributed Systems funded by the Ministry of Science, Education and Sport of the Republic of Croatia.

REFERENCES

- [1] L. Bavoil, M. Sainz and R. Dimitrov "Image-space horizon-based ambient occlusion," In *ACM Siggraph 2008 talks*, Article 22, 2008, DOI=10.1145/1401032.1401061
- [2] M. Bunnell, "Dynamic Ambient Occlusion and Indirect Lighting," In *GPU Gems 2*, M. Pharr, Ed. Addison Wesley, Ch. 2, 2005, pp. 223-233.
- [3] D. Fillion, R. McNaughton, "Starcraft ii: Effects and techniques," "Advances in Real-Time Rendering in 3D Graphics and Games, In *ACM Siggraph 2008 classes*, 2008, pp. 133-164. DOI=10.1145/1404435.1404441.
- [4] H. Landis, "Production-ready global illumination", *Siggraph Course Note 16: RenderMan in Production (29)*, 2002, pp. 87-102.
- [5] B.J. Loos, P.P. Sloan, "Volumetric obscurance," *Proceedings of the ACM Siggraph Symposium on Interactive 3D Graphics and Games I3D 10*, 151, 2010, doi=10.1145/1730804.1730829.
- [6] M. Mittring, "Finding next gen - cryengine 2," *Advanced Real-Time Rendering in 3D Graphics and Games*, 2007, pp. 97-121. DOI=10.1145/1281500.1281671.
- [7] C. Reinbothe, T. Boubekeur, M. Alexa, "Hybrid ambient occlusion", *Eurographics 2009 Areas Papers*, 2009, pp. 51-57.
- [8] T. Ritschel, T. Grosch, H. Seidel, "Approximating dynamic global illumination in image space", In *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, 2009, pp. 75-82.
- [9] P. Shanmugam, O. Arikani, "Hardware accelerated ambient occlusion techniques on GPUs," In: *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, 2007, pp. 73-80.



Robert Sajko received Bachelor Degree in 2008 in Computing from the University of Zagreb and he received Masters Degree in Computer Science in 2010 from the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia. His passion is computer graphic and his reach interest includes real-time rendering techniques especially for game design and image-based special effects on GPUs.



Željka Mihajlović received the B.Sc. degree in Electrical Engineering in 1988, the M.Sc. and Ph.D. degrees in Computer Science in 1993 and 1998 respectively, from the Faculty of Electrical Engineering and Computing, University of Zagreb. She is currently an Associate Professor at the Department of Electronics, Microelectronics,