

Multi-view Ambient Occlusion with Importance Sampling

Kostas Vardis* Georgios Papaioannou† Athanasios Gaitatzes‡
Department of Informatics, Athens University of Economics & Business

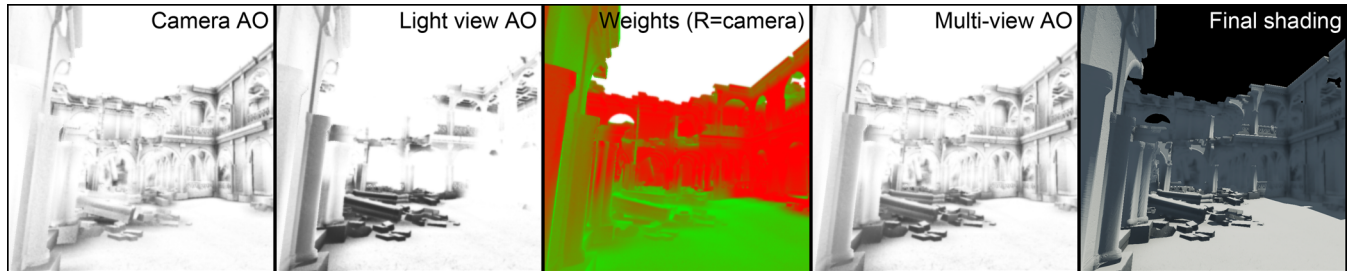


Figure 1: Multi-view ambient occlusion combines depth information from any available views to fill in missing occlusion.

Abstract

Screen-space ambient occlusion and obscurance (AO) techniques have become de-facto methods for ambient light attenuation and contact shadows in real-time rendering. Although extensive research has been conducted to improve the quality and performance of AO techniques, view-dependent artifacts remain a major issue. This paper introduces Multi-view Ambient Occlusion, a generic per-fragment view weighting scheme for evaluating screen-space occlusion or obscurance using multiple, arbitrary views, such as the readily available shadow maps. Additionally, it exploits the resulting weights to perform adaptive sampling, based on the importance of each view to reduce the total number of samples, while maintaining the image quality. Multi-view Ambient Occlusion improves and stabilizes the screen-space AO estimation without overestimating the results and can be combined with a variety of existing screen-space AO techniques. We demonstrate the results of our sampling method with both open volume- and solid angle-based AO algorithms.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Shading, shadowing

Keywords: ambient occlusion, ambient obscurance, screen space, real-time rendering

1 Introduction

Over the last few years, graphics processing power has been increasing significantly and, as a result, demands in realism in real-time applications have been increasing as well. This requires developers to implement and integrate efficient and highly realistic

rendering algorithms into their applications. One of the most popular techniques is ambient occlusion [Akenine-Möller et al. 2008]. Ambient occlusion is a non-physically based method, which tries to approximate the amount of indirect light that reaches a point, based on its surrounding occluders, without taking into account inter-reflections. Hence, the incident radiance from blocked directions is zero. This produces darker results than a full indirect illumination simulation, yet visually convincing, especially for distant lighting such as environment maps.

Two main approaches are taken when approximating ambient occlusion and obscurance of complex, dynamic scenes; object-space and image-based methods. Object-space methods produce stable, high-quality results, but the proposed techniques are not suited for applications with limited per-frame time budget for global illumination effects. On the other hand, image-based methods introduce a trade-off between quality and performance. They produce reasonably convincing results and offer bounded rendering times since they do not depend on scene complexity, since they make use of information already stored in the camera G-buffer. Image-based methods have been an active area of research over the last few years and a number of different techniques have been proposed, some of which are discussed in the next section.

In general, image-space techniques use buffers from available views, such as the camera G-buffer or shadow maps to obtain information related to a world-space point and its surrounding geometry. Typically, samples in the vicinity of a point are projected onto the buffers and the resulting image-space locations derived through the view's depth map constitute an approximate reconstruction of the geometry at it. The recovered information, is used in calculations such as ambient occlusion, obscurance and indirect lighting.

Screen-space ambient occlusion techniques, focus on the use of user-centered information, i.e. from the G-buffer of the camera viewpoint. Despite their stable and fast performance, all screen-space methods suffer from view-dependent artifacts, mainly caused by the absence of occluders, unseen in the image buffers of the current view (inside and outside the view frustum). These artifacts are manifested in the form of shadows that pop up as the camera moves and haloing effects.

Similar in spirit to previous techniques that exploit image-space data from fixed configurations of multiple camera views (e.g. [Bavoil and Sainz 2009]) to improve visual stability of the rendered image, this paper proposes a generic method to address the view-

*e-mail:kvardis@aub.gr

†e-mail:gepap@aub.gr

‡e-mail:gaitat@yahoo.com

dependent inconsistencies of the screen-space ambient occlusion algorithms by taking advantage of buffers containing geometric information from other view points, already generated as part of the rendering process. Furthermore, our method introduces an importance sampling scheme for effectively fusing ambient occlusion from arbitrary viewpoints, without overestimating the result or requiring special passes such as depth peeling. The method is not limited to a particular image-based ambient occlusion algorithm and is thus orthogonal to previous work. The proposed work maintains the advantages of a screen-space technique, while significantly reducing view-dependent artifacts, in general.

2 Related Work

Ambient occlusion and obscurity. Zhukov et al. introduced *ambient obscurity* (AO) as an empirical model to account for a better approximation to ambient lighting [Zhukov et al. 1998]. Their method simulated the indirect diffuse illumination from distant uniform lighting by calculating how much a point is obscured by neighboring geometry. Unlike global illumination methods, which simulate the light transport in a scene using approximate solutions of the rendering equation, ambient obscurity calculates a local light attenuation factor based on the “openness” of the geometry above an arbitrary surface point. It also takes into account the probability of the light not being blocked but rather reflected on nearby geometry towards our point of interest, by introducing an empirical distance-based fall-off function for the occlusion:

$$A(\mathbf{p}) = \frac{1}{\pi} \int_{\Omega} \rho(d(\mathbf{p}, \omega)) \cos(\theta_i) d\omega \quad (1)$$

where Ω is the hemisphere centered at the normal vector of the receiving point \mathbf{p} . $\rho(d(\mathbf{p}, \omega))$ is an obscurity attenuation function of distance $d(\mathbf{p}, \omega)$ of the closest point to \mathbf{p} in the incident direction $\omega = (\theta_i, \phi_i)$. When only visibility of the distant environment is considered, the distance attenuation function is replaced by a binary visibility term and ambient obscurity becomes *ambient occlusion*.

Screen-space AO. Next we review some of the recent screen-space ambient occlusion techniques, since object-space ones are not relevant to our work.

Luft et al. used a high pass filter (unsharp mask) over the depth buffer to increase visual contrast and simulate ambient occlusion in screen-space [Luft et al. 2006]. Their method is fast but only resembles the intended effect.

Screen-Space Ambient Occlusion (SSAO) was introduced in [Mitting 2007], where point samples are chosen from within a sphere centered at the shaded fragment and tested against the depth buffer. The ambient occlusion approximation is calculated using the weighted result of the depth tests to estimate the open hemispherical volume above the shaded point. Loos and Sloan proposed Volumetric Obscurity [Loos and Sloan 2010], which improved the convergence of the open volume estimator of the SSAO technique. Volumetric Obscurity measures the exact distance between the hemisphere and the actual surface at a number of samples uniformly selected on a disk around the shaded point.

Bavoil et al. proposed Horizon-Based Ambient Occlusion (HBAO [Bavoil et al. 2008]), which is based on the concept of horizon mapping [Max 1986]. They perform a circular sweep of the image space near a shaded point, by ray marching along the radial directions up to an r_{max} range. Each sample is tested against the depth buffer in order to find the highest elevation (horizon) in the point’s tangent space in every search direction. The final set of horizon angles is then used to estimate the solid angle subtended by the horizon, which is considered to contribute to ambient occlusion.

In [McGuire et al. 2011], the Alchemy Ambient Obscurity was presented, which combines the sampling scheme of Volumetric Obscurity with a statistical horizon-based approach to the open solid angle estimation; samples in an image-space disc above the shaded point are projected to the depth buffer and a horizon is determined directly for each one of them according to their apparent elevation, thus dispensing with the radial samples of the HBAO algorithm. Finally, a user-adjustable attenuation factor based on the projected sample distance from the shaded point similar to [Filion and McNaughton 2008] is applied to each partial obscurity estimate.

Multi-image approaches. [Bavoil and Sainz 2009] use several depth-peeled screen-space layers generated with an enlarged field of view for accessing visibility information behind the nearest depth layer and outside the screen bounds. They calculate AO for each depth layer and keep the highest occlusion. They also propose a dual resolution adaptive sampling approach to increase performance. The use of depth peeling however, still misses geometric information due to polygons that remain parallel to the view direction, requires additional passes to generate the required visibility information and can overestimate occlusion due to the max operator.

Finally, the screen-space near-field diffuse global illumination technique proposed in [Ritschel et al. 2009] uses multiple views to account for the missing geometric information in the camera frame buffer. They use a fixed configuration of additional depth-peeled views relative to the user’s camera target position and render the scene from these view points. However, their method requires several additional passes and does not guarantee that important direct lighting is present in any of the generated views, especially in environments with high occlusion.

Our contribution. We build on this concept of multiple view-points for screen-space visibility sampling, but propose a weighting scheme for combining arbitrary views, such as shadow maps already generated in the rendering pipeline, so that a multi-view screen-space technique can be applicable to environments with high occlusion and geometric complexity. The exploitation of depth information from existing views translates to no additional computation cost or extra passes. The weighting scheme balances the contribution of each view, without overestimating occlusion and without the need for special view setup. Furthermore, we show how to utilize the view weighting function in order to perform importance sampling and an adaptive sample reduction, thus significantly reducing the rendering time, while maintaining high quality. Multi-view occlusion can yield better results than simple depth peeling but can also include importance-sampled depth-peeled views as a special case. Finally, user-defined (phantom) views like the ones used in [Ritschel et al. 2009] can be exploited and seamlessly blended according to their importance. We propose such a camera configuration to address view-dependent artifacts in hard cases, such as confined, indoor scenes with high occlusion. We apply our multi-view method to ambient occlusion/obscurity calculation based on the Alchemy AO, SSAO and HBAO algorithms, but the method can be applied to other similar techniques.

3 Method Overview

As screen-space AO techniques evolved, they attempted to improve mostly issues related to sampling quality, memory access efficiency and faster and more accurate convergence of the occlusion integral estimator. A prominent example of this constant refinement is the Scalable AO by McGuire [McGuire et al. 2012]. However the “trademark” problems of screen-space methods, i.e. view-dependent occlusion fluctuations and erroneous shadowing still remain (see for example Figure 2). View dependency commonly

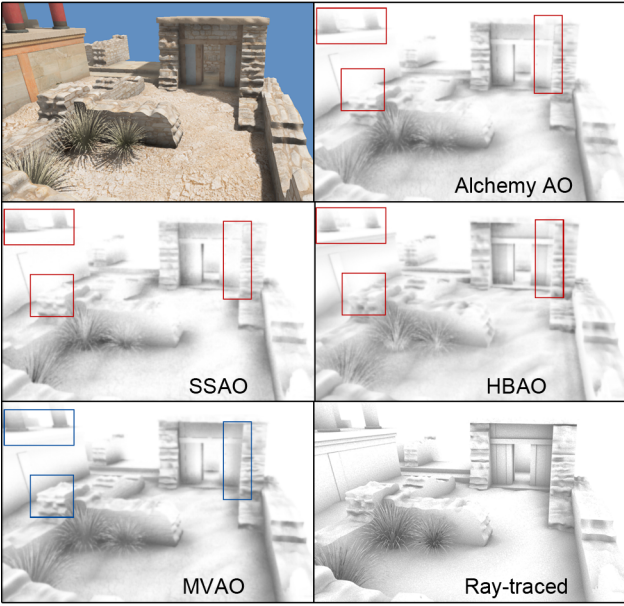


Figure 2: View dependencies of screen-space AO algorithms. Marked regions indicate erroneous occlusion due to parallax.

manifests itself either as a gradual darkening at the edges of the frame buffer, as new depth values start contributing to the occlusion estimation, or as erroneous occlusion due to lack of geometric information in the depth buffer. The first problem is remedied either by fading in the occlusion at the image boundaries or by virtually extending the image buffer beyond the viewport. The second problem, i.e. the lack of depth information within the image, is the root of severe artifacts in many cases, where layered geometry parallax with respect to the camera view point causes depth discontinuities and makes samples fall on different surfaces than the shaded point lies on. If these samples are not culled or attenuated (usually with respect to the maximum AO range r_{max}), they cause a darkening halo at the depth discontinuities. If on the other hand they are rejected, the result is an unnatural, view-dependent brightening of normally shadowed crevices. The latter is also demonstrated in Figure 2.

Multi-view Ambient Occlusion (MVAO) takes advantage of image buffers from secondary views, such as shadow maps, which are already available at AO calculation time, in order to improve and stabilize the latter, when screen-space techniques are used. For each fragment to be shaded in the camera frame buffer, the occlusion or obscurity is estimated using any number of available views and the final AO is the combination of the partial results. The (partial) occlusion in each one of the image buffers is calculated using a standard AO algorithm, such as the Alchemy AO method or SSAO. The combined occlusion however, is not the average or maximum occlusion of all partial results, but rather a weighted sum of them, since we need to favor views with better visibility of both the shaded point and the samples drawn in its neighborhood and subsequently projected in the corresponding depth buffer. For instance, in Figure 2, AO samples for a point on geometry visible just above the broken wall in the foreground are more reliably projected in the shadow map of the overhead light in the scene rather than in the camera view. Since we allow multiple arbitrary views, the shaded points and near-field samples may be occluded in any other view but the camera buffer, and therefore the importance of these views must be adjusted per fragment accordingly. The weighting functions for combining the partial AO results from the multiple views are discussed below in Section 3.2.

3.1 The Algorithm

Given a number of n views we calculate the occlusion for a point \mathbf{p} as:

$$O(\mathbf{p}) = \frac{\sum_{v=1}^n O_v(\mathbf{p})w(v, \mathbf{p})}{\sum_{v=1}^n w(v, \mathbf{p})} \quad (2)$$

where v is the current view, O_v is the occlusion for view v and $w(v, \mathbf{p})$ is a weighting function describing the contribution of view v to the occlusion at \mathbf{p} .

3.2 View Weighting Function

Any screen-space sampling implementation assumes that the point of interest \mathbf{p} is always visible in the image buffer and that the same is true for the samples taken within its near field. Additionally, the eye-space normals at all sampling locations, including the central point \mathbf{p} , always point in the direction of the center of projection. However, when projecting a camera buffer point to an arbitrary view, any of the above assumptions may not hold. Furthermore, even if \mathbf{p} and the projected samples in view v comply with the above conditions, the view-dependent projection of the samples, especially at oblique angles, makes certain view points unreliable for a given point \mathbf{p} .

The importance of a particular view v in the calculation of the occlusion $O(\mathbf{p})$ at any point \mathbf{p} visible in the camera frame buffer depends on three factors: the visibility of \mathbf{p} in view v , the orientation of the surface at \mathbf{p} with respect to the view direction and finally, the proximity of the projected samples on the v -th depth buffer to \mathbf{p} . These three factors affect the overall contribution of the partial result $O_v(\mathbf{p})$ to the final occlusion in the form of a weighting function $w(v, \mathbf{p})$:

$$w(v, \mathbf{p}) = bw_d(v, \mathbf{p}) + (1 - b)w_n(v, \mathbf{p}) \quad (3)$$

where $w_d(v, \mathbf{p})$ is the sample distance weighting function, $w_n(v, \mathbf{p})$ is the directional weighting function and b is a blending coefficient to bias the view weight in favor of the distance or the orientation importance weight. The visibility indirectly affects the weight through the distance weight. Directly accounting for the visibility of a point in a given view would cause discontinuities in the weighting function, resulting in visible banding, for instance at shadow boundaries, if light views are used. The blending coefficient value depends on the AO method used and is discussed in Section 4.1.

Distance weight. To avoid sampling occlusion at depth discontinuities and across depth layers or image regions where geometry is obliquely projected, we favor views where the projected samples land closer to the central (shaded) point \mathbf{p} (see Figure 3). This is achieved by keeping track of the average distance from \mathbf{p} to the projected samples \mathbf{s}_i on view v :

$$w_d(v, \mathbf{p}) = 1 - \frac{1}{N_s r_{max}} \sum_{i=1}^{N_s} \min(\|\mathbf{p} - \mathbf{s}_i\|, r_{max}) \quad (4)$$

where N_s is the total number of samples per view.

Visibility of the central point \mathbf{p} is handled indirectly through the above weighting function; for an occluded point \mathbf{p} in the current view, many of the projected samples \mathbf{s}_i will lie on the occluder, thus increasing the average distance and lowering the importance of view v . However, the decrease of $w_d(v, \mathbf{p})$ will happen gradually, as more and more samples land on the occluding depth layer when \mathbf{p} moves behind an occluder, thus avoiding any visible banding.

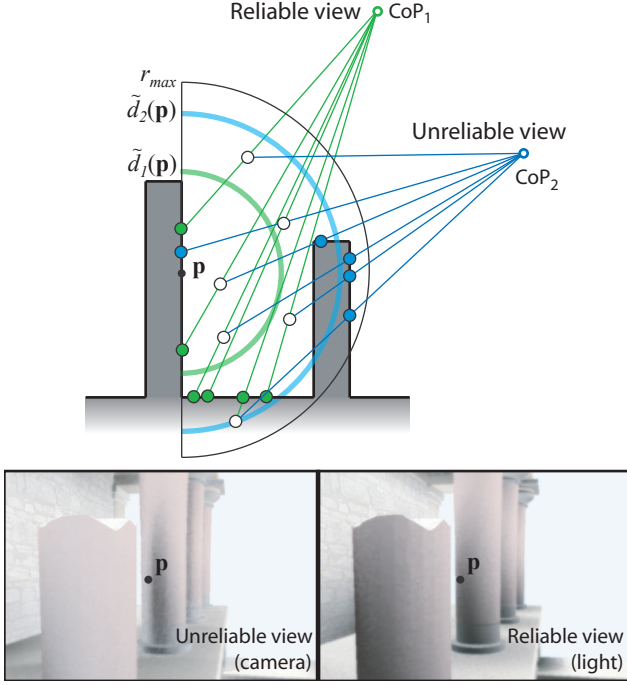


Figure 3: Distance weighting of a view with respect to the average distance of projected samples from a given shaded point. Intense parallax in the depth buffer can cause many false positive samples in the projection of the sampling pattern on the depth buffer.

Note that the calculation of the distance weight incurs no additional cost, since \mathbf{s}_i are already available and $\|\mathbf{p} - \mathbf{s}_i\|$ is needed for the distance-based obscurance attenuation function anyway.

Directional weight. A common problem in screen-space AO techniques is that, when the normal at a shaded point \mathbf{p} diverges from the viewing direction, the probability that a sample is projected on a disjoint or distant patch of geometry, as registered in the depth map of the view, increases. Worse, in the case of multi-view projection, where the normal \mathbf{n} at \mathbf{p} may point away from the view, the projection of the sample has a significantly higher probability to land on a disjoint region than near \mathbf{p} , causing erroneous occlusion. Figure 4 demonstrates this issue; in a highly oblique view, samples on an image disk or in a hemisphere above \mathbf{p} exhibit a strong directional bias towards one side of the sampling pattern and are prone to be projected out of the near-field range r_{max} . Although the effect of this issue varies according to the screen-space algorithm used, the problem may occur in all views, including the camera view (e.g. at grazing angles with respect to the geometry).

A simple criterion to favor views facing along the normal vector at \mathbf{p} is the dot product of \mathbf{n} and \mathbf{l}_v , \mathbf{l}_v being the direction from \mathbf{p} to the center of projection of view v :

$$w_n(v, \mathbf{p}) = \max(0, \mathbf{l}_v \cdot \mathbf{n}) \quad (5)$$

3.3 Importance Sampling

We can use the weighting function of Section 3.2 in order to direct more samples towards views with higher contribution to the final AO. However, since the distance-based weight $w_d(v, \mathbf{p})$ depends on the yet unknown position of the depth samples, we initially generate a first estimate of the occlusion $\tilde{O}_v(\mathbf{p})$ and the corresponding

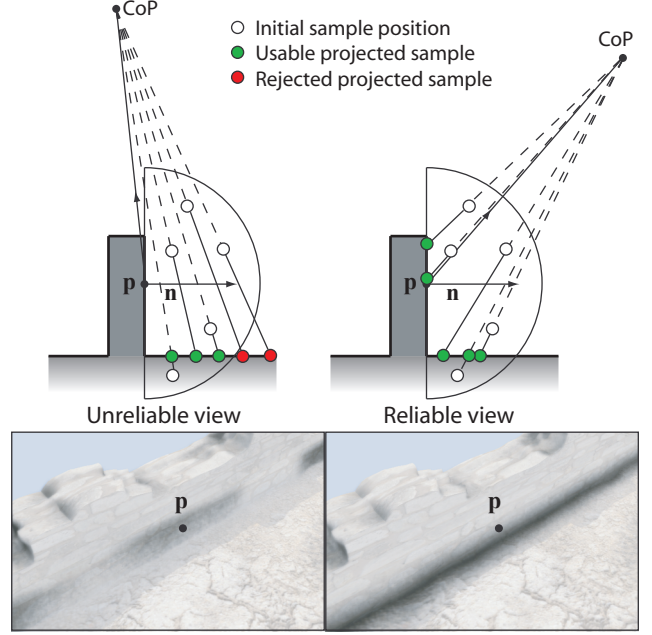


Figure 4: Directional weighting of a view with respect to the normal vector at the shaded point. Oblique projections or viewpoints facing away from the normal can cause severe artifacts in view-dependent near-field geometry recovery from depth.

weights $\tilde{w}(v, \mathbf{p})$ with a small number of samples N_{init} for each view v (e.g. 5-7). The choice of N_{init} depends on the specific AO algorithm used (see quality comparison in Figure 8). $\tilde{O}_v(\mathbf{p})$ and $\tilde{w}(v, \mathbf{p})$ are used to perform importance sampling on the image buffers in a second iteration, with $N_v(\mathbf{p})$ samples for each view v .

The importance of a particular view v and consequently, $N_v(\mathbf{p})$, is directly proportional to the initial weighting function estimate $\tilde{w}(v, \mathbf{p})$. Additionally, since we have a first estimation of the occlusion at \mathbf{p} from view v , $N_v(\mathbf{p})$ can be biased against regions with low occlusion to avoid wasting samples on uninteresting regions. This way we combine importance-driven sample distribution among the views and occlusion-driven adaptive sampling in a single, *adaptive step*. The proposed heuristic for determining $N_v(\mathbf{p})$ is directly proportional to both the initial estimate of the occlusion and the current view weight:

$$N_v(\mathbf{p}) = (N_s - N_{init}) \tilde{w}(v, \mathbf{p}) \frac{(1 + \tilde{O}_v(\mathbf{p}))}{2} \quad (6)$$

The new importance-sampled occlusion $O_{v,is}(\mathbf{p})$ of each view v from the adaptive step is added to the initial estimate $\tilde{O}_v(\mathbf{p})$ and the updated occlusion becomes:

$$O_v(\mathbf{p}) = \frac{O_{v,is}(\mathbf{p})N_v(\mathbf{p}) + \tilde{O}_v(\mathbf{p})N_{init}}{N_v(\mathbf{p}) + N_{init}} \quad (7)$$

This second, adaptive step also calculates new weights $w_{v,is}(\mathbf{p})$ for each view v , which are used to update the weighting function estimate $w_v(\mathbf{p})$, similar to Eq (7):

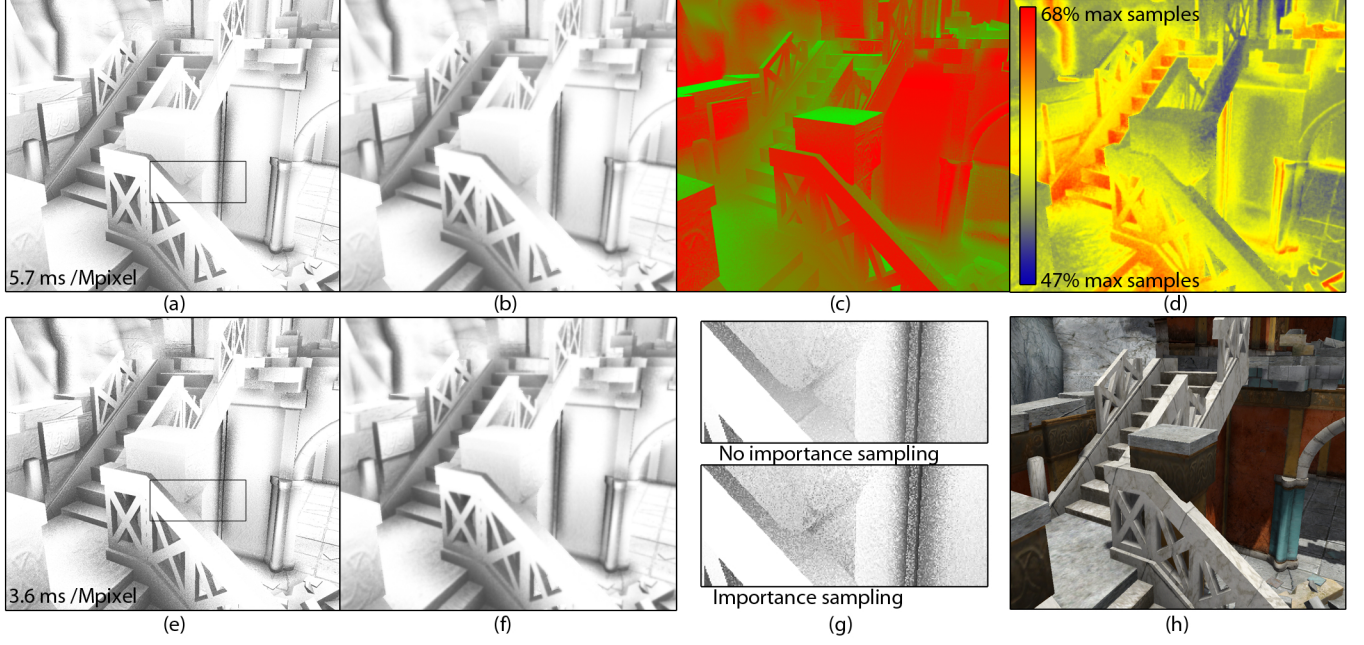


Figure 5: MVAO with importance sampling (IS) using the Alchemy AO variation. Example using the camera and the main light shadow map. (a,b) Raw and filtered obscuration without IS. (c) view weights: red denotes camera weights, green corresponds to light view. (d) Number of samples with respect to maximum samples. (e,f) Raw and filtered obscuration using IS. (g) Close up view of the highlighted region in (a) and (e). (h) The final rendered view.

$$w_v(\mathbf{p}) = \frac{w_{v, \text{is}}(\mathbf{p})N_v(\mathbf{p}) + \tilde{w}_v(\mathbf{p})N_{\text{init}}}{N_v(\mathbf{p}) + N_{\text{init}}} \quad (8)$$

The example in Figure 5 demonstrates how a significant reduction in the total number of samples and the respective rendering time of the AO buffer is achieved, without significantly affecting the image quality. Since more samples are shifted towards more reliable views in terms of depth variance and view orientation, occlusion variance due to view selection is minimized. The occlusion-driven adaptive reduction of the number of samples on the other hand, introduces some noise (see Figure 5(g)), which is however suppressed in the post-filtering stage. In this particular example, the Alchemy AO algorithm was used with a 15-tap rotating kernel and time measurements correspond to rendering 1 megapixel of the AO buffer with 100% geometry coverage. The joint bilateral filtering time to produce the smoothed AO buffers is not included. Both the filtered and unfiltered buffers are provided for comparison (a,b,e,f,g). The heatmap of inset (d) indicates the percentage of the total samples used with respect to the maximum number of samples for both views, i.e. 2×15 here.

The partial occlusion results $O_v(\mathbf{p})$ can be estimated and combined in a single fragment shader for the camera view using an image-space AO algorithm *AOEstimation*:

Algorithm 1 MVAO Fragment Shader

```

for all views  $v$  do                                     ▷ init step
     $\tilde{O}_v(v, \mathbf{p}), \tilde{w}_v(v, \mathbf{p}) \leftarrow \text{AOEstimation}(N_{\text{init}})$ 
for all views  $v$  do                                     ▷ adaptive step
    Estimate  $N_v(\mathbf{p})$  using Eq. 6
     $O_{v, \text{is}}(\mathbf{p}), w_{v, \text{is}}(\mathbf{p}) \leftarrow \text{AOEstimation}(N_v(\mathbf{p}))$ 
    Estimate  $O_v(\mathbf{p}), w_v(\mathbf{p})$  using Eq. 7, 8
    Estimate  $O(\mathbf{p})$  using Eq. 2

```

4 Implementations and Results

Integrating MVAO in a deferred lighting scheme is easy, since the method is a drop-in replacement for a single screen-space AO calculation shader. We ran our tests on a Geforce GTX 570 and we report times in milliseconds per megapixel of an AO buffer with 100% geometry coverage. Reporting the timings for multiple resolutions was not useful, since our experiments showed that they scale linearly with the pixel count, as expected. Performance and quality for each individual algorithm are discussed below.

4.1 Multi-view Ambient Occlusion Variations

We have integrated MVAO with representatives of both open volume and solid angle genres of screen-space algorithms. Figure 6 provides a comparison of the AO produced by a camera-only screen-space technique and its multi-view implementation, as well as the resulting bent normals and final shading. In these implementations, the distance weighting function is the predominant view selection criterion, while the directional weighting function is required only in techniques, which use the surface normal for the occlusion estimation. Figure 7 includes relative timings of the MVAO variations for 2 and 3 views with respect to our single-view algorithm implementations, including importance sampling where applicable. We also report the absolute AO rendering times for our single view versions for completeness. These are only indicative as the relative performance of MVAO to the single-view methods matters.

4.1.1 Solid Angle Algorithms

Alchemy AO. First, we have implemented a variation of the recently developed Alchemy Ambient Obscuration algorithm, due to its efficiency and stochastic rather than explicit horizon estimation. The MVAO version of this algorithm produces stable results and

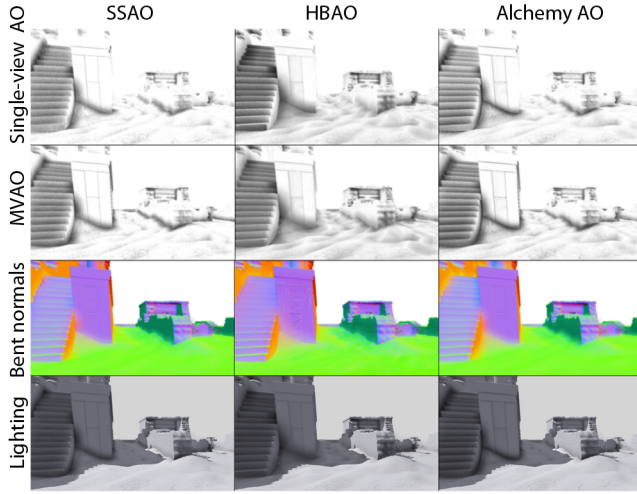


Figure 6: Multi-view implementations of various AO algorithms. In this example, MVAO uses the camera, the shadow map and a fixed phantom view.

the view-dependent artifacts are reduced significantly. In terms of performance, Alchemy AO scales proportionally with the number of views, as shown in the top chart in Figure 7. Additionally, in all test scenes, importance sampling was particularly effective in drastically reducing AO buffer rendering times, which in some cases were lower than the single-view Alchemy AO, for the same maximum number of samples (see the “2 views + IS” column).

HBAO. Horizon-Based AO produces more accurate radial horizon measurements in exchange for more samples and increased computation time, but for the same reason tends to enhance errors when depth information is inaccurate due to high occlusion. When applied to MVAO, it produces banding in the form of silhouettes in regions, where the variance of the distance weights is high, such as the shadow boundaries, if light views are used. In the latter case however, this does not perceptually affect the final image, since the occlusion transitions coincide with the shadow boundaries. Performance-wise, our HBAO variation is the heaviest algorithm, so the stochastic horizon construction of Alchemy AO should be preferred instead.

In our tests, we have found that both solid angle techniques need to favor the distance weighting function ($b \in [0.8, 0.9]$).

4.1.2 Open Volume Algorithms

Screen-space Ambient Occlusion. In our implementation of sample-rejection occlusion, we sample the hemisphere above \mathbf{p} , instead of the entire sphere centered at \mathbf{p} . The depth test mechanism used for the occlusion calculation performs well when adapted to the MVAO scheme, but two modifications are needed. First, the directional weighting function is not necessary as SSAO does not use normals for evaluating the occlusion integral ($b = 1.0$). Second, points on surfaces facing away from the view tend to produce shadowing instead of occlusion. Relying on the distance weighting function to lower the significance of the problematic view would not be sufficient however, since both the shadowed and visible regions with respect to the view can have similar average sample distances. On the other hand, if for all samples that are not visible in the view we assign a distance equal to r_{max} when computing $w_d(v, \mathbf{p})$ we effectively bias the weight against regions in shadow and therefore eliminate the erroneous shadowing.

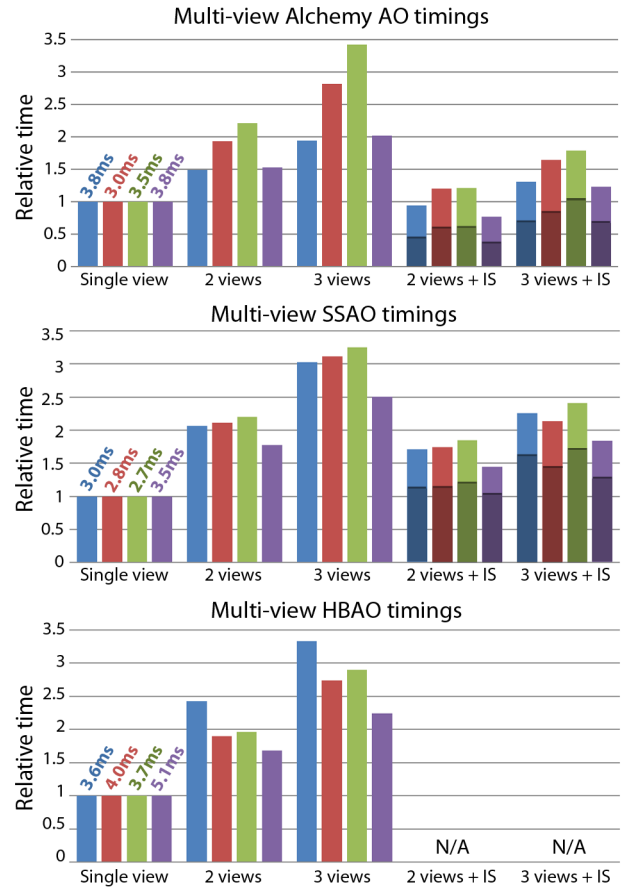
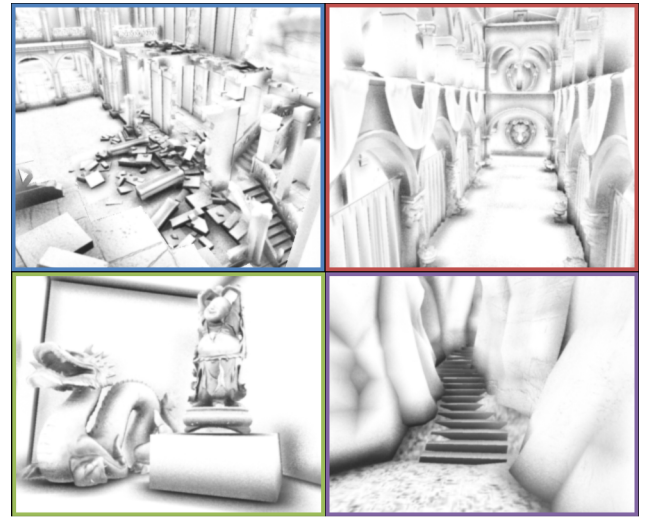


Figure 7: Relative timings of MVAO for various scenes (color-coded insets) with respect to the corresponding single view AO. IS denotes importance sampling and the dark part of the bars corresponds to the initial samples taken (5 and 7 samples for the Alchemy AO and SSAO respectively). IS is not available in HBAO, due to its fixed sampling strategy. Alchemy AO and SSAO both use maximum 15 samples, while HBAO uses 6 slices with 4 radial samples per slice.

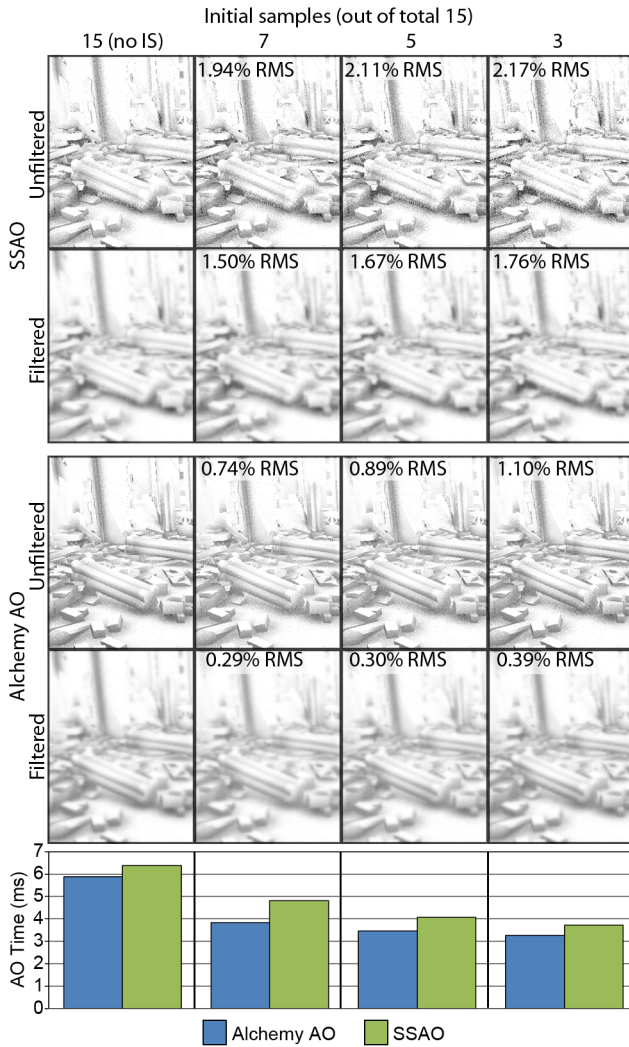


Figure 8: Comparison of AO quality for the multi-view Alchemy AO and SSAO methods, using importance sampling with different N_{init} number of samples for the first estimate of the occlusion and weights. Corresponding AO rendering time is shown for the two methods for a 1Mpixel AO buffer and 1Mpixel shadow map.

In terms of quality, multi-view SSAO results in stable and plausible images. The performance of multi-view SSAO is comparable to that of Alchemy MVAO. However, importance sampling is less effective compared to Alchemy AO, mainly due to higher variance in the initial weight and occlusion estimates. Figure 8 demonstrates this fact; both the visual comparison and the RMS difference of the importance sampling results relative to the basic MVAO method reveal that the slower convergence of SSAO increases the deviation of the initial estimates for the occlusion and the view weights, which in turn lead to a slight degradation of the overall occlusion (1.3%) with respect to the Alchemy AO.

Volumetric Obscure. VO improves the convergence of the occlusion integral by measuring the exact distance between a point on a hemisphere and the projected sample. However, VO is incompatible with a multi-view algorithm, since it relies on distance measurements rather than point sampling; if the sphere of radius r_{max} around \mathbf{p} is below the depth horizon, e.g. when all samples in the vicinity of \mathbf{p} are invisible in the current view, VO produces shadows instead of occlusion. When the extra views are light sources,

the calculated obscuration accentuates deep shadows and softens the shadow boundaries. Since the results deviate from the intended effect we do not present relative timings for this method. Still, VO can be used as an alternative integrated obscuration/shadow generation technique.

4.2 Bent Normals in Multi-view Ambient Occlusion

Multi-view AO also supports the generation of bent normals using the weighted results from different views. A uniform way to do this with negligible overhead is to accumulate for every view v the vectors from the central point \mathbf{p} to every original sample (not projected on the depth buffer) that is not obscured in v , i.e. lies in front of the recorded depth and thus signifies an open direction. The result is blended with the normal at \mathbf{p} . The partial results from all contributing views are weighted in the same manner as the AO values. Figure 6 includes bent normal calculation results for the multi-view implementations described above.

5 View Positioning and Phantom AO Views

MVAO can account for any number of views, but our tests have shown that one or two additional views are usually sufficient for improving significantly the occlusion estimation, even in highly occluded parts of the environment. Adding a large number of views to MVAO is impractical, since the AO computation time would be prohibitive for real-time applications, using today's hardware.

For open environments, or parts of indoor environments where shadow maps provide a significant coverage of the geometry, the latter can be effectively used as secondary views (for example, see teaser and Figure 5). It is important to keep in mind that views that do not reveal complementary geometric information are unlikely to improve the result compared to a single-view AO method, as expected. Therefore, such view points should not be included into the MVAO calculations. For a discussion about typical failure cases, the interested reader may refer to the supplemental material.

For confined spaces and parts of a scene that are not present in a shadow map, instead of using the shadow map(s) as additional views, one may opt to add "phantom" views and render the scene at a low resolution from this points of view. Phantom views can also be set up to follow the user from a different relative view, similar to [Ritschel et al. 2009], complementing the visible geometry of the camera viewpoint. We have tested the four-view configuration described in the above paper and found that it is ineffective in a typical indoor environment, where high occlusion from walls and other pieces of geometry obstruct the visual range of additional cameras pointing towards the user's point of interest. Furthermore, the latter is ill-defined in most cases and thus unreliable.

We propose instead the use of a stereo pair of phantom views arranged as shown in Figure 9; the additional low-resolution cameras render the scene from positions behind the user, but at a fixed, close distance so that the possibility of one of the views being obstructed is minimized. Having the look-at directions of the additional cameras cross each other in front of the user, allows for high coverage from significantly different vantage points near the camera, where accuracy is needed the most. The parallel-to-the-ground configuration was chosen having in mind that in most practical scenarios, the view is obstructed by objects whose vertical extents are dominant (e.g. walls, trees, people, etc). Placing the phantom views behind the camera prevents clipping the geometry in front of the camera against their near planes. For a discussion regarding the resolution of the phantom views, the interested reader should refer to the supplemental material.

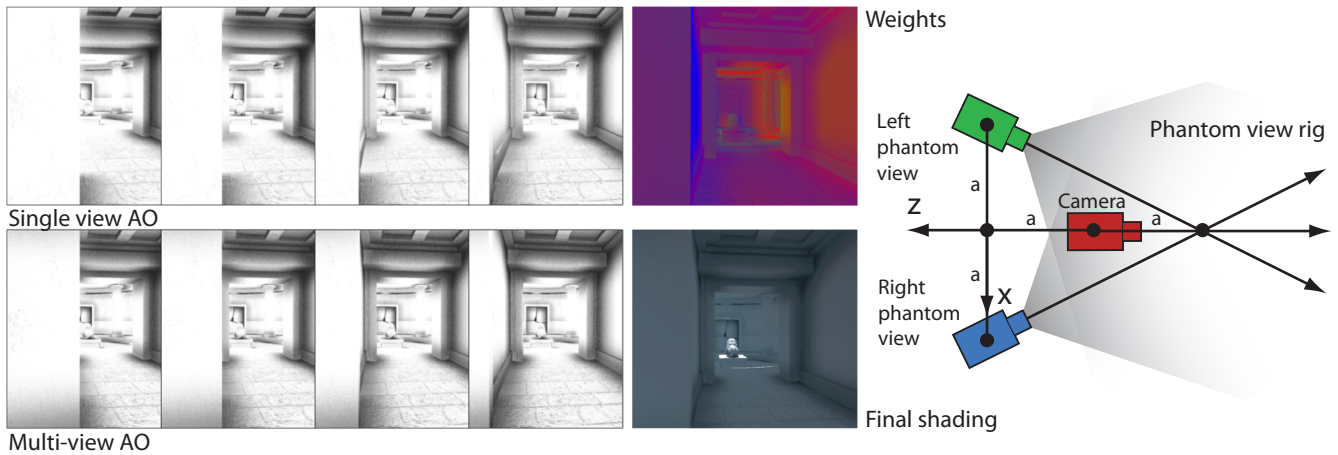


Figure 9: MVAO in confined spaces with heavy occlusion. Using an additional pair of cross-eyed low-resolution “phantom” depth buffers eliminates most view-dependent artifacts.

6 Discussion

Multi-view Ambient Occlusion provides a cost-effective solution to the view dependency issues of image-space methods. We weight the contribution of the camera and other views, such as light sources or phantom views depending on the visibility of point p and its samples, the direction relative to the view and the average distance between the projected samples. Importance Sampling uses the first estimate of the occlusion and weight information to adaptively reduce the total sample count and therefore minimize the overhead imposed by the AO calculations for the extra viewpoints. We show that our algorithm works reasonably well with a variety of techniques and increases the visibility information. However, the selection of viewpoints directly affects the completeness of the occlusion information; geometry not properly represented by any of the depth buffers will still produce erroneous, view-dependent occlusion.

Since our view weighting function and importance sampling scheme is generic, we will investigate its applicability to screen-space indirect lighting and color bleeding techniques. Furthermore, in the same spirit as [Mattausch et al. 2010], we will attempt to exploit temporal coherence by using previous image buffers from different viewpoints, effectively spreading the multiple views over time instead of space. Finally, we will examine the possibility of a dynamic phantom view placement, taking into account factors such as occlusion by adjacent geometry.

Acknowledgements

The Sponza Atrium model (Crytek version) was downloaded from <http://www.crytek.com/cryengine/cryengine3/downloads>. This work was supported by EC FP7 STREP Project PRESIOUS, grant no. 600533.

References

- AKENINE-MÖLLER, T., HAINES, E., AND HOFFMAN, N. 2008. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA.
- BAVOIL, L., AND SAINZ, M. 2009. Multi-layer dual-resolution screen-space ambient occlusion. In *SIGGRAPH 2009: Talks*, ACM, New York, NY, USA, SIGGRAPH ’09, 45:1–45:1.
- BAVOIL, L., SAINZ, M., AND DIMITROV, R. 2008. Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 talks*, ACM, New York, NY, USA, SIGGRAPH ’08, 22:1–22:1.
- FILION, D., AND MCNAUGHTON, R. 2008. Effects & techniques. In *ACM SIGGRAPH 2008 Games*, ACM, New York, NY, USA, SIGGRAPH ’08, 133–164.
- LOOS, B. J., AND SLOAN, P.-P. 2010. Volumetric obscurity. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D ’10, 151–156.
- LUFT, T., COLDITZ, C., AND DEUSSEN, O. 2006. Image enhancement by unsharp masking the depth buffer. *ACM Trans. Graph.* 25, 3 (July), 1206–1213.
- MATTAUSCH, O., SCHERZER, D., AND WIMMER, M. 2010. High-quality screen-space ambient occlusion using temporal coherence. *Computer Graphics Forum* 29, 8 (Dec.), 2492–2503.
- MAX, N. L. 1986. Shadows for bump-mapped surfaces. In *Proceedings of Computer Graphics Tokyo ’86 on Advanced Computer Graphics*, Springer-Verlag New York, Inc., New York, NY, USA, 145–156.
- MCGUIRE, M., OSMAN, B., BUKOWSKI, M., AND HENNESSY, P. 2011. The alchemy screen-space ambient obscurity algorithm. In *High-Performance Graphics 2011*.
- MCGUIRE, M., MARA, M., AND LUEBKE, D. 2012. Scalable ambient obscurity. In *High-Performance Graphics 2012*.
- MITTRING, M. 2007. Finding next gen: Cryengine 2. In *ACM SIGGRAPH 2007 courses*, ACM, New York, NY, USA, SIGGRAPH ’07, 97–121.
- RITSCHER, T., GROSCH, T., AND SEIDEL, H.-P. 2009. Approximating Dynamic Global Illumination in Screen Space. In *Proceedings ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*.
- ZHUKOV, S., INOES, A., AND KRONIN, G. 1998. An Ambient Light Illumination Model. In *Rendering Techniques ’98*, Springer-Verlag Wien New York, G. Drettakis and N. Max, Eds., Eurographics, 45–56.