# Volumetric Ambient Occlusion

*László Szirmay-Kalos*[1], *Tamás Umenhoffer*[1], *Balázs Tóth*[1], *László Szécsi*[1], *Mateu Sbert*[2]

[1] *Budapest University of Technology and Economics, Hungary*

[2] *University of Girona, Spain*

### Abstract

This paper presents a new GPU-based algorithm to compute ambient occlusion. We first examine how ambient occlusion is related to the physically founded rendering equation. The correspondence is made by introducing a fuzzy membership function that defines what "near occlusions" mean. Then we develop a method to calculate ambient occlusion in real-time without any pre-computation. The proposed algorithm is based on a novel interpretation of ambient occlusion that measures how big portion of the tangent sphere of the surface belongs to the set of occluded points. The integrand of the new formula has low variation, thus can be estimated accurately with a few samples. Thus, the algorithm can effectively be used in real-time systems and games to cheaply approximate global illumination effects.

**Keywords:** Real-time ambient occlusion, obscurances, GPU, importance sampling

## 1 Introduction

This paper focuses on the fast computation of the reflection of the ambient light[1]. We shall assume that the primary source of illumination in the scene is a homogeneous sky light source of radiance $L^a$. For the sake of simplicity, we consider only diffuse surfaces. According to the rendering equation, *reflected radiance* $L^r$ in *shaded point* $\vec{x}$ can be obtained as:

$$L^r(\vec{x}) = \frac{a(\vec{x})}{\pi} \int_{\Omega} L^{in}(\vec{x}, \vec{\omega}) \cos^+ \theta \mathrm{d}\omega, \tag{1}$$

where $\Omega$ is the set of directions in the hemisphere above $\vec{x}$, $L^{in}(\vec{x}, \vec{\omega})$ is the *incident radiance* from direction $\vec{\omega}$, $a(\vec{x})$ is the *albedo* of the surface, and $\theta$ is the angle between the surface normal and illumination direction $\vec{\omega}$. If incident angle $\theta$ is greater than 90 degrees, then the negative cosine value should be replaced by zero, which is indicated by superscript $^+$ in $\cos^+$.

If no surface is seen from $\vec{x}$ at direction $\vec{\omega}$, then shaded point $\vec{x}$ is said to be *open* in this direction, and incident radiance $L^{in}$ is equal to ambient radiance $L^a$. If there is an occluder nearby, then the point is called *closed* at this direction and the incident radiance is the radiance of the occluder surface. The exact determination of this radiance would require a global illumination solution, which is too costly in real-time applications. Thus, we simply assume that the radiance is proportional to the ambient radiance and to a factor expressing the *openness* — also called *accessibility* — of the point. The theory of classical ambient occlusion considers an occluder to be "nearby" if its distance is smaller than a predefined threshold $R$. However, such an uncertain property like "being close" is better to handle by a *fuzzy measure* $\mu(d(\vec{\omega}))$ that defines how strongly direction $\vec{\omega}$ belongs to the set of open directions based on distance $d$ of the occlusion at direction $\vec{\omega}$. In other words, this fuzzy measure expresses how an occluder at distance $d$ allows the ambient lighting to take into effect. Relying on the physics analogy of the non-scattering participating media, a possible fuzzy measure could be $\mu(d) = 1 - \exp(-\tau d)$ where $\tau$ is the *absorption coefficient* of the media [6, 1]. Unfortunately, this interpretation requires the distance of far occlusions as well, while the classical ambient occlusion does not have to compute occlusions that are farther than $R$, localizing and thus simplifying the shading process.

[1] Should the scene contain other sources, e.g. directional or point lights, their effects can be added to the illumination of the ambient light.

Thus, for practical fuzzy measures we use functions that are non-negative, monotonously increasing from zero and reach 1 at distance $R$. The particular value of $R$ can be set by the application developer. When we increase this value, shadows due to ambient occlusions get larger and softer.

Using the fuzzy measure of openness, the incident radiance is

$$L^{in}(\vec{x},\vec{\omega}) = L^a \mu(d(\vec{\omega})),$$

thus the reflected radiance (Equation 1) can be written in the following form:

$$L^r(\vec{x}) = a(\vec{x}) \cdot L^a \cdot O(\vec{x}),$$

where

$$O(\vec{x}) = \frac{1}{\pi} \int\limits_{\Omega} \mu(d(\vec{\omega})) \cos^+ \theta \mathrm{d}\omega. \tag{2}$$

is the *ambient occlusion* representing the local geometry. It expresses how strongly the ambient lighting can take effect on point $\vec{x}$. The computation of ambient occlusion requires the distances $d(\vec{\omega})$ of occluders in different directions, which are usually obtained by computationally expensive ray-tracing.

This paper proposes an ambient occlusion algorithm where the directional integral of Equation 2 is replaced by a volumetric one that can be efficiently evaluated. The resulting method

- runs at high frame rates on current GPUs,
- does not require any pre-processing and thus can be applied to general dynamic models,
- can provide smooth shading with just a few samples due to partial analytic integration and interleaved sampling.

These properties make the algorithm suitable for real-time rendering and games.

## 2 Previous work

The oldest ambient lighting model assumes that incident radiance $L^{in}$ is equal to constant $L^a$ in all points and directions, and a point reflects $k_a L^a$ intensity, where $k_a$ is the ambient reflectivity of the surface. As this model ignores the geometry of the scene, the resulting images are plain and do not have a 3D appearance. A physically correct approach would be the solution of the rendering equation that can take into account all factors missing in the classical ambient lighting model. However, this approach is too expensive computationally when dynamic scenes need to be rendered in real-time.

Instead of working with the rendering equation, local approaches examine only a neighborhood of the shaded point. *Ambient occlusion* [4, 11, 8] and *obscurances* [15, 6] methods compute just how "open" the scene is in the neighborhood of a point, and scale the ambient light accordingly. Originally, the neighborhood had a sharp boundary in ambient occlusion and a fuzzy boundary in the obscurances method, but nowadays these terms refer to similar techniques. As the name of ambient occlusion became more popular, we also use this term in this paper.

If not only the openness of the points is computed but also the average of open directions is obtained, then this extra directional information can be used to extend ambient occlusion from constant ambient light to environment maps [11]. In the *spectral* extensions the average spectral reflectivities of the neighborhood and the whole scene are also taken into account, thus even *color bleeding* effects can be cheaply simulated [9, 2].

Since ambient occlusion is the "local invisibility of the sky", real-time methods rely on scene representations where the visibility can be easily determined. These scene representations include the approximation of surfaces by disks [2, 5] or spheres [14]. Instead of dealing directly with the geometry, the visibility function can also be approximated [3]. A cube map or a depth map [10, 13] rendered from the camera can also be considered as a sampled representation of the scene. Since these maps are already in the texture memory of the GPU, a fragment shader program can check the visibility for many directions. The method called *screen-space ambient occlusion* [10] took the difference of the depth values. *Horizon split ambient occlusion* [13] generated and evaluated a horizon map on the fly.

# 3 Volumetric ambient occlusion

The evaluation of the directional integral in the ambient occlusion formula (Equation 2) requires rays to be traced in many directions, which is rather costly and needs complex GPU shaders. Thus, we transform this directional integral to a volumetric one that can be efficiently evaluated on the GPU. The integral transformation involves the following steps:

1. Considering its derivative instead of the fuzzy membership function, we replace the expensive ray tracing operation by a simple containment test. This replacement is valid if neighborhood $R$ is small enough to allow the assumption that a ray intersects the surface at most once in interval $[0, R]$.
2. Factor $\cos^+ \theta$ in Equation 2 is compensated by transforming the integration domain mapping the hemisphere above the surface to the tangent sphere. This makes ambient occlusion depend on the volume of that portion of the tangent sphere which belongs to the "free" space that is not occupied by objects.

## 3.1 Replacing ray tracing by containment tests

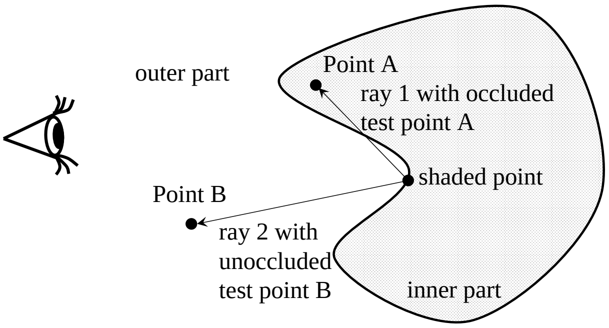


Fig. 1: Replacing ray tracing by containment tests. If a test point (Point B) along a ray starting at the shaded point being in the outer region or on the region boundary is also in the outer region, then the ray either has not intersected the surface or has intersected at least two times. Supposing that the ray is short enough and thus may intersect the surface at most once, the condition of being in the outer region is equivalent to the condition that no intersection happened.

Let us assume that the surfaces subdivide the space into an *outer part* where the camera is and into *inner parts* that cannot be reached from the camera without crossing a surface. We define the *characteristic function* $\mathcal{I}(\vec{p})$ that is 1 if point $\vec{p}$ is an outer point and zero otherwise. The boundary between the inner and outer parts, i.e. the surfaces, belong to the outer part by definition (Figure 1).

The form of the indicator function $\mathcal{I}(\vec{p})$ depends on the type or representation of the surfaces (Figure 2).

1. *Implicit surfaces* are defined by equation $f(\vec{p}) = 0$. In this case $\mathcal{I}(\vec{p})$ is 1 if $f(\vec{p})$ is positive or zero (the point is outside of the object or on its surface), and zero otherwise.
2. *Height fields* represent a particularly important special case of implicit surfaces. A height field is formed by points $(x, y, z)$ satisfying equation $z = h(x, y)$. Since the eye is usually "above" the height field, the characteristic function should indicate the case when for point $\vec{p} = (p_x, p_y, p_z)$, relation $p_z \geq h(p_x, p_y)$ holds. Note that a polygon of a *displacement mapped* mesh can also be considered as a height field in the tangent space of the macrostructure polygon.
3. As in screen-space ambient occlusion methods [10, 13], the content of the z-buffer can also provide an inner–outer distinction. If a point is reported to be occluded by the depth map, then surfaces separate this point from the eye, thus its characteristic value is zero. If the point passes the depth test, then it is in the same region as the eye, so it gets indicator value 1.

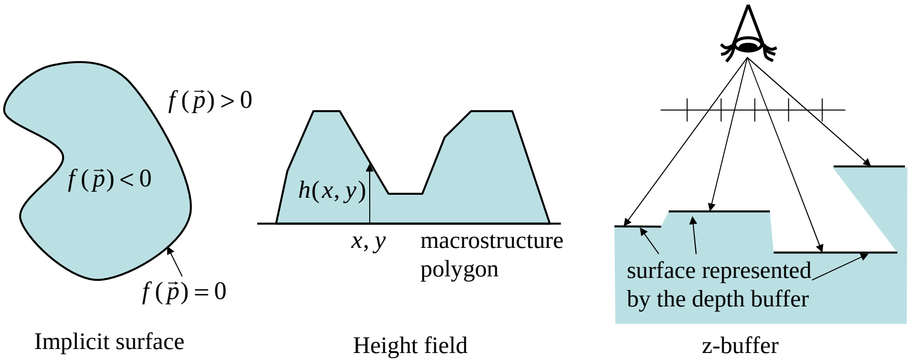


Fig. 2: Three examples for the definition of the characteristic function. Outer regions where $\mathcal{I} = 1$ are blank, inner regions where $\mathcal{I} = 0$ are filled.

Note that in all these three cases, not only are we able to classify a point as inner or outer, but we can also determine the distance between point $(p_x, p_y, p_z)$ and the surface along axis $z$. When the content of the z-buffer defines the separation, the $z$-direction is the viewing direction in clipping space. Reading depth value $z^*$ with the $p_x, p_y$ coordinates of the point, the distance between the point and the surface can be expressed as $z^* - p_z$. If the characteristic function is defined by height field $h(x, y)$, then the distance along axis $z$ is $p_z - h(p_x, p_y)$. Finally, when the surface is defined by implicit equation $f(x, y, z) = 0$, we can use Taylor's approximation to estimate the distance between point $(p_x, p_y, p_z)$ and point $(p_x, p_y, z^*)$ that is on the surface:

$$f(p_x, p_y, p_z + (z^* - p_z)) = 0 \quad \Rightarrow \quad z^* - p_z \approx -\frac{f(p_x, p_y, p_z)}{\partial f/\partial z}.$$

In order to evaluate the ambient occlusion using containment tests, we express it as a three dimensional integral. For a point at distance $d$, fuzzy measure $\mu(d)$ can be found by integrating its derivative from 0 to $d$ since $\mu(0) = 0$. Then the integration domain can be extended from $d$ to $R$ by multiplying the integrand by a step function which replaces the derivative by zero when the distance is greater than $d$:

$$\mu(d) = \int\limits_0^d \frac{\mathrm{d}\mu(r)}{\mathrm{d}r}\mathrm{d}r = \int\limits_0^R \frac{\mathrm{d}\mu(r)}{\mathrm{d}r}\epsilon(d-r)\mathrm{d}r,$$

where $\epsilon(x)$ is the step function, which is 1 if $x \geq 0$ and zero otherwise. Note that this formulation requires generalized derivatives for classical ambient occlusion since its membership function is a step function, and the derivative of the step function is a Dirac-delta (Figure 3).

Substituting this integral into the ambient occlusion formula, we get

$$O(\vec{x}) = \frac{1}{\pi}\int\limits_{\Omega}\int\limits_0^R \frac{\mathrm{d}\mu(r)}{\mathrm{d}r}\epsilon(d-r)\cos^+\theta \mathrm{d}r\mathrm{d}\omega. \tag{3}$$

Let us consider a ray of equation $\vec{x} + \vec{\omega}r$ where shaded point $\vec{x}$ is the origin, $\vec{\omega}$ is the direction, and distance $r$ is the ray parameter. Indicator $\epsilon(d-r)$ is 1 if the distance of the intersection $d$ is larger than current distance $r$ and zero otherwise, that is, it shows whether or not intersection has happened. If we assume that the ray intersects the surface at most once in the $R$-neighborhood, then the condition that $\vec{x} + \vec{\omega}r$ is in the outer part, i.e. $\mathcal{I}(\vec{x} + \vec{\omega}r) = 1$, also shows that no intersection has happened yet. Thus, provided that only one intersection is possible in the $R$-neighborhood, indicators $\epsilon(d-r)$ and $\mathcal{I}(\vec{x} + \vec{\omega}r)$ are equivalent. Replacing step function $\epsilon(d-r)$ by indicator function $\mathcal{I}$ in Equation 3, we obtain

$$O(\vec{x}) = \frac{1}{\pi}\int\limits_{\Omega}\int\limits_0^R \frac{\mathrm{d}\mu(r)}{\mathrm{d}r}\mathcal{I}(\vec{x} + \vec{\omega}r)\cos^+\theta \; \mathrm{d}r\mathrm{d}\omega. \tag{4}$$

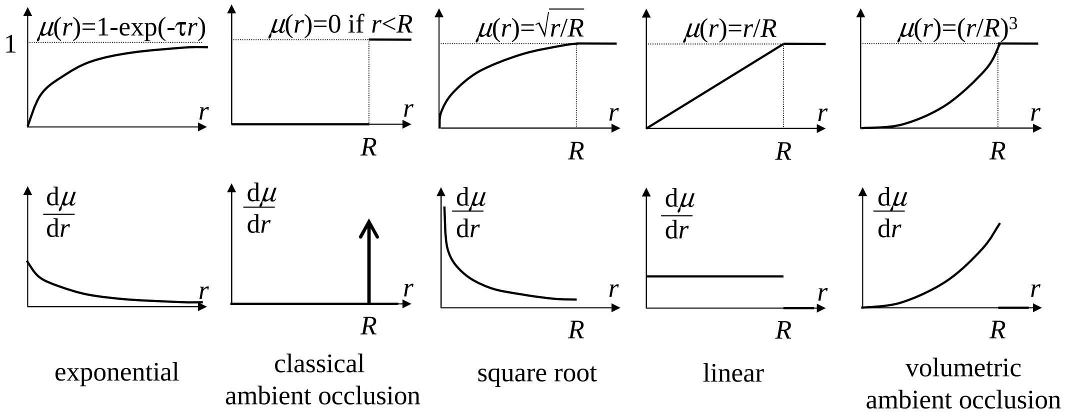


Fig. 3: Example fuzzy membership functions and their derivatives. Note that the exponential function [6, 1] can be given a physical interpretation but it requires global visibility computations, while other functions need visibility checks only in a neighborhood of radius $R$. Classical ambient occlusion [11] uses a non-fuzzy separation. The localized square root is a good compromise between the global exponential and the non-fuzzy separation [9]. We also included the membership function of the volumetric ambient occlusion that is proposed in the next subsection.

One possibility for the estimation of this integral is the Monte Carlo quadrature. According to the concept of *importance sampling* we use cosine distributed sample directions $\vec{\omega}_i$ and distance samples $r_i$ following density $\mathrm{d}\mu(r)/\mathrm{d}r$, that are obtained by transforming uniformly distributed samples $\xi_i$ as $r_i = \mu^{-1}(\xi_i)$. Note that in our case the number of samples $n$ is small and the random samples should be generated only once, thus we can hardwire samples $(X_i, Y_i, Z_i) = \vec{\omega}_i r_i$ into the shader code computing

$$O(\vec{x}) \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}\left(\vec{x} + X_i \vec{T} + Y_i \vec{B} + Z_i \vec{N}\right). \tag{5}$$

where $\vec{T}$, $\vec{B}$, and $\vec{N}$, are the tangent, binormal, and normal vectors, respectively. We call this method the *containment test based algorithm.* This method would approximate the quadrature averaging $n$ binary numbers, thus the average would have *binomial distribution* with mean $O(\vec{x})$ and standard deviation $\sqrt{O(\vec{x})(1 - O(\vec{x}))/n}$. Ambient occlusion $O(\vec{x})$ is in $[0, 1]$ and the standard deviation reaches its maximum $1/\sqrt{4n}$ when the ambient occlusion value is 1/2. In order to reduce the maximum standard deviation (i.e. the error) below 0.01 in a pixel, we need 2500 random samples, which are too many for real-time applications.

Thus, we need a better estimate for the ambient occlusion integral that requires less samples to achieve the same accuracy. We combine three techniques to reach this goal. Most importantly, the integral is reformulated to reduce the variation of the integrand and to allow the incorporation of all available information into the quadrature. In particular, we use the distance to the separating surface as such additional information. Secondly, we replace statistically independent random samples by low-discrepancy samples following the Poisson-disk distribution. Finally, we use interleaved sampling, i.e. exploit the samples obtained in the neighboring pixels to improve the estimate in the current pixel.

## 3.2 Exploiting the distance to the separating surface

Inspecting Equation 4 we can observe that the ambient occlusion is a double integral inside a hemisphere where the integrand includes factor $\cos^{+}\theta$, thus directions enclosing a larger angle with the surface normal are less important. Instead of the multiplication, the effect of this cosine factor can also be mimicked by reducing the size of the integration domain proportionally to $\cos^{+}\theta$. Note that this is equivalent to the original integral only if the remaining factors of the integrand are constant, and can be accepted as an approximation in other

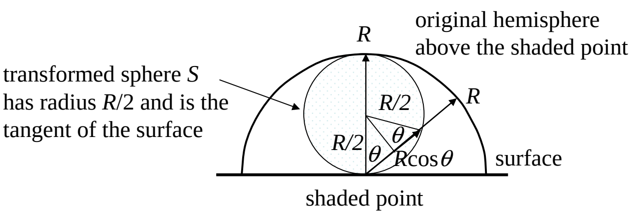


Fig. 4: Transforming a hemisphere of radius $R$ by shrinking distances by $\cos\theta$ results in another sphere of radius $R/2$.

cases:

$$O(\vec{x}) \approx \frac{1}{\pi} \int_{\Omega} \int_{0}^{R\cos^{+}\theta} \frac{\mathrm{d}\mu(r)}{\mathrm{d}r} \mathcal{I}(\vec{x} + \vec{\omega} r) \mathrm{d}r \mathrm{d}\omega.$$

Transforming a hemisphere by shrinking distances in directions enclosing angle $\theta$ with the surface normal by $\cos\theta$ results in another sphere, which is denoted by $S$ (Figure 4). This new sphere has radius $R/2$ and its center is at distance $R/2$ from shaded point $\vec{x}$ in the direction of the surface normal. While the original hemisphere had the shaded point as the center of its base circle, the surface will be the tangent of the new sphere at shaded point $\vec{x}$.

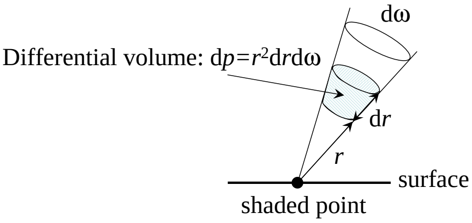


Fig. 5: Differential volume swept when $r$ changes by $dr$ and direction $\vec{\omega}$ is in $d\omega$.

In order to replace integrals over directions and distances by a volumetric integral, let us examine the volume swept when distance $r$ changes by $\mathrm{d}r$ and direction $\vec{\omega}$ varies in solid angle $\mathrm{d}\omega$ (Figure 5). During this, we sweep a differential volume $\mathrm{d}p = r^2 \mathrm{d}r \mathrm{d}\omega$. Thus, we can express the ambient occlusion as a volumetric integral in $S$ instead of a double integral of directions and distances in the following way:

$$O(\vec{x}) \approx \frac{1}{\pi} \int_{\vec{p} \in S} \frac{\mathrm{d}\mu(r(\vec{p}))}{\mathrm{d}r} \frac{1}{(r(\vec{p}))^2} \mathcal{I}(\vec{p}) \mathrm{d}p$$

where $r(\vec{p})$ is the distance of point $\vec{p}$ from the shaded point. If we set the fuzzy membership function such that $\mathrm{d}\mu(r)/\mathrm{d}r$ is proportional to $r^2$ (last column of Figure 3), then the ambient occlusion integral becomes just the volumetric integral of the membership function. This observation leads us to a new definition of the openness of a point, which we call the *volumetric ambient occlusion* and denote by $V(\vec{x})$ to distinguish it from ambient occlusion $O(\vec{x})$. The volumetric ambient occlusion is the relative volume of the unoccluded part of the tangent sphere $S$. Formally, the volumetric ambient occlusion function is defined as:

$$V(\vec{x}) = \frac{\int_S \mathcal{I}(\vec{p}) \mathrm{d}p}{|S|}, \tag{6}$$

where $|S| = 4(R/2)^3\pi/3$ is the volume of the tangent sphere, which makes sure that the volumetric ambient occlusion is also in $[0, 1]$. Figure 6 compares the computation of volumetric ambient occlusion to ray tracing based methods and to the containment test based algorithm.

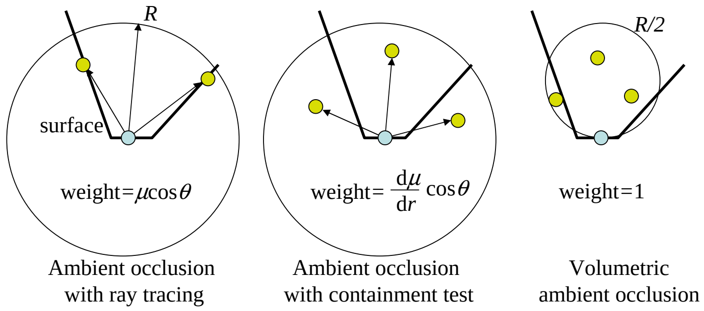

R
surface
weight=$\mu\cos\theta$
Ambient occlusion with ray tracing
weight=$\frac{d\mu}{dr}\cos\theta$
Ambient occlusion with containment test
R/2
weight=1
Volumetric ambient occlusion

Fig. 6: Comparison of the ambient occlusion computation with ray-tracing, containment test, and with the volumetric interpretation. Note that these methods differ both in sample generation and weighting.

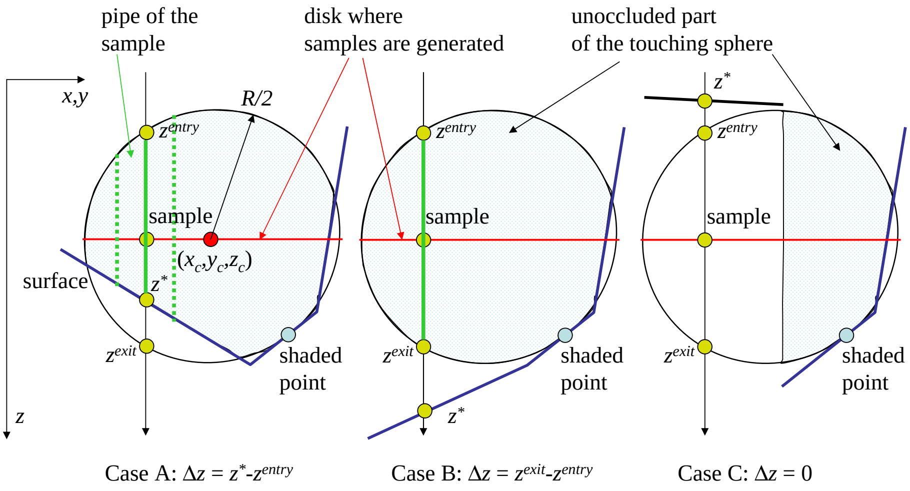

pipe of the sample
disk where samples are generated
unoccluded part of the touching sphere
x,y
z
R/2
$z^{entry}$
sample
$(x_c,y_c,z_c)$
surface
$z^*$
$z^{exit}$
shaded point
Case A: $\Delta z = z^*-z^{entry}$
Case B: $\Delta z = z^{exit}-z^{entry}$
Case C: $\Delta z = 0$

Fig. 7: Computation of the volumetric ambient occlusion. The volume of the unoccluded part of the tangent sphere of radius $R/2$ is approximated by uniformly sampling $n$ points called *sample* on the disk that is perpendicular to axis $z$. Each sample represents the same base area $(R/2)^2\pi/n$ of the disk. Such a base area is the cross section of a "pipe" inside the unoccluded part of the sphere. The volume of the pipe associated with a sample is the base area multiplied with the length of the line segment that is in the unoccluded part of the tangent sphere. The line crossing the sample point enters the sphere at $z$ coordinate $z^{entry}$, exits it at $z^{exit}$, and intersects the surface at $z^*$. To obtain length $\Delta z$ of the line segment that is in the unoccluded part of the sphere, we identify three main cases. In Case A the line–surface intersection is in the sphere. In Case B the surface is behind the sphere. In Case C the surface is in front of the sphere.

When evaluating volumetric integral $\int_S \mathcal{I}(\vec{p})\ dp$, we can take advantage of the distances to the separating surface. The volume is computed as a sum of "pipes" (Figure 7). The axes of these pipes are parallel to axis $z$ and they have the same cross section area. The pipes are limited by either the separating surface or the surface of the tangent sphere.

Let us denote the center of the tangent sphere by $(x_c, y_c, z_c)$ and consider a disk of radius $R/2$ around this point that is perpendicular to axis $z$. We sample $n$ uniformly distributed points $(x_i, y_i)$ in a unit radius disk, and transform them onto the considered disk of the tangent sphere, that is perpendicular to direction $z$. A transformed point has coordinates $(x_i R/2, y_i R/2, z_c)$ and is called *sample* in Figure 7. A line crossing the $i$th sample point and being parallel with axis $z$ enters the sphere at $z_i^{entry} = z_c - \frac{R}{2}\sqrt{1 - x_i^2 - y_i^2}$ and exits it at $z_i^{exit} = z_c + \frac{R}{2}\sqrt{1 - x_i^2 - y_i^2}$. The points on this line belong to the outer region when their $z$ coordinates are less than $z^*$, where $z^*$ represents the intersection of this line with the surface.

The length of traveling in the outer, i.e. unoccluded part of the sphere is $\Delta z_i = z_i^* - z_i^{entry}$ when $z_i^{entry} \leq z_i^* \leq z_i^{exit}$ (Case A in Figure 7). The traveled distance is $\Delta z_i = z_i^{exit} - z_i^{entry}$ if the surface is behind the sphere (Case B). If $z_i^* < z_i^{entry}$, then $\Delta z_i = 0$ since this part of the sphere is occluded (Case C).

If $n$ sample points are uniformly distributed on the disk of radius $R/2$, then a line is associated with $(R/2)^2\pi/n$ area of the disk. Thus, we get the following approximation of the volume of the unoccluded part of the tangent sphere:

$$\int_S \mathcal{I}(\vec{p})\mathrm{d}p \approx \frac{R^2\pi}{4n}\sum_{i=1}^{n}\Delta z_i. \tag{7}$$

The fragment shader gets samples $(x_i, y_i)$ as a constant array and estimates the volumetric ambient occlusion of the point as

$$V(\vec{x}) = \frac{\int_S \mathcal{I}(\vec{p})\mathrm{d}p}{|S|} \approx \frac{3}{2Rn}\sum_{i=1}^{n}\Delta z_i = \frac{1}{F}\sum_{i=1}^{n}\Delta z_i. \tag{8}$$

Note that the normalization constant $F$ is the ratio of the tangent sphere's volume $|S| = 4(R/2)^3\pi/3$ and $R^2\pi/(4n)$, which can be analytically computed. However, instead of the exact value of $|S|$, it is worth approximating it with the same samples as used to compute the volume of the unoccluded part (Equation 7), which results in the following constant

$$F = R\sum_{i=1}^{n}\sqrt{1 - x_i^2 - y_i^2}.$$

In this case, the approximations of the volume of the unoccluded part and the volume of the tangent sphere have correlated error. Thus, when their ratio is computed, the error of the volume of the unoccluded part is reduced, as proposed by *weighted importance sampling* [12].

## 3.3 Noise reduction with interleaved sampling

The quasi-Monte Carlo quadrature has some error in each pixel, which depends on the particular samples used in the quadrature. If we used different quasi-random numbers in neighboring pixels, then dot noise would show up. Using the same quasi-random numbers in every pixel would make the error correlated and replace dot noise by "stripes". Unfortunately, both stripes and pixel noise are quite disturbing. In order to reduce the error without taking excessive number of samples, we apply *interleaved sampling* [7] that uses different sets of samples in the pixels of a 4 × 4 pixel pattern, and repeat the same sample structure periodically. The 16 different sample sets can be obtained from a single set by a rotation around the surface normal vector by random angle $\alpha$. The rotation is executed in the fragment shader that gets 16 $(\cos\alpha, \sin\alpha)$ pairs in addition to quasi-random samples $(x_i, y_i)$. The errors in the pixels of a 4 × 4 pixel pattern are uncorrelated, and can be successfully reduced by a low-pass filter of the same size. Thus, interleaved sampling using a 4 × 4 pixel pattern multiplies the effective sample number by 16 but has only the added cost of a box-filtering with a 4 × 4 pixel window. When implementing the low-pass filter, we also check whether or not the depth difference of the current and the neighbor pixels exceeds a given limit. If it does, then the neighbor pixel is not included in the averaging operation.

## 4 Results

The proposed methods have been implemented in DirectX/HLSL environment and their performance has been measured on an NVIDIA GeForce 8800 GTX GPU at 800 × 600 resolution.

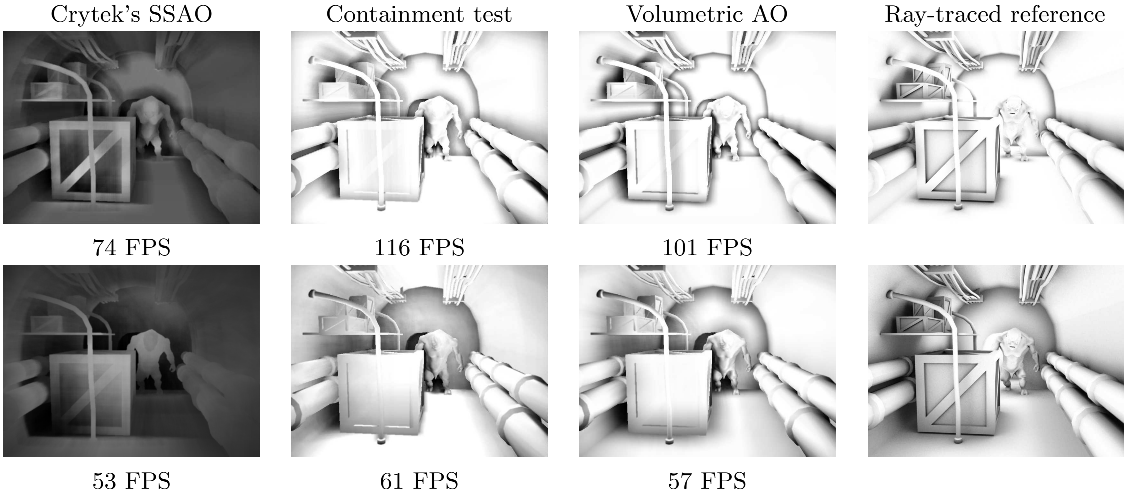


Fig. 8: Comparison of Crytek's screen-space ambient occlusion, the containment test based method, volumetric ambient occlusion, and a reference that is obtained with ray-tracing. Images in the upper and lower rows are rendered with a smaller ($R = 3$) and a larger ($R = 9$) neighborhood, respectively. We used 32 samples per pixel in all cases. We incorporated membership function $\mu = (r/R)^3$ in the containment test based and in the ray-tracing methods, which is the same membership function that is implicitly used by volumetric ambient occlusion.

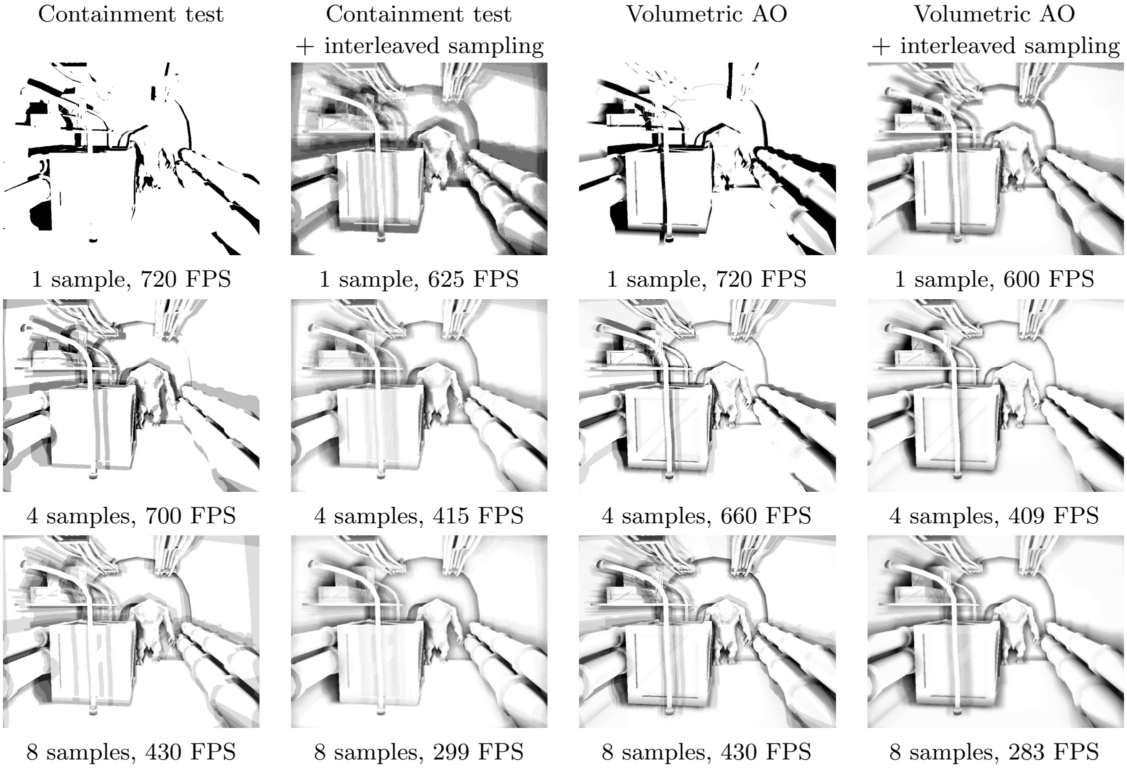
Figure 8 compares Crytek's screen-space ambient occlusion[2], the containment test based method (Equation 5), the volumetric ambient occlusion (Equation 8), and a reference that is obtained with the software ray-tracer of MentalRay. We note that screen-space ambient occlusion takes samples on the whole sphere and is not based on the original ambient occlusion formula, thus it assigns middle gray even for completely unoccluded surfaces, giving an unrealistically dark touch to the image. Note that both the containment test based and the volumetric methods are faster than Crytek's SSAO, and the speed is sensitive to the size of the neighborhood in all cases. This sensitivity is due to the degraded cache efficiency of z-buffer accesses when the large $R$ requires distant samples to be fetched. This degradation can be avoided by reducing the z-buffer resolution by an additional z-buffer filtering before the ambient occlusion computation. In terms of quality the volumetric ambient occlusion is the closest to the ray-traced reference. The quality degradation with respect to the ray-traced result is mainly due to the assumption that in the $R$-neighborhood at most one intersection can happen. This assumption limits the simultaneous consideration of close and distant occlusions, which reduces the level of details in the final image. This is the price we should pay for real-time rendering.

Figure 9 evaluates the performance-quality tradeoff for the containment test based and the volumetric ambient occlusion algorithms, and also shows the effect of interleaved sampling. As the volumetric approach evaluates a part of the integration analytically, its results are smoother when just a few samples per pixel are computed. The additional interleaved sampling helps eliminating sampling artifacts in all cases, but it also has a cost and mildly blurs the image.

Figure 10 shows the power of weighted importance sampling. We rendered the left image with the analytic formula (Equation 8) and the middle image with weighted importance sampling taking only 4 samples per pixel. Note that the image obtained with weighted importance sampling is closer to the reference rendered with 32 samples.

Figure 11 demonstrates the differences of Crytek's screen-space ambient occlusion, the classical ambient occlusion obtained with containment tests, and our new volumetric ambient occlusion. We took only $n = 8$ samples per pixel in all cases, and frame rates were also similar (about 250 FPS). As the volumetric ambient

[2] http://en.wikipedia.org/wiki/Screen_Space_Ambient_Occlusion



Containment test | Containment test + interleaved sampling | Volumetric AO | Volumetric AO + interleaved sampling

1 sample, 720 FPS | 1 sample, 625 FPS | 1 sample, 720 FPS | 1 sample, 600 FPS

4 samples, 700 FPS | 4 samples, 415 FPS | 4 samples, 660 FPS | 4 samples, 409 FPS

8 samples, 430 FPS | 8 samples, 299 FPS | 8 samples, 430 FPS | 8 samples, 283 FPS

Fig. 9: Comparison of the containment test based and the volumetric ambient occlusion (AO) using different numbers of samples per pixel and turning interleaved sampling on and off.

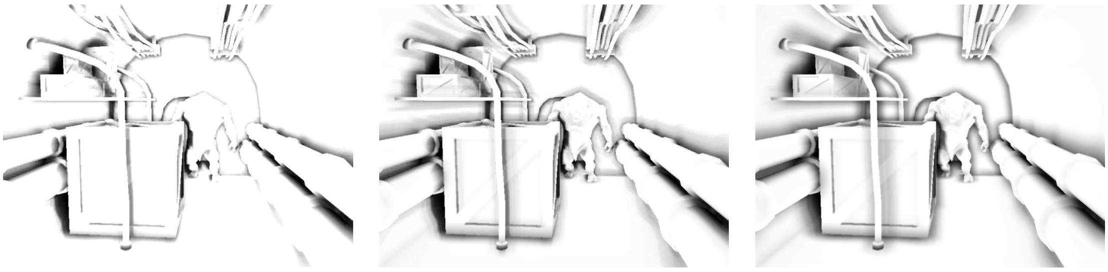
occlusion method needs to integrate a lower variation integrand, it provides smoother and better results than screen-space ambient occlusion or containment test based methods.

In our implementation the characteristic function may be a product of two characteristic functions. One separates the inner and outer parts according to the z-buffer. The other handles displacement mapped polygons considering them as height fields defined in tangent space. The z-buffer based classification is responsible for occlusions of other objects, while the height field based classification handles self-shadowing. Note that this way the ambient occlusion value can be obtained for displacement mapped surfaces even if the depth value is not modified in the fragment shader, and the accuracy problems of the z-buffer are also eliminated.

Figure 12 compares images rendered with only the z-buffer based characteristic function and when height field occlusions are also considered, and also shows the final image due to environment map lighting. The volumetric ambient occlusion is computed with $n = 12$ samples per pixel. Note that our ambient occlusion computation is just slightly more expensive than environment lighting, but significantly enhances the details both on the level of macrostructure geometry and on the level of displacement maps.

## 5 Conclusions

This paper proposed a fast method for the computation of ambient occlusion. The new approach is based on rewriting the ambient occlusion integral to evaluate the volume of the unoccluded part of the tangent sphere, and on its accurate approximation taking advantage of uniform sequences transformed to mimic the new integrand. We also examined the correspondence of the new volumetric ambient occlusion formula to the classical obscurances or ambient occlusion. However, we have to emphasize that the volumetric ambient occlusion can also be considered as a new definition for the openness or accessibility of a point. The important advantage of the new formula is that it can be evaluated more accurately with the same number of samples, thus, it is more appropriate in real-time systems where low noise results are needed with just a few cheap samples. The method also works for dynamic geometry with dynamic height fields and displacement maps.



Analytic, 4 samples

WIS, 4 samples

Reference, 32 samples

Fig. 10: Effects of the proposed weighted importance sampling (WIS). Note that the image rendered with weighted importance sampling from 4 samples is much closer to the reference image generated from 32 samples.

It does not require pre-processing and runs at high frame rates, since it only needs as few as 8-10 samples per pixel.

## 6 Acknowledgement

This work has been supported by the National Scientific Research Fund (OTKA ref. No.: T042735) and by TIN 2007-68066-C04-01 from the Spanish Government.

## References

[1] T. Annen, T. Mertens, H.-P. Seidel, E. Flerackers, and J. Kautz. Exponential shadow maps. In *GI '08: Proceedings of graphics interface 2008*, pages 155–161, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[2] M. Bunnel. Dynamic ambient occlusion and indirect lighting. In M. Parr, editor, *GPU Gems 2*, pages 223–233. Addison-Wesley, 2005.

[3] P. Clarberg and T. Akenine-Möller. Exploiting Visibility Correlation in Direct Illumination. *Computer Graphics Forum (Proceedings of EGSR 2008)*, 27(4):1125–1136, 2008.

[4] L. Hayden. Production-ready global illumination. Technical report, SIGGRAPH Course notes 16, 2002. http://www.renderman.org/RMR/Books/ sig02.course16.pdf.gz.

[5] J. Hoberock and Y. Jia. High-quality ambient occlusion. In Hubert Nguyen, editor, *GPU Gems 3*, pages 257–274. Addison-Wesley, 2007.

[6] A. Iones, A. Krupkin, M. Sbert, and S. Zhukov. Fast realistic lighting for video games. *IEEE Computer Graphics and Applications*, 23(3):54–64, 2003.

[7] A. Keller and W. Heidrich. Interleaved sampling. In *Rendering Techniques 2001 (Proceedings of the 12th Eurographics Workshop on Rendering)*, pages 269–276, 2001.

[8] J. Kontkanen and T. Aila. Ambient occlusion for animated characters. In *Proceedings of the 2006 Eurographics Symposium on Rendering*, 2006.

[9] A. Méndez, M. Sbert, and J. Catá. Real-time obscurances with color bleeding. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 171–176, New York, NY, USA, 2003. ACM.

[10] M. Mittring. Finding next gen — CryEngine 2. In *Advanced Real-Time Rendering in 3D Graphics and Games Course - Siggraph 2007*, pages 97–121. 2007.

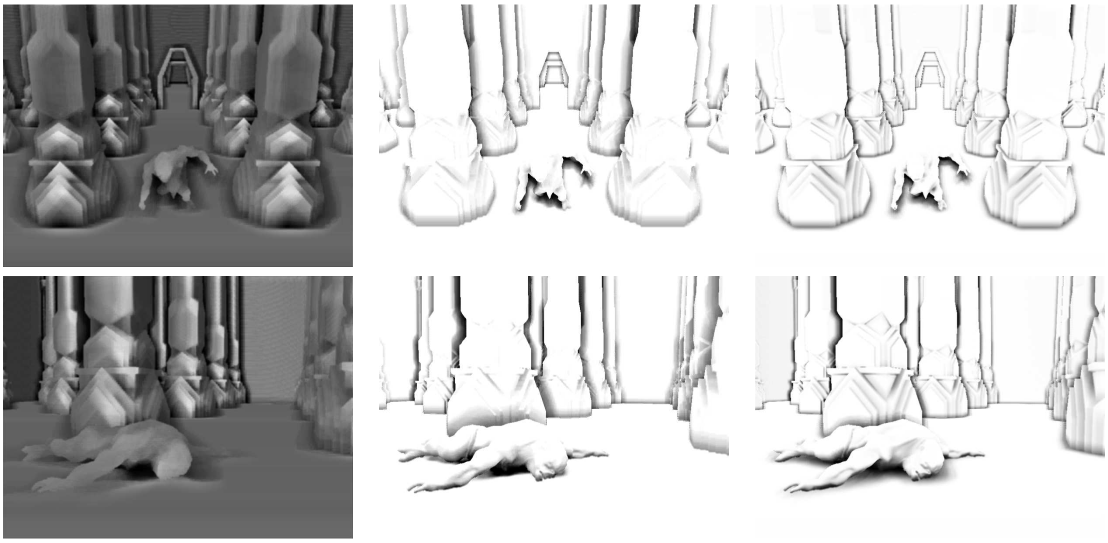
[11] M. Pharr and S. Green. Ambient occlusion. In *GPU Gems*, pages 279–292. Addison-Wesley, 2004.

[12] M. Powell and J. Swann. Weighted importance sampling — a Monte-Carlo technique for reducing variance. *Inst. Maths. Applics.*, 2:228–236, 1966.

[13] M. Sainz. Real-time depth buffer based ambient occlusion. In *Games Developers Conference '08*. 2008.

[14] P. Shanmugam and O. Arikan. Hardware accelerated ambient occlusion techniques on GPUs. In *Proceedings of the 2007 Symposium on Interactive 3D graphics*, pages 73–80, 2007.

[15] S. Zhukov, A. Iones, and G. Kronin. An ambient light illumination model. In *Proceedings of the Eurographics Rendering Workshop*, pages 45–56, 1998.



Crytek's SSAO

Classical AO<br>$\mu = \epsilon(d - R)$

Volumetric AO

Fig. 11: Comparison of Crytek's screen space ambient occlusion, the containment test based ambient occlusion using the step membership function, and volumetric ambient occlusion. We used as few as 8 samples per pixel. The scene is built of 70138 triangles. Crytek's SSAO is a little slower than the other two algorithms that run at 250 FPS.

## Authors

### László Szirmay-Kalos

is the head of the computer graphics group and the Department of Control Engineering and Information Technology at the Budapest University of Technology and Economics. He received PhD in 1992 and full professorship in 2001 in computer graphics. He also had guest lecturer and researcher positions at the University of Girona, the Technical University of Vienna, and the University of Minnesota. His research area includes Monte-Carlo global illumination algorithms, distributed high-performance visualization, and GPU implementation. He published about two hundred papers, scripts and books on this topic. He obtained Eurographics Fellow title in 2008. Contact: szirmay@iit.bme.hu.

### Tamás Umenhoffer

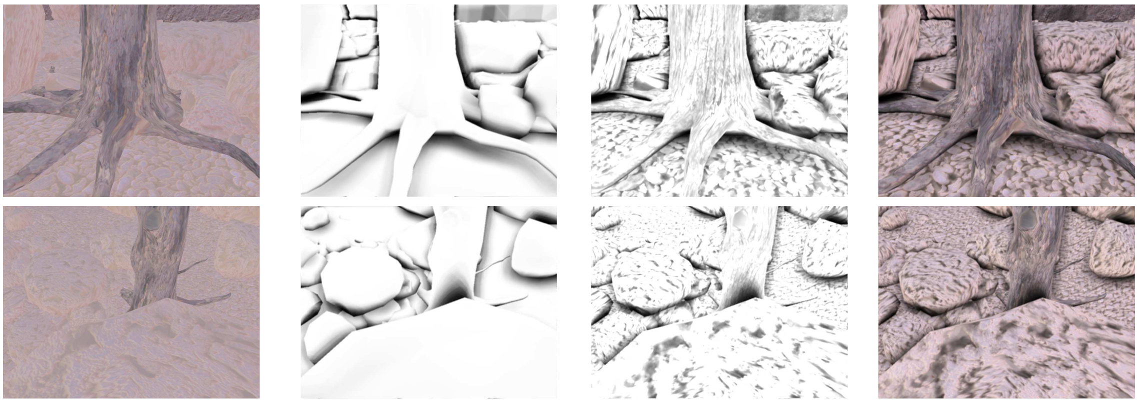
is an assistant professor at the Department of Control Engineering and Information Technology of Budapest University of Technology and Economics since 2007. His research focuses on games, realistic lighting effects, and medical visualization. He is about to defend his PhD dissertation on global illumination lighting in games. Contact: umitomi@gmail.com.

### Balázs Tóth

is an assistant professor at the Department of Control Engineering and Information Technology of Budapest University of Technology and Economics since 2008. His research area is screen space rendering methods, including tone mapping, depth of field, deferred shading, and ambient occlusion. Recently, he turned toward the application of GPUs in medical image reconstruction. Contact: tbalazs@sch.bme.hu.

### László Szécsi

is an assistant professor at the Department of Control Engineering and Information Technology of Budapest University of Technology and Economics since 2005, where he is responsible for game development courses. His research includes real-time Monte Carlo algorithms running both on the CPU and on the GPU. He will defend his PhD dissertation soon on the application of virtual point lights in real-time global illumination. Contact: szecsi@iit.bme.hu.



environment lighting only
115 FPS

z-buffer based occlusions
112 FPS

+ height-field occlusions
80 FPS

+ environment lighting
70 FPS

Fig. 12: A tree with stones rendered by volumetric ambient occlusion taking 12 samples per pixel. The scene is defined by 109784 triangles.

## Mateu Sbert

is a professor in computer science at the University of Girona, Spain. His research interests include application of Monte Carlo and Information Theory techniques to computer graphics and image processing. Sbert received an MS in theoretical physics from the University of Valencia, an MS in mathematics from U.N.E.D. (National Distance University of Spain), Madrid, and a PhD in computer science from the Technical University of Catalonia (best PhD award). Readers may contact Mateu Sbert at Institute of Informatics and Applications, Campus Montilivi, Edifici PIV, Univ. of Girona, E-17071 Girona, Spain; mateu@ima.udg.edu.