Eurographics Symposium on Rendering 2013
Nicolas Holzschuch and Szymon Rusinkiewicz
(Guest Editors)

Volume 32 (2013), Number 4

Probabilistic Visibility Evaluation for Direct Illumination

Niels Billen, Björn Engelen, Ares Lagae and Philip Dutré[†]

Department of Computer Science, KU Leuven, Belgium

Abstract

The efficient evaluation of visibility in a three-dimensional scene is a longstanding problem in computer graphics. Visibility evaluations come in many different forms: figuring out what object is visible in a pixel; determining whether a point is visible to a light source; or evaluating the mutual visibility between 2 surface points. This paper provides a new, experimental view on visibility, based on a probabilistic evaluation of the visibility function. Instead of checking the visibility against all possible intervening geometry, the visibility between 2 points is now evaluated by testing only a random subset of objects. The result is not a Boolean value that is either 0 or 1, but a numerical value that can even be negative. Because we use the visibility evaluation as part of the integrand in illumination computations, the probabilistic evaluation of visibility becomes part of the Monte Carlo procedure of estimating the illumination integral, and results in an unbiased computation of illumination values in the scene. Moreover, the number of intersections tests for any given ray is decreased, since only a random selection of geometric primitives is tested. Although probabilistic visibility is an experimental and new idea, we present a practical algorithm for direct illumination that uses the probabilistic nature of visibility evaluations.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

The evaluation of visibility in a three-dimensional scene has a long history in the field of computer graphics. Early algorithms focused mainly on the spatial relationship between geometric entities, and were closely related to techniques derived from descriptive geometry or technical drawings. This research led to many different techniques that involved line- and polygon clipping and point-in-polygon tests . With the introduction of the framebuffer and later the Z-buffer, visibility became discretized. Instead of performing continuous operations on lines and polygons, the focus now lay on operations in the image plane consisting of pixels. While many of the visibility operations were designed to determine what is visible as seen from the camera, the visibility problem using the position of the light sources (determining shadow regions) spawned its own collection of algorithms. Again, this led to continuous operations such as the computation of shadow volumes, shadow polygons, etc.; as well as to discretized data-structures such as the shadow map. A full survey of these techniques is outside the scope of this paper, but the interested reader can find a lot of information about the development of visibility calculations in general computer graphics books, e.g. [FvDFH95, AMHH08, ESAW11].

With the advent of ray tracing, two specific visibility evaluations became more prominent. For determining the visibility from the camera through a pixel, one is interested in the first visible surface point along a viewing ray. This type of visibility query is also useful when tracing recursive rays (reflective and refractive rays, or more generally, indirect rays in stochastic ray tracing). For shadow calculations, tracing a ray between a point to be shaded and a light source, a slightly different type of evaluation is needed. Now, one is not interested in the first intersection point encountered, but whether an intersection with surrounding geometry is present at all. The efficiency of both types of queries is speeded up significantly by the use of proper acceleration structures (e.g., [Wal07, WMG*08]). Most acceleration schemes follow the ray from start to end, moving through a number of spatial grid cells or hierarchical bounding boxes, each containing a number of geometric primitives. Once an

[†] e-mail: philip.dutre@cs.kuleuven.be

intersection is found, any primitives positioned farther along the ray do not need to be considered any further.

In the context of global illumination algorithms, the visibility evaluation between two points exchanging light energy (e.g. a point to be shaded and a point located at the surface of a light source), is written as part of the integrand of the illumination integral [DBB06, PH10]. These integrals are often evaluated using Monte Carlo integration, sampling a number of rays in the total integration domain [RAMN12]. Well-known and tested optimizations include importance sampling, in which either a BRDF-kernel or an incoming radiance distribution is sampled proportionally. The visibility function, however, is almost always evaluated exactly. Only a limited number of approaches have explore non-traditional evaluation of visibility [HDG99, DSDD07].

In this paper we introduce a new and experimental view on visibility evaluations, which considers the visibility estimation as a numerical part of the Monte Carlo estimation of the illumination integral. This stochastic evaluation of visibility is performed by testing only a random subset of the geometry likely to be located between two surface points. We will show that — perhaps somewhat counterintuitive — such an evaluation produces unbiased images. Since this technique reduces the number of intersection tests for each ray, efficiency can be gained. However, this process also introduces a new source of stochastic noise in the image. Therefore, a careful balance between noise and efficiency has to be found. By exploring this new approach on visibility, we enlarge the toolbox of methods available that can be used to evaluate visibility in graphics algorithms.

In summary, the contributions of this paper are:

- We develop a theoretical framework to evaluate visibility on stochastic subsets of geometric primitives. Our proposed approach considers visibility as a numerical function that can be sampled stochastically, and produces unbiased images in the context of global illumination algorithms.
- We propose a practical algorithm that uses the notion of stochastic visibility for computing direct illumination in a scene, based on the idea of an occlusion map. For each shadow ray to be processed, a random subset of the geometry most likely to intersect the shadow ray is tested for intersection. We show that we can reach the same image quality while performing fewer intersections tests
- We provide some additional theoretical insights in how the visibility function can be evaluated in the context of graphics algorithms.

2. Theoretical framework

The visibility function $V(x,y)$ in direct illumination algorithms is usually evaluated between 2 surface points x and y [DBB06]. If x and y are mutually visible, then $V(x,y) = 1$. Otherwise, $V(x,y) = 0$. In order to evaluate $V(x,y)$, all candidate geometric primitives that are located between x and y need to be checked for intersection. A proper acceleration structure can limit the list of possible intersecting geometry, or is able to process them in sorted order between x and y.

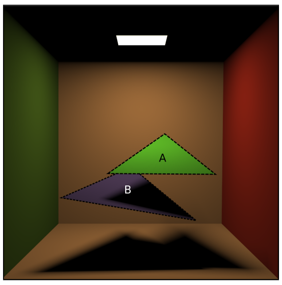


Figure 1: Box containing two candidate blockers A and B. Note that the walls, ceiling and floor are not considered for intersection testing.

The set of candidate geometric primitives that could intersect the line segment xy can be written as $Z = \{z_1, z_2, z_3, ..., z_n\}$. For now, we make abstraction of how this set is determined, and merely assume that the geometric primitives in Z are the ones that need to be tested explicitly for intersection. The total visibility $V(x,y)$ can then be written as the product of individual visibility evaluations:

$$V(x,y) = V_{z_1}(x,y) \cdot V_{z_2}(x,y) ... V_{z_n}(x,y) = \prod_{i=1}^{n} V_{z_i}(x,y) \quad (1)$$

$V_{z_i}(x,y)$ represents the visibility value (0 or 1) of the line segment xy, only taking into account the possible intersection with primitive z_i. The product represents the total visibility against the entire set of candidate blocking geometry. If at least one geometric primitive from the set results in a visibility value of 0, the total visibility equals 0.

It is important to realize that we will only concern ourselves with the numerical evaluation of visibility (0 or 1), as needed for direct illumination, and not with visibility in the sense of trying to determine what surface is visible along a given (viewing) ray. We will also refer to geometric primitives that need to be tested for intersection as candidate blockers or potential blockers.

2.1. Stochastic evaluation of visibility

To explain the core idea of probabilistic visibility, let us simplify Eqn. 1 by only considering two candidate intersecting blockers, A and B (Fig. 1). The visibility term $V(x,y)$ equals $V_A(x,y) \cdot V_B(x,y)$, in which V_A represents the visibility checking only against blocker A, and V_B checks only against blocker B.

The key insight of our procedure is to rewrite this product as a sum, and stochastically select only one of the terms of the sum to evaluate the overall function. A sum $S = s_1 + s_2 +$

Exact values			Stochastic evaluation			Var
V_A	V_B	$V_A.V_B$	$3V_A$	$3V_B$	$3(\overline{V_A}\,\overline{V_B}-1)$	-
0	0	0	0	0	0	0
0	1	0	0	3	-3	6
1	0	0	3	0	-3	6
1	1	1	3	3	-3	8

Table 1: Stochastic visibility evaluation for two polygons A and B. Each line shows one of 4 possible visibility configurations. Exact visibility values (leftmost columns), stochastic value for each of the 3 terms that can be selected (middle columns), variance of the stochastic process (right column).

$\ldots + s_n$ can be stochastically estimated by selecting a single term s_i with probability p_i. Provided all $p_i > 0$ and their sum equals 1, $\tilde{S} = s_i/p_i$ is then an unbiased estimator for S. One of the most simple decomposition of a product into a sum of terms is based on the algebraic product $ab = a + b + (1-a)(1-b) - 1$. Using complement notation ($\overline{a} = 1-a$), Eqn. 1 for 2 blockers is then written as:

$$\begin{aligned} V(x,y) &= V_A(x,y)\cdot V_B(x,y) \\ &= V_A(x,y) + V_B(x,y) + (\overline{V_A(x,y)}\cdot\overline{V_B(x,y)} - 1) \end{aligned} \quad (2)$$

By stochastically selecting one of the 3 terms, each with probability p_1, p_2 and p_3 ($p_1 + p_2 + p_3 = 1$), we obtain the following evaluation scheme:

$$\tilde{V}(x,y) = \begin{cases} \frac{V_A(x,y)}{p_1} & \text{with probability } p_1 \\ \frac{V_B(x,y)}{p_2} & \text{with probability } p_2 \\ \frac{\overline{V_A(x,y)}\cdot\overline{V_B(x,y)}-1}{p_3} & \text{with probability } p_3 \end{cases} \quad (3)$$

The numerical value of the total visibility $V(x,y)$ is therefore estimated, *in an unbiased manner*, by only intersecting with polygon A in the first case, only intersecting with polygon B in the second case, or intersecting *both* in the third case. Note that, depending on the choices of $p1$, $p2$ and $p3$, visibility values can be larger than 1. The third term can even become negative. As an example, assume $p1$, $p2$ and $p3$ are all equal to $1/3$, leading to the stochastic evaluations listed in Tab. 1. For each of 4 possible visibility configurations, the exact values are listed, along with the stochastic evaluation when each of the 3 possible terms would be selected. The last column lists the exact value of variance (computed analytically) of the stochastic process. It is straightforward to check that the expected value of this stochastic process yields the correct visibility value in each of the 4 visibility configurations. It might be counterintuitive to deal with visibility values different from 0 or 1, but since we are interpreting visibility as a numerical value rather than a geometric property, this is not unsurprising.

This numerical interpretation of visibility can be exploited when including $V(x,y)$ in the illumination integral for direct illumination calculations [DBB06]. Radiance L at a surface point x is written as (without loss of generality, we assume a diffuse surface):

$$L(x) = \int_S f_r(x) L(y \to x) V(x,y) G(x,y) dS_y \quad (4)$$

with S the area of the light source, y being a surface point on the light source, f_r the BRDF, and $G(x,y)$ the geometric coupling term containing two cosine factors and the inverse of the distance squared between x and y. Splitting the visibility term as above:

$$\begin{aligned} L(x) = &\int_S f_r(x) L(y \to x) V_A(x,y) G(x,y) dS_y \\ &+ \int_S f_r(x) L(y \to x) V_B(x,y) G(x,y) dS_y \\ &+ \int_S f_r(x) L(y \to x)(\overline{V_A(x,y)}\cdot\overline{V_B(x,y)} - 1) G(x,y) dS_y \end{aligned} \quad (5)$$

The evaluation of $L(x)$ can now be considered as evaluating the above sum of integrals, by stochastically selecting only one of the 3 terms for each shadow ray cast towards the light source, resulting in an unbiased estimator for $L(x)$.

The first row of images shown in Fig. 2 are rendered using this stochastic evaluation of visibility, for a gradually increasing number of shadow rays per pixel. Notice that the area where the shadows of both polygons overlap is rendered correctly irrespective of the number of shadow rays. This is also obvious from Tab. 1, since in this case, the numerical variance equals 0. The areas in which only a single shadow is present do contain noise, since for one-third of the shadow rays, the relevant polygon will not be checked for visibility. The areas without any shadows at all do exhibit dark noise as well, since their visibility values are the result of adding positive and negative values to converge to an average value of 1 (see Tab. 1).

The overall advantage is that for each shadow ray, on average, only $(1+1+2)/3 = 1.33$ polygons need to be checked for intersection, versus 2 polygons for a regular visibility evaluation. Although we expect an increase in image noise, there is room for a trade-off between speed (number of intersections per shadow ray) and image quality (noise).

2.2. Alternative decompositions

We can expect that other decompositions of the visibility product into a sum can lead to less noise in the final direct illumination image. An interesting choice is to distribute the -1 term evenly over all 3 visibility terms. With p_1, p_2 and p_3 again all equal to $1/3$, the different possibilities are listed in Tab. 2.

Again, it is easy to check that the expected value in each of the 4 configurations equals the correct visibility value. Although the variance generally is lower in value compared to the previous evaluation in Tab. 1, we have introduced variance for areas which are covered by both shadows, and lowered the variance in completely visible areas or areas only covered by one of both shadows. This is an expected result,

Exact values			Stochastic evaluation			Var
V_A	V_B	$V_A.V_B$	$3V_A - 1$	$3V_B - 1$	$3\overline{V_A V_B} - 1$	-
0	0	0	-1	-1	2	2
0	1	0	-1	2	-1	2
1	0	0	2	-1	-1	2
1	1	1	2	2	-1	2

Table 2: Stochastic visibility evaluation for two polygons A and B with equal distribution of the -1 term, each line showing one of 4 possible visibility configurations. Exact visibility values (leftmost columns), stochastic value for each of the 3 terms that can be selected (middle columns), variance of the stochastic process (right column).

Exact values			Stochastic evaluation			Var
V_A	V_B	$V_A.V_B$	$-\frac{3V_A}{254}$	$-\frac{3V_B}{254}$	$\frac{3(V_A+V_B)^8}{254}$	-
0	0	0	0	0	0	0
0	1	0	0	-3/254	3/254	4.65e-5
1	0	0	-3/254	0	3/254	4.65e-5
1	1	1	-3/254	-3/254	768/254	2.048

Table 3: Stochastic visibility evaluation for two polygons A and B using binomial decomposition, each line showing one of 4 possible visibility configurations. Exact visibility values (leftmost columns), stochastic value for each of the 3 terms that can be selected (middle columns), variance of the stochastic process (right column).

since we are effectively always evaluating the constant -1 term present in the sum decomposition, whereas before, we only evaluated it for $1/3$ of the shadow rays. Fig. 2, second row, shows images rendered with this decomposition. We see noise in the completely shadowed area of the image, but compared to the previous case, visible noise in other areas of the image has decreased. One can observe an overall faster convergence towards the exact solution.

We also experimented with a decomposition of the visibility function based on the binomial theorem. Since V_A and V_B equal either 0 or 1, we can write:

$$\begin{aligned}(V_A(x,y)+V_B(x,y))^n =& V_A(x,y)+V_B(x,y)\\ &+(2^n-2)V_A(x,y)V_B(x,y)\end{aligned} \tag{6}$$

and thus:

$$\begin{aligned}V(x,y) =& -\frac{V_A(x,y)}{2^n-2}-\frac{V_B(x,y)}{2^n-2}\\ &+\frac{(V_A(x,y)+V_B(x,y))^n}{2^n-2}\end{aligned} \tag{7}$$

Choosing $n = 8$, and selecting each of 3 possible terms with probability $1/3$ provides the values listed in Tab. 3. The variance all shadowed regions is now very small, but variance is present in completely illuminated regions.

The resulting images are shown in the third row of Fig. 2. We have no variance in areas where both polygons cast a shadow. Noise in shadowed areas caused by a single polygon is far less as in both previous cases, as is predicted by the variance values in Tab. 3. The illuminated areas though, show quite some noise for a low number of shadow rays. In this case, a pixel might end up will a small negative visibility value (subsequently capped to 0). Experimenting with larger values for n in the binomial decomposition did not produce significantly different results.

2.3. Further discussion

Multiple blockers. So far, we have considered the case for 2 blocking polygons. The theory still holds when we have a larger set of potentially intersecting geometry. By subdividing the set in 2 groups A and B, the above equations still hold. A visibility evaluation for a group then involves explicitly checking all polygons in that group for intersection. A recursive further subdivision of a selected group can also be considered, but various tests showed that this option increases noise significantly in the image. This is not surprising, since reducing visibility tests for a large set of potential blockers to only a single test vs. a single blocker, is unlikely to provide accurate (although unbiased) results. For a numerical insight, consider the values in Tab. 1. Stochastic visibility values for a single shadow ray can be either 0, 3 or -3. Averaging a large number of samples either produces a value of 0 or 1. If groups are recursively subdivided, the stochastic visibility values are multiplied by an additional factor of 3 for each recursive subdivision, resulting in very large positive and negative values, that still need to average to 0 or 1. Such a process with increasingly larger values inherently increases variance. This increase in variance is also noticeable when we use an acceleration structure such as a regular grid to process the shadow ray. Each cell the ray passes through contains a number of polygons, whose visibility is evaluated using the procedures above. However, by stacking up all stochastic evaluations for all cells the ray passes through, we reach visibility estimates which have high absolute values, but still need to average out to 0 or 1.

Negative visibility values. It is possible that due to negative visibility values, the total illumination integral for a surface point evaluates to a negative value, especially for points for which the exact visibility value equals 0. Although the variance decreases with a growing number of shadow rays, there still is a significant probability that the average visibility value will not end up being exactly 0 due to the averaging of positive and negative evaluations. In our implementation, we clamp all pixels with negative values to 0. This makes the image a little bit too bright on average, but this can be corrected for by accumulating all negative light in pixels and distributing it equally over (neighboring) pixels. This is a similar approach as used in the redistribution of unshot energy in radiosity algorithms [CCWG88].

Similarity to BRDF decomposition. The decomposition of the visibility term in a sum of terms, thereby splitting the illumination integral in a sum of integrals (Eqn. 5), is quite similar to the evaluation of the indirect illumination integral

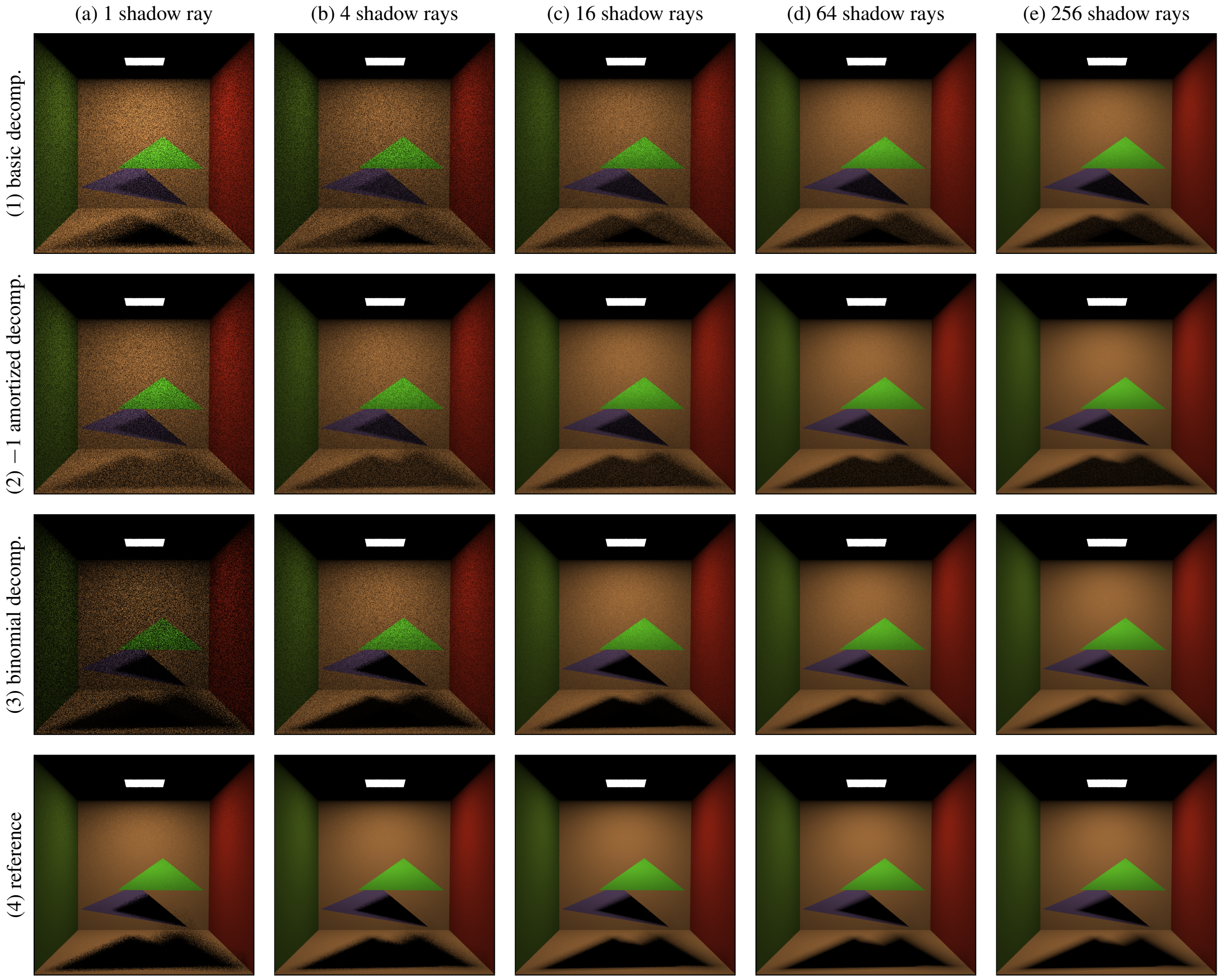


Figure 2: Stochastic visibility evaluation for different decomposition of the visibility function (rows 1-3) and for a gradually increasing number of shadow rays per pixel (columns a-e) for the scene shown in Fig. 1. A reference image using deterministic visibility evaluation is shown in row 4.

in global illumination, in which the BRDF is split in a sum of a diffuse and a specular term. An indirect illumination ray is then chosen with discrete probability as being reflected either through the diffuse or the specular part of the BRDF, and the resulting evaluation is an estimate for the reflection behavior of the complete BRDF.

3. Practical algorithm using occlusion mapping

3.1. Design decisions and algorithm description

To use probabilistic visibility in a direct illumination algorithm, we need to have knowledge of the potential blockers between a surface point to be shaded and the light source(s) in the scene. We need to consider the entire group of blockers that can possibly intersect the ray as a whole, and split this group in two subgroups. Ideally, we would like to use an acceleration structure that quickly selects potential blockers based on their geometric proximity along the entire length of the shadow ray. Traditional acceleration structures such as spatial grids or hierarchical bounding volumes are not immediately suitable since they are specialized in finding the first blocker along a ray by proceeding through a number of spatial cells the ray passes through. As mentioned before, splitting up the visibility evaluation by subsequently evaluating visibility one cell at a time, yields a negative effect on the variance of the visibility evaluation.

For our purposes, directional structures such as a light-buffer [HG86] or a 5D ray hierarchy [AK87] are much better suited. These structures are able to select all blockers located in a narrow directional shaft around the entire length of the ray. When processing a shadow ray, we can then retrieve the blockers that have an effect on the total visibility function

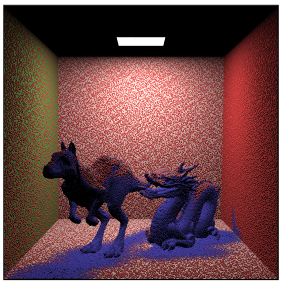


Figure 3: Location of light photons (red) and occlusion photons (blue) for the scene shown in Fig. 7(b).

for the ray, and proceed with that group for our stochastic evaluation.

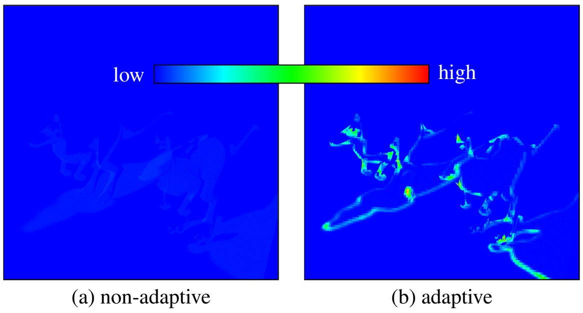
As a directional structure that selects blockers based on ray direction, we employ an *occlusion map*. Our occlusion map is closely related to shadow photons, introduced in the context of photon mapping [Jen01]. The occlusion map stores information about all blockers located between an occlusion photon (a surface point) and the light source. The location of occlusion photons tells us something about the location of shadows in the scene, as well as about the geometry casting these shadows. We will use the occlusion map to discriminate between umbra, penumbra, and illuminated regions. We will only use probabilistic visibility for the penumbra regions.

Our direct illumination algorithm is composed of the following steps:

1. Construction of the occlusion map, by storing information at surface points (*occlusion photons*) regarding blockers between that point and the light source (Sec. 3.2). If no blockers are present, we consider the point as a *light photon*.
2. During rendering, for each pixel a viewing ray is traced through the scene. A lookup in the occlusion map retrieves nearby occlusion and light photons for the visible point through the pixel (Sec. 3.3). The following possibilities occur:

   a. No occlusion photons are found: the light source is deemed to be completely visible and the illumination will be evaluated analytically.
   b. Only occlusion photons are found: the point to be shaded is considered to be located in the umbra region and nu further illumination computations are made.
   c. A mix of occlusion photons and light photons is detected: the point is likely to be situated in a penumbra region w.r.t. the light source. The potential blockers from the occlusion photons are gathered and split in two groups A and B for probabilistic visibility evaluation.

We will now describe each of these steps in more detail.



(a) non-adaptive (b) adaptive

Figure 4: Occlusion photon density for (a) non-adaptive and (b) adaptive generation of occlusion photons for the scene shown in Fig. 7(a).

3.2. Construction of the occlusion map

We need to be able to quickly identify the set of candidate blockers between a point to be shaded and a light source. Besides creating light photons in the regions illuminated by a light source, we also create occlusion photons in the regions invisible to the light source. These occlusion photons store the set of all occluders between the position of the occlusion photon and the light source. Since occlusion photons are only present in shadowed regions, the location of occlusion photons is an indication of the presence of a shadow in that particular part of the scene.

The occlusion map can be constructed in a couple of different ways. A straightforward approach distributes photons from the light source, and a light photon is created at the first intersection point visible from the light source. The photon ray is then traced further through intersecting geometry and occlusion photons are created at all subsequent intersection points. All previously intersected geometric elements are stored with the occlusion photon. Since we will only trace shadow rays in the penumbra regions where a mix of light photons and occlusion photons are detected, we would like to distribute more occlusion photons in these penumbra regions. This is difficult to achieve while distributing the photons from the light source.

We therefore opted for a camera-driven generation of occlusion photons. A first batch of viewing rays (we used 200K rays for our images) is uniformly distributed over the image plane. A shadow ray is traced from each of those 200K intersection points. If the light source is reached unhindered, a light photon is created. Otherwise, an occlusion photon which stores all the blockers encountered along the shadow ray is stored in the occlusion map (see Fig. 3). For each group of 8x8 pixels in the image plane, the number of occlusion and light photons are compared. Subsequent batches of viewing rays (again, 200K each), are distributed according to a density distribution that favors previously detected penumbra regions (see Fig. 4). The net result is that relatively more occlusion photons are generated in penumbra regions.

3.3. Rendering phase

The rendering phase generates a viewing ray through each pixel, finding the first visible point x. In order to compute the illumination at x, the corresponding occlusion map is queried to locate the nearest 100 occlusion photons within a maximum set distance. As explained before, the following cases are considered:

- If no occlusion photons are found, x is assumed to be completely illuminated by the light source, and the illumination is computed analytically [Arv95]. This currently limits our technique to diffuse surfaces only.
- If occlusion photons are detected, we search for the closest 100 light photons, again within a set maximum distance. If no light photons are present, we assume x is in the umbra region, and no further computations are made. Otherwise, we assume x is located within the penumbra.

In the last case, we collect all unique blockers that are stored in the 100 closest occlusion photons. These blockers are considered the most likely candidates to intersect the shadow ray. Note that this set is not conservative, i.e. it is possible that a blocker intersecting a shadow ray starting at x will not be present in this set. This can happen if this particular blocker was not detected during the construction of the occlusion map. Especially with very small blockers relative to the density of the occlusion map, this might be a problem.

We subsequently subdivide the blockers in two groups, and then proceed with probabilistic visibility evaluation using Eqn. 7. To subdivide all blockers in 2 groups A and B, they are sorted according to their subtended solid angle as seen from x. Blockers are subsequently put into one of both groups, keeping the summed solid angle of both groups roughly equal. The rationale behind this split is that we will initially attribute equal probabilities to A and B during probabilistic visibility evaluations. If they both subtend equal solid angle, chances are good they both are as likely to intersect the shadow ray.

Note that the occlusion map can also be used to evaluate visibility deterministically by intersecting *all* blockers stored in the occlusion photons. In our algorithm, we merely employ the occlusion map to quickly select a number of candidate blockers to serve as input for the probabilistic visibility evaluation. Discussing advantages and limitations of the occlusion map without probabilistic visibility evaluation falls outside the scope of this paper.

3.4. Efficiency and implementation issues

Importance sampling of visibility terms. To improve performance, importance sampling for the three different visibility terms of Eqn. 7 is implemented (see Fig. 5). Initially, all probabilities are set to $1/3$. After the first group of 64 shadow rays is evaluated, the probabilities are refined based on the hit ratios of the shadow rays versus groups A and B. This allows us to assign probabilities to the different visibility terms proportional to their actual numerical values, and therefore lower the variance of the stochastic visibility evaluations.

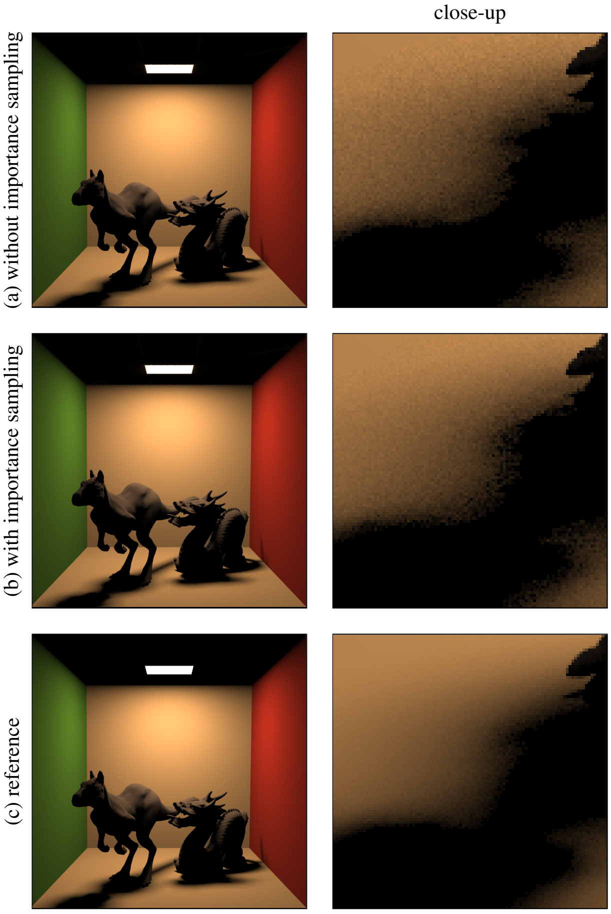
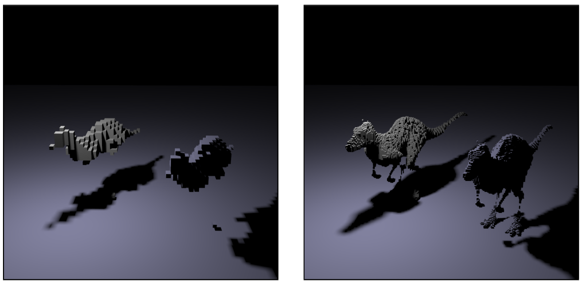


Figure 5: Importance sampling of visibility terms. Stochastic visibility evaluation (a) without and (b) with importance sampling of visibility terms for the scene shown in Fig. 7(b). A reference using deterministic visibility evaluation (c) and close-ups are also shown.

Volumetric occluders. Probabilistic visibility performs better if the selection of candidate blockers is accurate. For scenes with a large number of small polygons, such small polygons might be missed entirely in the occlusion map, and therefore not show up in the visibility evaluations. Volumetric occluders can be used to fill up solid objects with a solid octree [DKH09] (Fig. 6). If these occluders are also stored as part of the occlusion map, the chance of missing visibility events is largely reduced.



(a) depth 4

(b) depth 6

Figure 6: Visualization of the volumetric occluder tree at (a) depth 4 and (b) depth 6 for the scene shown in Fig. 7(a). Volumetric occluders reduce missed visibility events and enhance the reliability of probabilistic visibility.

4. Discussion

4.1. Results

We evaluated stochastic visibility evaluation with occlusion mapping on different scenes (Fig. 7). To test the limits of our algorithm, we included some test scenes with very challenging penumbra shadow patterns (e.g. a Menger grid and the Yeah Right model). For solid objects, volumetric occluders were used as well.

Tab. 4 illustrates the performance of our algorithm for the scenes in Fig. 7. Both execution times and number of intersection tests are listed. Since we want to investigate the isolated effect of probabilistic visibility, we compared two rendering approaches, both using the occlusion map, but with and without the use of probabilistic visibility. Our stochastic visibility estimation reduces the number of intersections for an equal number of shadow rays on average by 22% while decreasing the rendering time on average by 14%. This is due to the probabilistic evaluation of visibility in detected penumbra regions. The corresponding images indicate equal visual quality.

When comparing reference images with images rendered with the occlusion map, we notice that shadow regions tend to become smaller. This is due to the difficulty of finding all potential blockers in the penumbra regions. This is clearly seen in the scene with the Yeah Right model where very fine shadows tend to disappear. However, regions which are categorized as either in umbra or completely illuminated are always fully converged due to analytic integration of the illumination.

4.2. Limitations

The proposed algorithm using the occlusion map and the probabilistic evaluation of visibility has a number of limitations:

- With an exact evaluation of the visibility, noise due to visibility is only present in penumbra regions. Probabilistic visibility introduces noise in umbra and illuminated regions as well. However, the number of intersections per shadow ray is reduced. Further research might be able to exploit this trade-off between speed and quality in a more fundamental way.
- The occlusion map is an acceleration structure designed to quickly select potential blockers for a given surface point. The occlusion map itself is constructed stochastically, hence introducing some additional sources of error (small polygons might be missed completely). The use of volumetric occluder alleviates these missed visibility events.
- Especially for large amounts of geometry present in the scene, it remains doubtful whether probabilistic visibility can compare favorably with more traditional, often very efficient, accelerations structures. Hierarchical grids are able to find an intersection very quickly, often after only a handful of intersection tests. Future work on the use and exploitation of probabilistic visibility in conjunction with acceleration structures might therefore be interesting.

5. Conclusion

We have presented a new and experimental view on the evaluation of visibility, based on a probabilistic evaluation of various visibility terms. By integrating this numerical evaluation into the illumination integral for direct illumination, we are able to generate unbiased images. Although the number of intersection tests per shadow ray are reduced, additional noise is introduced into the image. A careful balance is therefore needed between computation time and image quality.

A practical algorithm using stochastic visibility should use a directional acceleration scheme, since potential occluders located close to the shadow ray should be selected quickly. We have shown that using an occlusion map, encoding visibility events in a large set of occlusion photons, is suited for this task. Using various optimizations, a competitive algorithm has been constructed.

We hope that by having presented a novel evaluation of the visibility function, new types of visibility algorithms can be constructed, and additional new insights may be gained by using probabilistic visibility.

Acknowledgements

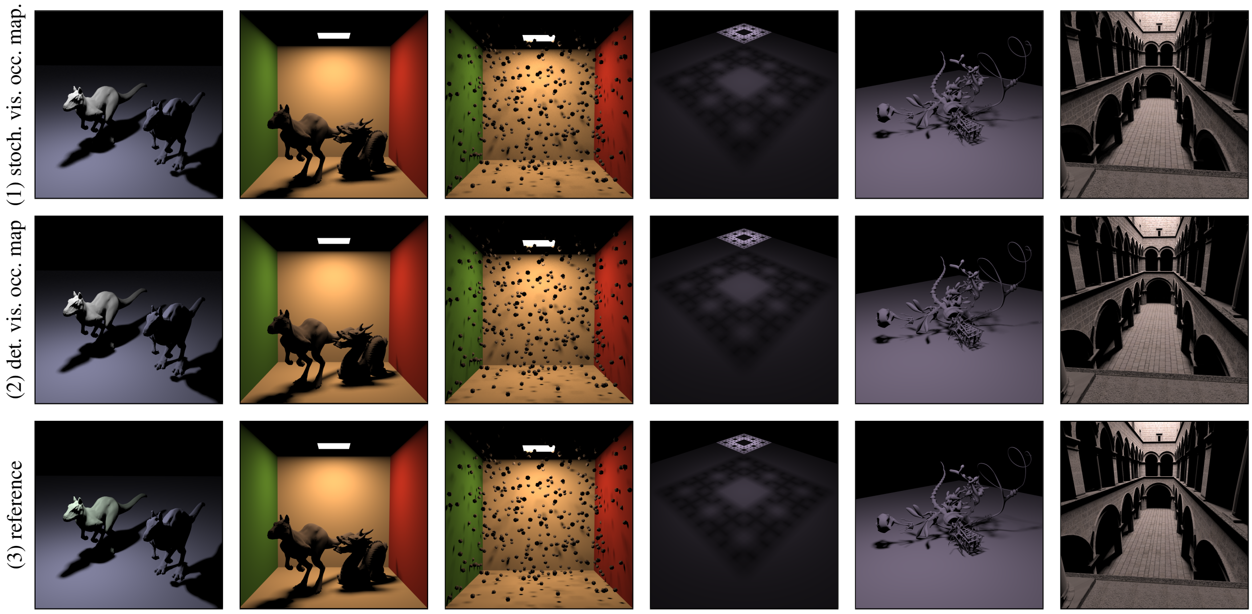
The scenes used in this paper are courtesy of PBRT [PH10]. We would like to thank the anonymous reviewers. Ares Lagae is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO).

References

[AK87] Arvo J., Kirk D.: Fast ray tracing by ray classification. In *Computer Graphics* (1987), vol. 21, pp. 55–64. 5

scene	time (in sec)		intersections (in millions)	
	occlusion map	prob.vis.	occlusion map	prob.vis.
Killeroo	306s	262s (-14.4%)	27494M	21047M (-23.4%)
Cornell w. Dragon and Killeroo	990s	827s (-16.4%)	96453M	76283M (-20.9%)
Icosahedrons	714s	533s (-25.4%)	61561	37913 (-38.2%)
Menger cube	637s	578s (-9.3%)	63470M	51927M (-18.2%)
Yeah Right	663s	560s (-15.5%)	70392M	55203M (-21.5%)
Sponza	628s	588s (-6.3%)	58449M	44867M (-23.2%)

Table 4: Performance of stochastic visibility evaluation.



(a) Killeroos (b) Cornell w. Dragon (c) Icosahedron (d) Menger (e) YeahRight (f) Sponza

Figure 7: Stochastic visibility renderings with the occlusion map for different scenes (a-f). For comparison purposes, deterministic visibility evaluation using the occlusion map (row 2) and a reference image using deterministic visibility evaluation (row 3) are also shown.

[AMHH08] Akenine-Möller T., Haines E., Hoffman N.: *Real-Time Rendering 3rd Edition*, 3rd ed. A K Peters/CRC Press, 2008. 1

[Arv95] Arvo J.: *Analytic methods for simulated light transport*. PhD thesis, Yale University, 1995. 7

[CCWG88] Cohen M. F., Chen S. E., Wallace J. R., Greenberg D. P.: A progressive refinement approach to fast radiosity image generation. In *Computer Graphics* (1988), vol. 22, pp. 75–84. 4

[DBB06] Dutré P., Bala K., Bekaert P.: *Advanced Global Illumination*, 2nd ed. A K Peters/CRC Press, 2006. 2, 3

[DKH09] Djeu P., Keely S., Hunt W.: Accelerating shadow rays using volumetric occluders and modified kd-tree traversal. In *Proceedings of the Conference on High Performance Graphics 2009* (2009), pp. 69–76. 7

[DSDD07] Dachsbacher C., Stamminger M., Drettakis G., Durand F.: Implicit visibility and antiradiance for interactive global illumination. *ACM Transactions on Graphics 26*, 3 (2007), 61:1–61:10. 2

[ESAW11] Eisemann E., Schwarz M., Assarsson U., Wimmer M.: *Real-Time Shadows*. A K Peters/CRC Press, 2011. 1

[FvDFH95] Foley J. D., van Dam A., Feiner S. K., Hughes J. F.: *Computer graphics: principles and practice*, 2nd ed. Addison-Wesley Professional, 1995. 1

[HDG99] Hart D., Dutré P., Greenberg D. P.: Direct illumination with lazy visibility evaluation. In *Proceedings of SIGGRAPH 99* (1999), pp. 147–154. 2

[HG86] Haines E. A., Greenberg D. P.: The light buffer: A shadow-testing accelerator. *IEEE Computer Graphics and Applications 6*, 9 (1986), 6–16. 5

[Jen01] Jensen H. W.: *Realistic image synthesis using photon mapping*, 2nd revised ed. A K Peters/CRC Press, 2001. 6

[PH10] Pharr M., Humphreys G.: *Physically Based Rendering*, 2nd ed. Morgan Kaufmann, 2010. 2, 8

[RAMN12] Ramamoorthi R., Anderson J., Meyer M., Nowrouzezahrai D.: A theory of monte carlo visibility sampling. *ACM Transactions on Graphics 31*, 5 (2012), 121:1–121:16. 2

[Wal07] Wald I.: On fast construction of sah-based bounding volume hierarchies. In *Proceedings of the 2007 IEEE Symposium on Interactive Ray Tracing* (2007), pp. 33–40. 1

[WMG*08] Wald I., Mark W. R., Günther J., Boulos S., Ize T., Hunt W., Parker S. G., Shirley P.: State of the art in ray tracing animated scenes. *Computer Graphics Forum 28*, 6 (2008), 1691–1722. 1