Eurographics Symposium on Rendering 2014
Wojciech Jarosz and Pieter Peers
(Guest Editors)

Volume 33 (2014), Number 4

Probabilistic visibility evaluation using geometry proxies

Niels Billen, Ares Lagae, Philip Dutré

Department of Computer Science, KU Leuven, Belgium

Abstract

Evaluating the visibility between two points is a fundamental problem for ray-tracing and path-tracing algorithms. Ideally, visibility computations are organized such that a minimum number of geometric primitives need to be checked for each ray. Replacing complex geometric shapes by a simpler set of primitives is one strategy to control the amount of intersection calculations. However, approximating the original geometry introduces inaccuracies in e.g. shadow regions when shadow rays are intersected with the approximate geometry. This paper presents a theoretical framework for probabilistic visibility evaluation. When intersecting a shadow ray with the scene, we randomly select the original geometry, the approximated geometry, or one of several correction terms, to be tested. Not all shadow rays will therefore intersect the original geometry, but our method is able to produce unbiased images that converge to the correct solution. Although probabilistic visibility evaluation is an experimental idea, we show several example scenes that highlight the potential for future improvements.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Three-Dimensional Graphics and Realism - Raytracing

1. Introduction

Visibility evaluation along a ray is a common operation in many rendering algorithms. We can consider two types of visibility queries: finding the first surface point along a directed line (so-called *first-hit* visibility), or figuring out whether two points are mutually visible to each other (so called *any-hit* visibility). In this paper, we will focus specifically on evaluating the *any-hit* visibility of shadow rays between surface points in the scene and surface points at a light source.

A naïve ray tracing algorithm needs to test all the geometric primitives for intersection for each ray. This evaluation can be accelerated significantly by using an appropriate acceleration structure. Most acceleration structures follow the ray from start to end, moving through a spatial grid or a hierarchy of bounding boxes. Once an intersection is found, the primitives positioned further along the ray can be ignored.

Visibility evaluations can also be simplified by testing against a geometry proxy, a less complex version of the original geometry. Intersecting a geometry proxy instead of the original will introduce errors, which become more pronounced when the geometry proxy becomes less representative. However, there is a potential for gains in efficiency, since fewer geometric primitives need to be processed.

In this paper we introduce a new probabilistic visibility algorithm which estimates the visibility using geometric proxies, but is still able to provide unbiased visibility estimates.

The contributions of this paper are the following:

- We develop a theoretical framework that allows for unbiased shadow computations while testing geometry proxies.
- We present several special cases where the proxies bound the original geometry or are completely contained in the original geometry.
- We evaluate the performance of our stochastic algorithm for a number of scenes of varying complexity.

2. Related work

Acceleration structures. A lot of research towards visibility evaluation has focused on fast acceleration structures (see [WMG*09] for a survey). These can be classified roughly in adaptive and non-adaptive structures. A regular grid is a non-adaptive acceleration structures which divides the three dimensional space into cells, where each cell only contains a small number of primitives [FCK*88], [LD08]. Rays are traversed front-to-end, stepping from cell to cell, only testing the contents of each cell intersected by the ray.

Adaptive acceleration structures, such as Bounding Volume Hierarchies [Wal07] and kD-trees [WMS06] follow a divide-and-conquer approach, where each node in the hierarchy divides either the objects or the three-dimensional space in two partitions. During traversal, we start in the root node and only descend in a child node if the ray passes through it, achieving logarithmic time efficiency.

Approximate geometry. Several methods exist for generating approximate geometry for a complex model [CMS98], [HG97]. The most common simplification algorithms remove or cluster vertices based on a quality heuristic. A complex model can also be approximated by a set of textured planes [DDSD03], which allows for extreme simplifications while maintaining good quality. Silvennoinen et al. [SSLL14] propose a new algorithm which converts a complex model to a triangle soup while maintaining its occlusion properties.

Approximate visibility. Several techniques exist for approximating visibility values. Lacewell et al. prefilter the occlusion of aggregate geometry and store in each node of the acceleration structure the opacity of its content [LBBS08]. The traversal of the acceleration structure for shadow rays can be aborted early, returning the prefiltered opacity of the node. Imperfect Shadow Maps (ISMs) create several low resolution shadow maps for a sparse point sampled representation of the scene [RGK*08], allowing realtime calculation of indirect shadows. While ISMs can handle low-frequency indirect shadows well, they have difficulty dealing with high-frequency shadows. While our technique uses approximate geometry, our resulting images are unbiased and converge to the exact solution.

Stochastic visibility. A novel view on visibility was presented by [BELD13]. Potential occluders between light source and points to be shaded are split in two separate groups and visibility is evaluated by either testing the first, second or both groups of primitives. The number of intersection tests is reduced on average, since testing a single group requires less intersection tests than testing both groups. In contrast to [BELD13], our method introduces geometric proxies in the scene and we stochastically select either the geometric proxies or a special correction term which keeps the visibility estimation unbiased.

3. Theoretical framework

The visibility function $V(x,y)$ in direct illumination algorithms is usually evaluated between two surface points x and y. When both points are mutually visible, $V(x,y) = 1$, otherwise $V(x,y) = 0$. To evaluate the visibility function, the set of geometric primitives $\mathcal{P} = \{p_1, p_2, \dots p_n\}$ between x

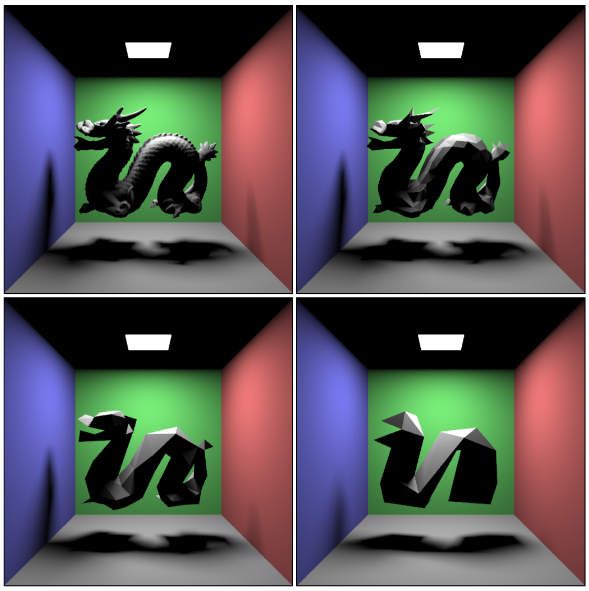


Figure 1: *Approximations of the Stanford Dragon using a quadric edge collapse decimation [HG97]. The original model (top left) consists of 871414 triangles. The approximation in the top right, bottom left and bottom right consist of 1024, 256 and 64 triangles respectively. Note that we are only concerned about shadow cast by the dragon. The actual geometry simplification is only visualized for illustrative purposes.*

and y need to be tested for intersection with the line segment $\overline{xy}$ until an intersection is found.

We consider $V_{p_i}(x,y)$ as the visibility with respect to a single primitive p_i:

$$V_{p_i}(x,y) = \begin{cases} 0 & \text{if } \overline{xy} \text{ intersects primitive } p_i \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

The visibility function for a set of n geometric primitives can be expressed as the product of the visibilities of the individual primitives. The visibility $V(x,y)$ with respect to the set of geometric primitives $\mathcal{P} = \{p_1, p_2, \dots p_n\}$ can be written as:

$$V_{\mathcal{P}}(x,y) = V_{p_1}(x,y) \cdot V_{p_2}(x,y) \cdot \ldots \cdot V_{p_n}(x,y) \tag{2}$$

The exact visibility $V_{\mathcal{P}}(x,y)$ can be approximated by a different set of primitives $\mathcal{P}' = \{p'_1, p'_2, \dots, p'_m\}$:

$$V_{\mathcal{P}'}(x,y) = V_{p'_1}(x,y) \cdot V_{p'_2}(x,y) \cdot \ldots \cdot V_{p'_m}(x,y) \tag{3}$$

Using $V_{\mathcal{P}'}(x,y)$ instead of $V_{\mathcal{P}}(x,y)$ offers potential gains in efficiency, if a set of primitives $\mathcal{P}'$ is used for which the visibility can be evaluated more efficiently. However, using the approximate visibility $V_{\mathcal{P}'}(x,y)$ introduces errors in the resulting image, which become more pronounced when the discrepancy between $\mathcal{P}$ and $\mathcal{P}'$ becomes larger (see Figure 1).

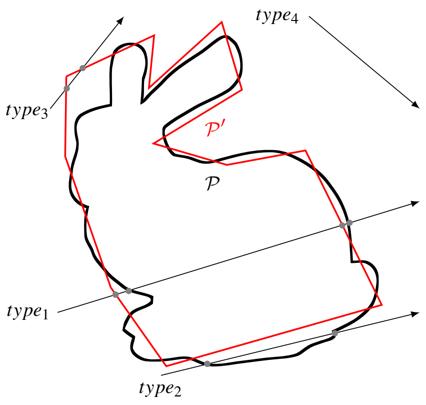


Figure 2: *$Type_1$ rays hit both model ($\mathcal{P}$) and the proxy ($\mathcal{P}'$). Rays of $type_2$ hit the model, but miss the proxy. $Type_3$ rays miss the model, but hit the proxy. Rays of $type_4$ miss both the proxy and the original model.*

Ray type	$V_{\mathcal{P}}(x,y)$	$V_{\mathcal{P}'}(x,y)$	$c(x,y)$
$type_1$	0	0	0
$type_2$	0	1	-1
$type_3$	1	0	1
$type_4$	1	1	0

Table 1: *Visibility values for rays of each type as shown in Figure 2.*

The exact visibility can be expressed as the sum of the approximate visibility and a correction term:

$$V_{\mathcal{P}}(x,y) = V_{\mathcal{P}'}(x,y) + c(x,y) \tag{4}$$

To find an expression for $c(x,y)$, we inspect the different types of rays which intersect $\mathcal{P}$ and $\mathcal{P}'$. Four types of rays can be distinguished, illustrated in Figure 2. Each ray type corresponds to a visibility configuration as summarized in Table 1. The approximate visibility $V_{\mathcal{P}'}(x,y)$ differs from $V_{\mathcal{P}}(x,y)$ for rays of type $type_2$ and $type_3$. The visibility discrepancy for rays of $type_2$ occur due to under-occlusion. The approximate primitives do not block all the rays would be blocked by the original primitives. The errors for rays of $type_3$ occur due to over-occlusion, caused by the approximate primitives which block more rays than the original primitives.

To correct for under- and overocclusion, we subtract and add the visibility configuration of $type_2$ and $type_3$ rays from the approximate visibility:

$$\begin{aligned} V_{\mathcal{P}}(x,y) = V_{\mathcal{P}'}(x,y) \\ + V_{\mathcal{P}}(x,y)\,\overline{V_{\mathcal{P}'}(x,y)} \\ - V_{\mathcal{P}'}(x,y)\,\overline{V_{\mathcal{P}}(x,y)} \end{aligned} \tag{5}$$

where $\overline{a}$ indicates the complementary boolean value of a.

This equation can also be derived from set theory by the following observation:

$$\begin{aligned} &\{\text{rays that hit } \mathcal{P}\} \cup \{\text{rays that hit } \mathcal{P}' \text{ but not } \mathcal{P}\} \\ &= \{\text{rays that hit } \mathcal{P}'\} \cup \{\text{rays that hit } \mathcal{P} \text{ but not } \mathcal{P}'\} \end{aligned} \tag{6}$$

By substituting these sets of rays with the appropriate boolean expressions, one can deduce Equation 5.

3.1. Stochastic visibility evaluation

In this section we will review the theory on how to evaluate Equation 5 stochastically, following the same approach as [BELD13]. Any sum $S = s_1 + s_2 + \ldots + s_n$ can be estimated by selecting a single term s_i and dividing by the probability p_i of selecting that term. The value $\tilde{S} = s_i / p_i$ is an unbiased estimator of the sum S when the probability $p_i > 0$ for every term s_i, as can be seen from the following equation:

$$E\left[\tilde{S}\right] = p_1 \cdot \frac{s_1}{p_1} + p_2 \cdot \frac{s_2}{p_2} + \ldots + p_n \cdot \frac{s_n}{p_n} = S \tag{7}$$

We can apply this theory to stochastically evaluate Equation 5 to obtain the following unbiased estimator for the exact visibility $V_{\mathcal{P}}(x,y)$:

$$\tilde{V}(x,y) = \begin{cases} \frac{V_{\mathcal{P}'}(x,y)}{p_1} & \text{with probability } p_1 \quad \text{(a)} \\ \frac{\overline{V_{\mathcal{P}'}(x,y)}V_{\mathcal{P}}(x,y)}{p_2} & \text{with probability } p_2 \quad \text{(b)} \\ -\frac{V_{\mathcal{P}'}(x,y)\overline{V_{\mathcal{P}}(x,y)}}{p_3} & \text{with probability } p_3 \quad \text{(c)} \end{cases} \tag{8}$$

To evaluate the first term (a), we only need to intersect the approximate geometry $\mathcal{P}'$. The terms (b) and (c) require us to test both the exact and the approximate geometry. We can skip the evaluation of the exact geometry when $\overline{V_{\mathcal{P}'}(x,y)} = 0$ for the second term (b) and when $V_{\mathcal{P}'}(x,y) = 0$ for the third term (c).

Table 2 summarizes the values of the estimator for each ray type. From this table we can easily verify that the expected value of the estimator is equal to the exact visibility.

Ray type	**Exact values**		**Stochastic evaluation**		
	$V_{\mathcal{P}}$	$V_{\mathcal{P}'}$	$\frac{V_{\mathcal{P}'}}{p_1}$	$\frac{V_{\mathcal{P}}\overline{V_{\mathcal{P}'}}}{p_2}$	$-\frac{V_{\mathcal{P}'}\overline{V_{\mathcal{P}}}}{p_3}$
$type_1$	0	0	0	0	0
$type_2$	0	1	$\frac{1}{p_1}$	0	$-\frac{1}{p_3}$
$type_3$	1	0	0	$\frac{1}{p_2}$	0
$type_4$	1	1	$\frac{1}{p_1}$	0	0

Table 2: *Stochastic visibility evaluation for each type of ray. Each line shows for of the four ray types; the exact visibility values (leftmost columns) and the stochastic values (rightmost columns).*

3.2. Variance analysis

We estimate the variance of the $\tilde{V}(x,y)$ with respect to the population of all the possible rays $\overline{xy}$ in the scene. The average visibility over all the rays can be considered as the fraction of rays for which $V_{\mathcal{P}}(x,y)=1$. We define f_i to be the relative fraction of all the rays which belong to $type_i$. The average visibility can be expressed as:

$$V_{avg} = f_3 + f_4 \tag{9}$$

The fractions f_i allow us to model the quality of the approximation $\mathcal{P}'$. We define:

$$f_{hit} = \frac{f_1}{f_1+f_2} \tag{10}$$

$$f_{miss} = \frac{f_4}{f_3+f_4} \tag{11}$$

Intuitively, of all the rays which hit the original geometry, f_{hit} is the fraction of these rays that also hit the proxy. Likewise, of all the rays which miss the original geometry, f_{miss} is the fraction of these rays that also miss the proxy.

The variance associated with the estimator in Equation 8 is equal to:

$$Var\left[\tilde{V}(x,y)\right] = E\left[\tilde{V}(x,y)^2\right] - E\left[\tilde{V}(x,y)\right]^2 \tag{12}$$

The expected value of $\tilde{V}(x,y)$ for the population of all rays is equal to the average visibility V_{avg}, because $\tilde{V}(x,y)$ is an unbiased estimator for the exact visibility $V_{\mathcal{P}}(x,y)$. This allows us to calculate the variance for each $type_i$ as summarized in Table 3.

Ray type	Fraction	Variance
$type_1$	f_1	$0 - V_{avg}^2$
$type_2$	f_2	$\frac{1}{p_1} + \frac{1}{p_3} - V_{avg}^2$
$type_3$	f_3	$\frac{1}{p_2} - V_{avg}^2$
$type_4$	f_4	$\frac{1}{p_1} - V_{avg}^2$

Table 3: *The variance of the visibility estimator for each ray type.*

The variance in function of the average visibility V_{avg} and the quality of the approximation is equal to:

$$Var\left[\tilde{V}(x,y)\right] = E\left[\tilde{V}(x,y)^2\right] - V_{avg}^2 \tag{13}$$

$$= f_2\left(\frac{1}{p_1}+\frac{1}{p_3}\right) + \frac{f_3}{p_2} + \frac{f_4}{p_1} - V_{avg}^2 \tag{14}$$

If we express the fractions f_i in function of f_{hit}, f_{miss} and V_{avg}:

$$Var\left[\tilde{V}(x,y)\right] = E\left[\tilde{V}(x,y)^2\right] - V_{avg}{}^2 \tag{15}$$

$$= \frac{(1-V_{avg})(1-f_{hit})(p_1+p_3)}{p_1p_3} + \frac{V_{avg}(1-f_{miss})}{p_2} + \frac{V_{avg}f_{miss}}{p_1} - V_{avg}{}^2 \tag{16}$$

Equation 16 expresses the variance for a single ray in a scene, where the population of all rays has an average visibility of V_{avg}. The fractions f_{hit} and f_{miss} model how closely the exact visibility $V_{\mathcal{P}}(x,y)$ matches the approximate visibility $V_{\mathcal{P}'}(x,y)$.

We will use this expression in Section 4 to choose the optimal probabilities which minimize the variance.

3.3. Outside proxies

We examine the special case when the approximate primitives $\mathcal{P}'$ form a bounding volume for the original primitives $\mathcal{P}$ (see Figure 3).

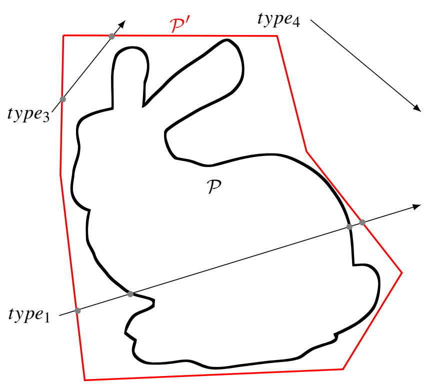


Figure 3: *Rays of $type_2$ do not exist, because it is impossible for a ray to hit the original model ($\mathcal{P}$) without hitting the outside proxy ($\mathcal{P}'$)*

This case has the special property that rays of $type_2$, which hit $\mathcal{P}$ but miss $\mathcal{P}'$, do not exist. Therefore the fraction f_{hit}, of rays which hit an outside proxy and hit an original primitive, equals 1. This allows us to simplify Equation 5 to:

$$V_{\mathcal{P}}(x,y) = V_{\mathcal{P}'}(x,y) + \overline{V_{\mathcal{P}'}(x,y)}V_{\mathcal{P}}(x,y) \tag{17}$$

The variance simplifies to equation:

$$Var\left[\tilde{V}(x,y)\right] = \frac{V_{avg}(1-f_{miss})}{1-p_1} + \frac{V_{avg}f_{miss}}{p_1} - V_{avg}{}^2 \tag{18}$$

Outside proxies are a useful special case since they are frequently used, in the form of axis aligned bounding boxes.

3.4. Inside proxies

We can define a proxy which is fully contained inside the original geometry as illustrated in Figure 4.

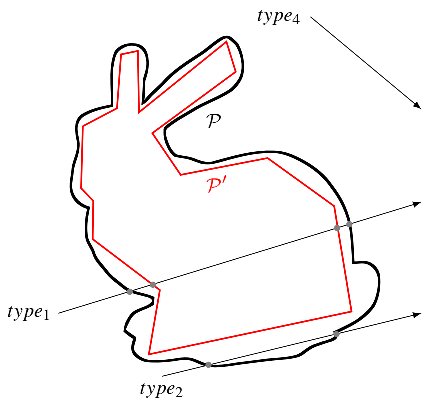


Figure 4: *Rays of $type_3$ do not exist, because it is impossible for a ray to miss the original model ($\mathcal{P}$) after hitting the inside proxy ($\mathcal{P}'$). Note that this requires the original geometry to form a watertight mesh.*

Rays of $type_3$ do not exist when inside proxies are used, because it is impossible for a ray to miss the original primitives $\mathcal{P}$ and also hit the inside proxy. Note that this requires the original primitives $\mathcal{P}$ to form a *watertight* mesh. This implies we always miss the inside proxy when we miss the original primitives and therefore the fraction $f_{miss} = 1$. This allows us to simplify Equation 5 to:

$$V_{\mathcal{P}}(x,y) = V_{\mathcal{P}'}(x,y) + \overline{V_{\mathcal{P}}(x,y)} V_{\mathcal{P}'}(x,y) \tag{19}$$

The variance simplifies to equation:

$$Var\left[\tilde{V}(x,y)\right] = \frac{(1-V_{avg})(1-f_{hit})}{p_1(1-p_1)} + \frac{V_{avg}}{p_1} - V_{avg}{}^2 \tag{20}$$

4. Practical algorithm

Our goal is to evaluate the direct illumination, using stochastic visibility. The direct illumination is evaluated at the points x visible from the camera as given by the direct illumination integral [DBB06]:

$$L(x \to \theta) = \int_S f_r(x, \theta \leftrightarrow \overline{yx}) L(y \to x) V_{\mathcal{P}}(x,y) G(x,y)\, dS_y \tag{21}$$

where y are points on the light source, f_r is the BRDF, $L(y \to x)$ the emission of the light source, $G(x,y)$ the geometric coupling term and $V_{\mathcal{P}}(x,y)$ the exact visibility evaluated by testing all the primitives $\mathcal{P}$ in the scene.

To evaluate the direct illumination using stochastic visibility, we replace the exact visibility $V_{\mathcal{P}}(x,y)$ evaluated for each shadow ray with the stochastic estimator $\tilde{V}(x,y)$. For an efficient evaluation of $\tilde{V}(x,y)$ we construct seperate acceleration structures for the sets $\mathcal{P}$ and $\mathcal{P}'$, to accelerate the evaluation of the approximate $V_{\mathcal{P}'}(x,y)$ and exact visibility $V_{\mathcal{P}}(x,y)$.

In the remainder of this section, we will focus on how we determine the probabilities and various implementation details.

4.1. Probabilities for minimum variance

To reduce the stochastic noise, we choose the probabilities p_1, p_2 and p_3 such that the variance given by Equation 16 is minimized. This gives us the following optimal probabilities:

- **Inside proxies**:

$$p_1 = \frac{1 - f_{hit}(1-V_{avg})}{V_{avg}} - \frac{\sqrt{(1-f_{hit})(1-V_{avg})(1-f_{hit}(1-V_{avg}))}}{V_{avg}}$$
$$p_2 = 0$$
$$p_3 = 1 - p_1$$

The probability p_2 can be set to zero, because the corresponding term $\overline{V_{\mathcal{P}'}(x,y)} V_{\mathcal{P}}(x,y)$ will always be equal to zero (as shown in Section 3.4).

- **Outside proxies**:

$$p_1 = \frac{f_{miss} - \sqrt{f_{miss} - f_{miss}{}^2}}{2f_{miss} - 1}$$
$$p_2 = 1 - p_1$$
$$p_3 = 0$$

The probability p_3 can be set to zero, because the corresponding term $-V_{\mathcal{P}'}(x,y)\overline{V_{\mathcal{P}}(x,y)}$ will always be equal to zero (as shown in Section 3.3).

- **General proxies**: an analytical formula for the optimal probabilities p_1, p_2 and p_3 could not be found for proxies where the set $\mathcal{P}$' can be arbitrarily inside and outside the set $\mathcal{P}$. These probabilities are approximated numerically for a set of sample points and tabulated.

The fractions f_{hit}, f_{miss} and the average visibility V_{avg} are unknown. To estimate these, we trace a number of probe rays which evaluate the exact and the approximate visibility for every point x. V_{avg} is set to the average visibility of the probe rays. f_{hit} and f_{miss} are set respectively to the fraction of rays which hit/miss a proxy after hitting/missing the original geometry.

Figure 5 illustrates the reduction in variance when using adaptive probabilities. Columns 1, 3 and 5 illustrate the stochastic visibility evaluation where the probabilities are chosen to be equal. Columns 2, 4 and 6 show the stochastic visibility evaluation where the probabilities are chosen adaptively after sending 16 probe rays.

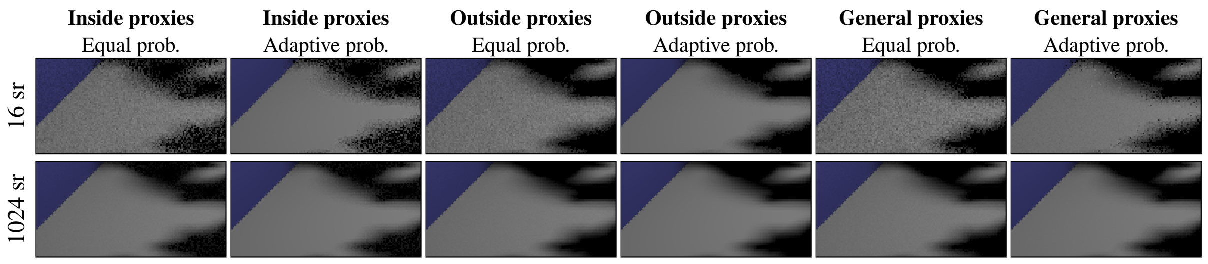


Figure 5: *Stochastic visibility evaluation while using different types of proxies in the Cornell box with dragons (see Table 5). We used 16 probe rays to select the adaptive probabilities. The contributions of the probe rays are not taken into account. The first row shows that using the adaptive probabilities are very effective at reducing the stochastic noise. The second row show that all techniques converge to the correct solution.*

4.2. Probabilities for minimizing the cost

Minimization of the variance does not give the most cost-efficient probabilities. In regions where both $\mathcal{P}'$ and $\mathcal{P}$ are hit, minimization of the variance will assign a high value to the probability p_2 of the term $\overline{V_{\mathcal{P}'}(x,y)}V_{\mathcal{P}}(x,y)$. This is a logical result because in the regions where $V_{\mathcal{P}'}(x,y)$ is hit, the term $\overline{V_{\mathcal{P}'}(x,y)}V_{\mathcal{P}}(x,y)$ becomes equal to the exact visibility $V_{\mathcal{P}}(x,y)$. However, to minimize the cost it is more efficient to only intersect the proxy, in regions where both the proxy and the original geometry are hit.

We were not able to minimize the product of the variance and cost analytically. Therefore, we propose a heuristic which increases the probability p_1 and decreases the probabilities p_2 and p_3 when $V_{\mathcal{P}}(x,y)$ and $V_{\mathcal{P}'}(x,y)$ lead to the same results:

$$p'_1 = 1 - p'_2 - p'_3 \tag{22}$$

$$p'_2 = p_2 \cdot \sqrt[4]{1-(f_1+f_4)} \tag{23}$$

$$p'_3 = p_3 \cdot \sqrt[4]{1-(f_1+f_4)} \tag{24}$$

The heuristic $\sqrt[4]{1-(f_1+f_4)}$ decreases the probabilities of p_2 and p_3 only when f_1+f_4 is almost equal to one. The sum f_1+f_4 is a measure for the number of rays which hit or miss both the proxy and the original geometry. Figure 6 shows that the cost heuristic is able assign a higher chance to p_1 in the umbra of the Cornell box with dragons scene. The probabilities in the penumbra and illuminated regions of the scene remain unchanged.

Note that although we assign a higher value to p_1 which only tests the proxy in regions where $V_{\mathcal{P}}(x,y)$ and $V_{\mathcal{P}'}(x,y)$ are nearly identical, the visibility evaluation still remains unbiased and converges to the correct solution.

4.3. Splitting the visibility

Naïvely replacing the exact visibility $V_{\mathcal{P}}(x,y)$ with $\tilde{V}(x,y)$ introduces unwanted side effects in the visibility evaluation. Imagine a scene where every object has an outside proxy and where the visibility is evaluated stochastically. When we evaluate the visibility from a point x on an object in the scene, the point x will always be inside the bounding box of the object and therefore, $V_{\mathcal{P}'}(x,y)=0$. The stochastic visibility equation would degenerate to:

$$\begin{aligned} V_{\mathcal{P}}(x,y) &= V_{\mathcal{P}'}(x,y) + V_{\mathcal{P}}(x,y)\overline{V_{\mathcal{P}'}(x,y)} \\ &\quad - V_{\mathcal{P}'}(x,y)\overline{V_{\mathcal{P}}(x,y)} \end{aligned} \tag{25}$$

$$= 0 + V_{\mathcal{P}}(x,y)\overline{0} - 0 \cdot \overline{V_{\mathcal{P}}(x,y)} \tag{26}$$

To solve this problem, we split the visibility in two separate terms: the visibility of the object hit at the point x and the visibility of other primitives in the scene. Assume that the object hit at x consists of the set $\mathcal{O}$ primitives. We split the visibility as:

$$\tilde{V}(x,y) = V_{\mathcal{O}}(x,y) \cdot \tilde{V}_{\mathcal{P}\setminus\mathcal{O}}(x,y) \tag{27}$$

The visibility against the object $\mathcal{O}$ is evaluated exactly to account for self intersections. The visibility for the other primitives in the scene $\mathcal{P}\setminus\mathcal{O}$ is evaluated stochastically.

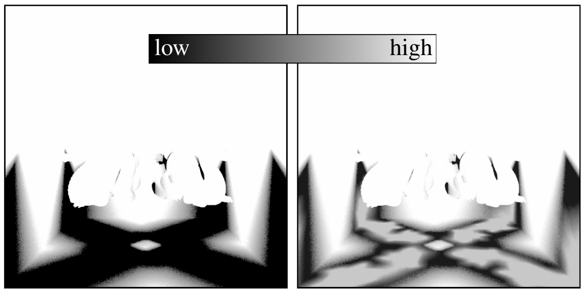


Figure 6: *Probability p_1 assigned to the term $V_{\mathcal{P}'}(x,y)$ for the Cornell box with Dragons from low (black) to high (white). On the left, we minimize the variance. On the right, we apply the correction term. Note that in the hard shadow regions of the scene, p_1 has a higher value when applying the correction term.*

To practically implement the visibility splitting, we used object instancing. The set of primitives $\mathcal{O}$ is the intersected instance at the position x. To calculate the visibility $\tilde{V}_{\mathcal{P}\setminus\mathcal{O}}(x,y)$, we intersect the primitives in $\mathcal{P}$, but ignore any intersections with the instance $\mathcal{O}$.

5. Experiments

We evaluated our algorithm in several scenes of varying complexity (see Table 4).

- **Cornell box with dragons**: this scene features simple geometry and regular shadow patterns with large umbra regions on the floor. The outside proxies for the dragons are the tightest bounding boxes for the model. A general proxy is created with a quadratic edge collapse algorithm and consists of 512 triangles (examples in Figure 1). The inside proxies are generated by placing 40 maximum inscribed spheres along the medial axis of the dragon.
- **Nature and forest scene**: these scenes feature moderate complex geometry and very irregular shadows. The outside proxies are chosen to be the tightest bounding boxes for every model in the scene. The general proxies are generated using a clustering decimation algorithm. For every model in the scene, the vertices of the triangles are clustered in a grid, where the cells have a size equal to 3% of the size of the bounding box of the model.
- **Hairball scene**: this scene features highly complex geometry. The outside proxy is constructed by calculating the convex hull of the hairball. A general proxy is constructed by simplifying the convex hull to a triangle count of 32 triangles.

We rendered the scenes using exact visiblity evaluation and with our techniques using various settings. Table 5 shows an equal time comparison of the exact visibility estimation compared to stochastic visibility using outside and general proxies. Table 6 shows an equal time comparison of exact visibility estimation compared to stochastic visibility using inside proxies (we included difference images against a ground truth image in the supplementary materials). Note that we were only able to generate the inside proxies for the Cornell box with dragons, since inside proxies require the original geometry to form watertight models. Figure 8 shows the mean squared error for increasingly longer render times.

For scenes with moderate to complex geometry (Nature, Forest and Hairball scene), our algorithms are able to trace more shadow rays than when using exact visibility. The only exception is the Nature scene, where we were unable to cast more shadow rays than exact visibility when using mixed proxies. This is due the fact that the mixed proxies consist of 5.4M triangles, compared to 138K bounding boxes for the outside proxies.

Our algorithm performs worse in the Cornell box with dragons. This scene contains large umbra regions. An acceleration structure is very efficient in finding an intersection in these regions, but has more trouble in the penumbra where the rays nearly miss an object. In the umbra, testing the exact geometry is roughly as efficient as testing the approximate geometry.

	Cornell box with dragons	**Nature scene**	**Forest scene**	**Hairball scene**
Scene primitives	3.5M	19.0M	291.1M	2.9M
Inside proxy primitives	180	-	-	-
Outside proxy primitives	9	134K	577	743
Mixed proxy primitives	2000	5.4M	13824	32

Table 4: *Summary of the primitive count in the scene. Each column shows the number of primitives of the scene and the number of primitives used for the inside, outside and mixed proxies.*

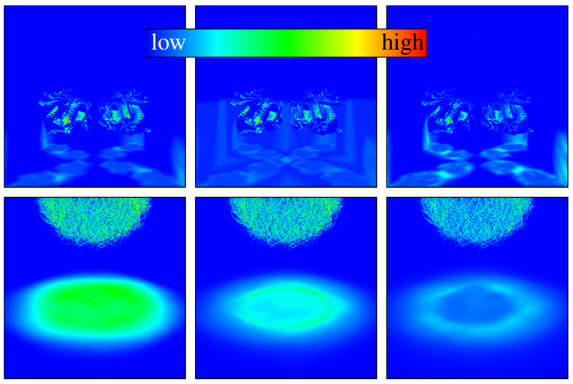


Figure 7: *The number of shadow ray intersection tests in the Cornell box with dragons and the Hairball scene. For simple geometry, the highest intersection count is in the penumbra. For complex geometry, the intersection count is high in umbra and penumbra. Stochastic visibility has an advantage in scenes with complex geometry, because it is more efficient to intersect a simple proxy than the complex geometry. Our algorithm fails for simple geometry, because an acceleration structure can find an intersection very efficiently.*

To support this claim, we rendered the number of shadow ray intersections in the Cornell box with dragons and in the Hairball scene (see Figure 7). In the Cornell box with dragons, most of the shadow ray intersections are located in the penumbra, where the shadow rays nearly miss the dragon model. In the umbra, finding an intersection with the exact geometry is as efficient as using outside and general proxies. However, the Hairball scene features a complex geometrical model and an acceleration structure requires a high number of intersections tests to find an intersection with the hairball. This figure clearly shows that for highly complex geometry, intersecting a proxy is much more efficient than the actual geometry. Therefore, in scenes with large umbra regions, there is no advantage in using stochastic visibility. Scenes with very complex geometry can benefit from using stochastic visibility.

Figure 8 shows that our stochastic visibility estimation has the same convergence speed as exact visibility estimation. Although our stochastic visibility evaluation is able to trace more shadow rays in scenes with high complex geometry (see Forest and Hairball scene in Table 5), the exact visibility achieves lower mean squared errors. This can be explained by the fact that the additional noise introduced by stochastically evaluating the visibility cannot be compensated by the additional shadow rays. The graphs also show that the adaptive probabilities are an effective means to decrease the mean squared error. However, our adaptive technique fails in the Forest scene, where the 16 probe rays are not able to estimate the probabilities correctly.

6. Conclusions

We have presented a theoretical framework for unbiased visibility evaluation using geometry proxies. We demonstrated that our stochastic visibility algorithm is able to produce unbiased images. Although probabilistic visibility evaluation is an experimental idea, we show several example scenes that highlight the potential for future improvements.

7. Acknowledgements

We would like to thank our anonymous reviewers for their comments and helpful suggestions. The scenes used in this paper are courtesy of *PBRT* [PH10]. The *Hairball* model is courtesy of Samuli Laine and Tero Karras. The *Tree* model is generated by Karl vom Berge. Niels Billen is funded by the Agency for Innovation by Science and Technology in Flanders (IWT). Ares Lagae is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO).

References

[BELD13] BILLEN N., ENGELEN B., LAGAE A., DUTRÉ P.: Probabilistic visibility evaluation for direct illumination. *Computer Graphics Forum 32(4) (Proceedings of Eurographics Symposium on Rendering 2013) 32*, 4 (2013), 39–47. 2, 3

[CMS98] CIGNONI P., MONTANI C., SCOPIGNO R.: A comparison of mesh simplification algorithms. *Computers & Graphics 22*, 1 (1998), 37–54. 2

[DBB06] DUTRÉ P., BALA K., BEKAERT P.: *Advanced Global Illumination*, 2nd ed. A K Peters/CRC Press, 2006. 5

[DDSD03] DÉCORET X., DURAND F., SILLION F. X., DORSEY J.: Billboard clouds for extreme model simplification. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, ACM, pp. 689–696. URL: http://doi.acm.org/10.1145/1201775.882326, doi:10.1145/1201775.882326. 2

[FCK*88] FRANKLIN W., CHANDRASEKHAR N., KANKANHALLI M., SESHAN M., AKMAN V.: Efficiency of uniform grids for intersection detection on serial and parallel machines. In *New Trends in Computer Graphics*, Magnenat-Thalmann N., Thalmann D., (Eds.). Springer Berlin Heidelberg, 1988, pp. 288–297. URL: http://dx.doi.org/10.1007/978-3-642-83492-9_25, doi:10.1007/978-3-642-83492-9_25. 2

[HG97] HECKBERT P. S., GARLAND M.: *Survey of polygonal surface simplification algorithms*. Tech. rep., DTIC Document, 1997. 2

[LBBS08] LACEWELL D., BURLEY B., BOULOS S., SHIRLEY P.: Raytracing prefiltered occlusion for aggregate geometry. In *Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on* (2008), IEEE, pp. 19–26. 2

[LD08] LAGAE A., DUTRÉ P.: Compact, fast and robust grids for ray tracing. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1235–1244. 2

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. 8

[RGK*08] RITSCHEL T., GROSCH T., KIM M. H., SEIDEL H.-P., DACHSBACHER C., KAUTZ J.: Imperfect Shadow Maps for Efficient Computation of Indirect Illumination. *ACM Trans. Graph. (Proc. of SIGGRAPH ASIA 2008) 27*, 5 (2008). 2

[SSLL14] SILVENNOINEN A., SARANSAARI H., LAINE S., LEHTINEN J.: Occluder simplification using planar sections. *Computer Graphics Forum 33*, 1 (2014), 235–245. URL: http://dx.doi.org/10.1111/cgf.12271, doi:10.1111/cgf.12271. 2

[Wal07] WALD I.: On fast construction of sah-based bounding volume hierarchies. In *Interactive Ray Tracing, 2007. RT'07. IEEE Symposium on* (2007), IEEE, pp. 33–40. 2

[WMG*09] 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 (2009), 1691–1722. 1

[WMS06] WOOP S., MARMITT G., SLUSALLEK P.: B-kd trees for hardware accelerated ray tracing of dynamic scenes. In *SIGGRAPH/EUROGRAPHICS Conference On Graphics Hardware: Proceedings of the 21 st ACM SIGGRAPH/Eurographics symposium on Graphics hardware: Vienna, Austria* (2006), vol. 3, pp. 67–77. 2

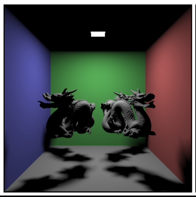
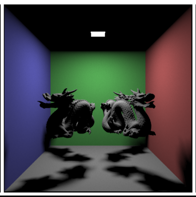
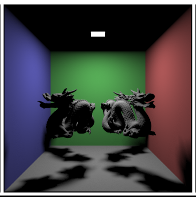
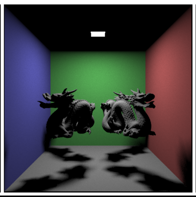
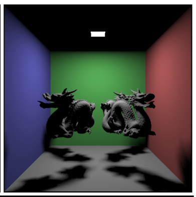
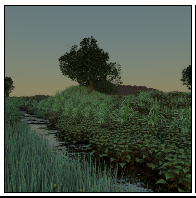
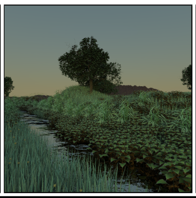
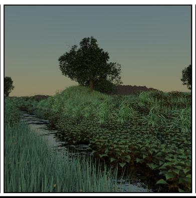
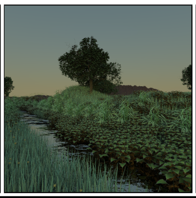
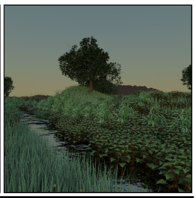
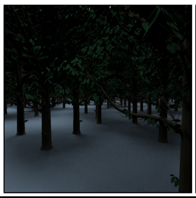
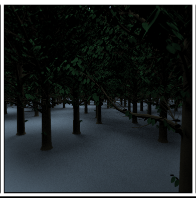
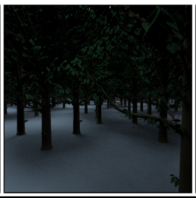
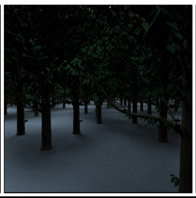
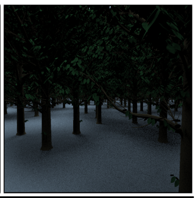
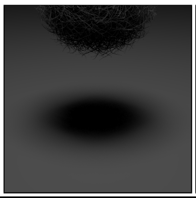
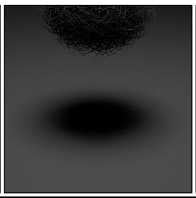
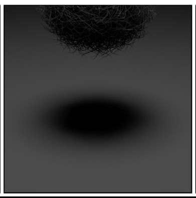
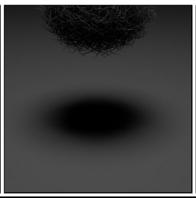
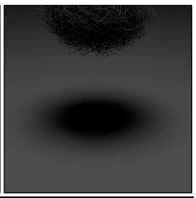
	Exact visibility	**Outside proxies** (equal prob.)	**Outside proxies** (adaptive prob.)	**General proxies** (equal prob.)	**General proxies** (adaptive prob.)
Cornell box with dragons					
Time	157.7s	157.2s	154.7s	149.0s	165.5s
Shadow rays	256	256	192	192	240
Intersection tests	134M	194M	173M	115M	168M
Traversal steps	6369M	4101M	4264M	3067M	3677M
MSE	6.158E-7	4.089E-5	9.689E-6	1.083E-4	9.916E-6
Nature scene					
Time	491.5s	507.4s	491.7s	495.4s	493.3s
Shadow rays	256	400	272	240	208
Intersection tests	1099M	1100M	1036M	762M	786M
Traversal steps	29919M	33398M	31470M	31204M	27828M
MSE	3.894E-6	1.041E-5	7.522E-6	4.214E-5	3.223E-5
Forest scene					
Time	567.8s	575.9s	557.4s	573.5s	570.6s
Shadow rays	256	384	336	352	336
Intersection tests	1206M	1205M	1184M	1298M	1278M
Traversal steps	24902M	22115M	22028M	19997M	19265M
MSE	3.548E-5	5.455E-5	6.008E-5	9.371E-5	1.633E-4
Hairball scene					
Time	658.5s	656.0s	649.5s	651.0s	653.9s
Shadow rays	256	368	352	496	480
Intersection tests	1395M	1284M	1272M	1250M	1387M
Traversal steps	13130M	12555M	12228M	11236M	11147M
MSE	4.461E-7	5.36E-6	3.125E-6	7.628E-6	2.488E-6

Table 5: *Equal time comparison of exact visibility estimation against stochastic visibility estimation. For a rougly equal amount of time, this table summarizes the number of shadow rays the algorithm was able to cast to evaluate the visibility at a point, the number of intersection tests performed for shadow rays, the number of traversal steps through the acceleration structures for the shadow rays and the mean squared error compared to a reference image (for difference images against the reference images, we refer to the supplementary materials).*

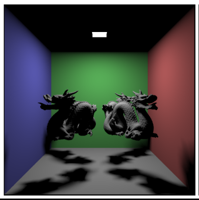
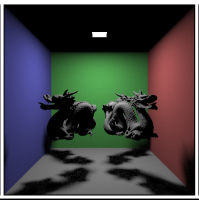
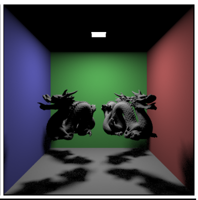
	Exact visibility	**Inside proxies** (equal prob.)	**Inside proxies** (adaptive prob.)
Cornell box with dragons			
Time	157.7s	160.0s	158.6s
Shadow rays	256	192	240
Intersection tests	134M	78M	102M
Traversal steps	6369M	3288M	3121M
MSE	6.158E-7	6.798E-5	1.481E-5

Table 6: *Equal time comparison of exact visibility estimation against stochastic visibility estimation using inside proxies. Note that the inside proxies are only generated for the dragons, because the models have to be watertight to be able to create inside proxies.*

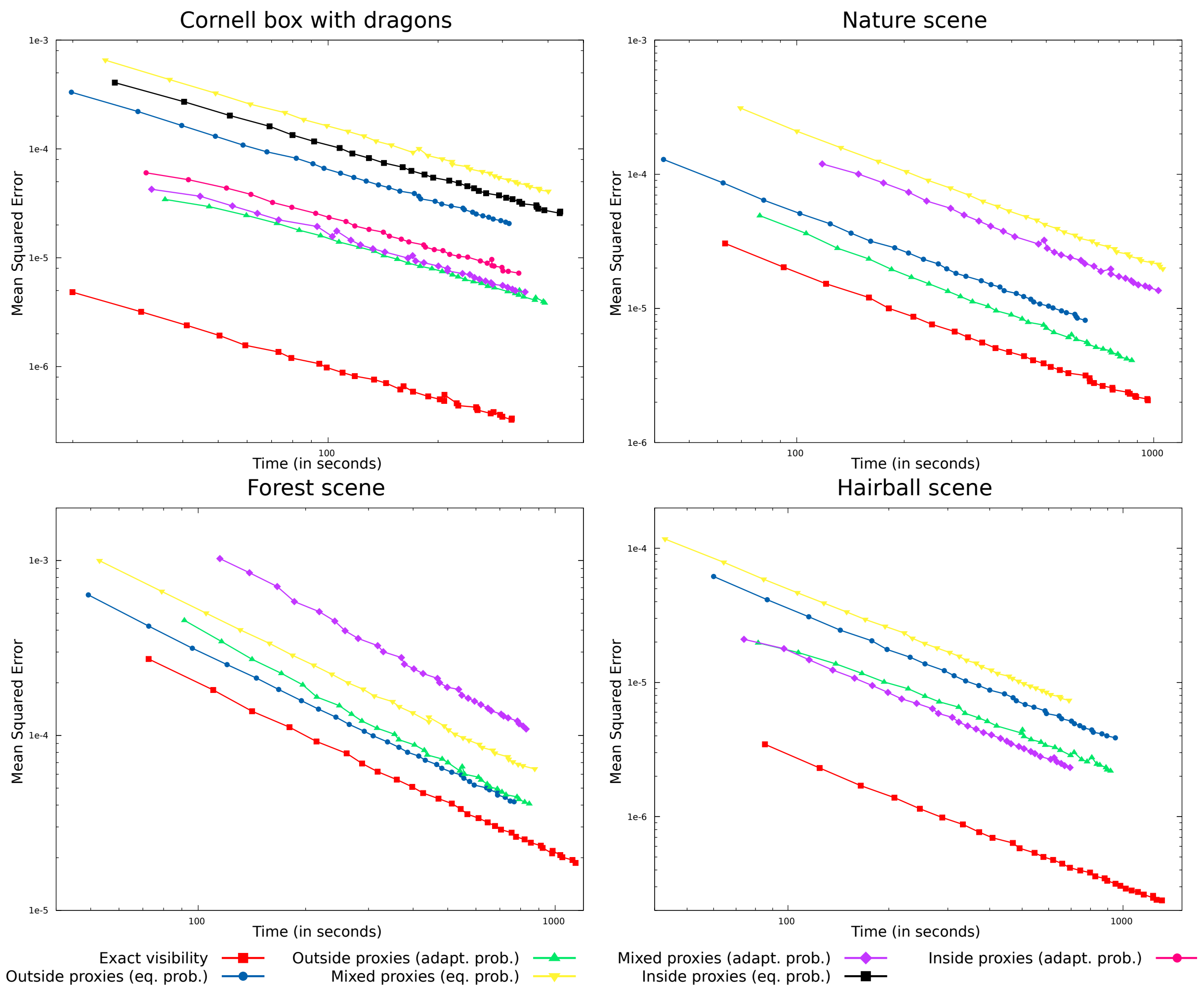


Figure 8: *Mean squared error versus time for our four test scenes.*