Visibility Sweeps for Joint-Hierarchical Importance Sampling of Direct Lighting for Stochastic Volume Rendering

Thomas Kroes* Delft University of Technology
Martin Eisemann† Delft University of Technology
Elmar Eisemann‡ Delft University of Technology

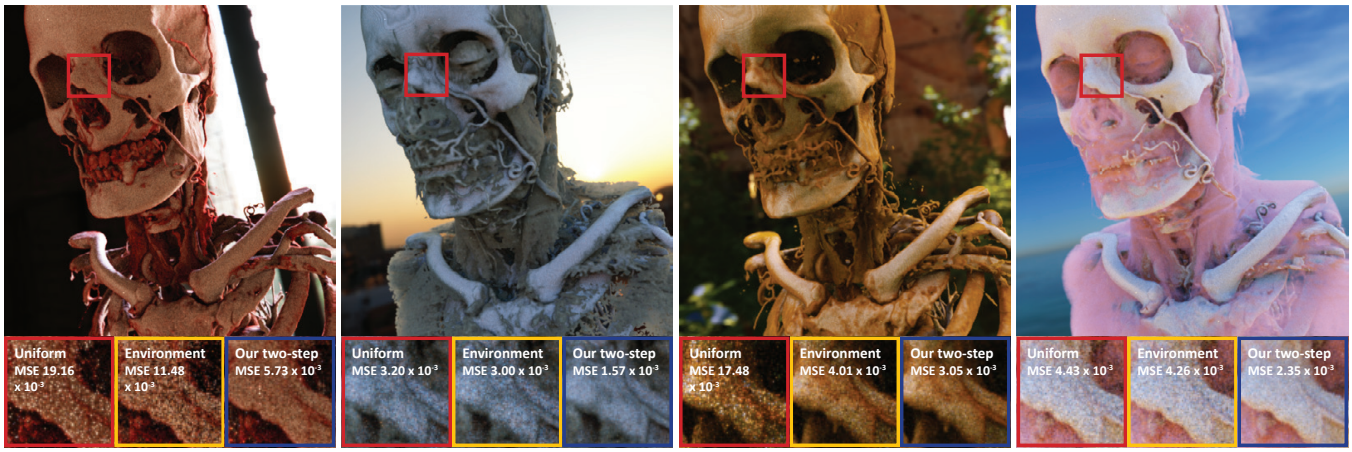

Figure 1: We compute the product of approximated visibility and environment map lighting in a stochastic Monte Carlo volume renderer to steer a joint importance sampling of the direct lighting. Our proposed two-step approach is well suited for dynamic changes in visibility and lighting functions due to a fast sweeping-plane algorithm to estimate visibility. The insets show how our technique (blue) achieves faster convergence with less samples compared to a uniform sampling (red) and importance sampling of the environment map (yellow). Here, 64 samples per pixel have been used. The Manix data set consists of $512 \times 512 \times 460$ voxels.

Abstract

Physically-based light transport in heterogeneous volumetric data is computationally expensive because the rendering integral (particularly visibility) has to be stochastically solved. We present a visibility estimation method in concert with an importance-sampling technique for efficient and unbiased stochastic volume rendering. Our solution relies on a joint strategy, which involves the environmental illumination and visibility inside of the volume. A major contribution of our method is a fast sweeping-plane algorithm to progressively estimate partial occlusions at discrete locations, where we store the result using an octahedral representation. We then rely on a quadtree-based hierarchy to perform a joint importance sampling. Our technique is unbiased, requires little precomputation, is highly parallelizable, and is applicable to a various volume data sets, dynamic transfer functions, and changing environmental lighting.

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1 Introduction

Stochastic volume rendering is computationally intensive. To evaluate the rendering equation, many samples (rays) are required in order to compute the light distribution within a volume. In practice, rays are sent from the camera through the volume and a scattering event occurs at random positions along the ray based on the current transfer function, which maps the volume's density values to material properties. Each scattering event requires generating one or more sample rays to evaluate the rendering equation via Monte Carlo (MC) integration. These rays are ultimately absorbed or potentially hit a light source, e.g., the environmental light. Using standard sampling techniques at the scattering events can be inefficient, as no knowledge about the volume absorption or light characteristics is used. As a result, many rays might contribute little or nothing to the final image.

Importance-sampling techniques [7, 5, 23] incorporate knowledge about the scene to place more effort on potentially light-carrying paths to accelerate the convergence of the result. Some approaches combine information about the material and light positions. However, one important factor, the (volumetric) scene, and, hence, visibility is not taken into account. Previously, visibility approximations were only used directly in the shading evaluation, resulting in biased images [26]. Furthermore, a brute-force visibility precomputation is costly and transfer-function changes require a complete reevaluation.

In this work, we focus specifically on evaluating direct lighting for a volume data set with arbitrary and interactively changing transfer functions defining varying diffuse materials in the context of an unbiased MC-based stochastic volume renderer. The volume is lit by a natural illumination in the form of environmental lighting.

The key idea of our approach is to use environmental light and visibility represented as a joint probability density function (pdf), Fig. 2. As a result, the sampling process, steered by this pdf, becomes more efficient than in previous work, while keeping the result unbiased. The sampling technique allows us to evaluate the

*e-mail: t.kroes@tudelft.nl
†e-mail: m.eisemann@tudelft.nl
‡e-mail: e.eisemann@tudelft.nl

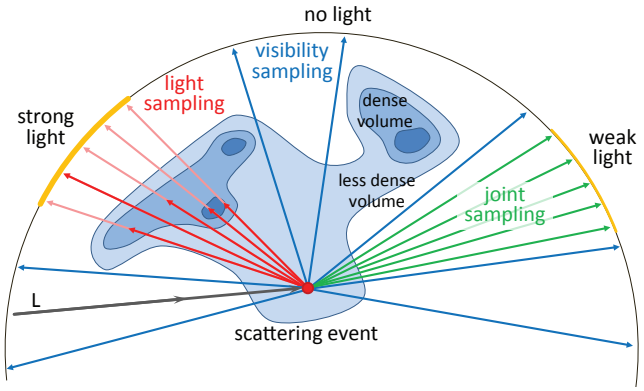

Figure 2: **Problem Statement:** For efficient sampling, samples with both strong light and strong visibility need to be found. Sampling according to the lighting only (red) may give bad results as the samples get absorbed, sampling only according to the visibility (blue) might miss important lights. Product sampling (green) solves the problem. Unfortunately, the visibility is usually unknown beforehand.

direct light at any scattering event within the volume. While our results could be generalized, we illustrate the application to single scattering.

Our contributions are as follows:

- An efficient sweeping-plane algorithm to compute approximate visibility within a 3D volume;
- A product importance sampling solution based on joint environmental light and visibility information;
- A GPU-adapted and highly-parallel implementation.

Our technique is useful for any volumetric renderer with dynamically changing content, such as environmental light, transfer functions, etc., making it an interesting addition to visualization and rendering systems aiming for unbiased results.

2 Related Work

The literature on volumetric-illumination techniques is vast, which is why we will focus only on certain aspects to put our approach in perspective. A recent survey on this topic can be found in [10].

Ambient Occlusion helps to better perceive certain shapes and their relative positions by measuring the light accessibility for each scene point. Luminance is linked to the degree of local occlusion [34]. Multi-resolution variants [16], and even dynamic ambient occlusion variants [28], which allow changes to the transfer function, have been considered. Nonetheless, ambient occlusion computes only a statistical scalar value to approximate the ambient light, which means that directional information is lost. We incorporate full directional support for high-quality unbiased physically-based rendering.

Visibility Approximation for Semi-Transparent Structures are most common in physically-based volume rendering. Opacity shadow maps [11] are an extension of shadow maps [33] using a stack that stores alpha values instead of depth values to support shadow computation for complex, potentially semi-transparent structures. Deep shadow maps [17] are a more compact representation, which store a shadow-function approximation per pixel. They have quickly been adopted for volume rendering [9, 27].

All such techniques are fast but inapplicable in our scenario of stochastic MC volume rendering. First, using approximate visibility directly for shading introduces a bias, which is unacceptable for certain applications. Second, these techniques support only point and directional light sources, whereas we aim for environmental lighting. Third, visibility is costly to compute and even approximating it can usually involve many rays, although not all locations might ultimately contribute to the image. Our approach computes visibility in a coarse 3D grid and uses it only to carefully steer the sample generation. In this way, our result remains unbiased, exact, and supports arbitrary environmental lighting.

Basis-Function Techniques decouple light-source radiance and visibility, which allows for dynamically changing the illumination. Spherical harmonics (SH) are prominent basis functions, used for example for precomputed radiance transfer [30], and were first used in the context of volume rendering to precompute and store isosurface illumination [2]. They have also been used to store visibility for volume rendering under natural illumination [26]. Other research in this area mostly aimed at generalizations to support advanced material properties [15] or reduce memory costs [14].

While SH are well suited to represent low-frequency functions, their direct use for visibility is a strong approximation and introduces bias. Further, only low-frequency illumination is supported, in contrast to our solution.

Image Plane-Sweep Volume Illumination Approaches move a virtual plane through a scene to invoke the shading computations for all positions within the plane in parallel. The parallelism makes these approaches highly applicable to modern architectures, such as the GPU. Using carefully-chosen approximations (e.g., a forward peaked phase function, single point or directional light source), single and forward multiple scattering effects can be simulated at interactive frame rates [31]. We decouple the plane sweep from a particular light source to enable general illumination and efficient sampling in stochastic MC volume rendering.

Recently, iterative convolutions on volume slices have been used to approximate direct lighting [22]. The results are approximate, some parameter settings have to be carefully chosen, and only particular light-source configurations are efficiently supported (e.g., usually Gaussian and behind the observer).

MC Ray Tracing for volume rendering gained attention with the advances of modern GPUs, which made interactive progressive rendering possible. First attempts sacrificed generality for performance [25] and did not support translucent materials. New approaches, such as Exposure Render [13] achieve images of very high realism. They employ all the benefits of physically-based MC techniques: arbitrary natural illumination, real-world cameras with lens and aperture (e.g., for depth-of-field effects). We implemented our approach building upon this open source solution. Only recently, specialized algorithms have been developed to efficiently handle participating media by splitting the evaluation into an analytical and a numerically evaluated part [20].

Importance Sampling is a powerful sampling technique to render objects illuminated by natural or complex lighting [7] under an environmental illumination. An efficient method for non-specular materials is to place pre-integrated directional lights at the brightest locations [1, 12, 21]. These methods work extremely well in the absence of occlusion, but shadowed regions may appear noisy. When materials are increasingly specular, a large number of lights is needed to adequately represent the environment map. Consequently, many physically-based MC techniques sample the environment map directly to avoid any artifacts and its intensity can even be used as a pdf to steer the sampling [23].

If also visibility or material properties are to be included, the pdfs can be combined in a single MC estimator via multiple importance sampling (MIS) [32]. MIS is most efficient if only one of the

sampled functions is complex and will pick the best one. If both are complex, MIS provides little advantage and is likely to waste samples in regions with little influence. Visibility and lighting can both be complex and only a joint sampling of both functions can be efficient (Fig. 2). A first step towards this direction was taken in [3]. Their technique importance samples the environment map to produce a candidate sample. Its probability is then evaluated again using a special pdf involving the BRDF to determine if an evaluation is triggered. Such a sampling can quickly become costly, due to potential high rejections rates (in the order of 90%) [3].

More related to our sampling approach are techniques for joint importance sampling that compute the BRDF/environment-map product [5, 6, 4] and BRDF/visibility/environment-map product [29] to steer sample placement. In the context of participating media, joint importance sampling can also be employed to optimize volumetric paths [8]. In this article, we focus on efficient visibility/environment-map sampling. Nonetheless, we also rely on a quadtree-based product to hierarchically warp samples [4].

3 Overview

In the following, we will describe our algorithm in detail. First, we provide the necessary background knowledge (Sec. 3.1). Then, we describe our actual solution, starting with our data structures and data representations (Sec. 3.2), which are designed with GPU-efficiency in mind. Our visibility-sweep algorithm (Sec. 3.3) is used to compute an approximate visibility within the volume. It is then used in conjunction with the scene illumination to yield a joint sampling technique to steer the MC evaluation (Sec. 3.4). Finally, we describe the necessary implementation details (Sec. 3.5). The benefits for convergence behavior and the support of dynamic lighting and transfer-function changes will be demonstrated in Sec. 4.

3.1 Background and Goal

We adopt the notation from [26] for the emission-absorption volume-rendering equation [18] in an isotropic medium:

$$L = \int_0^\infty A(t)E(\mathbf{x}(t))\mathrm{d}t. \tag{1}$$

It describes the recorded radiance L along a camera-ray position $\mathbf{x}(t)$ parameterized by t, where

$$A(t) = exp\left(-\int_0^t \tau_\alpha(D(\mathbf{x}(s)))\mathrm{d}s\right) \tag{2}$$

$$E(\mathbf{x}(t)) = \int_\Omega \tau_\rho(D(\mathbf{x}(t)))V(\mathbf{x}(t),\omega)L_i(\omega)\mathrm{d}\omega. \tag{3}$$

E is the emission and $A(t)$ is the absorption up to position $\mathbf{x}(t)$. The volume density at location $\mathbf{x}(t)$ is denoted as $D(\mathbf{x}(t))$. The visibility at a position $\mathbf{x}(t)$ in direction ω is denoted as $V(\mathbf{x}(t),\omega)$. The incoming light $L_i(\omega)$ from direction ω, integrated over all possible directions Ω, is assumed to be independent of $\mathbf{x}(t)$, i.e., we assume an environmental light. A transfer function τ maps a density value y to an extinction coefficient $\tau_\alpha(y)$ and scattering albedo $\tau_\rho(y)$. For brevity, we will omit the ray parameter t and write only $\mathbf{x}$ to denote a certain location.

We use stochastic ray marching to solve the integral in Eqs. (1) and (2) based on [13]. To solve Eq. (3) stochastically, MC integration is applied:

$$E(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N}\frac{\tau_\rho(D(\mathbf{x}))V(\mathbf{x},\omega_j)L_i(\omega_j)}{p(\mathbf{x},\omega_j)}.$$

Here, p is a pdf that is used to weigh and generate the random sample vectors ω_j. The MC integration can become highly inefficient with a bad choice of the pdf p as it may create many samples ω_j which contribute little to the final result, Fig. 2.

The focus of this paper is on choosing an effective pdf p and its efficient computation. In order to achieve this, we split p into two components

$$p(\mathbf{x},\omega) = \frac{1}{W(\mathbf{x})}p_V(\mathbf{x},\omega)p_{L_i}(\omega).$$

p_V is a pdf based on the visibility, which changes locally throughout the volume based on the location $\mathbf{x}$, p_{L_i} is a pdf based on the position-independent environmental lighting and $W(\mathbf{x}) = \int p_V(\mathbf{x},\omega)p_{L_i}(\omega)\mathrm{d}\omega$ is a normalization factor to produce a valid pdf. p_{L_i} is known and based on the intensity of the environmental lighting, normalized by its overall intensity. The representation of these functions, the computation of $p_V(\mathbf{x},\omega)$, $p(\mathbf{x},\omega)$, and how to draw samples from $p(\mathbf{x},\omega)$ are the core of our method and explained in the following sections. We explain the data structures, then the visibility approximation, which will be used to derive p_V, before combining all the elements.

3.2 Octahedral Representation

Before explaining the algorithmic part of our approach, we will focus on the chosen data structures. They were developed to ensure an efficient evaluation on modern hardware and to simplify generation, sampling, and product computation. These elements will be necessary to drive the MC sampling process.

As we are dealing with potentially semi-transparent media in volume rendering, we assume V to be locally smooth with respect to $\mathbf{x}$ and ω. This allows us to estimate V at discrete positions $\mathbf{x}_d$ and a few discrete directions ω_d. We arrange the locations within a 3D voxel grid of user-defined size encompassing the original volume. These local estimates are then interpolated during rendering to obtain an approximation of the actual visibility in each location.

For a fixed location $\mathbf{x}$ our functions V and L_i and their respective pdfs p_V and p_{L_i} are spherical functions, i.e., they solely depend on a direction vector ω. Given that we only consider piecewise-constant pdfs, we will represent these functions as octahedral maps, which is a discrete image-based area-preserving representation [24] and can be saved/accessed as a 2D texture (Fig. 3). Each texel p in these maps is associated with one direction $\omega_d(p)$ and indicates the accumulated volumetric visibility in direction $\omega_d(p)$ from the maps location. We will refer to these visibility maps for each discrete location $\mathbf{x}_d$ as *visibility voxels* $\mathbf{V}_d$.

3.3 Visibility Approximation

In this section, we describe how to compute the entries of the visibility voxels via our sweeping plane algorithm. The visibility is

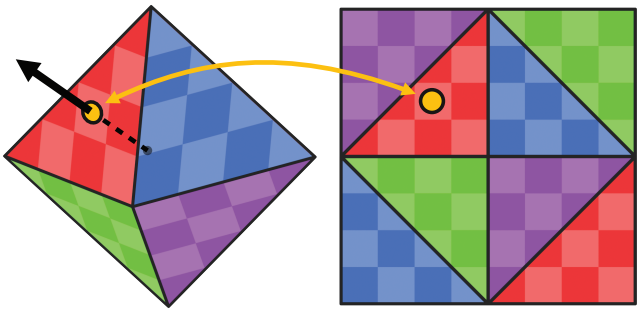

Figure 3: **Octahedral representation:** We present spherical functions using an octahedral representation. (a) 3D representation, (b) unfolded 2D representation.

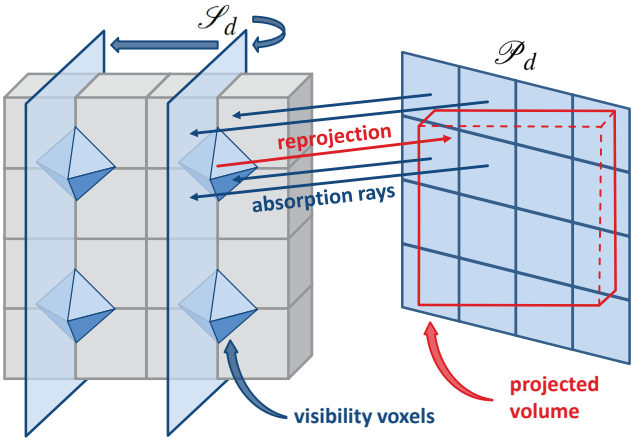

Figure 4: **Visibility Sweeps:** We compute the absorption of sample rays starting at plane n along direction ω_d up to the positions coinciding with a sweeping plane, which is orthogonal to the main component of ω_d. To compute the absorption at a visibility voxel in direction $-\omega_d$ we reproject its position onto $\mathscr{P}_d$ and query the interpolated absorption value. All components can be efficiently computed on the GPU.

computed for one direction $\omega_d(p)$ at the time. In each step, one slice of $\mathbf{V}_d$ is evaluated in parallel. Previous results are reused, making only a few value lookups per step necessary. Therefore, the amortized cost over all $\mathbf{x}_d$ is very low. After all directions were treated, the resulting $\mathbf{V}_d$ is used to derive the pdf p_V, which will guide the sampling process. An illustration of a single sweep step is given in Fig. 4 and described in the following.

Here, we describe the process for one given direction ω_d, for brevity we omit the direction parameter in brackets. First, a plane $\mathscr{P}_d$ with normal ω_d is defined, the orthogonal projection of the data volume's bounding box defines its size. We then create a set of r rays at uniformly distributed positions within this projection having the same direction as ω_d. To coordinate the ray traversal, we introduce a sweeping plane $\mathscr{S}_d$ which is orthogonal to one of the main axes of the original volume data. This axis is chosen based on the main direction of ω_d (defined as the maximum of the absolute values of its three components). $\mathscr{S}_d$ is initialized to intersect the first 2D slice of $\mathbf{V}_d$, so that it coincides with the position of the visibility voxels within this slice. We traverse the volume along the rays starting at $\mathscr{P}$ and accumulate visibility changes until they hit $\mathscr{S}$. This accumulation effectively keeps track of all relevant information that lies behind the ray when reaching a new visibility voxel.

The main loop of the algorithm moves $\mathscr{S}_d$ forward along the main direction by one visibility-voxel slice at a time until all slices are processed. After each step of $\mathscr{S}_d$, the rays advance via ray marching from their previous position until they reach $\mathscr{S}_d$ again. On their way, the absorption values along the ray are accumulated and, when reaching $\mathscr{S}_d$, stored in a 2D texture mapped onto the initial positions on $\mathscr{P}_d$. Please note that the resolution of the visibility voxel grid and the original volume can be different. Next, the visibility voxels coinciding with $\mathscr{S}_d$ (now reached by the rays) are updated by querying the interpolated absorption values produced by the rays. This gathering operation is highly parallelizable and more efficient than a scattering strategy.

After the algorithm finishes and all directions have been processed, we have a discrete approximation $\mathbf{V}_d$ of the visibility within the volume, which, if normalized, results in the pdf p_V. We add a small ε-value beforehand to prevent zero probabilities, which would introduce a bias. The main observation is that this iterative update is more efficient than individual visibility computations per visibility voxel.

3.4 Joint Importance Sampling

At a scattering event during rendering, we want to make use of a joint importance sampling combining visibility and environmental lighting. We have explained how to produce the pdfs for p_V and p_{L_i}. Here, we explain how to combine both. The computation is divided into a preprocess, taking place whenever the environment map or the transfer function changes, and an online process, taking place whenever a scattering event occurs during rendering.

Preprocess For the preprocess, we assume that the environment map is also given as an octahedral map, otherwise we convert it first. As a reminder, p_{L_i} is defined as the normalized intensity value of the environmental lighting, giving higher importance to the brighter parts. In general, the resolution of the octahedral maps of p_{L_i} will be higher than for p_V. To combine both, we first adapt the resolution of p_{L_i}. To simplify explanations, we assume that the resolution in width and height is chosen to be a power of two.

Similar to [4], we create a multiresolution pdf in the form of a quadtree, i.e., each node saves the average of its four child nodes, with the leaves being the individual pixels. To match the resolution between lighting and visibility, we choose a level l in p_{L_i} whose resolution is equal to the directional resolution of a single visibility voxel. We then multiply all entries in $\mathbf{V}_d$ with the respective information in p_{L_i} at level l. The result is an unnormalized joint pdf of the combined product.

Rendering In the rendering phase, we create a final combined pdf p for each scattering event at location $\mathbf{x}$. This pdf is used to draw a single sample, as this strategy is often more efficient in a stochastic volume renderer with semi-transparent media [13]. Nonetheless, the sampling algorithm naturally extends to any number and distribution of initial samples, including quasi-MC methods [19].

To derive the pdf p, we first linearly interpolate the neighboring visibility voxels which now carry the information of both visibility and lighting as described in the preprocess. Initially, this interpolated result is not a pdf. Nevertheless, we do not normalize it right away, but compute a multiresolution representation in the form of a quadtree where each node is the average of its child nodes. Following the hierarchical warping technique [5], we can then transform a uniformly distributed $[0,1)^2$-variable into one that is distributed according to p by passing the sample down in the quadtree according to the local probabilities. In contrast to [5], we need to normalize each 2×2 tile that we encounter during the quadtree sampling but as we only draw a single sample per scattering event the effort is only $O(\log n)$ compared to $O(n)$ if we would create a complete pdf for the interpolated visibility voxel. Here, n is the number of texels in the lowest level of the quadtree.

In case the environment map has a high resolution, we propose to use a *two-step approach* that continues the descent on the remaining quadtree of the higher resolved environment map [6, 4]. This step is especially beneficial for complex high-frequency illumination, which is otherwise not well taken into account during the sampling.

3.5 Implementation Details

We found that using 8×8 maps to represent the visibility at each $\mathbf{x}_d$ is generally a good memory/performance trade-off. The amount of visibility voxels should be based on the scene/feature size and $\mathbf{V}_d$ should be chosen slightly larger to encompass the original volume and ensure a correct boundary sampling. The number of ray-marching steps along the ray should be based on the resolution of the data set and the number of visibility voxels. The maximal step size should be equal to the voxel size in the original data set in order to not miss any details.

For improved cache usage, we actually do not compute all visibility-voxel octahedral maps in advance. Instead, we avoid that rays write to different textures during the ray traversal in the visibility precomputation and store the occlusion values for a direction ω_d in a separate 3D texture. Once the pass for ω_d is completed, we perform the multiplication with the environment map as explained above and keep this texture in memory. During the rendering process, when a scattering event occurs, it might not lie directly on a visibility voxel. We thus retrieve the interpolated values using hardware filtering from these 3D textures and construct the visibility-voxel octahedral map on the fly. Although this might sound costly, it turned out that in practice, this cost is outweighed by the cache advantages due to a better data locality and we avoid constructing visibility voxels in areas where no scattering occurs. Nonetheless, on future hardware a first reconstruction pass might become preferable.

4 Results

We integrated our algorithms into the stochastic CUDA-based volume renderer from [13]. All images have been generated on a 64bit Intel© Core™ i7 920 with 2.67GHz, 12GB of RAM, and an NVIDIA GeForce GTX 760.

We compare the performance and do a qualitative comparison between the existing and our approach. We compute the mean-squared error (MSE) and compared to reference solutions using 8196 samples per pixel with uniform sampling. Our environment maps all had a resolution of 2048×2048 pixels (in the octahedral representation). For all tests, the joint importance pdf p was constructed on the fly for each scattering event via interpolation of eight visibility voxels.

Timings and Parameters The overhead during rendering using our visibility sweeps is low compared to the gain in quality, especially as the sweeping-plane algorithm to update the visibilities in $\mathbf{V}_d$ is evoked only if the transfer function or illumination changes. As standard parameters, we use a 8^2 directional map for each visibility voxel and set one visibility voxel for each 4^3 voxel subset of the original data volume. The overhead during rendering is roughly only 10%, compared to rendering the same number of samples per pixel using plain uniform sampling. This includes interpolating the visibilities, creating the multiresolution 2D pdf representation on the fly and the joint importance sampling itself.

We compared our visibility sweeps approach to a brute-force computation of the visibility where each entry for the visibility voxels is computed exactly using ray marching. Table 1 shows a comparison of the timings for different parameters. For our proposed standard parameters and a reasonable number of absorption rays our approach is approximately $6\times$ faster than the brute-force computation. It is important to note that this factor becomes larger with an increasing number of visibility voxels (up to a factor of 15 in our tests in Table 1). Further, the test scene (Manix) resembles and isosurface due to its very steep transfer function, hence, the brute-force ray marching stops if the ray hits the isosurface. In our sweeping-plane algorithm the rays need to traverse the whole volume. So, we deliberately chose a difficult scenario - the benefit will be even bigger for a higher number of visibility voxels and more transparent volumes.

Additionally, we checked our assumption that we can interpolate the queried visibility during the reprojection step in the sweeping-plane algorithm. It should be pointed out that the results will always remain unbiased, independent of the resolution because the values are only used to guide the sampling process. To this extent, we reduced the number of absorption rays that are traversed through the volume shown in Fig. 5, which is of size $512 \times 512 \times 373$ voxels. Consequently, the visibility voxels will have to rely on an interpolated result. The number of rays influences the result only slightly and no visible errors are introduced. This result indicates that we can rely on a relatively cheap preprocess to approximate the visibility, which reduces the additional overhead in our approach.

Vis. Voxels	256x256x230	128x128x115	64x64x57	32x32x28
Memory	964.7	120.6	15.0	1.8
Sweep (16^2)	3.76	1.93	0.76	0.53
Sweep (32^2)	3.89	1.97	0.79	0.54
Sweep (64^2)	4.05	2.03	0.79	0.55
Sweep (128^2)	4.31	2.40	1.04	0.59
Sweep (256^2)	6.36	3.23	1.63	1.41
Brute-Force	97.35	15.17	2.79	0.57

Table 1: Memory requirements (MB) and timings (seconds) for the visibility sweep algorithm and varying input parameters in comparison to a brute-force visibility computation. We shoot 16^2, 32^2, 64^2, 128^2 and 256^2 absorption rays per sweeping direction. All experiments are performed on the Manix data set ($512 \times 512 \times 460$ voxels).

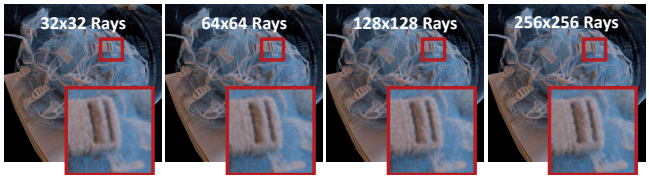

Figure 5: Influence of the visibility sampling precision (number of absorption rays) on the result.

Qualitative Evaluation We compare our approach to uniform sampling, importance sampling of the environment map only, importance sampling of the visibility only, a combined approach, where the visibility pdf is multiplied with the downsampled pdf from the environment map of the same resolution, as well as our combined two-step approach, which makes use of the combined sampling but switches to the full environment-map resolution as soon as a leaf in the combined pdf representation is reached to further support high-frequency lighting.

Fig. 8 shows an equal-time comparison of all the techniques after 10 s render time, excluding the visibility precomputation. For comparison, we show the computed number of samples per pixel (SPP) and the Mean-Squared Error (MSE) for each approach. Though the number of samples is lower, due to the computational overhead induced by the joint sampling, the noise is significantly reduced with our approach. Due to a lower ray coherency the uniform sampling creates less samples per pixel in the same time than most of the other approaches.

Figs. 1 and 6 show an equal sample comparison. Fig. 7 shows an equal quality comparison, where we precomputed the result for various power-of-two number of samples and illustrate the ones closest to the indicated error.

Additionally, we provide error plots for the Statue and Engine Block scene with respect to the number of samples in Figs. 9 and 10. As expected, uniform sampling performs worst. Interestingly, the visibility sampling alone performs better in the beginning but has a worse convergence. We found that the images also contain a lot more firefly artifacts. We, therefore, believe the reason for the convergence behavior lies in the approximate visibility function at occlusion boundaries. If a direction is supposed to be occluded it is sampled with a low probability and therefore a high weighting. If it accidentally hits a bright light source behind it, high energy samples are added to the result which result in the fireflies. As the direction around this occlusion boundary is rarely sampled it takes a lot of

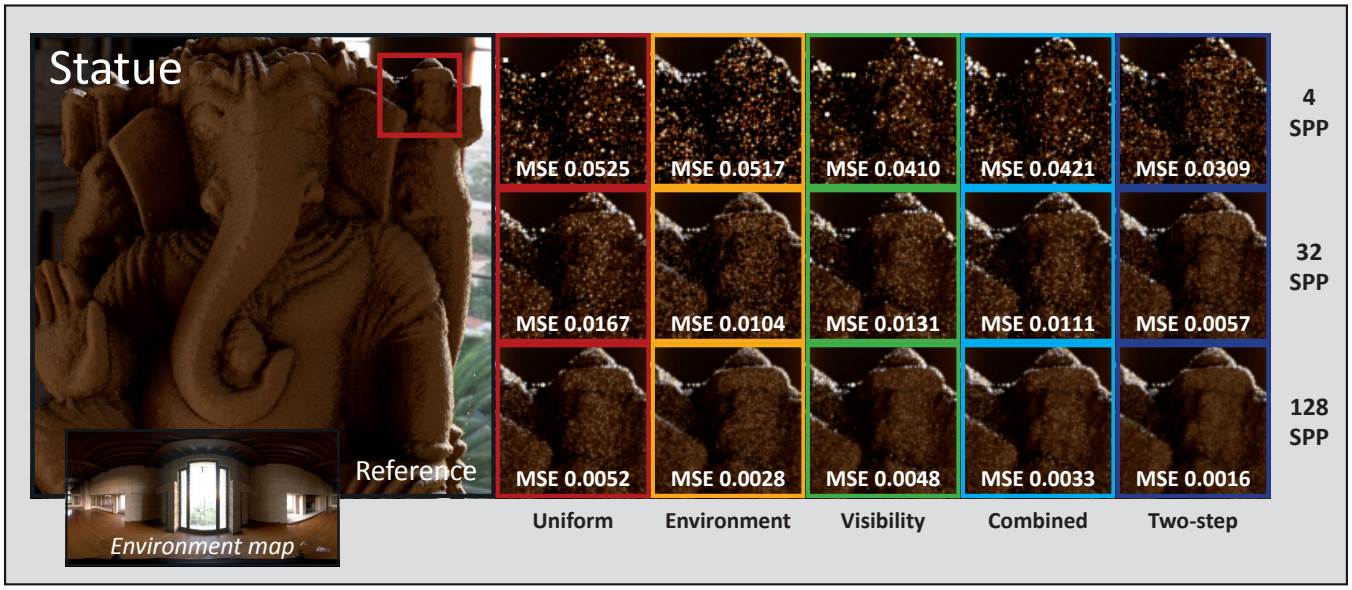

Figure 6: **Equal Sample Comparison:** We compare our proposed two step importance sampling technique (dark blue) using 4, 32 and 128 samples to uniform sampling (red) and importance sampling of the environment map only (yellow), the visibility only (green), and the combined low-resolution product (light blue). All images are unbiased and a reference, as well as the environment map are shown on the left.

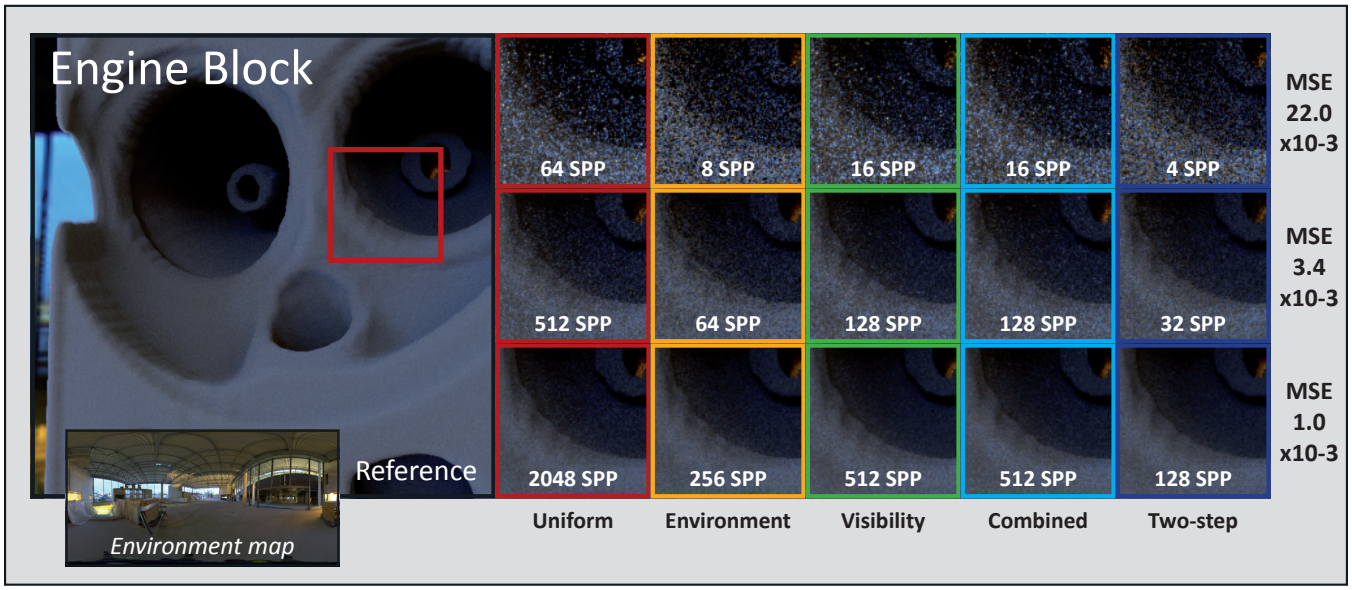

Figure 7: **Equal Quality Comparison:** We compare our proposed two step importance sampling technique (dark blue) to uniform sampling (red) and importance sampling of the environment map only (yellow), the visibility only (green), and the combined low-resolution product (light blue). All images are unbiased and a reference, as well as the environment map are shown on the left. For approximately the same quality, our two-step approach requires significantly less samples.

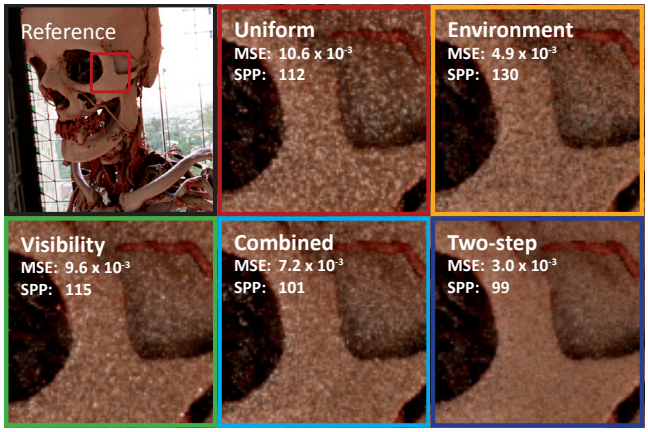

Figure 8: **Equal time comparison:** All images, except the reference image, have been created using 10 seconds of rendering time.

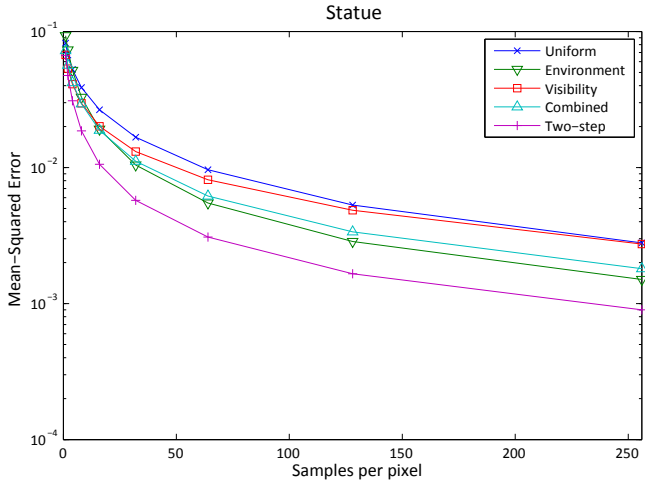

Figure 9: Convergence graphs for the Statue scene (Fig. 6)

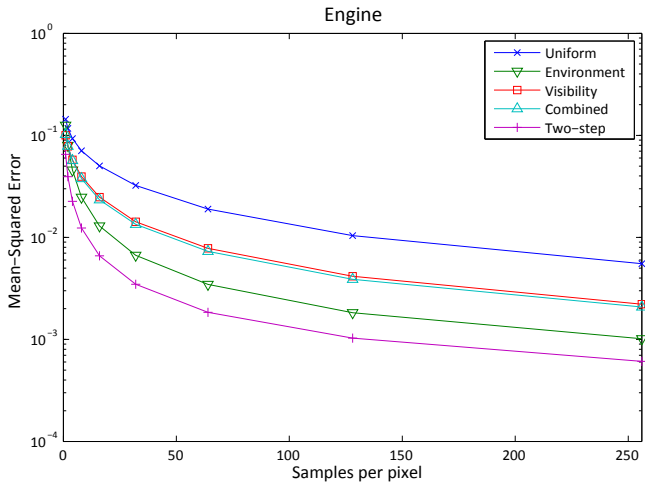

Figure 10: Convergence graphs for the Engine Block scene (Fig. 7).

samples to correct for these errors. If the lighting is incorporated in the pdf, these cases are taken care of sufficiently. Environment map and the low-resolution combined sampling perform almost equally well on the Statue scene. Presumably, importance sampling the light wastes a lot of samples that are absorbed within the volume. The combined sampling approach suffers to some extent from the low resolution of the visibility function and, therefore, the combined pdf is not able to capture the high frequency details of the environment map. This disadvantage is compensated by the two-step approach, which can make use of both the visibility and the high-frequency illumination information and shows better convergence rates even at high sampling rates. The results suggest that it is highly beneficial to incorporate the proposed visibility sweeps and joint sampling in the two-step approach for stochastic MC volume rendering.

5 Conclusion

We presented a joint sampling approach relying on visibility and lighting information within an interactive unbiased stochastic volume renderer. The core of our solution is an efficiently-computed visibility approximation based on a sweep-plane algorithm. Its performance allows us to change environmental lighting and transfer functions dynamically. We carefully designed our algorithm for GPU execution and have demonstrated its applicability to different volume data sets.

Visibility sweeps could prove beneficial for traditional boundary-representation rendering as well. To some extent this is illustrated by using transfer functions, which lead to very sharp features. Our approach usually lowers the amount of needed samples significantly compared to previous solutions at equal quality, which is an important result as the evaluation of samples is a very costly element in most production and rendering contexts.

There are still some options for minor optimizations for the traversal algorithm. First, a pruning of the absorption rays that do not intersect with the volume at all, and second, an early exit strategy for rays that are already fully absorbed, could potentially result in a traversal speed-up at the cost of a more complex algorithm. Though not yet implemented, interactive clipping (slicing) of the volume is naturally supported in our approach, as it simply requires disregarding the intensity values in front of the slicing plane during the visibility computation. A remaining challenge is the incorporation of non-diffuse media. For isotropic media one could precompute several pdfs based on the angle of the incoming and outgoing ray and interpolate these during rendering, but general reflection models remain future work.

Acknowledgements

This work was partially supported by the FP7 European Project Harvest4D and the IVCI at Saarland University.

REFERENCES

[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *ACM Trans. Graph.*, 22(3):605–612, 2003.

[2] K. M. Beason, J. Grant, D. C. Banks, B. Futch, and M. Y. Hussaini. Pre-computed illumination for isosurfaces. In *Conference on Visualization and Data Analysis*, pages 1–11, 2006.

[3] D. Burke. Bidirectional importance sampling for illumination from environment maps. Master's thesis, UBC, 2004.

[4] P. Clarberg and T. Akenine-Möller. Practical product importance sampling for direct illumination. *Computer Graphics Forum (Proceedings of Eurographics)*, 27(2), 2008.

[5] P. Clarberg, W. Jarosz, T. Akenine-Möller, and H. W. Jensen. Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Trans. Graph.*, 24(3):1166–1175, 2005.

[6] D. Cline, P. K. Egbert, J. F. Talbot, and D. L. Cardon. Two Stage Importance Sampling for Direct Lighting. In *Eurographics Symposium on Rendering*, 2006.

[7] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 189–198. ACM, 1998.

[8] I. Georgiev, J. Křivánek, T. Hachisuka, D. Nowrouzezahrai, and W. Jarosz. Joint importance sampling of low-order volumetric scattering. *ACM Transactions on Graphics (TOG)*, 32(6):164, 2013.

[9] M. Hadwiger, A. Kratz, C. Sigg, and K. Bühler. GPU-accelerated deep shadow maps for direct volume rendering. In *Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics Hardware*, GH '06, pages 49–52. ACM, 2006.

[10] D. Jönsson, E. Sundén, A. Ynnerman, and T. Ropinski. A Survey of Volumetric Illumination Techniques for Interactive Volume Rendering. *Computer Graphics Forum*, 33(1):27–51, 2014.

[11] T.-Y. Kim and U. Neumann. Opacity shadow maps. In *In Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 177–182. Springer-Verlag, 2001.

[12] T. Kollig and A. Keller. Efficient illumination by high dynamic range images. In *Rendering Techniques*, volume 44, pages 45–51. Eurographics Association, 2003.

[13] T. Kroes, F. H. Post, and C. P. Botha. Exposure render: an interactive photo-realistic volume rendering framework. *PLoS ONE*, 7(7), 2012.

[14] J. Kronander, D. Jönsson, J. Löw, P. Ljung, A. Ynnerman, and J. Unger. Efficient Visibility Encoding for Dynamic Illumination in Direct Volume Rendering. *IEEE TVCG*, 18(3):447–462, 2012.

[15] F. Lindemann and T. Ropinski. Advanced light material interaction for direct volume rendering. In *IEEE/EG International Symposium on Volume Graphics*, pages 101–108. Eurographics Association, 2010.

[16] P. Ljung, C. Lundström, and A. Ynnerman. Multiresolution interblock interpolation in direct volume rendering. In *Proc. of EuroVis*, pages 259–266. Eurographics Association, 2006.

[17] T. Lokovic and E. Veach. Deep shadow maps. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 385–392. ACM Press/Addison-Wesley Publishing Co., 2000.

[18] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[19] H. Niederreiter. *Random Number Generation and quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[20] J. Novák, A. Selle, and W. Jarosz. Residual ratio tracking for estimating attenuation in participating media. *ACM Transactions on Graphics (TOG)*, 33(6):179, 2014.

[21] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23(3):488–495, 2004.

[22] D. Patel, V. Šoltészová, J. M. Nordbotten, and S. Bruckner. Instant convolution shadows for volumetric detail mapping. *ACM Transactions on Graphics (TOG)*, 32(5):154, 2013.

[23] M. Pharr and G. Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., 2nd edition, 2010.

[24] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Trans. Graph.*, 22(3):340–349, 2003.

[25] C. Rezk Salama. GPU-based monte-carlo volume raycasting. In *Proc. Pacific Graphics*, pages 411–414, 2007.

[26] T. Ritschel. Fast GPU-based Visibility Computation for Natural Illumination of Volume Data Sets. In P. Cignoni and J. Sochor, editors, *Short Paper Proceedings of Eurographics 2007*, pages 17–20, 2007.

[27] T. Ropinski, J. Kasten, and K. H. Hinrichs. Efficient shadows for GPU-based volume raycasting. In *Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2008)*, pages 17–24, 2008.

[28] T. Ropinski, J. Meyer-Spradow, S. Diepenbrock, J. Mensmann, and K. Hinrichs. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Comput. Graph. Forum*, 27(2):567–576, 2008.

[29] F. Rousselle, P. Clarberg, L. Leblanc, V. Ostromoukhov, and P. Poulin. Efficient product sampling using hierarchical thresholding. *The Visual Computer*, 24(7-9):465–474, 2008.

[30] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 527–536. ACM, 2002.

[31] E. Sundén, A. Ynnerman, and T. Ropinski. Image Plane Sweep Volume Illumination. *IEEE TVCG(Vis Proceedings)*, 17(12):2125–2134, 2011.

[32] E. Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 419–428. ACM, 1995.

[33] L. Williams. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12(3):270–274, 1978.

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