

Efficient Volume Illumination with Multiple Light Sources through Selective Light Updates

Erik Sundén*
Interactive Visualization Group,
Linköping University, Sweden

Timo Ropinski†
Visual Computing Research Group,
Ulm University, Germany



Figure 1: A CT scan of a mummy ($438 \times 289 \times 1700$ voxels) rendered using the proposed volumetric illumination method. To enable high contrast in bright and shadowed areas, we use a light setup that consists of two warm (orange and red) point lights and two directional white lights.

ABSTRACT

Incorporating volumetric illumination into rendering of volumetric data increases visual realism, which can lead to improved spatial comprehension. It is known that spatial comprehension can be further improved by incorporating multiple light sources. However, many volumetric illumination algorithms have severe drawbacks when dealing with multiple light sources. These drawbacks are mainly high performance penalties and memory usage, which can be tackled with specialized data structures or data under sampling. In contrast, in this paper we present a method which enables volumetric illumination with multiple light sources without requiring precomputation or impacting visual quality. To achieve this goal, we introduce selective light updates which minimize the required computations when light settings are changed. We will discuss and analyze the novel concepts underlying selective light updates, and demonstrate them when applied to real-world data under different light settings.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1 INTRODUCTION

Since interactive volume rendering has matured, several researchers have focused on advanced volumetric illumination models. These models not only enable a more realistic representation of volumetric data sets, recent studies also indicate the perceptual benefits which come along with these models [17]. This discovery, together with more general perceptual findings which suggest that spatial perception is heavily dependent on the used illumination model [15], have motivated researchers to push the boundaries of advanced volume illumination. Novel models have been designed to support various

realistic lighting phenomena, such as shadowing [6], multiple scattering [8], or advanced light material interactions [16]. While these techniques have advanced the degree of realism in volume rendered images, the integration of multiple light sources as one crucial aspect has received relatively little attention.

Multiple light source setups are an important factor when aiming for realistic scene representations. They enable increased contrast in both shadowed and illuminated scene parts, and thus result in a more vivid representation (see Figure 1). Besides this impact on realism, multiple light source setups are also associated with improved spatial comprehension [5]. These advantages are widely exploited through according lighting setups in photography, while similar benefits could also be observed and exploited in visualization research. Halle and Meng could for instance show the improved perceptual benefits of complex light setups in visualization [7], and Šoltészová et al. found evidence that contrast rich colored shadow regions improve perceptual qualities of volume rendered images [29]. While these and other findings demonstrate the importance of complex light setups in visualization, there are unfortunately several technical challenges to be solved to support these setups in an efficient manner.

Many volumetric illumination techniques have been designed by taking into account a tradeoff between rendering time and memory usage, as precomputing illumination-relevant information has a positive impact on rendering performance. Unfortunately, at the same time it results in an increased memory footprint, which is often prohibitive, especially when handling large data sets. Dealing with multiple light sources poses an additional challenge, as more than one light source needs to be taken into account either during precomputation or during rendering, which naturally results in a performance decrease. Beyond this, more data needs to be allocated to represent the illumination for each individual light source. Furthermore, as current volumetric illumination techniques mostly follow an all-or-nothing approach when it comes to light updates, frame update times and memory consumption are often proportional to the total number of light sources rather than the number of changed light sources.

*erik.sunden@liu.se

†timo.ropinski@uni-ulm.de

To reduce the impact of these factors, we exploit two main observations which can be made when dealing with multiple light source setups.

Coherent propagation. While light sources may illuminate a scene from opposite directions, the more lights that are added to a scene, the higher is the similarity of the illumination directions, which enables the exploitation of performing the propagation of multiple coherent lights simultaneously.

Sequential editing. As in photography, also when designing multiple light setups in the context of visualization, light sources are usually edited in a sequential manner.

In this paper, we propose *selective light updates*, a novel approach for dealing with complex lighting setups in interactive volume rendering. By taking into account the observation that often only a few light sources of a complex light setup are changed interactively, selective light updates minimize the necessary recalculations. This becomes possible by exploiting a novel propagation scheme, *coherence-based light propagation*, which incrementally generates and maintains a global, and selectively updatable, illumination volume. Thus, we can obtain fast rendering times, without requiring additional memory for caching individual light source information. In practice this means that independent of the number of light sources, change of a single light source does not require recomputation of the other none-modified light sources. Furthermore, even when recalculating the entire illumination information, our coherence-based propagation results in a significant speedup. At the same time we reduce the size of the GPU memory footprint substantially, in comparison to previous approaches.

2 RELATED WORK

In recent years, several interactive volume illumination techniques have been proposed with the goal to improve realism and the perceptual qualities of volume rendered images [9]. In the following paragraphs we discuss these related approaches based on the number of supported light sources.

No light source. Stewart was the first author who proposed illumination-like enhancements of volume rendered images without requiring to setup individual light sources [26]. His vicinity shading technique takes into account the occlusion of a voxel to represent the amount of light energy it is exposed to. While this works only for iso-surfaces, Ropinski et al. presented how to interactively realize ambient occlusion for Direct Volume Rendering (DVR) [22]. Independently, Diaz et al. proposed the vicinity occlusion shading technique, which exploits summed area tables to achieve similar visual effects in volume rendered images [4].

Single light source. Most volume illumination techniques have been designed to support only a single light source. This is most likely due to the fact, that with a single light source setup it is possible to propagate illumination simultaneously during rendering, a concept first proposed by Kniss et al. [12]. Schott et al. applied a similar proceeding to support head lights [24], which have been identified as perceptually most effective for volumetric illumination [17]. Šoltészová et al. extended this directional occlusion shading method to support light sources at an arbitrary position inside the viewer's hemisphere [28]. While all these approaches facilitate slice-based volume rendering, other approaches employ a similar procedure when using volume rendering based on ray-casting [27]. Some similar techniques can even be combined with arbitrary volume rendering techniques [6, 21].

Multiple light sources. The method proposed in this paper is not the first one that enables the incorporation of multiple light sources in DVR. While many of those techniques originally proposed for single light setups, that do not exploit the simultaneous light propagation mentioned above, can be extended to support multiple light

sources, this results in a penalty of rendering time or memory, or both. In this paragraph we discuss those techniques which have originally been proposed as being able to support multiple light sources.

Spherical harmonics (SH) are frequently used in computer graphics to support the interactive application of low frequency environmental lighting. Ritschel was the first who has applied this concept in the area of DVR [19]. Later, Lindemann and Ropinski have shown how to use this approach also to incorporate advanced material effects [16] and Kaplanyan et al. [10] used SHs and introduce light propagation volumes to achieve indirect lighting. Kronander et al. described how to improve the performance of the SH computation by exploiting space partitioning data structures [14]. However, SH techniques are limited to low frequency lighting. Accordingly, it is not possible to exploit the perceptual benefits which are often associated with hard shadow borders [31].

Photon mapping has also been applied in the context of volume rendering. Jönsson et al. were the first who have realized interactive volumetric illumination through photon mapping [8]. While their approach enables multiple light sources, data sizes become prohibitive as the size of the used data structure is directly dependent on the size of the volume and the number of traced photons. In a recent approach by Zhang et al. [33] they directly address the usage of multiple light sources, which is supported by introducing certain basis functions for light projection. However, the basis functions needs to be precomputed. Monte Carlo ray-tracing has been proposed as another alternative for volumetric illumination [13], as it also enables realistic volume illumination with multiple light sources. Unfortunately, Monte Carlo based methods come with a performance penalty that becomes visually apparent through the progressive nature of the algorithm. When the user interacts with the scene, the initial image needs to be refined over time before a crisp image appears. To cope with such issues, several researchers have presented caching techniques. Weber et al. [32] present an interactive approach of a multiple light setup by approximating and caching light transport with a set of virtual lights, and progressively updating them when the transfer functions changes. Schlegel et al. [23] use summed area tables (SATs) to speed up the computation of dynamic lighting. However, high frequency illumination effects are not supported, due to a use of a low-resolution SAT. A recent article by Ament et al. [1] utilize various SATs, with different sampling patterns, such as cuboid and sphere-based sampling. They support coherent calculation of up to six orthodirectional light sources. However, if one light sources changes all light contributions has to be recomputed. To precompute lighting and support interactive updates, Zhang et al. [34] utilize a convection diffusion equation. However, they only support soft shadows due to the use of a first-order upwind scheme.

The specific use of multiple lights for improving visual perception in volume rendering is also acknowledged by other authors. Wang and Kaufman propose a lighting system together with a light design algorithm, targeted towards the perceptual improvement of volume rendered images [30]. A similar goal has been targeted by Zhang and Ma, who propose a lighting design system for automatically generating three-point light setups [35]. As these systems mainly focus on the lighting setup, the described concepts can be combined with the proposed approach to generate an initial light setup.

To the best of our knowledge, the proposed method is the first which enables the integration of multiple colored light sources in an interactive volume illumination setup together with a selective update process where only the contribution of changed light sources are recalculated. The required storage is constant with the number of light sources and it supports both low and high frequency illumination effects.

3 VOLUME ILLUMINATION MODEL

In this section, we detail the volumetric illumination model underlying our approach. We start with a requirement analysis, detailing the desired lighting capabilities. As stated in Section 1, our main goal is to support multiple light setups, as they are perceived as more realistic and support a better scene perception [5]. The most popular and also most basic multiple light setup used in photography is the *three-point lighting*, used in the setup of Figure 2 (c), which is often used in portrait photography and known to bring out details and emphasize 3D structures in the scene. *Butterfly lighting*, utilized in the rendering shown in Figure 2 (b), uses only two lights, whereby one is placed on top of the main object, and one which serves as a rim light, is placed beneath. The importance of these and other lighting setups, together with the impact of complex light scenarios on the perception of visualizations [7], was a great motivation for us to achieve similar effects with volumetric illumination.

A general rule underlying multiple lighting scenarios is that each individual light source has its special purpose. Thus, often accessory lights are added to provide contrast in certain scene parts or to fill the background. We see this fact nicely correlating with a study conducted by Šoltészová et al., where they show that chromatic shadows, an effect easily achieved with strategically placed accessory lights, have a positive impact on the perception of volume rendered images [29].

While these observations underline the importance of the interplay of different light sources in DVR, the way each light source illuminates the scene is also of importance. As it was shown in earlier studies, both diffuse lighting [15] and hard shadow borders [31, 21] have a positive effect on scene perception. Finally, colored lights can be used to emphasize moods in a picture. While this effect plays an important role in photography, it might be of lesser importance in the area of visualization. Nevertheless, all the criteria discussed in this paragraph motivated us to design the proposed illumination approach.

To see how to incorporate these multiple light source effects into interactive DVR, we consider the standard volume rendering integral with single scattering extended for multiple light support. While its most basic form only incorporates emission and absorption, it does not account for any external light sources or indirect illumination effects. Therefore, we follow Max’s theory on volumetric illumination models [18], and add a single scattering term, thus that we can rewrite the standard volume rendering integral as:

$$I(x_e, \vec{\omega}_o) = T(x_b, x_e) \cdot I(x_b, \vec{\omega}_o) + \int_{x_l}^{x_e} T(x', x_e) \cdot \rho(x', \vec{\omega}_o) dx', \quad (1)$$

where $I(x_e, \vec{\omega}_o)$ denotes the radiance reaching the eye point x_e from direction $\vec{\omega}_o$, $T(x_0, x_1)$ depicts the extinction between x_0 and x_1 , and $\rho_i(x, \vec{\omega}_o)$ can be considered as the local shading function that reflects light into direction $\vec{\omega}_o$. To consider the single scattering contribution, we define $\rho(x, \vec{\omega}_o)$ as follows:

$$\rho(x, \vec{\omega}_o) = \tau(x) \cdot \left(q(x) + \int_{\Omega} s(x, \vec{\omega}_i, \vec{\omega}_o) \cdot T(x_l, x) \cdot I(x_l, -\vec{\omega}_i) d\vec{\omega}_i \right) \quad (2)$$

Here, $\tau(x)$ and $q(x)$ are the local absorption and emission properties at x , and $s(x, \vec{\omega}_i, \vec{\omega}_o)$ is the scattering function at x that is dependent on the incoming light $\vec{\omega}_i$ as well as the outgoing light $\vec{\omega}_o$. To take into account all potential light sources, we integrate over all directions Ω and use the extinction of the radiance $I(x_l, -\vec{\omega}_i)$ coming from the individual directions $-\vec{\omega}_i$. We negate the direction here to be consistent with Equation 1 and denote that this is radiance leaving from x_l , being the position of the respective light source.

When dealing with multiple light sources, the integral in Equation 2 is the compute-intensive part, as it has to be solved for all

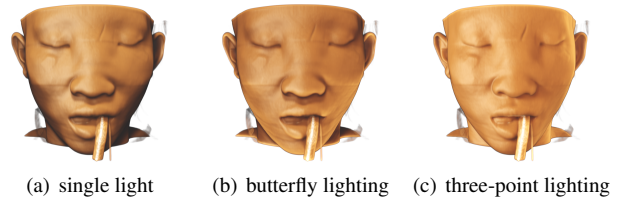


Figure 2: Application of our technique with lighting setups used in photography. We show the setups butterfly lighting (b) and three-point lighting (c) as compared to a single point light setup (a).

samples along all viewing rays. However, when dealing with multiple point light sources, as it is done in computer graphics when using virtual point light sources [11], the integral can be simplified, as one needs to integrate only over the light sources in the scene. A similar observation holds for directional light sources. Nevertheless, when using this simplification still the extinction $T(x_l, x)$ needs to be solved for each individual light source, which results in a vast performance impact, dependent on the total number of lights.

The observation of coherent propagation enables us to combine several light sources when computing $T(x_l, x)$. Kniss et al. were the first to show that a slightly offset light direction results in convincing results when propagating light information [12]. In our work, we exploit this finding in a systematic manner, by limiting the number of light propagation directions to a maximum of six, which covers the principal positive and negative axis directions of a volume. Thus we can rewrite Equation 2 as follows:

$$\rho_{\vec{d}}(x, \vec{\omega}_o) = \tau(x) \cdot \left(q(x) + \int_{\Omega} g_{\vec{d}}(x, \vec{\omega}_i) \cdot s(x, \vec{\omega}_i, \vec{\omega}_o) \cdot T(x_l, x) \cdot I(x_l, -\vec{\omega}_i) d\vec{\omega}_i \right), \quad (3)$$

where \vec{d} is one of the principal volume axes $[-x, +x, -y, +y, -z, +z]$, and $g_{\vec{d}}(x, \vec{\omega}_i)$ is a weighting factor that depends on the incoming light direction $\vec{\omega}_i$, which is defined as follows:

$$g_{\vec{d}}(x, \vec{\omega}_i) = \begin{cases} 1 & \text{if } \angle(\vec{\omega}_i, \vec{d}) < \Theta_{\vec{d}} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Here, $\Theta_{\vec{d}}$ is a 90° open angle associated with each principal axis direction $[-x, +x, -y, +y, -z, +z]$, and $\angle(\vec{\omega}_i, \vec{d})$ denotes the angle between the incoming light direction $\vec{\omega}_i$ and \vec{d} .

Several previous approaches have integrated the thus exploited offset light direction in the context of slice-based volume rendering to achieve a single illumination pass supporting a single light source, e. g., [24, 28]. Since we aim to cover a multitude of light sources, potentially located at opposite ends of the volume, we cannot combine these within a single rendering pass. Thus, for each x we compute $\rho_{\vec{d}}(x, \vec{\omega}_o)$ up to six times depending on the distribution of light positions, encoded in $g_{\vec{d}}(x, \vec{\omega}_i)$.

Based on these computations, we propagate the light for the up to six directions, and combine the results in an illumination volume, as it has been proven viable for single light sources before [3, 20]. While this procedure enables us to achieve a constant maximum number of propagation passes, independent of the number of light sources (see Figure 3), it will be still more costly than just considering a single light source. To resolve this, we take into account our second observation describing the sequential editing of light sources.

Using an illumination volume with sequential editing of light sources enables us to realize selective light updates by exploiting the additive nature of light. The key idea is that if light can be added to an illumination volume during a light propagation pass, it can consequently also be removed. Thus, while utilizing coherent propagation, we cannot only add light via propagation, but we can also cancel it out by subtracting its energy. In practice this means that we can modify a subset of a multiple light source setup for a given $\Theta_{\vec{d}}$ in constant time independent of the total number of light sources. Accordingly, our algorithm enables interactive editing of single light sources in multiple light source setups in constant time. Furthermore, our storage requirement for the resulting illumination information is independent of the total number of light sources.

In the next section we describe the underlying coherence-based light propagation in more detail, whereby we discuss the selective light update strategy in Section 5.

4 COHERENCE-BASED LIGHT PROPAGATION

Our foundation for selective light updates is a novel coherence-based light propagation scheme that enables a single illumination volume to be used even for multiple light sources. We minimize the required calculations by performing simultaneous light propagations, which exploit the coherence between different light sources. Our method supports directional, point and spot light sources through a coherence-based scheme that takes the correlation of the light sources' directions into account. For the point and spot light sources the correlation is based on the main direction which we define as the direction from the position of the light source $l_{pos}(i)$ to the center of the volume. Thus, the directional component $l_{dir}(i)$ for the i th point light can be written as $l_{dir}(i) = v_{center} - l_{pos}(i)$ with v_{center} being the geometric center of the volume. To associate the individual light source's directions with the propagation direction \vec{d} as specified in Equation 3 we compute $h_{\vec{d}}(l_{dir}(i)) = l_{dir}(i) \cdot -n_{v_{face}}$, where $j \in [0..5]$ denotes the identifier for one of the six faces of the volume. For each light source we maximize $h_{\vec{d}}$ and thus obtain the relevant propagation group, which groups all relevant light sources for the current propagation direction. Each propagation group has between 0 and n lights associated with it, where n corresponds to the total number of light sources.

All lights associated to a propagation group are processed simultaneously, and unnecessary computations are avoided by discarding empty propagation groups before processing. Then the actual propagation is performed axis-aligned in volume space, which ensures a traversal of all voxels in sequential order. This is of importance, as it avoids race conditions when updating the illumination information for the current position of the illumination volume. Furthermore, this considerably reduces the data transfer bandwidth requirements and thus has a large positive impact on the overall performance.

To propagate the light without requiring an additional volume as an intermediate cache, we exploit a pair of 2D read/write buffers for each light source. The read buffer is used to retrieve the incoming radiance from the previous propagation step, while the write buffer is used to store the outgoing radiance from the current location. Therefore, we compute the outgoing radiance $l_i(\vec{x}_i, \vec{\omega})$ for each individual voxel as $l_i(\vec{x}_i, \vec{\omega}) = (1 - \alpha(x_i)) \cdot l_i(\vec{x}_{i-1}, \vec{\omega}) + \alpha(x_i) \cdot c(x_i)$, where $l_i(\vec{x}_{i-1}, \vec{\omega})$ represents the light intensity that reached the previous voxel in direction $\vec{\omega}$ from the light source, $\alpha(x_i)$ the material opacity at x_i and $c(x_i)$ the material color at x_i . We would like to point out that, while previous approaches had to generate one illumination volume per light source [33], our novel scheme only requires a single illumination volume and at the same time enables selective light updates, which we detail in the upcoming section.

At the end of each propagation step, the content of the write buffer is added to the appropriate slice in the resulting illumination volume, which might already contain light contributions from other propagation passes. Afterwards, the read and the write buffer are

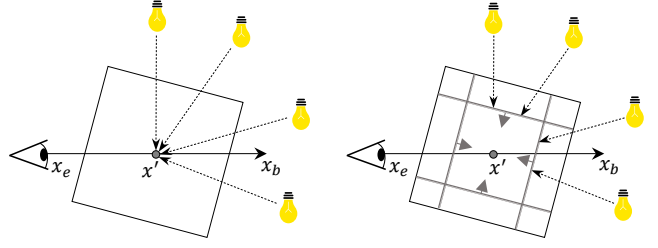


Figure 3: By exploiting coherence-based light propagation, we achieve a constant maximum number of propagation passes independent of the number of light sources. While in standard multi-light rendering approaches (left) all light sources and their visibility would be evaluated per sample during rendering, the coherence-based propagation (right) groups light sources into propagation groups associated with the principal axis of the volume.

swapped until all steps along the propagation direction have been processed. Once all steps for all propagation directions are processed, the illumination volume contains the complete light contribution from all light sources in the scene. During this process, the read/write buffers are, besides the illumination volume, the only required memory allocations. As a read/write buffer typically has the same resolution as the volumes cross-section along the propagation axis their memory footprint is rather small. Thus, we are able to realize the integration of a large number of light sources without reaching the memory limits of modern GPUs.

To further optimize light propagation, we sort the propagation groups prior propagation. As we employ read/write buffers, additional memory reallocations might become necessary when the sizes of these buffers change. To avoid these time-consuming allocations, the propagation groups are sorted such that propagations along the same axes, in positive and negative direction, are performed consecutively.

5 SELECTIVE LIGHT UPDATES

As expressed in our second observation in Section 3, it is common that only a few of the lights present in a scene are changed simultaneously. Therefore, based on the coherence-based light propagation scheme described in the previous section, we introduce selective light updates, which only update light emitted from modified light sources. The in-place update mechanism of the illumination volume detailed above, together with a high dynamic range illumination volume are essential to realize these selective updates. With these capabilities, we are able to add and subtract light energy to/from the illumination volume in a single propagation pass. Thus, we can enable a change in position, direction or color of a light source, whereby the light contribution from the light source at the state after the change is added to the illumination volume, and the light contribution from the light source at the state before the change is subtracted from the illumination volume.

To realize adding and subtracting light contributions, we modify the sign of the contribution factor α , as detailed in Equation 5, such that it becomes negative rather than of positive for the light source being removed. Thus, the light contribution added in a previous state for the changed light source can be removed without modifying the propagation pipeline. Furthermore, the operation of turning on and off a light source is solved with even less computations as only one subtraction or addition operation needs to be performed for the switched light source.

In our method, we define the term of initial light contribution as the initial light intensity $l_0 = l_p \cdot l_c(r, g, b)$, modulated with light power l_p and light color l_c , through which we take into account the absorption of light in a high dynamic range setup. The high dy-

dynamic range lighting setup is essential to enable the update of a light independent of other static lights in the scene, as an arbitrary number of lights can be present at various locations and with varying light power, as seen in Figure 4a and 4b. The contribution from all light sources would need to be balanced against each other to compensate for an insufficient dynamic range, which makes independent update of arbitrary lights impossible.

Cost estimation. To take maximum advantage of selective light updates, the projected performance of changing the scene needs to be taken into account. For instance, when two light sources are present in the scene, the projected performance of doing a selective update may be lower or higher than the projected performance of an entire light update (recalculation of the contribution from all lights in the scene), as a selective update of one light requires propagations of two lights (before and after the change), which is the same as an entire light update for the current setup. In order to deal with these, and similar scenarios, the selective update is performed only if the estimated performance determines that a selective update is less costly than an entire light update. Therefore, we have developed different cost measures.

Our least complicated cost estimation is performed by determining the percentage of how many of the available light sources have changed. If 50 percent or less have changed, a selective update is considered preferable over an entire light update. This cost is based on the fact that a selective update of a changed light source is treated as propagation of two light sources. However, it should be noted that the selective update is preferable when the percentage is exactly 50, due to the fact that in most scenarios the add and remove operations are performed along the same propagation direction, which enables the propagation coherence, explained in Section 4, to be used.

We can further extend our cost estimation to focus on the number of propagation passes needed to determine if a selective light update is more preferable than an entire light update. Thus, it is based on propagation coherence and uses the same assumptions as above. However, it is preferable to utilize selective updates when the number of propagations is lower than the number of propagations for an entire light update, even if the number of lights changed are more than 50 percent of the total number of lights.

A detailed evaluation of the impact on different light setups is discussed in Section 7.

6 TECHNICAL REALIZATION

We have implemented the proposed concepts utilizing the OpenGL API, whereby we keep the illumination volume as a 4 channel 32-bit 3D texture, and the read/write buffers are 4 channel 32-bit 2D textures. We could also store all illumination data as single channel data, but then it would not be possible to propagate colored lights. To be able to realize random read/write operations in the illumination volume in the fragment shader stage, OpenGL 3.0 and above is required for our technical realization. Besides the memory reduction, the in-place illumination propagation also results in improved performance, due to the faster read/write operations for 2D textures, compared to 3D textures.

For the coherence-based light propagation detailed in Section 4, we manage the processing of lights in the same propagation group during one propagation pass by attaching all read and write caches to one frame buffer object and binding all caches to individual texture units. Thus, we can exploit multiple render targets to update the write caches, and perform a ping-pong swap between the texture units and the attachment IDs, without the need to change the frame buffer object. By varying the number of render targets a trade-off between performance and memory can be made. If the number of lights in one propagation pass is larger than the number of render targets, the propagation group is divided into multiple groups.

The propagation follows the scheme proposed by Ropinski et

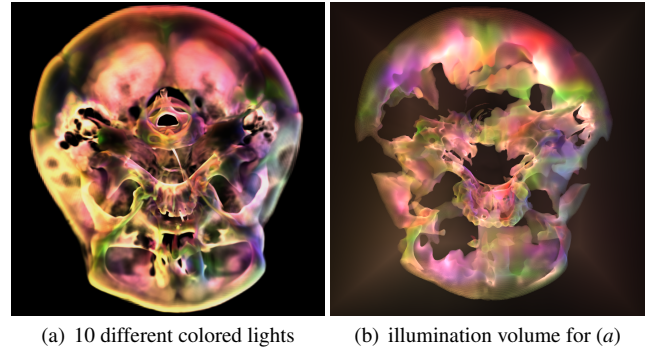


Figure 4: Rendering of a human skull (a) and associated light volume (b) with 10 light sources using our method.

al. [20], thus the propagation steps are briefly presented here. During the propagation, axis permutation is applied in the form of permutation matrices, which enables the possibility to propagate in the illumination volume in all axis-aligned directions, positive and negative, without requiring specific sampling code per axis. To avoid popping artifacts which would occur when a light source changes its propagation group, a light source is represented in two propagation groups, corresponding to the propagation along the normal of the closest and the second closest volume face to the light source. Before the light is added to the illumination volume a contribution factor is applied, as detailed in Equations 5a and 5b:

$$\alpha_{cA} = 1 - \frac{2 \arccos(l_{\text{pos}} \cdot -n_{F_l})}{\pi} \quad (5a)$$

$$\alpha_{scA} = 1 - \alpha_{cA} \quad (5b)$$

The contribution factor describes the contribution of light between the two propagation passes and is based on the angle β between the normal of the closest volume face n_{F_l} and the light source l_{pos} .

Light initialization. Currently, we support three different light source types. These light types are directional, point, and spot light sources. Within our algorithm, directional light is the most straightforward to handle, as the direction towards the light is always the same. Due to the lack of a light position, the direction towards the point and spot light needs to be calculated at every position inside the volume. Furthermore, when performing propagation of a spot light, the cone angle needs to be accounted for.

To optimize and simplify the propagation, we initiate the read buffer to match the projection of the light as shown in Figure 5. For directional lights, the read cache is initialized with a constant color based on the color and intensity of the light. For point and spot lights, the read cache is generated by projecting the light with projective texture mapping [25] towards a plane where the propagation will start, before spreading out the color contribution. Thus, both point and spot light can be treated the same way during propagation, and we do not need to perform any specific operation for the first propagation step.

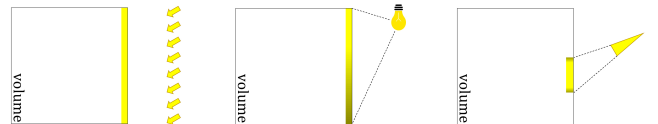


Figure 5: For the supported light source types (directional, point, and spotlight (from left to right), we project the light sources on the initial volume slice before we perform our coherence-based propagation.

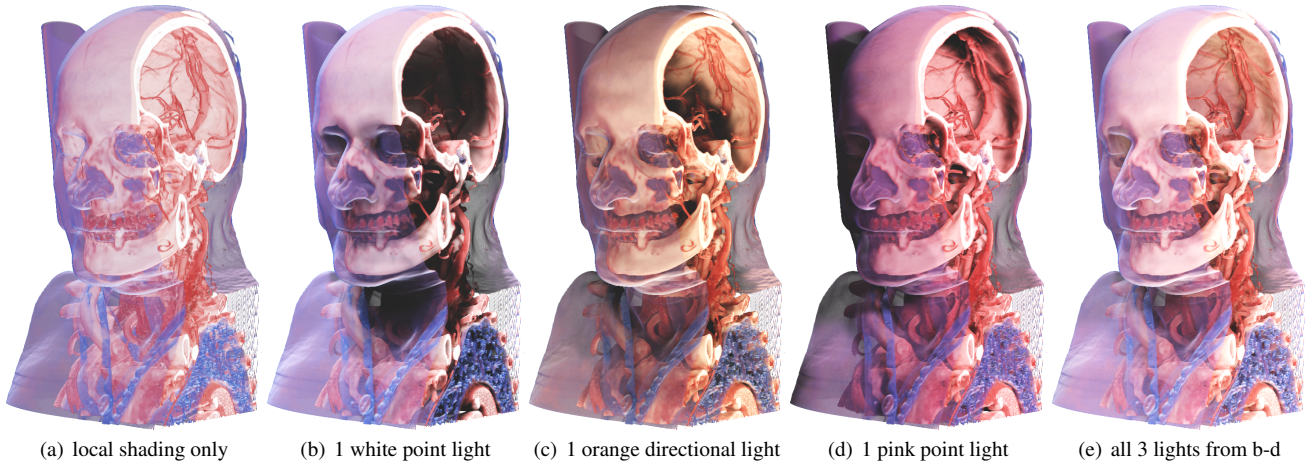


Figure 6: Renderings of CT data set of a male's upper body. In (a) only local shading is used with the Ashikhmin-Shirley model [2]. One light has been added in (b), (c) and (d) respectively, with varying characteristics of light type, direction and color. All the three light sources from (b), (c) and (d) are then used together in (e), which is rendered with our volume illumination approach that supports multiple light sources.

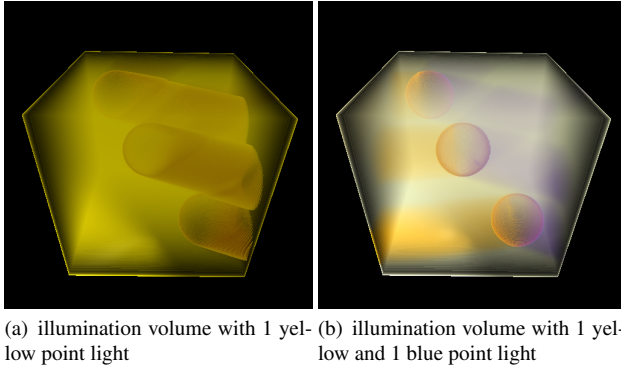


Figure 7: Light contribution to a scene. (a) shows the illumination volume, of three spheres, under lighting from one yellow point light from the left of the volume, relative to the camera. In (b) an additional blue point light has been added, whose location is on the right side of the volume, thus there are areas to the left of the spheres where the blue light is not present, and areas to the right of the spheres where the yellow light is not present.

7 RESULTS & DISCUSSION

To demonstrate the utility of the presented approach, we have produced visual results and performance measurements as well as discussed limitations, which is outlined in the following subsections.

7.1 Visual Results

To validate the correctness of the presented light propagation algorithm, we have applied it to a simple synthetic scene containing of three spheres (see Figure 7). We compare the achieved effects when applying lighting from one yellow point light from the left (Figure 7 (a)) to adding an additional blue point light from the right (Figure 7 (b)). As can be seen, there are areas to the left of the spheres not covered by the blue light, and areas to the right of the spheres where the yellow light is not covered accordingly.

Figure 6 show renderings of a male's upper body, with various lighting setups. The setup in Figure 6 (a) contains only local shading [2]. In Figure 6 (b), (c) and (d) we use the same setup as (a) but add global illumination in form of one light source with different characteristics. Finally, in Figure 6 (e) we combine all previously

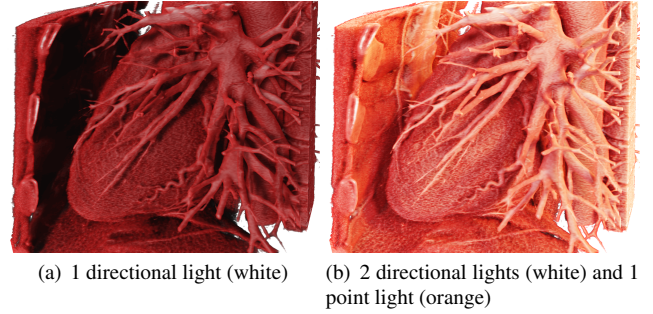


Figure 8: When applied to complex 3D structures, the use of multiple light sources not only increases the level of realism, but also the contrast in lit and shadowed areas, which emphasize the structures.

introduced light sources to show the correctness of our technique. We can see that the combination in (e) illuminates almost all areas of the volume, while still providing sufficient depth cues.

Figure 8 (a) shows renderings of a heart data set with only one light source, which is similar to what can be achieved with other high-quality propagation methods [20, 27]. In Figure 8 (b) the light setup has been change of accommodate three different lights. The fill lights (directional) are utilized to increase contrast so that previously invisible structures can emerge more prominently.

As we can practical achieve and maintain an interactive and exploratory multiple light source setup efficiently, we further provide visual results of rendering including various multiple light source setups to motivate our method.

Figure 1 shows a rendering of a mummy with a lighting setup consisting of four different lights. Two white lights are placed slightly below and above the mummy, relative to the camera, which resembles the butterfly lighting setup. The remaining lights (one orange and one red), are introduced to enhance the warm materials.

Finally, Figure 9 shows two renderings of the classic engine data set under a light setup of three different lights, with varying light color, to show the influence of this light parameter. Two of the lights in Figure 9 (a) are white and the third is yellow/orange, while all three lights in in Figure 9 (b) varying bright blue colors. As can be seen, the engine can be perceived as warm in in Figure 9 (a) while cool tones dominate in Figure 9 (b).

Data	Data Volume Resolution (DR)	Illumination Volume Resolution	Light Count	Entire Light Update		Selective Light Update (One light)	
				Sequential propagation	Coherence-based propagation	Coherence-based propagation	State change
Engine	256x256x256	DR * 1	2	129 ms	114 ms	72 ms	30 ms
		DR * 1	3	192 ms	124 ms	72 ms	30 ms
		DR * 1/4	2	22 ms	18 ms	11 ms	5 ms
		DR * 1/4	3	30 ms	20 ms	11 ms	5 ms
Heart	512x512x338	DR * 1	3	1093 ms	747 ms	430 ms	183 ms
		DR * 1/4	3	153 ms	102 ms	58 ms	27 ms
Male	512x512x460	DR * 1/4	4	273 ms	152 ms	78 ms	35 ms
Mummy	438x289x1700	DR * 1/4	4	533 ms	340 ms	170 ms	64 ms

Table 1: Performance measurements of our illumination calculations for various light setups and data sets. The numbers show the performance increase achieved with coherence-based propagation when all lights have to be recomputed (entire light update) and for selective light update of only one light. For the updates, we differentiate between two cases, light state changes (e.g., color, position or type) and toggling a light.

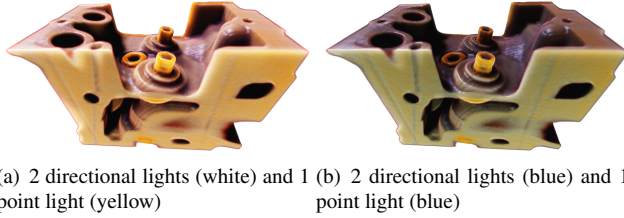


Figure 9: The engine data set rendered with our approach and colored light setups to shift the image tone from warm (a) to cool (b).

7.2 Performance Measurements

We have further analyzed the performance outcome of coherence-based light propagation and selective light update independently. The results are shown in Table 1, where we have applied the approach to different data sets, with varying number of light sources, varying illumination volume and buffer sizes, and with 1 or 1/4 times the resolution of the actual data. For each of the resulting scenarios, we first analyze the gain of the coherence-based propagation by comparing it with sequential propagation carried out for each individual light source. Secondly, we analyze the benefit of selective light updates by measuring the performance of a single light update. As light state changes, e.g., changing a light’s color, position or type, require one update pass to remove and one update pass to add the light, we also measured the performance for switching lights on or off.

The measurements show that coherence-based light propagation increases the performance in all our real-world scenarios, where in the general case it is 1.5 times faster than sequential light propagation, which is defined as one light per propagation pass. However, if all propagation passes have only one light source, which is possible in a setup with 2-3 lights, due to the 6 possible propagation directions, the performance of coherence-based propagation would be identical to sequential propagation, for an update of all lights.

The measurements also show that selective light updates decrease calculation time further, and when combined with coherence-based light propagation, a state change performed on one light resulted in a speedup proportional to the number of light sources in the scene, and light toggling(on/off) additional twice as fast. The results are expected as a setup with many lights should gain substantial performance increase by utilizing selective light update for single light source changes.

Our least beneficial setup for selective light update, when a state change only results in 1.68 times speedup, occur when only two light sources are utilized, and even then, the calculation time for

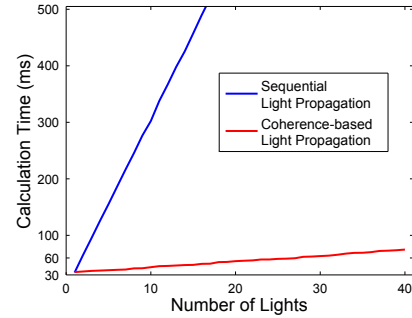


Figure 10: Performance measurements of our coherence-based propagation compared to sequential propagation. The results have been achieved for a data resolution of 256^3 for read and storage.

selective light updates are substantially faster than computing the contribution of all lights. This is the case, as the selective light propagation passes are almost always processed in the same pass, due to small directional/positional change of the one light that moves around in the scene. We have also performed a scaling test, where we measured the calculation time for a scene containing 1-40 lights, all with the same direction (see Figure 10). Based on the results of these measurements, we can see that calculation time for both methods increases linearly. However, while the sequential approach increases by 30 ms per added light, the increase for our coherence-based light propagation is only 1-2 ms per added light. All measurements were performed on a computer equipped with an Intel CPU (Xeon W3550 3.07GHz), 6 GB RAM and an NVIDIA GPU (Asus GTX670-DC2-4GD5G).

7.3 Limitations

The dynamic range of the illumination is limited to the storage precision, currently 32-bit in our setup. If the total light exceeds this precision, the intensity of all light has to be scaled by reducing them equally. Fortunately, we did not reach this limit in practice.

A spot light could be placed inside the volume without overhead if the cone angle does not result in propagation of more than two passes. However, propagating a point light located inside the volume is expensive as all six axis-aligned propagations are needed.

In computationally heavy situations, for instance in a setup of many light sources spread out in all directions and inside the volume, interactive exploration might not be possible when a recomputation of the contribution of all lights is needed. To achieve an interactive setup the illumination volume can be incrementally updated from a specific number of propagations per frame.

8 CONCLUSIONS & FUTURE WORK

Within this paper, we have proposed a novel approach for dealing with multiple light sources in the context of interactive volume rendering with global volumetric illumination. With coherence-based light propagation we could achieve performance improvements of at least 50 percent, and when combined with selective light updates the speedup increased proportional to the number of light sources present in the setup. A controlled scalability test performed by iteratively increasing the number of light sources could also show a speedup of up to 15 times.

Furthermore, the selective light updates enabled updating of only modified light sources, which results in constant performance independent of the number of used light sources. To increase the advantage of selective light updates we further have discussed different cost estimations, which regulate when it is advantageous to perform a selective update. While coherence-based light propagation is the foundation for the selective light update concept, both concepts benefit from an in-place light propagation which minimizes the required memory. The presented results show realistic illumination effects, with high and low frequency shadows, achieved under lighting setups with multiple light sources.

We have currently not addressed the integration of multiple scattering into our proposed concepts. Such an incorporation would yield additional calculation time and should optimally be made in such a way that independent addition and removal of multiple scattering is performed in a selective update. Furthermore, we would like to extend the selective updates to take structures inside the volume into account, thereby only updating affected regions.

Acknowledgements

The data set shown in Figure 1 is courtesy of Sectra AB, Sweden. The presented concepts have been developed and evaluated in the Inviwo framework (www.inviwo.org).

REFERENCES

- [1] M. Ament, F. Sadlo, C. Dachsbacher, and D. Weiskopf. Low-Pass Filtered Volumetric Shadows. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2437–2446, 2014.
- [2] M. Ashikhmin and P. Shirley. An Anisotropic Phong BRDF Model. *J. Graph. Tools*, 5(2):25–32, 2000.
- [3] U. Behrens and R. Ratering. Adding Shadows to a Texture-Based Volume Renderer. In *IEEE/EG Symp. on Volume Visualization*, 1998.
- [4] J. Díaz, H. Yela, and P.-P. Vázquez. Vicinity Occlusion Maps - Enhanced depth perception of volumetric models. In *Computer Graphics Int.*, pages 56–63, 2008.
- [5] R. W. Fleming, R. O. Dror, and E. H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, 3(5):3, 2003.
- [6] M. Hadwiger, A. Kratz, C. Sigg, and K. Bühler. GPU-Accelerated Deep Shadow Maps for Direct Volume Rendering. In *ACM SIGGRAPH/EG Conference on Graphics Hardware*, pages 27–28, 2006.
- [7] M. Halle and J. Meng. LightKit: A lighting system for effective visualization. In *IEEE Visualization*, pages 363–370, 2003.
- [8] D. Jönsson, J. Kronander, T. Ropinski, and A. Ynnerman. Histograms: Enabling Interactive Global Illumination in Direct Volume Rendering using Photon Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2364–2371, 2012.
- [9] 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.
- [10] A. Kaplanyan and C. Dachsbacher. Cascaded Light Propagation Volumes for Real-time Indirect Illumination. In *Proc. of SIGGRAPH Symp. on Interactive 3D Graphics and Games*, pages 99–107, 2010.
- [11] A. Keller. Instant Radiosity. In *Proc. of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 49–56, 1997.
- [12] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson. A Model for Volume Lighting and Modeling. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):150–162, 2003.
- [13] T. Kroes, F. H. Post, and C. P. Botha. Exposure Render: An Interactive Photo-Realistic Volume Rendering Framework. *PLoS ONE*, 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 Transactions on Visualization and Computer Graphics*, 18(3):447–462, 2012.
- [15] M. S. Langer and H. H. Bühlhoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29(6):649–660, 2000.
- [16] F. Lindemann and T. Ropinski. Advanced Light Material Interaction for Direct Volume Rendering. In *IEEE/EG International Symp. on Volume Graphics*, pages 101–108, 2010.
- [17] F. Lindemann and T. Ropinski. About the Influence of Illumination Models on Image Comprehension in Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2011.
- [18] N. Max. Optical Models for Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [19] T. Ritschel. Fast GPU-based Visibility Computation for Natural Illumination of Volume Data Sets. In *Eurographics Short Paper Proc.*, pages 17–20, 2007.
- [20] T. Ropinski, C. Döring, and C. Rezk Salama. Advanced Volume Illumination with Unconstrained Light Source Positioning. *IEEE Computer Graphics and Applications*, 30(6):29–41, 2010.
- [21] T. Ropinski, J. Kasten, and K. Hinrichs. Efficient Shadows for GPU-based Volume Raycasting. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 17–24, 2008.
- [22] T. Ropinski, J. Meyer-Spradow, S. Diepenbrock, J. Mensmann, and K. H. Hinrichs. Interactive Volume Rendering with Dynamic Ambient Occlusion and Color Bleeding. *Computer Graphics Forum*, 27(2):567–576, 2008.
- [23] P. Schlegel, M. Makhinya, and R. Pajarola. Extinction-based Shading and Illumination in GPU Volume Ray-Casting. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1795–1802, 2011.
- [24] M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, and K. Bouatouch. A Directional Occlusion Shading Model for Interactive Direct Volume Rendering. *Computer Graphics Forum*, 28(3):855–862, 2009.
- [25] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haeberli. Fast shadows and lighting effects using texture mapping. *ACM SIGGRAPH Computer Graphics*, 26(2):249–252, 1992.
- [26] A. J. Stewart. Vicinity Shading for Enhanced Perception of Volumetric Data. In *IEEE Visualization*, pages 355–362, 2003.
- [27] E. Sundén, A. Ynnerman, and T. Ropinski. Image Plane Sweep Volume Illumination. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2125–2134, 2011.
- [28] V. Soltészová, D. Patel, S. Bruckner, and I. Viola. A Multidirectional Occlusion Shading Model for Direct Volume Rendering. *Computer Graphics Forum*, 29(3):883–891, 2010.
- [29] V. Soltészová, D. Patel, and I. Viola. Chromatic Shadows for Improved Perception. In *Proc. of Non-Photorealistic Animation and Rendering (NPAR)*, pages 105–115, 2011.
- [30] L. Wang and A. E. Kaufman. Lighting System for Visual Perception Enhancement in Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(1):67–80, 2013.
- [31] L. C. Wanger, J. A. Ferwerda, and D. P. Greenberg. Perceiving Spatial Relationships in Computer-Generated Images. *IEEE Computer Graphics and Applications*, 12(3):44–51, 54–58, 1992.
- [32] C. Weber, A. Kaplanyan, M. Stamminger, and C. Dachsbacher. Interactive Direct Volume Rendering with Many-light Methods and Transmittance Caching. In *Proc. of the International Workshop on Vision, Modeling, and Visualization*, pages 195–202, 2013.
- [33] Y. Zhang, Z. Dong, and K. Ma. Real-time Volume Rendering in Dynamic Lighting Environments Using Precomputed Photon Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1317–1330, 2013.
- [34] Y. Zhang and K.-L. Ma. Fast Global Illumination for Interactive Volume Visualization. In *Proc. of SIGGRAPH Symp. on Interactive 3D Graphics and Games*, pages 55–62, 2013.
- [35] Y. Zhang and K.-L. Ma. Lighting Design for Globally Illuminated Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2946–2955, 2013.