Linköping University Pre Print

Efficient Visibility Encoding for Dynamic Illumination in Direct Volume Rendering

Joel Kronander, Daniel Jönsson, Joakim Löw, Patric Ljung,
Anders Ynnerman and Jonas Unger

N.B.: When citing this work, cite the original article.

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Joel Kronander, Daniel Jönsson, Joakim Löw, Patric Ljung, Anders Ynnerman and Jonas Unger, Efficient Visibility Encoding for Dynamic Illumination in Direct Volume Rendering, 2011, IEEE Transactions on Visualization and Computer Graphics.

<http://dx.doi.org/10.1109/TVCG.2011.35>

Preprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-66839>

Efficient Visibility Encoding for Dynamic Illumination in Direct Volume Rendering

Joel Kronander[†], Daniel Jönsson[†], Joakim Löw[†], Patric Ljung[‡], Anders Ynnerman[†], Jonas Unger[†]

Fig. 1. Computed Tomography (CT) scan of a human abdomen with a stent supporting the aorta. The image displays the abdomen rendered using: (a) gradient based diffuse shading, (b) our method for shading and (c) a close up of (b). The local shading and shadows generated by our method present visual cues that reveal the shape of the heart as well as the structure of the stent and bones. It is evident that the efficiently encoded visibility information enables detailed shading and increased spatial perception of the data set.

Abstract—We present an algorithm that enables real-time dynamic shading in direct volume rendering using general lighting, including directional lights, point lights and environment maps. real-time performance is achieved by encoding local and global volumetric visibility using spherical harmonic (SH) basis functions stored in an efficient multi-resolution grid over the extent of the volume. Our method enables high frequency shadows in the spatial domain, but is limited to a low frequency approximation of visibility and illumination in the angular domain. In a first pass, Level Of Detail (LOD) selection in the grid is based on the current transfer function setting. This enables rapid on-line computation and SH projection of the local spherical distribution of visibility information. Using a piecewise integration of the SH coefficients over the local regions, the global visibility within the volume is then computed. By representing the light sources using their SH projections, the integral over lighting, visibility and isotropic phase functions can be efficiently computed during rendering. The utility of our method is demonstrated in several examples showing the generality and interactive performance of the approach.

Index Terms—Volumetric Illumination, Precomputed Radiance Transfer, Volume Rendering.

1 INTRODUCTION

Light is the fundamental carrier of visual information. It not only transports digital pixel values from the computer screen onto the retina where it is registered for interpretation, but also plays a key role in the computation of visually rich visualizations that allow us to understand complex systems and make decisions based on multidimensional data. The lighting used in the rendering of an image has an enormous impact on how it is interpreted by a human observer. By careful design of the setup, the lighting can reveal strong visual cues [15], describing

global structures, local details and curvature of surfaces and volumetric structures that would otherwise be hidden from the observer. Advanced illumination models have therefore been widely adopted both in cinematography and computer graphics [4], as well as scientific and medical visualization [10], to convey structure and spatial relationships.

However, dynamic and interactive lighting is, in the context of interactive volume rendering, still a great challenge. Its importance has motivated research and development of a variety of sophisticated shading techniques. Nevertheless, illumination models allowing real-time interaction have been limited to texture slicing, local lighting effects and non-physically based computations, or have put constraints on the number, position and generality of the light sources. In contrast to general volumetric illumination methods used in traditional computer graphics, in interactive volume rendering, fast updates of the visibility when changing the transfer function (TF) are necessary for an interactive workflow. This leads to several additional constraints on the illumination model used, and rules out traditional methods based on pre-computed radiance transfer, which requires lengthy pre-computations when TF updates occur.

In this paper, we focus on *Direct Volume Rendering* (DVR) and the

[†] is with C-Research, Linköping University, Sweden, email : [(joel.kronander), (daniel.jonsson), (joakim.low), (anders.ynnerman), (jonas.unger)]@liu.se

[‡] is with Princeton Corporate Research, NJ, USA, email: patric.ljung@siemens.com

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org .

real-time ray-casting paradigm. We present a technique for computing volumetric illumination effects, including both local and global self shadowing using lighting setups that include arbitrary positioned directional and point light sources. This is realized by encoding spherical distributions of local and global visibility information in the volume using spherical harmonics (SH) basis functions, that are spatially distributed using a high performance multi-resolution grid [21]. The local visibility is adaptively represented in the multi-resolution grid, whereas the global visibility is computed and stored in a regular grid. The visibility information and SH projections are rapidly computed online in two steps using an efficient volume rendering framework. First, we compute a local visibility distribution within a radius centered at each voxel in the multi-resolution grid. The global visibility in the volume is then computed in a second pass by integrating the local distributions in a number of sample directions. By decoupling of the directionally varying visibility from the lighting, our technique enables the viewpoint and position of light sources to be varied arbitrarily. Another benefit of using SH basis functions for storing the radiance transport within the volume is that our technique also enables the use of real world lighting environments, so called *High Dynamic Range* (HDR) environment maps [35]. While previous methods using spherical harmonics for volumetric lighting have been limited to static attribute data [35, 30, 39], our fast recomputation of both the local and global visibility enables interactive workflows, as the performance penalty introduced by TF updates is in the order of 1s for standard high resolution volumes. This performance is an order of magnitude faster than previous SH volumetric lighting techniques.

The adaptive multi-resolution grid structure enables accurate representation of high frequency spatial variations. The finite spherical harmonics expansions, however, limit the distributions in the angular domain around each sample point to low frequency lighting environments. In this paper, we limit ourselves to include only isotropic phase functions, and do not consider more complex scattering properties of volumetric media and materials.

Contributions - The main contributions in this paper are:

- An efficient approach for computing and representing local and global visibility enabling real-time illumination and interactive TF updates.
- Support for dynamic lighting environments, including lighting designs consisting of several point lights and rim lights (e.g environment maps).
- A comprehensive evaluation of the novel rendering parameters introduced with our approach. We also demonstrate our approach on a number of practical scenarios to show the utility of the method.

2 PREVIOUS WORK

Our method is inspired by previous work considering realtime illumination of volumetric media and methods using precomputed radiance transfer.

Illumination models for volume rendering - The complex effects of self-shadowing and illumination of volumetric media have received a lot of attention in previous work. In the seminal work by Max [25] the foundations for volumetric illumination models are discussed along with offline methods for efficient light propagation. Shading techniques for interactive volume rendering have, in practice, predominantly been constrained to strictly local effects such as the *Blinn-Phong* model [7]. Since such methods rely local information only, they give poor depth and proximity cues for comprehension of spatial features in the volume data. They also often use gradients to provide surface normals which results in poor shading for homogenous regions and clip plane surfaces.

Using texture slicing, Kniss *et al.* [14], estimate forward scattering, shadows and color bleeding through half angle slicing. This technique has also been extended to incorporate dilation of light to reduce hard shadow edges by Desgranges *et al.* [6]. Schott *et al.* [34] presented a method using a view-oriented backward-peaked cone phase function.

These methods require recomputation when changing viewpoint, and can only handle a single light source. Furthermore, they are limited to a slice based volume renderer and can not readily be used in ray-casting pipelines.

Recent works have been directed towards including neighborhood structure and self-shadowing to provide important perceptual cues for better comprehension of spatial ordering and orientation [15]. Ambient occlusion techniques compute the amount of light reaching a point from a uniform spherical environment. Hernell *et al.* [11, 13] perform a ray-tracing pass per voxel over a local spherical neighborhood to compute a scalar occlusion factor that has to be recomputed each time the transfer function changes. The computations are accelerated with an efficient volume renderer; implementing a multi-resolution level of detail selection. Desranges *et al.* [5] proposed a method to perform fast recomputation of an ambient occlusion term from several pre-filtered volume representations. To handle dynamic updates of the transfer function, Ropinski *et al.* [32] compute local data histograms for each voxel neighborhood in an extensive precomputation step. The compressed local data statistics combined with a transfer function setting allow them to approximate dynamic ambient occlusion and color bleeding effects. These methods are limited to local effects due to their computational complexity and do not, in contrast to our approach, incorporate any directional shading effects.

Previous work has also considered global illumination and scattering effects for real time rendering. Hernell *et al.* [12] extended their framework for local ambient occlusion to incorporate global shadows, and first order scattering effects through piecewise integration of local ray segments. Global shadows are calculated for each light source separately, and require recalculation when moving or adding light sources. Global illumination and scattering effects using a Face Centered Cubic lattice for improved sampling efficiency was presented by Qui *et al.* [27]. Rezk-Salama [33] presented semi-interactive methods for Monte Carlo ray casting on the GPU, allowing global illumination and general scattering effects. Although these approaches are promising, their methods are still too computationally intensive to reach interactive frame rates, especially for dynamic lighting environments.

To simulate global shadow effects for realtime rendering, methods computing shadow volumes [2] and deep shadow maps [22, 9] have been presented. These methods sample the volumetric occlusion from the light sources utilizing efficient pre-computed data structures. To include scattering effects Ropinski *et al.* [31] compute a volumetric representation for each light source using a slice-based volume renderer. While these methods are efficient for a single static light source, moving the light source or adding additional light sources require a recomputation of the shadow representation, leading to poor frame rates for dynamic lighting environments.

In the recent work by Lindemann and Ropinski [18] spherical harmonics are utilized for interaction and rendering of advanced material properties for volume rendering tasks. However, they do not consider multi-resolution structures or rapid visibility updates.

Global illumination effects for isosurfaces extracted from volumetric data sets have been considered by Wyman *et al.* [38]. The authors present a method incorporating a precomputed illumination representation using spherical harmonics, allowing interactive change of viewpoint, illumination and isovalue. Beason *et al.* [1] also incorporated translucent materials and caustics, but, limited to static lighting. These methods are limited to isosurfaces of a single isovalue, and cannot incorporate shadowing effects between multiple isosurfaces.

Precomputed radiance transfer - Precomputed radiance transfer (PRT) has a long history in computer graphics. State of the art PRT techniques for polygonal geometry can, as described in a recent survey by Ramamoorthi [28], handle all frequency effects, varying illumination and viewpoint, editing of material properties, and approximations to dynamic scenes. Although a large amount of work has been directed towards PRT for polygonal models, volumetric data and volume rendering have received little attention. In the seminal paper by Sloan *et al.* [35], illumination for volumetric data is discussed. However their method requires extensive pre-computation and cannot handle interactive updates of transfer functions. Ritschel [30] presented a method for

more efficient precomputation, however for reasonably large volume data changing the transfer function for interactive volume rendering still requires several seconds or minutes, and the reported frame rates are low. In this paper, we utilize a grid of spherical harmonic coefficients to encode self-shadowing of the volumetric data. Volumetric grids of spherical harmonics have also been used for rendering hair [26] and to encode spatially varying illumination [23]. Due to the large number of spherical harmonic coefficients needed for general scenes, compression methods based on clustered principal component analysis have previously been presented [36]. For polygonal meshes fast updates of material properties represented by BRDFs have been the focus of several works [3, 37]. However to maintain interactive updates they have either assumed approximations of the underlying materials or used static lighting and viewpoint settings. Furthermore, these works do not consider volumetric media, and do not propose methods to handle efficient coding of visibility information in the five dimensional space represented by spatial and angular variations. Recently a GPU implementation rendering participating media (smoke) in realtime was presented by Ren *et al.* [39]. They take single and multiple scattering into account by using low-frequency approximations of the volume density with radial basis functions. Using fast basis rotation and spherical harmonic multiplication through spherical harmonic exponentiation [29], they achieve realtime updates of the visibility function. However, their method assumes an absorption coefficient that is proportional to density. For volume rendering, where the absorption/emission is mapped nonlinearly by density values, their approach based on approximating the underlying density field with radial basis functions does not apply.

In summary, no previous work has enabled the use of dynamic and general lighting environments for volumetric media while also enabling interactive TF updates. Since both these properties are highly important for usable and flexible illumination models in DVR, our work is clearly motivated and outperforms previously presented methods.

3 THEORETICAL OVERVIEW

In this section we present the theoretical foundations of our work.

3.1 Volumetric Light Transport

To simulate light transport within a volume and compute the accumulated intensity contribution reaching each pixel, the volume rendering integral is evaluated along each ray in direction ω_o from 0 to D [25]:

$$I(D) = \int_0^D g(x, \omega_o) \cdot \exp\left(-\int_0^x \tau(s(\hat{x})) d\hat{x}\right) dx, \quad (1)$$

where $\tau(s(x))$ denotes the absorption and $g(x, \omega_o)$ the source radiance (scattering and emission) at each position x along the ray. Assuming single scattering and emission, the radiance contribution at each sample position x can be described as:

$$g(x, \omega_o) = g_e(x) + g_s(x, \omega_o), \quad (2)$$

where $g_e(x)$ describes the emission at x and $g_s(x, \omega_o)$ describes the radiance scattered in the outgoing direction ω_o of the sample ray. The scattering $g_s(x, \omega_o)$ is, for a given outgoing direction ω_o at x , the integral of a product of the radiance distribution function $L(x, \omega_i)$, the visibility $V(x, \omega_i)$ through the volume from the position x , and the phase function $\rho(x, \omega_i, \omega_o)$:

$$g_s(x, \omega_o) = \int_{S^2} L(x, \omega_i) V(x, \omega_i) \rho(s(x), \omega_i, \omega_o) d\omega_i. \quad (3)$$

For the purposes of the algorithm presented here, the phase-function is assumed to be isotropic: $\rho(x, \omega_i, \omega_o) = 1/4\pi$, that is light is scattered equally in all directions. This yields:

$$g_s(x, \omega_o) = g_s(x) = \frac{1}{4\pi} \int_{S^2} L(x, \omega_i) V(x, \omega_i) d\omega_i. \quad (4)$$

For general lighting environments, $L(x, \omega_i)$ varies with position, $x \in R^3$, and angle of incidence, $\omega_i \in S^2$. The visibility, $V(x, \omega_i)$, is a function defined on the same 5D domain, $R^3 \times S^2$. In our work we consider both local and global visibility through the volume, see Section 4. To enable general and interactive illumination, it is necessary to decouple $V(x, \omega_i)$ and $L(x, \omega_i)$ in such a way that the integral in (4) can be computed rapidly during rendering.

3.2 Spherical Harmonics Representation

To enable efficient representation of $L(x, \omega_i)$ and $V(x, \omega_i)$, our method employs real-valued spherical harmonic basis functions. The set of SH basis functions represent an orthonormal basis (ON-basis) for all square-integrable functions defined on the sphere S^2 . Hence it is possible to expand, and find an approximation $\tilde{f}(\omega)$ to, a function $f(\omega)$ defined on S^2 :

$$\tilde{f}(\omega) = \sum_{l=0}^{n_l} \sum_{m=-l}^l f_{l,m} Y_l^m(\omega), \quad (5)$$

where $Y_l^m(\omega)$ are the SH basis functions, defined by the degree l and order m , with $l \geq 0$ and $-l \leq m \leq l$ [8]. The coefficients $f_{l,m}$ are computed by integration:

$$f_{l,m} = \int_{S^2} f(\omega) Y_l^m(\omega) d\omega. \quad (6)$$

For practical use, the outer sum in (5) is truncated to yield a function approximation of a finite number of coefficients. Here, we use $n_l \leq 3$, yielding a maximum of 16 coefficients (truncating the degree to n results in $(n+1)^2$ coefficients). Being an ON-basis, SH enables efficient integral evaluation. Using truncated SH expansions $\tilde{L}(x, \omega_i) \approx L(x, \omega_i)$ and $\tilde{V}(x, \omega_i) \approx V(x, \omega_i)$, the integral describing the source radiance (4) reduces, for a given position, x , to a scalar product between the corresponding coefficient vectors:

$$g_s(x) \approx \frac{1}{4\pi} \sum_{l=0}^{n_l} \sum_{m=-l}^l L_{l,m} V_{l,m}. \quad (7)$$

Note that the coefficients $L_{l,m}$ and $V_{l,m}$ depend on the position, x . For $n_l \leq 3$, the scalar product in (7) involves only 16 terms or less, and can be efficiently evaluated in real-time.

Another property of SH which proves valuable for the application in this paper, is the ability to perform efficient rotation. Using the basis expansion, the rotation of a function around an arbitrary axis is expressed as a matrix, describing the relation between original and rotated coefficients. For the SH basis, only basis functions of the same degree, l , affect each other during rotation, giving a memory-efficient block diagonal matrix:

$$\begin{pmatrix} f_{0,0}^r \\ f_{1,*}^r \\ f_{2,*}^r \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{R}_0 & 0 & 0 & \dots \\ 0 & \mathbf{R}_1 & 0 & \\ 0 & 0 & \mathbf{R}_2 & \\ \vdots & & & \ddots \end{pmatrix} \begin{pmatrix} f_{0,0} \\ f_{1,*} \\ f_{2,*} \\ \vdots \end{pmatrix}. \quad (8)$$

Restriction of rotation around a single coordinate axis yields an even more sparse matrix, which means efficient rotation using Euler angles can be performed. An in-depth discussion of how we perform rotations in practice is given in Section 4.5. A more detailed discussion of SH functions, their definition, and efficient evaluation in practice can be found in the tutorial by Green [8].

4 ALGORITHM

The algorithm, as outlined in Figure 3.2, consists of three main steps: computing the volumetric visibility $V(x, \omega_i)$, computing and storing the light sources $L(x, \omega_i)$, and rendering using general and dynamic illumination environments.

Fig. 2. The algorithm consists of three main steps: visibility computation, light source representation and rendering. The local visibility SH expansions $\tilde{V}^L(x, \omega_i)$ are computed in a first step, and then used to compute the global visibility $\tilde{V}^G(x, \omega_i)$. The SH representation of the lighting environment $L(x, \omega_i)$ enables efficient evaluation of the interaction between the volumetric self-occlusion and incident illumination during rendering.

The evaluation of volumetric visibility is performed online in two steps using a multi-resolution grid, where the level of detail (LOD) is adapted to the current TF settings. First, local visibility $V^L(x, \omega_i)$, is sampled in spherical neighborhoods, adaptively distributed according to the LOD selection, and stored as truncated SH expansions $\tilde{V}^L(x, \omega_i)$. Global visibility SH expansions $\tilde{V}^G(x, \omega_i)$ are then calculated by piecewise integration using the stored local visibility $\tilde{V}^L(x, \omega_i)$. To enable fast calculation of $g_s(x)$ in the computation of the volume rendering integral (1), the lighting environment is also stored as a set of SH expansions, one for each light source. Parallel rotations of the SH expansions then enable real-time rendering of both spatially and angularly varying light sources, including point lights, directional lights and environment maps. These steps are described in detail below.

4.1 Sparse Visibility Representation

The visibility function, $V(x, \omega_i)$, is estimated by tracing rays through the data volume. The sampled spherical visibility at a position, x , is then compactly encoded using an SH expansion. To efficiently represent the spatial distribution of SH coefficients (in R^3), a multi-resolution data structure is used. For an overview of the combined visibility representation, see Figure 3. Using a multi-resolution data structure allows adaptive computation and storage of the SH-coefficients, thus avoiding unnecessary computations in empty and homogeneous regions, as defined by the TF. For this purpose we have adopted a flat blocking multi-resolution method, as presented by Ljung et al. [21, 20]. In a pre-processing step, the volume is divided into blocks with a resolution of fixed size. In this paper a block size of 16^3 voxels is used. Each block is analyzed to determine LOD selection parameters, and all levels of detail are computed and encoded (here corresponding to $16^3, 8^3, 4^3, 2^3, 1^3$ voxels). These blocks are then stored in a multi-resolution representation enabling efficient loading of arbitrary LOD selections at runtime. When the TF is changed or the volume is cropped, a fast, adaptive LOD selection of the data is performed per block. The LOD selection strives to minimize the variation in the TF domain (data mapped through the TF) given a fixed memory budget. Blocks with no TF content can be skipped entirely. The LOD selection process starts by computing a TF based significance for each block. The significance of each block is computed as the difference in the CIE Luv color space (for details see [21]) between the lowest resolution level and the original data. To allow for interactive LOD selection when the TF changes, a piecewise constant approximation of the block data histogram is used for computing the TF based significance measure. First, all blocks are set to the minimum resolution level. The LOD selection process then uses an adaptive prioritization queue to select for which blocks to increase the resolution level, until the memory budget is reached. More details regarding the priority queue is given in the original paper [21].

Fig. 3. To store visibility functions, $V(x, \omega_i)$, on the 5D domain $R^3 \times S^2$ we utilize a multi-resolution structure to store the variation in R^3 and SH expansions to store the variation in S^2 . The flat blocked multi-resolution structure adapts for $16^3, 8^3, 4^3, 2^3, 1$ or no SH expansions to be stored in each 16^3 voxel block.

For our GPU implementation, the multi-resolution structure is packed into a set of regular 3D textures, one for the scalar data values and a few for storing SH coefficients (the number of 3D textures depends on how many SH coefficients are used). The mapping from volume coordinates to packed coordinates in the 3D textures is performed by a lookup in an explicit index texture, containing the number of blocks, size of each block and the block offset in the index texture.

The multi-resolution data structure is used for storing both the volume scalar data and the local visibility, similarly to Hernell et al. [13], but with the addition of storing SH expansions, $\tilde{V}^L(x, \omega_i)$, of the visibility function, instead of an occlusion value only.

4.2 Computing Local Visibility

The local visibility SH expansions, $\tilde{V}^L(x, \omega_i)$, are computed by sampling a spherical neighborhood around each grid point, $x \in R^3$, specified by the LOD selection. Inspired by previous work on local ambient occlusion [13], the extent of the neighborhood is limited to a radius, r_s , of approximately the same number of voxels as the multi-resolution grid block size, which in the present case is $r_s \approx 16$ voxels. The local visibility is, as described in Figure 4(a), sampled using a number of rays distributed over the sphere. The distribution of ray directions, $\omega_i \in S^2$, is selected as the vertices of a tetrahedron, icosahedron or octahedron subdivided to the desired level using the partition method presented by Lessing et. al. [17]. Using a model of the volume rendering integral (1), taking only absorption into account, an opacity value is obtained for each ray direction by adaptive sampling of the volume within the radius r_s . In practice, the entire volume is processed by mapping slices of the volume to a framebuffer object, where each pixel maps to one grid point in the multi-resolution structure. Since the computation is driven by the same LOD structure used for storing the visibility function, unnecessary computations are avoided. The integration along each sampling ray utilizes an adaptive step size according to the LOD selection in the multi-resolution structure, as also proposed in [20]. The slice offset and the location of the current fragment in the framebuffer provides the packed texture coordinates of the corresponding multi-resolution grid. To perform the mapping from packed coordinates to volume coordinates, obtaining the origin for the sample rays, a reverse index texture is used. For each sample along the rays the forward index texture, described in 4.1, is used to find and sample the packed volume.

Iteration is performed for all slices in the packed volume, one ray at a time, to increase texture cache coherency. When a ray direction has been sampled, the corresponding visibility value is used to update the numerical computation of the SH coefficients. Iterating over all the rays yields the final SH coefficients, $V_{l,m}^L$, describing the spherical distribution of local visibility at any grid point, x . Figure 14(b) displays the effect of only using local visibility for a CT-scan of a human heart. Figure 14(c) displays a rendering of the same dataset but

Fig. 4. In a first rendering pass (a), for each grid point in the multi-resolution structure the visibility function is adaptively sampled in a spherical neighborhood with a finite radius r_s . In a second pass (b), the local visibility SH expansions are reused to evaluate the global visibility information, sampling several local visibility contributions in each direction.

instead using global visibility. Using only local visibility for shading more details can be identified without moving the light source but at the cost of a reduced depth perception and spatial comprehension of the global properties in the data set.

4.3 Computing Global Visibility

If larger spherical neighborhoods are to be considered, visibility information can be prohibitively expensive to sample using the method described above. Instead, global visibility, $\tilde{V}^G(x, \omega_i)$, is computed after the local visibility data has been generated. For a point, x , the global visibility computation utilizes local visibility data, again along a set of ray directions distributed over the sphere. Along each ray, local visibility is sampled at locations, r_s , voxels apart, yielding samples, $\tilde{V}^L(x_r, \omega_i)$, at sample locations, x_r . To compute the visibility in the current ray direction, ω_r , the sampled visibility function, $\tilde{V}^L(x_r, \omega_i)$, is evaluated for ω_r :

$$\tilde{V}^L(x_r, \omega_r) = \sum_{l=0}^{n_l} \sum_{m=-l}^l V_{l,m}^L Y_l^m(\omega_r). \quad (9)$$

This value is taken as representative of the visibility between the current sample position and the consecutive sample position. Since the algorithm operates one ray direction at a time for the entire volume, the values of the basis functions, $Y_l^m(\omega_r)$, requires computation only once for each direction.

Traversing the ray and using standard front-to-back compositing [7], a final global visibility value per ray can be calculated. Using a ray tracing approach, the method also allows for early ray termination, avoiding unnecessary samples. Similarly to the computation of the local SH expansion, the visibility data is projected onto an SH basis, but the global SH expansions are stored in a regular grid instead of a multi-resolution data structure.

4.4 Light Source Representation

To accommodate lighting into the framework, as outlined in Section 3.2, SH expansions are computed for all types of light sources. Expansions of global lighting from environment maps, $L^e(\omega_i)$, are precomputed and rotated into place during rendering. Directional light sources, $L^d(\omega_i)$, are considered Dirac deltas in the directions, ω_d , of the light sources, which implies a straight-forward computation of the corresponding SH coefficients:

$$L_{l,m}^d = \int_{S^2} c \delta(\omega_i - \omega_d) Y_l^m(\omega_i) d\omega_i = c Y_l^m(\omega_d), \quad (10)$$

where c is an intensity constant. These coefficients need to be computed only once each time the directions to the light sources change. Point light sources, $L^p(x, \omega_i)$, which enable local lighting, utilize

Dirac delta SH coefficients similarly to directional light sources, but are computed once for every voxel position x to get the correct direction to the light source with respect to x . In addition, the intensity of the point light source is attenuated by the distance between the light source and x .

The final light contribution to a point, x , is computed using (7), with coefficients, $L_{l,m}$, given by the SH expansion of the function (possibly using several light sources of each kind):

$$L(x, \omega_i) = L^e(\omega_i) + L^d(\omega_i) + L^p(x, \omega_i), \quad (11)$$

which, by using the corresponding SH expansions at x , results in the basis coefficients

$$L_{l,m} = L_{l,m}^e + L_{l,m}^d + L_{l,m}^p. \quad (12)$$

4.5 Rendering

The volume rendering integral (1) is evaluated at runtime using standard ray-casting accelerated by the multi-resolution grid. At each sample point, x , along a ray, the light source SH representation, described above, and the local or global SH expansions encoding visibility enable efficient computation of $g_s(x)$ using the scalar product in (7). Through the decoupling of visibility, $V(x, \omega_i)$, and lighting, $L(x, \omega_i)$, the fast SH rotation, described in Section 3.2, enables translations and rotations of both the volume data and the light sources without significant degradation in rendering performance. It is also possible to interactively switch between local and global visibility without any recalculation. The local visibility, $\tilde{V}^L(x, \omega_i)$, is especially useful for cut-plane renderings, but is also used as an approximation to enable light sources to be positioned inside the volume.

For directional and environment light sources, the rotation needs to be performed on the CPU only when the light direction changes. However, when point light sources are used, the rotation needs to be performed on the GPU for each sample along the ray since the direction from the sample point to the light source varies. The block diagonal parts of the rotation matrix (described in (8)) R_0, R_1, R_2 and R_3 are of sizes $1 \times 1, 3 \times 3, 5 \times 5$ and 7×7 , respectively. To compute the rotation efficiently on the GPU we divide the block diagonal matrices into sub parts of 4×4 matrices and 1×4 vectors. Thus SH degree one requires one matrix-vector multiplication. Degree two requires more work, since the 5×5 matrix does not fit naturally into the GPU computation framework; it results in two matrix-vector multiplications, one matrix-scalar multiplication, one dot product and one addition. Similar work needs to be done for degree three, which also consists of a 7×7 matrix. For our GPU implementation, we generated the expression for the rotation coefficients using Maple, but the CPU implementation uses a more general approach of which a detailed description, including source code, can be found in [19].

5 PARAMETER SETTINGS

One of the most important aspects in the assessment of the efficiency of any interactive visualization technique is the tradeoff between performance and accuracy. In the context of the proposed method, this section presents a qualitative overview of these tradeoffs along with a quantitative evaluation of the relation between the visual accuracy, and the two time critical steps in our approach: recomputing visibility and online rendering performance.

To quantify how the rendering parameters affect the performance and visual quality, we have conducted a study in which we have varied the parameters in a systematic way, and for each parameter configuration rendered an image. We have used two data sets: one synthetic for the qualitative overview, see Figures 5, 6, 7, and one real data set for the quantitative evaluation, see Figures 8(f), 9(f), 10(f) and 11(f). The synthetic data set is chosen to be a corner case that emphasize the limitations of the approach. The opaque sphere and torus exhibit sharp silhouettes and crisp shadows specifically chosen to introduce features that are not band limited. The high spatial and angular frequencies are problematic to sample and particularly challenging to represent using spherical harmonics expansions. The perfectly flat plane is chosen to fully reveal ringing and discretization artifacts. The real data

Fig. 5. The number of ray samples used to compute the visibility $V(x, \omega_i)$ can be varied arbitrarily. The figure displays the difference in visual quality using (a) 20, (b) 128 and (c) 512 rays encoded using 9 SH coefficients at each point for global visibility renderings of a synthetic data set. Using a low number of sampling rays lead to visual banding artifacts in shadow regions due to the coarse quantization of angular visibility.

Fig. 6. Using a finite number of spherical harmonic coefficients, a low-frequency approximation to the spherical variation in visibility per grid point is obtained. The images show a synthetic volume rendered with the SH expansion truncated to (a) 4, (b) 9, (c) 16 coefficients respectively.

Fig. 7. Images rendered with a global visibility grid of resolution (a) $32 \times 32 \times 32$, (b) $64 \times 64 \times 64$ and (c) $128 \times 128 \times 128$ respectively.

set used is the Bonsai Tree data set illuminated with the Grace Cathedral environment map. The rendered images have been compared to a high quality reference image using the perceptually based visual difference predictor (VDP) norm [24] (default settings for HDR-VDP version 1.7). We quantify the results of the VDP norm by considering the ratio of pixels that contain distortions of a probability greater than 75%. All presented images are rendered with a viewport resolution of 1024×1024 pixels on a PC equipped with dual Intel Xeon 3.2 GHz processors, 6GB of RAM and an NVIDIA GeForce GTX 295 graphics card with 896 MB onboard memory.

5.1 Visibility calculations

Each time the TF settings are changed, the local and global visibility needs to be re-computed. The computational performance and quality of the rendered images are affected by several parameters:

- Number of rays used to sample visibility, $V(x, \omega_i)$.
- Radius of the spherical regions used for sampling local visibility, $V^L(x, \omega_i)$.
- The number of spherical harmonic coefficients.

- The memory budget for the multi-resolution grid used for storing and computing local visibility, $\tilde{V}^L(x, \omega_i)$.
- Spatial resolution of the regular grid used for storing the global visibility, $\tilde{V}^G(x, \omega_i)$.

The first parameter in our evaluation is the number of rays used to sample the local, $\tilde{V}^L(x, \omega_i)$, and global, $\tilde{V}^G(x, \omega_i)$, visibility at each grid point. Using an insufficient number of rays will result in artifacts caused by strong edges or small features in the volume. The number of sampling rays required is related to the size (in voxels) of the local neighborhood and the number of SH coefficients used in the basis projection. Figure 5 displays an example of how the visual quality is affected by varying the number of sampling rays. For this comparison, the other parameters have been fixed (using a local visibility radius of 16 voxels, 9 SH coefficients, a $128 \times 128 \times 128$ resolution for the global visibility), and a single directional light source have been used. Increasing the number of sample rays will result in smoother shadows and reduced discretization artifacts. Visual artifacts when using too few rays often manifests themselves as noisy and distorted shadow regions and are especially emphasized on flat surfaces and when using low grid resolutions (see e.g. Figure 5(a) using 20 sampling rays).

Fig. 8. The example scene using (a) 16, (b) 64 and (c) 2048 sampling rays for computing visibility. (d) and (e) depicts the visual difference predicted with the HDR-VDP norm, where pixels are colored according to the probability of detection as: 100% magenta, 75%-90% green, 62,5% yellow, 50% green and less than 25% gray. (f) the variation in update time (solid blue line) and visual quality (dashed red curve) with varying the number of sampling rays. The visual quality corresponds to the ratio of pixels that contain distortion of the probability greater than 75%.

Fig. 9. The example scene rendering using (a) 4, (b) 9 and (c) 16 SH coefficients. (d) and (e) depicts the visual difference predicted with the HDR-VDP norm. (f) shows the variation in update time (solid blue line) and visual quality (dashed red curve) with varying number of SH coefficients.

Fig. 10. The example scene rendered using a data reduction setting of (a) 1:33.6, (b) 1:6.7 and (c) 1:1.7 for the multiresolution grid. (d) and (e) depicts the visual difference predicted with the HDR-VDP norm, where pixels are colored according to the probability of detection as: 100% magneta, 75%-90% green, 62,5% yellow, 50% green and less than 25% gray. (f) shows the effects on visibility update time (solid blue line) and visual quality (dashed red line) when varying the data reduction rate ratio for the grid used for storing local visibility.

Fig. 11. The example scene rendered using a resolution of (a) $32 \times 32 \times 16$, (b) $64 \times 64 \times 32$ and (c) $512 \times 512 \times 256$ for the global visibility grid. (d) and (e) depicts the visual difference predicted with the HDR-VDP norm. (f) shows the effects on visibility update time (solid blue line) and visual quality (dashed red line) when varying the size of the regular grid storing the global visibility.

Fig. 12. TF update times for different radius of the local spherical regions used for visibility computation, (solid blue curve) using 32 sampling rays and (dashed red curve) using 64 sampling rays.

The bright spots at the base of the sphere and the torus, visible in the reference image rendered using 512 rays (Figure 5(c)) are most likely due to ringing artifacts (aliasing) introduced by the low frequency SH approximation of the hemispherical visibility. The same visual artifact can also be observed in images rendered with only local visibility. As expected, the size and intensity of the bright region also varies with the number of SH coefficients, see Figure 6.

Figure 8(f) (solid blue line) shows that the relationship between the number of rays used and the computation time (left scale) is linear. This behavior is expected, given that all other parameters are fixed, (4 SH-coefficients, 16 voxels local radius, 1:6.7 data reduction rate for the multi-resolution grid and a $64 \times 64 \times 32$ global visibility grid resolution). Figure 8(f) (red dashed line) also displays the relation between the number of rays and the perceived visual quality (right scale) quantified using VDP as described above. The reference image for the VDP comparison was generated using 2048 sample rays per grid point. It can be seen that the visual error compared to the reference decreases dramatically, up to 40 - 60 rays, and that more rays have little effect on the quality. In contrast to traditional SH lighting, pre-computed radiance transfer, where the visibility in most cases is sampled using several thousands of rays or more, we use a relatively small number of sampling rays. This is possible since the limited size of local neighborhoods results in an adequate sampling density per volume element. Figures 8(a),(b),(c) shows the visual result of the example scene obtained using 16, 64 and 2048 sampling rays.

The number of SH coefficients affects the memory footprint, the update time and the angular frequency of the encoded visibility at each grid point. The onboard GPU memory limits the number of coefficients. Given the resolution of the data sets used in our experiments and the 896 MB of onboard memory on the GeForce GTX 295 graphics card, we have for all data sets used in this paper, see Figure 15, considered 1 - 16 coefficients. Note that using 1 coefficient corresponds to ambient occlusion, [13], and that the number of coefficients depends on the degree of the SH expansion. A higher number of coefficients allows for reconstruction of higher frequency visibility distributions with higher response to light changes in the angular domain. This is displayed in Figure 6, where different number of coefficients are used (images rendered using 512 sample rays, 16 voxels local radius, 1:3 data reduction rate for the multi-resolution grid and a $128 \times 128 \times 128$ global grid resolution). Using a higher number of SH coefficients, increases the sharpness of shadows, and enables detailed variations in the shading when light sources are moved. The number of SH coefficients affects the update time, as more computations and writes to texture memory have to be performed during the projection of the samples to the coefficient basis. The effects on update time and visual quality is shown in Figure 9(f) (comparison images generated with 32 sampling rays, 16 voxel radius, 1:6.7 data reduction rate for the multi-resolution grid and a $64 \times 64 \times 32$ global grid resolution).

Fig. 13. Rendering performance (fps) for varied number of SH coefficients. (dashed red line) Using one point light source, and (solid blue curve) using the Grace Cathedral environment map.

One of the major tradeoffs in terms of quality and performance is the resolution of the visibility representation encoding the spatial variation. The memory budget set for the LOD selection of the multi-resolution grid affects the total memory footprint, the adaptive sampling along the visibility sample rays, and enables early ray termination, hence also the time required for updating the local visibility $\tilde{V}^L(x, \omega_i)$. Figure 10(f) (solid blue line) shows how the update time varies with the total data reduction rate of the volume, and Figure 10(f) (dashed red line) the effect on visual quality quantified using VDP metric. For the images used in the comparison, the other parameters were kept fixed (4 SH-coefficients, 32 sample rays, 16 voxels local radius, and a $64 \times 64 \times 32$ global grid resolution).

To represent the global visibility, $\tilde{V}^G(x, \omega_i)$, we limit ourselves to a regular grid. For low grid resolutions, this limitation can introduce unwanted visual artifacts. Unnecessary computations are also performed in some parts of the volume, as the global resolution is not adapted to the current TF setting. In Figure 7, the resolution of the global grid has been varied. Figure 11(f) shows how the global grid resolution relates to the update time (solid blue line) and visual quality (dashed red line); computed using the VDP with a reference image computed using a global grid of the same resolution as the original data. For the images used in the comparison, the other parameters were kept fixed (4 SH-coefficients, 32 sample rays, 16 voxels local radius, and a 1:6.7 data reduction rate for the multi-resolution grid).

The local neighborhood radius affects the number of voxels traversed by each ray in the computation of the local visibility, $\tilde{V}^L(x, \omega_i)$, and the number of local neighborhoods traversed in the global visibility computation $\tilde{V}^G(x, \omega_i)$. Using a large local radius requires more computation of the local visibility, but less samples have to be evaluated to compute the global visibility. Figure 12 shows how the update time for global visibility varies with different radius using 32 sample rays (solid blue line) and 64 sample rays (dashed red line). For the data sets and TF settings used in this paper, see Figure 15, the optimal performance is obtained using a radius of around 12 - 16 voxels. It should be noted that the sampling density along each ray depends on the local block LOD in the multi-resolution grid.

5.2 Rendering performance

The rendering performance depends primarily on two factors: the number of spherical harmonic coefficients, and the type of light sources used. Figure 13 illustrates how the performance is affected by the number of SH coefficients used for the illumination computations in Eq. 7: using the Grace Cathedral environment map (solid blue line) and a single point light source (dashed red line). The number of coefficients varies from 1 (first order SH projection) to 16 (fourth order SH projection). It can be seen that increasing the number of SH coefficients has a significant effect on rendering performance. This is due to the fact that more computations and texture fetches are required at

Fig. 14. A Computed Tomography (CT) scan of a human heart rendered using (a) gradient based diffuse shading, (b) our method local visibility and (c) our method using global visibility.

Volume	Resolution	Visibility update (sec) Local/Global/Total	Rendering (fps)	Sample Rays	Local Radius	SH Coefficients	Data Reduction	Global Visibility Grid Resolution
Abdomen	$512 \times 512 \times 880$	0.47/1.16/1.63	30	32	16	4	1:15.6	$64 \times 64 \times 128$
Golden Lady	$512 \times 512 \times 624$	0.72/0.52/1.24	15	32	16	4	1:8.9	$128 \times 128 \times 128$
Heart	$512 \times 448 \times 416$	0.71/0.5/1.21	22	32	16	4	1:6.4	$64 \times 64 \times 64$
MRI Brain	$512 \times 512 \times 256$	0.7/0.21/0.91	24	32	16	4	1:4.4	$64 \times 64 \times 32$
Bonsai	$512 \times 512 \times 189$	0.73/0.75/1.48	21	64	16	4	1:6.7	$64 \times 64 \times 32$
Engine	$256 \times 256 \times 256$	0.35/0.1/0.45	7	32	16	9	1:2.8	$32 \times 32 \times 32$
Lobster	$301 \times 324 \times 56$	0.29/0.05/0.34	45	32	16	4	1:1.2	$64 \times 64 \times 32$
Fuel	$64 \times 64 \times 64$	0.05/0.1/0.15	6	128	16	16	1:4	$64 \times 64 \times 64$

Table 1. Performance and parameter settings for the scenarios displayed in Figure 1, Figure 14 and Figure 15. For the measured rendering performance a 1024×1024 viewport have been utilized.

Volume	Data Size (MB)	Reverse Index Map (MB)	Local Visibility SH Coeffs. (MB)	Global Visibility SH Coeffs. (MB)	Total Memory Requirement (MB)
Abdomen	29.58	29.58	118.30	4.19	181.65
Golden Lady	30.16	30.16	120.64	16.78	197.74
Heart	29.82	29.82	119.28	2.10	181.02
MRI Brain	30.50	30.50	122.02	1.05	184.07
Bonsai	14.79	14.79	59.16	1.05	89.79
Engine	11.98	11.98	143.80	0.79	168.55
Lobster	9.10	9.10	36.40	1.05	55.65
Fuel	0.14	0.14	2.10	8.39	10.77

Table 2. Storage statistics for the scenarios displayed in Figure 1, Figure 14 and Figure 15. The reported figures correspond to the sparse multiresolution representation stored in GPU memory.

each sample point in the volume during ray casting. The difference in performance between the point light and the environment map is due to that; for each point light it is necessary to recompute SH coefficients for each sample point along the ray, while for directional light sources and environment maps, only one SH rotation for the whole volume is necessary. All images rendered for the performance tests in Figure 13 were generated using the Bonsai tree data set using a data reduction rate of 1:6.8 for the multi-resolution grid.

6 RESULTS

To display the advantages of our technique, we present renderings of several data sets using complex lighting environments, Figures 1, Figure 14 and 15, along with a description of the runtime performance, Table 1. The results show detailed shading with local and global volumetric self-shadowing as well as translucency effects. Based on the evaluation performed in Section 5, the rendering parameters were set

to obtain a suitable tradeoff between speed and accuracy for practical use in a real-time rendering system. For all the displayed images, the radius, r_s , of the local visibility neighborhoods is set to 16 voxels. Generally, 4 or 9 Sh coefficients and 32 or 64 sampling rays have been used for rendering and visibility computations. All images are rendered with viewport resolution of 1024×1024 pixels on a PC equipped with dual Intel Xeon 3.2 GHz processors, 6GB of RAM and an NVIDIA GeForce GTX 295 graphics card with 896 MB onboard memory.

Figure 1 (a),(b),(c) displays three renderings of a data set from a Computed Tomography (CT) scan of an abdomen, where an artificial stent have been placed around the aorta. A comparison between the diffusely shaded rendering in (a), and the image using global visibility (b) shows that our method increases the depth perception, separation of large features and reveals local structure. This can be seen in the details of the heart and vessels, as well as the structure of the bones

Fig. 15. Volume renderings using the methods described in this paper. **Top row:** (a) Fuel injection, simulation data illuminated with one directional light. (b) CT scan of human head and torso, Golden Lady, illuminated with single directional light, (c) Rendering with clip plane using another TF for the same data set as in (b). **Middle row:** (d), (e) The Brain MRI-scan data set is illuminated by two light sources: one directional light and one point light. The directional light works as a rim light to enhance the silhouette of the brain, while the point light is interactively moved by the user to reveal local structures. (e) The method also enables accurate rendering with clip planes. In (f) the engine data set is rendered using one directional rim light and one point light to display local curvature and transparent materials. **Bottom row:** The Lobster in (g) is rendered using a single point light positioned above and to the right of the volume. The Bonsai Tree in (h) is illuminated by two point light sources to bring out the details in the foliage and branches. The Bonsai Tree (i) illuminated with the Grace Cathedral HDR light probe environment map.

and the stent. Using 4 SH coefficients at each grid point, the update to the displayed TF takes 3.2 seconds using 64 sample rays, 1.63 seconds using 32 sample rays, and with 9 SH coefficients 2.9 seconds using 32 rays.

Figure 14 (a),(b),(c) displays volume renderings of a human heart. The volume is illuminated by a single point light source. Comparing

the diffusely shaded rendering (left) to both the renderings using local (middle) and global (right) visibility it is clearly visible that the encoded directionally varying visibility presents strong visual cues that reveal both local details and global structures. This is especially noticeable from the overall increased depth perception and the details of the coronary arteries supplying blood to the heart muscle.

Technique	Dynamic Lighting	Directional Shading	Global Shadows	Ray-Casting Compatible	Complex Lighting Setups	Interactive* TF Update
Blinn-Phong with Gradients [7]	✓	✓	✗	✓	✗	✓
Half-Angle Slicing [14, 6, 34]	✓	✓	✓	✗	✗	✓
Ambient Occlusion [13, 5, 32]	—†	✗	✓	✓	—†	✓
Deep Shadow Maps [22, 9]	$\approx 1 - 2$ fps ‡	✓	✓	✓	✗	≈ 1 sec per light‡
Volumetric PRT [35, 30, 39]	✓	✓	✓	✓	✓	✗
Our method	✓	✓	✓	✓	✓	✓

Table 3. Qualitative comparison of the most similar related work. *By Interactive TF updates, we here refer to update times not significantly decreasing an interactive DVR workflow, e.g. our method performs TF updates in the order of $\approx 1s$, with an additional interleaved rendering to allow interactivity during updates. † For local ambient occlusion, no external light environment is considered. ‡ as reported in [9]

Figure 15 (a) shows a fuel injection simulation rendered with 16 SH coefficients using global visibility and one directional light source. The global visibility update for this volume takes only 0.15 seconds. Figure 15 (b), (c) displays renderings of the Golden Lady data set, using our methods for shading. In (b) the TF updates takes 1.24 seconds for full global visibility update and 0.72 seconds for only the local visibility update. In (c) a rendering with a cut plane is displayed, showing the blood vessels inside the skull.

The T1 weighted Brain MRI data set is shown in Figure 15 (d),(e), is illuminated by two point light sources. In (d), detailed local structure is revealed by positioning one of the lights behind the brain to act as a rim light, while interactively moving the second to explore the data. Our technique is, as displayed in (e), also suitable for rendering volumes with clip planes.

The Engine data set displayed in Figure 15 (f) is rendered using two light sources: one directional and one point light. The directional light illuminates the volume from behind, while the point light is used to interactively explore the data. The local structures can clearly be seen through the transparent material. The Lobster CT data set shown in figure 15 (g) renders at a sustained rate of 45 fps. Using 32 sample rays and 4 SH coefficients at each grid point, the TF update takes 0.3 seconds. Figure 15 (h) shows the Bonsai Tree data set rendered using two directional light sources with 64 sample rays and 4 SH coefficients at each grid point. Figure 15 (i) shows the Bonsai Tree data set rendered with the Grace Cathedral high dynamic range environment map.

Table 2 gives an overview of the storage statistics of each scenario. The figures display the run time memory footprint required for storing: the volume data, the SH coefficients for local and global visibility and the index maps for the multi-resolution grid (see Section 4.1 and Section 4.2). The SH coefficients are stored in texture memory as 16bit floats (half float). Note that all figures have been generated using the multi-resolution grid with the same transfer function settings as in Figure 15, and that the original resolution and specific reduction ratio of each data set is presented in Table 1.

7 DISCUSSION

Table 3 lists a qualitative comparison of our method with standard and other similar advanced shading approaches. In comparison to standard methods, such as diffuse shading based on gradients, our method increases spatial comprehension by incorporating local and global volumetric self-occlusion effects. Another advantage is that our method supports advanced shading in a ray-tracing pipeline. Although half-angle slicing techniques are efficient for a slice based renderer, texture based slicing fails to incorporate adaptive sampling, empty space skipping and other acceleration techniques, making them less attractive for data sizes commonly used in today’s practice. Furthermore, half-angle techniques have previously been limited to a single light source, and general lightning environments have not been considered.

Ambient occlusion methods represent a special case of our method, only using a single SH coefficient (DC term) and a uniformly bright environment lighting. Using only a single SH coefficient does not allow for directional shading effects, and limiting the illumination to uniform lighting removes the perceptual benefits of lighting setups.

Deep shadow maps represent the volumetric shading in the light source coordinate frame. However, moving the light source forces re-computation of the shadow representation, which should be compared to our method which only requires a fast SH rotation of the lightning environment, which does not affect the framerate. Furthermore, our method computes visibility in all directions, permitting general omnidirectional light environments, while the deep shadow map techniques do not scale well with several light sources.

Previous methods using PRT techniques have successfully incorporated advanced volumetric shading effects for general lighting environments. However, such methods require extensive precomputation each time the transfer function changes, making them far less suitable for interactive volume rendering than our method.

8 CONCLUSIONS AND FUTURE WORK

In this paper we have presented a method that enables fully interactive direct volume rendering with dynamic illumination, involving an arbitrary number and kind of light sources. The work constitutes an important step towards solutions that can lead to more general acceptance and wide spread use of advanced shading in direct volume rendering. The key to obtaining this unprecedented performance is the decoupling of the visibility information (Absorption of light in the volume) and the radiance distribution (Light information). The visibility information is efficiently encoded using Spherical Harmonics. Another key component is the separation of the visibility information into local and global contributions, and their representation on multi-resolution grids.

The method is described in the context of several examples, showing the image quality and the improved visual clarity over other, more basic, illumination models. The limitations and performance of the approach are documented by systematic exploration of parameter settings and corresponding benchmarking. It is emphasized that the rendering performance is insensitive to illumination changes, but a penalty in the order of 1s, depending on TF settings and data resolution, is introduced when the transfer function is changed. This does not, however, significantly interrupt the DVR workflow, especially since the visibility updates can be interleaved in the rendering.

There are several areas in which future work will be done to further improve the method. In this work we have limited the representation of global visibility to a regular grid, while using a multi-resolution grid for the local visibility representation. An adaptive global visibility representation, yielding higher quality shadows and faster response during TF changes, can be obtained by also using a multi-resolution grid for the global visibility. We will also explore the use of other basis functions, such as wavelets [17] to incorporate higher frequency effects. An interesting area for future work is the use the directional dependence encoded in the visibility representation to emulate multiple scattering effects. Previous related work in computer graphics and visualization has focused on automatic lighting design [16], optimizing the position and direction of several light sources to increase spatial comprehension of data. While these methods have previously been limited to polygonal meshes and isosurfaces, our method would enable lighting design algorithms to be used in interactive volume rendering. Investigating the union of these methods is an interesting path

for future work.

ACKNOWLEDGMENTS

We gratefully acknowledge Stefan Lindholm, Stefan Gustavsson, Tan Khoa Nguyen, Per Larsson, Frida Hernell and Erik Sundén for invaluable discussions. We would also like to thank the anonymous reviewers for their feedback. A special thank you goes to Matt Cooper for proofreading the manuscript. This work was supported through the Swedish Foundation for Strategic Research through the strategic research centre MOVIII grant A3:05:193, the Swedish Knowledge Foundation grant 2009/0091, Forskning och Framtid grant ITN 2009-00116, the Swedish Research Council through the Linnaeus Center for Control, Autonomy, and Decision-making in Complex Systems (CADICS), and the Excellence Center at Linköping and Lund in Information Technology (ELLIIT).

REFERENCES

- [1] K. Beason, J. Grant, D. Banks, B. Futch, and M. Hussaini. Pre-computed illumination for isosurfaces. In *IST/SPIE Symposium on Electronic Imaging (2006)*, pages 98–108, 2006.
- [2] U. Behrens and R. Ratering. Adding shadows to a texture-based volume renderer. In *IEEE Symposium on Volume Visualization*, pages 39–46, 1998.
- [3] A. Ben-Artzi, K. Egan, F. Durand, and R. Ramamoorthi. A precomputed polynomial representation for interactive brdf editing with global illumination. *ACM Transactions on Graphics*, 27(2):1–13, 2008.
- [4] J. Birn. *[digital] Lighting and Rendering*. New Riders, second edition, 2006.
- [5] P. Desgranges and K. Engel. Fast ambient occlusion for direct volume rendering. US patent application 2007/0013696 A1, 2007.
- [6] P. Desgranges, K. Engel, and G. Paladini. Gradient-free shading: A new method for realistic interactive volume rendering. In *VMV (Vision, Modeling, and Visualization)*, pages 209–216, 2005.
- [7] K. Engel, M. Hadwinger, J. M. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-Time Volume Graphics*. A K Peters, Ltd, 2006.
- [8] R. Green. Spherical harmonic lightning: The gritty details. In *Game Developers Conference*, 2003.
- [9] M. Hadwiger, A. Kratz, C. Sigg, and K. Bühler. Gpu-accelerated deep shadow maps for direct volume rendering. In *Graphics hardware*, 2006.
- [10] M. Hadwiger, P. Ljung, C. Salama, and T. Ropinski. Advanced illumination techniques for GPU volume raycasting. In *SIGGRAPH Courses*, 2009.
- [11] F. Hernell, P. Ljung, and A. Ynnerman. Efficient ambient and emissive tissue illumination using local occlusion in multiresolution volume rendering. In *IEEE Symposium on Volume Graphics*, pages 1–8, 2007.
- [12] F. Hernell, P. Ljung, and A. Ynnerman. Interactive global light propagation in direct volume rendering using local piecewise integration. In *IEEE/EG Symposium on Volume and Point-Based Graphics*, pages 105–112, 2008.
- [13] F. Hernell, P. Ljung, and A. Ynnerman. Local Ambient Occlusion in Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 16(4):548–559, 2010.
- [14] 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.
- [15] M. Langer and H. Bülthoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29:649–660, 2000.
- [16] C. H. Lee, X. Hao, and A. Varshney. Light collages: Lighting design for effective visualization. In *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE Visualization 2004)*, pages 281–288, 2004.
- [17] C. Lessig and E. Fiume. SOHO: Orthogonal and Symmetric Haar Wavelets on the Sphere. *ACM Transactions on Graphics*, 27(1), 2008.
- [18] F. Lindemann and T. Ropinski. Advanced light material interaction for direct volume rendering. In *IEEE/EG Symposium on Volume Graphics*, pages 101–108, 2010.
- [19] I. G. Lisle and S.-L. T. Huang. Algorithms for spherical harmonic lighting. In *GRAPHITE*, pages 235–238, 2007.
- [20] P. Ljung. Adaptive sampling in single pass, GPU-based raycasting of multiresolution volumes. In *IEEE/EG Workshop on Volume Graphics*, pages 39–46, 2006.
- [21] P. Ljung, C. Lundström, A. Ynnerman, and K. Museth. Transfer function based adaptive decompression for volume rendering of large medical data sets. In *IEEE Symposium on Volume Visualization and Graphics*, pages 25–32, 2004.
- [22] T. Lokovic and E. Veach. Deep shadow maps. In *SIGGRAPH*, pages 385–392, 2000.
- [23] J. Löw, A. Ynnerman, P. Larsson, and J. Unger. Hdr light probe sequence resampling for realtime incident light field rendering. In *SCCG (Spring Conference for Computer Graphics)*, 2009.
- [24] R. Mantiuk, S. Daly, K. Myszkowski, and H.-P. Seidel. Predicting Visible Differences in High Dynamic Range Images - Model and its Calibration. In *IST/SPIE Symposium on Electronic Imaging (2005)*, pages 204–214, 2005.
- [25] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [26] J. Moon, B. Walter, and S. Marschner. Efficient multiple scattering in hair using spherical harmonics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3):31–31, 2008.
- [27] F. Qiu, F. Xu, Z. Fan, N. Neophytou, A. Kaufman, and K. Mueller. Lattice-based volumetric global illumination. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE Visualization 2007)*, 13(6):1576–1583, 2007.
- [28] R. Ramamoorthi. Precomputation-based rendering. *Foundations and Trends in Computer Graphics and Computer Vision*, 3(4):281–369, 2009.
- [29] Z. Ren, R. Wang, J. Snyder, K. Zhou, X. Liu, B. Sun, P.-P. Sloan, H. Bao, Q. Peng, and B. Guo. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, pages 977–986, 2006.
- [30] T. Ritschel. Fast GPU-based Visibility Computation for Natural Illumination of Volume Data Sets. In *Eurographics*, pages 17–20, 2007.
- [31] T. Ropinski, C. Döring, and C. Salama. Interactive volumetric lighting simulating scattering and shadowing. In *PacificVis (IEEE Pacific Visualization)*, 2010.
- [32] T. Ropinski, J. Meyer-Spradow, S. Diepenbrock, J. Mensmann, and K. Hinrichs. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2):567–576, 2008.
- [33] C. Salama. Gpu-based monte-carlo volume raycasting. In *Pacific Conference on Computer Graphics and Applications*, pages 411–414, 2007.
- [34] M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, and K. Bouatouch. A directional occlusion shading model for interactive direct volume rendering. *Computer Graphics Forum (Proceedings of EG/IEEE Symposium on Visualization 2009)*, 28(3):855–862, 2009.
- [35] P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH*, pages 527–536, 2002.
- [36] P. Sloan, J. Snyder, and J. Hall. Clustered principal components for pre-computed radiance transfer. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 22(3):382–391, 2003.
- [37] X. Sun, K. Zhou, Y. Chen, S. Lin, J. Shi, and B. Guo. Interactive relighting with dynamic brdfs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):27, 2007.
- [38] C. Wyman, S. Parker, P. Shirley, and C. Hansen. Interactive display of isosurfaces with global illumination. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):186–196, 2006.
- [39] K. Zhou, Z. Ren, S. Lin, H. Bao, B. Guo, and H.-Y. Shum. Real-time smoke rendering using compensated ray marching. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3), 2008.