IEEE/ EG Symposium on Volume and Point-Based Graphics (2008)
H.- C. Hege, D. Laidlaw, R. Pajarola, O. Staadt (Editors)

Interactive Global Light Propagation in Direct Volume Rendering using Local Piecewise Integration

Frida Hernell[†1,2], Patric Ljung[‡3] and Anders Ynnerman[§1]

[1]Norrköping Visualization and Interaction Studio, Linköping University, Sweden
[2]Center for Medical Image Science and Visualization, Linköping University, Sweden
[3]Imaging & Visualization, Siemens Corporate Research, Inc. NJ, USA

Abstract

A novel technique for efficient computation of global light propagation in interactive DVR is presented in this paper. The approach is based on a combination of local shadows from the vicinity of each voxel with global shadows calculated at high resolution but stored in a sparser grid. The resulting intensities are then used as the initial illumination for an additional pass that computes first order scattering effects. The method captures global shadowing effects with enhanced shadows of near structures. A GPU framework is used to evaluate the illumination updates at interactive frame rates, using incremental refinements of the in-scattered light.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Color, shading, shadowing, and texture; I.4.10 [Image Processing and Computer Vision]: Volumetric;

1. Introduction

Direct Volume Rendering (DVR) has become an invaluable tool for improved interaction and understanding of complex structures in medical visualization. A well-known problem in current practice is the limited depth perception in DVR on standard displays. For instance, the relative distance and connectivity of intertwined blood vessels with complex structures at varying depths can sometimes be difficult to determine [HWSB99]. Since interaction with a global light source changes the visual appearance of structures and the projected shadows give intuitive cues for spatial relations inside the volume an improvement in depth perception can be achieved by the inclusion of advanced illumination with self-shadowing. Algorithms for realistic global illumination include direct and indirect illumination. This means that both light arriving directly from the light source and light reflected by other objects in the volume are taken into account. A photon can thus be scattered within the volume until it is absorbed. General methods for the solution of the full light transport equations are, however, computationally demanding and interactive frame rates are desired in many applications, including medical. Reaching interactive frame rates in medical DVR with approximated high quality global lighting is the main challenge and motivation for the work presented in this paper.

The presented method progresses in several steps to simulate the light transport in the volume and, at each step, captures the main physical contributions. In the first step opacities are composited into a global shadow volume for a given light source and transfer function (TF) settings. The calculation is based on piecewise integration techniques on the GPU and sparse representations of intermediate results. The method then proceeds to include first order scattering of the global light arriving at each voxel by integration of scattered light in a local spherical neighborhood around each voxel. This step is similar to the local ambient occlusion (LAO) method described in [HLY07], which enhances perception of local shapes and tissue properties. In the last step a single pass ray-casting approach is used to render the final image. The resulting application enables updates of light position and transfer functions while maintaining interactive frame rates yet simulating a realistic light model. The main contributions of this paper are:

† frida.hernell@cmiv.liu.se
‡ patric.ljung@siemens.com
§ anders.ynnerman@itn.liu.se

- an efficient approximation of volumetric propagation of light from a point light source using local piecewise integration.
- support for illumination calculations of large scale volumes due to the inclusion of multi-resolution data management
- a fast approach for refinements of first order scattering supporting phase functions
- support for interactive and arbitrary positioning of the light source, outside as well as inside the volume.

The remainder of this paper is organized as follows: An overview of related work in this research field is given in section 2. The theory and implementation details are presented in sections 3 and 4, respectively. Visual and performance results are described in section 5. The paper is finally concluded with a discussion and outlook on future work.

2. Related Work

Over the past decades a vast amount of research has been published in the area of realistic lighting, for instance [Bli82, KH84, Max94, Rus88, JC98]. The focus of our paper is to compute global light in participating media at interactive frame rates. In this section we give an overview of previous research with the same goal.

Calculations of global illumination from a light source in volume rendering applications are often pre-computed and are commonly stored in an additional 3D texture [BR98]. Hadwiger et al. [HKSB06] presented a deep shadow map in which pre-compressed visibility functions are stored. Kniss et al. [KPH*03] presented an efficient method that computes attenuation of light interleaved with the rendering of the volume using texture slicing. A half-angle approach is used that slices the volume at an angle between the light source and the view direction. Limitations of their method are that only one light source can be applied, the rendering approach of texture slicing is required and only forward scattering, approximated with a blur-function, is possible. Desgranges et al. [DEP05] proposed a similar method but also integrated dilation of light reducing hard shadow edges and enhancing translucent appearances. Qiu et al. [QXF*07] presented a method that uses a Face-Centered Cubic lattice for increased sampling efficiency. In their method computations of multiple scattering are simplified through spatial and angular discretization on lattices. However, their approach is still too slow to reach interactive frame rates. Interactive volume rendering with dynamic ambient occlusion using local histograms was presented by Ropinski et al. [RMSD*08]. Their method requires a preprocessing time of approximately 2 days for a volume of 512x512x294 voxels.

The method described in our paper approximates propagation of light with single scattering effects based on the approach of local ambient occlusion (LAO) [HLY07]. LAO is an efficient approximation of ambient occlusion from a local volumetric light source that surrounds each voxel. The intensity of incident light is derived by integrating the optical depth along rays cast from the voxel towards the sphere in a number of directions. Promising results for enhanced perception of shapes and tissue properties can be achieved with LAO, however, the improved depth perception is limited to local features since each voxel is only occluded by structures in the vicinity.

Further speed-up of the method described in our paper is achieved by utilizing local piecewise integration and a flat blocking multi-resolution data management [LLYM04]. With this data management the volume is divided into small blocks of 16^3 voxels which are stored in a multi-resolution structure. The blocks are assigned a level-of-detail (LOD) which determines the resolution of the block in the multi-resolution structure. This data reduction is based on preservation of the resulting image quality in the current TF domain and non-contributing/empty blocks are ignored. The blocks are, in the multi-resolution structure, not organized according to their spatial location but their LOD.

3. Global Lighting with Scattering Effects

The method presented in this paper constitutes an approximation of the the distribution of global light in participating media including first order scattering effects. The goal of the approach is to support interactive frame rates both during and in between illumination updates. As our starting point, we take the classical volume rendering integral [Max95]:

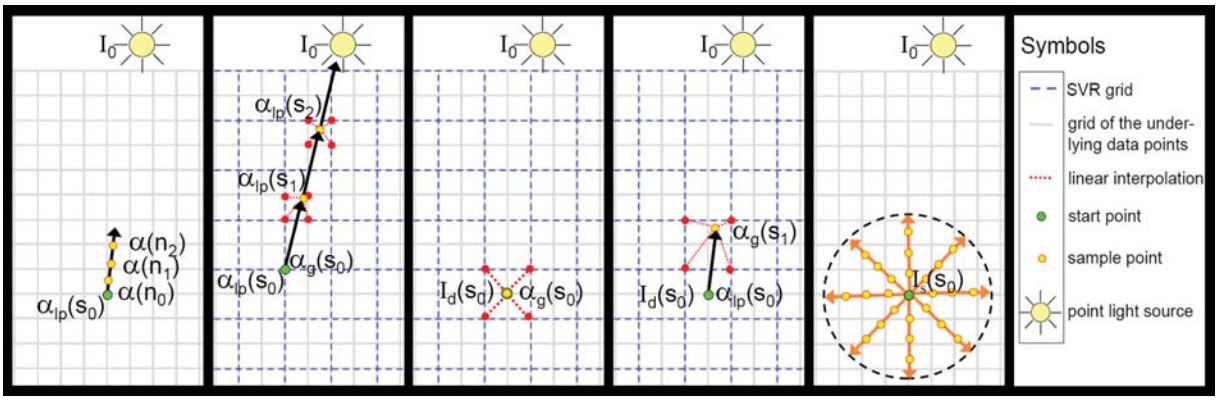
$$I(D) = I_0 \cdot e^{-\int_0^D \tau(t)dt} + \int_0^D g(s) \cdot e^{-\int_s^D \tau(t)\mathrm{d}t}\mathrm{d}s \qquad (1)$$

I_0 is the background light intensity, τ is the absorption coefficient and $g(s)$ is the source term that describes the illumination model. The reflected color is modulated based on the lighting conditions, I, according to:

$$g(s) = I(s) \cdot c(s) \cdot \alpha(s) \qquad (2)$$

where $c(s)$ and $\alpha(s)$ are the color and opacity at the current location, s, along the ray. Simulating fully realistic lighting is computationally demanding. Most approaches for obtaining $I(s)$ thus do not take global light transport into account. The most commonly used local lighting model in DVR is the Blinn-Phong shading model [Bli77] which includes three terms: ambient, diffuse and specular. In our illumination model $I(s)$ is the sum of both direct and indirect light contributions, as in [KPH*03], but in our case with a physically-based estimation of the first order scattering. Our approach to the calculation of $I(s)$ is based on four steps that are carried out to compute the illumination of the volume:

1. *Enumeration of the opacity along piecewise ray paths*
2. *Computation and representation of a global shadow volume*
3. *Estimation of the global light contribution to each voxel*
4. *Inclusion of local first order scattering effects*



(a) (b) (c) (d) (e)

Figure 1: *Opacity for a local segment is integrated in (a) and global opacity is integrated in (b). A simple method to compute I_d is to perform a direct interpolation from α_g, as in (c), however, a better approximation is obtained, in (d), by utilizing the piecewise integration as an initial step. The quality of nearby shadow contributions is thereby increased. In-scattering of the initial intensity approximation, I_s, is illustrated in (e).*

The resulting $I(s)$ is then used in interactive ray casting DVR to illuminate each sample point, in which $I(s)$ only needs to be updated when the TF or light settings change. The following subsections, 3.1-3.5, describe the details of the separate steps of the method.

3.1. Local Piecewise Integration

In the first step of the algorithm, opacity is integrated in a user defined local neighborhood by casting rays from each voxel simultaneously in the direction of the light source. The purpose of calculating these local piecewise opacity segments is to enable high resolution shadows from occluding objects in the vicinity of each voxel and also to form the basis for fast calculations of global opacity.

We start by examining the intensity of the direct light that reaches a position, s, along a ray from a light source with intensity I_0:

$$I(s) = I_0 \cdot e^{-\int_0^s \tau(t)\mathrm{d}t}. \tag{3}$$

By dividing the integral into k segments of length Δs, eq. 3 can be rewritten as:

$$I(s) = I_0 \cdot \prod_{n=0}^{k} T_{lp}(s_n) \tag{4}$$

where

$$T_{lp}(s_n) = e^{-\int_{s_n}^{s_{n+1}} \tau(t)\mathrm{d}t} \tag{5}$$

and $s_n = n\Delta s$, where s_0 originates at the current voxel location. Eq. 5 [Max95] is then discretized into eq. 6, where α_{lp} is the opacity of the segment that originates in point s_n and ends in s_{n+1}. The opacity, α_{lp}, is related to the transparency as $\alpha_{lp} = 1 - T_{lp}$.

$$\alpha_{lp}(s_n) = 1 - \prod_{i=0}^{N}(1 - \alpha(n_i)) \tag{6}$$

where N is the number of samples taken along the segment and $\alpha(n_i)$ is the opacity of a point, n_i, of the segment in the current TF domain. $\alpha(n_i)$ can be adjusted with the opacity correction formula [PH89] for further control of the attenuation impact.

The first step in the proposed method computes $T_{lp}(s_n)$ for $n = 0$ for each voxel location in the volume separately. The resulting opacities, $\alpha_{lp}(s_0)$, constitute the piecewise segments that will be reused throughout the algorithm. Detailed local opacity segments are thus obtained for each grid point, see also the illustration in fig.1a. These segments must be recomputed each time the transfer function is changed or the volume is moved in relation to the light source but, as will be shown in section 5, interactivity can still be maintained. The step length in the global light integration is equal to the length of the local piecewise segments. The optimal length, providing a good balance of high quality and high performance, is discussed in the results section.

The concept of piecewise integration is illustrated for a 1D example in fig. 2, starting with the opacity values along a ray cast in the direction of the light source. Fig. 2b then shows how the integrated opacity builds up along a few segments of the ray. The results of the local piecewise integration is the opacity of each segment (see fig. 2c). A detailed description of how to compute the global integral (fig. 2d), using the local integrations, is provided in section 3.2.

3.2. Global Shadow Volume

In this step the opacities, α_{lp}, obtained in the local piecewise integration for each voxel in the volume, are used to obtain

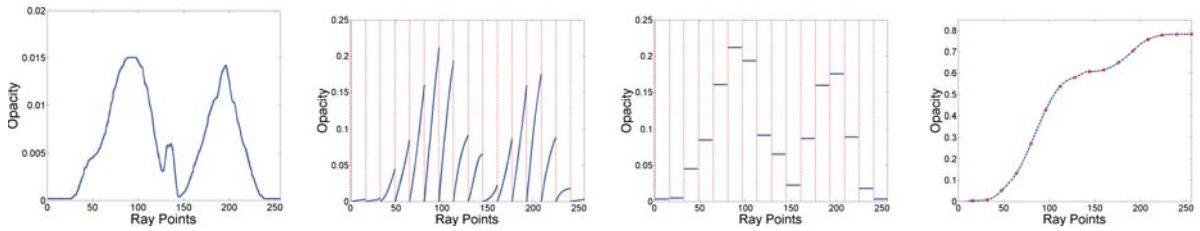


Figure 2: *The opacities for each point along a ray, in one-dimension, are illustrated in (a). Opacity of local segments is integrated in (b) and the resulting opacities for each segment are shown in (c). Computing opacity along the whole ray, in (d), from the first point to the light source, is performed by sampling (red circles) the opacities in (c), which allows for a sparse sampling density. Note that the local piecewise integration must be performed and stored for each point along the ray to allow the global sampling to start at an arbitrary point.*

a global shadow volume that is stored in a sparse data structure. The global opacity, α_g, is determined by computing the product of the transparency for each segment, as indicated in eq. 4. The global opacity can be computed as:

$$\alpha_g(s_0) = 1 - \prod_{i=0}^{k}(1 - \alpha_{lp}(s_i)) \tag{7}$$

Since linear interpolation is used to estimate local segments, illustrated in fig. 1b, some small errors are introduced.

To speed up the calculations it is possible to reduce the resolution of the shadow volume representation (SVR) used to store the computed results. The ray integrations are, however, still performed at high resolution to maintain the quality of the visibility estimates along each path. It is usually sufficient to use a grid resolution at 1:4, that is each sample in the SVR covers 4^3 voxels in the data volume. Since this resolution can be adjusted interactively it is possible to compute the global distribution for a lower number of points during interaction and incrementally increase the resolution of the SVR when the viewpoint or light source is static.

3.3. Global Light Contribution

For each voxel in the volume the direct light contribution from the source is now estimated. Paths of light rays converge close to the light source and large differences in the occlusion between neighboring voxels therefore primarily depend on objects in the near vicinity. Therefore, to improve the accuracy of the computations of the direct light contribution, $I_d(s_0)$ in eq. 8, the opacity of the first ray segment is taken from the previously computed local piecewise integration, $\alpha_{lp}(s_0)$. The remaining opacity, $\alpha_g(s_1)$, from point s_1 along the ray towards the light source is interpolated from the SVR (see fig. 1d).

$$I_d(s_0) = I_0 \cdot (1 - \alpha_g(s_1)) \cdot (1 - \alpha_{lp}(s_0)) \tag{8}$$

This minor additional computational cost improves the final quality, compared to only using α_g (see fig. 1c), as it ensures that local occlusion is calculated at the highest possible resolution. The intensity, I_0, is a user defined parameter.

3.4. First Order Scattering

In the final step of the illumination calculation, in-scattering from all directions on a sphere to each point in the volume [Max95] is included. To approximate the in-scattering contribution we use the results from eq. 8 and treat the scattered part of the direct global light as emittance (eq. 9). In

$$I_e(s_0) = \int_{s_0}^{R_\Omega} q(s) e^{-\int_{s_0}^{r} \tau(t)\mathrm{d}t} \mathrm{d}r \tag{9}$$

$q(s)$ is thus taken from the direct global light distribution and R_Ω is the radius of the scattering sphere, Ω. Since the first order scattering is treated as LAO [HLY07] only short range scattering effects are captured (see fig. 1e) which are the dominant contributions and also, from the medical perspective, the most relevant ones. The complete integral for the in-scattering contribution becomes:

$$I_s(s_0) = \int_\Omega \varphi(\omega) \int_{s_0}^{R_\Omega} I_d(r) \cdot e^{-\int_{s_0}^{r} \tau(t)\mathrm{d}t} \mathrm{d}r \mathrm{d}\omega \tag{10}$$

A discretization of this integral gives:

$$I_s = \sum_{j=0}^{J} \varphi_j \sum_{m=0}^{M} I_d(s_m) \prod_{i=0}^{m-1} (1 - \alpha(s_i)) \tag{11}$$

J is the number of ray directions and M is the number of sampling steps along each ray. The light contribution from each direction, j, is normalized to sum to one. A phase function, φ, can be used to weight the incoming light from different directions [Max95]. An isotropic phase function is used in this paper, which is the simplest case, weighting the light equally in all directions.

3.5. Volume Ray Casting

Having obtained the scattered intensity of each voxel, the final intensity of a pixel in the image, I_{eye}, is computed using

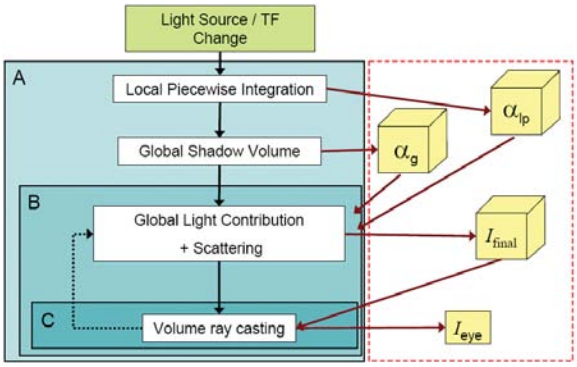


Figure 3: *GPU flow chart of the presented methods, which includes local piecewise integration, global light integration and estimations of in-scattering. The last step is incrementally refined until the intensity from all the in-scattering directions are updated. The objects in the dotted square are the intermediate results of the shader programs.*

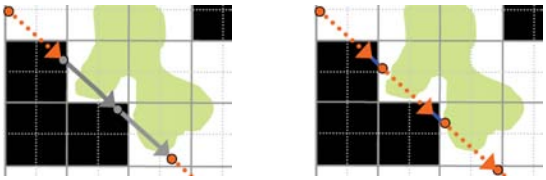


Figure 4: *Occluding objects can be missed when computing α_g in the left image since opacity is not pre-computed for rays (gray) that originate in empty blocks (black squares) ignored by the multi-resolution structure. This problem is avoided by advancing the ray (blue line) to the next non-empty block, as shown in the right image, for this special case. The samples taken are marked with orange dots.*

a Riemann sum to discretize the ray casting integral:

$$I_{eye} = \sum_{k=0}^{n} g_k \prod_{p=0}^{k-1} (1 - \alpha(s_p)) \tag{12}$$

where g_k is

$$g_k = \underbrace{I_{bias} + (1 - I_{bias}) \cdot I_s(s_k)}_{I_{final}} \cdot c(s_k) \cdot \alpha(s_k)$$

c and α are the color and opacity, for each point, s_k, according to the current TF domain. A parameter, I_{bias}, in the range [0..1], is introduced in order to adjust the minimum light level. The light source is considered to distribute only white light but the equations can easily be extended to handle colored light.

4. Implementation

This section describes the implementation utilized to efficiently solve the equations presented in the previous section. Three shader programs have to be executed to update the light computations. The intermediate results obtained by these programs are volumes containing α_{lp} (eq. 6), α_g(eq. 7) and I_{final} (eq. 12), respectively, as illustrated in fig. 3. To fill these three volumes each slice must be rendered, one-by-one, to a frame buffer object (FBO) that is attached to the corresponding slice in the respective 3D texture target. An additional shader program is used for the final ray casting step, computing I_{eye}. To speed up the interaction during an illumination update the stored light, I_{final}, does not have to include calculations for the whole scattering sphere at once. The in-scattering contribution can be progressively refined by adding computations for one additional direction, each time I_{eye} is updated, until all directions are computed. The direction of the light source is computed first so that the shadows seen during interaction are the best approximation of the final illumination. The subsequent directions of the scattering sphere are found by subdividing a tetrahedron, icosahedron or octahedron to a level that gives the desired number of rays. For a correct incremental update the new results of I_{final} have to be blended (eq. 13) with the previously stored intensities, I_{stored}. $I_{blended}$ is the new value to store and j is the current ray among the J rays used to discretize the scattering sphere.

$$I_{blended} = \frac{1}{j} \cdot I_{final} + (1 - \frac{1}{j}) \cdot I_{\text{stored}} \tag{13}$$

The calculations performed in the presented method result in a large amount of data since each step needs to store computations for each voxel. For example, if the illumination is computed for a volume of 512^3 then three additional 512^3 volumes are needed to store the results of the α_{lp}, α_{g} and I_{final} computations. However, it is not reasonable to use this much space for light calculations. As mentioned in 3.2 the resolution of the SVR can be reduced, which implies that neighboring points use the same global integration. Further reduction of storage space is obtained by employing a flat blocking multi-resolution data management [LLYM04]. This approach gives the possibility to reduce the amount of data to process without significant loss in visual quality (see section 2). With this approach a medical dataset of 512 cubed voxels can commonly be stored using 256x256x128 voxels without significant loss of important information. By performing the light computations directly in the coordinates of the multi-resolution data structure, as in [HLY07], it is possible to skip calculations for regions that are empty. It is therefore feasible to reduce the texture sizes by at least a factor of eight, and the number of render passes required for an illumination update decreases greatly. Interpolation problems can, however, occur since the blocks in the multi-resolution structure are not packed according to their spatial location. Potential artifacts can, in most cases, be avoided or reduced by clamping the sample locations to a proper distance from the block borders [LLY06]. This solution is sufficient to prevent artifacts in α_{lp} and I_{final}. Since the data storage in the SVR is allowed to be reduced by a large factor, however, clamping is not a possible solution when computing the global opacity, α_{g}. In this case it is therefore nec-

Data reduction	A		B	C
	32^3	256^3		
8.9:1	284	552	233	68
14.8:1	178	515	145	68
22.1:1	121	403	96	48
35.2:1	81	365	62	46

Piecewise segment length (voxels)	A			
	32^3	64^3	128^3	256^3
4	261	267	439	1428
8	284	297	373	552
16	331	339	380	641
32	436	439	463	862

Table 1: *Performance, in milliseconds (ms), for different levels of illumination updates ((A), (B) and (C) as in fig. 3) versus varying data reductions (left table). The piecewise segments are 8 voxels long. In the right table the performance is shown for different sizes of the SVR versus varying lengths of the local piecewise segments for light update (A). The data reduction is 8.9:1. Measurements in both tables are performed for a volume of* 512^3 *voxels, rendered in a 1024x1024 window. The same volume, TF and rendering settings are used in fig. 7, using a step length of 16 voxels.*

(a) $I_{bias} = 0$, $R_\Omega = 16$ *(b)* $I_{bias} = 0$, $R_\Omega = 48$ *(c)* $I_{bias} = 0.2$, $R_\Omega = 16$ *(d)* $I_{bias} = 0.2$, $R_\Omega = 48$

Figure 5: *Light is integrated, for a CT scan of a carp, with varying settings for* I_{bias} *and* R_Ω*. The shadows becomes less distinct with a large radius (denoted in units of voxels) of the scattering sphere,* Ω*.* I_{bias} *adjusts the minimum intensity of each voxel.*

essary to compute the estimates in a regular linear volume so that linear interpolation can be used without causing artifacts. Another problem that arises with the flat blocking data management is that blocks that do not include any content in the TF domain are ignored in the multi-resolution structure. Local rays that originate in the empty blocks might, in the computation of α_g, cover parts of neighboring blocks that are not empty. The occlusion in those blocks will then be lost and artifacts can appear. For this reason, the ray is advanced to the next non-empty block if a sample point in eq. 7 is located in an empty block. An illustration of this problem is shown in fig. 4.

5. Results

The results shown in this section have been generated using a standard PC equipped with an Nvidia GeForce 8800 Ultra graphics board with 768 MB of graphics texture memory. The overall performance of the presented illumination method depends mainly on the size of the SVR, the length of the piecewise segments and the data reduction in the multi-resolution data management. Speed-ups due to an increased data reduction are shown in table 1(left). This table also shows performance of the different update levels, (A), (B) and (C), illustrated in fig. 3. (A) is updated once when the TF or the light conditions are changed, (B) is updated once for each refinement of the in-scattered light and (C) shows the performance of a rendering with an already existing light computation. The right table show timings of varying sizes of the SVR versus different lengths of the piecewise segments for the update level (A). Visual error due to the data reduction, of 8.9:1 used in this table, is hardly noticeable with the employed TF. As can be seen in table 1(right), the length of 8-16 voxels, with these settings, gives significantly better performance. With a length of 8 voxels the global light integration is updated in 284-552 ms, depending on the SVR size, and 62-233 ms for first order scattering incremental refinements. High frame rates, 15-22 frames per seconds (fps), are reached once the light calculations are updated, since only one additional texture look-up must be performed for each sample during ray casting.

The variation in illumination effects obtained when adjusting the user defined parameters can make important features in the volume more prominent. Fig. 5 illustrates how the parameters I_{bias} and R_Ω affect the illumination. I_{bias} adjusts the minimum lighting so that no region is completely dark and an increase of the scattering sphere radius, R_Ω, results in less distinct shadows. The shadows are, in this implementation, generally quite indistinct since an isotropic phase function is used.

Approximating the absorption integral using local piecewise linear integration can introduce interpolation errors. A volume rendered without this approximation is used to analyze artifacts in fig. 6. Integration of the light intensities, for each point in the volume to the light source, without the local piecewise method, is approximately 10 times slower. An error image, using a perceptually adapted color error in the CIE 1976 L*u*v* color space (CIELUV) [Fai98, LLYM04], is shown in fig. 6c. The scale of the pixel-wise error, ΔE, has a Just Noticeable Difference (JND) at 1.0. The errors in this comparison are very small, with a root mean square (RMS) of 0.19 for ΔE, and mainly appear at sharp edges and thin structures. An RMS measure, however, captures individual pixels with high distortion quite poorly. The rate of pixels with a ΔE above 6, ΔE_6, is therefore also reported (as

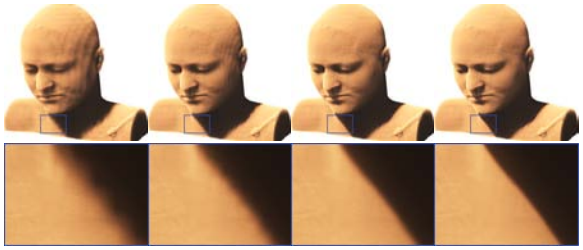
(a) Opacity integration for each point to the light source (5110 ms). *(b) Opacity computed with piecewise integrations (545 ms).* *(c) Pixel-wise error, ΔE*

Figure 6: *A comparison of illumination integrated for each point to the light source, in (a), with intensities computed with our approximation using piecewise integration, in (b). The volume and the SVR are 512^3 voxels, the view-port is 1024x1024 pixels and no in-scattering is considered. A color-coded error image that shows the pixel-wise error, ΔE, is provided in (c). The errors that appear are small (ΔE_{RMS} = 0.19 and ΔE_6=6%), and mainly appear at sharp edges and thin structures.*

in [LLYM04]), which is 6% for the measurements in fig. 6c. The size of the SVR used to store α_g is, in these calculations, equal to the size of the volume. A reduction of the SVR size results in a performance increase but with a potential quality loss. Block artifacts, in the form of jagged edges, appear on sharp edges and surfaces if too low a resolution is chosen (see fig. 7). Also, banding effects appear on the side of the head due to the interpolation between shadowed and un-shadowed regions. This image series is used for the measurements in table 1 using a piecewise segment length of 16 voxels. A significant reduction of the SVR size is possible during interaction since a sufficient context of the illumination is still maintained.

6. Conclusions

Local piecewise integration is employed to efficiently compute the propagation of light from a global light source towards each voxel in a volumetric dataset. The method is interactive and the obtained scattering effects are nearly physically correct, compared to Kniss et al. [KPH*03] that utilizes only a blur function. Additional advantages of our method are that the light source can easily be positioned anywhere in the scene, even within the volume (fig. 8), and light can be computed for large scale volumes since a multi-resolution data management scheme is used. These possibilities are difficult to achieve with methods that compute light propagation in object order from the light source. The realism of the presented method can easily be further increased by allowing different phase functions for different tissue types, instead of using a simple isotropic phase function. The method can also easily be extended to use up to four light sources by casting four rays simultaneously and using the storage space in the three additional channels of each texture. In future work it would also be interesting to extend the method for multiple scattering. This would be possible with an incremental update of I_{final}. However, the phase functions used after the first order scattering must be limited to isotropic scattering since no information of scattering directions is stored.



(a) $16^3 \rightarrow 1$ *(b)* $8^3 \rightarrow 1$ *(c)* $4^3 \rightarrow 1$ *(d)* $2^3 \rightarrow 1$

Figure 7: *Block artifacts appear due to reduced size of the shadow volume representation. The original volume size is 512^3 with a data reduction of 8.9:1 and the SVR is computed for (a) 32^3 (b) 64^3 (c) 128^3 and (d) 256^3 voxels. Jagged edges appear in (a) and (b), as can be seen in the close ups. Also, band artifacts arise at the side of the head. Computation times for these images are as shown in table 1. The length used for the piecewise segments is 16 voxels.*

Shadows from a global light source emphasize anatomical structures that otherwise can be difficult to see, but it can be debated whether global shadowing is beneficial in medical applications since important areas such as tumors could be obscured. With control of the minimum intensity, I_{bias}, of each point in the volume it is possible to see features without missing important structures due to complete occlusion (see fig. 5). Interactively moving the light source further decreases the problem of overly shadowed regions. In addition, global illumination does not have to exclude other shading methods but can rather serve as a complement. The best solution might be to allow the user to efficiently blend local and global shading effects for an optimal exploration of features in a dataset. In medical applications it is, in many cases, preferable to allow static illumination, fixing the light source relative to the volume. The light intensities need then only be recomputed when the transfer function is changed or a new light position is set with respect to the volume of data. A comprehensive user study will be needed to evaluate the use of this method in medical applications to ensure

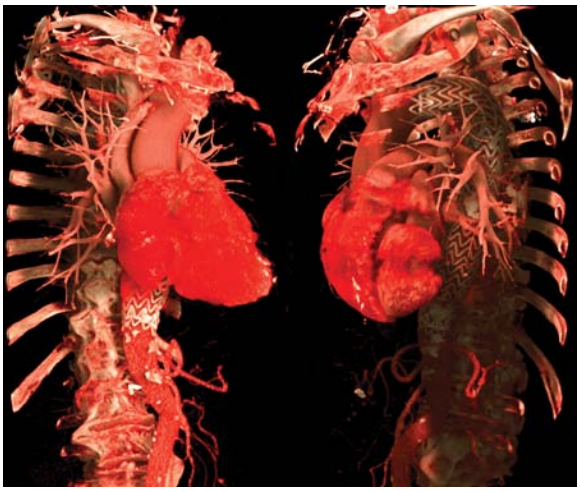


Figure 8: *Two examples, with different positions of the light source, using the presented illumination technique for a medical volume (512^3 voxels) of a heart. A stent has been inserted into the aorta. ($I_{bias} = 0.1$, $R_\Omega = 16$ and $J = 32$)*

clinical usefulness and value. A pilot study has, however, already been conducted on twelve test persons, comparing the perceived depth using the presented illumination method and the gradient based diffuse shading that is common in medical applications today. One of the tasks in the study was to determine the closest of two blood vessels in a synthetic dataset. The results of the study indicate that less errors and faster answers are given using the presented illumination method.

Acknowledgements

This work has been funded by the Strategic Research Center MOVIII, founded by the Swedish Foundation for Strategic Research, (SSF). The data sets used are provided by the Center for Medical Image Science and Visualization (CMIV), Siemens and The Volume Library (http://www9.informatik.uni-erlangen.de/External/vollib/). Many thanks to co-workers at the division for Visual Information Technology and Applications.

References

[Bli77] BLINN J. F.: Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph. 11*, 2 (1977), 192–198.

[Bli82] BLINN J. F.: Light reflection functions for simulation of clouds and dusty surfaces. In *Proc. of SIGGRAPH '82* (New York, NY, USA, 1982), ACM, pp. 21–29.

[BR98] BEHRENS U., RATERING R.: Adding shadows to a texture-based volume renderer. In *IEEE Symposium on Volume Visualization* (1998), pp. 39–46.

[DEP05] DESGRANGES P., ENGEL K., PALADINI G.: Gradient-free shading: a new method for realistic interactive volume rendering. In *Proc. of Vision, Modelling, and Visualization* (Nov. 2005). Berlin, Akademische Verlagsgesellschaft Aka GmbH.

[Fai98] FAIRCHILD M. D.: *Color Appearance Models*. Addison Wesley Longman, Inc., 1998.

[HKSB06] HADWIGER M., KRATZ A., SIGG C., BÜHLER K.: Gpu-accelerated deep shadow maps for direct volume rendering. In *GH '06: Proc. of SIGGRAPH/Eurographics symp. on Graphics hardware* (New York, NY, USA, 2006), ACM, pp. 49–52.

[HLY07] HERNELL F., LJUNG P., YNNERMAN A.: Efficient ambient and emissive tissue illumination using local occlusion in multiresolution volume rendering. In *Proc. of Eurographics/IEEE Volume Graphics 2007* (2007), pp. 1–8.

[HWSB99] HUBONA G. S., WHEELER P. N., SHIRAH G. W., BRANDT M.: The relative contributions of stereo, lighting, and background scenes in promoting 3d depth visualization. *ACM Trans. Comput.-Hum. Interact. 6*, 3 (1999), 214–242.

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH '98: Proc. of Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM, pp. 311–320.

[KH84] KAJIYA J. T., HERZEN B. P. V.: Ray tracing volume densities. In *SIGGRAPH: Proc. of Comput. graph. and interactive techniques* (New York, USA, 1984), ACM, pp. 165–174.

[KPH*03] KNISS J., PREMOZE S., HANSEN C., SHIRLEY P., MCPHERSON A.: A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 150–162.

[LLY06] LJUNG P., LUNDSTRÖM C., YNNERMAN A.: Multiresolution interblock interpolation in direct volume rendering. In *Proc. of Eurographics/IEEE Visualization* (2006), pp. 259–266.

[LLYM04] LJUNG P., LUNDSTRÖM C., YNNERMAN A., MUSETH K.: Transfer function based adaptive decompression for volume rendering of large medical data sets. In *Proc. of IEEE Volume Visualization and Graphics* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 25–32.

[Max94] MAX N. L.: Efficient Light Propagation for Multiple Anisotropic Volume Scattering. In *Fifth Eurographics Workshop on Rendering* (Darmstadt, Germany, 1994), pp. 87–104.

[Max95] MAX N.: Optical models for direct volume rendering. *IEEE Trans. on Visualization and Computer Graphics 1*, 2 (1995), 99–108.

[PH89] PERLIN K., HOFFERT E. M.: Hypertexture. In *SIGGRAPH '89: Proc. of Computer graphics and interactive techniques* (New York, NY, USA, 1989), ACM, pp. 253–262.

[QXF*07] QIU F., XU F., FAN Z., NEOPHYTOU N., KAUFMAN A. E., MUELLER K.: Lattice-based volumetric global illumination. *IEEE Trans. Vis. Comput. Graph. 13*, 6 (2007), 1576–1583.

[RMSD*08] ROPINSKI T., MEYER-SPRADOW J., DIEPENBROCK S., MENSMANN J., HINRICHS K. H.: Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Eurographics 2008) 27*, 2 (2008), 567–576.

[Rus88] RUSHMEIER H. E.: *Realistic image synthesis for scenes with radiatively participating media*. PhD thesis, Ithaca, NY, USA, 1988.