

# Lighting Design for Globally Illuminated Volume Rendering

Yubo Zhang, *Student Member, IEEE*, and Kwan-Liu Ma, *Fellow, IEEE*

**Abstract**—With the evolution of graphics hardware, high quality global illumination becomes available for real-time volume rendering. Compared to local illumination, global illumination can produce realistic shading effects which are closer to real world scenes, and has proven useful for enhancing volume data visualization to enable better depth and shape perception. However, setting up optimal lighting could be a nontrivial task for average users. There were lighting design works for volume visualization but they did not consider global light transportation. In this paper, we present a lighting design method for volume visualization employing global illumination. The resulting system takes into account view and transfer-function dependent content of the volume data to automatically generate an optimized three-point lighting environment. Our method fully exploits the back light which is not used by previous volume visualization systems. By also including global shadow and multiple scattering, our lighting system can effectively enhance the depth and shape perception of volumetric features of interest. In addition, we propose an automatic tone mapping operator which recovers visual details from overexposed areas while maintaining sufficient contrast in the dark areas. We show that our method is effective for visualizing volume datasets with complex structures. The structural information is more clearly and correctly presented under the automatically generated light sources.

**Index Terms**—Global illumination, lighting design, volume rendering, tone mapping

## 1 INTRODUCTION

In volume data visualization, lighting design is a process of setting up light sources for volume rendering. The goal is to enhance the visual perception of the volume data. In real world, human eyes can perceive objects better with the assistance of light. It is also true for volume rendering if light sources are placed appropriately. For example, local structures of isosurfaces can be perceived through local diffuse shading and ambient occlusion. Depth order and thickness can be perceived through global shadow casting and multiple scattering. Lighting design is as important as transfer-function design in volume visualization. Because a good lighting can improve the effectiveness of the visualization while a bad lighting may give a wrong impression of the volume dataset to the user.

In conventional volume visualization systems, setting up the light sources is often done manually by the user. However, this requires the user to have related knowledge in computer graphics including the shading models and light behaviors as well as the knowledge in arts. It takes time to setup a good lighting environment even for experienced users. The number of light sources, the position, direction and intensity of each light source are all unknown variables which need to be determined. Although a simple directional light can illuminate the volume data, it is not optimal and sometimes can give misleading results. On the other hand, a nicely designed lighting environment can present the volume structural information correctly and accurately, which greatly improves the effectiveness of the volume visualization.

Due to the limitation of graphics hardware in the early years, interactive volume visualization systems generally adopt simple local illumination models (e.g. local Phong shading) for rendering volume datasets. Therefore, previous studies on lighting design for volume data visualization are also based on local shadings. These proposed techniques mainly focus on enhancing the details of local surfaces and optimize the visual perception on the local structures. However, such methods do not fully exploit the benefit from global illumination and hence, may not produce optimized solutions for globally illuminated volume rendering. For example, the depth order of a cluttered volume dataset may not be well presented. Since global illumination has already been proven useful for volume data visualization [20, 21, 11],

we have developed a practical automatic lighting design system for such cases which takes global shadow casting and multiple scattering into account.

In order to achieve interactive global illumination for volume rendering, we adopt a high quality real-time volume rendering technique which fully models light propagation, absorption and scattering under multiple light sources [32]. Based on the renderer, we implemented a three-point lighting system where the key light and the fill light are used to shade the volume dataset with appropriate contrast and cast the shadows to enhance the depth perception. The back light is used to highlight the rim and relatively thin structures, and to illuminate certain overshadowed regions from the back. The role of back light is often ignored in previous work (e.g. [27, 30]) because it is impossible to fully exploit the back light without light absorption and multiple scattering effects. Our lighting design system can utilize the global light absorption, shadow casting, and multiple scattering to enhance the visual perception including depth and shape perception.

One issue of using back light in our lighting design system is that certain areas may be overexposed because the opacity can be unevenly distributed and certain thin structures can be penetrated by the back light and illuminated by all the light sources. Without sacrificing the shading quality, overexposure in certain regions of volume datasets is unavoidable with limited amount of light sources. However, increasing the number of light sources will not only increase the complexity of lighting design process, but also increase the cost of real-time volume rendering. Therefore, we further propose an automatic tone mapping operator to recover details from overexposed regions such as the rim and thin structures. Our tone mapping operator also produces better contrast in the dark areas compared to conventional tone mapping operators.

## 2 RELATED WORK

In recent years, involving advanced shading techniques into the interactive direct volume rendering (DVR) has been received much attention. We refer interested readers to the survey [7]. The screen-space ambient occlusion (SSAO) technique [24] which renders local shadows by sampling nearby depth values in the image-space, was extended to DVR in [3]. The limitation of SSAO is that it can only produce local shadows. To avoid certain drawbacks in SSAO such as limited accuracy and overshadowing, Kronander et al. [11] proposed an efficient visibility encoding technique for object-space occlusion calculation. They first estimate the local visibility at each voxel by sampling nearby voxels and then approximately estimate the global visibility by using the local visibility information and encode the visibility using spherical harmonics. However, as the opacity mapping

• Yubo Zhang is with UC Davis. E-mail: ybzhang@ucdavis.edu.  
• Kwan-Liu Ma is with UC Davis. E-mail: ma@cs.ucdavis.edu.

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.

For information on obtaining reprints of this article, please send e-mail to: [ivcg@computer.org](mailto:ivcg@computer.org).

changes, they need a second to update the visibility volume, which limits the ability of editing the transfer-function in real-time. In order to further reduce the occlusion estimation time, summed area table (SAT) based techniques [3, 21] are developed to efficiently lookup the occlusion information approximately. In certain cases the SAT needs to be rebuilt if the opacity mapping is changed. In [20] light propagation is simulated in the volume-space for shadow and single scattering. Kroes et al. [10] applied Monte Carlo ray tracing to the DVR which can achieve various global illumination effects including shadow and single scattering. However, it still takes several seconds to converge which affects the interactivity and the rendering results are blurred due to the strong noise filtering. In [32] a high quality DVR method is developed which fully simulates global shadow and multiple scattering effects in real-time.

Lighting design is a fundamental topic of computer animation in storytelling and expressing mood [1] and has been studied for decades. In general, lighting design systems can be classified as three types including direct manipulation, indirect light generation and automatic light generation. Direct light manipulation allows the user to fully control the parameters of each light source. It is widely used in many rendering software applications such as Autodesk Maya [2]. But it requires the user to have knowledge in computer graphics and digital arts in order to efficiently setup a good lighting environment. It is also not intuitive compared to the indirect light generation. Indirect light generation systems often let the user to specify certain intuitive parameters which are affected by light sources such as highlight and shadow areas in a scene. For example, in [18, 17] users are allowed to place shadows or highlights directly and the system can indirectly modify the light sources accordingly to match the user input. In [22, 16, 25] users can also paint lights or shadows in the image-space. These systems can solve for the optimal lighting parameters based on the user's paint. Indirect lighting systems provide intuitive interfaces but the light sources cannot be controlled precisely.

Although intuitive, indirect light generation still requires the user possesses domain knowledge in computer graphics. In many cases such as medical visualization, the user can be an expert from other fields without the knowledge of shading models and light behaviors. Therefore, automatic light generation is desired in these situations. Shacked and Lischinski [23] proposed a perceptual quality metric for automatic lighting design. The system searches for optimal lighting parameters based on the perceptual quality metric which can be evaluated from the rendered image. Similar strategies such as maximizing the perceptual entropy or visual information are presented in [4, 28, 30]. These methods try to optimize the lighting parameters based on an objective function. Lee et al. [12] introduced a novel light source placement technique where the objects are segmented into local surface patches and assigned to different light sources. Zupko and El-Nasr [33] developed a system which adapts lighting to the user interaction in real-time by involving perceptual theories and cinematic lighting techniques. Tao et al. [27] developed an automatic lighting design system by measuring the structural differences between images. It takes the structural information into account during the lighting design process. For general evaluations, we refer readers to [9] where the benefits of different types of lighting design interfaces are evaluated through a comprehensive user study.

### 3 OVERVIEW

In general, our lighting system consists of three parts: the automatic light source generation, volume rendering, and exposure optimization. The light source generation module follows the classic three-point lighting setup and computes the light directions and intensities. Note that we only use directional light sources which are easy to manipulate. Our volume rendering module renders the volume dataset under the generated light sources and estimates the global light transportation. The exposure optimization step can recover more details from overexposed areas. Fig. 1 shows the workflow of our volume visualization and how the lighting system fits into the whole process. The design of our automatic lighting system meets the following criteria:

- **High Quality Shading** The generated light sources should pro-

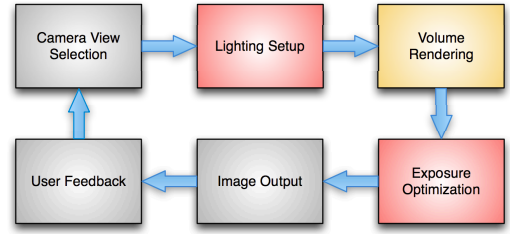


Fig. 1. The workflow of our volume visualization where the colored boxes are implemented in our system including automatic lighting setup, volume rendering with global illumination and exposure optimization. The red boxes are our major focuses which are fully integrated into the volume rendering stage.

duce high quality shading which leads to a good visual perception of depth and occlusion on the volume dataset.

- **Real-time Response** The light source generation should be fast enough in the real-time volume visualization context in order to provide a good user experience.
- **Simple and Intuitive Light Settings** The results should involve only a few light sources for better performance and easy adjustments. The light settings should also be intuitive to the user. The direction and the intensity of the light sources should be relatively stable as the camera view changes.

In the following sections, we first briefly introduce the rendering system and then discuss the light source generation and exposure optimization.

### 4 THE RENDERING SYSTEM

Our rendering system is based on the state of the art techniques [20, 21, 10, 32], where the global light transportation is calculated in the volume-space. For each light source, we compute the light propagation, absorption and multiple scattering on a volume grid. The transportation is modeled using a convection-diffusion partial differential equation and solved numerically. The general model equation has the form

$$\frac{\partial}{\partial t} \rho(\mathbf{x}) = -c\mathbf{u}(\mathbf{x}) \cdot \nabla \rho(\mathbf{x}) - \sigma_a(\mathbf{x})\rho(\mathbf{x}) + \sigma_s \nabla^2 \rho(\mathbf{x}) \quad (1)$$

where  $t$  is the time,  $\rho$  is the energy density,  $c$  is the speed of light,  $\mathbf{u}$  is the local propagation direction of light,  $\sigma_a$  and  $\sigma_s$  are the absorption and scattering coefficients, respectively. For any directional light source, we assume the light direction is  $\mathbf{u}_D$  and the energy density of the directional light is  $\rho_D$ . Assuming that directional lights are used, the following boundary condition is used for Eq. 1

$$\rho(\mathbf{x})|_{\partial\Omega} = \rho_D, \quad \text{if } \mathbf{n}(\mathbf{x}) \cdot \mathbf{u}_D < 0, \quad (2)$$

$$\frac{\partial \rho(\mathbf{x})}{\partial \mathbf{n}} \Big|_{\partial\Omega} = 0, \quad \text{otherwise.} \quad (3)$$

where  $\mathbf{x} \in \partial\Omega$  is the boundary of the entire volume domain  $\Omega$  and  $\mathbf{n}(\mathbf{x})$  is the unit outward normal at  $\mathbf{x}$ . The steady state solution of Eq. 1 represents the final energy distribution which is emitted from a single light source.

Typically, light propagation in the data volume is computed using ray-tracing [8] or volume photon mapping [5]. However, these geometry-based or particle-based methods do not fit the volume data structure perfectly because frequent sampling and gathering operations are needed during the computation. Therefore, the first term on the right hand side of Eq. 1 is used for modeling the light propagation over time  $t$  from a single light source. The term is a convection term which ensures energy conservation and it is the basis of estimating the global illumination effects.

The light absorption is modeled using the second term of the right hand side of Eq. 1. This term simply reduces light energy according to the local absorption coefficients which are computed from the local opacity values. Due to the energy reduction, shadows can occur along the light paths. The global shadow effects can greatly enhance the visual perception of depth and occlusion which is important for accurate volume visualization.

We assume that the volume media is optically thick, so the multiple scattering can be modeled using diffusion [26]. We use a simple isotropic diffusion term as the last term on the right hand side of Eq. 1 which is adequate for modeling realistic multiple scattering for volume visualization. It adds another layer of realism such that we can have a better sense of object thickness if the volume is illuminated from the back. In addition, the back light also highlights the rim of the volumetric object.

Eq. 1 together with the boundary conditions Eq. 2 and Eq. 3 are solved numerically on a volume grid, which is called light volume. The numerical computation can be highly parallelized with the assistance of the GPU. We refer readers to [32] for details. The solution to the equation is then stored into the light volume. A typical volume ray casting algorithm can be performed for the final volume rendering where the global light energy distribution is combined with the local shading at each sampling point along the ray

$$L_{final}(\mathbf{x}) = L(\mathbf{x}) (I_a(\mathbf{x}) + I_d(\mathbf{x}) + I_s(\mathbf{x})) \quad (4)$$

where  $I_a$ ,  $I_d$  and  $I_s$  are the colors of ambient, diffuse and specular lights, respectively.  $L_{final}(\mathbf{x})$  is then used as the final lighting color for material shading.

## 5 LIGHTING DESIGN UNDER GLOBAL ILLUMINATION

The light source generation for volume visualization under global illumination has not been well studied. With global illumination effects, more structural information can be presented if the light sources are placed appropriately. We discuss the placement of each light source in our system, including the key light, the fill light and the back light. Instead of modeling a quality metric and iteratively refine the light parameters, we try to collect the statistical information from the volume dataset and setup the light sources using an empirical lighting design model. This leads to a lighting setup that is close to a manually designed lighting environment and it is unlikely to generate unexpected light configurations in any case. The details are discussed in the following subsections.

### 5.1 Key Light

The key light in the lighting system controls the global shading. Intuitively, a good key light source can produce appropriate contrast on the volume dataset. Instead of iteratively searching the direction which maximizes the contrast measurement in the image-space, we propose a fast and intuitive method for placing the key light, which is stable and closer to human design process. This is done by observing the distribution of surface normals and placing the key light to generate balanced highlight, mid-tone and shadow areas. Fig. 2 shows a good example of placing the key light source. To achieve this, we employ a statistical approach to guide the key light direction which is explained in the following subsections.

#### 5.1.1 Normal Vector Distribution

To study the distribution of the visible surface normals, we have to compute the mean and spatial variation of these normal vectors. We analyze the distribution of the normal vectors in the image-space. This is faster than volume-space computation due to the space and time complexity. For semi-transparent volumes, there can be multiple visible samples that contribute to each pixel. The sample with highest contribution dominates the visual perception based on the adopted rendering model. Here, the contribution refers to the final alpha value that contributes to a pixel from a sample located at position  $\mathbf{x}$ , which can be evaluated through

$$\hat{\alpha}(\mathbf{x}) = \alpha(\mathbf{x}) e^{-\int_{\mathbf{x}}^{\mathbf{x}_c} \alpha(t) dt} \quad (5)$$

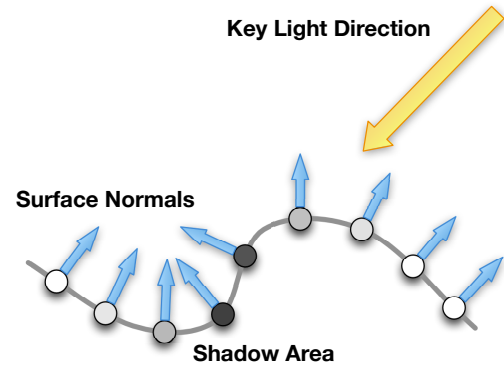


Fig. 2. An example of the key light source placement. The light direction is chosen such that the highlight, mid-tone, and shadow areas are balanced. The gray curve is the surface. The blue arrows are local surface normals. The yellow arrow is the light direction. The color of each circle dot indicates the illumination level.

where  $\hat{\alpha}$  is the final alpha contribution,  $\alpha$  is the local alpha value in the volume space, and  $\mathbf{x}_c$  is the camera position. So for each pixel, we pick the sample along the corresponding ray with the maximum  $\hat{\alpha}$  value. Assume the selected samples have local unit surface normals  $\mathbf{n}_k$  where  $k = 1, 2, \dots, m$  and  $m$  is the number of non-empty pixels. We first compute the mean of these normals  $\mathbf{n}_a = \frac{1}{m} \sum_{k=1}^m \mathbf{n}_k$ . Then the offset of each normal vector from the mean normal vector is computed through  $\mathbf{d}_k = \mathbf{n}_k - \mathbf{n}_a$  where  $\theta_k$  is the angle difference and  $\mathbf{d}_k$  is the rotation axis. Finally, the principal component analysis [6] is performed on the set  $\{\mathbf{d}_k | k = 1, 2, \dots, m\}$  to get the principal vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$  and its corresponding variance  $\{\lambda_1, \lambda_2, \lambda_3\}$ . We use the resulting data as a guide for key light setup.

#### 5.1.2 Key Light Setup

Assume the diffuse shading  $C_d[\mathbf{n} \cdot (-\mathbf{l})]$  is dominant where  $C_d$  is the light color,  $\mathbf{n}$  is the outward unit normal, and  $\mathbf{l}$  is the light direction. It is obvious that we can produce higher shading contrast along the direction of the first principal vector  $\mathbf{v}_1$  because the normal vector distribution has the highest variance  $\lambda_1$  along  $\mathbf{v}_1$ . An extreme case would be the curved surface of a cylinder. Following this intuition, we define two characteristic normal vectors  $\mathbf{n}_+ = \mathbf{n}_a + \sqrt{\lambda_1} \mathbf{v}_1$  and  $\mathbf{n}_- = \mathbf{n}_a - \sqrt{\lambda_1} \mathbf{v}_1$ . If we set the key light direction to  $-\mathbf{n}_+$ , then the areas with normal vectors close to  $\mathbf{n}_+$  will be highlighted and the areas with normal vectors close  $\mathbf{n}_-$  will be shadowed unless the visible areas are relatively flat. Setting key light direction to either  $-\mathbf{n}_+$  or  $-\mathbf{n}_-$  can produce good shading contrast where the highlight, mid-tone and shadow areas are balanced. According to the study [15], we choose the one from top where the y-component of the direction vector of key light is negative. The other direction is then used for fill light which is discussed in the next section. We set the intensity of the key light  $I_k = 1$ .

### 5.2 Fill Light

The fill light is used to illuminate the shadow areas in order to make the details in the shadow areas visible. Following the previous section, the direction of the fill light is already determined, which can be either  $-\mathbf{n}_-$  or  $-\mathbf{n}_+$  according to the selection of the key light. The y-component of the direction vector of fill light is non-negative. In addition to the direction, we also have to choose an appropriate intensity for the fill light. If the variance of the normal vector distribution is low, which means the visible areas are relatively flat, then there is no over-shadowed areas and the fill light is even not necessary. On the other hand, we need the fill light if the variance  $\lambda_1$  is high because there would be more over-shadowed areas. Therefore, we set the fill light intensity based on the variance  $I_f = c_f \sqrt{\lambda_1}$  where  $c_f < I_k$  is a scaling



constant for the fill light intensity. In our system  $c_f$  is a configurable parameter. Fig. 3 shows two typical examples with different variance values of the normal vector distribution. It gives the intuition of how the intensity of the fill light is set based on the variance  $\lambda_1$ .

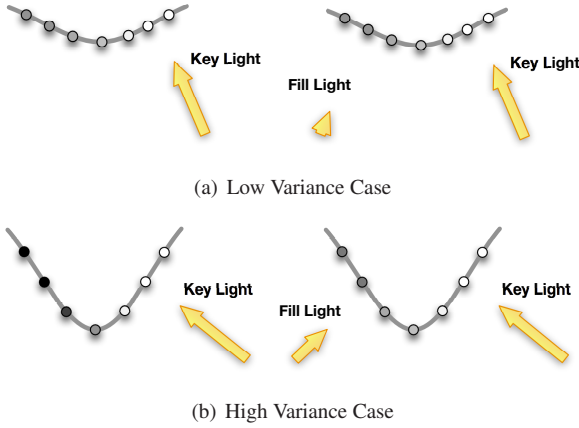


Fig. 3. (a) A typical case where the surface is relatively flat and the variance  $\lambda_1$  is low. In this case, the intensity of the fill light should be low in order to maintain the contrast produced by the key light. (b) Another typical case where the variance  $\lambda_1$  is close to  $1/2$  and half of the surface is completely shadowed. In this case the fill light can have a relatively high intensity value.

### 5.3 Back Light

The back light has an important role in the lighting design. Its capability can be fully exploited by global illumination including light absorption and multiple scattering. In photography, the back light is often used for highlighting the rim of the object or character. It can also be applied to the volume rendering and an example is shown in Fig. 4. We can see that the back light highlighted the backside of the

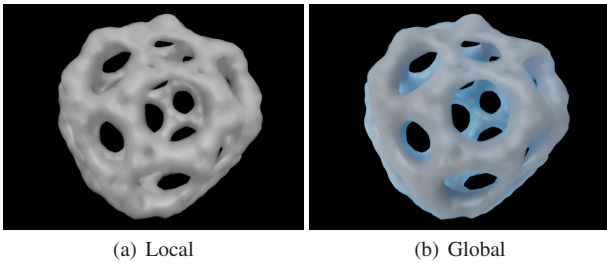


Fig. 4. Comparison of the back light effects under local and global illumination. In the global case (b), the backside of the object is highlighted in blue, which provides additional depth cue.

object and improves the depth cue. In our system, the direction of the back light is simply set to the opposite direction of the camera view. The tricky part is to estimate an appropriate intensity value for the back light. While a lower intensity value can make the back light fully occluded by the object, a higher intensity value can also penetrate the object and make it over-illuminated. Therefore, we have to measure the opacity distribution of the volume dataset.

#### 5.3.1 Accumulated Opacity Distribution

Similar to estimating the final alpha contribution in Eq. 5, we can also estimate the accumulated opacity  $A_k$  along each camera ray direction  $\mathbf{d}_k$  in the image-space

$$A_k = \int_{\mathbf{x}_c}^{\mathbf{x}_c + s_{\max} \mathbf{d}_k} \alpha(\mathbf{t}) d\mathbf{t} \quad (6)$$

where  $\mathbf{x}_c$  is the camera position,  $s_{\max}$  is the maximum ray distance,  $\alpha(\mathbf{t})$  is the opacity at position  $\mathbf{t}$ ,  $k = 1, 2, \dots, m$ , and  $m$  is the number of non-empty pixels. Then we can compute the mean  $A_a$  and the variance  $A_v$  from the set  $\{A_k | k = 1, 2, \dots, m\}$ . With the opacity distribution, we can estimate the intensity of the back light at the reasonable scale.

#### 5.3.2 Back Light Setup

Assume that the back light can penetrate most part of the volume dataset, the following condition must hold

$$I_b e^{-A_a - \sqrt{A_v}} \geq I_e \quad (7)$$

where  $I_b$  is the intensity of the back light,  $A_a$  is the mean accumulated opacity,  $A_v$  is the variance of accumulated opacity, and  $I_e$  is the lowest perceivable light intensity. The intuition behind Inequality 7 is that most areas have accumulated opacity values less than  $A_a + \sqrt{A_v}$  by the definition of variance. Therefore, it is straightforward to find the lowest back light intensity that satisfies Inequality 7:

$$I_b = I_e e^{A_a + \sqrt{A_v}}. \quad (8)$$

$I_e \ll 1$  is a configurable variable in our lighting system. Note that  $I_e$  is a data-independent value. Once set, the system can automatically compute the appropriate back light intensity for each case. This is much more convenient than adjusting the back light intensity manually.

#### 5.3.3 Color of the Back Light

In addition to the direction and intensity of the back light, the color of the back light can also be utilized. Using colors to illuminate shadow areas can improve the visual perception [29]. In certain cases, if we set the back light source to a different color than the key light, we can have a better depth perception on the volume dataset since the rear part of the data will be illuminated in this color. It can also be used to evaluate the thickness of local structures because thin structures will be penetrated by the back light and the color will look different than thick structures. The use of the back light is evaluated in Section 8.

In certain extreme cases, The generated key and fill light directions can be quite different from the camera direction if the surface variance is big. Hence the light path between the camera focus point and the key light or even the fill light can be occluded. However, the structures around the focus point would not be overshadowed due to our strong back light penetration and scattering. They can be even highlighted by the back light color. We show a good example in Fig. 12.

## 6 IMAGE-SPACE OPTIMIZATION

With the approach discussed in Section 5, we can already generate an appropriate three-point lighting environment for a volume dataset. However, certain areas in the image can be overexposed. This often happens when there are thin structures in the volume dataset which can be over-illuminated by the strong back light together with the key light. To avoid this, we automatically generate a special tone curve to optimize the exposure in highlight and shadow regions. Tone mapping is typically used in digital photo processing (e.g. [19]) and is also applied to HDR rendering (e.g. [31]). In our system, the goal is to generate a tone curve that can compress the highlight while maintain the contrast in the shadow area. In [19], a simple tone mapping operator was discussed which maps an average luminance value to 0.5 and compresses the highlight region:

$$L' = \frac{L}{L + M} \quad (9)$$

where  $L$  is the original luminance value,  $L'$  is the mapped luminance value, and  $M$  is the log-average luminance. It is obvious that Eq. 9 can map the value  $M$  to 0.5 and map the infinity to 1 which compresses the overexposed areas. However, Eq. 9 also reduces the contrast in the



shadow area where  $L < M$ . To remedy this, we introduce a modified tone mapping operator

$$L' = \frac{tL}{L+M} + \frac{(1-t)L^2}{2M^2} \quad (10)$$

where  $t = \min\{L/M, 1\}$ . Here the term  $\frac{L^2}{2M^2}$  is used to control the contrast in the shadow areas and it is blended into the original tone mapping operator. The tone curves of Eq. 9 and Eq. 10 are plotted in Fig. 5. We can see that the contrast of the shadow area is enhanced in

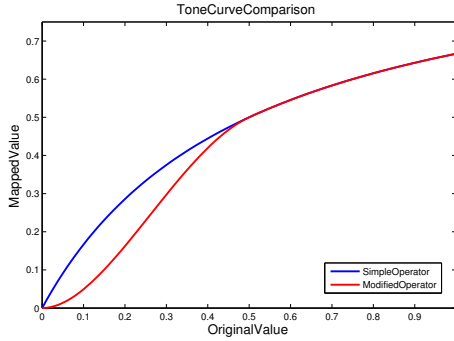


Fig. 5. Comparison between the simple tone mapping operator and our modified operator. Here we set  $M = 0.5$  as an example. The blue line is the simple operator and the red line is our modified operator. The red curve can enhance the contrast in the shadow area where  $L < M$  while maintain some of the properties of the simple operator including the mid-point mapping and highlight compressing.

our modified tone curve. We discuss the implementation and results in the following sections.

## 7 IMPLEMENTATION

The implementation of our system is based on the OpenGL library and NVIDIA CUDA [14]. We also use G3D [13] as the application framework and user interface. The subsystems include lighting setup, volume rendering and exposure optimization and the detailed implementations are discussed in the following subsections. We summarize our method in Algorithm 1.

### Algorithm 1 Lighting Setup for Volume Visualization under Global Illumination

- 1: Estimate normals and opacities in image-space
- 2: Compute PCA on normal vectors
- 3: Setup key light
- 4: Setup fill light
- 5: Compute mean and variance of opacities
- 6: Setup back light
- 7: Simulate global light transportation
- 8: Render volume into HDR frame buffer
- 9: Apply tone mapping operator
- 10: Output image

In the automatic light source generation stage, we first use a standard volume ray casting to render the whole volume into a float-point frame buffer. Instead of writing colors into the frame buffer, we record the normal vectors and accumulated opacity values in the float-point RGBA channels. These values can be estimated during the volume ray casting. Then the frame buffer values are packed into an array for PCA and mean/variance computation where pixels with zero alpha values are discarded. Here the PCA computation is performed on a separate CPU thread. The results are then used for setting the direction and intensity of the light sources. The light source parameters are finally passed to the volume renderer.

The volume renderer creates two 3D textures for storing the original dataset and the light volume. It also creates several GPU arrays for light simulation. In the lighting simulation step, Eq. 1 is firstly solved using CUDA acceleration for each light source. Then the result is written into the light volume. Finally, the light volume is combined with the standard volume ray casting and the image is rendered into an HDR frame buffer for tone mapping.

In the exposure optimization stage, we first compute the log-average luminance from the HDR frame buffer through parallel reduction. Then Eq. 10 is applied to all the pixels in the frame buffer and the final image is obtained. The tone mapping is implemented in a GLSL shader.

## 8 RESULTS AND EVALUATION

In this section, we present various results produced to validate and evaluate our lighting system. The evaluation includes basic functional evaluation and real case studies.

### 8.1 Exposure Optimization

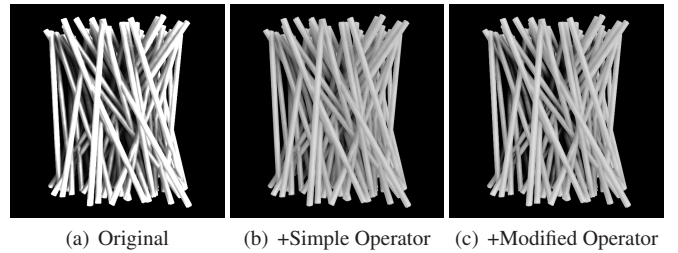


Fig. 6. Evaluation of our tone mapping operator. (a) Rendering without tone mapping; (b) Rendering with simple tone mapping operator; (c) Rendering with our modified operator. Note that our operator produces better contrast in the darker area.

We first validated our modified tone mapping operator discussed in Section 6. A volume dataset with randomly distributed tubes is used. We set up a directional light with strong intensity and made the light source directly face to the volume. Then we rendered the volume dataset and applied tone mapping operators Eq. 9 and Eq. 10. We compared the rendering results with and without applying tone mapping operators in Fig. 6(a)-(c). We found that the simple operator in Eq. 9 is effective in compressing the highlights and our modified operator Eq. 10 can further improve the contrast in the dark areas.

### 8.2 Depth Perception

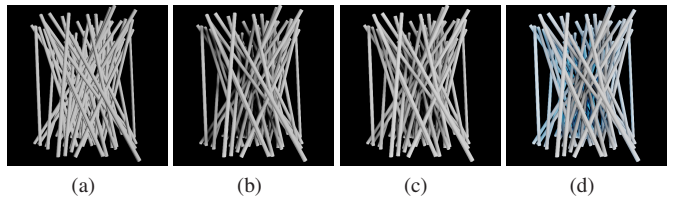


Fig. 7. Comparison between local illumination (a) and global illumination (b)-(d) under the automatically generated light sources by our lighting system. Here we also split the contribution of these light sources into (b) key light, (c) key+fill lights and (d) key+fill+back lights. We can clearly see how these light sources can enhance the depth perception of the volume dataset under global illumination.

We also evaluated how our automatically generated light sources can improve the presentation of structural information. In this test, a dataset with cluttered tubes are used which has rich structural information. From this kind of data, we usually get poor depth perception under local illumination and we try to find out if our lighting system helps. In Fig. 7, we compare the rendering results with different light

settings. It is shown that the distribution of the tubes looks flat in Fig. 7(a) where the volume is rendered using conventional volume renderer with local illumination. Fig. 7(b)-7(d) show the rendering results using our lighting system. To evaluate the contribution of each light source, we split the lighting into three settings. It is clearly shown that with only the key light which comes from the front-right side, the depth perception can already be improved greatly. The fill light from the front-left side then illuminates certain overshadowed areas. Finally, the blue colored back light further improves the shading such that we can realize the depth of the tubes from the shading color. The overshadowed deep areas are also illuminated in blue by the back light. From this experiment, we conclude that our lighting system is effective for visualizing volume datasets with cluttered spatial structures and can greatly improve the depth perception on the datasets.

### 8.3 Light Direction Evaluation

To evaluate the automatically generated key light and fill light directions, we show a set of examples in Fig. 8 where two datasets are rendered with different camera positions and transfer functions. It is shown that the key light directions always deviate from the view directions, which leads to better shading and contrast. It is close to the use of key lights in photography which can produce images with better 3D-look than using headlights. The fill light effects are subtle and cannot be ignored in most cases. While in Fig. 8(g)-(i) the fill light effect is obvious, it can also illuminate most overshadowed small structures in certain cases. Fig. 9 compared our automatically generated light settings to the headlight and randomly generated lights which deviate from the headlight direction. The flow dataset used has higher surface normal variance in the vertical direction. Our algorithm can capture this structural feature and generate an appropriate key light direction which produces good shading contrast and cast shadows vertically, as shown in 9(a). Using headlight, however, tends to produce a flat shading, as shown in 9(b), and the global illumination effect is weakened. We also generate a number of random light directions within 90 degrees from the camera direction. Here 9(c) and 9(d) are two representative cases. The specular lighting effects are missing in 9(c) due to a large deviation of the key and fill light directions. The depth perception in 9(d) is better than 9(b) and 9(c) but not as good as 9(a).

### 8.4 Frame-to-frame Coherence

We also studied the frame-to-frame coherence of our automatically generated light directions by moving the camera around the volume dataset. We used a separate thread to calculate the PCA and continuously update the light directions. Fig. 10 shows several frames captured from a camera moving test. These frames are chosen where the change of light directions are more obvious than other frames. The whole animation is provided in the supplementary video. In general, the directions of light sources change smoothly according to the continuous camera motion as the distribution of surface normals remains stable. The only exception is that the key light and the fill light can swap when they have similar y-component. This is caused by the strategy of choosing the key light described in Section 5.1.2. We also tested the frame-to-frame coherence when changing the transfer functions. Fig. 11 shows several examples where the light sources are generated based on different transfer functions. It is shown that changing transfer functions may affect the light directions.

### 8.5 Case Studies

We applied our lighting system to several datasets including a scanned mechanical part, a vortex field from a combustion simulation, and a CT scan of a patient's hand who has orthopedic disease.

#### 8.5.1 Mechanical Part

Our first case is a scanned mechanical part dataset shown in Fig. 12. The mechanical part has some nested structure around the center. In Fig. 12(e), the dataset is rendered using local shading under our automatically generated light sources. Although all the details can be clearly seen, we cannot determine the depth order of the structures

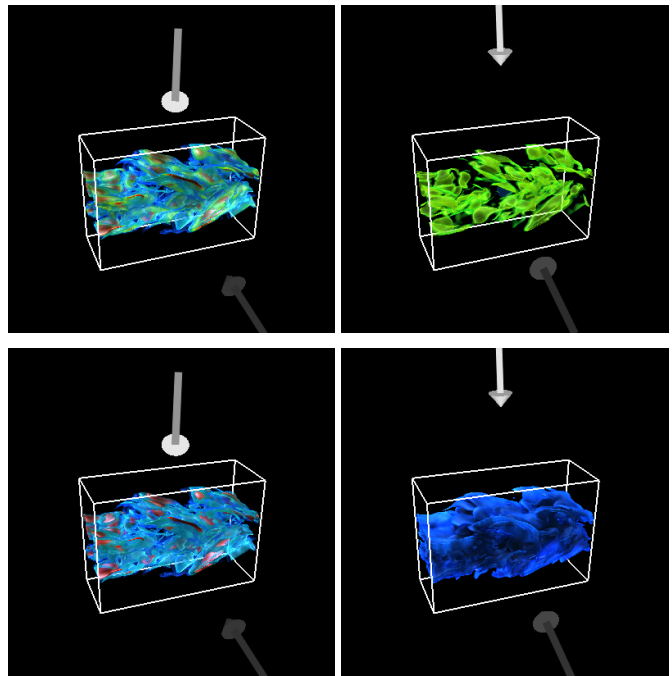


Fig. 11. An example of frame-to-frame coherence test by changing the transfer function. Light sources are updated if the opacity mapping changes. Here we use the camera view for rendering.

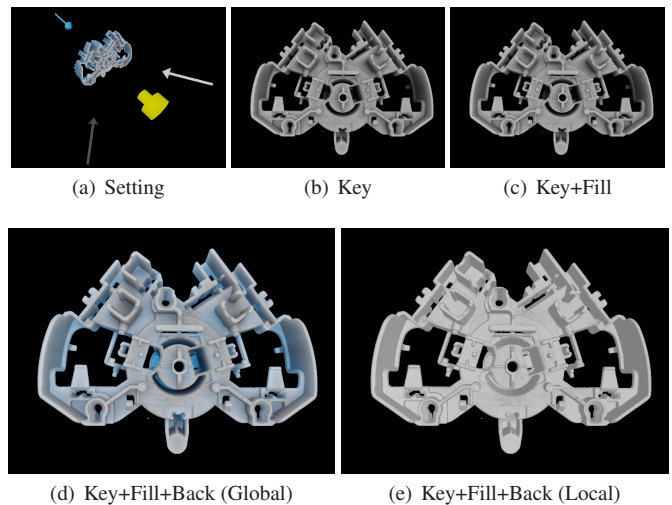


Fig. 12. Comparing the local shading and the global shading using the same light settings generated using our algorithm for a scanned mechanical part. Note the center area in (e), where local shading can give a wrong impression of the depth order. The blue back light together with light absorption and multiple scattering effects can enhance the depth perception. Note that our algorithm can compute an appropriate intensity which makes the back light penetrates the volume object while maintain a good contrast.

around the center since the back light effect is not visible in local illumination. It seems that the concave regions are closer to the camera and the surrounding vertical surfaces look flat. By using global shading under the same light setting, the concave regions are correctly shaded. In Fig. 12(d), these regions are shadowed by the key light and are highlighted by the back light. The surrounding vertical surfaces also have a smooth transition from white to blue, which indicates the depth change. This case also shows the importance of the back light under global illumination. Without the back light, the concave regions

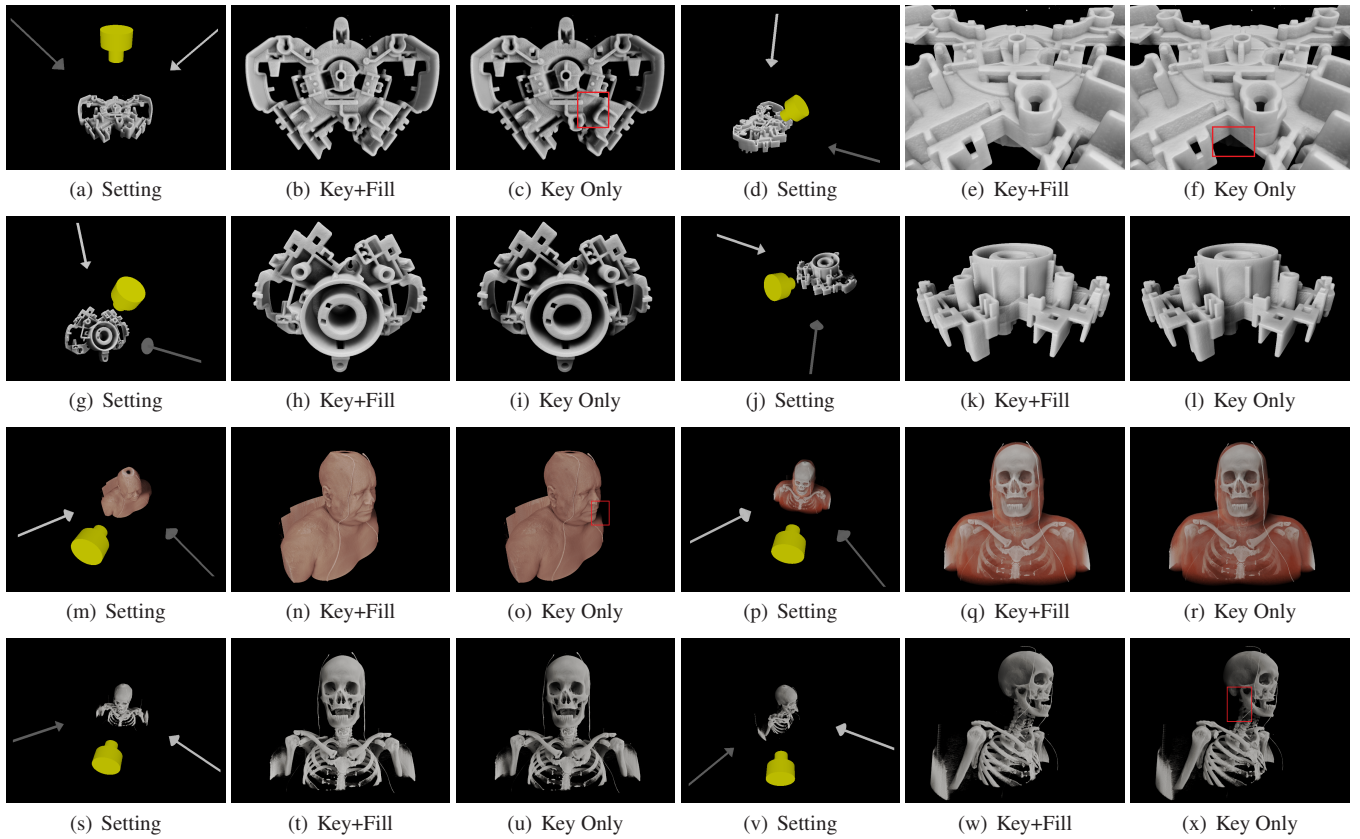


Fig. 8. Evaluation of automatically generated key light and fill light positions based on our algorithm. Key lights are rendered in white and fill lights are rendered in gray. We also provide the results where fill lights are ignored. In certain cases there are overshadowed areas marked by red boxes. (g)-(i) shows an example where the fill light effects are strong.

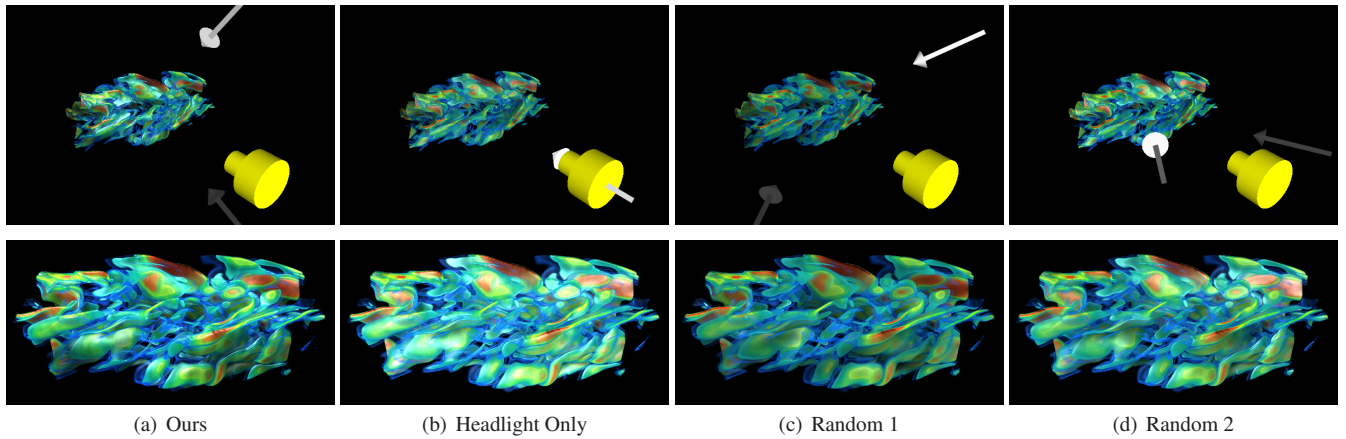


Fig. 9. Comparison among automatically generated light directions (a), headlight (b) and randomly generated light directions (c)&(d). The top row shows the light settings and the bottom row shows the corresponding rendering results.

can be overshadowed, as shown in Fig. 12(b) and 12(c) where both the key light and the fill light cannot cover the overshadowed regions.

### 8.5.2 Vortex Field

Our second case is a turbulent vortex field which comes from a combustion simulation. The tangled vortex structures are difficult to visualize using local shading, as shown in Fig. 13(a). It seems all the structures are in a flat plane and the fine scale details are not clearly presented. In Fig. 13(b), we use global illumination to render the vortex field. As a result, the depths of all different vortex layers are clearly visualized. The key light is placed to illuminate the top-front

of the dataset, which is similar to Fig. 9(a). The shadows are cast from the upper vortex layers to the lower vortex layers which make the structural information well presented.

### 8.5.3 Patient Hand

Our third case is a CT scan of a patient's left hand. The patient has orthopedic disease and there are several holes at the joints due to the erosion. The goal is to observe these holes and analyze the situation of the recovery process. In Fig. 14(e), the dataset is visualized using local shading under our automatically generated light sources. We use red boxes to mark certain areas which were ambiguous to the experts. In



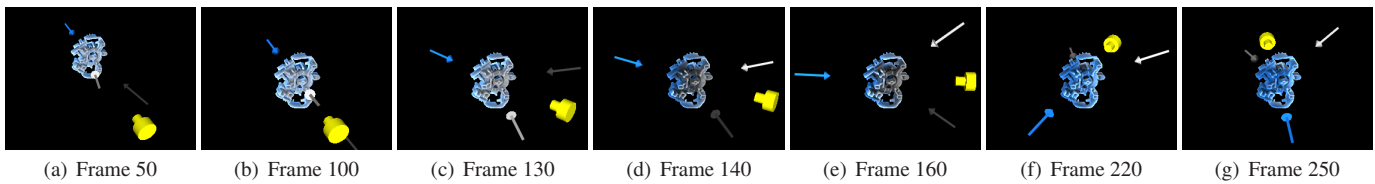


Fig. 10. An example of frame-to-frame coherence test by moving the camera around the volume dataset. Light sources are continuously updated using our system. Here the back light is shown in blue. These frames are chosen where the change of light directions are more obvious than other frames. Note the swap between key light and fill light after frame 130 which is caused by the strategy of choosing the key light described in Section 5.1.2.

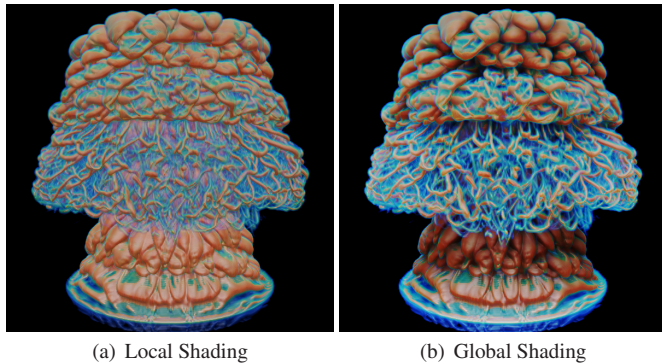


Fig. 13. (a) Vortex field visualized using local shading; (b) Vortex field visualized using global shading. The automatically generated light sources are similar to Fig. 9(a) with an additional back light source in white. Note that the shadows in (b) make the structural information better presented.

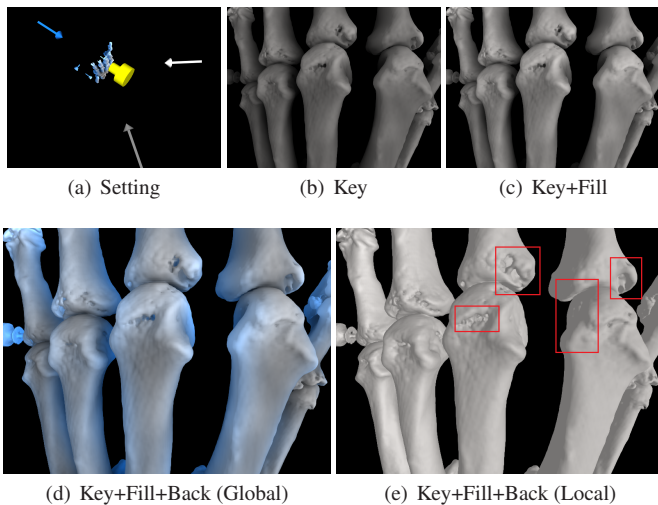


Fig. 14. Comparing the local shading and the global shading using our lighting design system for a CT scan of a patient's hand. Note the red boxes in (e), where local shading can give a wrong impression of the depth order. These structures are correctly visualized in (d) with the assistance of light absorption, multiple scattering and an appropriate back light intensity.

Fig. 14(e), they could only identify rapid changes in surface topology, but were not able to tell if these corresponded to true erosions. The local shading does not always convey the appropriate depth information, which makes interpretation sub-optimal. In Fig. 14(d), with the same light sources, the expert users concluded that they could better identify the true erosions and their extent. The more realistic portrayal of the erosions seems to provide adequate level of detail and allows

the experts to focus on the relevant parts of the bone. The blue colored back light also highlights the deep holes and the rim of the bones which are overshadowed in Fig. 14(c). Based on the evaluation from expert users, we conclude that our lighting design system is effective for certain cases in medical visualization.

#### 8.5.4 Supernova

We also evaluated our lighting system for a supernova dataset with different camera views and transfer-functions, which is shown in Fig. 15. For each camera and transfer-function setting, our system can generate optimized lighting that achieves good depth cue. Fig. 15(a) is a global view of the supernova. Fig. 15(b) is a close up view which visualizes the outer core. In 15(c) we removed the outer core to reveal the internal structures. In 15(d), we rendered the dataset from a different view. All four images have different transfer functions and the illuminations from our generated lights provide good visual perceptions.

#### 8.5.5 Additional Results

Some additional results are shown in Fig. 16 and Fig. 17. In Fig. 16, a scanned upper human body is visualized using different camera views. Our lighting system can generate appropriate view-dependent light sources based on each camera view. In Fig. 17, we show another vortex field rendered using our automatic lighting. We also reduced the opacity of outer layers. We can see that all the features are well shaded. The contrast makes these features distinguishable to the viewer.

#### 8.6 Performance

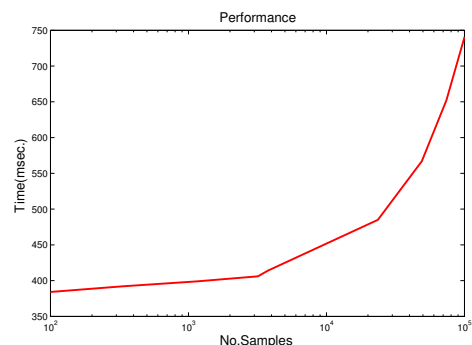


Fig. 18. The performance of our light source generation mainly depends on the number of samples in the image-space that we used for estimating normal vector distribution and accumulated opacity distribution. Note that the x-axis is in log scale.

We measured the time needed for light source generation on a laptop with Intel Quad-core 1.7GHz CPU and NVIDIA GeForce GTX460M GPU, which is plotted in Fig. 18. The computation time is related to the number of samples we used for analyzing normal vector distribution and accumulated opacity distribution. The number of samples highly depends on the rendering resolution. The main steps include the sample collection which needs a volume ray casting operation and the PCA computation. In our experiments, the average

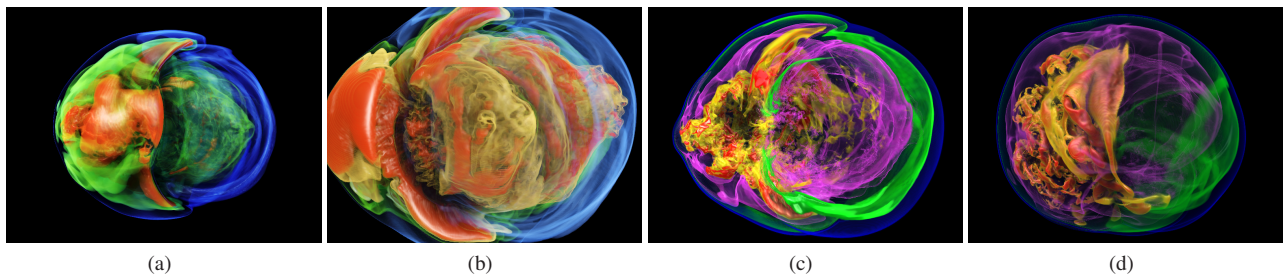


Fig. 15. Visualization of a supernova dataset using different camera views and transfer-functions. Our system automatically generates optimized light sources for each camera and transfer-function combination where the global structures and the local features are well shaded.

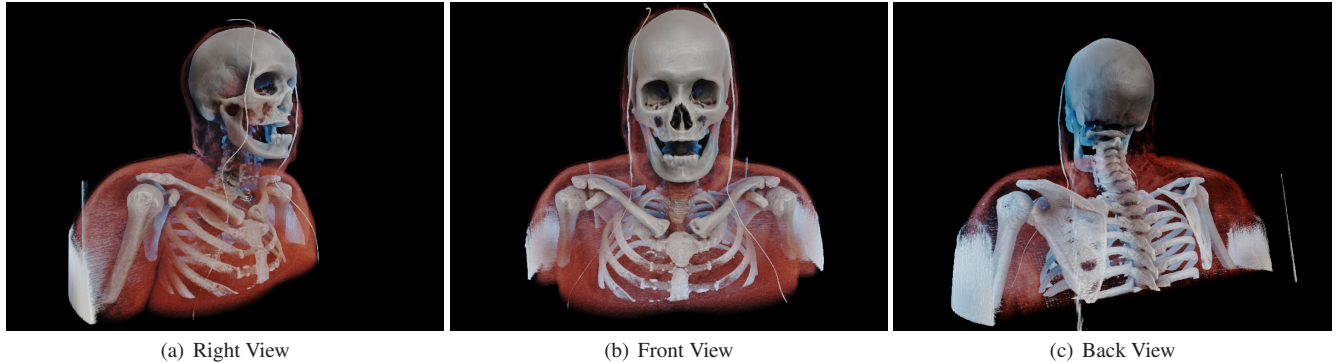


Fig. 16. Visualization of a scanned human body using different camera views. Our lighting system can generate appropriate light sources for each view. Here a blue back light is used.

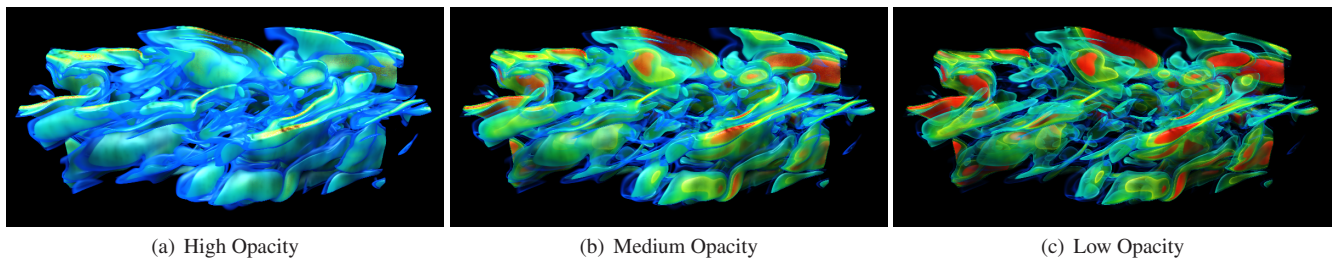


Fig. 17. Another visualization of a vortex field from combustion simulation. The global opacity is adjusted to reveal the internal structures. Good contrast is achieved for both the large and the small scale features.

time taken for light source generation is less than 500ms. Since we use a separate CPU thread for the PCA computation, it is fast enough for interactive rendering systems. The lights are updated every 10-20 frames approximately while the rendering frame rate is above 30 fps. Currently the PCA computation is implemented on CPU which takes up 95 percent of the total computation time. We believe a full GPU implementation can further increase the performance.

## 9 CONCLUSION

In this paper, we present a novel automatic lighting design system for volume visualization under global illumination. It takes global light transportation into account during the lighting design process. Our system can automatically generate the key light, the fill light and the back light for a volume dataset by analyzing the structural information statistically and can fully utilize the back light for volume visualization. We verified our technique through various experiments and evaluated our lighting system through several case studies. We conclude that our lighting design method for globally illuminated volume visualization is effective for volume datasets with complex and fine structures. The performance of our method also makes it practical for interactive systems which should provide instant feedback to the user. Most importantly, we free the users of our system from worry-

ing about setting up lighting parameters so they can focus on locating features of interest by picking good views and transfer functions for property mapping. However, our method also has limitations. First, it may not be applied to view-independent cases because the result is optimized for the camera view. Second, it is not applicable in cases where more than three light sources must be used. In addition, our method may not give optimal results for extremely complex datasets where point light sources are needed to illuminate internal structures that directional lights cannot reach. It does not take depth information into account.

In the future, we will extend our lighting design method to time-varying volume datasets, which are commonly found in scientific applications. We will also improve our system by supporting more than three light sources, including point light sources which can be used to highlight certain features in the volume.

## ACKNOWLEDGMENTS

This research has been sponsored in part by the National Science Foundation through grants OCI-0905008, OCI-0850566, OCI-0749227, and CCF-0811422, and also the Department of Energy through grants DEFC02-06ER25777, DE-CS0005334, and DE-FC02-12ER26072.

## REFERENCES

- [1] A. A. Apodaca and L. Gritz. *Advanced RenderMan: Creating CGI for Motion Picture*. Morgan Kaufmann Publishers Inc., 1st edition, 1999.
- [2] Autodesk. Maya 2013.
- [3] J. Díaz, P. Vázquez, I. Navazo, and F. Duguet. Real-time ambient occlusion and halos with summed area tables. *Computers & Graphics*, 34(4):337–350, 2010.
- [4] S. Gumhold. Maximum entropy light source placement. In *Proc. of IEEE Visualization*, pages 275–282, 2002.
- [5] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. of SIGGRAPH*, pages 311–320. ACM, 1998.
- [6] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag New York, 1986.
- [7] D. Jönsson, E. Sundén, A. Ynnerman, and T. Ropinski. Interactive Volume Rendering with Volumetric Illumination. In *Eurographics STAR program*, 2012.
- [8] J. T. Kajiya and B. P. Von Herzen. Ray tracing volume densities. *SIGGRAPH Comput. Graph.*, 18:165–174, 1984.
- [9] W. B. Kerr and F. Pellacini. Toward evaluating lighting design interface paradigms for novice users. *ACM Trans. Graph.*, 28(3):26:1–26:9, July 2009.
- [10] T. Kroes, F. H. Post, and C. P. Botha. Exposure render: An interactive photo-realistic volume rendering framework. *PLoS ONE*, 7(7):e38586, 07 2012.
- [11] J. Kronander, D. Jonsson, J. Low, P. Ljung, A. Ynnerman, and J. Unger. Efficient visibility encoding for dynamic illumination in direct volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 18(3):447–462, march 2012.
- [12] C. H. Lee, X. Hao, and A. Varshney. Light collages: Lighting design for effective visualization. In *Proceedings of IEEE Visualization*, pages 281–288, 2004.
- [13] M. McGuire. G3D innovation engine. <http://g3d.sourceforge.net/>.
- [14] NVIDIA. CUDA C Programming Guide, 2012. <http://docs.nvidia.com/>.
- [15] J. P. O'Shea, M. S. Banks, and M. Agrawala. The assumed light direction for perceiving shape from shading. In *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, APGV '08, pages 135–142, 2008.
- [16] F. Pellacini, F. Battaglia, R. K. Morley, and A. Finkelstein. Lighting with paint. *ACM Trans. Graph.*, 26(2), June 2007.
- [17] F. Pellacini, P. Tole, and D. P. Greenberg. A user interface for interactive cinematic shadow design. *ACM Trans. Graph.*, 21(3):563–566, July 2002.
- [18] P. Poulin and A. Fournier. Lights from highlights and shadows. In *Proc. of I3D*, pages 31–38, 1992.
- [19] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. Graph.*, 21(3):267–276, July 2002.
- [20] T. Ropinski, C. Döring, and C. Rezk-Salama. Interactive volumetric lighting simulating scattering and shadowing. In *PacificVis*, pages 169–176. IEEE, 2010.
- [21] P. Schlegel, M. Makhinya, and R. Pajarola. Extinction-based shading and illumination in gpu volume ray-casting. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1795–1802, Dec. 2011.
- [22] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, and D. Greenberg. Painting with light. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 143–146, 1993.
- [23] R. Shacked and D. Lischinski. Automatic lighting design using a perceptual quality metric. *Comput. Graph. Forum*, 20(3):215–227, 2001.
- [24] P. Shanmugam and O. Arikan. Hardware accelerated ambient occlusion techniques on gpus. In *Proc. of I3D*, pages 73–80, 2007.
- [25] A. Shesh and B. Chen. Crayon lighting: sketch-guided illumination of models. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 95–102, 2007.
- [26] J. Stam. Multiple scattering as a diffusion process. In *Proc. of EGWR*, pages 41–50, 1995.
- [27] Y. Tao, H. Lin, F. Dong, C. Wang, G. Clapworthy, and H. Bao. Structure-aware lighting design for volume visualization. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2372–2381, 2012.
- [28] P.-P. Vázquez. Automatic light source placement for maximum visual information recovery. *Comput. Graph. Forum*, 26(2):143–156, 2007.
- [29] V. Šoltészová, D. Patel, and I. Viola. Chromatic shadows for improved perception. In *Proc. of NPAR*, pages 105–116, 2011.
- [30] L. Wang and A. E. Kaufman. Lighting system for visual perception enhancement in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(1):67–80, Jan. 2013.
- [31] X. Yuan, M. X. Nguyen, B. Chen, and D. H. Porter. High dynamic range volume visualization. In *Proc. of IEEE Visualization*, 2005.
- [32] Y. Zhang and K.-L. Ma. Fast global illumination for interactive volume visualization. In *Proc. of I3D*, 2013.
- [33] J. Zupko and M. S. El-Nasr. A tool for adaptive lighting design. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, Sandbox '08, pages 135–142.