

A Versatile Optical Model for Hybrid Rendering of Volume Data

Fei Yang, Qingde Li, Dehui Xiang, Yong Cao, and Jie Tian, *Fellow, IEEE*

Abstract—In volume rendering, most optical models currently in use are based on the assumptions that a volumetric object is a collection of particles and that the macro behavior of particles, when they interact with light rays, can be predicted based on the behavior of each individual particle. However, such models are not capable of characterizing the collective optical effect of a collection of particles which dominates the appearance of the boundaries of dense objects. In this paper, we propose a generalized optical model that combines particle elements and surface elements together to characterize both the behavior of individual particles and the collective effect of particles. The framework based on a new model provides a more powerful and flexible tool for hybrid rendering of isosurfaces and transparent clouds of particles in a single scene. It also provides a more rational basis for shading, so the problem of normal-based shading in homogeneous regions encountered in conventional volume rendering can be easily avoided. The model can be seen as an extension to the classical model. It can be implemented easily, and most of the advanced numerical estimation methods previously developed specifically for the particle-based optical model, such as preintegration, can be applied to the new model to achieve high-quality rendering results.

Index Terms—Direct volume rendering, optical models, isosurfaces, preintegration, ray casting, transfer function.

1 INTRODUCTION

UNLIKE TRADITIONAL polygon-based rendering, there is generally no such thing as geometric primitives in direct volume rendering. Instead, objects contained in a volume data set are usually considered as collections of particles that absorb, transmit, and emit light. The series of optical models proposed by Max [1] are based on this concept. It can be clearly seen that these models tend to assume that light-object interaction can be reduced to light's interaction with individual particles, and the macro optical behavior can be predicted based on the density distribution of the particles. For materials formed by loosely coupled particles such as dust or mist, these assumptions are obviously reasonable.

However, the particle-based optical models are limited when describing light-object interaction, especially around the boundary of a dense object. For a dense object, there is a strong collective effect between particles of the same type. In this case, light's interaction with the object behaves more like a direct interaction with the boundary surfaces rather than with the individual particles (see Fig. 1). In reality, this is the foundation of the most frequently used optical models in traditional surface-based graphics. Most

of the traditional shading models, such as Blinn-Phong shading, were developed to simulate the light-surface interaction, which cannot be used readily to model the light-particle interaction.

Motivated by these observations, we propose a hybrid model that combines particle elements and surface elements together to characterize both the behaviors of individual particles and the collective effect of particles. For a scalar volume, the scalar value is used for two purposes. First, by using a transfer function, it is mapped to the density of the particles. Second, as a field, it defines a set of isosurfaces, which are used as ideal geometric shapes to capture the collective effect of the particles. Particles and surfaces behave differently both in the way light is propagated to the viewer and in the way it is attenuated along its direction. For the particles, light is randomly scattered to the viewer, and the attenuation is related to the density of the particles and the distance a light ray travels. For the surfaces, light may be reflected and refracted, and the attenuation depends on the opacities of the surfaces and the number of surfaces a light ray passes. For a sufficiently short ray segment, the total light propagated to the viewer can be flexibly modeled as a linear combination of the amount of light from the particles and that from the surfaces.

Based on the hybrid model, a unified rendering framework is developed. It benefits both the integration of the surface elements and local illumination, which are two important aspects of volume rendering. On one hand, it provides a more flexible tool for hybrid rendering of isosurfaces and transparent clouds of particles in a single scene. The display of isosurfaces can be controlled in a continuous way using transfer functions rather than having them discretely placed at finite isovalues, which is capable of generating soft boundaries that have less image space aliasing problems. On the other hand, the new model provides a more rational basis for local illumination. In

• F. Yang, D. Xiang, and J. Tian are with the Institute of Automation Chinese Academy of Sciences, 95 Zhongguancun East Road, Beijing 100190, China.

E-mail: yangfei09@mails.gucas.ac.cn, dehui.xiang@ia.ac.cn, tian@ieee.org.

• Q. Li is with the Department of Computer Science, University of Hull, Cottingham Road, Hull, HU1 7RX, United Kingdom.

E-mail: Q.Li@hull.ac.uk.

• Y. Cao is with the Virginia Polytechnic Institute and State University, 2202 Kraft Drive, Blacksburg, VA 24060. E-mail: yongcao@vt.edu.

Manuscript received 30 Aug. 2010; revised 20 Apr. 2011; accepted 1 June 2011; published online 16 June 2011.

Recommended for acceptance by H.-W. Shen.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number TVCG-2010-08-0202. Digital Object Identifier no. 10.1109/TVCG.2011.113.

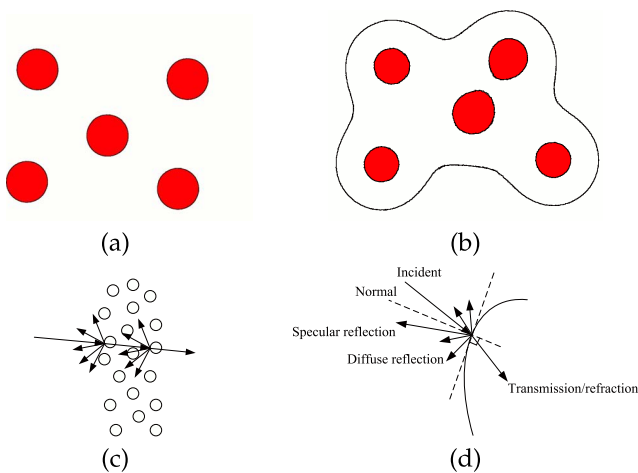


Fig. 1. Particle and surface. (a) Illustration of isolated particles. (b) Illustration of the formation of a surface from the collective effect of particles. (c) Illustration of light-particle interaction. (d) Illustration of light-surface interaction.

conventional volume rendering, one often encounters difficulty in illuminating homogeneous regions, a problem usually said to be due to a lack of a “well-defined” normal vector in homogeneous regions. However, this problem can be easily avoided using the proposed hybrid model. Thanks to the model, directed reflection is bound with the light-surface interaction just as in surface graphics. A surface is always defined (the isosurface) before shading is applied, so the problem of finding normal vectors will never occur. Compared with some solutions to this problem which resolves it based on numerical hints such as the gradient magnitude, our method tackles the issues at an optical model level, which makes it physically sound.

As a generalization to the conventional shading model, the hybrid model inherits all of the simplicity of the conventional particle-based model. It is simple to implement and easy to use with some well-established volume rendering techniques concerning rendering efficiency and effectiveness, such as preintegration, shadowing and ambient occlusion. However, some subtle modifications might be required. In this paper, we will only describe the way we applied the preintegration algorithms introduced in [2] and [3] for the new model.

Although more transfer functions are introduced into the framework, there is not as much increased effort required from the user to manipulate them, because each set of transfer functions has a distinct physical meaning, which allows the user to incrementally add controllers into the transfer functions with a well-defined objective in mind. In practice, this is a more efficient approach in adjusting the visual appearance.

In the remainder of this paper, we will discuss in detail various issues related to the development and applications of our model. In Section 2, we give a brief overview of related techniques previously used for direct volume rendering. In Section 3, we describe the concepts of the new model and the new volume rendering integral. In Section 4, two estimation algorithms of the model are given. In Section 5, implementation issues, including transfer function specification and some simplifications, are discussed. The experimental results are presented in Section 6.

Some potential applications and the possible extensions of the model are discussed in Section 7.

2 RELATED WORK

2.1 Conventional Volume Rendering

In conventional volume rendering, optical behavior of a volumetric object is modeled using the concept of particles. Max [1] proposed a series of models based on the discussions about “geometric optics effects of the individual particles.” As long as shadows and multiple scattering are not taken into account, the basic behaviors of light-particle interaction, such as absorption, emission, and single scattering, can be expressed using the volume rendering integral

$$I = I_0 \exp\left(-\int_0^{Far} \mu(l)dl\right) + \int_0^{Far} c(l)\mu(l) \exp\left(-\int_0^l \mu(t)dt\right)dl. \quad (1)$$

The integration is performed for a viewing ray cast from the viewpoint, where $l = 0$, to the far end, where $l = Far$. I_0 is the light coming from the background, μ is the per-unit-length extinction coefficient, and c is a wavelength-dependent intensity factor of emission or scattering, which is also called the color of the particles.

In object-ordered rendering techniques, sometimes particles can also be generated explicitly [4]. At an optical model level, it is basically the same as the continuous integration form.

2.2 Shading Problem and Existing Solutions

The Blinn-Phong illumination model [5] was originally developed for surface shading in polygon-based rendering, and has been popularly used for volume rendering since it was introduced in this area by Levoy [6]. Even though Phong shading is well suited for rendering material boundaries, its application to homogenous regions can be a problem, which is said to be due to a lack of a “well-defined” normal vector in such regions. This has been widely observed by researchers such as Kniss et al. [7] and Hadwiger et al. [8], and different solutions have been given.

Kniss et al. [7] tackled this problem using a “surface scalar” term S to interpolate between shaded and unshaded effects, where S is between 0 and 1 and is positively correlated with the gradient magnitude.

It has also been suggested that by considering multiple scattering, ambient occlusion, and the attenuation of incident light, surface shading can be avoided in some cases [8], [9], [10]. However, the specular reflection feature provided by surface rendering is still important for many applications. In such cases, the “surface scalar” method is still often used.

With clipping surfaces, the shading problem of particle-based volume rendering methods can become even more obvious, partially due to the inconsistency of the normal vectors calculated from the volume and the clipping surface. Weiskopf et al. [11] suggested that the normal vector of a clipping surface should be used in the vicinity of

the clipping surface. The situation was quite similar when segmentation was applied [12] before volume rendering.

However, we find none of the previous work attempt to analyze the cause of the problem and solve it at an optical model level. Although sometimes the “surface scalar” method can generate some plausible rendering results, the optical mechanism behind such formulation can hardly be imagined. To fundamentally solve the problem, an optical-model-based solution will be provided in this paper.

2.3 Surface Elements in Volume Rendering

In order to simulate light’s behavior at the boundaries in volume rendering, it is crucial that the boundary surfaces should be properly represented in the first place. Various techniques for representing the surfaces have been proposed. Polygons are introduced into volume rendering by Kreeger and Kaufman [13]. Rotteger et al. [14] suggested that an opaque isosurface can be implicitly inserted by adding Dirac’s delta function with infinite amplitude into the extinction transfer function. Contribution of a semi-transparent isosurface should be estimated every time the isosurface is passed. This can be done efficiently using preintegration (for isosurfaces) proposed by Engel et al. [2]. Kraus [15] pointed out that the limit of an infinite number of semitransparent isosurfaces is equivalent to an “integral in data space.” In this paper, a similar idea is used for modeling the surface response. Surface elements are also extremely important for the simulation of light refraction [16], [17].

Our work is significantly inspired by these research ideas. We found that shading is smoothly integrated when surface elements are used, and the problem of shading homogeneous region never occurs in these frameworks. An interesting finding is that the model behind these surface elements is different from the particle model used for volume rendering both in physical meaning and mathematical expression, but it is possible to merge different models.

In this paper, we extend the existing hybrid rendering frameworks by establishing a versatile optical model including both the particle aspect and the surface aspect of a volume data to allow multiple semitransparent (may be soft) isosurfaces to be rendered together with particle-like contents within a unified framework.

2.4 Preintegrated Volume Rendering

Preintegrated volume rendering [2] is an algorithm for reducing the required sampling frequencies along a ray by preintegrating the ray segments with all possible combinations of starting values and ending values in transfer-function precision. Some improvements to the original method were made by Rotteger and Ertl [18] and Lum et al. [3]. In Lum’s work, a fast algorithm for calculating the lookup tables without problematic approximations and a method for smoothly shading multiple transparent isosurfaces were given.

Any modification to the rendering model can be a challenge to the applicability of preintegration techniques. Fortunately, we find that they can be applied to the new model proposed in this paper with some subtle modifications, with all of their good properties well preserved.

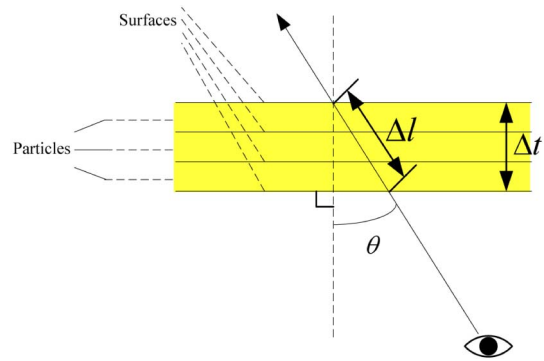


Fig. 2. Microstructure of the hybrid model.

2.5 Other Volume Rendering Models

Apart from the basic particle model mentioned above, many other models have also been proposed for volume rendering, especially in recent years. Global illumination for volume rendering can date back to Sobierajski (1994) [19]. Various approximation methods including shadowing [20], [7], [21], ambient occlusion [8], multiple scattering [10], [22],[23], and the Kubelka-Munk model [24], [25] have been used. Note that using the model proposed in this paper doesn’t prevent one from using the above techniques. Our technique can be useful whenever surface shading is required.

As for modeling color in volume rendering, there are two main approaches. One is based on the trichromatic theory using the RGB color space. The other is to use a spectral representation of color [26], [27], [24]. Theoretically, modeling color using the concept of the wavelength can be more flexible and physically appropriate. However, representing color in RGB mode can be much simpler in terms of implementation if physical correctness regarding color is not important. As a result, we chose to describe our model using wavelength while implementing the model in RGB color space.

3 THE HYBRID MODEL

3.1 Assumptions

Before illustrating the details of our proposed model, let us describe some simple assumptions of this work first. As Fig. 2 shows, a volumetric object is considered to be composed of two kinds of contents interlacing each other: particles, which characterize the individual behavior of particles, and surfaces, which characterize the collective behavior of particles.

For the particle part, just like the conventional model, the attenuation of light is proportional to the density of the particles and the distance a light ray travels. The response from particles can be described by μ_p , the extinction of a unit Δl , and c_p , the scattering intensity of the particles.

For the surface part, the attenuation of a light ray is proportional to the number of surfaces it passes and the opacities of the surfaces. Therefore, instead of using the extinction of a unit Δl , we define μ_s as the average extinction of a unit Δt , so that μ_s is independent of the direction of the light ray. We use c_s for the reflection intensity of the surfaces, which can be calculated from a surface shading model. Note that these two parameters still make sense when the space

between neighboring surfaces tends to be zero, although the number of surfaces tends to be infinite. Refraction is not considered in this paper.

3.2 Total Response of the Mixture

We begin with an ideal situation where the surfaces are simply a set of parallel planes with finite space between neighboring layers. The parameters of both the surfaces and the particles can be assumed to be uniform in a subvolume when it is sufficiently small. Using μ_p , the opacity of each ray segment among the particles is $\alpha_{p-n}(\theta) = 1 - \exp(-\frac{\mu_p \Delta l(\theta)}{n})$. Similarly, the opacity of each surface layer should be $\alpha_{s-n} = 1 - \exp(-\frac{\mu_s \Delta t}{n})$. It can be easily seen that the total opacity along a viewing ray going through the mixture is

$$\begin{aligned} \alpha(\theta) &= 1 - (1 - \alpha_{s-n})^n (1 - \alpha_{p-n}(\theta))^n \\ &= 1 - \exp(-\mu_s \Delta t - \mu_p \Delta l(\theta)). \end{aligned} \quad (2)$$

The average extinction coefficient of a unit Δl is

$$\begin{aligned} \mu(\theta) &= -\ln(1 - \alpha(\theta)) / \Delta l(\theta) \\ &= \mu_s \cos \theta + \mu_p. \end{aligned} \quad (3)$$

It can also be derived that when we let n tends to infinity and the space between surfaces $\frac{\Delta t}{n}$ tends to be zero, the total intensity from a view ray can be expressed in the following form:

$$c(\theta)\mu(\theta) = c_s \mu_s \cos \theta + c_p \mu_p. \quad (4)$$

See the Appendix, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.113>, for more details.

3.3 New Volume Rendering Integral

Based on (3) and (4) which describe the local feature of the hybrid model, we will derive a new volume rendering integral for the case of a single scalar volume in this section.

Let $f(x, y, z)$ be the continuous spatial function interpolated from the scalar volume. In math, the distribution density of isosurfaces should be proportional to the gradient magnitude of f to keep the consistency of isosurfaces. Hence, the extinction coefficient of surfaces μ_s should be considered as the product of the density $|\nabla f|$ and the per-surface extinction coefficient μ_{sg} as (5) shows. μ_{sg} is considered as a constant of an isosurface, which can be specified for each scalar value

$$\mu_s = \mu_{sg} |\nabla f|. \quad (5)$$

A viewing ray is a parameterized path through the volume, which can be expressed as $(x(l), y(l), z(l))$. $x(l)$, $y(l)$, and $z(l)$ are linear functions of l . The speed vector of the viewing ray $\mathbf{r}(l) = (x'(l), y'(l), z'(l))$ is constant for the viewing ray, and the scalar value along the viewing ray is $v(l) = f(x(l), y(l), z(l))$.

Using the normalized gradient vector $\frac{\nabla f}{|\nabla f|}$ as the normal vector of the surface, the term $\cos \theta$ in (3) and (4) can be replaced by a dot product of the speed vector \mathbf{r} and the vector $\frac{\nabla f}{|\nabla f|}$, as (6) and (7) show:

$$\cos \theta = |\mathbf{r} \cdot \mathbf{N}| = \frac{|\mathbf{r} \cdot \nabla f|}{|\nabla f|} = \left| \frac{dv}{dl} \right| / |\nabla f| \quad (6)$$

$$\mu_s \cos \theta = \mu_{sg} \left| \frac{dv}{dl} \right|. \quad (7)$$

Equations (3) and (4) can be transformed into

$$\begin{cases} \mu = \mu_{sg} \left| \frac{dv}{dl} \right| + \mu_p \\ c\mu = c_s \mu_{sg} \left| \frac{dv}{dl} \right| + c_p \mu_p. \end{cases} \quad (8)$$

A new volume rendering integral can then be formulated as

$$\begin{aligned} I &= I_0 \exp \left(- \int_0^{Far} (\mu_{sg}(l) |dv(l)| + \mu_p(l) dl) \right) \\ &+ \int_0^{Far} (c_s(l) \mu_{sg}(l) |dv(l)| + c_p(l) \mu_p(l) dl) \\ &\times \exp \left(- \int_0^l (\mu_{sg}(t) |dv(t)| + \mu_p(t) dt) \right). \end{aligned} \quad (9)$$

3.4 Surface Shading

Just as in polygon-based rendering, the reflection intensity of surface c_s is calculated from some normal-vector-based shading model, such as the Blinn-Phong model. To clarify the wavelength dependency of parameters, we subscript all wavelength-dependent terms with λ . Using surface shading, a primary color $\tilde{c}_{s-\lambda}$ should be defined for the surface. Using a shading model, the surface intensity $c_{s-\lambda}$ can then be calculated, which often takes form such as $c_{s-\lambda} = k_\lambda \tilde{c}_{s-\lambda} + b_\lambda$, where k_λ and b_λ are special functions dependent on the position and posture of the volumetric object as well as the property of light sources.

3.5 Different Kinds of Particles

The new model prevents the use of surface shading for individual particles. However, to extend the flexibility to render different kinds of materials, the individual behavior of particles can be assumed to be a mixture of different kinds of particles, with the total particle response described as μ_p and c_p . In this section, we give the details about what kinds of particles can be defined and how they can be mixed. Here, we define three kinds of different particles, each having unique behaviors that cannot be covered by others, and then combine them together into a general form.

- **Isotropic scattering.** Isotropic scattering particles scatter light from the light source evenly in all directions, and the attenuation of light is equal for all wavelengths. Such behavior is equal to the absorption plus emission particles described in [1]. We use a wavelength-dependent term $c_{p-s-\lambda}$ and a wavelength-independent term μ_{p-s} to describe the contribution of this kind of particles.
- **Selective absorptive.** Selective absorptive particles are purely absorbant, which means that they don't emit nor scatter light. However, the attenuation of light is wavelength dependent. Such behavior is called selective absorption in spectral volume rendering [24]. We describe it using the term $\mu_{p-a-\lambda}$.
- **Pure emissive.** Pure emissive particles only emit light without extinction. Since the light is self-emitted, it doesn't need to be weighted by the attenuation. We use the term $e_{p-\lambda}$ to describe it.

The mixed effect of these three kinds of particles can be formulated as

$$\begin{cases} \mu_{p-\lambda} = \mu_{p-s} + \mu_{p-a-\lambda} \\ c_{p-\lambda}\mu_{p-\lambda} = c_{p-s-\lambda}\mu_{p-s} + e_{p-\lambda}. \end{cases} \quad (10)$$

3.6 Special Cases

By setting some parameters in (9) to zero, two extreme cases can be reached. On one hand, if we set μ_{sg} to zero, we will get a particle only case which is the same as the traditional volume rendering integral without considering the collective effect. On the other hand, if we set μ_p to zero, we will get a surface only case which is the same as the “integral in data space” described in [15].

In principle, two transfer functions should be defined for the isosurfaces, $TF\mu_{sg}(v)$ and $TF\tilde{c}_{s-\lambda}(v)$. There’s yet another special case, which addresses the transfer function $TF\mu_{sg}(v)$. If we add Dirac’s delta function item $\mu_{\delta_{-i}}\delta(v - v_{\delta_{-i}})$ (the limit case of a pulse function) into the transfer function, a discrete isosurface is inserted, and the opacity of the discrete isosurface is $\alpha_i = 1 - e^{-\mu_{\delta_{-i}}}$. This case actually has the same optical model as the volume-polygon hybrid rendering techniques. A preintegrated algorithm should be used; otherwise, the sampling rate required will be infinite.

4 ESTIMATION ALGORITHMS

Estimating the new volume rendering integral (9) is similar to estimating a conventional volume rendering integral. However, some subtle changes are needed, especially when trying to apply a preintegrated algorithm to achieve high-quality rendering.

Graphics hardware has observed a boost in its programmability and processing power during recent years. As a result, GPU-based ray casting becomes a popular volume rendering technique. The advantages of using ray casting over slice-based rendering can be found in [28], [17]. We will discuss three types of ray-casting algorithms for our newly proposed optical model: one is the direct ray-casting algorithm which uses a straightforward sampling and summing scheme, and the other two are preintegrated ray-casting schemes evolved from [2] and [3]. In this section, we assume that the parameters μ_{sg} , $\tilde{c}_{s-\lambda}$, $\mu_{p-\lambda}$, and $c_{p-\lambda}$ relating to surfaces and particles can be achieved using corresponding transfer functions $TF\mu_{sg}$, $TF\tilde{c}_{s-\lambda}$, $TF\mu_{p-\lambda}$, and $TFc_{p-\lambda}$.

4.1 Direct Ray Casting

In our direct ray-casting scheme, scalar samples were taken along each viewing ray. Using the transfer functions, we have μ_{sg} , $\tilde{c}_{s-\lambda}$, $\mu_{p-\lambda}$, and $c_{p-\lambda}$. Applying Blinn-Phong shading to $\tilde{c}_{s-\lambda}$, we get $c_{s-\lambda}$. Normal vectors of the isosurfaces can be estimated using either gradient interpolation or on the fly gradient estimation. For the estimation of $\frac{dv}{dl}$, the previous scalar sample is kept, and the next scalar sample is preloaded. Letting v_0 , v_1 , v_2 be the previous, the current, and the next sample, in each step of a ray, respectively, we pass the value of v_1 to v_0 , v_2 to v_1 , and load the next sample value into v_2 . Using a sample distance Δl , $\frac{dv}{dl}$ is estimated by a central difference as (11) shows:

$$\frac{dv}{dl} \approx \frac{v_2 - v_0}{2\Delta l}. \quad (11)$$

The opacity of a sample is estimated as (12), letting $\hat{\mu}_\lambda\Delta l$ denote $\mu_{sg}\frac{|v_2-v_0|}{2} + \mu_{p-\lambda}\Delta l$:

$$\begin{aligned} \alpha_\lambda &\approx 1 - \exp(-\mu_\lambda\Delta l) \\ &= 1 - \exp\left(-\left(\mu_{sg}\left|\frac{dv}{dl}\right| + \mu_{p-\lambda}\right)\Delta l\right) \\ &\approx 1 - \exp\left(-\mu_{sg}\frac{|v_2-v_0|}{2} - \mu_{p-\lambda}\Delta l\right) \\ &= 1 - \exp(-\hat{\mu}_\lambda\Delta l). \end{aligned} \quad (12)$$

The opacity weighted intensity of the sample is estimated as

$$\begin{aligned} c_\lambda\alpha_\lambda &\approx c_\lambda\mu_\lambda\Delta l\frac{1 - \exp(-\hat{\mu}_\lambda\Delta l)}{\hat{\mu}_\lambda\Delta l} \\ &= \left(c_{s-\lambda}\mu_{sg}\left|\frac{dv}{dl}\right| + c_{p-\lambda}\mu_{p-\lambda}\right)\Delta l\frac{1 - \exp(-\hat{\mu}_\lambda\Delta l)}{\hat{\mu}_\lambda\Delta l} \\ &\approx \left(c_{s-\lambda}\mu_{sg}\frac{|v_2-v_0|}{2} + c_{p-\lambda}\mu_{p-\lambda}\Delta l\right)\frac{1 - \exp(-\hat{\mu}_\lambda\Delta l)}{\hat{\mu}_\lambda\Delta l}. \end{aligned} \quad (13)$$

Then, the composition is done sample by sample using (14), where $A_\lambda(i)$ is the accumulated opacity and $C_\lambda(i)$ is the accumulated intensity of the i th step:

$$\begin{cases} A_\lambda(i) = A_\lambda(i-1) + (1 - A_\lambda(i-1))\alpha_\lambda(i) \\ C_\lambda(i) = C_\lambda(i-1) + (1 - A_\lambda(i-1))\alpha_\lambda(i)c_\lambda(i). \end{cases} \quad (14)$$

Assuming there are n steps casting a ray, the background intensity $I_{0-\lambda}$ is added using

$$I_\lambda = C_\lambda(n) + (1 - A_\lambda(n))I_{0-\lambda}. \quad (15)$$

4.2 Ray Casting with Preintegration

In volume rendering, it is often required to let the transfer functions contain some sharp structures such as edges or spikes, which often lead to an extremely high requirement of sampling frequency for direct ray casting. The preintegration algorithm has been an efficient tool for tackling this kind of problem.

In order to use the existing preintegration algorithms for the new model and have their good properties well preserved, some subtle modifications are required. When using the hybrid model, shading is only applied to the isosurfaces, not to the particles. Thus, two sets of lookup tables are required: one for the isosurfaces, and the other for the particles. Note that shading can only be done during ray casting because normal vectors are not available in the process of building the lookup tables.

The following will discuss the application of the two preintegrated ray-casting schemes. The first one is based on the approximative preintegration given in [2] while the second one is based on the scheme given in [3].

4.2.1 Approximative Preintegration

The approximative preintegration scheme was proposed for a fast estimation of the lookup tables. By neglecting the extinction within the segments, some integral functions are precomputed, and the integration of segments is estimated using the difference of the integral functions.

The integral functions are computed using

$$\begin{aligned}
M_s(x) &= \int_{-\infty}^x TF\mu_{sg}(v)dv \\
\tilde{C}_{s-\lambda}(x) &= \int_{-\infty}^x TF\tilde{c}_{s-\lambda}(v)TF\mu_{sg}(v)dv \\
M_{p-\lambda}(x) &= \int_{-\infty}^x TF\mu_{p-\lambda}(v)dv \\
C_{p-\lambda}(x) &= \int_{-\infty}^x TFc_{p-\lambda}(v)TF\mu_{p-\lambda}(v)dv.
\end{aligned} \tag{16}$$

Opacities and intensities of the segments are estimated using the difference of the integral functions as shown in (17). The intensities $\tilde{c}_{s-\lambda}(v_0, v_1)$ and $c_{p-\lambda}(v_0, v_1)$ here are opacity weighted

$$\begin{cases}
\alpha_{s-\lambda}(v_0, v_1) = |M_s(v_1) - M_s(v_0)| \frac{\alpha_\lambda(v_0, v_1)}{M_\lambda(v_0, v_1)} \\
\tilde{c}_{s-\lambda}(v_0, v_1) = |\tilde{C}_{s-\lambda}(v_1) - \tilde{C}_{s-\lambda}(v_0)| \frac{\alpha_\lambda(v_0, v_1)}{M_\lambda(v_0, v_1)} \\
\alpha_{p-\lambda}(v_0, v_1) = (M_{p-\lambda}(v_1) - M_{p-\lambda}(v_0)) \frac{\Delta l}{v_1 - v_0} \frac{\alpha_\lambda(v_0, v_1)}{M_\lambda(v_0, v_1)} \\
c_{p-\lambda}(v_0, v_1) = (C_{p-\lambda}(v_1) - C_{p-\lambda}(v_0)) \frac{\Delta l}{v_1 - v_0} \frac{\alpha_\lambda(v_0, v_1)}{M_\lambda(v_0, v_1)}
\end{cases}$$

where

$$\begin{aligned}
M_\lambda(v_0, v_1) &= |M_s(v_1) - M_s(v_0)| \\
&\quad + (M_{p-\lambda}(v_1) - M_{p-\lambda}(v_0)) \frac{\Delta l}{v_1 - v_0} \\
\alpha_\lambda(v_0, v_1) &= 1 - \exp(-M_\lambda(v_0, v_1)).
\end{aligned} \tag{17}$$

In comparison with the equations given in [2], (17) is slightly different in that we added in a correcting factor $\frac{\alpha_\lambda(v_0, v_1)}{M_\lambda(v_0, v_1)}$ to avoid oversaturation of the intensity. It can be seen that for the surface part, there's no such factor as $\frac{\Delta l}{v_1 - v_0}$.

Normal vectors needed for shading are estimated by interpolating the gradient vectors between the two ends of each segment. The result is then normalized. Engel et al. [2] suggested using a weighted average. For the case of a single isosurface, the weights of the two ends are decided by the relationship of the isovalue and the value of the two ends. Let v_0 be the starting value, v_1 be the ending value, and v_C be the isovalue of the isosurface; weights are determined by

$$w_0 = \frac{v_1 - v_C}{v_1 - v_0}, \quad w_1 = \frac{v_C - v_0}{v_1 - v_0}. \tag{18}$$

However, in the generalized case there is not necessarily an isovalue. Thus, instead of using an isovalue, we redefined v_C as the "barycenter" of the segment, which is a visibility weighted average scalar value. As shown in (19), it produces another lookup table:

$$\begin{aligned}
V(x) &= \int_{-\infty}^x vTF\mu_{sg}(v)dv \\
v_C(v_0, v_1) &= \frac{V(v_1) - V(v_0)}{M_s(v_1) - M_s(v_0)}.
\end{aligned} \tag{19}$$

The numerical processes of the integrations are straightforward. Note that some Dirac's delta function items may be

contained in $TF\mu_{sg}(v)$. They are converted into step function items by the integrations before being sampled.

4.2.2 Full Preintegration with Improved Shading

Neglecting the extinction within the segments can introduce ordering problems, especially when the value of extinction is large. For quality considerations, such an approximation should be avoided, and a full preintegration should be performed. By brute force, a full preintegration requires too much calculation for the interactive transfer-function manipulation. Lum et al. [3] proposed an algorithm called incremental subrange integration that can calculate an $n \times n$ lookup table in $O(n^2)$ time without problematic approximation. The numerical process of this algorithm can be straightforwardly applied to the new model proposed in this paper.

Another problem for approximated preintegration is the gradient interpolation method used in the algorithm which can cause discontinuities. To tackle this problem, Lum et al. [3] proposed an algorithm called "interpolated preintegrated lighting." This idea can also be borrowed for implementing the new model. For the sake of clarity, we reformulated the idea in the new context.

For a segment of a viewing ray from l_0 to l_1 , its partial integration can be written as

$$\begin{aligned}
\alpha_\lambda(l_0, l_1) &= 1 - \exp\left(-\int_{l_0}^{l_1} (\mu_{sg}(l)|dv(l)| + \mu_{p-\lambda}(l)dl)\right) \\
c_\lambda(l_0, l_1) &= \int_{l_0}^{l_1} (c_{s-\lambda}(l)\mu_{sg}(l)|dv(l)| + c_{p-\lambda}(l)\mu_{p-\lambda}(l)dl) \\
&\quad \times \exp\left(-\int_{l_0}^l (\mu_{sg}(t)|dv(t)| + \mu_{p-\lambda}(t)dt)\right).
\end{aligned} \tag{20}$$

As has been described in Section 3.4, a surface shading model can usually be written in a $c_{s-\lambda} = k_\lambda \tilde{c}_{s-\lambda} + b_\lambda$ form, where k_λ and b_λ are space-dependent functions that are unavailable during the generation of the lookup tables. Following the idea of "interpolated preintegrated lighting," we consider $k_\lambda(l)$ and $b_\lambda(l)$ to be linear between l_0 and l_1 , and the item $c_{s-\lambda}(l)\mu_{sg}(l)$ in (20) can be expanded as (21) shows. Thus, the space-dependent factors can be decoupled from the segment integration:

$$\begin{aligned}
k_\lambda(l) &= \frac{(l_1 - l)k_\lambda(l_0) + (l - l_0)k_\lambda(l_1)}{l_1 - l_0} \\
b_\lambda(l) &= \frac{(l_1 - l)b_\lambda(l_0) + (l - l_0)b_\lambda(l_1)}{l_1 - l_0} \\
c_{s-\lambda}(l)\mu_{sg}(l) &= k_\lambda(l_0) \frac{(l_1 - l)\tilde{c}_{s-\lambda}(l)\mu_{sg}(l)}{l_1 - l_0} + k_\lambda(l_1) \frac{(l - l_0)\tilde{c}_{s-\lambda}(l)\mu_{sg}(l)}{l_1 - l_0} \\
&\quad + b_\lambda(l_0) \frac{(l_1 - l)\mu_{sg}(l)}{l_1 - l_0} + b_\lambda(l_1) \frac{(l - l_0)\mu_{sg}(l)}{l_1 - l_0}.
\end{aligned} \tag{21}$$

In addition, as in all preintegration schemes, $v(l)$ is assumed to be linear between l_0 and l_1 , with $v_0 = v(l_0)$ and $v_1 = v(l_1)$ sampled at the two ends. The task now is to build a pair of lookup table sets for the isosurfaces (a front-weighted and a back-weighted), and another lookup table set for the particles as (22) shows:

$$\begin{cases}
 \alpha_{s_fw-\lambda}(v_0, v_1) \\
 = \int_{v_0}^{v_1} \frac{(v_1 - v)TF\mu_{sg}(v)}{v_1 - v_0} [1 - \alpha_\lambda(v_0, v, v_1 - v_0)] |dv| \\
 \tilde{c}_{s_fw-\lambda}(v_0, v_1) \\
 = \int_{v_0}^{v_1} \frac{(v_1 - v)TF\tilde{c}_{s-\lambda}(v)TF\mu_{sg}(v)}{v_1 - v_0} [1 - \alpha_\lambda(v_0, v, v_1 - v_0)] |dv| \\
 \alpha_{s_bw-\lambda}(v_0, v_1) \\
 = \int_{v_0}^{v_1} \frac{(v - v_0)TF\mu_{sg}(v)}{v_1 - v_0} [1 - \alpha_\lambda(v_0, v, v_1 - v_0)] |dv| \\
 \tilde{c}_{s_bw-\lambda}(v_0, v_1) \\
 = \int_{v_0}^{v_1} \frac{(v - v_0)TF\tilde{c}_{s-\lambda}(v)TF\mu_{sg}(v)}{v_1 - v_0} [1 - \alpha_\lambda(v_0, v, v_1 - v_0)] |dv| \\
 \alpha_{p-\lambda}(v_0, v_1) \\
 = \int_{v_0}^{v_1} \frac{TF\mu_{p-\lambda}(v)\Delta l}{v_1 - v_0} [1 - \alpha_\lambda(v_0, v, v_1 - v_0)] dv \\
 c_{p-\lambda}(v_0, v_1) \\
 = \int_{v_0}^{v_1} \frac{TFc_{p-\lambda}(v)TF\mu_{p-\lambda}(v)\Delta l}{v_1 - v_0} [1 - \alpha_\lambda(v_0, v, v_1 - v_0)] dv
 \end{cases} \quad (22)$$

where

$$\begin{aligned}
 \alpha_\lambda(v_0, v_1, \Delta v) \\
 = 1 - \exp\left(-\int_{v_0}^{v_1} TF\mu_{sg}(v)|dv| + \frac{TF\mu_{p-\lambda}(v)\Delta l}{\Delta v} dv\right).
 \end{aligned}$$

Then, during ray casting, $\tilde{c}_{s_fw-\lambda}$ is shaded with $k_\lambda(l_0)$, $b_\lambda(l_0)$ and $\tilde{c}_{s_bw-\lambda}$ is shaded with $k_\lambda(l_1)$, $b_\lambda(l_1)$ as (23) shows:

$$\begin{aligned}
 \alpha_\lambda(l_0, l_1) \\
 = & \alpha_{s_fw-\lambda}(v(l_0), v(l_1)) \\
 & + \alpha_{s_bw-\lambda}(v(l_0), v(l_1)) \\
 & + \alpha_{p-\lambda}(v(l_0), v(l_1)) \\
 c_\lambda(l_0, l_1) \\
 = & k_\lambda(l_0)\tilde{c}_{s_fw-\lambda}(v(l_0), v(l_1)) \\
 & + b_\lambda(l_0)\alpha_{s_fw-\lambda}(v(l_0), v(l_1)) \\
 & + k_\lambda(l_1)\tilde{c}_{s_bw-\lambda}(v(l_0), v(l_1)) \\
 & + b_\lambda(l_1)\alpha_{s_bw-\lambda}(v(l_0), v(l_1)) \\
 & + c_{p-\lambda}(v(l_0), v(l_1)).
 \end{aligned} \quad (23)$$

A fast algorithm to numerically estimate (22) can be derived following the ideas provided in [3]. The difference between our process and theirs is mainly in the integrals to be estimated. In their case, only the particle lookup tables were involved. These lookup tables consisted of a front-weighted part and a back-weighted part, so a method for the “weighted lookup tables” was given. In our case, a similar strategy is used for the estimation of the surface lookup tables, while the particle lookup tables are similar to the unweighted case (see the Appendix, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.113>, for more details).

Also, Dirac’s delta function items can be converted into step function items via the integrations prior to the process of sampling.

5 IMPLEMENTATION ISSUES

Over the past few years, GPUs have undergone tremendous development and evolved into extremely flexible and powerful processors in terms of programmability. In the meantime, various GPGPU APIs and techniques have been developed to harness the processing power of modern GPUs. Our implementations were based on CUDA [29], and the codes were written in CUDA C.

The implementation is generally straightforward without using any acceleration techniques such as empty-spacing skipping. Even early ray-termination is not used when the attenuation is wavelength dependent. The volume data set and the lookup tables are simply stored as textures. The overall scheme goes as follows: first, some necessary preparation is performed at the CPU side, including the calculation of the lookup tables when the transfer functions are modified. Then, the rendering kernel is launched, and each GPU thread is in charge of a ray casted through a pixel. The ray is first transformed to the voxel coordinate and is trunked by the bounding box of the volume. Then, sample values are fetched along the ray from the volume texture with uniform sampling distance. For the direct ray casting, optical properties are fetched from the 1D-LUTs, and then shaded and merged as described in Section 4.1. For the preintegrated ray casting, the two neighboring samples are used together to achieve the optical properties of the segment from the 2D-LUTs, and then shaded and merged as described in Section 4.2. The normal vector needed for shading is calculated on the fly. The merged values are stored into a GPU buffer. The buffer is mapped to an OpenGL pixel-buffer-object which can be displayed in an OpenGL environment.

Apart from the overall scheme, in this section, we mainly focus on two implementation issues. The first is about some conversions of the parameters needed to make the specification of the transfer functions more intuitive. The second is about some simplifications we made to reduce the number of transfer functions and lookup tables.

5.1 Conversion of Parameters

In Section 4, the transfer functions needed for the model are assumed to be available. However, in practice, some parameters are not intuitive enough. In this section, we define a series of conversions aiming at producing a friendly interface to allow the user to tune the effects.

First, $\mu_{p-\lambda}$ and $c_{p-\lambda}$ represent the collective behavior of different kinds of particles; they are calculated from μ_{p-s} , $\mu_{p-a-\lambda}$, $c_{p-s-\lambda}$, μ_{p-s} , and $e_{p-\lambda}$ according to (10).

Second, the value range of the extinction coefficients μ_{p-s} , $\mu_{p-a-\lambda}$, and μ_{sg} is infinite. Therefore, a direct specification of these parameters is not easy. It is better to set an extinction coefficient indirectly using an opacity term ranging from zero to one. However, to do so requires certain context for the opacity values to be converted to extinction values.

For particles, an opacity value can be defined by using a “standard sampling distance” Δl_{std} (similar concept can be found in [30]). An opacity α can be generally converted into a corresponding extinction value μ using

$$\mu = \frac{-\ln(1 - \alpha)}{\Delta l_{std}}. \quad (24)$$

Using conversions such as this, μ_{p-s} can be calculated from α_{p-s} and $\mu_{p-a-\lambda}$ can be calculated from $\alpha_{p-a-\lambda}$.

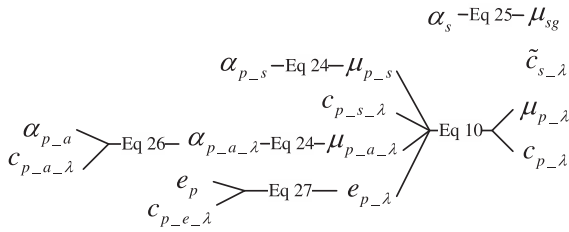


Fig. 3. Conversion of parameters.

Similarly for isosurfaces, an opacity value α_s can be defined by using a “standard scalar difference” Δv_{std} . The corresponding extinction value μ_{sg} can be calculated using

$$\mu_{sg} = \frac{-\ln(1 - \alpha_s)}{\Delta v_{std}}. \quad (25)$$

Third, some parameters may still not be intuitive enough. For selective absorptive particles, the parameter $\alpha_{p-a-\lambda}$ is a wavelength-dependent opacity. It can be more intuitively specified using a transmission color term $c_{p-a-\lambda}$ and a wavelength-independent density term α_{p-a} as (26) shows:

$$\alpha_{p-a-\lambda} = \alpha_{p-a}(1 - c_{p-a-\lambda}). \quad (26)$$

The parameter of pure emissive particles $e_{p-\lambda}$ is specified using a wavelength-independent term e_p and an emission color term $c_{p-e-\lambda}$ as (27) shows. There is actually no upper limit for the term e_p . However, for the oversaturation nature of the light emission effect, the value is usually not very large. We used a range from 0 to 2:

$$e_{p-\lambda} = \frac{c_{p-e-\lambda} e_p}{\Delta l_{std}}. \quad (27)$$

In addition, we may want to include Dirac’s delta function items into $TF\mu_{sg}(v)$. A series of discrete scalar values $v_{\delta-i}$ can be directly used to specify the positions of the spikes, and the corresponding magnitude terms $\mu_{\delta-i}$ can be indirectly specified using opacities of the discrete isosurfaces $\alpha_{\delta-i}$, and we can simply convert them using $\mu_{\delta-i} = -\ln(1 - \alpha_{\delta-i})$. Optionally, colors of the spikes can be specified using a series of colors $\tilde{c}_{\delta-i}$, and values of $TF\tilde{c}_{s-\lambda}(v)$ at $v_{\delta-i}$ can be replaced.

There are four pairs of transfer functions which can be set by the users. Each pair contains a wavelength-dependent term and a wavelength-independent term. From these transfer functions, the four transfer functions used in Section 4 can be built indirectly as Fig. 3 shows. Although the number of transfer functions increases, their meanings are more intuitive so they can be more easily set up.

- $TF\alpha_s(v)$. Opacities for isosurfaces.
- $TF\tilde{c}_{s-\lambda}(v)$. Base colors for isosurfaces.
- $TF\alpha_{p-s}(v)$. Opacities for isotropic scattering particles.
- $TFc_{p-s-\lambda}(v)$. Colors for isotropic scattering particles.
- $TF\alpha_{p-a}(v)$. Densities for selective absorptive particles.
- $TFc_{p-a-\lambda}(v)$. Transmission colors for selective absorptive particles.
- $TFe_p(v)$. Wavelength-independent intensities for pure emissive particles.
- $TFc_{p-e-\lambda}(v)$. Colors for pure emissive particles.

Moreover, there are two global parameters Δl_{std} and Δv_{std} to be set. The first one can be used to adjust the

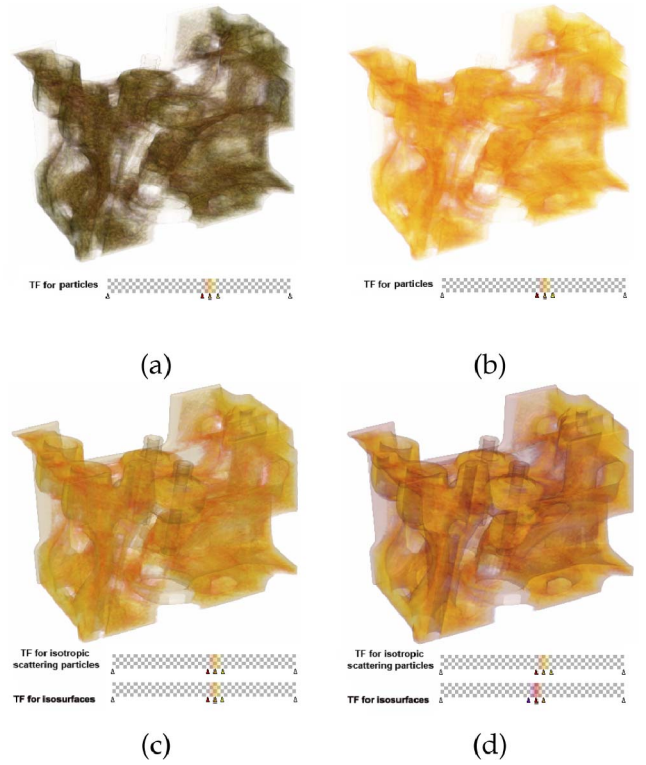


Fig. 4. Rendering the engine data set using different methods. (a) Conventional shaded volume rendering. (b) Conventional unshaded volume rendering. (c) Our new model, using exactly the same transfer functions. (d) Our new model, with a small modification to the surface transfer function.

densities of all particles, and the second one can be used to adjust the opacities of all isosurfaces.

In the experiment part, we will show some examples of tuning the effects using the above set of transfer functions.

5.2 Simplifications

By far, we have discussed issues related to the new model in general. An issue relating to the generality of the model is that too many lookup tables may be required during the process of implementation, which may affect the performance.

To avoid the excessive requirement of lookup tables, we made several simplifications in the implementation of the new model. First, we didn’t implement a full spectrum rendering. Instead, the RGB color space was used. That means we only sampled the spectrum of wavelengths at three points corresponding to the standard red, green, and blue colors; hence, the concept of wavelengths is directly mapped to the RGB color channels. Second, we did not use all kinds of the particles together, and in some cases, the same color transfer functions were used for all the color terms, so the four-color transfer functions $TF\tilde{c}_{s-\lambda}(v)$, $TFc_{p-s-\lambda}(v)$, $TFc_{p-a-\lambda}(v)$, and $TFc_{p-e-\lambda}(v)$ could be reduced into only one shared color transfer function $TFc_{\lambda}(v)$.

6 EXPERIMENTAL RESULTS

6.1 A Model-Oriented Solution to the Shading Problem

In conventional shaded volume rendering, shading artifacts are frequently seen where the gradient magnitude is small. Such artifacts can be avoided using our new model. In Fig. 4a,

the engine data set is rendered using the conventional shaded volume rendering. The transfer function is set to extract some particles from the volume, which are mostly distributed in homogeneous areas. As can be seen from the figure, the shading result is very poor. Compared with its unshaded counterpart in Fig. 4b, it provides little extra information. In contrast, when the new model is used, with exactly the same transfer function for both the particles and the surfaces, some surface features can be clearly seen as Fig. 4c shows. Even better results can be achieved by using a slightly modified surface transfer function (Fig. 4d). Two more experimental results are shown in Fig. 5 to further demonstrate the strength of our model. This time, a clipping plane is placed in the middle of the volume. As can be seen from this figure, the details in homogeneous regions cannot be properly rendered using the conventional model, while the details in these regions can be clearly seen using our model.

6.2 Soft Boundary Surfaces

Image space aliasing is frequently seen in surface rendering and volume-surface hybrid rendering. With the use of transfer functions, our hybrid optical model allows isosurfaces corresponding to different isovalues to be specified in such a way that they transit smoothly from one isosurface to another instead of being discretely placed. As a result, soft boundary surfaces can be displayed in a way that they look the same as discretely placed isosurfaces except that the aliasing at the boundary is smoothed. A comparison between the effect of a discretely placed isosurface and that of a soft boundary surface is shown in Fig. 6. In Fig. 6a, the isosurface is defined by a delta function, while in Fig. 6b, the isosurface is defined by a simple triangular pulse with finite width.

6.3 Tuning the Effects

As described in Section 5, four pairs of transfer functions are provided to control the different components of the model, but they are seldom used all together. At the initial stage, all of the transfer functions are empty, so nothing is displayed. In our implementation, we use controllers to set up the transfer functions. A controller is simply a draggable GUI shape which can be used to assign a pair of optical properties to a scalar value, so that the optical properties between the

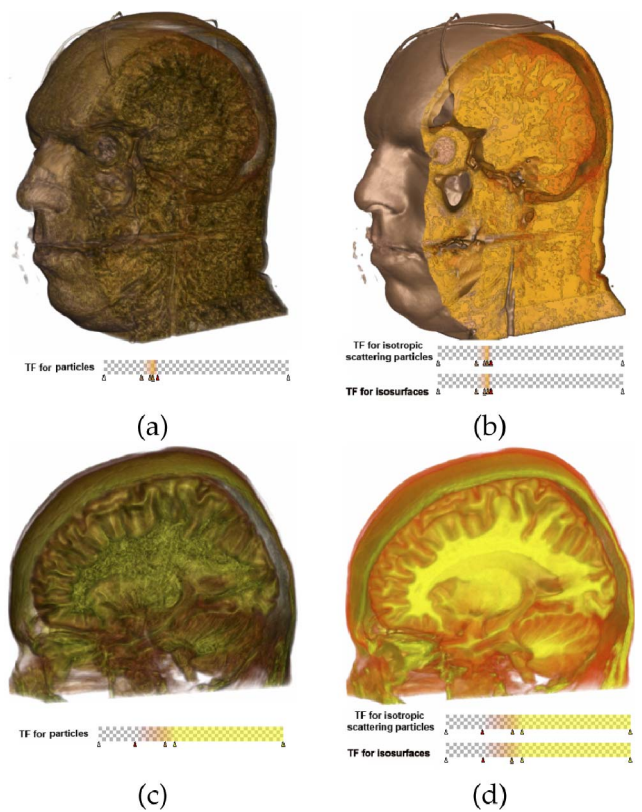


Fig. 5. Comparison of conventional shaded volume rendering with our new model using a clipping plane. (a) The visible human head CT data set rendered using conventional shaded volume rendering. (b) The visible human head CT data set rendered using our new model. (c) An MRI head data set rendered using conventional shaded volume rendering. (d) An MRI head data set rendered using our new model.

controllers can be interpolated. The transfer-function pair is visualized with a colored bar, with the wavelength-dependent property mapped to color channels and the wavelength-independent property mapped to the alpha channel. Dirac's delta function items and the corresponding colors can be assigned with a special kind of controller which we call a delta function controller, each corresponding to a delta function item. See Fig. 7 for the mentioned GUI components.

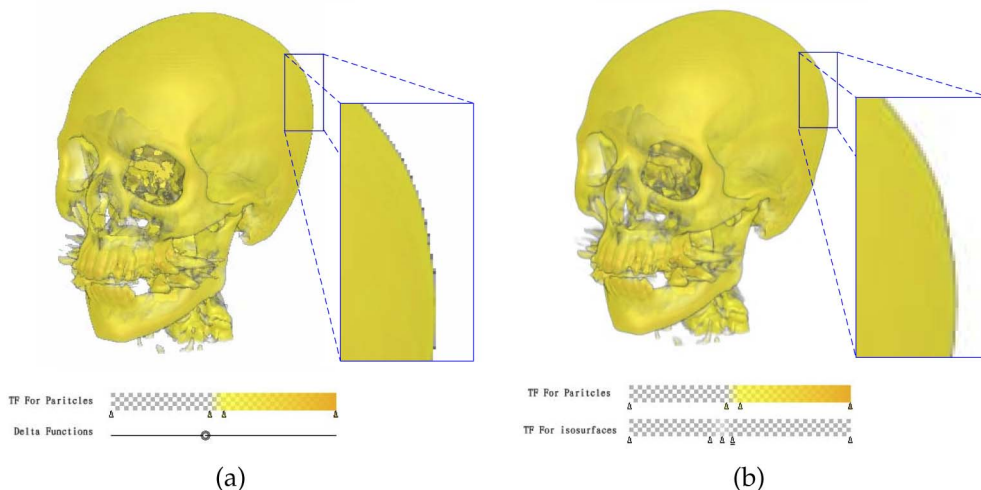


Fig. 6. Antialiasing by soft boundary surfaces. (a) Hybrid rendering of isotropic scattering particles with a discretely placed isosurface. (b) Hybrid rendering of isotropic scattering particles with a soft boundary surface.

User Interface Explanation

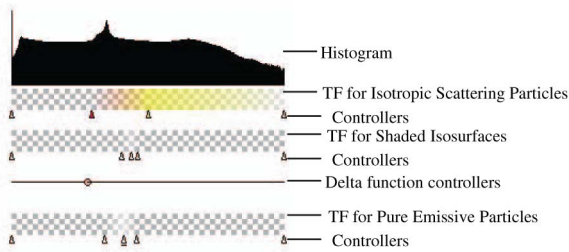


Fig. 7. An explanation to the UI design and the terms used.

When a volume data set is to be visualized, first, the user may choose to play a quick test of critical isovalues by temporarily placing and dragging a delta function controller. Then, according to the requirement of visualization, controllers can be successively added to the transfer functions. To visualize an isosurface, the user can use the delta function controller or set up a finite pulse in the TF for shaded isosurfaces to generate a soft boundary; to visualize some dust-like content within a value range, simply put some controllers into the TF for isotropic scattering particles to define that value range and properties; if some color filtering material (something like dyestuff) is needed, then try adding some controllers into the TF for pure absorptive particles; if the feature of interest is obscured by other contents, then adding some controllers into the TF for pure emissive particles might be helpful to increase the penetration of that feature. The resulted image will be a mixture of all of these factors.

The example in Fig. 8 shows how different kinds of particles and surface elements can be successively added to the scene. In Fig. 8a, a simple transfer function for isotropic scattering particles is set. In Fig. 8b, controllers are added into the transfer function for isosurfaces to display the boundary of the vessels. In Fig. 8c, more controllers are added around the scalar value of the boundary of the heart, so the boundary can be clearly seen. As a side effect, some of the inner structures are concealed. In Fig. 8d, some controllers are added into the transfer function for pure emissive particles to enhance the display of the vessels. In Figs. 8e and 8f, isotropic scattering particles are replaced with selective absorptive particles. Compared with the former, these particles appear more transparent.

The global parameters Δv_{std} and Δl_{std} are manipulated using sliders in exponential scale. They can be used to adjust the global transparency of the surface and particle elements, respectively, while keeping the other element unaffected. In the example shown in Fig. 9, we first set up the transfer function for the pure emissive particles. Then, some controllers are added into the transfer function for isosurfaces, so the boundary of the vessels appeared. From Fig. 9b to Fig. 9d, Δv_{std} is gradually decreased, so that the isosurfaces became more opaque; from Fig. 9d to Fig. 9f, Δl_{std} is gradually increased, so that the intensity of the pure emissive particles is reduced. Overall, a gradual change from particle only rendering to surface only rendering is exhibited.

Fig. 10 shows another example. This time, we first put controllers into the isosurface transfer functions to extract surface features. Then, the transfer functions for isotropic scattering particles are also set up. The figure demonstrates the effect of the global parameter Δl_{std} which controls the global transparency of particles while keeping the surfaces unchanged.

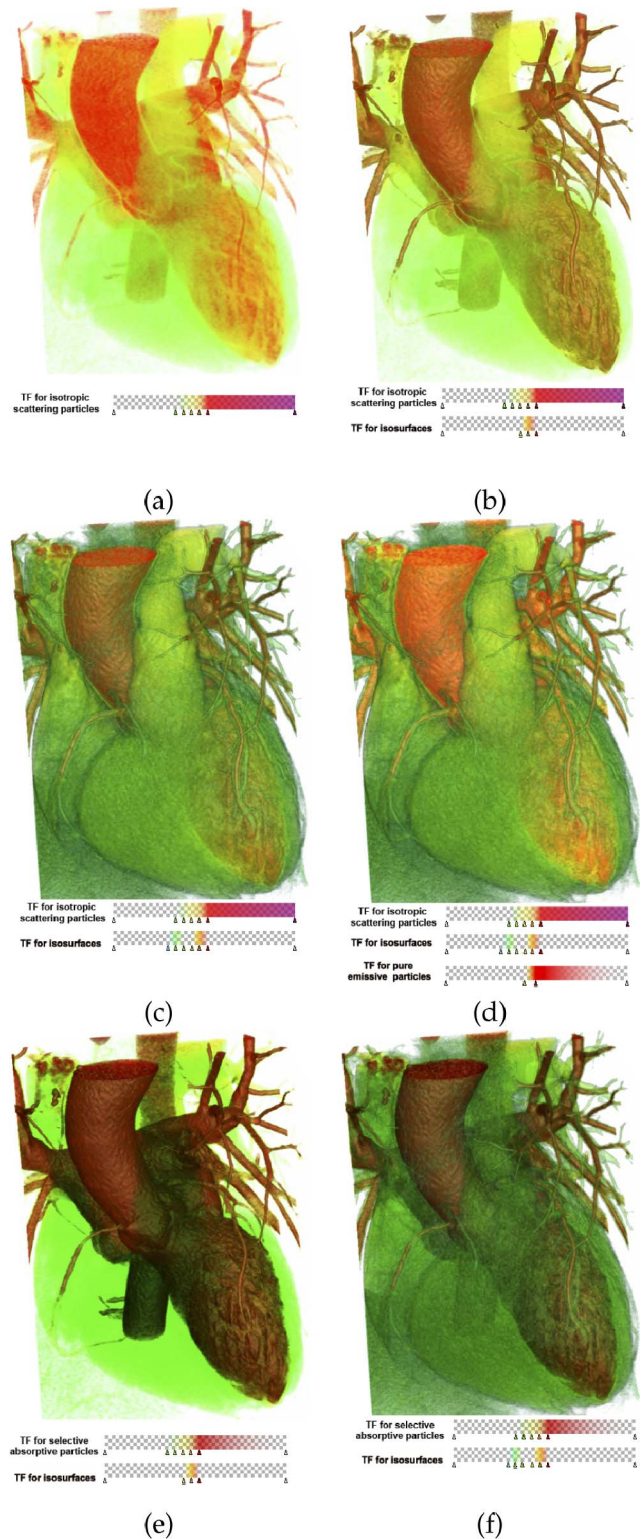


Fig. 8. Rendering the CT heart data set with different combinations of contents. (a) Isotropic scattering particles only. (b) Added a family of shaded isosurfaces to (a). (c) Added another family of shaded isosurfaces to (b). (d) Added some pure emissive particles to (c). (e) Selective absorptive particles with a family of shaded isosurfaces. (f) Added another a family of shaded isosurfaces to (e).

6.4 Using Advanced Estimation Algorithms

As described in Section 4, we adapted two of the preintegration algorithms to the new model with subtle modifications. Here, we tried to make sure that these algorithms actually

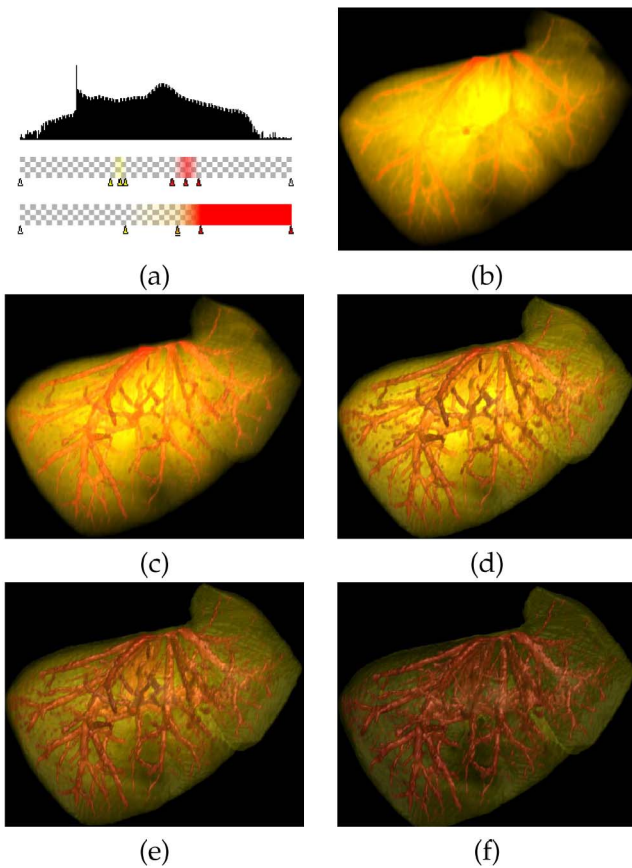


Fig. 9. Rendering the same segmented liver CT data set using pure emissive particles and shaded surfaces. In (a), there's the logarithm histogram of the data set and the transfer functions. The first transfer function is for the shaded surfaces, and the second one is for the pure emissive particles. With the fixed transfer functions, from (b) to (f), the global parameter Δv_{std} are 600, 40, 13, 13, and 13. Δl_{std} are 4, 4, 4, 12, and 600.

worked as well as they did for the particle only model. Sampling artifacts can be most evident when the sampling rate is low and the transfer functions are sharp. Using preintegration algorithms, such artifacts should be dramatically reduced. The results are shown in Fig. 11. The synthetic data set (64^3) is rendered using three different algorithms; all of them are based on the hybrid model and the transfer functions used are identical. Here, we only use isotropic scattering particles. When the data set is rendered with direct ray casting, obvious artifacts are generated due to the inadequate sampling rate as Fig. 11b shows. When it is rendered with an approximative preintegrated ray-casting algorithm, the artifact is reduced, but some artifacts can still be seen when closely observed, as Fig. 11c shows. The remaining artifacts are due to the approximations used. When rendering the data set with fully preintegrated ray casting, the sampling artifacts are totally removed, as Fig. 11d shows.

6.5 Performance Results

To have a better assessment of the performance, we will give the timing results of the above cases in this section. Note that we are not trying to achieve an optimized performance result. In reality, no acceleration technique is used. The tests are performed on a standard PC platform with an Intel Core 2 Duo E6550 CPU at 2.33 GHz and a PCI-E graphics card

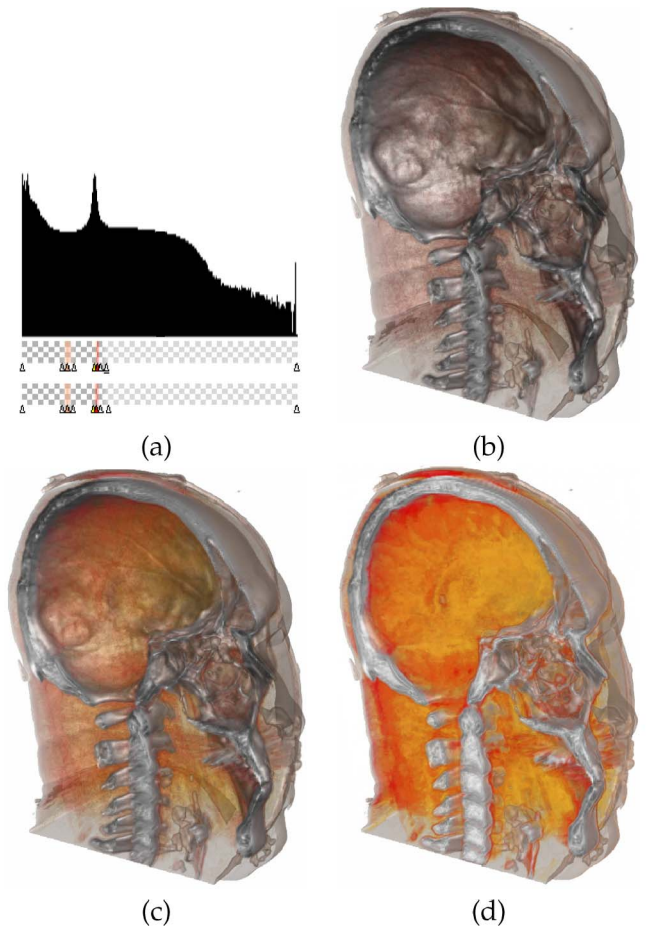


Fig. 10. The visible human head CT data set rendered using the hybrid model with a sagittal clipping plane in the middle. In (a), there's the logarithm histogram of the data set and the transfer functions. The first transfer function is for the shaded surfaces, and the second is for isotropic scattering particles. With the fixed transfer functions, the global parameter Δl_{std} in (b) to (d) are 220, 22, and 7.

powered by an NVIDIA Geforce GT240 GPU. The timing results are shown in Table 1. Most of the cases are rendered at an interactive frame rate. From the Fig. 5 cases, we can see some performance impact of the relatively complicated hybrid model, which is obviously not as significant as it appears to be.

7 DISCUSSION

In this paper, we proposed a novel model for volume rendering that combines particle elements and surface elements together to characterize both the behavior of individual particles and the collective effect of particles. The proposed hybrid optical model originates from an analysis of the microstructure of the volume that interacts with the light rays. The main advantage of the model is its flexibility to smoothly combine different optical elements into a single scene in a physically explainable way. Apart from that, it benefits volume rendering in different aspects. On one hand, compared to the existing methods that address the shading problem, it enables the rendering of isosurfaces with proper attenuation pattern which is consistent to the surface assumption (view-independent transparency). On the other hand, compared to the polygon-based methods, the proposed method doesn't require the reconstruction of

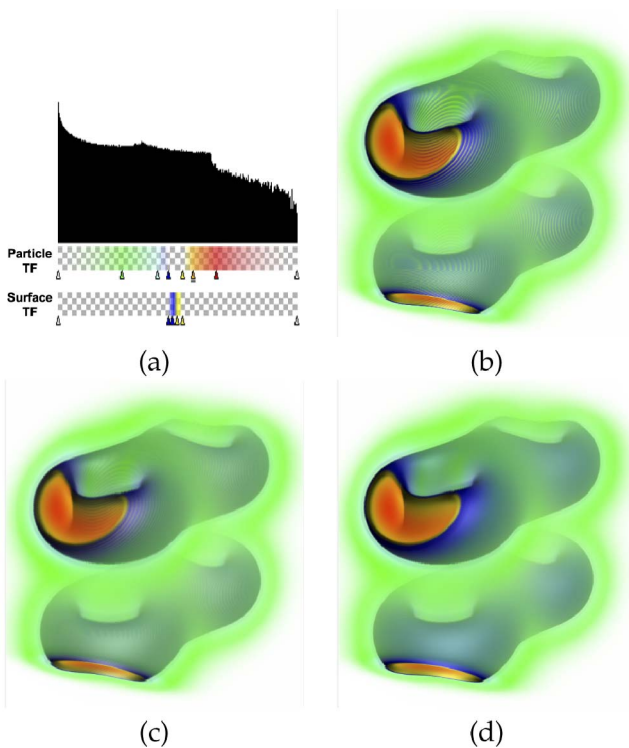


Fig. 11. Rendering a synthetic data set using the hybrid model. All are rendered using the same transfer functions, but with different estimation algorithms. (a) Logarithm histogram of the data set, and the transfer functions used. (b) Rendered with direct ray casting. (c) Rendered with approximative preintegrated ray casting. (d) Rendered with fully preintegrated ray casting.

polygons during the rendering process when the isovalues is changed, and it provides the soft boundary feature which mitigates the image space aliasing problem.

In this paper, we have only discussed about single scalar data sets. However, it is also possible to utilize this idea for visualizing multivariable data sets or integrating scalar fields from different sources such as implicit surface modeling [31]. In that case, we can potentially have multiple choices to assign the scalar values different meanings by mapping them to different roles in the model. For example, we can use one of the scalar values to define a set of isosurfaces and use another scalar value for the color map, just like rendering the surfaces with a texture attached. The preintegration schemes may not be applicable in that case, so more efforts are required to develop a suitable numerical estimation algorithm for the case.

The biggest challenge in using this model is perhaps the adjustment of the extra transfer functions introduced by the new model. Since each set of transfer functions has a distinct physical meaning, the user can adjust these transfer functions with a well-defined objective in mind, so it is not as difficult as it appears to be. However, we believe that there're still rooms for further simplifications to the manipulation of the transfer functions. The use of the automatic transfer-function generation or adaption techniques in the model may also be considered in future applications.

ACKNOWLEDGMENTS

This paper is supported by the National Basic Research Program of China (973 Program) under Grant 2011CB707700;

TABLE 1
Timing Results

Datasets		
Name	Size	Data Type
Engine	256x256x128	8 bit unsigned integer
Visible Human Head	128x256x256	8 bit unsigned integer
Head MRI	181x217x181	8 bit unsigned integer
Head CT	208x256x225	8 bit unsigned integer
Segmented Liver CT	400x310x100	16bit signed integer
Heart CT	299x289x218	16bit unsigned integer
Synthetic Data	64x64x64	32bit float

Timing

Figure	Dataset	Image Size	Rendering Time(ms)
Fig.4(a)	Engine	528x599	114.19
Fig.4(b)	Engine	528x599	116.11
Fig.4(c)	Engine	528x599	114.71
Fig.4(d)	Engine	528x599	115.30
Fig.5(a)	Visible Human Head	543x490	90.88
Fig.5(b)	Visible Human Head	543x490	97.90
Fig.5(c)	Head MRI	558x449	98.68
Fig.5(d)	Head MRI	558x449	106.04
Fig.6(a)	Head CT	558x526	141.62
Fig.6(b)	Head CT	558x526	141.52
Fig.9(b)	Segmented Liver CT	658x674	154.23
Fig.9(c)	Segmented Liver CT	658x674	153.03
Fig.9(d)	Segmented Liver CT	658x674	152.44
Fig.9(e)	Segmented Liver CT	658x674	152.01
Fig.9(f)	Segmented Liver CT	658x674	152.98
Fig.10(b)	Visible Human Head	554x507	113.71
Fig.10(c)	Visible Human Head	554x507	113.29
Fig.10(d)	Visible Human Head	554x507	112.40
Fig.8(a)	Heart CT	367x476	151.03
Fig.8(b)	Heart CT	367x476	148.23
Fig.8(c)	Heart CT	367x476	145.77
Fig.8(d)	Heart CT	367x476	145.48
Fig.8(e)	Heart CT	367x476	159.76
Fig.8(f)	Heart CT	367x476	145.50
Fig.11(b)	Synthetic Data	558x560	40.18
Fig.11(c)	Synthetic Data	558x560	35.71
Fig.11(d)	Synthetic Data	558x560	30.52

the National Natural Science Foundation of China under Grant Nos. 81071218, 60910006, and 30873462; and the Knowledge Innovation Project of the Chinese Academy of Sciences under Grant No. KSCX2-YW-R-262. The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

- [1] N. Max, "Optical Models for Direct Volume Rendering," *IEEE Trans. Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99-108, June 1995.
- [2] K. Engel, M. Kraus, and T. Ertl, "High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading," *Proc. ACM SIGGRAPH/EUROGRAPHICS Workshop Graphics Hardware*, pp. 9-16, 2001.
- [3] E. Lum, B. Wilson, and K. Ma, "High-Quality Lighting and Efficient Pre-Integration for Volume Rendering," *Proc. Joint Eurographics-IEEE TVCG Symp. Visualization (VisSym '04)*, pp. 25-34, 2004.
- [4] N. Sakamoto, T. Kawamura, K. Koyamada, and K. Nozaki, "Technical Section: Improvement of Particle-Based Volume Rendering for Visualizing Irregular Volume Data Sets," *Computer Graphics*, vol. 34, pp. 34-42, <http://dx.doi.org/10.1016/j.cag.2009.12.001>, Feb. 2010.
- [5] B.T. Phong, "Illumination for Computer Generated Pictures," *Comm. ACM*, vol. 18, no. 6, pp. 311-317, 1975.

[6] M. Levoy, "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications*, vol. 8, no. 5, pp. 29-37, May 1988.

[7] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional Transfer Functions for Interactive Volume Rendering," *IEEE Trans. Visualization and Computer Graphics*, vol. 8, no. 3, pp. 270-285, July-Sept. 2002.

[8] M. Hadwiger, P. Ljung, C.R. Salama, and T. Ropinski, "Advanced Illumination Techniques for GPU Volume Raycasting," *SIGGRAPH Asia '08: Proc. ACM SIGGRAPH ASIA 2008 Courses*, pp. 1-166, 2008.

[9] J. Kniss, S. Premoze, C. Hansen, and D. Ebert, "Interactive Translucent Volume Rendering and Procedural Modeling," *Proc. IEEE Visualization*, pp. 109-116, 2002.

[10] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson, "A Model for Volume Lighting and Modeling," *IEEE Trans. Visualization and Computer Graphics*, vol. 9, no. 2, pp. 150-162, Apr.-June 2003.

[11] D. Weiskopf, K. Engel, and T. Ertl, "Interactive Clipping Techniques for Texture-Based Volume Visualization and Volume Shading," *IEEE Trans. Visualization and Computer Graphics*, vol. 9, no. 3, pp. 298-312, July-Sept. 2003.

[12] D. Xiang, J. Tian, F. Yang, Q. Yang, X. Zhang, Q. Li, and X. Liu, "Skeleton Cuts—An Efficient Segmentation Method for Volume Rendering," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 9, pp. 1295-1306, Sept. 2011.

[13] K. Kreeger and A. Kaufman, "Mixing Translucent Polygons with Volumes," *Proc. Conf. Visualization: Celebrating Ten Years*, pp. 191-198, 1999.

[14] S. Röttger, M. Kraus, and T. Ertl, "Hardware-Accelerated Volume and Isosurface Rendering Based on Cell-Projection," *Proc. Conf. Visualization*, pp. 109-116, 2000.

[15] M. Kraus, "Scale-Invariant Volume Rendering," *Proc. IEEE Visualization*, pp. 295-302, 2005.

[16] S. Li and K. Mueller, "Accelerated, High-Quality Refraction Computations for Volume Graphics," *Proc. Int'l Workshop Vol. Graphics*, pp. 73-229, 2005.

[17] S. Stegmaier, M. Strengert, T. Klein, and T. Ertl, "A Simple and Flexible Volume Rendering Framework for Graphics-Hardware-Based Raycasting," *Proc. Int'l Workshop Vol. Graphics*, pp. 187-241, 2005.

[18] S. Röttger and T. Ertl, "A Two-Step Approach for Interactive Pre-Integrated Volume Rendering of Unstructured Grids," *Proc. IEEE Symp. Vol. Visualization and Graphics*, pp. 23-28, 2002.

[19] L.M. Sobierajski, "Global Illumination Models for Volume Rendering," PhD dissertation, State Univ. of New York, 1994.

[20] R. Ratering and U. Behrens, "Adding Shadows to a Texture-Based Volume Renderer," *Proc. IEEE Symp. Vol. Visualization and Graphics*, pp. 39-46, 1998.

[21] T. Ropinski, J. Kasten, and K.H. Hinrichs, "Efficient Shadows for GPU-Based Volume Raycasting," *Proc. 16th Int'l Conf. Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '08)*, pp. 17-24, 2008.

[22] T. Ropinski, C. Döring, and C. Rezk-Salama, "Interactive Volumetric Lighting Simulating Scattering and Shadowing," *Proc. IEEE Pacific Visualization Symp. (PacificVis '10)*, pp. 169-176, Mar. 2010.

[23] C.R. Salama, "GPU-Based Monte-Carlo Volume Raycasting," *Proc. Pacific Conf. Computer Graphics and Applications*, pp. 411-414, 2007.

[24] A. Abdul-Rahman and M. Chen, "Spectral Volume Rendering Based on the Kubelka-Munk Theory," *Computer Graphics Forum*, vol. 24, no. 3, pp. 413-422, 2005.

[25] M. Strengert, T. Klein, R. Botchen, S. Stegmaier, M. Chen, and T. Ertl, "Spectral Volume Rendering Using GPU-Based Raycasting," *The Visual Computer*, vol. 22, no. 8, pp. 550-561, 2006.

[26] H. Noordmans, H. van der Voort, and A. Smeulders, "Spectral Volume Rendering," *IEEE Trans. Visualization and Computer Graphics*, vol. 6, no. 3, pp. 196-207, July-Sept. 2000.

[27] S. Bergner, T. Möller, M.S. Drew, and G.D. Finlayson, "Interactive Spectral Volume Rendering," *VIS '02: Proc. IEEE Visualization Conf.*, pp. 101-108, 2002.

[28] K. Engel, M. Hadwiger, J.M. Kniss, A.E. Lefohn, C.R. Salama, and D. Weiskopf, "Real-Time Volume Graphics," *SIGGRAPH: Proc. ACM SIGGRAPH 2004 Course Notes*, p. 29, 2004.

[29] NVIDIA "Nvidia Cuda Programming Guide Version 2.3," NVIDIA Corporation, July 2009.

[30] M. Chen and J. Tucker, "Constructive Volume Geometry," *Computer Graphics Forum*, vol. 19, no. 4, pp. 281-293, 2000.

[31] Q. Li and J. Tian, "2D Piecewise Algebraic Splines for Implicit Modeling," *ACM Trans. Graphics*, vol. 28, pp. 13:1-13:19, <http://doi.acm.org/10.1145/1516522.1516524>, May 2009.



Fei Yang received the BE degree in biomedical engineering from Beijing Jiaotong University, Beijing, China, in 2009. He is currently working toward the PhD degree in the Medical Image Processing Group at the Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include volume rendering, medical image analysis, and GPU computing.



Qingde Li received the PhD degree in computer science from the University of Hull, Hull, United Kingdom, in 2002. From 1998 to 2002, he was a professor in the School of Mathematics and Computer Science, Anhui Normal University, Anhui, China. He was a visiting scholar in the Department of Applied Statistics, University of Leeds, United Kingdom, from October 1990 to May 1992, and in the Department of Computing, University of Bradford, Bradford, United Kingdom, from September 1996 to August 1997. He is currently a lecturer in the Department of Computer Science, University of Hull. His recent research interests include GPU-based volume data visualization, implicit geometric modeling, and computer graphics.



Dehui Xiang received the BE degree in automation from Sichuan University, Sichuan, China, in 2007. He is currently working toward the PhD degree in the Medical Image Processing Group at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include volume rendering, medical image analysis, computer vision, and pattern recognition.



Yong Cao received the PhD degree in computer science from the University of California at Los Angeles, Los Angeles, in 2005. He is currently an assistant professor of computer science at the Virginia Polytechnic Institute and State University and the director of Animation and Gaming Research Lab. His research focuses on animation, high-performance computing, and video game-based learning.



Jie Tian (M'02, SM'06, F'10) received the PhD degree (Hons) in artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1992. From 1995 to 1996, he was a postdoctoral fellow at the Medical Image Processing Group, University of Pennsylvania, Philadelphia. Since 1997, he has been a professor at the Institute of Automation, Chinese Academy of Sciences, where he has been involved in the research in Medical Image Processing Group. His research interests include the medical image process and analysis and pattern recognition. He is the Beijing Chapter Chair of The Engineering in Medicine and Biology Society of the IEEE. He was elected as a fellow of the IEEE in 2010.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.