Interactive Translucent Volume Rendering and Procedural Modeling

Joe Kniss* Simon Premože† Charles Hansen‡ David Ebert§

*‡Scientific Computing and Imaging Institute
†‡School of Computing, University of Utah
§School of Electrical and Computer Engineering
Purdue University

Figure 1: A translucent fish rendered from a CT scan of a carp. Left: Blinn-Phong shading. Center and right: translucent volume shading.

ABSTRACT

Direct volume rendering is a commonly used technique in visualization applications. Many of these applications require sophisticated shading models to capture subtle lighting effects and characteristics of volumetric data and materials. Many common objects and natural phenomena exhibit visual quality that cannot be captured using simple lighting models or cannot be solved at interactive rates using more sophisticated methods. We present a simple yet effective interactive shading model which captures volumetric light attenuation effects to produce volumetric shadows and the subtle appearance of translucency. We also present a technique for volume displacement or perturbation that allows realistic interactive modeling of high frequency detail for real and synthetic volumetric data.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—3D Graphics

Keywords: Volume rendering, shading model, volume modeling, procedural modeling

1 INTRODUCTION

Direct volume rendering is widely used in visualization applications. Many of these applications render semi-transparent surfaces without shadows lit by an approximation to the Blinn-Phong local surface shading model. This shading model adequately renders such surfaces but it does not provide sufficient lighting characteristics for translucent materials or materials where scattering dominates the visual appearance. Shadows also substantially add to the visual perception of volume rendered data but shadows are not typically utilized in interactive direct volume rendering because of their high computational expense.

Several studies have shown that the appearance of many common objects is dominated by subsurface scattering effects [4] [13]. This is especially true for natural phenomena such as smoke and clouds but is also true for wax, skin, and other translucent materials.

While the effects of multiple scattering are important, physically accurate computation of them is not necessarily required. In fact, interactive rendering for visualization already often employs such approaches (e.g. ambient light, OpenGL style fog, even the Blinn-Phong surface shading model). Interactivity for visualization is important since it aids in rapidly and accurately setting transfer functions[18], as well as providing important visual cues about spatial relationships in the data. While it is possible to precompute multiple scattering effects, for direct volume rendering such methods are dependent on the viewpoint, light position, transfer function and other rendering parameters which defeat the purpose of interactivity.

The ability to add detail to volumetric data has also been a computational bottleneck. Volume perturbation methods allow such effects at the cost of the time and memory to precompute such details before rendering. Volume perturbation methods have also been employed for modeling natural phenomena such as clouds. Such models, when coupled with a shading model with the visual appearance of scattering, can produce high quality visualizations of clouds and other natural phenomena, as well as introduce visually pleasing details to material boundaries (e.g. direct volume rendered isosurfaces).

In this paper, we present a simple yet effective interactive shading model that captures volumetric light attenuation effects to produce volumetric shadows and the qualitative appearance of scattering. This is shown in Figure 1. On the left is the standard surface shading of a CT scan of a carp. In the middle, rendered with the same transfer function, is our improved shading model. The image on the right has the light repositioned slightly behind the fish. In this paper we also present a technique for volume displacement or perturbation that allows realistic interactive modeling of clouds as well as the introduction of details to volumetric data.

*jmk@cs.utah.edu
†premoze@cs.utah.edu
‡hansen@cs.utah.edu
§ebertd@ecn.purdue.edu

IEEE Visualization 2002 Oct. 27 - Nov. 1, 2002, Boston, MA, USA
0-7803-7498-3/02/$17.00 © 2002 IEEE

2 Background and Previous Work

Volume Shading

The classic volume rendering formulation for visualization was proposed by Levoy [22]. This model is a very simplified approximation of volumetric light transport utilizing only emission, absorption and surface shading. This model has been used extensively for interactive and graphics hardware based volume rendering techniques [5, 7, 38]. Krueger applied the volume rendering equation to create meaningful scientific visualizations[19]. Direct attenuation through the volume or shadows can be added to the classic approximation using a brute force method where the attenuation is computed from each sample to the light source. Typical optimizations precompute the attenuation from the light source and store it in an additional 3D volume. This approach has been implemented in both software [20] and hardware [2, 29]. Recently a method has been proposed that computes direct attenuation in screen space, providing sharper shadows with less blur by using a half angle slicing technique [18]. Max emphasized the importance of light transport and surveyed several models for light interactions inside a volume, special attention is given to calculation methods for the multiple scattering [23]. Although the described methods are accurate, the computational expense and storage requirements are large and do not allow interactivity.

Participating Media

A vast amount of literature on light transport and scattering exists. The non-linear integral scattering equation that describes the light transport and scattering events inside a volume has been studied extensively by Ambarzumian[1], Chandrasekhar[6] and van de Hulst[36]. Their work ranges in complexity from semi-infinite homogeneous isotropic atmospheres to finite inhomogeneous anisotropic atmospheres.

Blinn was one of the first researchers to recognize importance of volumetric scattering for computer graphics and visualization applications. He presented a model for the reflection and transmission of light through thin clouds of particles based on probabilistic arguments and single scattering approximation in which Fresnel effects were considered[3]. Kajiya and von Herzen described a model for rendering arbitrary volume densities that included expensive multiple scattering computation. The radiative transport equation[14] cannot be solved analytically except for some simple configurations. Expensive and sophisticated numerical methods must be employed to compute the radiance distribution to a desired accuracy. Finite element methods are commonly used to solve transport equations. Rushmeier presented zonal finite element methods for isotropic scattering in participating media[34][33]. Max *et al.*[24] used a one-dimensional scattering equation to compute the light transport in tree canopies by solving a system of differential equations through the application of the Fourier transform. The method becomes expensive for forward peaked phase functions, as the hemisphere needs to be more finely discretized. Spherical harmonics were also used by Kajiya and von Herzen[16] to compute anisotropic scattering as well as discrete ordinate methods (Languenou *et al.*[21]).

Monte Carlo methods are robust and simple techniques for solving light transport equation. Hanrahan and Krueger modeled scattering in layered surfaces with linear transport theory and derived explicit formulas for backscattering and transmission[11]. The model is powerful and robust, but suffers from standard Monte Carlo problems such as slow convergence and noise. Pharr and Hanrahan described a mathematical framework[32] for solving the scattering equation in context of a variety of rendering problems and also described a numerical Monte Carlo sampling method. Jensen and Christensen described a two-pass approach to light transport in participating media[12] using a volumetric photon map. The method is simple, robust and efficient and it is able to handle arbitrary configurations. Dorsey *et al.*[8] described a method for full volumetric light transport inside stone structures using a volumetric photon map representation.

Recently, Jensen *et al.* introduced computationally efficient analytical diffusion approximation to multiple scattering[13], which is especially applicable for homogeneous materials that exhibit considerable subsurface light transport. The model does not appear to be easily extendible to volumes with arbitrary optical properties. Several other specialized approximations have been developed for particular natural phenomena. Nishita *et al.*[27] presented an approximation to light transport inside clouds and Nishita[26] an overview of light transport and scattering methods for natural environments[26]. These approximations are not generalizable for volume rendering applications because of the limiting assumptions made in deriving the approximations.

Procedural Volume Modeling

A number of authors have developed techniques for procedurally simulating volumetric features. These approaches can be generally classified into full volumetric simulations and thin surface volumes. Kajiya and Kay[15] introduced the idea of modeling fine surface structures with thin volumes to simulate hair and fur. This approach has been extended by Neyret[25] for simulating more complex natural structures. Perlin's seminal work in procedural simulation of noise and turbulence [30] forms the foundation for most volumetric procedural simulation techniques. Perlin created the first volumetric procedural models in 1989[31]. This work has been extended by Musgrave[9] for modeling clouds and Ebert for modeling steam, fog, smoke, and clouds[9].

3 Model

Optical properties that affect the appearance of an illuminated volume are density, absorption, scattering and emission from individual particles or molecules in the volume. For realistic visualization of volumetric data or participating media, an optical model must take into account the optical properties, external illumination, as well as light transport within the volume. The light transport that ultimately determines the appearance of the volume can be described with the volume rendering equation [14] as a series of scattering, absorption and emission events. Max surveyed many optical models for volume rendering applications [23] ranging from very simple to very complex, and accurate models that account for all interactions within the volume.

Max [23] and Jensen *et al.* [13] clearly demonstrate that the effects of multiple scattering and indirect illumination are important for volume rendering applications. However, accurate simulations of full light transport are computationally expensive and do not permit interactivity such as changing the illumination or transfer function. Analytical approximations exist, but they are severely restricted by underlying assumptions, such as homogeneous optical properties and density, simple lighting or unrealistic boundary conditions. These analytical approximations cannot be used for arbitrary volumes or real scanned data where optical properties such as absorption and scattering coefficients are hard to obtain.

Our goal is to realistically visualize arbitrary volumetric materials at interactive rates without any restrictions on external illumination or optical properties. Therefore, we have developed an empirical optical model for volume rendering that captures complex light transport and is based on empirical observations of appearance of volumetric materials, as well as some recent theoret-

ical results ([4][17][37]). While the effects of multiple scattering and indirect illumination are very important, they do not need to be computed accurately for visualization purposes. We are only concerned with the qualitative properties and appearance rather than quantitative accuracy. Our model focuses on capturing the appearance of translucency which is a result of significant multiple scattering events. Color bleeding and diffusion of light across boundaries are also consequences of multiple scattering captured by our model. However, other global effects such as full global illumination, backward scattering and volumetric light sources (emission) are not part of our model.

The classic volume rendering model is:

$$I_{eye} = I_B * T_e(0) + \int_0^{eye} T_e(s) * g(s) * f_s(s) ds \quad (1)$$

$$T_e(s) = exp\left(-\int_s^{eye} \tau(x) dx\right) \quad (2)$$

Where I_B is the background light intensity, $g(s)$ is the emission term at sample s, $f_s(s)$ is the Blinn-Phong surface shading model evaluated using the normalized gradient of the scalar data field at s, and $\tau(x)$ is the extinction coefficient at the sample x. For a concise derivation of this equation and the discrete solution used in volume rendering, see [23].

Shadows can be added to the model as such:

$$I_{eye} = I_B * T_e(0) + \int_0^{eye} T_e(s) * g(s) * f_s(s) * I_l(s) ds \quad (3)$$

$$I_l(s) = I_l(0) * exp\left(-\int_s^{light} \tau(x) dx\right) \quad (4)$$

Where $I_l(0)$ is the light intensity, $I_l(s)$ is the light intensity at the sample s, and $g(s)$ can now be thought of as a reflective term rather than an emissive term. Notice that $I_l(s)$ is essentially the same as $T_e(s)$ except that the integral is computed toward the light rather than the eye.

Our empirical volume shading model adds a blurred indirect light contribution at each sample:

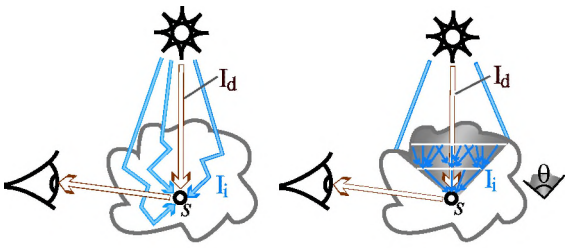
$$I_{eye} = I_0 * T_e(0) + \int_0^{eye} T_e(s) * C(s) * I_l(s) ds \quad (5)$$

$$C(s) = g(s)\left((1 - S(s)) + f_s(s) S(s)\right) \quad (6)$$

$$I_l(s) = I_l(0) * exp\left(-\int_s^{light} \tau(x) dx\right) + I_l(0) * exp\left(-\int_s^{light} \tau_i(x) dx\right) \mathbf{Blur}(\theta) \quad (7)$$

Where $\tau_i(s)$ is the indirect light extinction term, $C(s)$ is the reflective color at the sample s, $S(s)$ is a surface shading parameter, and I_l is the sum of the direct light and the indirect light contributions.

In general, light transport in participating media must take into account the incoming light from all directions, as seen in Figure 2(a). The difficulty in solving this problem is similar to other global illumination problems in computer graphics: each element in the scene can potentially contribute indirect light to every other element. It is therefore quite understandable that accurate solutions to this problem are too computationally expensive for interactive applications and approximations are needed. The net effect of indirect lighting, however, is effectively a diffusion of light through the volume. Light travels farther in the volume than it would if only direct attenuation is taken into account. Translucency implies blurring of the light as it travels through the medium due to scattering effects. We can approximate this effect by simply blurring the light



(a) General Light Transport (b) Our Approximation

Figure 2: On the left is the general case of direct illumination I_d and scattered indirect illumination I_i. On the right is our shading model which includes the direct illumination I_d and approximates the indirect, I_i, by blurring within the shaded region. Theta is the angle indicated by the shaded region.

in some neighborhood and allowing it to attenuate less in the light direction. Figure 3 shows how the effect of translucency is captured by our model. The upper left image, a wax candle, is an example of a common translucent object. The upper right image is a volume rendering using our model. Notice that the light penetrates much deeper into the material than it does with direct attenuation alone (volumetric shadows), seen in the lower right image. Also notice the pronounced hue shift from white to orange to black due to an indirect attenuation term that attenuates blue slightly more that red or green. The lower left image shows the effect of changing just the reflective color to a pale blue.

The diffusion approximation[37, 10] models the light transport in multiple scattering media as a random walk. This results in the light being diffused within the volume. The $\mathbf{Blur}(\theta)$ operation in Equation 7 averages the incoming light within the cone with an apex angle θ in the direction of the light (Figure 2(b)). The indirect lighting at a particular sample is only dependent on a local neighborhood of samples computed in the previous iteration and shown as the arrows between slices. This operation models light diffusion by convolving several random sampling points with a Gaussian filter.

The surface shading parameter ($S(s)$) in Equation 6 is a number between one and zero that describes the degree to which a sample should be surface shaded. It is used to interpolate between surface shading and no surface shading. This value can be added to the transfer function allowing the user to specify whether or not a classified material should be surface shaded. It can also be set automatically using the gradient magnitude at the sample, as in [18]. Here, we assume that classified regions will be surface-like if the gradient magnitude is high and therefore should be shaded as such. In contrast, homogeneous regions, which have low gradient magnitudes, should only be shaded using light attenuation.

4 Implementation

Our approach for simulating light transport is designed to provide interactive or near interactive frame rates for volume rendering when the transfer function, light direction, or volume data are not static. Therefore, the light intensity at each sample must be recomputed every frame. Our method for computing light transport is done in screen space resolution, allowing the computational complexity to match the level of detail. Since the computation of light transport is decoupled from the resolution of the volume data, we can also accurately compute lighting for volumes with high fre-

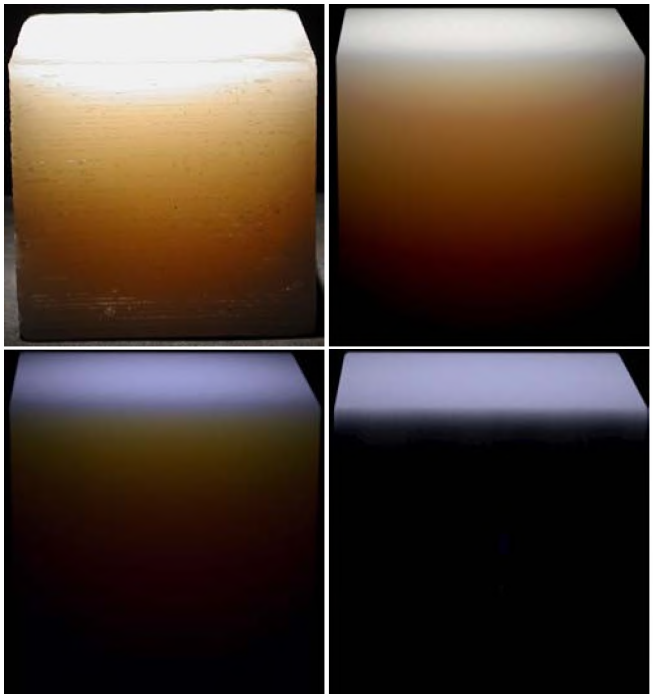


Figure 3: Translucent volume shading. The upper left image is a photograph of wax block illuminated from above with a focused flashlight. The upper right image is a volume rendering with a white reflective color and a desaturated orange transport color (1− *indirect attenuation*). The lower left image has a bright blue reflective color and the same transport color as the upper right image. The lower right image shows the effect of light transport that only takes into account direct attenuation.

quency displacement effects, which are described in the second half of this section.

4.1 Volume Shading

The traditional volume rendering pipeline only requires two optical properties for each material: extinction and material color. However, rather than specifying the extinction term, which is a value in the range zero to infinity, a more intuitive opacity, or alpha, term is used:

$$\alpha = 1 - exp(-\tau(x)). \tag{8}$$

The material color is the light emitted by the material in the simplified absorption/emission volume rendering model, however, the material color can be thought of as the diffuse reflectance if shadows are included in the model. In addition to these values, our model adds an indirect attenuation term to the transfer function. This term is spectral, meaning that it describes the indirect attenuation of light for each of the R, G, and B color components. Similar to extinction, the indirect attenuation can be specified in terms of an indirect alpha:

$$\alpha_i = 1 - exp(-\tau_i(x)) \tag{9}$$

While this is useful for computing the attenuation, we have found it non-intuitive for user specification. We prefer to specify a *transport color* which is $1 - \alpha_i$ since this is the color the indirect light will become as it is attenuated by the material. The alpha value can also be treated as a spectral term; the details of this process can be found in [28]. For simplicity sake, we will treat the alpha as an achromatic

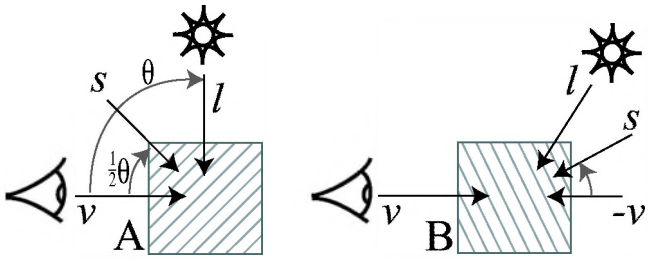


Figure 4: Half angle slice axis for light transport.

value since our aim is to clearly demonstrate indirect attenuation in interactive volume rendering.

Our volume rendering pipeline computes the transport of light through the volume in lock step with the accumulation of light for the eye. The method uses the half angle slicing proposed for volume shadow computation in [18], where the slice axis is halfway between the light and view directions or halfway between the light and inverted view directions depending on the sign of the dot product of the two (see Figure 4). This modification of the slicing axis allows us to render each slice from the point of view of both the observer and the light. This achieves the effect of a high resolution shadow map without the requirement of pre-computation and storage. A total of three image buffers are required to implement our model. Two buffers are maintained for the attenuation of light in the light direction (**current** and **next**), in addition to the buffer for the accumulation of light for the observer, which is typically the frame buffer.

In the first pass, a slice is rendered from the observer's point of view. In this step, the transfer function is evaluated using a dependent texture read for the reflective color and alpha. In the hardware fragment shading stage, the reflective color is multiplied by the sum of one minus the indirect and direct light attenuation previously computed at that slice position in the **current** light buffer. This color is then blended into the observer buffer using the alpha value from the transfer function.

In the second pass, a slice is rendered into the **next** light buffer from the light's point of view to compute the lighting for the next iteration. Two light buffers are maintained to accommodate the blur operation required for the indirect attenuation. Rather than blend slices using a standard OpenGl blend operation, we explicitly compute the blend in the fragment shading stage. The **current** light buffer is sampled once in the first pass, for the observer, and multiple times in the second pass, for the light, using the *render to texture* OpenGL extension. Whereas, the **next** light buffer, is rendered to only in the second pass. This relationship changes after the second pass so that the **next** buffer becomes the **current** and *vice versa*. We call this approach *ping pong blending*. In the fragment shading stage, the texture coordinates for the **current** light buffer, in all but one texture unit, are modified per-pixel using a random noise texture as discussed in the next section. The number of samples used for the computation of the indirect light is limited by the number of texture units. Currently, we use four samples. Randomizing the sample offsets masks some artifacts caused by this coarse sampling. The amount of this offset is bounded based on a user defined blur angle (θ) and the sample distance (d):

$$offset \leq d \tan(\frac{\theta}{2}) \tag{10}$$

The **current** light buffer is then read using the new texture coordinates. These values are weighted and summed to compute the blurred inward flux at the sample. The transfer function is evaluated for the incoming slice data to obtain the indirect attenuation

(α_i) and direct attenuation (α) values for the current slice. The blurred inward flux is attenuated using α_i and written to the RGB components of the **next** light buffer. The alpha value from the **current** light buffer with the unmodified texture coordinates is blended with the α value from the transfer function to compute the direct attenuation and stored in the alpha component of the **next** light buffer.

4.2 Volume Perturbation

One drawback of volume based graphics is that high frequency details cannot be represented in small volumes. These high frequency details are essential for capturing the characteristics of many volumetric objects such as clouds, smoke, trees, hair, and fur. Procedural noise simulation is a very powerful tool to use with small volumes to produce visually compelling simulations of these types of volumetric objects. Our approach is similar to Ebert's approach for modeling clouds[9]; use a coarse technique for modeling the macrostructure and use procedural noise based simulations for the microstructure. We have adapted this approach to interactive volume rendering through two volume perturbation approaches which are efficient on modern graphics hardware. The first approach is used to perturb optical properties in the shading stage while the second approach is used to perturb the volume itself.

Both volume perturbation approaches employ a small 3D perturbation volume, 32^3. Each texel is initialized with four random 8-bit numbers, stored as RGBA components, and blurred slightly to hide the artifacts caused by trilinear interpolation. Texel access is then set to repeat. An additional pass is required for both approaches due to limitations imposed on the number of textures which can be simultaneously applied to a polygon, and the number of sequential dependent texture reads permitted. The additional pass occurs before the steps outlined in the previous section. Multiple copies of the noise texture are applied to each slice at different scales. They are then weighted and summed per pixel. To animate the perturbation, we add a different offset to each noise texture's coordinates and update it each frame.

Our first approach is similar to Ebert's lattice based noise approach [9]. It uses the four per-pixel noise components to modify the optical properties of the volume after the the transfer function has been evaluated. This approach makes the materials appear to have inhomogeneities. We allow the user to select which optical properties are modified. This technique is used to get the subtle iridescence effects seen in Figure 6(bottom).

Our second approach is closely related to Peachey's vector based noise simulation technique [9]. It uses the noise to modify the location of the data access for the volume. In this case three components of the noise texture form a vector, which is added to the texture coordinates for the volume data per pixel. The data is then read using a dependent texture read. The perturbed data is rendered to a pixel buffer that it is used instead of the original volume data. Figure 5 illustrates this process. **A** shows the original texture data. **B** shows how the perturbation texture is applied to the polygon twice, once to achieve low frequency with high amplitude perturbations (large arrows) and again to achieve high frequency with low amplitude perturbations (small arrows). Notice that the high frequency content is created by allowing the texture to repeat. Figure 5 **C** shows the resulting texture coordinate perturbation field when the multiple displacements are weighted and summed. **D** shows the image generated when the texture is read using the perturbed texture coordinates. Figure 6 shows how a coarse volume model can be combined with our volume perturbation technique to produce an extremely detailed interactively rendered cloud. The original 64^3 voxel dataset is generated from a simple combination of volumetric blended implicit ellipses and defines the cloud macrostructure [9]. The final rendered image in Figure 6(c), produced with our volume perturbation technique, shows detail that would be equivalent to unperturbed voxel dataset of at least one hundred times the resolution. Figure 7 demonstrates this technique on another example. By perturbing the volume with a high frequency noise, we can obtain a fur-like surface on the Teddy bear.

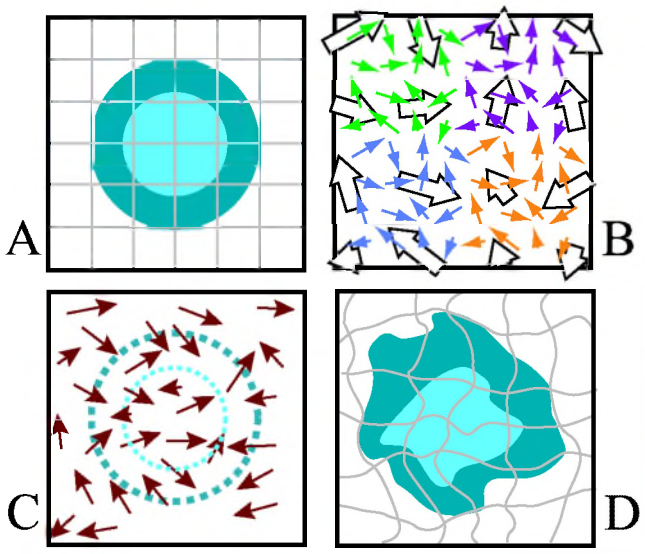


Figure 5: Texture coordinate perturbation in 2D. **A** shows a square polygon mapped with a un-perturbed texture. **B** shows a low resolution vector noise texture applied the polygon multiple times at different scales to achieve low frequency high amplitude offsets (large arrows) and high frequency, low amplitude offsets (small colored arrows). These offset vectors are weighted and summed to offset the original texture coordinates as seen in **C**. The texture is then read using the modified texture coordinates, producing the image seen in **D**.

5 Results and Discussion

We have implemented our volume shading model on both the NVIDIA GeForce 3 and the ATI Radeon 8500/9500. By taking advantage of the OpenGL *render to texture* extension, which allows us to avoid many time consuming copy to texture operations, we have attained frame rates which are only 50 to 60 percent slower than volume rendering with no shading at all. The frame rates for volume shading are comparable to volume rendering with surface shading (e.g. Blinn-Phong shading). Even though surface shading does not require multiple passes on modern graphics hardware, the cost of the additional 3D texture reads for normals induces a considerable performance penalty compared to the 2D texture reads required for our two pass approach. The latest generations of graphics hardware, such as the ATI Radeon 9500, have very flexible fragment shading capabilities, which allow us to implement the entire shading and perturbation pipeline in a single pass.

While our volume shading model is not as accurate as other, more time consuming software approaches, the fact that it is interactive makes it an attractive alternative. Accurate simulations of light transport require material optical properties to be specified in terms of scattering and absorption coefficients. Unfortunately, these values are difficult to acquire. There does not yet exist a comprehensive database of common material optical properties. Interactivity combined with a higher level description of optical properties (e.g. diffuse reflectivity, indirect attenuation, and alpha) allow the user the freedom to explore and create visualizations that achieve a desired effect. Figure 8 (top) demonstrates the familiar appearance

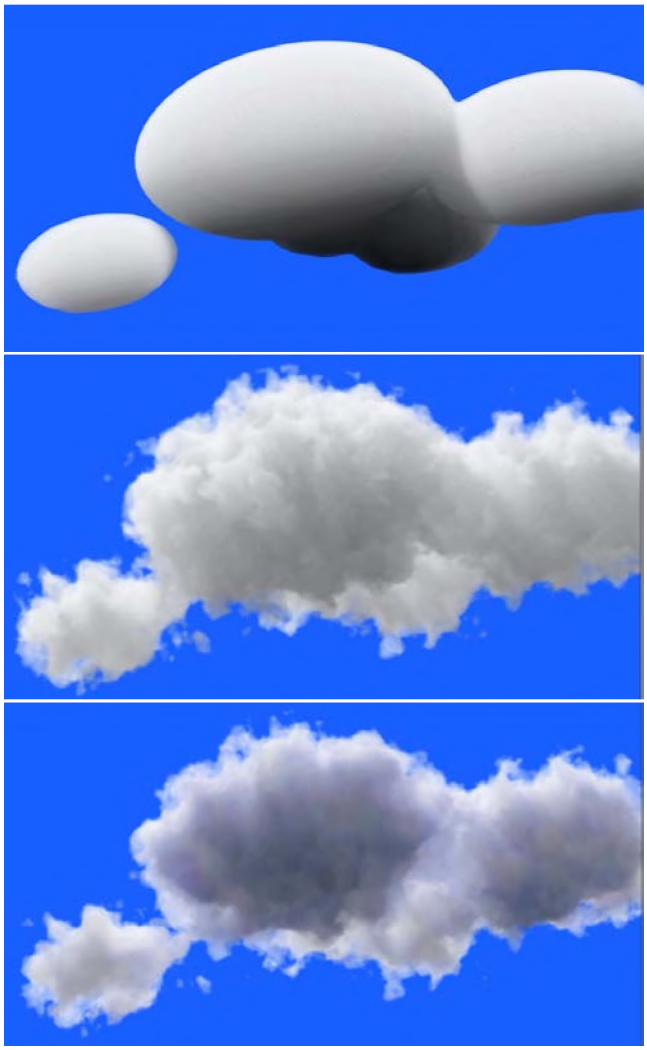


Figure 6: Procedural clouds. The image on the top shows the underlying data, 64^3. The center image shows the perturbed volume. The bottom image shows the perturbed volume lit from behind with low frequency noise added to the indirect attenuation to achieve subtle iridescence effects.

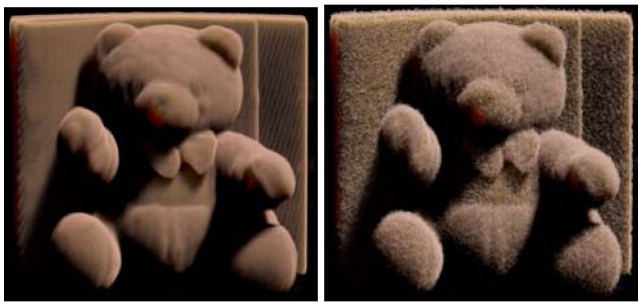


Figure 7: Procedural fur. Left: Original Teddy bear CT scan. Right: Teddy bear with fur created using high frequency texture coordinate perturbation.

Figure 8: The feet of the Visible Female CT.

of skin and tissue. The optical properties for these illustrations were specified quickly (in less than 5 minutes) without using measured optical properties. Even if a user has access to a large collection of optical properties, it may not be clear how to customize them for a specific look. Figure 8 (bottom) demonstrates the effectiveness of our lighting model for scientific visualization.

Our approach is advantageous over previous hardware volume shadow approaches [2, 29, 35] in several ways. First, since this method computes and stores light transport in image space resolution rather than in an additional 3D texture, we avoid an artifact known as attenuation leakage. This can be observed as materials which appear to shadow themselves and blurry shadow boundaries caused by the trilinear interpolation of lighting stored on a coarse grid. Second, even if attenuation leakage is accounted for, volume shading models which only compute direct attenuation (shadows) will produce images which are much darker than intended. These approaches often compensate for this by adding a considerable amount of ambient light to the scene, which may not be desirable. The addition of indirect lighting allows the user to have much more control over the image quality. All of the images in this paper were generated without ambient lighting. Figure 9 compares different lighting models. All of the renderings use the same colormap and alpha values. The image on the upper left is a typical volume rendering with surface shading using the Blinn-Phong shading model. The image on the upper right shows the same volume with only direct lighting, providing volumetric shadows. The image on the lower right uses both direct and indirect lighting. Notice how indirect lighting brightens up the image. The image on the lower left uses direct and indirect lighting combined with surface shading where surface shading is only applied to the leaves using the surface scalar in section 3. Figure 10 shows several examples

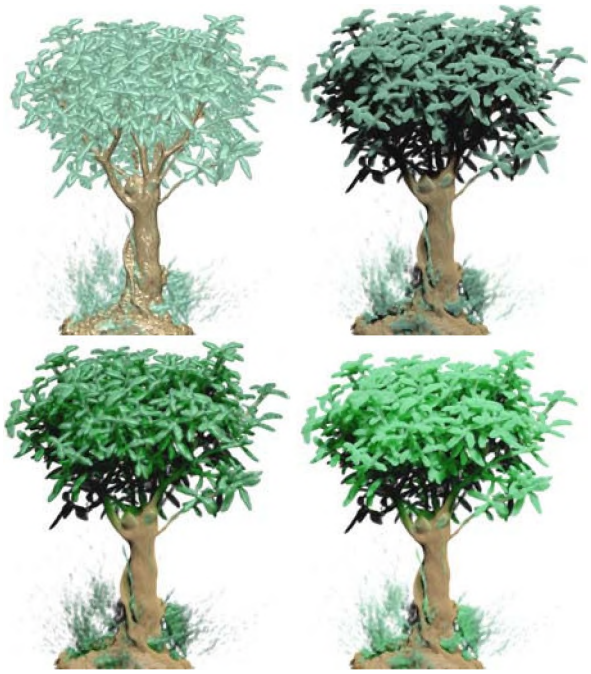


Figure 9: A comparison of shading techniques. Upper left: Surface Shading only, Upper right: Direct lighting only (shadows), Lower right: Direct and indirect lighting, Lower left: Direct and Indirect Lighting with surface shading only on leaves.

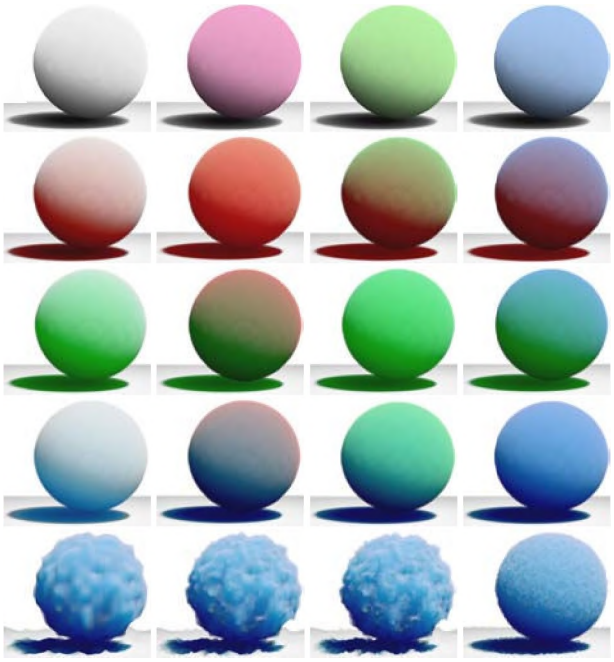


Figure 10: Example material shaders. Rows: Grey, Red, Green, and Blue transport colors respectively. Columns: White, Red, Green, and Blue reflective colors respectively. Bottom row: Different noise frequencies; low, low plus medium, low plus med plus high, and just high frequencies respectively.

of translucent shading. The columns vary the transport color, or the indirect attenuation color, and the rows vary the reflective color, or simply the material color. This illustration demonstrates only a small subset of the shading effects possible with our model.

Procedural volumetric perturbation provides a valuable mechanism for volume modeling effects such as the clouds seen in Figure 6 and for adding high frequency details which may be lost in the model acquisition process, such as the fur of the Teddy bear in Figure 7. Its value in producing realistic effects, however, is largely dependent on the shading. As you can imagine, the clouds in Figure 6 would look like nothing more than deformed blobs with a surface based shading approach. By combining a realistic shading model with the perturbation technique, we can achieve a wide range of interesting visual effects. The importance of having a flexible and expressive shading model for rendering with procedural effects is demonstrated in Figure 11. This example attempts to create a mossy or leafy look on the Visible Male's skull. The upper left image shows the skull with texture coordinate perturbation and no shading. To shade such a perturbed volume with surface shading, one would need to recompute the gradients based upon the perturbed grid. The upper right image adds shadows. While the texture is readily apparent in this image, the lighting is far too dark and harsh for a leafy appearance. The lower right image shows the skull rendered with shadows using a lower alpha value. While the appearance is somewhat brighter, it still lacks the luminous quality of leaves. By adding indirect lighting, as seen in the lower left image, we not only achieve the desired brightness, but we also see the characteristic hue shift of translucent moss or leaves.

Figure 11: The "Chia Skull". A comparison of shading techniques on the Visible Male skull using texture coordinate perturbation. Upper Left: No shading. Upper Right: Shadows. Lower Right: Shadows with a lower opacity skull. Lower Left: Indirect and direct lighting.

6 Future Work

The lighting model presented in this paper was designed to handle volume rendering with little or no restrictions on external lighting,

transfer function, or volume geometry setup. However, if some assumptions can be made, the model can be modified to gain better performance for special purpose situations. We will be exploring extensions of this model that are tailored for specific phenomena or effects, such as clouds, smoke, and skin.

We are also interested in developing more accurate simulations of volumetric light transport that can leverage the expanding performance and features of modern graphics hardware. Such models would be useful for high quality off-line rendering as well as the qualitative and quantitative assessment of our current lighting model, thereby guiding future improvements. As the features of programmable graphics hardware become more flexible and general, we look forward to enhancing our model with effects such as refraction, caustics, back scattering, and global illumination.

Our work with volume perturbation has given us valuable insight into the process of volume modeling. We have been experimenting with approaches for real time volume modeling which do not require any underlying data. We will be developing implicit volume representations and efficient simulations for interactive applications. We are also exploring the use of volume perturbation in the context of uncertainty visualization, where regions of a volume are deformed based on uncertainty or accuracy information.

7 ACKNOWLEDGMENTS

The authors would like to thank Al McPherson from the ACL at LANL for fruitful and provocative discussions about volumetric shadows. This material is based upon work supported by the National Science Foundation under Grants: NSF ACI-0081581, NSF ACI-0121288, NSF IIS-0098443, NSF ACI-9978032, NSF MRI-9977218, NSF ACR-9978099, and the DOE VIEWS program.

REFERENCES

[1] AMBARZUMIAN, V. A. A new method for computing light scattering in turbid media. *Izv. Akad. Nauk SSSR 3* (1942).

[2] BEHRENS, U., AND RATERING, R. Adding Shadows to a Texture-Based Volume Renderer. In *1998 Volume Visualization Symposium* (1998), pp. 39–46.

[3] BLINN, J. F. Light reflection functions for simulation of clouds and dusty surfaces. vol. 16, pp. 21–29.

[4] BOHREN, C. F. Multiple scattering of light and some of its observable consequences. *American Journal of Physics 55*, 6 (June 1987), 524–533.

[5] CABRAL, B., CAM, N., AND FORAN, J. Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. In *ACM Symposium On Volume Visualization* (1994).

[6] CHANDRASEKAR, S. *Radiative Transfer*. Dover, New York, 1960.

[7] C.REZK-SALAMA, K.ENGEL, BAUER, M., GREINER, G., AND ERTL, T. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. In *Siggraph/Eurographics Workshop on Graphics Hardware 2000* (2000).

[8] DORSEY, J., EDELMAN, A., JENSEN, H. W., LEGAKIS, J., AND PEDERSEN, H. Modeling and rendering of weathered stone. In *Proceedings of SIGGRAPH 1999* (August 1999), pp. 225–234.

[9] EBERT, D., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., AND WORLEY, S. *Texturing and Modeling: A Procedural Approach*. Academic Press, July 1998.

[10] FARRELL, T. J., PATTERSON, M. S., AND WILSON, B. C. A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the non-invasive determination of tissue optical properties in vivo. *Medical Physics 19* (1992), 879–888.

[11] HANRAHAN, P., AND KRUEGER, W. Reflection from layered surfaces due to subsurface scattering. In *Computer Graphics (SIGGRAPH '93 Proceedings)* (Aug. 1993), J. T. Kajiya, Ed., vol. 27, pp. 165–174.

[12] JENSEN, H. W., AND CHRISTENSEN, P. H. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of SIGGRAPH 98* (Orlando, Florida, July 1998), Computer Graphics Proceedings, Annual Conference Series, pp. 311–320.

[13] JENSEN, H. W., MARSCHNER, S. R., LEVOY, M., AND HANRAHAN, P. A practical model for subsurface light transport. In *Proceedings of SIGGRAPH 2001* (August 2001), Computer Graphics Proceedings, Annual Conference Series, pp. 511–518.

[14] KAJIYA, J. T. The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)* (Aug. 1986), D. C. Evans and R. J. Athay, Eds., vol. 20, pp. 143–150.

[15] KAJIYA, J. T., AND KAY, T. L. Rendering fur with three dimensional textures. In *Computer Graphics (Proceedings of SIGGRAPH 89)* (Boston, Massachusetts, July 1989), vol. 23, pp. 271–280.

[16] KAJIYA, J. T., AND VON HERZEN, B. P. Ray tracing volume densities. In *Computer Graphics (SIGGRAPH '84 Proceedings)* (July 1984), H. Christiansen, Ed., vol. 18, pp. 165–174.

[17] KEIJZR, M., STAR, W. M., AND STORCHI, P. R. M. Optical diffusion in layered media. *Applied Optics 27*, 9 (1988), 1820–1824.

[18] KNISS, J., KINDLMANN, G., AND HANSEN, C. Multi-Dimensional Transfer Functions for Interactive Volume Rendering. *TVCG* (2002 to appear).

[19] KRUEGER, W. The application of transport theory to visualization of 3-D scalar data fields. *Computers in Physics* (Jul/Aug 1991), 397–406.

[20] LACROUTE, P., AND LEVOY, M. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform. In *ACM Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), pp. 451–458.

[21] LANGUENOU, E., BOUATOUCH, K., AND CHELLE, M. Global illumination in presence of participating media with general properties. In *Fifth Eurographics Workshop on Rendering* (Darmstadt, Germany, June 1994), pp. 69–85.

[22] LEVOY, M. Display of surfaces from volume data. *IEEE Computer Graphics & Applications 8*, 5 (1988), 29–37.

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

[24] MAX, N., MOBLEY, C., KEATING, B., AND WU, E. Plane-parallel radiance transport for global illumination in vegetation. In *Eurographics Rendering Workshop 1997* (New York City, NY, June 1997), J. Dorsey and P. Slusallek, Eds., Eurographics, Springer Wien, pp. 239–250. ISBN 3-211-83001-4.

[25] NEYRET, F. Modeling, animating, and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics 4*, 1 (January-March 1998), 55–70. ISSN 1077-2626.

[26] NISHITA, T. Light scattering models for the realistic rendering. In *Proceedings of the Eurographics* (1998), pp. 1–10.

[27] NISHITA, T., NAKAMAE, E., AND DOBASHI, Y. Display of clouds and snow taking into account multiple anisotropic scattering and sky light. In *SIGGRAPH 96 Conference Proceedings* (Aug. 1996), H. Rushmeier, Ed., Annual Conference Series, ACM SIGGRAPH, Addison Wesley, pp. 379–386. held in New Orleans, Louisiana, 04-09 August 1996.

[28] NOORDMANS, H. J., VAN DER VOORT, H. T., AND SMEULDERS, A. W. Spectral Volume Rendering. In *IEEE Transactions on Visualization and Computer Graphics* (July-September 2000), vol. 6, IEEE.

[29] NULKAR, M., AND MUELLER, K. Splatting With Shadows. In *Volume Graphics 2001* (2001), pp. 35–49.

[30] PERLIN, K. An image synthesizer. In *Computer Graphics (Proceedings of SIGGRAPH 85)* (San Francisco, California, July 1985), vol. 19, pp. 287–296.

[31] PERLIN, K., AND HOFFERT, E. M. Hypertexture. In *Computer Graphics (SIGGRAPH '89 Proceedings)* (July 1989), J. Lane, Ed., vol. 23, pp. 253–262.

[32] PHARR, M., AND HANRAHAN, P. M. Monte carlo evaluation of non-linear scattering equations for subsurface reflection. In *Proceedings of SIGGRAPH 2000* (July 2000), Computer Graphics Proceedings, Annual Conference Series, pp. 75–84.

[33] RUSHMEIER, H. E. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. Ph.d. thesis, Cornell University, 1988.

[34] RUSHMEIER, H. E., AND TORRANCE, K. E. The zonal method for calculating light intensities in the presence of a participating medium. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (July 1987), M. C. Stone, Ed., vol. 21, pp. 293–302.

[35] STAM, J. Stable Fluids. In *Siggraph 99* (1999), pp. 121–128.

[36] VAN DE HULST, H. *Multiple Light Scattering*. Academic Press, New York, NY, 1980.

[37] WANG, L. V. Rapid modelling of diffuse reflectance of light in turbid slabs. *J. Opt. Soc. Am. A 15*, 4 (1998), 936–944.

[38] WILSON, O., GELDER, A. V., AND WILHELMS, J. Direct Volume Rendering via 3D Textures. Tech. Rep. UCSC-CRL-94-19, University of California at Santa Cruz, June 1994.