

# Lighting Transfer Functions Using Gradient Aligned Sampling

Eric B. Lum

Kwan-Liu Ma

University of California Davis

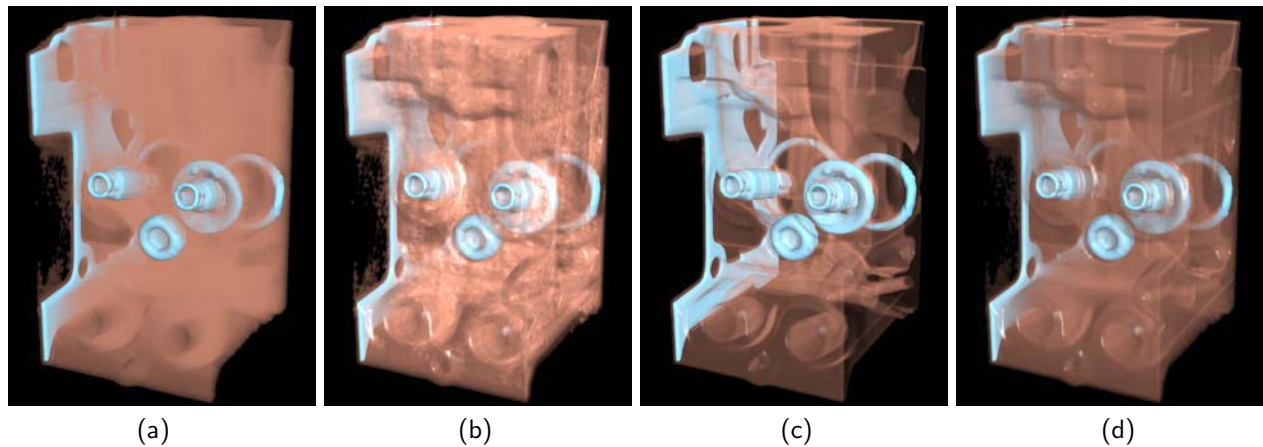


Figure 1: With the absence of lighting in the brown region in (a), very few of the inner structures are visible. With the addition of shading in (b) these structures become more visible, but erratic variations of lighting in near homogeneous regions make the larger boundaries of interest difficult to see. By increasing the opacity at these boundaries in (c) the boundary surface are clearly visible, however, the sense of material thickness from absorption in homogeneous regions is lost. With our technique shown in (d), lighting transfer functions are used to selectively enhance boundary surfaces of interest, allowing opacity to be used to illustrate material thickness and depth. Figures (a), (b), and (d) share the same color and opacity transfer function and have variations only in their lighting transfer function.

## ABSTRACT

An important task in volume rendering is the visualization of boundaries between materials. This is typically accomplished using transfer functions that increase opacity based on a voxel's value and gradient. Lighting also plays a crucial role in illustrating surfaces. In this paper we present a multi-dimensional transfer function method for enhancing surfaces, not through the variation of opacity, but through the modification of surface shading. The technique uses a lighting transfer function that takes into account the distribution of values along a material boundary and features a novel interface for visualizing and specifying these transfer functions. With our method, the user is given a means of visualizing boundaries without modifying opacity, allowing opacity to be used for illustrating the thickness of homogeneous materials through the absorption of light.

**CR Categories:** I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.7 [Computer Graphics]: Three-Dimensional graphics and realism—color, shading, and texture

**Keywords:** direct volume rendering, volume visualization, multi-dimensional transfer functions, shading, transfer functions

## 1 INTRODUCTION

Much of the expressiveness of direct volume rendering comes from the transfer function, which refers to the mapping between data val-

ues and visual attributes used for rendering. The ability to control this mapping allows a scientist to add emphasis to those materials of interest, while removing those materials that could obscure or detract from the visualization. Often, it is desirable to enhance the boundaries between materials in a volume, and methods have been developed which extend the number of transfer function inputs to include neighboring voxel's characteristics. These methods have used the extended domain of the transfer function to more easily specify the *opacity* of material boundaries. The appearance of a surface however, is not only a function of the material color and opacity, but also the reflection of light. Lighting plays a critically important role in illustrating surfaces; in particular, lighting variations provide visual cues regarding surface orientation.

Most research in the use of higher dimensional transfer functions has focussed on the specification of color and opacity, despite the fact that transfer functions can be used for specifying of any of a number of optical properties [6] including surface illumination. In this paper, we present a novel method for specifying the appearance of surface illumination using a higher-dimensional transfer function, with an input domain consisting of samples read along the gradient direction. The technique provides users with precise control over how lighting is used to illustrate the different material boundaries in a volume. In addition, we discuss a novel visual interface for specifying lighting transfer functions that is designed to improve the user's intuition as to where material transitions occur with respect to scalar value.

In the real world, lit surfaces are typically seen where there is a transition between transparent and opaque material, such as a solid object in air. However, lit surfaces can also appear at the boundary between two similarly transparent materials such as at the boundary between glass and air. Our method allows the user to position lighting at surfaces independent of the opacity at a surface. This can be contrasted with previously developed methods that enhance

the appearance of surfaces through the manipulation of opacity. By avoiding the use of opacity for illustrating surfaces, opacity-based attenuation can be reserved for indicating material thickness and depth. Furthermore, our method can be used to illustrate the surfaces of materials of extremely low opacity through lighting. The use of such low opacity permits underlying material to be less obscured by light absorption. An example of an engine rendered with our technique is shown in Figure 1(d).

In this paper we also describe how the method can be used to give a high degree of control over the lighting of near-homogeneous materials in both pre- and post-filtered data space. We discuss how our technique can make use of a hybrid between pre-filtered and post-filtered gradient calculations, and how it is able to take advantage of the favorable attributes of both methods. Finally, we describe an efficient hardware implementation.

Our method takes advantage of lighting characteristics that are not accounted for in the conventional model used in direct volume rendering, namely, the variation of surface reflectivity independent of opacity. Our method was not designed with the goal of creating physically realistic images, but to incorporate additional lighting principals in a manner that can be easily controlled by the user to enhance the appearance of those materials of most interest.

## 2 BACKGROUND

The seminal works of Levoy [12] and Drebin et al. [2] describe methods for the direct volume rendering of scientific data. Although sharing many similarities, these two works have several subtle but important differences with regard to how shading is performed. Levoy argues one of the key features of direct volume rendering is that unlike polygon isosurface rendering, shading can be performed independent of classification. He makes use of the Phong shading model with normals computed directly from the gradient of the scalar field using central differences. In scientific visualization, direct volume rendering illumination is typically performed in this manner. Levoy also describes the enhancement of boundary surfaces by making them more opaque depending on the magnitude of the gradient vector.

Drebin et al. [2] map voxel scalar values to density values that are computed as a result of classification, but are separate from opacity. In contrast to the method described by Levoy, the normals used for lighting are computed from the gradient of scalar values *after* applying the density table lookup. Thus they make use of post-classified gradients rather than pre-classified gradients. Further, they describe the modulation of lighting based on gradient, which has the effect of not illuminating homogeneous materials in post-classified data space.

The computation of gradients from the derived density volume as described by Drebin et al. has the advantage of more closely following the physical interaction of light in a volume density, but requires the recomputing of gradients when the classification function is changed. The computation of gradients prior to classification, on the other hand, yields normal directions that are more closely tied to the original data, and are not influenced by potential errors in classification. Furthermore, these normals can be generated in a preprocessing step, and do not need to be recomputed if the classification function is changed. The fact that these gradients do not necessarily match the densities used during rendering, however, can yield normals that are oriented in directions contrary to the densities used during rendering, particularly when regions of high data value are mapped to low opacity.

The particle models used in volume rendering make the assumption that a volume consists of a distribution of small, *fully opaque*, particles of varying densities. Max [14] provides an in-depth tutorial on this model and its assumptions. The alpha (opacity) term used in the volume rendering can be thought of as the probability

that a photon collides with a particle for some unit depth. This explains the  $\alpha$  and  $(1 - \alpha)$  terms in the discrete back-to-front volume rendering compositing relation

$$C_{out} = \alpha C + (1 - \alpha)C_{in}$$

where  $C_{out}$  is the color of the ray leaving a sample location,  $C_{in}$  is the color entering the sample location,  $C$  is the color of the sample location, and  $\alpha$  is that sample's opacity.

When a ray is cast through a volume, some portion of light  $\alpha$  collides with particles, yielding color from scattering. The remaining light  $(1 - \alpha)$  continues through the volume, revealing additional material. The fact that alpha appears in both terms of this equation is a direct consequence of using a model that only accounts for fully opaque particles. This restricts how opacity modifies the appearance of material in volume visualization. For example, increasing the opacity at a material's surface to increase surface visibility might have the undesirable effect of also obscuring the material behind it. Furthermore, increasing the opacity at a material's boundary is often not physically valid. A large class of transparent or semitransparent materials such as glass and water have visible surfaces, not because their surfaces are opaque, but because of the increased reflectivity at their material boundaries. Surface reflectivity results from a variation in the index of refraction as approximated by the Fresnel equations. A variation in the index refraction, although related to material density, can occur independent of a variation in light obstruction.

Enhancing opacity at material boundaries is analogous to rendering frosted glass. And while it can provide a powerful means of viewing surfaces, it also limits how opacity can be used. The method we present uses a second two-dimensional lighting transfer function to decouple the specification of the light emitted at a sample location through scattering from the occlusion of light from further materials through absorption.

## 3 RELATED WORK

Kindlmann and Durkin [7] describe joint 2D scalar/gradient magnitude histograms for the semi-automatic specification of transfer functions that enhance surface opacity based on scalar value and gradient magnitude. In a later work, Kniss et al. [9] present a widget-based interface for the intuitive specification of higher-dimensional opacity/color transfer functions and describe how these higher-dimensional transfer functions can be implemented efficiently in graphics hardware. They also describe the rendering of shadows by accumulating attenuated light in an off-screen rendering buffer. They avoid the shading of homogeneous material using scalar gradient magnitude. Our work differs from theirs in that their work focused on giving users a high degree of control over opacity, with efficient conventional lighting. Our work focuses exclusively on giving users expressive control over lighting, with a new user interface geared for specifying the shading for different material boundaries.

Much work has been done on using more sophisticated volumetric lighting models for more realistic rendering of natural phenomena [1, 5, 13, 18, 4, 15]. In addition, some work has been done in applying more realistic models for visualizations. Krueger [11] applies transport theory to volume rendering using a simulation of particles that have various lighting interactions with a data set. Rodgman and Chen [17] allow for transfer functions that include index of refraction to model the effect of the bending of light due to refraction. Noordmans et al. [16] model spectral changes in the color of light as it interacts with material and allow for chromatic opacity, rather than a single scalar alpha. Kniss et al. [10] use graphics hardware to efficiently generate highly realistic renderings that model the effects of volumetric shadows, forward scattering, and chromatic attenuation. In contrast, the primary goal of our work

is not to create more realistic images, but to provide control over the use lighting to generate imagery better suited for illustrating the specific structures of interest to scientists.

Kindlmann and Weinstein [8] use illumination as a means of illustrating tensor fields. In addition to the use of color and opacity transfer functions, they describe the idea of using lit-tensors to indicate the type and orientation of anisotropy. Hauser et al [3] describe a multi-level volume rendering method that simultaneously uses different rendering techniques, such as direct volume rendering and maximum intensity projection, to illustrate different objects in volumetric data set.

#### 4 LIGHTING TRANSFER FUNCTIONS

The higher-dimensional transfer function user interface described by Kniss et al. [9] uses transfer function widgets shown above a joint 2D scalar value/gradient magnitude histogram. This type of 2D histogram for the engine data set in Figure 1 is shown in Figure 2(a) with the horizontal axis used for scalar value and the vertical value for gradient magnitude. The low gradient groupings correspond to different homogeneous regions in the volume. As described by Kindlmann and Durkin [7], the arches between these groups are the boundaries between the different materials in a volume. In our work, we would like to give the user the ability to enhance the boundary between any pair of materials, which corresponds to an arch on this joint 2D histogram. This is difficult since the arches intersect, a phenomenon that occurs frequently in the regions of low gradient above homogeneous regions on the 2D histogram.

Kindlmann and Durkin use the second derivative along the gradient direction to further disambiguate boundaries since an idealized boundary has a high gradient and zero second derivative. The accuracy of the calculated second derivative, however, is highly susceptible to noise. Furthermore, the combined scalar value, gradient magnitude, and second derivative along the gradient direction do not provide a direct means of distinguishing the different types of material boundaries in a volume. With this in mind, we make use of a lighting transfer function that takes into account data values perpendicular to a possible surface at a sample, which is accomplished by sampling points along a sample's normalized gradient direction. As illustrated in Figure 3, for each sample two additional scalar values are used: one in the direction of the gradient and one in the opposite direction. The results are three samples perpendicular to the direction of a possible boundary surface. The center sample is used with a 1D scalar transfer function for the assignment of material color and opacity. The two samples along the gradient, above and below, are used as input to a two-dimensional lighting transfer function for the assignment of ambient, specular and diffuse lighting coefficients.

With shading, the color of a rendered sample can be expressed as

$$C = C_{\text{voxel}}(I_a + I_d) + I_s$$

where  $I_a$ ,  $I_d$ , and  $I_s$  are the ambient, diffuse, and specular illumination of that sample. When a transfer function colormap is used along with the Phong shading model, this can be expanded to

$$C = \text{ColorTF}(S)(k_a + k_d \text{MAX}(N \cdot L, 0)) + k_s \text{MAX}((N \cdot R)^n, 0)$$

where  $S$  is the sample's scalar value,  $\text{ColorTF}()$  is the transfer function colormap lookup table,  $N$  is the normalized gradient direction,  $L$  is the light direction,  $R$  is the reflected light direction,  $n$  is a specular "shininess" exponent, and  $k_a$ ,  $k_d$  and  $k_s$  are the ambient, diffuse, and specular lighting coefficients respectively. With our lighting transfer function method, the lighting coefficients are replaced with lookup tables that are functions of the two scalar values,  $S_1$  and  $S_2$ ,

read along the gradient direction. The color of a rendered sample therefore becomes

$$C = \text{ColorTF}(S)(\text{LTF}_{k_a}(S_1, S_2) + \text{LTF}_{k_d}(S_1, S_2) \text{MAX}(N \cdot L, 0) + \text{LTF}_{k_s}(S_1, S_2) \text{MAX}((N \cdot R)^n, 0))$$

where  $\text{LTF}_{k_a}()$ ,  $\text{LTF}_{k_d}()$  and  $\text{LTF}_{k_s}()$  are lookup tables for the ambient, diffuse and specular lighting coefficients respectively.

The two samples read along the gradient direction are well-suited inputs for our lighting transfer functions since they provide an indication of whether a material boundary occurs at a given sample and which materials exist on each side of that boundary. For example, if the two samples have the same value, then the center sample position is likely to be in a homogeneous region. Furthermore, if the scalar values above and below the current sample belong to two different materials, then the sample is likely at a material boundary. Intuition as to why this approach works can be gained by further considering the nature of the 2D histogram material boundary arches described by Kindlmann and Weinstein. In their work, they define material boundaries as the finitely thin transitional regions between homogeneous materials and explain that due to the band-limited nature of most data acquisition systems, a reconstructed volume will always contain a degree of blurring. They demonstrate that if a volume is reconstructed with a Gaussian reconstruction kernel, the transitions between idealized homogeneous regions have highest gradient values at the center of the boundary transition and diminish to zero moving away from the boundary. They illustrate this is true in practice with a variety of data sets that have 2D joint scalar/gradient magnitude histograms containing distinct arches between scalar values corresponding to homogeneous regions. For a boundary between two materials, an arch starts with reduced gradient magnitude in a homogeneous region, increases to a higher gradient magnitude between the two materials, and then recedes moving toward the other homogeneous region.

Rather than relying on scalar value and gradient to identify material boundary transitions, our method uses two scalar values read on both sides of a rendered sample along the gradient direction. This provides a more direct means of selecting the various boundaries in a volume by essentially following the arches and reading the homogeneous scalar values on both sides of a material boundary. Consider the case where the two supplemental scalar samples are separated by a distance greater than the thickness of a material boundary. In this case, if the gradient direction is perpendicular to the material boundary, the scalar value pair will have the same values throughout a material boundary, with their values equal to the scalar values on each side of the boundary. By following the gradient direction, this method is able to read from within the adjacent homogeneous regions for the classification of these transitions. The technique is sensitive to the thickness of a material boundary. Since the scalar value pairs are read during rendering time, the distance between sample pairs is a parameter that can be adjusted depending on the amount of blur present in a volume. For the data sets used in our work, we found reading a distance of one voxel in each direction to be sufficient.

#### 4.1 User Interface

Displaying a histogram of a transfer function's domain can provide insight into the structures found in the volume and make transfer function specification more intuitive. Unfortunately, plotting the scalar values pairs read along the gradient direction yields the rather un-intuitive 2D histogram shown in Figure 2(b). Interpreting this histogram is made more difficult by the fact that both the vertical and horizontal axis correspond to the same type of data, namely scalar value. This can be contrasted with a scalar/gradient magnitude histogram which plots two distinctly different properties. The band along the diagonal corresponds to homogeneous

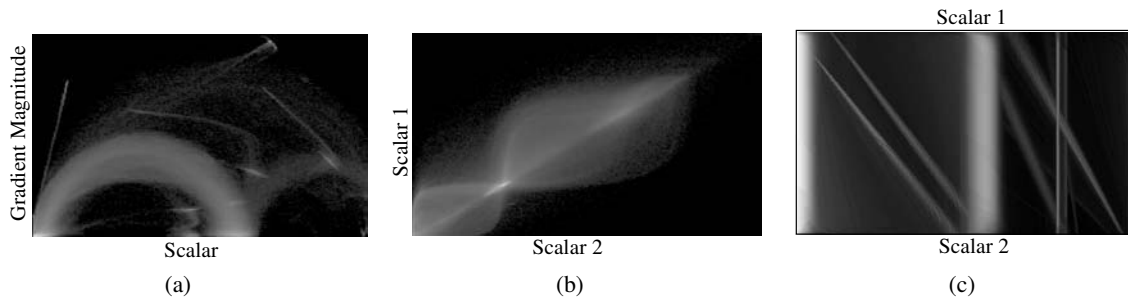


Figure 2: In (a), the joint scalar value/gradient magnitude 2D histogram of the engine data set is shown. The joint 2D histogram of scalar values pairs sampled along the gradient direction is shown in (b). The final image contains the line-based histogram of scalar value pairs used in our work. The vertical bands show homogeneous regions, while the diagonal lines indicate material boundaries.

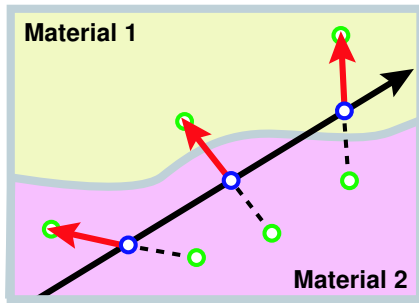


Figure 3: The two-dimensional lighting transfer functions in our work use as input scalar value pairs sampled along the gradient direction as shown in green. The center voxel samples used for the color and opacity transfer function are shown in blue, while the gradient direction is illustrated in red. The black line illustrates a ray being cast through a volume.

material, while vertical deviations illustrate boundaries. The fact that this vertical deviation is not perpendicular to the diagonal axis makes it difficult to get a sense for the strength of this deviation. Adding to the confusion, deviation in both the vertical and horizontal directions corresponds to increases in scalar value.

Recall that we would like to construct 2D transfer functions that allow the user to specify lighting for boundaries between materials. We would thus like to display the 2D histogram in a manner that makes the boundary relationships between materials immediately visible. To accomplish this, we have developed a method for displaying these 2D histograms using lines rather than the conventional point primitives. As shown in Figure 2(c), our line-based histogram makes use of two parallel axis for the two scalar values. The bottom axis corresponds to the scalar value read in the gradient direction, and increases from left to right. The top axis is used to plot the scalar sampled opposite to the gradient direction. For each pair of values in the 2D histogram a line is drawn from the top to the bottom axis with an intensity that increases with the number of such transitions in the data set. The resulting histogram gives an intuitive means of understanding the material boundaries found in a volume. Dense vertical lines represent homogeneous regions in a volume, while diagonal lines illustrate the boundaries between two materials. The start and end of these lines indicate the scalar values of the different materials at that boundary.

Specifying the lighting transfer function then becomes the task of assigning lighting coefficients to the different lines in a volume. We accomplish this using pairs of 1D transfer function widgets for the two axes. As shown in Figure 4 the user specifies a reflectivity distribution for both the source of the lines at the top, and the des-

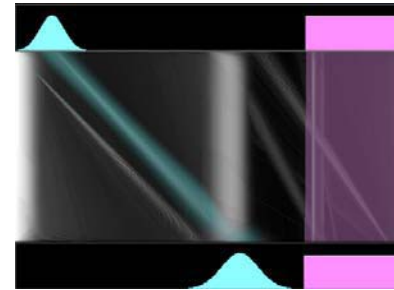


Figure 4: The user interface for our lighting transfer functions consists of widgets for specifying pairs of 1D transfer functions that indicate the intensity range for each side of a material boundary of interest. The selected boundary ranges are shown above the line-based boundary histogram.

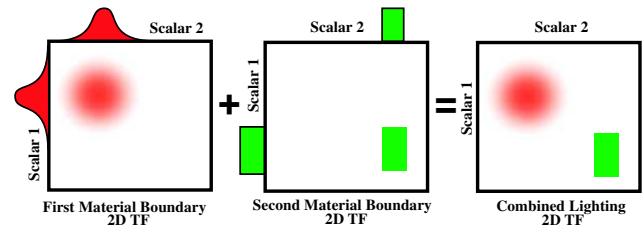


Figure 5: The lighting transfer function(TF) is specified using a pair of 1D transfer functions that are combined to create the separable 2D transfer functions shown below. These 2D transfer functions are summed to create the final 2D transfer function used for rendering.

tinuation of the lines at the bottom. The user also assigns ambient, diffuse, and specular terms for each widget that the transfer function will be multiplied by. As illustrated in Figure 5, for each pair of transfer function widgets, the product of one of the 1D transfer functions with the transpose of the other yields a separable 2D lighting transfer function. The sum of several of these 2D transfer functions yields the final 2D transfer function used for rendering, which is not separable.

This user interface gives a high degree control over how lighting is used to illustrate a volume and is independent of opacity as shown in Figure 6. The user can illustrate the surface between any pair of materials using lighting by selecting that material diagonal pair using this transfer function interface. Similarly, the user is given control over the lighting of near-homogeneous regions in the volume. Kniss et al. [9] attenuate surface lighting of regions that are homogeneous with respect to the original scalar data based on

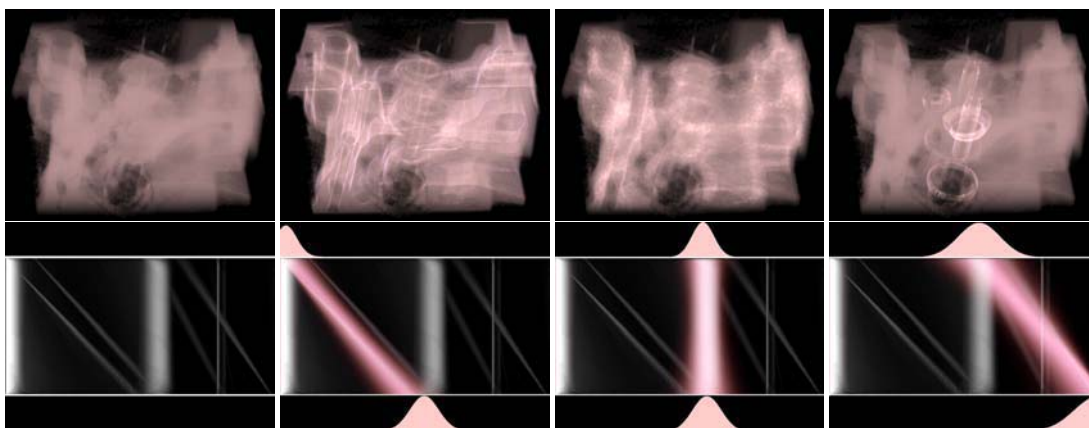


Figure 6: These examples illustrate how different material boundaries can be selected with our user interface and lighting transfer functions.

the gradient magnitude of that scalar data. Drebin et al.[2] attenuate surface lighting based on the gradient magnitude of post-classified density. With our technique, the user can quickly specify a lighting transfer function that attenuates using either measure of homogeneity. By not selecting vertical lines in the lighting transfer function, the user can attenuate surface lighting of homogeneous material in data space. By not specifying lines that link regions with similar opacity, attenuation of lighting based on post-classification can be performed. It should be emphasized that the main power of our technique lies not in its control over attenuating light in homogeneous regions, but rather in its use in enhancing lighting at those key boundaries of interest. These boundaries can even occur at subtle variations in scalar values that are near homogeneous in both scalar and classified data space.

## 4.2 Hardware Implementation

We implemented our technique in graphics hardware using view-aligned textured polygons [19]. The additional rendering computation is implemented using a fragment program. During rendering, each sample's scalar value and gradient are read. The gradient is normalized, and the pair of additional scalar values along the gradient direction are read. The center scalar value is used for a conventional opacity/color transfer function lookup, while the scalar value pair read along the gradient are used as texture indices into the 2D lighting transfer function. These lighting coefficients are used to modulate specular and diffuse lighting read from an environment map.

The fact that our method uses lighting to illustrate highly translucent material requires slight modifications to accommodate higher dynamic range lighting. Specifically, since the surfaces rendered with our method do not have enhanced opacity, it is often necessary that the alpha multiplied RGB color at a surface sample exceed its alpha value. The unmodulated color for a voxel must therefore be larger than one, the intensity often treated as brightest for hardware implementations. A light scaling factor that can be larger than one is therefore passed to the fragment program that modulates all light intensities. Compositing RGB colors larger than alpha also requires alpha modulation of the source color to be done in the fragment program rather than in the framebuffer blending stage. It is physically valid to composite colors that are brighter than the alpha value. Reflective water and flames are two physical phenomena that can reflect or emit large amounts despite their low absorption.

As discussed previously, gradient directions used for lighting are typically computed from the unclassified scalar data and not post-classified density as described by Drebin et al. [2]. Pre-classified gradients are more closely tied with the original data and can be

precomputed but can result in directions that point opposite to the classified opacity distribution. We use the additional scalar values read along the gradient direction to implement a hybrid between the two techniques. First, the gradients of the original scalar field are precomputed. The two additional scalar values are read along this linearly interpolated normalized direction. Then, based on the sign of the difference between the post-classified sample densities, the direction of that gradient is adaptively flipped to point in the direction of maximum variation in post-classified density. This difference is precomputed and stored in the alpha channel of the same 2D lookup table used for the lighting transfer function. The gradient directions computed with this method are not equivalent to post-classified gradients since pre-classified and post-classified gradients are rarely parallel. Figure 7(a) shows a tooth rendered with standard pre-classified gradients and single-sided lighting. The dentin structure is not lit since it is a region of relatively low scalar value, resulting in normals oriented inward. In Figure 7(b) the same normals are used with two-sided lighting. The dentin is properly lit, but some back-facing surfaces are lit as well, as indicated by the arrow. In Figure 7(c) single-sided lighting is used with our hybrid gradient computation technique. The dentin is lit with normals that have been flipped away from its surfaces, while the back-facing surfaces are not lit.

## 5 RESULTS

We experimented with our method using a wide variety of volumetric data sets generated from both measurement and numerical simulation. Figure 1 contains images generated from a CT scan of an engine. Notice in Figure 1(a) that with uniform opacity and the absence of lighting in the brown region, very few of the inner structures are visible. With the addition of lighting in Figure 1(b) the surface structures become more visible, but the erratic variations of lighting in near homogeneous regions make these boundaries more difficult to see. By increasing the opacity at these boundaries and reducing the opacity in the homogeneous regions in Figure 1(c), the boundary surface are clearly visible. However, the sense of material thickness from the uniform absorption of light in homogeneous regions is lost. Our lighting transfer function technique is illustrated in Figure 1(d), which uses the same opacity and color transfer function as (a) and (b). Notice how lighting is used to selectively enhance boundary surfaces of interest, while uniform opacity is used to illustrate material thickness.

In Figure 8 an electron probability distribution of a protein molecule is shown. In Figure 8(a) lighting is not used. The blue and yellow regions have the same uniform opacity, which allows



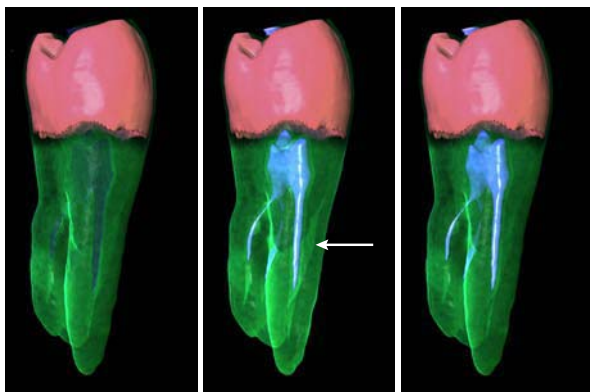


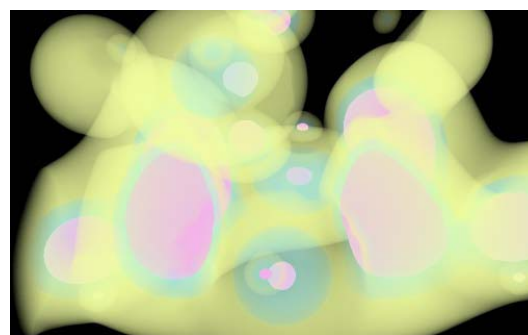
Figure 7: A hybrid between pre-classified and post-classified gradients is used to avoid the need for two-sided lighting. On the left, the dentin structure is not lit since its gradient direction is oriented toward its surface. The middle images was rendered with two-sided lighting which illuminates the dentin structure, but also lights back facing surfaces as indicated by the arrow. On the right, our hybrid normal-computation technique is used.

opacity attenuation to give a strong indication of the amount of material within the various structures. The absence of lighting makes it difficult to get a sense for the shape of the actual structures. The uniform application of shading in Figure 8(b) aids slightly in showing structure orientations, but is at time unclear from the uniform shading of homogeneous regions. In Figure 8(c) the use of opacity modulation at select density transitions communicates the shape of the spatial structures but loses many of the cues from opacity occlusion in thick regions. In (d) the volume is rendered with a lighting transfer function that illustrates several density transitions with lighting, while opacity is used to show depth-based occlusion. In particular, notice how lighting is used to make the boundary between blue and yellow material visible despite the uniform opacity at that transition. This boundary has been placed at a region of homogeneous opacity and relatively homogeneous scalar value.

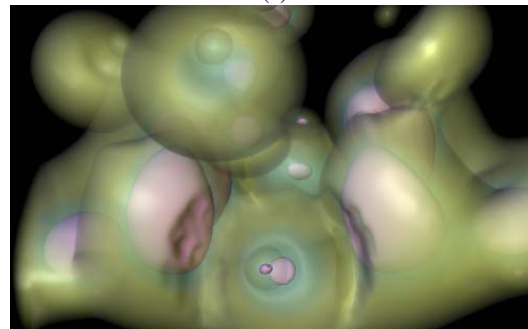
The left image of Figure 9 shows the use of opacity to illustrate a surface in a simulated vortex flow data set. The right rendering was generated using lighting transfer functions. Notice that since opacity is not modulated at the surface, the thickness of the pink material is made evident by the amount it occludes material behind it.

Figure 10 shows how lighting transfer functions can be used to illustrate surfaces independent of opacity with a data set consisting of a simulation of fuel injection into a combustion chamber. On the left, the green surface is fully opaque and clearly visible. On the right, the green region is rendered with extremely low opacity, but the shape of the boundary remains visible through the use of lighting. The low opacity of the green regions makes the underlying orange structure clearly visible. The middle image shows the volume rendered with intermediate opacity.

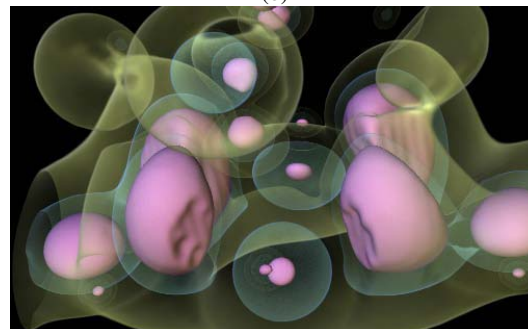
There are approximately twice as many texture reads required for our method compared to traditional opacity-based transfer function volume rendering methods. With an Nvidia Geforce 6800 graphics card rendering a  $256 \times 256 \times 256$  volume to a  $512 \times 512$  buffer with rendered sample spacing distance equal to that of the original volume, rendering occurs at approximately 7.3 frames per second versus 13.5 frames per second with uniform lighting. The frame rate is sufficient to allow interactive exploration of the rendering parameter space for the quick specification of a lighting transfer function to highlight boundaries of interest.



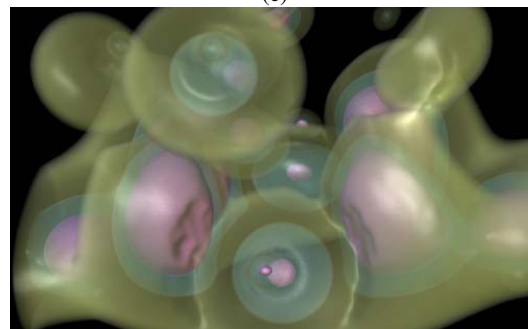
(a)



(b)



(c)



(d)

Figure 8: In (a), an electron probability distribution of a protein molecule is rendered without lighting. The uniform application of lighting in (b) aids slightly in showing structure orientations, but can add confusion from the illumination of homogeneous regions. Use of opacity modulation at select density transitions in (c) communicates the shape of the spatial structures, but loses many of the cues from opacity occlusions in thick regions. In (d) the volume is rendered with a lighting transfer function that illustrates a couple of density transitions with lighting, while opacity is used to show depth based occlusion.

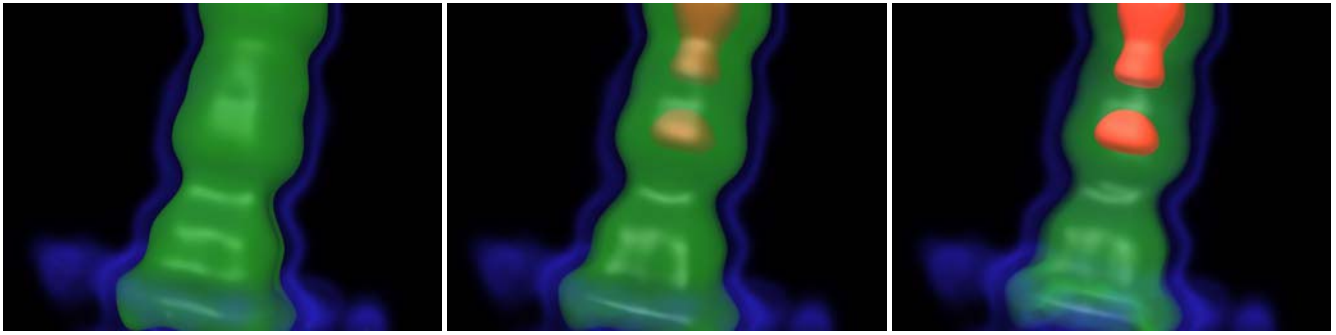


Figure 10: With our technique, lighting parameters can be manipulated independent of opacity. Notice that despite variation in opacity in the three images, the surface in green remains visible due to surface shading.

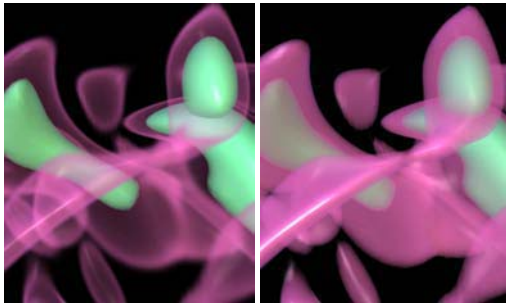


Figure 9: The left image of the vortex uses opacity to illustrate a surface. The right rendering was generated using lighting transfer functions. Notice that since opacity is not modulated at the surface, the thickness of the pink material is made evident by the amount it occludes material behind it.

## 6 DISCUSSION

This paper presents a method for the manipulation of surface lighting parameters to enhance material boundaries of interest. The manipulation of surface shading at arbitrary scalar value transitions is physically justifiable by the fact that reflections can exist at surfaces with homogeneous opacity or density because of variations in the index of refraction. However, the method can enhance the lighting at surfaces in a manner that is not physically realistic. We believe this is not a limitation of the work. Much of scientific visualization is far from photorealistic. For example, the assignment of color and opacity to a temperature or velocity field departs far from realism, but meets the goals of scientific visualization: the creation of imagery that helps illustrate scientific phenomena. Similarly, we believe the utility of the method we have developed lies not in necessarily being more physically valid, but rather in giving scientists the freedom to enhance lighting in a manner that gives greater control over the appearance of material boundaries.

As is the case with all methods that make use of gradients in direct volume rendering, our method depends on the ability to compute gradients in a volume with reasonable accuracy. Our method is thus less effective for volumes with a high amount of noise. The use of a higher-order smoothed gradient estimation mask, such as a Sobel Filter, helps in the presence of noise, but has the trade-off of potentially blurring the appearance of fine features of interest. Our method is also less effective in near homogenous regions where gradient directions are not well defined. In these near homogenous regions, however, the scalar values read along the estimated gradients tend to still fall within the same region yielding scalar values similar to those which would be read if the gradients were well de-

fined.

Much like other higher dimensional transfer function methods, our technique can add high frequency color variations that result in integration artifacts during rendering. Rather than color varying as a function of one variable (scalar value), the composited color varies as a result of the two additional scalar values read along the gradient direction. In practice we found that with some over sampling and the use of smoothly varying lighting transfer functions, artifacts from integration errors were not objectionable.

Our method shares a number of similarities to the use of scalar value/gradient magnitude pairs for transfer function specification. If one considers that gradient magnitude provides a measurement of how a scalar field varies at a point, then adding and subtracting this derivative from a scalar value sample provide an estimate of the values that would be acquired by sampling unit spacing along the gradient direction. The accuracy of this estimation depends on the degree that the scalar field varies linearly at that point, with material boundaries often having scalar fields that behave the least linearly. Our method, on the other hand, is able to take into account the actual distribution of scalar values along a material boundary directly and is thus better suited to deal with non-linear variations in scalar values at material transitions. The trade-off of using this information is the additional computation of performing these added scalar value reads.

## 7 FUTURE WORK AND CONCLUSION

In our current implementation we only use the scalar value pair read along the gradient direction for 2D lighting transfer functions. The center value could also be used as input for a three-dimensional lighting transfer function. Using this third variable could add significant complexity to the user interface and require additional texture storage for a 3D transfer function. We would like to investigate the use of this additional variable while avoiding these potential limitations.

The focus of this work is to use lighting to illustrate surfaces rather than opacity. The modulation of opacity can also provide a helpful means of illustrating semitransparent shapes. A future area of study is how the multi-sample transfer function method presented here could be used not only for lighting, but also opacity. The right of Figure 11 shows a preliminary result we obtained by applying our method for material classification to assign color and opacity. In this example, the branches have a higher density than the leaves, while the leaves have a higher density than air. Because of inherent blur in the acquisition of the data set, the regions surrounding the branches have the same density as the leaves. With the use of a conventional 1D transfer function, shown on the left, the regions surrounding the branches are misclassified as leaves, resulting in a moss-like appearance. By using a sample's scalar value,

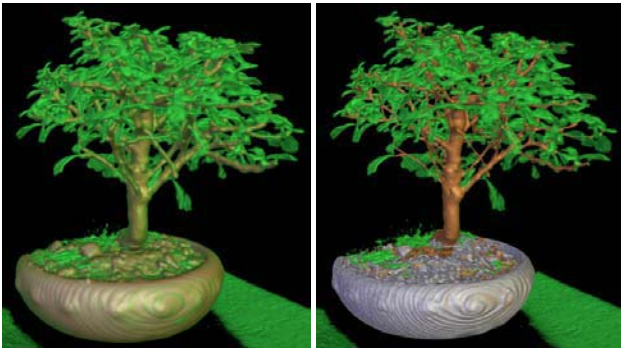


Figure 11: The regions surrounding the branches have the same intensity as the leaves and are incorrectly classified using a conventional transfer function resulting in a layer of green surrounding the branches as shown on the left. The right image shows the result of applying our multi-sample transfer function method for the specification of color and opacity. Notice that the voxels at transitions between wood and air are now classified as air or wood, and are no longer green.

and the two scalars read along the gradient direction as the domain of a 3D transfer function, a transfer function can be specified that takes into account the values on each side of a material boundary to properly segment the leaves from the regions surrounding the branches.

A major strength of the user interface presented in this paper is its simplicity. The user can see the different material boundaries that occur in the volume by looking at the line-based histogram, and can add illumination to a boundary by selecting scalar values that occur on each side of that boundary. The simplicity of this interface yields some limitations. With the interface we describe, each material boundary is selected using a 2D transfer function that is separable. Although an arbitrary transfer function can be constructed by combining multiple sets of widgets, it would be desirable to have a more direct means of providing this flexibility. In particular, one might want to enhance lighting based on the magnitude of the difference between the two scalar samples, which would not be possible using a separable transfer function. Providing a high degree of control over the construction of the transfer function, while maintaining the level of intuitiveness available with our current user interface, remains an important but difficult problem deserving future study.

Traditional transfer function methods use opacity to enhance material boundaries of interest. This paper describes a method that instead gives a high degree of control over illumination to illustrate surfaces, allowing opacity to be reserved for indicating thickness and occlusion. A novel user interface for visualizing and specifying these lighting transfer functions has also been presented. It is our belief that the technique we have developed can be used for the creation of more visually expressive imagery, to better meet the needs of scientists.

## 8 ACKNOWLEDGMENTS

This work has been sponsored in part by the U.S. National Science Foundation under contracts ACI 9983641 (PECASE award), ACI 0325934 (ITR), and ACI 0222991; the U.S. Department of Energy under Memorandum Agreements No. DE-FC02-01ER41202 (SciDAC program), and No. B523578 (ASCI VIEWS); the LANL/UC CARE program; the National Institute of Health through the Human Brain Project; and a United States Department of Education Government Assistance in Areas of National Need (DOE-GAANN)

grant P200A980307. We would like to thank the VIZLAB of CAIP at Rutgers University, SFB 382 of the German Research Council, General Electric, General Electric Aircraft Engines, Evendale, Ohio, and Stefan Roettger, VIS, University of Stuttgart for providing data sets. The authors would also like to thank ATI Technologies and NVIDIA Corporation.

## REFERENCES

- [1] James Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In *SIGGRAPH '82 Conference Proceedings*, pages 21–29, July 1982.
- [2] Robert Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. In *SIGGRAPH '88 Conference Proceedings*, pages 65–74, August 1988.
- [3] Helwig Hauser, Lukas Mroz, Gian-Italo Bisch, and M. Eduard Groller. Two-level volume rendering- fusing MIP and DVR. In *IEEE Visualization 2000 Conference Proceedings*, pages 211–218, 2000.
- [4] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH '98 Conference Proceedings*, pages 311–320, July 1998.
- [5] James T. Kajiya and Brian P. Von Herzen. Ray tracing volume densities. In *SIGGRAPH '84 Conference Proceedings*, pages 165–174, July 1984.
- [6] Gordon Kindlmann. Transfer functions in direct volume rendering: Design, interface, interaction. In *SIGGRAPH 2002 Course Notes*, 2002.
- [7] Gordon Kindlmann and James W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium on Volume Visualization*, pages 79–86, 1998.
- [8] Gordon Kindlmann and David Weinstein. Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields. In *Proceedings of IEEE Visualization '99 Conference*, pages 183–189, October 1999.
- [9] Joe Kniss, Gordon Kindlmann, and Charles Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001 Conference*, pages 255–262, October 2001.
- [10] Joe Kniss, Simon Premoze, Charles Hansen, and David Ebert. A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):150–162, April-June 2003.
- [11] Wolfgang Krueger. The application of transport theory to visualization of 3-D scalar data fields. *Computers in Physics*, pages 397–406, July-August 1991.
- [12] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.
- [13] Nelson Max. Light diffusion through clouds and haze. *Computer Vision, Graphics, and Image Processing*, 33(3):280–292, 1986.
- [14] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [15] Tomoyuki Nishita, Eiichi Nakamae, and Yoshinori Dobashi. Display of clouds and snow taking into account multiple anisotropic scattering and sky light. In *SIGGRAPH '96 Conference Proceedings*, pages 379–386, August 1996.
- [16] Herke Jan Noordmans, Hans T.M. van der Voort, and Arnold W.M. Smeulders. Spectral volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):196–207, 2000.
- [17] David Rodgman and Min Chen. Refraction in discrete ray tracing. In *Proceedings of Volume Graphics 2001*, pages 3–17, 2001.
- [18] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *SIGGRAPH '87 Conference Proceedings*, pages 293–302, July 1987.
- [19] Allen Van Gelder and Kwansik Kim. Direct volume rendering with shading via three-dimension textures. In *ACM Symposium on Volume Visualization '96 Conference Proceedings*, pages 23–30, 1996.