Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields

Gordon Kindlmann and David Weinstein

Scientific Computing and Imaging
Department of Computer Science, University of Utah
{gk|dmw}@cs.utah.edu

Abstract

With the development of magnetic resonance imaging techniques for acquiring diffusion tensor data from biological tissue, visualization of tensor data has become a new research focus. The diffusion tensor describes the directional dependence of water molecules' diffusion and can be represented by a three-by-three symmetric matrix. Visualization of second-order tensor fields is difficult because the data values have many degrees of freedom. Existing visualization techniques are best at portraying the tensor's properties over a two-dimensional field, or over a small subset of locations within a three-dimensional field. A means of visualizing the global structure in measured diffusion tensor data is needed. We propose the use of direct volume rendering, with novel approaches for the tensors' coloring, lighting, and opacity assignment. *Hue-balls* use a two-dimensional colormap on the unit sphere to illustrate the tensor's action as a linear operator. *Lit-tensors* provide a lighting model for tensors which includes as special cases both lit-lines (from streamline vector visualization) and standard Phong surface lighting. Together with an opacity assignment based on a novel two-dimensional barycentric space of anisotropy, these methods are shown to produce informative renderings of measured diffusion tensor data from the human brain.

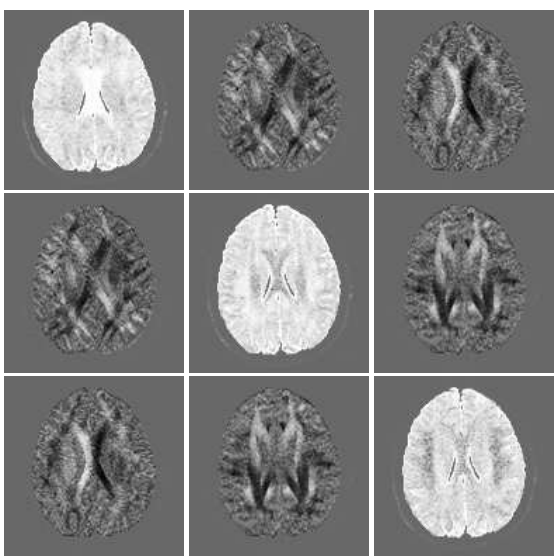


Figure 1: Matrix of images showing the individual tensor components within one dataset slice

1 Introduction

A fundamental property of biological tissue is the ability of water molecules to move within it by the action of Brownian motion. Rather than being one fixed velocity, this movement, called diffusion, is often anisotropic – happening faster in some directions than others. We use the term *anisotropy* to describe how different the rates of diffusion can be. Anisotropy is high when the rate can vary greatly as a function of direction. It is low when the rate is the same, regardless of direction. A complete description of the diffusion rate's directional dependence is afforded by a second-order tensor, representable as a three-by-three real-valued symmetric matrix.

To provide a feel for measured tensor data, a slice of a human brain diffusion tensor dataset is portrayed in Figure 1. Each subimage in the matrix of images is a gray-scale representation of the corresponding component of the tensor matrix, with medium gray representing zero. In the brain interior, the on-diagonal components of the tensor matrix are positive, while the off-diagonal components can be either positive or negative. This method of portraying the raw tensor data is not novel, nor is it a very intuitive way to display the orientation and shape of the diffusion tensors.

All three-by-three real-valued symmetric matrices have three real eigenvalues and three real-valued orthogonal eigenvectors [17]. The diffusion tensor matrix enjoys the additional constraint of having non-negative eigenvalues, implying it can be unambiguously represented as an ellipsoid. The ellipsoid's major, medium, and minor axes are along the tensor's eigenvectors, with the scalings along the axes being the eigenvalues. Such an ellipsoid is the image of the unit sphere under the linear transform induced by the tensor's matrix representation[1]. More intuitively, if one were to put a drop of ink into a diffusive material, the ink might be drawn in some directions faster that others, and the resulting shape would approximate the ellipsoid described.

The ellipsoid provides an elegant and powerful way to visualize the tensor because it has a simple shape, and yet it has just as many degrees of freedom as the diffusion tensor. Indeed, previous work in diffusion tensor visualization has used arrays of ellipsoids to depict the tensor field within a two dimensional region. Another tensor visualization method, hyperstreamlines, succeeds in faithfully depicting the tensor along one-dimensional paths in a volume dataset. These methods, and other previous approaches to this problem, are useful because they produce a means of visually decoding *all* the tensor's degrees of freedom at some set of locations in the field.

In some cases, however, it may be desirable to create renderings of tensor datasets by displaying only *some* of the information, but *everywhere* within a volume. The application motivating this research is to create an understanding of the fibrous structure of white matter throughout the brain. Because the white matter fiber tracts connect major regions of the brain, a detailed understanding

[1] This is not the only unambiguous ellipsoidal representation. One could also represent the tensor with the unit sphere's pre-image, or, for a tensor M one could also use the set of points $\mathbf{x}$ such that $\mathbf{x}^{\mathrm{T}}\mathrm{M}\mathbf{x} = 1$, as is done by Strang [17].

of their structure could foster advances in surgical planning, neurophysiology, and cognitive science [12, 15]. Fortunately, developments in magnetic resonance imaging have made it possible to accurately measure the water diffusion tensor within living brain tissue [2]. The white matter fiber tracts can be distinguished from their surroundings based on properties of the measured diffusion tensor, such as its anisotropy. Visualizing the intricate structure of the fiber tracts is inherently a three-dimensional problem. A technique that makes the large scale patterns within the diffusion tensor field visually apparent would be ideal.

Since this has historically been the goal of direct volume rendering for scalar data, we have explored the use of direct volume rendering for diffusion tensor visualization. To make this possible, the various ingredients of the direct volume rendering algorithm need to be supplied from the tensor data. We propose methods for assigning color and opacity to each location within the dataset, as well as a way to *illuminate* the tensors in a way that can be integrated into a direct volume rendering algorithm such as raycasting. *Hue-balls* permit coloring of the diffusion tensor based on the linear transform of the tensor's matrix form. A user-specified unit vector is mapped by the tensor to an output vector, whose direction is then visualized by a two-dimensional colormap on the sphere. For the opacity assignment, we use a two-dimensional barycentric space of anisotropy (based on three existing anisotropy measures) as the domain of an opacity function. Finally, *lit-tensors* provide a way of illuminating a tensor according to the type and orientation of anisotropy which it exhibits. Lit-tensors provide a shading model for the one-parameter family of anisotropy between linear anisotropy (where the shading model coincides with illuminated streamlines), to planar anisotropy (where it coincides with traditional surface illumination).

2 Related Work

Much previous work in tensor visualization has started by simplifying the data to a scalar or vector field, to which established visualization techniques can be applied. That is, the tensor is viewed only in terms of some salient scalar or vector characteristic. For example, *tensor field lines* allow one to see the patterns in the vector fields composed of the eigenvectors of a tensor matrix [7]. In the medical community there is much interest in visualizing two-dimensional slices of MR diffusion tensor data by colormapping the direction of the principal eigenvector (the eigenvector associated with the largest eigenvalue) [4, 9, 14].

When the tensor visualization isn't accomplished by showing only some of the information at all locations, it is often done by showing all the tensor information in a restricted subset of locations. A natural choice has been the ellipsoid representation of the tensor [10, 13, 16, 18], though rectangular prisms (with geometry determined by the eigensystem) also work very well [21]. A recent advance along these lines was inspired by artists who vary the characteristics of discrete brush strokes to convey information [11]. Through a carefully designed mapping from tensor attributes to brush stroke qualities, a two-dimensional MR diffusion tensor dataset can be rendered as an image with rich information content. Furthermore, the image can be understood at a range of scales, showing both the overall shape of the anisotropic regions, as well as the degree and direction of anisotropy at one particular location.

Another method of tensor visualization by explicit representation is hyperstreamlines [5, 6]. Streamlines are advected through a vector field of one of the eigenvectors, but instead of simply drawing a line to indicate a path, a surface is formed whose cross-section indicates the orientation of the other two eigenvectors and their associated eigenvalues. As with the ellipsoids, this type of representation must be unobstructed to be interpreted, and so the density of hyperstreamlines in the volume must be low in order to avoid visual cluttering. Also, as with any scheme in which high-dimensional information is carefully packed into a single image, it can take some time to learn how to "read" these visualizations.

One could argue that density of visual information is what limits the number of hyperstreamlines that can go into a single visualization, or prevents a stack of ellipsoid-based two-dimensional visualizations from being readily composited to form a volume rendering. However, volume rendering is precisely what is needed for our application. Three-dimensional rendering of tensor fields will almost certainly *require* the elision of some of the tensor information; the challenge is to choose which tensor characteristics to display and how to do so.

3 Methods

3.1 Hue-balls

The first part of our method uses color to visually distinguish regions exhibiting different diffusion tensors. It is based on a simple way to visualize unit-length vectors: a colormap on the sphere that gives a continuous mapping from direction to color. A closely analogous approach in vector visualization used a two dimensional hue-saturation colormap on the sphere to visualize perturbation velocity in an application of direct volume rendering to computational fluid dynamics [19]. What turns this into a tensor visualization technique is the application of the tensor matrix as a linear transform. At all locations in the tensor field, a single user-specified input vector is multiplied by the diffusion tensor matrix to create an output vector. The tensor is assigned color by using the direction of the output vector as the lookup into the spherical colormap. We use the term *hue-ball* to describe a spherical colormap used in this way. One motivation for the hue-ball is that throughout a region of the tensor field with high spatial coherence, multiplying by the tensor will tend to give the same result, and the assigned color will be nearly uniform. Discerning coherent structures in the tensor field becomes a task of visually detecting color coherence in the rendered image.

The properties of the tensor's matrix representation are important for understanding how the hue-ball functions. It is useful to consider the input vector in the basis formed by the eigenvectors. Given a tensor matrix M with unit-length eigenvectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$, an input vector $\mathbf{v}$ can be expressed as

$$\begin{aligned} \mathbf{v} &= (\mathbf{v}\cdot\mathbf{e}_1)\,\mathbf{e}_1 + (\mathbf{v}\cdot\mathbf{e}_2)\,\mathbf{e}_2 + (\mathbf{v}\cdot\mathbf{e}_3)\,\mathbf{e}_3 \\ &= (\mathbf{v}\cdot\mathbf{e}_1, \mathbf{v}\cdot\mathbf{e}_2, \mathbf{v}\cdot\mathbf{e}_3) \end{aligned} \tag{1}$$

Then the output vector $\mathrm{M}\mathbf{v}$ can be expressed as

$$\begin{aligned} \mathrm{M}\mathbf{v} &= \mathrm{M}(\mathbf{v}\cdot\mathbf{e}_1)\,\mathbf{e}_1 + \mathrm{M}(\mathbf{v}\cdot\mathbf{e}_2)\,\mathbf{e}_2 + \mathrm{M}(\mathbf{v}\cdot\mathbf{e}_3)\,\mathbf{e}_3 \\ &= \lambda_1(\mathbf{v}\cdot\mathbf{e}_1)\,\mathbf{e}_1 + \lambda_2(\mathbf{v}\cdot\mathbf{e}_2)\,\mathbf{e}_2 + \lambda_3(\mathbf{v}\cdot\mathbf{e}_3)\,\mathbf{e}_3 \\ &= (\lambda_1\mathbf{v}\cdot\mathbf{e}_1, \lambda_2\mathbf{v}\cdot\mathbf{e}_2, \lambda_3\mathbf{v}\cdot\mathbf{e}_3) \end{aligned} \tag{2}$$

where λ_i is the eigenvalue corresponding to eigenvector $\mathbf{e}_i$. The coordinates of the output vector in the eigenvector basis are the input vector's coordinates, scaled by the corresponding eigenvalues. We term the change in direction between the input and output vectors the *deflection* caused by the tensor.

Equation 2 indicates that the vector is always deflected towards the principal eigenvector, since the coordinate of the input vector in the principal eigenvector direction will by definition grow proportionally larger than the components along the other eigenvectors. There is also a relationship between the amount of deflection and the tensor's anisotropy. Because the anisotropy of a tensor is in general related to the disparity among its three eigenvalues, multiplying a vector by a tensor with high anisotropy will cause a greater relative change among its coordinates, and hence a greater deflection.

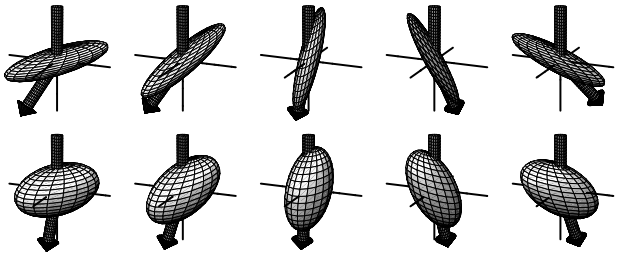
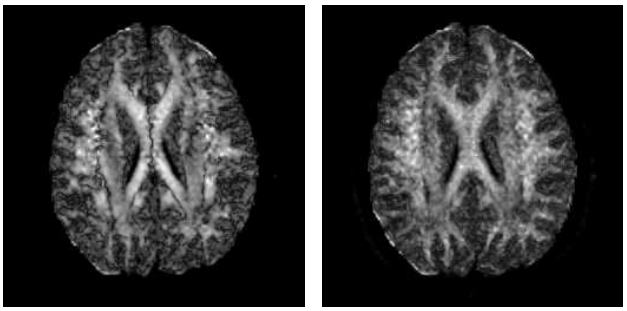


Figure 2: The diffusion tensor deflects the input vector by an amount related to its orientation and anisotropy



(a) Deflection (b) Anisotropy

Figure 3: Comparison within one dataset slice of the amount of deflection (a) from multiplication by the tensor matrix, with anisotropy (b) calculated from eigenvalues.

But there is a limit to how much the vector can be deflected. Since the diffusion tensor matrix has non-negative eigenvalues, multiplying by it can not change the sign of any of the vector's coordinates. Both the input and output vectors will be in the same octant of the eigenvector basis, so the angle between input and output vectors cannot exceed 90 degrees.

Figure 2 illustrates this deflection with two sequences of tensors, represented with their ellipsoidal form. Within each row, the anisotropy of the tensor is fixed, but the orientation is changing gradually. The upper row exhibits more anisotropy than the lower, but the orientations are matched between the two rows. The input vector is always pointing straight downward, above the ellipsoid, and the output vector is below the ellipsoid, pointing away from it. The deflection is always towards the principal eigenvector, but there is less deflection in the less anisotropic case. This illustrates the hue-ball's ability to de-emphasize isotropic regions (having low anisotropy). If the input vector points to a darker or more neutral color than the rest of the hue-ball, then isotropic regions will be inconspicuous, since in these regions there is little deflection of the input vector. Where the tensor field is more anisotropic, multiplying by the tensor has the potential to deflect the vector more, and the hue-ball can make these regions more noticeable.

Figure 3 shows how well the amount of deflection can correspond to anisotropy in real data. Figure 3(a) was generated with a gray-scale hue-ball which was black at one pole, with brightness increasing linearly with angular deviation from the pole. The input vector points into the page, towards the black pole. Figure 3(b) shows the result of evaluating a specific anisotropy measure (called the "anisotropy index") everywhere in the same slice of data. While the details of the anisotropy calculation are given in Section 3.2, the relevant point is that the anisotropy is calculated from the eigenvalues of the tensor at each point in the field. This requires finding the roots of the tensor's cubic characteristic polynomial, demanding much more computation than is needed to do the simple matrix multiplication used for the hue-ball. Yet, the results are comparable. Both have a large bright "X" shape in the center, indicating the corpus collosum (connecting the two sides of the brain), and other finer structures also appear in both images.

One important difference, however, is the dark seam which extends down the middle of the corpus collosum in Figure 3(a). This occurs when the input vector is perpendicular to the tensor's principal eigenvector, becoming aligned with either of the tensor's minor eigenvectors. In this configuration, there is no deflection towards the principal eigenvector. Using a different input vector can move the problem to another region of the dataset, but does not eliminate it. Also, if the tensor's major eigenvector is aligned with the input vector, here again there will be little deflection. These are basic limits on how well the deflection of the hue-ball's input vector can approximate the anisotropy.

The free parameters in the hue-ball method of tensor visualization are the color assignment on the sphere, and the input vector to use for multiplication with the diffusion matrix. For the sake of simplicity we have used only the hue-ball mapping shown in Figure 8(a) (colorplate). The sphere has a band of saturated colors around its equator, with saturation decreasing to the top and bottom poles, which are a medium gray. All the colors have the same "lightness" in the HSL color space [8], since for the sake of coloring a tensor field, it is visually less confusing if the hue-ball varies only color, letting the shading model control intensity[2]. All the hues appear twice on the hue-ball so as to create 180 degree rotational symmetry. As seen in Figure 2, the input vector can be deflected towards either end of the ellipsoid, depending on its orientation. The rotational symmetry insures that in either case the same hue is assigned.

In the same way that lights illuminating a scene can either be fixed in world or view coordinates, the hue-ball input vector can either be fixed relative to the dataset or relative to the viewpoint. If it is fixed relative to the dataset, changing the viewpoint allows one to inspect a single hue-ball mapping from different angles. When the hue-ball input vector is fixed in view space, the element of interactivity or motion can provide additional cues about the directionally dependent nature of the diffusion tensor. Image interpretation can be simplified by consistently orienting the hue-ball so that the input vector points to its neutral pole, and so that some hue is always aligned with an "up" direction. Regions in the volume which didn't significantly deflect the input vector will stay a neutral color, and a consistent mapping between color and deflection direction is maintained.

To illustrate how the hue-ball colors measured tensor data, the same dataset slice which was shown in Figures 1 and 3 has been mapped by the HSL hue-ball described above and is shown in Figure 8(b) (colorplate). Some previous techniques for colormapping diffusion tensor data assign color based on the direction of the principle eigenvector, and then modulate the color by the anisotropy, so that isotropic regions do not stand out. Using an appropriately chosen hue-ball mapping, with the input vector pointing to a neutral color, this happens automatically, so no anisotropy calculation is needed.

3.2 Barycentric Opacity Mapping

Another basic ingredient in direct volume rendering is an opacity assignment that makes uninteresting regions transparent so they do not occlude important structures. We term such a mapping from a diffusion tensor to opacity an *opacity function*. In our application, the objects of interest (white matter fiber tracts) are by definition very anisotropic, so high opacity should be assigned to them. Conversely, low or no opacity should be assigned to the gray matter on the exterior of the brain. The literature provides various metrics

[2] If one is seeking a truly constant luminance colormap, HSL colorspace is too simplistic.

for anisotropy based on the tensor matrix's three sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ [13, 20]. We have chosen to use the ones by Westin *et al*. [20] due to the simple geometric motivation behind them. Metrics for three different kinds of anisotropy are given:

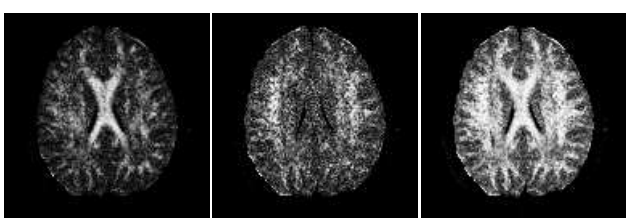
$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \quad (3)$$

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \quad (4)$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (5)$$

It can be shown that all the metrics fall in the range $[0, 1]$, and that they sum to unity: $c_l + c_p + c_s = 1$. The ellipsoids drawn next to the anisotropy metrics indicate the shape of diffusion tensor for which that metric will be high; it will be near zero for the other two shapes. Where only c_l is high, the tensor field is said to be *linearly anisotropic*, where only c_p is high, the tensor field is *planarly anisotropic*. The last metric, c_s is actually for *isotropy*; $c_s = 1$ only when all the eigenvalues are equal. Therefore, a single anisotropy metric called the "anisotropy index" is defined as:

$$c_a = 1 - c_s = c_l + c_p = \frac{\lambda_1 + \lambda_2 - 2\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (6)$$



(a) c_l (b) c_p (c) c_a

Figure 4: Different anisotropies measured on slice.

To see how the anisotropy can vary in measured data, Figure 4 shows the metrics c_l, c_p, and c_a evaluated over the same dataset slice seen in previous figures, with brighter areas indicating higher anisotropy.

These anisotropy measures provide a simple way of assigning opacity to diffusion tensors for direct volume rendering. A function whose domain and range are both $[0, 1]$ can map the values from an anisotropy measure (such as c_l) to opacity, assuming that the chosen anisotropy measure is high exactly in those regions that are deemed important for visualization.

A more flexible approach uses multiple anisotropy measures to facilitate more complex opacity functions. In light of the normalization built into c_l, c_p, and c_s, we propose the use of barycentric coordinates to depict the space of possible anisotropies, as shown in Figure 5. For every point in the triangle, there is a corresponding ellipsoid for which the anisotropy measures (c_l, c_p, and c_s) evaluate to the point's barycentric coordinates. In the figure, the three ellipsoids accompanying the corners of the triangle are representative of the ellipsoids which correspond to those corners. At each vertex of the triangle, one of the anisotropy measures is one, while the two others are both zero. Along the sides of the triangle, one of the anisotropy measures is zero, and the other two measures sum to one.

The same two-dimensional barycentric space of anisotropies can serve as the domain of an opacity function for direct volume rendering a diffusion tensor field. This class of opacity functions can accentuate the range of anisotropy types in a very flexible way, while including the previously described opacity functions (which depend on a single anisotropy metric) as special cases. Though space does not permit an exploration of the idea, we also note here that the barycentric space can also serve as the domain of a color mapping ("transfer function") applied to the tensor field, allowed different anisotropies to be discerned by varying their color.

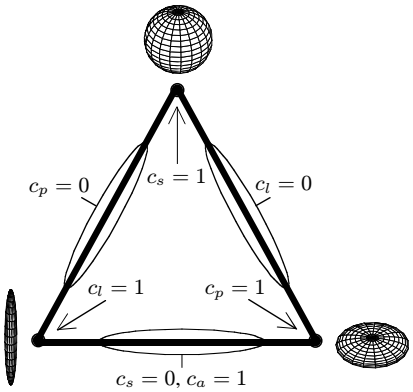


Figure 5: Barycentric space of anisotropies

Figure 6 demonstrates some barycentric opacity maps. Each opacity map is depicted by gray-scale representation: brighter regions in the triangle correspond to higher opacity assignment. For the purposes of this figure, the effect of the opacity map is demonstrated by applying the map to the the tensor dataset, resulting in a *scalar* volume of opacity values. This new scalar volume is visualized with a *linear* opacity function, and shaded according to the gradient of opacity values. A single white light is coming from the upper right. One can see that analogous to Figure 4, appropriately chosen opacity functions allow one to see the form of structures in the dataset which have one predominant type of anisotropy. It is these structures which will be colored by the action of the hue-ball, and illuminated according to lit-tensors.

3.3 Lit-Tensors

Streamlines used in vector visualization are sometimes hard to interpret because they lack the shading cues which we are accustomed to seeing on surfaces. However, illuminated streamlines ("lit-lines") have shading and highlights that give information about their direction and curvature, creating the appearance of shiny filaments [22]. In the case of diffusion tensor visualization, we have made tensors opaque based on their anisotropy, but no other information about the tensor is indicated. We aim to show that a means of illuminating diffusion tensors, called *lit-tensors*, can help disambiguate the type and orientation of anisotropy. The constraints we followed for designing such a scheme are:

1. In regions of complete linear anisotropy, the lighting model should be identical to that of illuminated streamlines. Complete linear anisotropy means movement by diffusion is constrained to one dimension, so it is sensible for the lighting model to degenerate to one already developed for vector fields.

2. In regions of complete planar anisotropy, the lighting model should be the same as with traditional surface rendering. The obvious choice for the "surface normal" for a planar anisotropic tensor is the third eigenvector, perpendicular to the plane formed by the span of the first two eigenvectors (associated with the largest two eigenvalues).

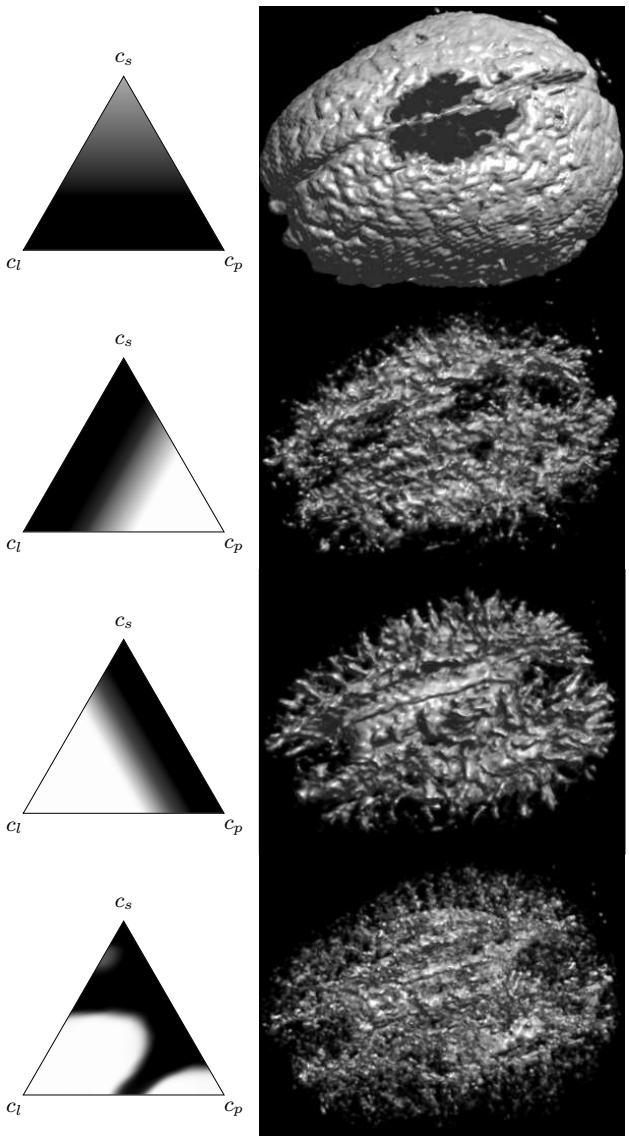


Figure 6: Examples of barycentric opacity maps and resulting volumes

3. There has to be a smooth interpolation between these two extremes. Since tensor data can exhibit a wide variety of anisotropies, allowing small variations in anisotropy to lead to large changes in shading will probably create a more confusing image.

This can be seen as a problem of how to interpolate illumination between different codimensions. The codimension of the diffusion tensor's representative ellipsoid is two in the linear anisotropy case, and one with planar anisotropy. Previous work [1] has rigorously developed illumination methods for general manifold dimension and codimension, but did not cover cases part-way between different codimensions. Unlike that work, no claim to physical accuracy or plausibility is made for the model presented here; it is just one simple way of satisfying the constraints above.

We take as our starting point the Blinn-Phong lighting model [3]:

$$\begin{aligned} I &= I_{\text{ambient}} + I_{\text{diffuse}} + I_{\text{specular}} \\ &= k_{\text{a}} A_\lambda O_\lambda + I_\lambda (k_{\text{d}} O_\lambda \mathbf{L} \cdot \mathbf{N} + k_{\text{s}} (\mathbf{H} \cdot \mathbf{N})^n) \end{aligned} \tag{7}$$

k_{a}, k_{d}, and k_{s} control the contributions of ambient, diffuse, and specular reflection to the final image. Following Foley *et al.* [8], we add the subscript λ to those variables which vary according to color. For example, there are separate values I_{r}, I_{g}, I_{b}, for the red, green, and blue components of the directional light source. The ambient light color is A_λ. Instead of representing the intrinsic object color with different k_{a} and k_{d} for red, green, and blue, we use O_λ for object color and keep k_{a} and k_{d} as separate controls. In our case, the intrinsic object color is determined by the output of the hueball. $\mathbf{L}$ is the vector pointing towards the directional light source, $\mathbf{V}$ points towards the eye, and $\mathbf{N}$ is the surface normal. Note that instead of using $(\mathbf{V} \cdot \mathbf{R})^n$ for the specular component, where $\mathbf{R}$ is the reflection of $\mathbf{L}$ across $\mathbf{N}$, we are using the "half-way" vector $\mathbf{H}$ in $(\mathbf{H} \cdot \mathbf{N})^n$. $\mathbf{H}$ is the normalized average of $\mathbf{L}$ and $\mathbf{V}$, and n is the shininess exponent.

Because a streamline is one-dimensional, at any given point along it there is an infinite set of normals, all perpendicular to the tangent direction $\mathbf{T}$, radiating outwards in a circle. If naively using Equation 7 to illuminate a streamline, one must find the normal which is in the plane spanned by $\mathbf{L}$ and $\mathbf{T}$ to evaluate $\mathbf{L} \cdot \mathbf{N}$. Similarly, another specific normal must be found to evaluate $\mathbf{H} \cdot \mathbf{N}$. The insight which makes lit-lines simple is that one does not need to actually find a specific normal in order to evaluate a dot product with it. With Pythagoras's theorem, the dot product can be expressed in terms of the tangent $\mathbf{T}$:

$$\mathbf{U} \cdot \mathbf{N} = \sqrt{1 - (\mathbf{U} \cdot \mathbf{T})^2} \tag{8}$$

where $\mathbf{U}$ is either $\mathbf{L}$ or $\mathbf{H}$, for the diffuse and specular terms, respectively.

The relevant property of Equation 8 is that the lighting calculation depends on a vector (the tangent) which gives the object's direction, instead of a depending on a normal. The direction and orientation of a diffusion tensor is determined by not one, but two vectors: the first and second eigenvectors[3]. Both of these could be interpreted as tangents, but their relative importance is determined by the magnitudes of the corresponding eigenvalues. To characterize the relative importance of the first two eigenvectors in determining the tensor's orientation, we introduce a parameter c_θ. Assuming that the eigenvalues are ordered $\lambda_1 \geq \lambda_2 \geq \lambda_3$, we define

$$c_\theta = \frac{\pi}{2} \frac{c_p}{c_a} = \frac{\pi(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 - 2\lambda_3} \tag{9}$$

As anisotropy varies from completely linear ($c_l = 1$; $c_p = 0$) to completely planar ($c_l = 0$; $c_p = 1$), c_θ varies from 0 to $\frac{\pi}{2}$. The role of c_θ is to control how much the second eigenvector contributes to the lighting of the diffusion tensor. In the linear case, only the first eigenvector determines the tensor orientation, and in the planar case, both the first and second eigenvectors matter equally.

The expression to be used in lieu of dot products with $\mathbf{N}$ is:

$$\text{“}\mathbf{U} \cdot \mathbf{N}\text{”} = \sqrt{1 - (\mathbf{U} \cdot \mathbf{e}_1)^2 - (\mathbf{U} \cdot \mathbf{e}_2 \sin(c_\theta))^2} \tag{10}$$

Note that in the case of linear anisotropy, $\sin(c_\theta) = \sin(0) = 0$, so the contribution from $\mathbf{e}_2$ vanishes, and the expression reduces to the formula for lit-lines (Equation 8), with the principal eigenvector $\mathbf{e}_1$ taking the role of the tangent $\mathbf{T}$. This is appropriate, since in linear anisotropy, the principal eigenvector points in the direction of movement, as does a streamline's tangent vector.

In planar anisotropy, $\sin(c_\theta) = \sin(\frac{\pi}{2}) = 1$, and the contributions of the two dot products are equal. This means that for any other vector $\mathbf{W}$ such that

$$(\mathbf{W} \cdot \mathbf{e}_1)^2 + (\mathbf{W} \cdot \mathbf{e}_2)^2 = (\mathbf{U} \cdot \mathbf{e}_1)^2 + (\mathbf{U} \cdot \mathbf{e}_2)^2 \tag{11}$$

[3]Because the eigenvectors always form an orthogonal basis, and because we are adopting two-sided lighting, the third eigenvector does not contribute any additional information.

Equation 10 will have the same value. Therefore, in planar anisotropy the lighting model is rotationally symmetric around $\mathbf{e}_3$. Rotational symmetry in this case is actually an important feature of the lighting model. In planar anisotropy, the diffusion tensor ellipsoid degenerates to a disc, and *any* vector in the plane spanned by the disc is an eigenvector. Because of this numerical instability, the calculated directions of the first and second eigenvectors will be essentially random. The illumination should not be sensitive to this arbitrary orientation, and should only be a function of the third eigenvector. In fact, one can use Pythagoras's theorem to show that if $c_\theta = \frac{\pi}{2}$, Equation 10 gives an exact formula for $\mathbf{U} \cdot \mathbf{e}_3$. Interpreting both $\mathbf{e}_1$ and $\mathbf{e}_2$ as surface tangents, then the surface normal $\mathbf{N}$ is aligned along $\mathbf{e}_3$. Therefore the model contains standard surface shading as a special case.

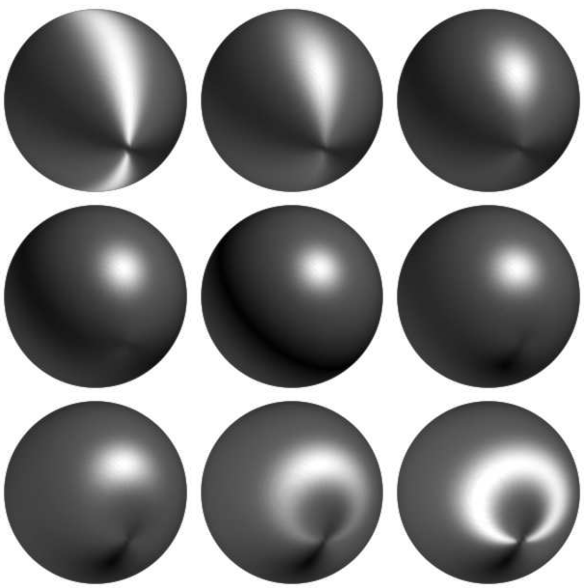


Figure 7: Sequence of volumes of differing anisotropy, rendered with lit-tensors. Anisotropy varies gradually between the nine volumes, going in scanline order.

To demonstrate lit-tensors, Figure 7 shows nine different synthetic diffusion tensor datasets which were direct volume rendered with a fixed viewpoint and light. The anisotropy index c_a of the sphere is also constant in every case, but c_θ is changing. The dataset in the upper left has complete linear anisotropy in a concentric circular pattern (along lines of latitude). The dataset in the middle has complete planar anisotropy (and hence looks just like a standard surface rendering). The dataset in the lower left has complete linear anisotropy along lines of longitude, going from pole to pole. The images provide a convincing sense of surface anisotropy, which is not a typical trait in direct volume renderings. Figure 8(c) on the colorplate shows the same sequence of volumes colored with a hue-ball.

4 Results

Figure 8 (colorplate) illustrates the action of hue-balls. Figure 9 shows all the components of diffusion tensor volume rendering that have been described so far. The diffusion tensor dataset being rendered (the same as in Figure 6) is a 128 by 128 by 60 voxel scan of a live brain. Our volume renderer is a simple ray-caster which uses one ray per pixel. When rendering a tensor volume, it calculates the eigensystem at each point along the ray, then uses the eigenvalues to determine opacity (according to the anisotropy opacity map), and the eigenvectors to determine lighting (according to lit-tensors).

Figure 9(a) is a rendering of a *scalar* dataset which was generating by mapping the tensor dataset through an anisotropy opacity map similar to the third map in Figure 6. The general structure of the anisotropic region as a whole is clear, but the distinct white matter fiber tracts are not easily distinguishable. This is remedied in Figure 9(b), where the volume is colored by the hue-ball, but the surface shading is the same as before (using only one white light). Now the regions with distinct anisotropy directions attain distinct colors. For example, the cingulum bundle (blue) is now clearly visible as a distinct tract from the corpus collosum (orange) below it. Some other major features are labeled. Figure 9(c) shows the result of using lit-tensors without a hue-ball. The light positions and colors are the same as in 9(a). The dark regions are mainly planarly anisotropic; they need to be correctly aligned to reflect light, whereas linearly anisotropic regions are inherently more reflective. Figure 9(d) shows the use of a hue-ball with lit-tensors. With one fixed rendering, the effect of lit-tensor highlights is not as clear as in the accompanying video, in which the highlights are seen to travel across the features with viewpoint motion.

5 Discussion

Applying direct volume rendering to diffusion tensor visualization is challenging because the complexity of the underlying data values prohibits "direct" mapping from the data to color and opacity. Counter to the appealing simplicity of typical scalar direct volume rendering, we *do* have to perform significant intermediate geometric processing in order to arrive at color and opacity values for each location in the field. The main contribution of this paper is a set of tensor analysis methods (hue-balls, barycentric opacity maps, and lit-tensors) which bridge the inherent complexity of the raw tensor data with the simplicity of direct volume rendering. Because anisotropy is a fundamental structural characteristic of the tensor field, the methods presented all portray information related to the tensor's anisotropy. Hue-balls strive to color only those regions which are anisotropic in a way that indicates the direction and orientation of the anisotropy. The barycentric opacity maps allow precise selection of structures within the dataset according to the type of anisotropy within them. Finally, lit-tensors accentuate large-scale patterns of anisotropy type and orientation across the volume.

One of the strategies of this work was to avoid explicit reliance on the principal eigenvector, since in a majority of positions within our data, the type of anisotropy (planar or spherical) means that the calculated direction of the principal eigenvector will be unreliable. The hue-ball embodies this intent. While it indicates information about the orientation of anisotropy without calculating the eigensystem, it is at the cost of requiring user interaction, since the hue-ball depends on a user-specified input-vector. Lit-tensors do depend on the calculated orientation of the principal eigenvector, but by design, the lighting model is insensitive to the direction of the principal eigenvector precisely in those situations (planar anisotropy) where its calculated direction is unreliable.

There is a subtle issue in the rendering process which strikes us as an interesting problem for future work. In scalar volume rendering, the usual technique is to resample data values by interpolation, and then map them through the transfer function. One could also resample vectors by interpolating the vector data component-wise. It is not clear to us, however, how best to interpolate diffusion tensor data for direct volume rendering. One could interpolate the matrix representation component-wise at each point along the ray, but then the entire eigensystem has to be recalculated at each step. In the interests of simplicity, this is the approach we are currently using.

We resort to this method because interpolating pre-computed eigenvectors or eigenvalues is complicated by a correspondence

problem. Given a set of three eigenvectors at one sample point $\mathbf{x}_1$, and another three eigenvectors at a second sample point $\mathbf{x}_2$, we do not immediately know the correspondence between the two sets of eigenvectors. Knowing the *continuous* tensor field from which the discrete dataset is sampled, one could learn if, for example, the principal eigenvector at $\mathbf{x}_1$ is continuous with the principal eigenvector at $\mathbf{x}_2$; this need not be the case in general. Even if the two sets of eigenvectors have identical orientations, it isn't necessarily the case that pairs of eigenvectors with the same orientation actually correspond to each other. A similar correspondence problem complicates eigenvalue interpolation. Further research on efficient and accurate strategies for diffusion tensor interpolation is required.

The current problems in the methods presented point to various possibilities for further work. The biggest draw-back to the current techniques is the lack of interactivity. However, Figure 3 hints that to a coarse approximation, the hue-ball could be used to measure anisotropy and hence form the basis of a very fast opacity function. If one foregoes directional lighting, this raises the possibility of interactive diffusion tensor visualization, which certainly warrants further investigation. An unfortunate draw-back of lit-tensors is the difficulty in discerning the edges of structures that have been illuminated by them. In the renderings of scalar volumes shown in Figure 6, the surface normal gave helpful cues to the shape of the object which was selected by the opacity map. While lit-tensors do succeed in indicating the direction of anisotropy relative to the lights and viewpoint, they do not give sufficient information about the shape of anisotropic structures to allow disambiguation of configurations such as partial self-occlusion. Also, the colormap used with the hue-ball could have been more carefully designed. A mapping from a perceptually uniform colorspace may permit a more precise representation of direction.

6 Acknowledgments

This work was supported in part by awards from the National Science Foundation. The authors would like to thank Andrew L. Alexander, Assistant Research Professor in the Department of Radiology, University of Utah School of Medicine, for involving them in his research on diffusion tensor imaging and for providing them with datasets. The authors are also grateful for the work of Eric Lundberg in processing the raw data, and for the helpful detailed comments from Chris Johnson, Peter Shirley, Sean Heffernan, Matthew Bane, and the anonymous reviewers.

References

[1] D.C. Banks. Illumination in diverse codimensions. In *SIGGRAPH 94 Proceedings*, volume 28, pages 327–334, 1994.

[2] P.J. Basser, J. Mattiello, and D. Le Bihan. Estimation of the effective self-diffusion tensor from the NMR spin-echo. *Magnetic Resonence*, pages 247–254, 1994.

[3] James F. Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics*, pages 192–198, July 1977.

[4] J. Coremans, R. Luypaert, F. Verhelle, T. Stadnik, and M. Osteaux. A Method for Myelin Fiber Orientation Mapping Using Diffusion-Weighted MR Images. *Magnetic Resonance Imaging*, pages 443–454, 1994.

[5] T. Delmarcelle and L. Hesselink. Visualization of second order tensor fields and matrix data. In *IEEE Visualization 92 Proceedings*, pages 316–323, 1992.

[6] T. Delmarcelle and L. Hesselink. *Computer visualization: graphics techniques for scientific and engineering analysis*, chapter A Unified Framework for Flow Visualization, pages 129–170. CRC Press, 1995.

[7] R. R. Dickinson. A unified approach to the design of visualization software for the analysis of field problems. In *Three-dimensional Visualization and Display Technologies (Proceedings of SPIE)*, pages 173–180, 1989.

[8] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics Principles and Practice*, pages 592–595, 722–731. Addison-Wesley, 1990.

[9] D. K. Jones, S. C. R. Williams, and M. A. Horsfield. Full Representation of White-Matter Fibre Direction on One Map via Diffusion Tensor Analysis. In *Proc. 5th Int. Soc. of Mag. Res. in Med.*, page 1743, 1997.

[10] G.D. Kerlick. Moving iconic objects in scientific visualization. In *IEEE Visualization 90 Proceedings*, pages 124–130, 1990.

[11] David H. Laidlaw, Eric T. Ahrens, David Kremers, Matthew J. Avalos, Russell E. Jacobs, and Carol Readhead. Visualizing Diffusion Tensor Images of the Mouse Spinal Cord. In *IEEE Visualization 98*, pages 127–134, 1998.

[12] N. Makris, A.J. Worth, G. Sorensen, G.M. Papadimitriou, O. Wu, T.G. Reese, V.J. Wedeen, T.L. Davis, J.W. Stakes, V.S. Caviness, E. Kaplan, B.R. Rosen, D.N. Pandya, and D.N. Kennedy. Morphometry of *in vivo* human white matter association pathways with diffusion weighted mri. *Annals of Neurology*, 42(6):951–962, 1997.

[13] C. Pierpaoli and P.J. Basser. Toward a Quantitative Assessment of Diffusion Anisotropy. *Magnetic Resonance Magazine*, pages 893–906, 1996.

[14] Carlo Pierpaoli. Oh no! One more method for color mapping of fiber tract direction using Diffusion MR Imaging Data. In *Proc. 5th Int. Soc. of Mag. Res. in Med.*, page 1741, 1997.

[15] C. Poupon, J. F. Mangin, V. Frouin, J. Régis, F. Poupon, M. Pashot-Clouard, D. Le Bihan, and I. Bloch. Regularization of MR Diffusion Tensor Maps for Tracking Brain White Matter Bundles. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pages 489–498, 1998.

[16] D. Silver, N. Zabusky, V. Fernandez, and M. Gao. Ellipsoidal quantification of evolving phenomena. *Scientific Visualization of Natural Phenomena*, pages 573–588, 1991.

[17] Gilbert Strang. *Linear Algebra and Its Applications*, chapter 5.5. Academic Press, Inc., Orlando, Florida, 1976.

[18] C. Upson, R. Wolff, R. Weinberg, and D. Kerlich. Two and three dimensional visualization workshop. In *Course Number 19, SIGGRAPH 89*, 1989.

[19] Samuel P. Uselton. Volume Rendering for Computational Fluid Dynamics: Initial Results. Technical Report RNR-91-026, NAS-NASA Ames Research Center, September 1991.

[20] C-F. Westin, S. Peled, H. Gubjartsson, R. Kikinis, and F.A. Jolesz. Geometrical diffusion measures for MRI from tensor basis analysis. In *Proceedings of ISMRM*, 1997.

[21] Andrew J. Worth, Nikos Makris, Van J. Wedeen, Jr. Verne S. Caviness, and David N. Kennedy. Fusion of MRI data for Visualization of White Matter Bundles, 1998. http://neuro-www.mgh.harvard.edu/cma/staff/ajw/MICCAI98/MICCAI98.html

[22] M. Zöckeler, D. Stalling, and H-C. Hege. Interactive visualization of 3d-vector fields using illuminated streamlines. In *IEEE Visualization 96 Proceedings*, pages 107–113, 1996.