3D modeling and segmentation of diffusion weighted MRI data

Leonid Zhukov, Ken Museth, David E. Breen, Ross T. Whitaker

Leonid Zhukov, Ken Museth, David E. Breen, Ross T. Whitaker, "3D modeling and segmentation of diffusion weighted MRI data," Proc. SPIE 4319, Medical Imaging 2001: Visualization, Display, and Image-Guided Procedures, (28 May 2001); doi: 10.1117/12.428081

SPIE.

Event: Medical Imaging 2001, 2001, San Diego, CA, United States

3D Modeling and Segmentation of Diffusion Weighted MRI Data

Leonid Zhukov*, Ken Museth*, David Breen*, Ross Whitaker**

*Department of Computer Science, Mail Code 350-74
California Institute of Technology, Pasadena, CA 91125-7400

**School of Computing, 3190 MEB
University of Utah, Salt Lake City, UT 84112-9205

ABSTRACT

Diffusion weighted magnetic resonance imaging (DW MRI) is a technique that measures the diffusion properties of water molecules to produce a tensor-valued volume dataset. Because water molecules can diffuse more easily along fiber tracts, for example in the brain, rather than across them, diffusion is anisotropic and can be used for segmentation. Segmentation requires the identification of regions with different diffusion properties. In this paper we propose a new set of rotationally invariant diffusion measures which may be used to map the tensor data into a scalar representation. Our invariants may be rapidly computed because they do not require the calculation of eigenvalues. We use these invariants to analyze a 3D DW MRI scan of a human head and build geometric models corresponding to isotropic and anisotropic regions. We then utilize the models to perform quantitative analysis of these regions, for example calculating their surface area and volume.

Keywords: DW MRI, diffusion tensor, eigenvalues, invariants, iso-surface, geometric model, level set methods.

1. INTRODUCTION

Diffusion weighted magnetic resonance imaging[1] (DW MRI) is sensitive to random thermal movement of water molecules known as Brownian motion. Consequently, DWI can be used to detect the diffusion of water molecules in three dimensions. Because water molecules can diffuse more easily along fiber tracts, for example in the brain, rather than across them, diffusion is anisotropic and can be used to segment DWI datasets. Since diffusion is characterized by six independent values at every voxel (which are usually grouped in a symmetric tensor - a 3x3 matrix $\mathbf{T}$), segmentation and visualization of the data are not entirely straightforward. Moreover, the diffusion matrix is not invariant with respect to rotations, and the elements that form the matrix will be different for different orientations of the sample in the scanner and therefore cannot be used for classification purposes. Finally, 3D visualization techniques available today are designed mostly for scalar and sometimes vector fields. Thus, there are two fundamental problems in tensor imaging: a) finding an invariant representation of a tensor that is independent of a frame of reference and constructing a mapping from the tensor field to a scalar or vector field, and b) visualization and classification of tissue using the derived scalar fields. In this paper we address both issues, developing a new invariant representation of a tensor field and using it for 3D visualization, segmentation and construction of geometric models.

Traditional approaches to diffusion tensor imaging[2] involve converting the tensors into an eigenvalue/eigenvector representation, which is rotation invariant. Every tensor may then be interpreted as an ellipsoid with principal axes oriented along the eigenvectors and radii equal to the corresponding eigenvalues. This ellipsoid describes the probabilistic distribution of a water molecule after a fixed diffusion time. The corresponding anisotropy is commonly estimated as $(\lambda_{max}/\lambda_{min})$ - the ratio of the maximal to the minimal eigenvalue. A much better measure is fractional anisotropy,[3] which measures average restriction of diffusion in all directions:

$$FA = \frac{\sqrt{3}}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \lambda_m)^2 + (\lambda_2 - \lambda_m)^2 + (\lambda_3 - \lambda_m)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}, \quad (1)$$

Further author information: (Send correspondences to David Breen) E-mail: {zhukov,kmu,david}@gg.caltech.edu, whitaker@cs.utah.edu

Visualization, Display, and Image-Guided Procedures, Seong Ki Mun, Editor,
Proceedings of SPIE Vol. 4319 (2001) © 2001 SPIE · 1605-7422/01/$15.00

where λ_1, λ_2, λ_3 are eigenvalues of the diffusion tensor and λ_m is the average magnitude of the diffusion (mean diffusivity):

$$\lambda_m = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3). \tag{2}$$

Another recently developed approach[4,5] classifies tensors into three categories according to the size of its eigenvalues: 1) the linear case, ($\lambda_1 >> \lambda_2 \approx \lambda_3$), where diffusion is mainly in the direction corresponding to the largest eigenvalue, 2) the planar case ($\lambda_1 \approx \lambda_2 >> \lambda_3$), where diffusion is restricted to the plane formed by the two eigenvectors corresponding to the two largest eigenvalues and 3) the spherical case ($\lambda_1 = \lambda_2 = \lambda_3$) of isotropic diffusion. This leads to linear, planar and spherical measures

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \qquad c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \qquad c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{3}$$

that may be combined into one anisotropy measure that describes the deviation from spherical (isotropic) diffusion,

$$c_a = c_l + c_p = \frac{\lambda_1 + \lambda_2 - 2\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}. \tag{4}$$

The goal of these methods is the quantitative identification of isotropic and anisotropic tissues. Both methods have the same drawback - they rely on eigenvalue computations. Though the diagonalization of the 3x3 matrix is not expensive, it must be repeatedly performed for every voxel in the volume. This calculation may become a bottleneck for large datasets. For example, computing eigenvalues and eigenvectors for a 512^3 volume requires over 20 CPU-minutes on a powerful workstation. Another problem associated with eigenvalue computation is stability - a small amount of noise will not only change the values but also the ordering of the eigenvalues, which then changes the orientation of the eigenvectors. Many anisotropy measures depend on the ordering of the eigenvalues (except for FA (1)), and the calculated direction of diffusion can be significantly altered by the noise normally found in diffusion tensor datasets. Thus it is desirable to have an anisotropy measure which is rotationally invariant, does not require eigenvalue computations and is stable with respect to noise. We present tensor invariants with these characteristics in Section 2.

Visualization and model extraction from the invariant 3D scalar fields is the second issue addressed in this paper. One of the most popular approaches to tensor visualization represents a tensor field by drawing the ellipsoids derived from the eigenvectors/values.[6] This method requires a significant amount of computation, and is slow for large 3D volumes. Furthermore, ellipsoidal visualization creates visual cluttering in 3D, and is only acceptable for 2D data. Recently Kindlmann[7] explored visualization methods where measures (3) were used to convert a 3D tensor volume into a scalar volume, which is then directly rendered.

In our work we perform iso-surfacing on the 3D scalar fields derived from our tensor invariants to visualize and segment the data. An advantage of iso-surfacing over other approaches is that it can provide the shape information needed for constructing geometric models, and computing internal volumes and external surface areas of the extracted regions. Detailed discussion of the modeling method is given in Section 3. Section 4 presents the results of our tensor invariant calculations and model segmentation technique with examples from a DWI scan of a human head. Section 5 then describes the quantitative analysis of the geometric models.

Finally, there has been a number of recent publications[8,9] devoted to fiber tracking. This is a different and more complex task than the one addressed in this paper and requires data with a much higher resolution and signal-to-noise ratio than the data used in our study.

2. TENSOR INVARIANTS

Tensor invariants (rotational invariants) are combinations of tensor elements that do not change after the rotation of the tensor's frame of reference, and thus do not depend on the orientation of the patient with respect to the scanner when performing DWI. To help better understand the above definition consider a vector (which is a tensor of rank 1). A vector, as a geometric object, does not depend on any coordinate system, but its components - projections on different axes - do change with the rotation of the coordinate system. In fact, if the rotation of the system is given by a matrix $\mathbf{R}$, which incorporates the axis and angle of rotation, the values of the components of the vector in the new, rotated, coordinate system will be $\mathbf{v}' = \mathbf{R}\mathbf{v}$. The length of the vector will stay the same in both coordinate

systems (proof: $||\mathbf{v}'|| = ||\mathbf{R}\mathbf{v}|| = ||\mathbf{R}||||\mathbf{v}|| = ||\mathbf{v}||$), because the rotation matrix is orthogonal with a norm $||\mathbf{R}|| = 1$. Thus, the length of the vector is an invariant with respect to rotation of the frame of reference. When length is used to describe a vector its frame of reference need not be considered.

Similarly, a tensor does not depend on any one frame of reference, but its components, which are in fact projections of the tensor onto the axes, do depend on the coordinate system and change with rotation of the frame of reference according to $\mathbf{T}' = \mathbf{R}\mathbf{T}\mathbf{R}^{\mathbf{T}}$, where $\mathbf{R}$ remains the same rotation matrix. Therefore, there is a need for tensor invariants, i.e. certain scalar functions defined for a tensor that do not change with the rotation of its frame of reference. Group theory shows that a tensor of rank n has $n(n+1)/2$ independent rotational invariants.[10] A vector is a tensor of rank 1, and therefore has only one such value—its length. A diffusion tensor, which is a 3x3 matrix, is a tensor of rank 2 and accordingly it has three invariants.

In order to present our invariants, we start with the characteristic equation for the eigenvalues of the matrix $\mathbf{T}$,

$$det||\mathbf{T} - \lambda\mathbf{I}|| = \begin{vmatrix} T_{xx} - \lambda & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} - \lambda & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} - \lambda \end{vmatrix} = 0.$$

Expanding the determinant leads to the characteristic equation,

$$\lambda^3 - D_1 \cdot \lambda^2 + D_2 \cdot \lambda - D_3 = 0, \tag{5}$$

where the coefficients are

$$\begin{aligned} D_1 &= T_{xx} + T_{yy} + T_{zz} \\ D_2 &= T_{xx}T_{yy} - T_{xy}T_{yx} + T_{xx}T_{zz} - T_{xz}T_{zx} + T_{yy}T_{zz} - T_{yz}T_{zy} \\ D_3 &= T_{xx}(T_{yy}T_{zz} - T_{zy}T_{yz}) \\ &- T_{xy}(T_{yx}T_{zz} - T_{zx}T_{yz}) + T_{xz}(T_{yx}T_{zy} - T_{zx}T_{yy}). \end{aligned}$$

The roots of this equation are the eigenvalues of matrix $\mathbf{T}$. It is well known that eigenvalues of a matrix are rotational invariants of that matrix, which means that the coefficients of the above equation should also be invariant. We can explicitly prove this statement. For example, for the first coefficient D_1,

$$D_1' = \sum_i T_{ii}' = \sum_i \sum_j \sum_k R_{ij}T_{jk}R_{ki} = \sum_j \sum_k \delta_{kj}T_{jk} = \sum_j T_{jj} = D_1, \tag{6}$$

which shows that it does not change under rotations, i.e. the trace of the matrix is a rotational invariant. Following the same approach one can prove that the two other coefficients, D_2 and D_3 are, also rotational invariants. If the columns of rotation matrix $\mathbf{R}$ are formed from the eigenvectors of $\mathbf{T}$, the resulting transformation when applied to $\mathbf{T}$ will rotate $\mathbf{T}$'s original coordinate system into alignment with the eigenvectors, producing a diagonal tensor. Then

$$\begin{aligned} D_1 &= \lambda_1 + \lambda_2 + \lambda_3 \\ D_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 \\ D_3 &= \lambda_1\lambda_2\lambda_3. \end{aligned}$$

Thus, considering the ellipsoidal representation for eigenvalues/eigenvectors, we can see that the proposed invariants are proportional to the sum of the radii, surface area and the volume of the ellipsoid. Instead of using $(\lambda_1, \lambda_2, \lambda_3)$ to describe the dataset, one can use (D_1, D_2, D_3). Moreover, a diagonal representation allows us to express some of the popular anisotropy measures with our invariants. For example, mean diffusivity is $\lambda_m = \frac{1}{3}D_1$.

Any combination of the above invariants is, in turn, an invariant. We consider the following dimensionless combination: D_1D_2/D_3. In the eigenvector frame of reference it becomes

$$\frac{D_1D_2}{D_3} = 3 + \frac{\lambda_2 + \lambda_3}{\lambda_1} + \frac{\lambda_1 + \lambda_3}{\lambda_2} + \frac{\lambda_1 + \lambda_2}{\lambda_3} \tag{7}$$

and we can define a new dimensionless anisotropy measure

$$D_a = \frac{1}{6}\left[\frac{D_1D_2}{D_3} - 3\right]. \tag{8}$$

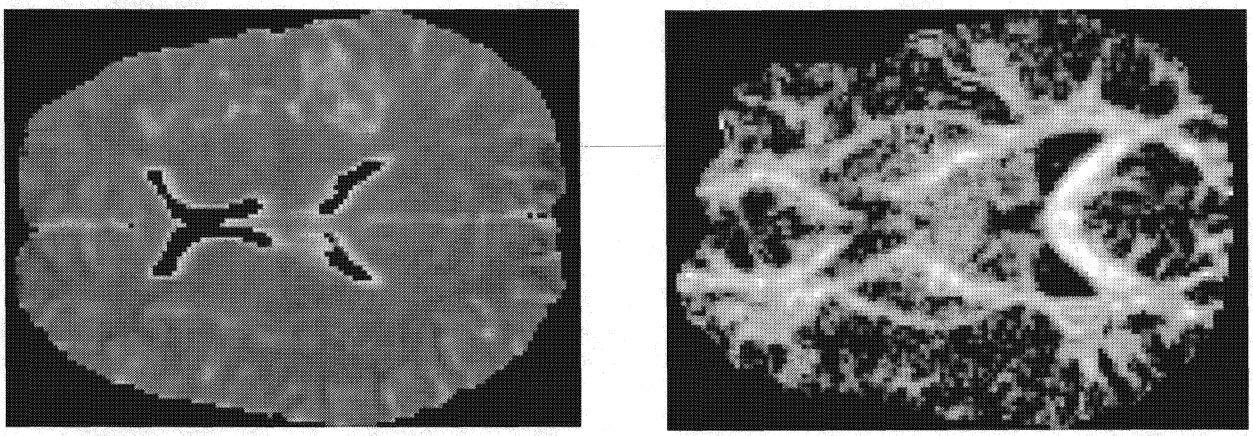

Figure 1. Slices from our two invariant volumes. a: $\mathcal{V}_1$ (D_1) (left). b: $\mathcal{V}_2$ (D_a) (right).

It is easy to show, that for isotropic diffusion, when $\lambda_1 = \lambda_2 = \lambda_3$, the coefficient $D_a = 1$. In the anisotropic case, this measure is identical for both linear, directional diffusion ($\lambda_1 >> \lambda_2 \approx \lambda_3$) and planar diffusion ($\lambda_1 \approx \lambda_2 >> \lambda_3$) and is equal to

$$D_a = \frac{1}{3}\left[1 + \frac{\lambda_1}{\lambda_3} + \frac{\lambda_3}{\lambda_1}\right]. \tag{9}$$

Thus D_a is always $\sim \lambda_{max}/\lambda_{min}$ and measures the magnitude of the diffusion's anisotropy. We again want to emphasize, that we use eigenvalue representation here only to analyze the behavior of the coefficient D_a, but we do not need eigenvalues to compute it.

3. EXTRACTING GEOMETRIC MODELS

We have used the invariants presented in the previous section (D_1, D_2, D_3) to extract geometric models from a $88 \times 121 \times 60$ DWI scan of a human head. We begin by generating two scalar volume datasets based on our invariants. The first scalar volume dataset $(\mathcal{V}_1)$ is formed by calculating the trace (D_1) of the tensor matrix for each voxel of the diffusion tensor volume. The second scalar volume dataset $(\mathcal{V}_2)$ is formed by calculating our three invariants for each voxel and combining them into D_a. The first scalar (trace) volume provides a single number that characterizes the total diffusivity at each voxel within the sample. Higher values signify greater total diffusion irrespective of directionality in the region represented by a particular voxel. A slice from this volume can be seen in Figure 1a. The second scalar calculation provides a measure of the magnitude of the anisotropy within the volume. Higher values identify regions of greater spatial anisotropy in the diffusion properties. A slice from the second scalar volume is presented in Figure 1b. The measure D_a does by definition not distinguish between linear and planar anisotropy. This is sufficient for our current study because the brain does not contain measurable regions with planar anisotropy. We therefore only need two scalar volumes in order to segment our DWI dataset.

These two derived scalar volumes may now be used to visualize structures with differing diffusion properties within the brain. Two options are available for viewing the scalar volume datasets, direct volume rendering[11,12] and volume segmentation[13] combined with conventional surface rendering. The first option, direct volume rendering, is only capable of supplying images of the data. While this method may provide useful views of the data, it is well-known that finding the exact transfer function that highlights the desired structures may be difficult to construct.[14] Our approach instead focuses on extracting geometric models of the structures imbedded in the volume datasets. The extracted models may be used for interactive viewing, but the segmentation of geometric models from the volume datasets provides a wealth of additional benefits and possibilities. The models may be used for quantitative analysis of the segmented structures, for example the calculation of surface area and volume; quantities that are important when studying how these structures change over time. The models may be used to provide the shape information necessary for computational simulation, e.g. EEG/MEG modeling within the brain.[15] Creating separate geometric models for each structure allows for the straightforward study of the relationship between the structures, even though they come from different datasets, as seen in Figures 6 and 7. The models may also be used within a surgical

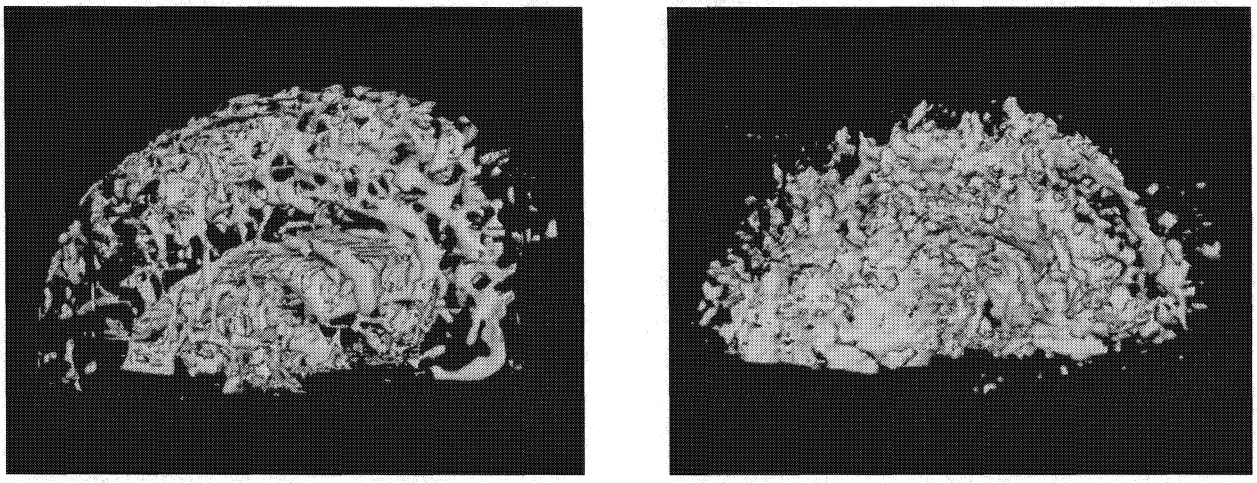

Figure 2. Meshes generated with the Marching Cubes algorithm on a: $\mathcal{V}_1$, iso-value = 8.75 (left); b: $\mathcal{V}_2$, iso-value = 1.36 (right).

planning/simulation/VR environment,[16] providing the shape information needed for collision detection and force calculations. The geometric models may even be used for manufacturing real physical models of the structures.[17] It is clear that there are numerous reasons to develop techniques for extracting geometric models from diffusion tensor volume datasets.

The most widely used technique for extracting polygonal models from volume datasets is the Marching Cubes algorithm.[18] This technique creates a polygonal model that approximates the iso-surface imbedded in a scalar volume dataset for a particular iso-value. The surface represents all the points within the volume that have the same scalar value. The polygonal surface is created by examining every "cube" of eight volume grid points and defining a set of triangles that approximates the piece of the iso-surface within the space bounded by the eight points. While the Marching Cubes algorithm is easy to understand and straightforward to implement, applying it directly to raw volume data from scanners can produce undesirable results, as seen in Figure 2. The algorithm is susceptible to noise and can produce many unwanted triangles that mask the central structures in the data. In order to alleviate this problem, we utilize a deformable model to smooth the data and remove the noise-related artifacts. Many types of models have been proposed for extracting structures from volumes.[13,19] We utilize level set models* because they have been shown to be flexible and effective for segmentation.[20–24] Most importantly, they easily change topology during deformation and have no fixed parameterization, allowing them to represent complex shapes.

Level set methods produce active deformable surfaces that may be directed to conform to features in a volume dataset while simultaneously applying a smoothing operation based on local surface curvature. The surfaces are viewed as a specific level set of a higher-dimensional function ϕ – hence the name level set methods.[25] These methods provide the mathematical and numerical mechanisms for computing surface deformations as iso-values of ϕ by solving a partial differential equation on the 3D grid.[26] That is, the level set formulation provides a set of numerical methods that describes how to manipulate the greyscale values in a volume, so that the iso-surfaces of ϕ move in a prescribed manner. ϕ is a dynamic function of $(x, y, z) \equiv \boldsymbol{s}$, that changes over time,

$$\phi(\boldsymbol{s}, t) = k. \tag{10}$$

To transform this definition into a partial differential equation that can easily be solved by standard numerical techniques, we differentiate both sides of Equation 10 with respect to time t, and apply the chain rule:

$$\frac{\partial \phi(\boldsymbol{s}, t)}{\partial t} + \nabla \phi(\boldsymbol{s}, t) \cdot \frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = 0. \tag{11}$$

$\mathrm{d}\boldsymbol{s}/\mathrm{d}t$ can be thought of as the movement of a point on a surface as it deforms, and can be expressed in terms of the position of $\boldsymbol{s}$ and the geometry of the surface at that point, which is, in turn, a differential expression of the implicit

*The VISPACK libraries were used in this study. See http://www.cs.utah.edu/~whitaker/vispack.

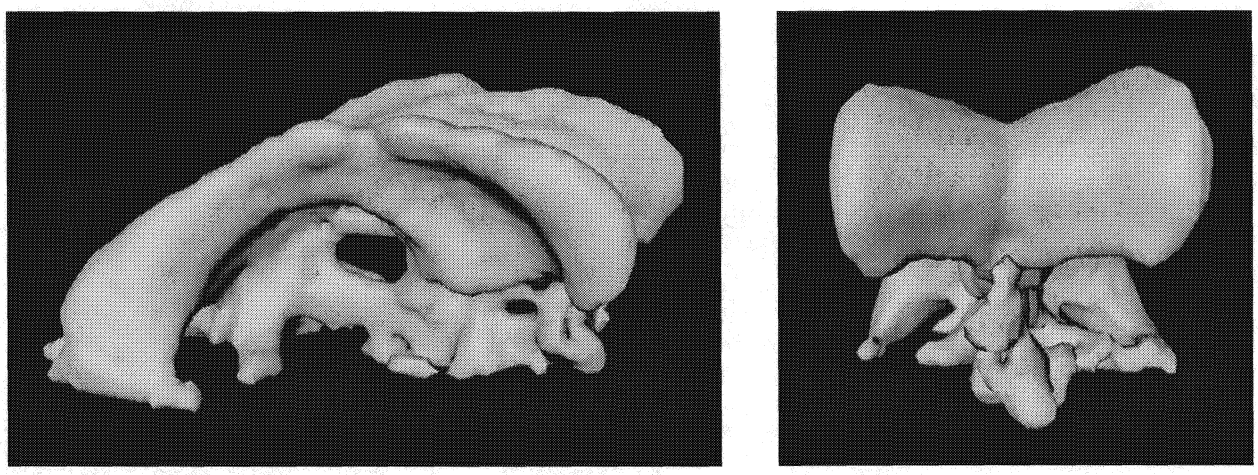

Figure 3. Brain ventricles extracted from volume $\mathcal{V}_1$. a: slightly oblique view (left). b: coronal,front view (right).

function ϕ. This produces another partial differential equation on ϕ,

$$\frac{\partial \phi}{\partial t} = -\nabla\phi \cdot \frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} \equiv -\nabla\phi \cdot \boldsymbol{F}(\boldsymbol{s}, \mathrm{D}\phi, \mathrm{D}^2\phi, \ldots), \tag{12}$$

where $\boldsymbol{F}$ is a user-defined "speed" term which depends on a set of order-n derivatives of ϕ, $D^n\phi$, evaluated at $\boldsymbol{s}$, as well as other functions of $\boldsymbol{s}$. Because this relationship applies to every level set of ϕ, i.e. all values of k, this equation can be applied to the whole dataset, and therefore the movements of *all* the level set surfaces embedded in ϕ can be calculated from Equation 12.

For our application $\boldsymbol{F}$ includes a term that attracts the surface to regions of high contrast, which are effectively "edges" in the volume. This can be done by calculating the gradient of the gradient magnitude of the scalar volume datasets. $G(\boldsymbol{x})$ is some greyscale feature, for instance $G(\boldsymbol{x}) = |\nabla I(\boldsymbol{x})|$, where $I(\boldsymbol{x})$ is the input data (appropriately filtered—we use Gaussian kernels with $\sigma \approx 0.5$). Then

$$\boldsymbol{F}(\boldsymbol{x}) = \pm\nabla G(\boldsymbol{x}), \tag{13}$$

where a positive sign moves the surfaces towards maxima and the negative sign towards minima. The smoothing term uses the mean curvature of the iso-surface H to form a vector in the direction of the surface normal $\boldsymbol{n}$ given by

$$\boldsymbol{F}(\boldsymbol{x}) = H\boldsymbol{n} = \left(\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}\right)\frac{\nabla\phi}{|\nabla\phi|}. \tag{14}$$

The smoothing term is weighted by a factor β, allowing the user to control the amount of smoothing, and is tuned for each dataset.

4. MODELING RESULTS

As stated in the previous section, we have generated two new scalar volumes ($\mathcal{V}_1$ and $\mathcal{V}_2$) by calculating and combining our invariants based on a DW MRI scan of a human head. $\mathcal{V}_1$ measures total diffusivity, and $\mathcal{V}_2$ is the combination D_a at each voxel which provides a measure of the level of anisotropy in the diffusion. We have extracted geometric models from these two scalar volumes utilizing our modeling and segmentation techniques. While meshes extracted with the Marching Cubes algorithm may provide the user with an understanding of the basic structure of the data and the noise included in it (See Figure 2), frequently a smoother, closed model is desirable. Figures 3 and 4 present typical models that can be extracted from DW MRI volume datasets using our techniques. Figure 3 presents the ventricles of the brain. This model was extracted from $\mathcal{V}_1$ using both manual/interactive segmentation tools implemented in our modeling system[27] and the computational level set models described in Section 3. Figures 4 and 5 present the model extracted from $\mathcal{V}_2$ with our level set model. These figures include regions of high diffusion anisotropy, for

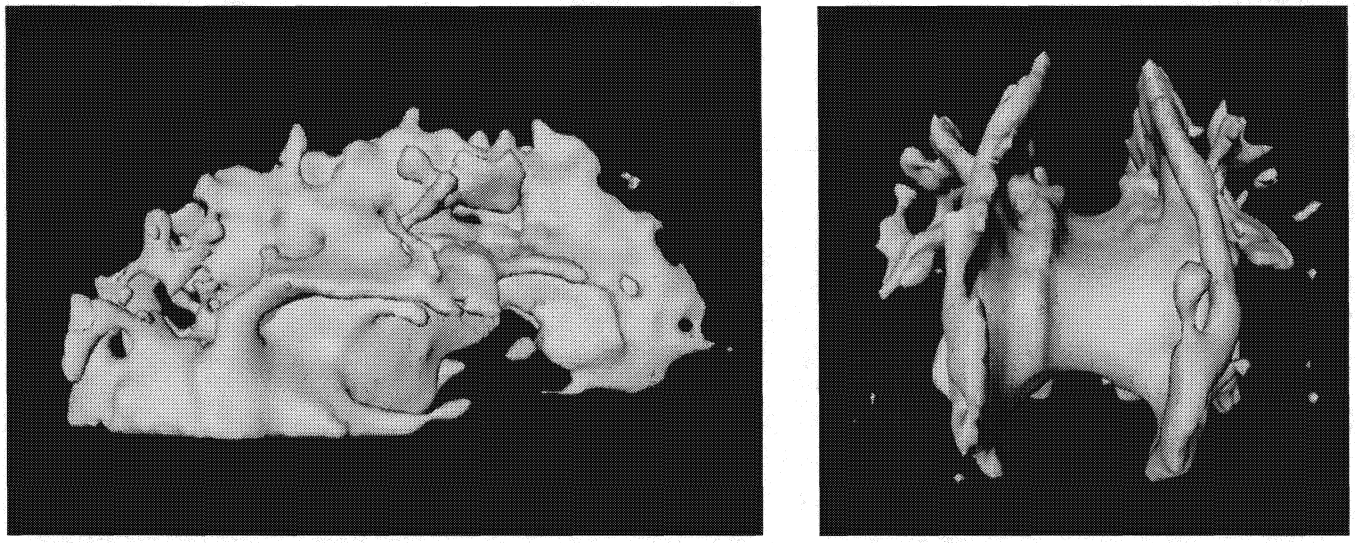

Figure 4. Model of anisotropic regions extracted from volume $\mathcal{V}_2$. a: sagittal view (left). b: coronal,front view (right).

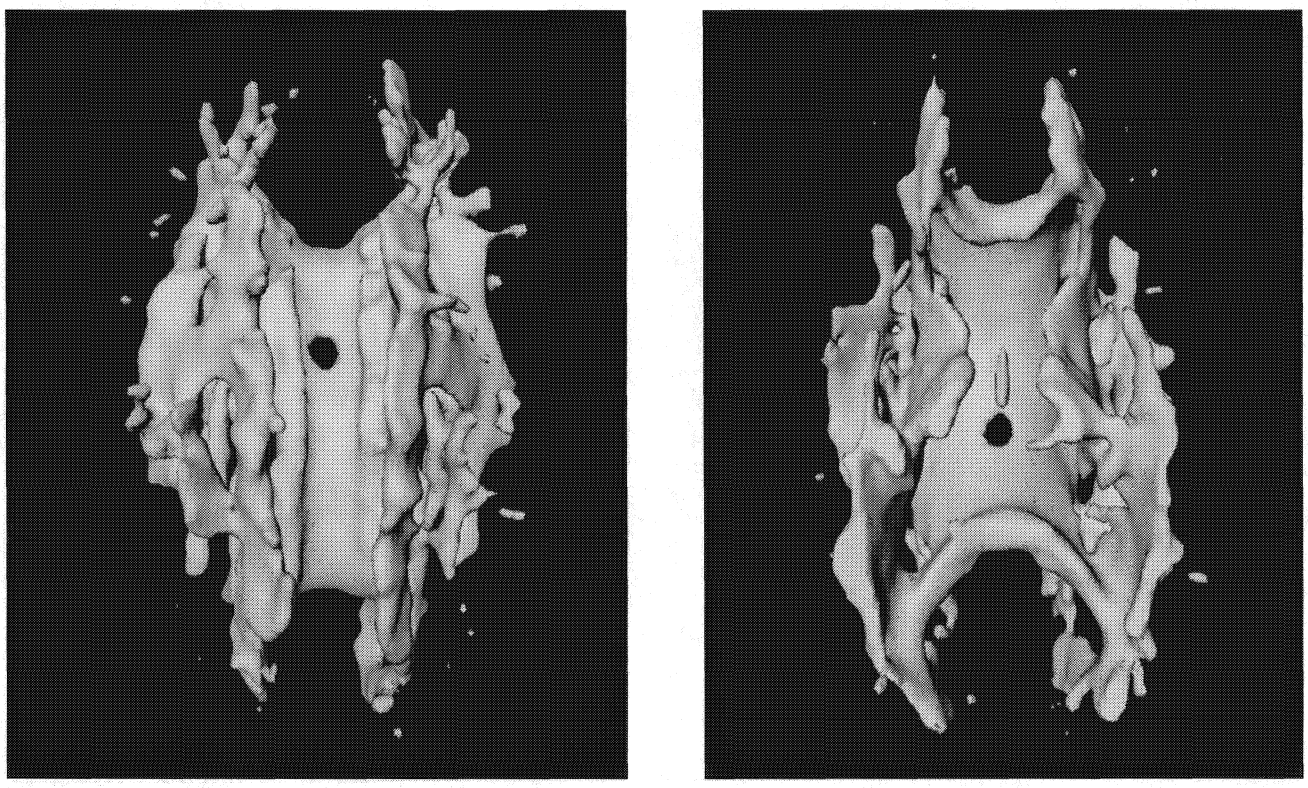

Figure 5. Model of anisotropic regions extracted from volume $\mathcal{V}_2$. a: axial,top view (left). b: axial,bottom view (right).

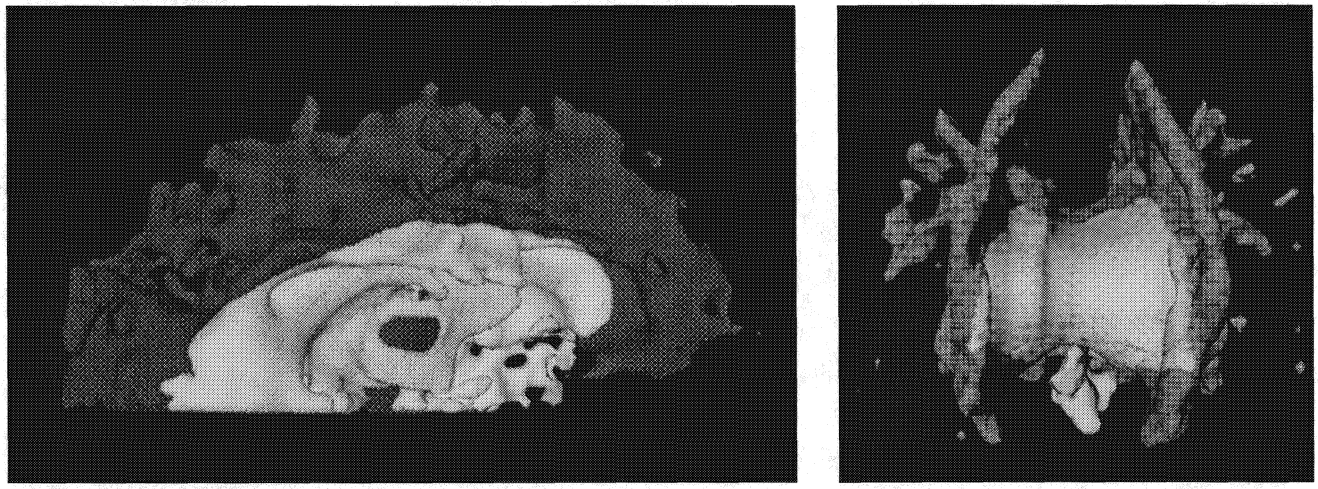

Figure 6. Combined model of ventricles and (semi-transparent) anisotropic regions. a: sagittal view (left). b: coronal,front view (right).

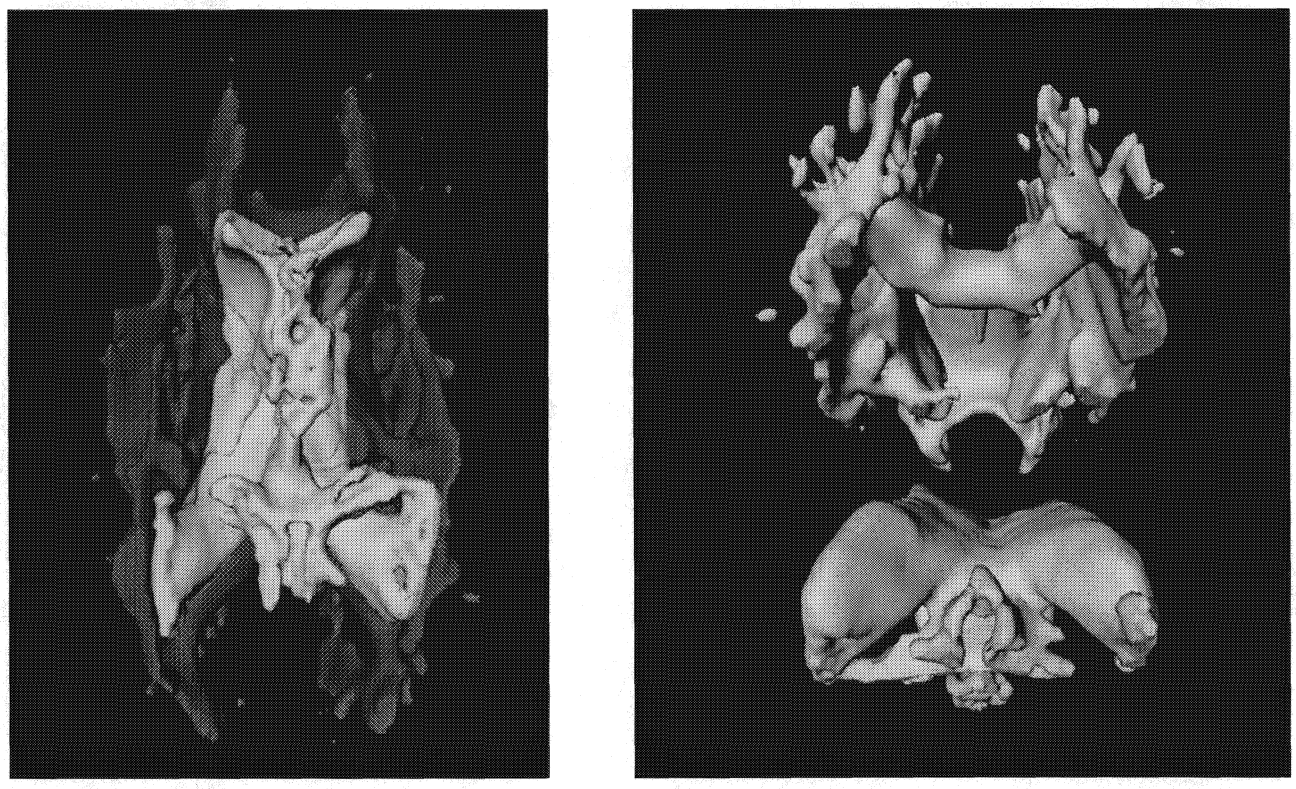

Figure 7. Combined model of ventricles and (semi-transparent) anisotropic regions. a: axial,bottom view (left). b: coronal,rear,exploded view (right).

Table 1. CPU-seconds on an SGI Onyx2 (250MHz R10K) for computing three different types of tensor invariants.

Mesh size $N_x = N_y = N_z$	Present method Invariants.	Eigenvalues Solve cubic eq.	Eigenvectors EISPACK
64	0.043	0.467	2.756
128	0.403	4.105	22.085
256	3.229	32.874	176.087
512	25.774	262.591	1408.651

example the corpus callosum, the internal capsul, the optical nerve tracks, and other white matter regions. These are areas of the brain with numerous nerve tracks that restrict diffusion, which are easily isolated with our techniques.

We can also bring together the two models extracted from $\mathcal{V}_1$ and $\mathcal{V}_2$ into a single image. Figures 6 and 7 demonstrate that we are able to isolate different structures in the brain and show their proper spatial inter-relationship. For example, it can be seen that the corpus callosum lies directly on top of the ventricles, and that the white matter fans out from both sides of the ventricles.

5. NUMERICAL COMPUTATIONS

The tensor invariants presented in Section 2 have the beneficial property that they are much faster to compute than the conventional eigenvalue/eigenvector invariants normally used to analyze diffusion tensor data. This is clear from Table 1 where we have listed CPU times for computing the three different types of tensor invariants mentioned in this paper for different sizes of the volume dataset. The three timings scale almost exactly as 2^3, which of course reflects the fact they measure computation of the invariants exclusively. It follows from this table that our proposed invariants are one order of magnitude faster to compute than the eigenvalues and approximately 50 times faster than the calculation of eigenvectors. For the last comparison it should be emphasized that the new invariants do *not* include directional information, as does the eigenvector calculation. However, it should be equally clear that when directional information is not desired our method offers a significant speedup over the conventional approach – especially for large datasets.

The Marching Cubes algorithm is applied to the volumes generated by the level set stage. We may then compute useful quantitative information, e.g. total area, volume, and Gaussian curvature, from the resulting polygonal model. The Marching Cubes algorithm produces a triangle mesh consisting of vertices and associated normal vectors. Thus, the total area of the mesh is computed by adding the areas, A_i, of each triangle

$$A_i = \frac{1}{2}|(\boldsymbol{v}_i^1 - \boldsymbol{v}_i^2) \times (\boldsymbol{v}_i^1 - \boldsymbol{v}_i^3)|, \tag{15}$$

where $\boldsymbol{v}_i^k$ is the k'th vertex of triangle i. The calculation of volume for the polygonal model is not as straightforward. First we assume that all of the extracted models are composed of *closed* polygonal surfaces. Next, using a simple idea by Eberly *et al.*[28] one can convert the volume calculation to a surface integral by means of the divergence theorem,

$$\int\int\int_{\mathcal{V}} \boldsymbol{\nabla} \cdot \boldsymbol{F}\, dV = \int\int_{\mathcal{S}} \boldsymbol{F} \cdot \boldsymbol{n}\, dS, \tag{16}$$

where $\boldsymbol{\nabla} \cdot \boldsymbol{F}$ is the divergence of an arbitrary vector function $\boldsymbol{F}$ and $\mathcal{V}$ is a volume in $\mathcal{R}^3$ enclosed by the surface $\mathcal{S}$ with associated normal vectors $\boldsymbol{n}$ *pointing outwards*. If we choose $\boldsymbol{F} = (\alpha_x x, \alpha_y y, \alpha_z z)$ such that $\alpha_x + \alpha_y + \alpha_z = 1$, then $\boldsymbol{\nabla} \cdot \boldsymbol{F} = 1$ and the left hand side of Equation 16 becomes the desired volume, V. Using standard integration theory, we can approximate the right hand side of Equation 16 with the Reimann sum,

$$V \approx \sum_{i=1}^{N_{poly}} \boldsymbol{v}_i \cdot \boldsymbol{n}_i A_i \tag{17}$$

where $\boldsymbol{v}_i \equiv (\alpha_x x_i, \alpha_y y_i, \alpha_z z_i)$ is a point on the i'th triangle with the associated normal vector $\boldsymbol{n}_i$ *pointing outwards* and the area A_i given by Equation 15. As stressed by Eberly *et al.* the choice of α_x, α_y, and α_z is crucial in

Table 2. Values of volume V, surface area A, total norm of the Gaussian curvature $|\overline{\mathcal{K}}|$ and total polygon count N_{poly} before and after the application of our level set model.

Data set	$V(cm^3)$	$A(cm^2)$	$\|\overline{\mathcal{K}}\|$	N_{poly}
Before($\mathcal{V}_1$)	18	231	294	39404
After($\mathcal{V}_1$)	33	203	152	36664
Before($\mathcal{V}_2$)	98	760	2673	142212
After($\mathcal{V}_2$)	87	473	188	81488

obtaining a good estimate of the volume. They describe a scheme for assigning values of the α_k based on an estimated bounding box around each closed surface, but this requires additional mesh connectivity information. Such information can of course be computed, but in our study where the Marching Cubes algorithm can generate very large numbers of closed surfaces this becomes very tedious and time-consuming. Instead, let us consider the special case $\boldsymbol{v}_i = \frac{1}{3}(x_i, y_i, z_i) \equiv \frac{1}{3}\boldsymbol{C}_i$ (i.e. $\alpha_k = \frac{1}{3}$), where $\boldsymbol{C}_i$ is chosen as the centroid of the i'th triangle. This leads to the expression

$$V \approx \frac{1}{6} \sum_{i=1}^{N_{poly}} \boldsymbol{C}_i \cdot \boldsymbol{N}_i \tag{18}$$

where we have defined the local vectors

$$\boldsymbol{C}_i = (\boldsymbol{v}_i^1 + \boldsymbol{v}_i^2 + \boldsymbol{v}_i^3)/3 \tag{19}$$

$$\boldsymbol{N}_i = (\boldsymbol{v}_i^1 - \boldsymbol{v}_i^2) \times (\boldsymbol{v}_i^1 - \boldsymbol{v}_i^3). \tag{20}$$

The geometric interpretation of Equation 18 is a signed sum of the pyramids with a base composed of the i'th triangle and a top vertex at the origin. We finally stress that the derived expression for an estimate of the total volume does *not* assume any knowledge of the connectivity of the triangular mesh.

Finally, in order to obtain a quantitative measure of the smoothness of the extracted triangular mesh we calculate a norm of the total Gaussian curvatures as follows. At each vertex the local Gaussian curvatures $\mathcal{K}$ is given by

$$\mathcal{K} = \frac{1}{(\dot{\phi}_x^2 + \dot{\phi}_y^2 + \dot{\phi}_z^2)^2} \left\{ \begin{array}{l} \dot{\phi}_x^2(\ddot{\phi}_{yy}\ddot{\phi}_{zz} - \ddot{\phi}_{yz}\ddot{\phi}_{yz}) + 2\dot{\phi}_y\dot{\phi}_z(\ddot{\phi}_{xz}\ddot{\phi}_{xy} - \ddot{\phi}_{xx}\ddot{\phi}_{yz}) + \\ \dot{\phi}_y^2(\ddot{\phi}_{xx}\ddot{\phi}_{zz} - \ddot{\phi}_{xz}\ddot{\phi}_{xz}) + 2\dot{\phi}_x\dot{\phi}_z(\ddot{\phi}_{yz}\ddot{\phi}_{xy} - \ddot{\phi}_{yy}\ddot{\phi}_{xz}) + \\ \dot{\phi}_z^2(\ddot{\phi}_{xx}\ddot{\phi}_{yy} - \ddot{\phi}_{xy}\ddot{\phi}_{xy}) + 2\dot{\phi}_x\dot{\phi}_y(\ddot{\phi}_{xz}\ddot{\phi}_{yz} - \ddot{\phi}_{zz}\ddot{\phi}_{xy}) \end{array} \right\}, \tag{21}$$

where for example $\ddot{\phi}_{xz}$ denotes the second order partial derivative of the volume with respect to the directions x and z.[25] The first and second order partial derivatives in Equation 21 can be approximated by finite-difference schemes. To obtain a measure of the total smoothness of the triangular mesh we found it useful to define the following total norm of the Gaussian curvature

$$|\overline{\mathcal{K}}| \equiv \frac{1}{4\pi} \sum_{i=1}^{N_{poly}} |\overline{\mathcal{K}}_i| A_i \tag{22}$$

where for each triangle i we define the average Gaussian curvature from the associated three vertices

$$\overline{\mathcal{K}}_i = (\mathcal{K}_i^{(1)} + \mathcal{K}_i^{(2)} + \mathcal{K}_i^{(3)})/3. \tag{23}$$

Equation 22 is an approximation of a special case of the Gauss-Bonnet theorem[29] that states that the surface integral of the Gaussian curvature (with sign) equals $2\pi E$ where E is the Euler-Pointcaré characteristic of the smooth surface ($E = 2$ for closed surfaces). Hence, to calculate a measure of the overall curvature, and not just the total number of closed surfaces, Equation 22 employs the absolute value of the Gaussian curvature.

Table 2 lists values of the total volume, surface area, norm of the Gaussian curvature and polygon count for the models extracted from our two derived scalar volume datasets ($\mathcal{V}_1$ and $\mathcal{V}_2$), before and after the level set algorithm is applied to the volumes. Comparing the first two rows for volume $\mathcal{V}_1$, we note that after the level set algorithm is employed the total volume increases, which is due to the closing of holes and cavities in the extracted surface.

Meanwhile, the surface area decreases, which is caused by the removal of internal structures. We also note that the total norm of the Gaussian curvature decreases due to the smoothing imposed by the level set model. Additionally, the polygon count drops, because of the simplified form of the final extracted triangular mesh. A similar behavior is observed when processing volume $\mathcal{V}_2$ (the last two rows of Table 2), except that the volume now decreases. This is caused by the removal (i.e. collapse) of small high frequency fragments cluttering the model.

6. CONCLUSION

In this paper we have proposed an alternative set of invariants that may be derived from diffusion tensors. We have shown that they are faster to compute than previous methods. These invariants may be used to generate scalar volumes that characterize the total diffusivity and diffusion anisotropy of a DW MRI scan of a human head. Applying modeling and segmentation techniques to the derived scalar volumes creates geometric models of specific brain structures, e.g. the ventricles, corpus callosum, and the internal capsul. The geometric models may then be used for quantitative analysis, e.g. volume and surface area calculations.

ACKNOWLEDGMENTS

We would like to thank Dr. J. Michael Tyszka and Dr. Miriam Scadeng for helping us to identify the 3-D structures extracted from the DWI dataset and for patiently teaching us neuro-anatomy. We would also like to thank Dr. Alan Barr for his support of this project. This work was supported by National Science Foundation grants #ACI-9982273 and #ASC-89-20219, the National Institute on Drug Abuse, the National Institute of Mental Health and the NSF, as part of the Human Brain Project, Office of Naval Research Volume Visualization grant #N000140110033, and the National Library of Medicine "Insight" Project #N01-LM-0-3503. The DWI dataset is courtesy of the University of Utah SCI Institute.

REFERENCES

1. T.G. Reese, R.M. Weisskoff, R.N. Smith, B.R. Rosen, R.E. Dinsmore and V.J. Wedeen. "Imaging Myocardial Fiber Architecture In Vivo with Magnetic Resonance". *Magn. Reson. Med.*, **34(6)**, pp 786-91, 1995.
2. C. Pierpaoli, P. Jezzard, P. J. Basser, A. Barnett and G. Di Chiro. "Diffusion Tensor MR Imaging of the Human Brain". *Radiology*, **201(3)**, pp 637-648, 1996.
3. P. J. Basser and C. Pierpaoli. "Microstructural and Physiological Features of Tissues Elucidated by Quantitative-Diffusion-Tensor MRI". *J. Magn. Reson. Ser. B*, **111**, pp 209-219, 1996.
4. C-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis and F. A. Jolesz. "Geometrical Diffusion Measures for MRI from Tensor Basis Analysis". *In Proc. ISMRM 5th Annual Meeting*, Vancouver, Canada, p 1742, 1997.
5. S. Peled, H. Gudbjartsson, C.F. Westin, R. Kikinis and F.A. Jolesz. "Magnetic Resonance Imaging Shows Orientation and Assymetry in White Matter Fiber Tracts". *Brain Research*, **780**, pp 27-33, 1998.
6. D.H. Laidlaw, E.T. Ahrens, D. Kremers, M.J. Avalos, R.E. Jacobs and C. Readhead. "Visualizing Diffusion Tensor Images of the Mouse Spinal Cord". *In Proc. IEEE Visualization 98*, pp 127-134, 1998.
7. G. Kindlmann and D. Weinstein. "Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields". *In Proc. IEEE Visualization 99*, pp 183-189, 1999.
8. P.J. Basser, S. Pajevic, C. Pierpaoli, J. Duda and A. Aldroubi. "In Vivo Fiber Tractography Using DT-MRI Data". *Magn. Reson. Med.*, **44**, pp 625-632, 2000.
9. C. Poupon, C.A. Clark, V. Frouin, J. Regis, I. Bloch, D. Le Bihan and J.-F Mangin. "Regularization of Diffusion-Based Direction Maps for the Tracking of Brain White Matter Fascicles". *Neuroimage*, **12**, pp 184-195, 2000.
10. T.M. Korn, A. Granino. *Mathematical Handbook for Scientists and Engineers.* McGraw-Hill, 1961.
11. R.A. Drebin, L. Carpenter and P. Hanrahan. "Volume Rendering". *Computer Graphics*, **22(4)**, pp 65-74, 1988.
12. S. Parker, M. Parker, Y. Livnat, P. Sloan, C. Hansen and P. Shirley. "Interactive Ray Tracing for Volume Visualization". *IEEE Trans. on Viz. and Comp. Graph.*, **5(3)**, pp 238-250, 1999.
13. *Deformable Models in Medical Image Analysis*, A. Singh, D. Goldgof and D. Terzopoulos (eds.). IEEE Computer Society Press, Los Alamitos, CA, 1998.
14. G. Kindlmann and J.W. Durkin. "Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering". *In Proc. IEEE Volume Visualization 98*, pp 79-86, 1998.

15. L.E. Zhukov, D.M. Weinstein and C.R. Johnson. "Independent Component Analysis for EEG Source Localization in Realistic Head Model". *IEEE Engineering in Medicine and Biology*, **19**, pp 87-96, 2000.
16. S. Gibson, et al. "Volumetric Object Modeling for Surgical Simulation". *Medical Image Analysis*, **2(2)**, pp 121-132, 1998.
17. M. Bailey. "Manufacturing Isovolumes". *In Volume Graphics*, M. Chen, A.E. Kaufman and R. Yagel (eds.), Springer, London, chapter 5, pp 79-93, 2000.
18. W.E. Lorensen and H.E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". *Computer Graphics*, **21(3)**, pp 163-169, 1987.
19. J. Miller, D. Breen, W. Lorensen, R. O'Bara and M. Wozny. "Geometrically Deformed Models: A Method for Extracting Closed Geometric Models from Volume Data". *Computer Graphics*, **25(4)**, pp 217-226, 1991.
20. R.T. Whitaker and D.T. Chen. "Embedded Active Surfaces for Volume Visualization". *In Proc. SPIE Medical Imaging*, pp 340-352, 1994.
21. R. Malladi, J. Sethian and B. Vemuri. "Shape Modeling with Front Propagation: A Level Set Approach". *IEEE Trans. PAMI*, **17(2)**, pp 158-175, 1995.
22. M. Leventon and O. Faugeraus and W. Grimson and W. Wells III. "Level Set Based Segmentation with Intensity and Curvature Priors". *In Proc. Mathematical Methods in Biomedical Image Analysis*, pp 4-11, 2000.
23. L. Staib, X. Zeng, R. Schultz and J. Duncan. "Shape Constraints in Deformable Models". *In Handbook of Medical Imaging*, I. Bankman (ed.), Academic Press, chapter 9, pp 147-157, 2000.
24. R.T. Whitaker, D.E. Breen, K. Museth and N. Soni. "A Framework for Level Set Segmentation of Volume Datasets". Submitted to Volume Graphics 2001.
25. *Level Set Methods and Fast Marching Methods*, J.A. Sethian. Cambridge University Press, Cambridge, UK, 1999.
26. S. Osher and J. Sethian. "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations". *Journal of Computational Physics*, **79**, pp 12-49, 1988.
27. P. Getto and D. Breen. "An Object-Oriented Architecture for a Computer Animation System". *The Visual Computer*, **6(2)**, pp 79-92, 1990.
28. D. Eberly, J. Lancaster, and A. Alyassin. "On Gray Scale Image Measurements". *Graphics Models and Image Processing*, **56(6)**, pp 550-62, 1991.
29. *Modern Differential Geometry of Curves and Surfaces with Mathematica*, A. Gray. CRC Press, 1998.