

Level set modeling and segmentation of diffusion tensor magnetic resonance imaging brain data

Leonid Zhukov
Ken Museth
David Breen
Alan H. Barr

California Institute of Technology
Department of Computer Science
Mail Code 350-74
Pasadena, California 91125-7400
E-mail: {zhukov,kmu,david,barr}@gg.caltech.edu

Ross Whitaker
University of Utah
School of Computing
3190 MEB
Salt Lake City, Utah 84112-9205

Abstract. Segmentation of anatomical regions of the brain is one of the fundamental problems in medical image analysis. It is traditionally solved by iso-surfacing or through the use of active contours/deformable models on a gray-scale magnetic resonance imaging (MRI) data. We develop a technique that uses anisotropic diffusion properties of brain tissue available from diffusion tensor (DT)-MRI to segment brain structures. We develop a computational pipeline starting from raw diffusion tensor data through computation of invariant anisotropy measures to construction of geometric models of the brain structures. This provides an environment for user-controlled 3-D segmentation of DT-MRI datasets. We use a level set approach to remove noise from the data and to produce smooth, geometric models. We apply our technique to DT-MRI data of a human subject and build models of the isotropic and strongly anisotropic regions of the brain. Once geometric models have been constructed they can be combined to study spatial relationships and quantitatively analyzed to produce the volume and surface area of the segmented regions. © 2003 SPIE and IS&T. [DOI: 10.1117/1.1527628]

1 Introduction

Diffusion tensor magnetic resonance imaging^{1–4} (DT-MRI) is a technique used to measure the diffusion properties of water molecules in tissues. Anisotropic diffusion can be described by the equation

$$\frac{\partial C}{\partial t} = \nabla \cdot (\mathbf{D} \nabla C), \quad (1)$$

where C is the concentration of water molecules, and \mathbf{D} is a diffusion coefficient, which is a symmetric second-order tensor

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix}. \quad (2)$$

Figure 1 presents a “slice” of the diffusion tensor volume data of human brain used in our study. Each subimage presents the scalar values of the associated diffusion tensor component for one slice of the dataset.

Tissue segmentation and classification based on DT-MRI offers several advantages over conventional MRI, since diffusion data contains additional physical information about the internal structure of the tissue being scanned. However, segmentation and visualization using diffusion data is not entirely straightforward. First, the diffusion matrix itself is not invariant with respect to rotations, and the elements that form the matrix will be different for different orientations of the sample or field gradient and therefore cannot themselves be used for classification purposes. Moreover, 3-D visualization and segmentation techniques available today are predominantly designed for scalar and sometimes vector fields. Thus, there are two fundamental problems in tensor imaging: (1) 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 (2) visualization and classification of tissue using the derived scalar fields.

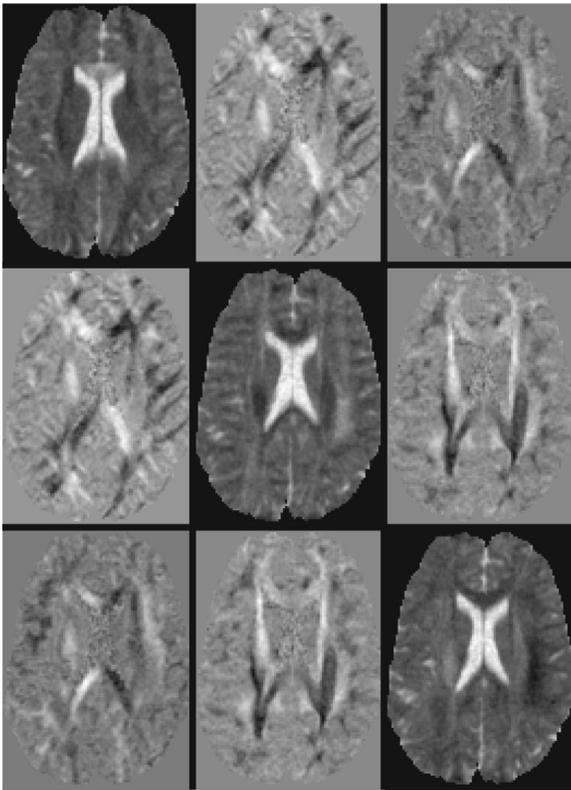


Fig. 1 Slice of a tensor volume where every “element” of the image matrix corresponds to one component of the tensor \mathbf{D} .

The traditional approaches to diffusion tensor imaging involve converting the tensors into an eigenvalue/eigenvector representation, which is rotationally invariant. Every tensor can 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.

Using eigenvalues and eigenvectors one can compute different anisotropy measures^{7,5-8} that map tensor data onto scalars and can be used for further visualization and segmentation. Although eigenvalue/vector computation of the 3×3 matrix is not expensive, it must be repeatedly performed for every voxel in the volume. This calculation easily becomes a bottleneck for large datasets. For example, computing eigenvalues and eigenvectors for a 512^3 volume requires over 20 CPU min on a powerful workstation. Another problem associated with eigenvalue computation is stability—a small amount of noise will change not only the values but also the ordering of the eigenvalues.⁹ Since many anisotropy measures depend on the ordering of the eigenvalues, the calculated direction of diffusion and classification of tissue will be significantly altered by the noise normally found in diffusion tensor datasets. Thus it is desirable to have an anisotropy measure that is rotationally invariant, does not require eigenvalue computations, and is stable with respect to noise. The tensor invariants with these characteristics were first proposed by Ulug and Zijl.¹⁰ In Sec. 2 of this paper we formulate a new anisotropy measure for tensor field based on these invariants.

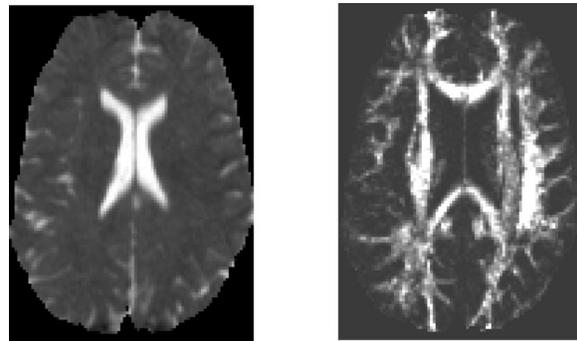


Fig. 2 Isotropic C_1 (left) and anisotropic C_a (right) tensor invariants for the tensor slice shown in Fig. 1.

Visualization and model extraction from the invariant 3-D scalar fields is the second issue addressed in this paper. One of the popular approaches to tensor visualization represents a tensor field by drawing ellipsoids associated with the eigenvectors/values.¹¹ This method was developed for 2-D slices and creates visual cluttering when used in three dimensions. Other standard computational fluid dynamics (CFD) visualization techniques such as tensor lines do not provide meaningful results for the MRI data due to rapidly changing directions and magnitudes of eigenvector/values and also amount of noise present in the data. Recently Kindlmann and Weinstein¹² developed a volume-rendering approach to tensor field visualization using eigenvalue-based anisotropy measures to construct transfer function and color maps, that highlight some brain structures and diffusion patterns.

In our work, we perform iso-surfacing on the 3-D 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. A detailed discussion of the modeling method is presented in Sec. 3. Section 4 presents the results of tensor-invariant calculations and model segmentation technique with examples from a DT-MRI scan of a human head. Section 5 then describes the quantitative analysis of obtained geometric models.

Finally, a number of recent publications^{13,14} have been devoted to brain 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 better SNR 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 DT imaging. The well-known invariants are the eigenvalues of diffusion tensor (matrix) \mathbf{D} , which are the roots of corresponding characteristic equation

$$\lambda^3 - C_1 \cdot \lambda^2 + C_2 \cdot \lambda - C_3 = 0, \quad (3)$$

with coefficients

$$\begin{aligned}
 C_1 &= D_{xx} + D_{yy} + D_{zz}, \\
 C_2 &= D_{xx}D_{yy} - D_{xy}D_{yx} + D_{xx}D_{zz} - D_{xz}D_{zx} + D_{yy}D_{zz} \\
 &\quad - D_{yz}D_{zy}, \\
 C_3 &= D_{xx}(D_{yy}D_{zz} - D_{zy}D_{yz}) - D_{xy}(D_{yx}D_{zz} - D_{zx}D_{yz}) \\
 &\quad + D_{xz}(D_{yx}D_{zy} - D_{zx}D_{yy}).
 \end{aligned} \tag{4}$$

Since the roots of Eq. (3) are rotational invariants, the coefficients C_1 , C_2 , and C_3 are also invariant. In the eigen frame of reference they can be easily expressed through the eigenvalues

$$\begin{aligned}
 C_1 &= \lambda_1 + \lambda_2 + \lambda_3, \\
 C_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3, \\
 C_3 &= \lambda_1\lambda_2\lambda_3,
 \end{aligned} \tag{5}$$

and are proportional to the sum of the radii, surface area and the volume of the “diffusion” ellipsoid. Then instead of using $(\lambda_1, \lambda_2, \lambda_3)$ to describe the dataset, we can use (C_1, C_2, C_3) . Moreover, since C_i are the coefficients of characteristic equation, they are less sensitive to noise, then roots λ_i of the same equation.¹¹

Any combination of the preceding invariants is, in turn, an invariant. We consider the following dimensionless combination: C_1C_2/C_3 . In the eigenvector frame of reference, it becomes

$$\frac{C_1C_2}{C_3} = 3 + \frac{\lambda_2 + \lambda_3}{\lambda_1} + \frac{\lambda_1 + \lambda_3}{\lambda_2} + \frac{\lambda_1 + \lambda_2}{\lambda_3}, \tag{6}$$

and we can define a new dimensionless anisotropy measure

$$C_a = \frac{1}{6} \left(\frac{C_1C_2}{C_3} - 3 \right). \tag{7}$$

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

$$C_a^{\text{limit}} \approx \frac{1}{3} \left(1 + \frac{\lambda_1}{\lambda_3} + \frac{\lambda_3}{\lambda_1} \right). \tag{8}$$

Thus C_a is always $\sim \lambda_{\text{max}}/\lambda_{\text{min}}$ and measures the magnitude of the diffusion anisotropy. Note that we use eigenvalue representation here only to analyze the behavior of the coefficient C_a , but we use invariants (C_1, C_2, C_3) to compute it using Eqs. (5) and (7). Isotropic C_1 and anisotropic C_a tensor invariants maps for the data slice from Fig. 1 is shown in Fig. 2.

3 Geometric Modeling

Two options are usually available for viewing the scalar volume datasets, direct volume rendering^{15,16} and volume

segmentation¹⁷ combined with conventional surface rendering. The first option, direct volume rendering, is capable of supplying only images of the data. While this method may provide useful views of the data, it is well known that it is difficult to construct the exact transfer function that highlights the desired structures in the volume dataset.¹⁸ Our approach instead focuses on extracting geometric models of the structures embedded in the volume datasets. The extracted models can 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 can 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 anatomical studies and computational simulation, for example, electroencephalogram/magnetoencephalogram (EEG/MEG) modeling within the brain.¹⁹ Creating separate geometric models for each structure enables the straightforward study of the relationship between the structures, even though they come from different datasets. The models can also be used within a surgical planning/simulation/VR environment,²⁰ providing the shape information needed for collision detection and force calculations. The geometric models can even be used for manufacturing real physical models of the structures.²¹ 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.²² This technique creates a polygonal model that approximates the iso-surface embedded 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 top row images in Figs. 4 and 7. The algorithm is susceptible to noise and can produce many unwanted triangles that mask the central structures in the data. To alleviate this problem, we utilize a deformable model approach to smooth the data and remove the noise-related artifacts. Many types of deformable models have been proposed for extracting structures from volumes.^{17,23} We utilize level set models as they have been shown to be flexible and effective for segmentation.^{24–28} 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.²⁸ Most importantly, they easily change topology during deformation and have no fixed parameterization, enabling them to represent complex shapes.

3.1 Level Set Models

A level set model^{29,30} specifies a surface as a level set (iso-surface) of a scalar volumetric function, $\phi: U \mapsto \mathfrak{R}$, where $U \subset \mathfrak{R}^3$ is the range of the surface model. Thus, a surface S is

$$S = \{\mathbf{s} \mid \phi(\mathbf{s}) = k\}, \quad (9)$$

and k is the isovalue. In other words, S is the set of points \mathbf{s} in \mathfrak{R}^3 that composes the k 'th iso-surface of ϕ . The embedding ϕ can be specified as a regular sampling on a rectilinear grid. The surfaces may propagate with (time-varying) curvature-dependent speeds. Level set methods provide the mathematical and numerical mechanisms for computing surface deformations as iso-values of ϕ by solving a partial differential equation on the 3-D grid (U). That is, the level set formulation provides a set of numerical methods that describes how to manipulate the gray-scale values in a volume, so that the iso-surfaces of ϕ move in a prescribed manner (see Fig. 3).

There are two different approaches to defining a deformable surface from a level set of a volumetric function, as described in Eq. (9). Either one can think of $\phi(\mathbf{s})$ as a static function and change the iso-value $k(t)$ or alternatively fix k and let the volumetric function dynamically change in time, i.e., $\phi(\mathbf{s}, t)$. Following the second approach, we can mathematically express the dynamic model as

$$\phi(\mathbf{s}, t) = k. \quad (10)$$

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

$$\frac{\partial \phi(\mathbf{s}, t)}{\partial t} + \nabla \phi(\mathbf{s}, t) \cdot \frac{d\mathbf{s}}{dt} = 0. \quad (11)$$

Equation (11) is sometimes referred to as a ‘‘Hamilton–Jacobi-type’’ equation and defines an initial value problem for the time-dependent ϕ . Let $d\mathbf{s}/dt$ be the movement of a point on a surface as it deforms, such that it can be expressed in terms of the position of $\mathbf{s} \in U$ and the geometry of the surface at that point, which is, in turn, a differential expression of the implicit function, ϕ . This gives a partial differential equation (PDE) on ϕ : $\mathbf{s} \equiv \mathbf{s}(t)$:

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{d\mathbf{s}}{dt} \equiv -\nabla \phi \cdot \mathbf{F}(\mathbf{s}, D\phi, D^2\phi, \dots), \quad (12)$$

where \mathbf{F} is a user-defined ‘‘speed’’ term which generally depends on a set of order- n derivatives of ϕ , $D^n \phi$, evaluated at \mathbf{s} , as well as other functions of \mathbf{s} . Typically $\mathbf{F}(\mathbf{x})$ combines a data term with a smoothing term, which prevents the solution from fitting too closely to noise-corrupted data. There are a variety of surface-motion terms that can be used in succession or simultaneously in a linear combination to form $\mathbf{F}(\mathbf{x})$. For the work presented in this paper, we combine a feature attraction term and a smoothing term weighted²⁸ by a factor β ,

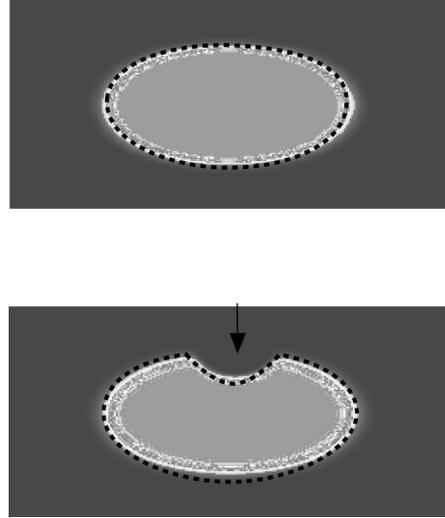


Fig. 3 Level set models represent curves and surfaces implicitly using gray-scale images. For example, an ellipse is represented as the level set of an image (top). To change the shape of the ellipse we modify the gray-scale values of the image by solving a PDE (bottom).

$$\mathbf{F} = \mathbf{F}_{\text{attr}} + \beta \mathbf{F}_{\text{curv}}. \quad (13)$$

The first term \mathbf{F}_{attr} is due to the attraction to the edges in the volume. It attracts the surface models to certain gray-scale features in the input data. For instance, the gradient magnitude indicates areas of high contrast in volumes. By following the gradient of such gray-scale features, surface models are drawn to minimum or maximum values of that feature. Typically, gray-scale features, such as the gradient magnitude are computed with a scale operator, e.g., a derivative-of-Gaussian kernel. If models are properly initialized, they can move according to the gradient of the gradient magnitude and settle onto the edges of an object at a resolution that is finer than the original volume. For this work we used the attraction force

$$\mathbf{F}_{\text{attr}} = \nabla |(\nabla(G * I(\mathbf{x})))|, \quad (14)$$

where the volume data $I(\mathbf{x})$ is convolved with a Gaussian kernel G with $\sigma \approx 0.5$, such that a positive sign moves surfaces toward maxima and the negative sign toward minima.

There are a variety of options for the curvature smoothing terms in Eq. (13), and the question of efficient, effective higher order smoothing terms is the subject of on-going research.³⁰ For the work presented in this paper the smoothing term uses the mean curvature \mathcal{K}_M of the level set \mathbf{S} to form a vector in the direction of the surface normal \mathbf{n} :

$$\mathbf{F}_{\text{curv}} = \mathcal{K}_M \mathbf{n} = (\nabla \cdot \mathbf{n}) \mathbf{n} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \frac{\nabla \phi}{|\nabla \phi|}. \quad (15)$$

It is weighted by a factor β , enabling the user to control the amount of smoothing, and is tuned for each dataset. The level set propagation stops when the \mathbf{F}_{attr} and $\beta \mathbf{F}_{\text{curv}}$ terms cancel each other, or when the number of computational iterations reaches a user-specified value.

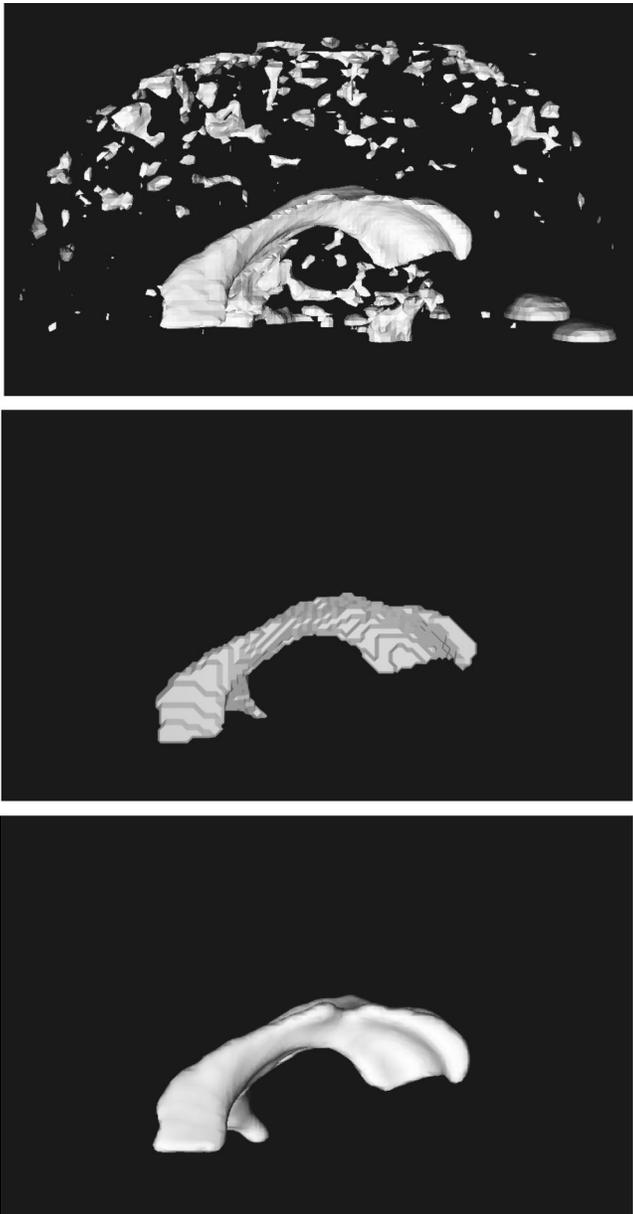


Fig. 4 Segmentation using isotropic measure \mathcal{V}_1 for the first DT-MRI dataset. The first row is the marching cubes iso-surface with 7.5-iso-value. The second row is the result of flood-fill algorithm applied to the same volume and used for level set initialization. The third row is the final level set model.

Level set models have a number of practical and theoretical advantages over conventional surface models, especially in the context of deformation and segmentation. Level set models are topologically flexible; they easily represent complicated surface shapes that can, form holes, split to form multiple objects, or merge with other objects to form a single structure. These models can incorporate many (millions) of degrees of freedom, and therefore they can accommodate complex shapes. Indeed, the shapes formed by the level sets of ϕ are restricted only by the resolution of the sampling. Thus, there is no need to reparameterize the model as it undergoes significant changes in shape.

The solutions to the partial differential equations described earlier are computed using finite differences on a

discrete grid. The use of a grid and discrete time steps raises a number of numerical and computational issues that are important to the implementation. However, it is outside of the scope of this paper to give a detailed mathematical description of such a numerical implementation. Rather we shall give a short outline here and refer to the actual source code, which is publicly available.*

Equations (12)–(15) can be solved using finite forward differences if one uses the up-wind scheme, proposed by Osher and Sethian,²⁹ to compute the spatial derivatives. This up-wind scheme produces the motion of level-set models over the entire range of the embedding, i.e., for all values of k in Eq. (10). However, this method requires updating every voxel in the volume for each iteration, which means that the computation time increases as a function of the volume, rather than the surface area, of the model. Because segmentation requires only a single model, the calculation of solutions over the entire range of iso-values is an unnecessary computational burden.

This problem can be avoided by the use of narrow-band methods, which compute solutions only in a narrow band of voxels that surround the level set of interest.²⁵ In previous work³¹ we described an alternative numerical algorithm, called the sparse-field method, that computes the geometry of only a small subset of points in the range and requires a fraction of the computation time required by previous algorithms. We have shown two advantages to this method. The first is a significant improvement in computation times. The second is increased accuracy when fitting models to forcing functions that are defined to subvoxel accuracy.

4 Segmentation

In this section, we demonstrate the application of our methods to the segmentation of DT-MRI data of the human head. We use a high-resolution data set from a normal volunteer, which contains 60 slices each of 128×128 pixels resolution. The raw data is sampled on a regular uniform grid.

We begin by generating two scalar volume datasets based on the invariants described in Sec. 2. The first scalar volume dataset (\mathcal{V}_1) is formed by calculating the trace (C_1) of the tensor matrix for each voxel of the diffusion tensor volume. It 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 Fig. 2 (left). The second scalar volume dataset (\mathcal{V}_2) is formed by calculating (C_1, C_2, C_3) invariants for each voxel and combining them into C_a . It 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 Fig. 2 (right). The measure C_a does not by definition distinguish between linear and planar anisotropy. This is sufficient for our current study since the brain does not contain

*The level-set software used to produce the morphing results in this paper is available for public use in the VISPACK libraries at <http://www.cs.utah.edu/~whitaker/vispack>.

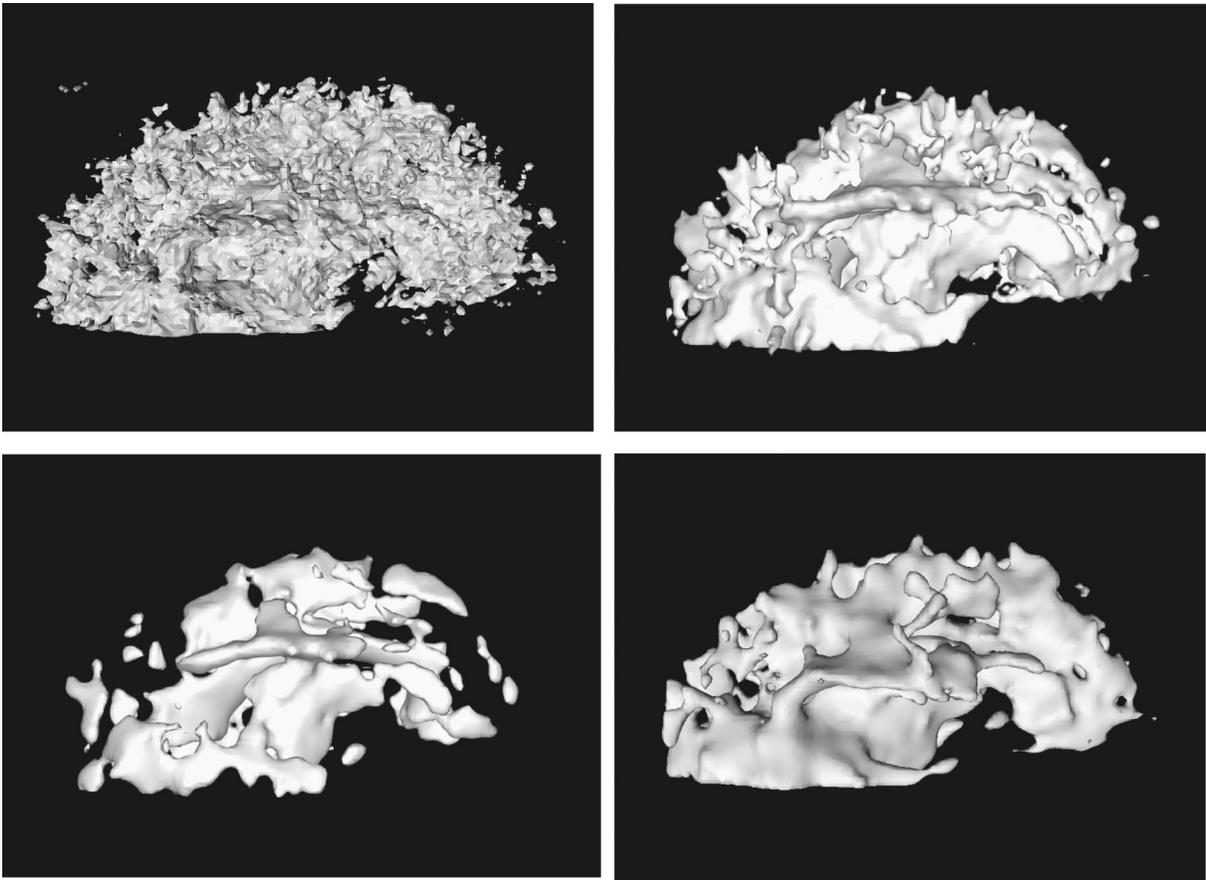


Fig. 5 Model segmentation from volume \mathcal{V}_2 . Top left image is an iso-surface of value 1.3, used for initialization of the level set. Clockwise, are the results of level set development with corresponding β values of 0.2, 0.4 and 0.5.

measurable regions with planar diffusion anisotropy. We therefore require only two scalar volumes to segment the DT dataset.

We then utilize level set methods to extract smoothed models from the two derived scalar volumes. Our level set segmentation approach consists of defining a set of suitable preprocessing techniques for initialization and selecting/tuning different feature-extracting terms in the level set equation to produce a surface deformation. Within our segmentation framework a variety of operations are available in each stage. A user must “mix-and-match” these operations to produce the desired result. We describe only those operations required to produce the models in this paper. A more detailed description of our segmentation methods may be found in Ref. 28.

Because level set models move using gradient descent, they seek local solutions, and therefore the results are strongly dependent on the initialization, i.e., the starting position of the surface. Thus, one controls the nature of the solution by specifying an initial model from which the surface deformation process proceeds. We are able to computationally construct reasonable initial estimates directly from the input data by combining a variety of techniques.

The first step involves filtering the input data with a low-pass Gaussian filter ($\sigma \approx 0.5$) to blur the data and thereby reduce noise. This tends to distort shapes, but the initialization can be only approximate. Next, the volume voxels are classified for inclusion/exclusion in the initial-

ization based on the filtered values of the input data ($k \approx 7.0$ for \mathcal{V}_1 and $k \approx 1.3$ for \mathcal{V}_2). For gray-scale images, such as those used in this paper, the classification is equivalent to high- and low-thresholding operations. These operations are usually accurate to only voxel resolution, but the deformation process will achieve subvoxel results. The final step before the actual level set deformation consists of performing a set of topological or logical operations on the voxels to “clean up” the initialization surface. This enables the removal of undesired internal and external structures, which is extremely helpful to obtain simple models. It includes unions or intersections of voxel sets to create the better initializations. Specifically, the topological operations consist of connected-component analyses (e.g., flood fill) to remove small pieces or holes from objects.

The initialization already described positions the model near the desired solution while retaining certain properties such as consistent geometry, connectivity, etc. Given this rough initial estimate, the level set surface deformation process, as described in Sec. 3.1, moves the surface model toward specific features in the data.

Figures 4 and 5 present two models that we extracted from DT-MRI volume datasets using our techniques. Figure 6 contains segmentations from volume \mathcal{V}_1 , the measure of total diffusivity. The image in the first row shows a marching cubes iso-surface using an iso-value of 7.5. In the bottom we have extracted just the ventricles from \mathcal{V}_1 . This is

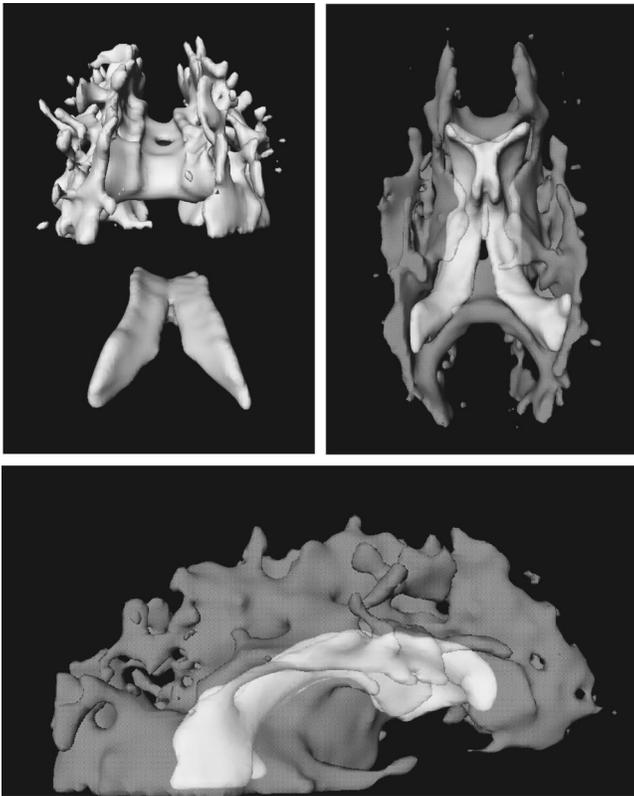


Fig. 6 Combined model of ventricles and (semitransparent) anisotropic regions: rear, exploded view (left), bottom view (right), side view (bottom). Note how model of ventricles extracted from isotropic measure dataset \mathcal{V}_1 fits into model extracted from anisotropic measure dataset \mathcal{V}_2 .

accomplished by creating an initial model with a flood-fill operation inside the ventricle structure shown in the middle image. This identified the connected voxels with value of 7.0 or greater. The initial model was then refined and smoothed with the level set method described in Section 3, using a β value of 0.2.

Figure 5 again provides the comparison between direct iso-surfacing and level set model, but on the volume \mathcal{V}_2 . The image in the top-left corner is a marching cubes iso-surface using an iso-value of 1.3. There is significant high-frequency noise and features in this dataset. The challenge here was to isolate coherent regions of high anisotropic diffusion. We applied our segmentation approach to the dataset and worked with neuroscientists from LA Childrens Hospital, City of Hope Hospital, and Caltech to identify meaningful anatomical structures. We applied our approach using a variety of parameter values, and presented our results to them, asking them to pick the model that they felt best represented the structures of the brain. Figure 5 contains three models extracted from \mathcal{V}_2 at different values of smoothing parameter β used during segmentation. Since we were not looking for a single connected structure in this volume, we did not use a seeded flood-fill for initialization. Instead we initialized the deformation process with an iso-surface of value 1.3. This was followed by a level set deformation using a β value of 0.2. The result of this segmentation is presented on the bottom-left side of Fig. 5. The top-right side of this figure presents a model extracted from

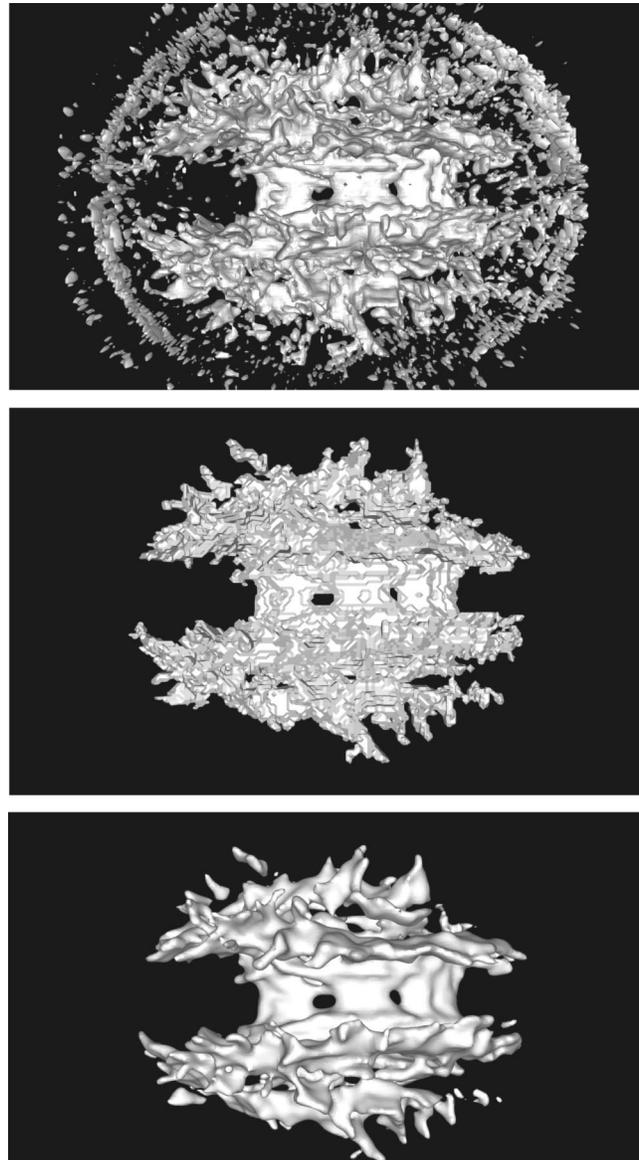


Fig. 7 Segmentation using anisotropic measure \mathcal{V}_2 from the second DT-MRI dataset. The first row is the marching cubes iso-surface with iso-value 1.3. The second row is the result of flood-fill algorithm applied to the volume and used for level set initialization. The third is the final level set model.

\mathcal{V}_2 using an initial iso-surface of value 1.4 and a β value of 0.5. The result chosen as the “best” by our scientific/medical collaborators is presented on the bottom-right side of Fig. 5. This model is produced with an initial iso-surface of 1.3 and a β value of 0.4. Our collaborators were able to identify structures of high diffusivity in this model, for example, the corpus callosum, the internal capsul, the optical nerve tracks, and other white matter regions.

We can also bring together the two models extracted from datasets \mathcal{V}_1 and \mathcal{V}_2 into a single image. Figure 6 demonstrates that we are able to isolate different structures in the brain and show their proper spatial interrelationship. 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.

Table 1 Total polygon count in the models N_{poly} , surface areas A and volumes V and before/after application of the level set smoothing to datasets \mathcal{V}_1 and \mathcal{V}_2 .

Data Set	N_{poly}	A (cm ²)	V (cm ³)
\mathcal{V}_1	36,620/15,096	188/85	26/22
\mathcal{V}_2	142,212/81,488	760/743	98/87

Finally, to verify the validity of our approach we applied it to the second data set of a different volunteer. This data set has 20 slices of the 256×256 resolution. We generated the anisotropy measure volume \mathcal{V}_2^2 and performed the level set model extraction using the same iso-values and smoothing parameters as for \mathcal{V}_2 . The results are shown in Fig. 7.

5 Model Properties

Once a user has produced a satisfactory model of the desired segmented structures, she or he can perform a number of quantitative geometric calculations on the resulting polygonal model, e.g., total area, volume, and average curvature. Though most of these measures are interesting from the modeling point of view, the volume of the ventricles, for example, can have clinical applications for disorder diagnosis and population comparison.

The models generated in the previous section are represented by triangle meshes consisting of vertices \mathbf{v}_i , connectivities, and associated normal vectors. The total surface area of the model can be easily computed by adding the areas A_i of each triangle

$$A = \sum_{i=1}^{N_{\text{poly}}} A_i = \sum_{i=1}^{N_{\text{poly}}} \frac{1}{2} |(\mathbf{v}_i^1 - \mathbf{v}_i^2) \times (\mathbf{v}_i^1 - \mathbf{v}_i^3)|, \quad (16)$$

where \mathbf{v}_i^k is the k 'th vertex of triangle i . Assuming that all of the extracted models are composed of closed polygonal surfaces, we can compute enclosed volume as a signed sum of the pyramids with a base composed of the i 'th triangle and a top vertex places at the origin of the dataset.³² Then

$$V \approx \frac{1}{6} \sum_{i=1}^{N_{\text{poly}}} A_i \cdot \frac{1}{3} (\mathbf{v}_i^1 + \mathbf{v}_i^2 + \mathbf{v}_i^3) \cdot \mathbf{N}_i. \quad (17)$$

Table 1 lists values of polygon count, surface area and total volume, for the models extracted from scalar volume datasets (\mathcal{V}_1 and \mathcal{V}_2), before and after the level set algorithm is applied to the volumes. We note that the polygon count drops, because of the simplified form of the final extracted triangular mesh. The total surface area decrease is also due to smoothing imposed by the level set model. Volume decrease is partially caused by the removal (i.e., collapse) of small high-frequency fragments cluttering the model and partially due to deformations of the model.

6 Conclusions

We developed a computational pipeline for DT-MRI level set modeling and segmentation. We proposed a new rota-

tionally invariant anisotropy measure, which does not require eigenvalue computations. We used the invariants to generate scalar volumes that characterize the total diffusivity and diffusion anisotropy of a DT-MRI scan of a human brain. Applying level set modeling and segmentation techniques to the derived scalar volumes we created geometric models of specific brain structures, e.g., the ventricles, corpus callosum, and the internal capsul. The geometric models were then used for quantitative analysis, including volume and surface area calculations.

Acknowledgments

We would like to thank Dr. J. Michael Tyszka, Dr. Miriam Scadeng, and Dr. David Dubowitz for helping us to identify the 3-D structures extracted from the DT dataset. Dr. Jason Wood developed the Iris Explorer modules used to produce part of the results in the paper. This work was supported by National Science Foundation (NSF) Grants No. ACI-9982273 and No. 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 No. N000140110033, and the National Library of Medicine "Insight" Project No. N01-LM-0-3503. The first DT-MRI dataset is courtesy of the University of Utah SCI Institute, the second dataset is courtesy of Dr. Mark Bastin, University of Edinburgh, United Kingdom. Finally, we would like to thank our reviewers for a very detailed review and multiple valuable suggestions.

References

1. P. J. Basser, J. Mattiello, and D. Le Bihan, "Estimation of the effective self-diffusion tensor from the NMR spin echo," *J. Magn. Reson., Ser. B* **103**(3), 247–254 (1994).
2. P. J. Basser, J. Mattiello, and D. Le Bihan, "MR diffusion tensor spectroscopy and imaging," *Biophys. J.* **66**(1), 259–267 (1994).
3. C. Pierpaoli and P. J. Basser, "Toward a quantitative assessment of diffusion anisotropy," *Magn. Reson. Med.* **36**(6), 893–906 (1996).
4. P. J. Basser, "Inferring microstructural features and the physiological state of tissues from diffusion weighted images," *NMR Biomed.* **8**(7–8), 333–344 (1995).
5. C. Pierpaoli, P. Jezzard, P. J. Basser, A. Barnett, and G. Di Chiro, "Diffusion tensor MR imaging of the human brain," *Radiology* **201**(3), 637–648 (1996).
6. P. J. Basser and C. Pierpaoli, "Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI," *J. Magn. Reson., Ser. B* **111**, 209–219 (1996).
7. 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*, p. 1742, Vancouver, Canada (1997).
8. S. Peled, H. Gudbjartsson, C. F. Westin, R. Kikinis, and F. A. Jolesz, "Magnetic resonance imaging shows orientation and asymmetry in white matter fiber tracts," *Brain Res.* **780**, 27–33 (1998).
9. P. J. Basser and S. Pajevic, "Statistical artifacts in diffusion tensor MRI caused by background noise," *Magn. Reson. Med.* **44**, 41–50 (2000).
10. A. M. Ulug and P. C. M. van Zijl, "Orientation-independent diffusion imaging without tensor diagonalization: anisotropy definitions based on physical attributes of the diffusion ellipsoid," *J. Magn. Reson. Imaging* **9**, 804–813 (1999).
11. 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).
12. 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).
13. P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi, "In vivo fiber tractography using DT-MRI data," *Magn. Reson. Med.* **44**, 625–632 (2000).
14. 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**, 184–195 (2000).
15. R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *Comput. Graph.* **22**(4), 65–74 (1988).
 16. S. Parker, M. Parker, Y. Livnat, P. Sloan, C. Hansen, and P. Shirley, "Interactive ray tracing for volume visualization," *IEEE Trans. Vis. Comput. Graph.* **5**(3), 238–250 (1999).
 17. A. Singh, D. Goldgof, and D. Terzopoulos, Eds., *Deformable Models in Medical Image Analysis*, IEEE Computer Society Press, Los Alamitos, CA (1998).
 18. 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).
 19. L. E. Zhukov, D. M. Weinstein, and C. R. Johnson, "Independent component analysis for EEG source localization in realistic head model," *IEEE Eng. Med. Biol. Mag.* **19**, 87–96 (2000).
 20. S. Gibson *et al.*, "Volumetric object modeling for surgical simulation," *Med. Image Anal.* **2**(2), 121–132 (1998).
 21. M. Bailey, "Manufacturing isovolumes," in *Volume Graphics*, M. Chen, A. E. Kaufman, and R. Yagel, Eds., pp. 79–93, Springer, London (2000).
 22. W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," *Comput. Graph.* **21**(3), 163–169 (1987).
 23. 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," *Comput. Graph.* **25**(4), 217–226 (1991).
 24. R. T. Whitaker and D. T. Chen, "Embedded active surfaces for volume visualization," in *Medical Imaging, Proc. SPIE*, 340–352 (1994).
 25. R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(2), 158–175 (1995).
 26. M. Leventon, O. Faugeras, 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).
 27. L. Staib, X. Zeng, R. Schultz, and J. Duncan, "Shape constraints in deformable models," in *Handbook of Medical Imaging*, I. Bankman, Ed., pp. 147–157, Academic Press, (2000).
 28. R. T. Whitaker, D. E. Breen, K. Museth, and N. Soni, "A framework for level set segmentation of volume datasets," in *Proc. Volume Graphics 2001*, pp. 159–168 (2001).
 29. S. Osher and J. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.* **79**, 12–49 (1988).
 30. J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge (1999).
 31. R. T. Whitaker, "A level-set approach to 3D reconstruction from range data," *Int. J. Comput. Vis.* **29**(3), 203–231 (1998).
 32. D. Eberly, J. Lancaster, and A. Alyassin, "On gray scale image measurements," *CVGIP: Graph. Models Image Process.* **56**(6), 550–562 (1991).



Leonid Zhukov is a senior research scientist with the California Institute of Technology Computer Graphics Laboratory. He received his BS and MS degrees in physics from Moscow Engineering and Physics Institute in 1991 and 1993 and his PhD degree from the University of Utah in 1998. His research interests are scientific computing, visualization, and computer graphics. His specific interests include development of computational and 3-D visualization methods for biomedical problems.



Ken Museth is a research scientist in computer graphics with the Computer Science Department, the California Institute of Technology (Caltech). His main research interests are implicit modeling, volume graphics, level set methods, and general 3-D graphics by means of solving partial differential equations. He came to Caltech in 1998 as a visiting faculty member in computational quantum chemistry, but quickly shifted field to computer science and joined the Computer Graphics Laboratory as a research staff member in 2000. He received his PhD degree from the University of Copenhagen in 1997.



David Breen is currently a senior research scientist with the Center for Advanced Computing Research at the California Institute of Technology, where he was previously the assistant director of the Computer Graphics Laboratory. He has held research positions at the European Computer Industry Research Centre, the Fraunhofer Institute for Computer Graphics, and the Rensselaer Design Research Center (formerly the RPI Center for Interactive Computer Graphics). His research interests include level-set models for graphics, volume graphics and segmentation, large data visualization, and geometric modeling. He is a member of ACM SIGGRAPH, the IEEE Computer Society and Eurographics. He is the coeditor of the recently published book *Cloth Modeling and Animation* (AK Peters). Breen received his BA degree in physics from Colgate University in 1982 and his MS and PhD degrees in computer and systems engineering from Rensselaer Polytechnic Institute in 1985 and 1993.



Ross Whitaker received his BS degree in electrical engineering and computer science from Princeton University in 1986, earning Summa Cum Laude. From 1986 to 1988 he was with the Boston Consulting Group, and entered the University of North Carolina (UNC) at Chapel Hill in 1989. At UNC he received an Alumni Scholarship Award, and completed his PhD in computer science in 1994. From 1994 to 1996 he was a research scientist with the European Computer-Industry Research Centre, Munich, Germany, in the User Interaction and Visualization Group. From 1996 to 2000 he was an assistant professor with the Department of Electrical Engineering at the University of Tennessee. Since 2000 he has been an assistant professor with the University of Utah in the College of Computing and a faculty member of the Scientific Computing and Imaging Institute.



Alan H. Barr is a professor of computer science with the California Institute of Technology. He received his PhD in mathematics from Rensselaer Polytechnic Institute. His research interests are generic methods for mathematical modeling, rendering, behavioral simulation, and scientific visualization so that reconfigurable models can have great predictive power in the physical sciences. In 1988 he received that ACM SIGGRAPH Computer Graphics Achievement Award, for research contribution to the field of computer graphics. In 1995 he became a fellow of the ACM.