

Level Set Segmentation of Biological Volume Datasets

*David Breen*¹ *Ross Whitaker*²
*Ken Museth*³ *Leonid Zhukov*⁴

¹Department of Computer Science, Drexel University, Philadelphia, PA 19104

²School of Computing, University of Utah, Salt Lake City, UT 84112

³Department of Science and Technology, Linköping University, 601 74 Norrköping,
Sweden

⁴Department of Computer Science, California Institute of Technology, Pasadena, CA
91125

Abstract

This chapter describes level set techniques for extracting surface models from a broad variety of biological volume datasets. These techniques have been incorporated into a more general framework that includes other volume processing algorithms. The volume datasets are produced from standard 3D imaging devices, and are all noisy samplings of complex biological structures with boundaries that have low and often varying contrasts. The level set segmentation method, which is well documented in the literature, creates a new volume from the input data by solving an initial value partial differential equation (PDE) with user-defined feature-extracting terms. Given the local/global nature of these terms, proper initialization of the level set algorithm is extremely important. Thus, level set deformations alone are not sufficient, they must be combined with powerful pre-processing and data analysis techniques in order to produce successful segmentations. This chapter describes the pre-processing and data analysis techniques that have been developed for a number of segmentation applications, as well as the general structure of our framework. Several standard volume processing algorithms have been incorporated into the framework in order to segment datasets generated from MRI, CT and TEM scans. A technique based on moving least-squares has been developed for segmenting multiple non-uniform scans of a single object. New scalar measures have been defined for extracting structures from diffusion tensor MRI scans. Finally, a direct approach to the segmentation of incomplete tomographic data using density parameter estimation is described. These techniques, combined with level set surface deformations, allow us to segment many different types of biological volume datasets.

0.1 Introduction

This chapter addresses the common problem of building meaningful 3D models of complex structures from noisy datasets generated from 3D imaging devices. In certain circumstances such data can be visualized *directly* [1, 2, 3, 4]. While direct techniques can provide useful insights into volume data, they are insufficient for many problems. For instance, direct volume rendering techniques typically do not remove occluding structures, i.e., they do not allow one to “peel back” the various layers of the data to expose the inner structures that might

be of interest. They also do not generate the models needed for quantitative study/analysis of the visualized structures. Furthermore, direct visualization techniques typically do not perform well when applied directly to noisy data, unless one filters the data first. Techniques for filtering noisy data are abundant in the literature, but there is a fundamental limitation—filtering that reduces noise tends to distort the shapes of the objects in the data. The challenge is to find methods which present the best tradeoff between fidelity and noise.

Level set segmentation relies on a surface-fitting strategy, which is effective for dealing with both small-scale noise and smoother intensity fluctuations in volume data. The level set segmentation method, which is well documented in the literature [5, 6, 7, 8], creates a new volume from the input data by solving an initial value partial differential equation (PDE) with user-defined feature-extracting terms. Given the local/global nature of these terms, proper initialization of the level set algorithm is extremely important. Thus, level set deformations alone are not sufficient, they must be combined with powerful initialization techniques in order to produce successful segmentations. Our level set segmentation approach consists of defining a set of suitable pre-processing techniques for initialization and selecting/tuning different feature-extracting terms in the level set algorithm. We demonstrate that combining several pre-processing steps, data analysis and level set deformations produces a powerful toolkit that can be applied, under the guidance of a user, to segment a wide variety of volumetric data.

There are more sophisticated strategies for isolating meaningful 3D structures in volume data. Indeed, the so called *segmentation problem* constitutes a significant fraction of the literature in image processing, computer vision, and medical image analysis. For instance, statistical approaches [9, 10, 11, 12] typically attempt to identify tissue types, voxel by voxel, using a collection of measurements at each voxel. Such strategies are best suited to problems where the data is inherently multi-valued or where there is sufficient prior knowledge [13] about the shape or intensity characteristics of the relevant anatomy. Alternatively, anatomical structures can be isolated by grouping voxels based on local image properties. Traditionally, image processing has relied on collections of edges, i.e. high-contrast boundaries, to distinguish regions of different types [14, 15, 16]. Furthermore deformable models, incorporating different degrees of domain-specific knowledge, can be *fitted* to the 3D input data [17, 18].

This chapter describes a level set segmentation framework, as well as the pre-processing and data analysis techniques needed to segment a diverse set of biological volume datasets. Several standard volume processing algorithms have been incorporated into the framework for segmenting conventional datasets generated from MRI, CT and TEM scans. A technique based on moving least-squares has been developed for segmenting multiple non-uniform scans of a single object. New scalar measures have been defined for extracting structures from diffusion tensor MRI scans. Finally, a direct approach to the segmentation of incomplete tomographic data using density parameter estimation is described. These techniques, combined with level set surface deformations, allow us to segment many different types of biological volume datasets.

0.2 Level Set Surface Models

When considering deformable models for segmenting 3D volume data, one is faced with a choice from a variety of surface representations, including triangle meshes [19, 20], superquadrics [21, 22, 23], and many others [18, 24, 25, 26, 27, 28, 29]. Another option is an implicit level set model, which specifies the surface as a *level set* of a scalar volumetric function, $\phi : U \mapsto \mathbb{R}$, where $U \subset \mathbb{R}^3$ is the range of the surface model. Thus, a surface S is

$$S = \{\mathbf{s} | \phi(\mathbf{s}) = k\}, \quad (1)$$

with an isovalue k . In other words, S is the set of points \mathbf{s} in \mathbb{R}^3 that composes the k th isosurface of ϕ . The embedding ϕ can be specified as a regular sampling on a rectilinear grid.

Our overall scheme for segmentation is largely based on the ideas of Osher and Sethian [30] that model propagating surfaces with (time-varying) curvature-dependent speeds. The surfaces are viewed as a specific level set of a higher-dimensional function ϕ – hence the name level set methods. These methods provide the mathematical and numerical mechanisms for computing surface deformations as isovalues of ϕ by solving a partial differential equation on the 3D grid. That is, the level set formulation provides a set of numerical methods that describes how to manipulate the grayscale values in a volume, so that the isosurfaces of ϕ move in a prescribed manner (shown in Figure 1). This paper does not present a comprehensive review of level set methods, but merely

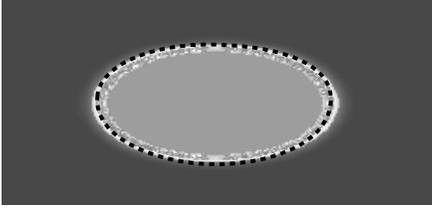


Figure 1a: Level set models represent curves and surfaces implicitly using grayscale images. For example an ellipse is represented as the level set of an image shown here.

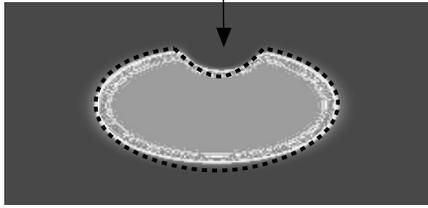


Figure 1b: To change the shape of the ellipse we modify the grayscale values of the image by solving a PDE.

introduces the basic concepts and demonstrates how they may be applied to the problem of volume segmentation. For more details on level set methods see [7, 31].

There are two different approaches to defining a deformable surface from a level set of a volumetric function as described in Equation 1. Either one can think of $\phi(\mathbf{s})$ as a *static* function and change the isovalue $k(t)$ or alternatively fix k and let the volumetric function *dynamically* change in time, i.e. $\phi(\mathbf{s}, t)$. Thus, we can mathematically express the static and dynamic model respectively as

$$\phi(\mathbf{s}) = k(t) \quad (2a)$$

$$\phi(\mathbf{s}, t) = k. \quad (2b)$$

To transform these definitions into partial differential equations which can be solved by standard numerical techniques, we differentiate both sides of Equation 2 with respect to time t , and apply the chain rule:

$$\nabla\phi(\mathbf{s}) \frac{d\mathbf{s}}{dt} = \frac{dk(t)}{dt} \quad (3a)$$

$$\frac{\partial\phi(\mathbf{s}, t)}{\partial t} + \nabla\phi(\mathbf{s}, t) \cdot \frac{d\mathbf{s}}{dt} = 0. \quad (3b)$$

The static Equation 3a defines a boundary value problem for the time-independent volumetric function ϕ . This static level set approach has been solved [32, 33] using “Fast Marching Methods”. However it inherently has some serious limitations following the simple definition in Equation 2a. Since ϕ is a function (i.e. single-valued), isosurfaces cannot self intersect over time, i.e. shapes defined in the static model are strictly expanding or contracting over time. However,

the dynamic level set approach of Equation 3b is much more flexible and shall serve as the basis of the segmentation scheme in this paper. Equation 3b is sometimes referred to as a “Hamilton-Jacobi-type” equation and defines an initial value problem for the time-dependent ϕ . Throughout the remainder of this paper we shall for simplicity refer to this dynamical approach as the level set method – and not consider the static alternative.

Thus, to summarize the essence of the (dynamic) level set approach; 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 on ϕ : $\mathbf{s} \equiv \mathbf{s}(t)$

$$\frac{\partial\phi}{\partial t} = -\nabla\phi \cdot \frac{d\mathbf{s}}{dt} = \|\nabla\phi\| \mathcal{F}(\mathbf{s}, \mathbf{n}, \phi, D\phi, D^2\phi, \dots) \quad (4a)$$

$$\mathcal{F}() \equiv \mathbf{n} \cdot \frac{d\mathbf{s}}{dt}, \quad (4b)$$

where $\mathcal{F}()$ is a user-created “speed” term that defines the speed of the level set at point \mathbf{s} in the direction of the local surface normal \mathbf{n} at \mathbf{s} . $\mathcal{F}()$ may depend on a variety of local and global measures including the order- n derivatives of ϕ , $D^n\phi$, evaluated at \mathbf{s} , as well as other functions of \mathbf{s} , \mathbf{n} , ϕ and external data. Because this relationship applies to every level set of ϕ , i.e. all values of k , this equation can be applied to all of U , and therefore the movements of *all* the level set surfaces embedded in ϕ can be calculated from Equation 4.

The level set representation has a number of practical and theoretical advantages over conventional surface models, especially in the context of deformation and segmentation. First, 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 such as the dendrite in Figure 6. 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 above 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 provide a summary in a later section and refer to the actual source code which is publicly available¹.

Equation 4 can be solved using finite forward differences if one uses the up-wind scheme, proposed by Osher and Sethian [30], 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 Equation 2. 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 [34, 35]. In previous work [36] 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 sub-voxel accuracy.

0.3 Segmentation Framework

The level set segmentation process has two major stages, initialization and level set surface deformation, as seen in Figure 2. Each stage is equally important for generating a correct segmentation. Within our framework a variety of core operations are available in each stage. A user must “mix-and-match” these operations in order to produce the desired result [37]. Later sections describe specialized operations for solving specific segmentation problems that build upon and extend the framework.

¹The 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>.

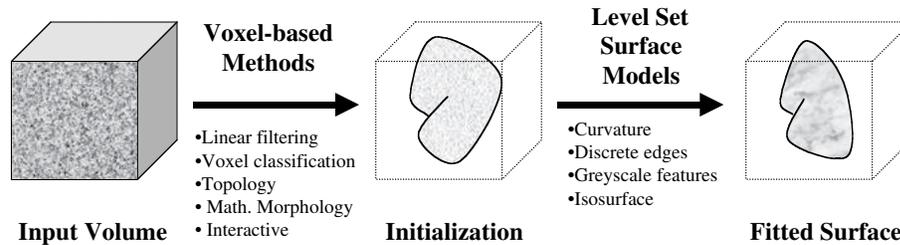


Figure 2: Level set segmentation stages – initialization and surface deformation.

0.3.1 Initialization

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 have implemented both computational (i.e. “semi-automated”) and manual/interactive initialization schemes that that may be combined to produce reasonable initial estimates directly from the input data.

Linear filtering: We can filter the input data with a low-pass filter (e.g. Gaussian kernel) to blur the data and thereby reduce noise. This tends to distort shapes, but the initialization need only be approximate.

Voxel classification: We can classify pixels based on the filtered values of the input data. For greyscale 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 (see [12] for alternatives), but the deformation process will achieve sub-voxel results.

Topological/logical operations: This is the set of basic voxel operations that takes into account position and connectivity. It includes unions or intersections of voxel sets to create better initializations. These logical operations can also incorporate user-defined primitives. Topological operations consist of connected-component analyses (e.g. flood fill) to remove small pieces or holes from objects.

Morphological filtering: This includes binary and greyscale morphological operators on the initial voxel set. For the results in the paper we

implement openings and closings using *morphological propagators* [38, 39] implemented with level set surface models. This involves defining offset surfaces of ϕ by expanding/contracting a surface according to the following PDE,

$$\frac{\partial\phi}{\partial t} = \pm|\nabla\phi|, \quad (5)$$

up to a certain time t . The value of t controls the offset distance from the original surface of $\phi(t = 0)$. A dilation of size α , D_α , corresponds to the solution of Equation 5 at $t = \alpha$ using the positive sign, and likewise erosion, E_α , uses the negative sign. One can now define a morphological opening operator O_α by first applying an erosion followed by a dilation of ϕ , i.e. $O_\alpha\phi = D_\alpha \circ E_\alpha\phi$, which removes small pieces or thin appendages. A closing is defined as $C_\alpha\phi = E_\alpha \circ D_\alpha\phi$, and closes small gaps or holes within objects. Both operations have the qualitative effect of low-pass filtering the isosurfaces in ϕ —an opening by removing material and a closing by adding material. Both operations tend to distort the shapes of the surfaces on which they operate, which is acceptable for the initialization because it will be followed by a surface deformation.

User-specified: For some applications it is desirable and easier for the user to interactively specify the initial model. Here, the user creates a Constructive Solid Geometry (CSG) model which defines the shape of the initial surface. In Figure 3a the CSG model in blue is interactively positioned relative to a Marching Cubes mesh extracted from the original dataset. The CSG model is scan-converted into a binary volume, with voxels simply marked as inside (1) or outside (0), using standard CSG evaluation techniques [40]. An isosurface of the initialization volume dataset generated from the torus and sphere is presented in Figure 3b. This volume dataset is then deformed to produce the final result seen in Figure 3c.

0.3.2 Level Set Surface Deformation

The initialization should position the model near the desired solution while retaining certain properties such as smoothness, connectivity, etc. Given a

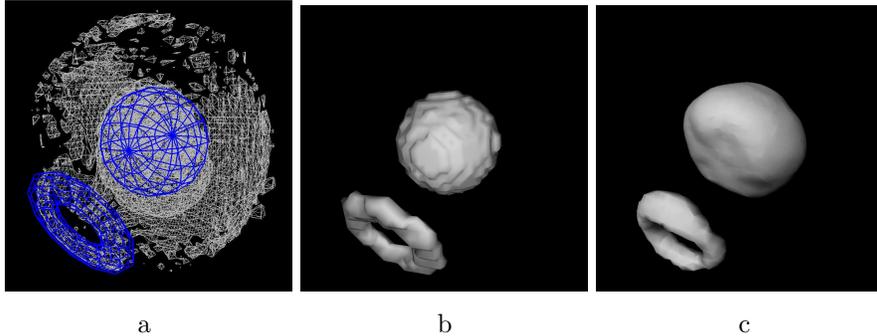


Figure 3: (a) Interactively positioning a CSG model relative to a Marching Cubes mesh. (b) Isosurface of a binary scan conversion of the initialization CSG model. (c) Final internal embryo structures.

rough initial estimate, the surface deformation process moves the surface model toward specific features in the data. One must choose those properties of the input data to which the model will be attracted and what role the shape of the model will have in the deformation process. Typically, the deformation process 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 $\mathcal{F}(\mathbf{x})$ in Equation 4.

Curvature: This is the smoothing term. For the work presented here we use the mean curvature of the isosurface H to produce

$$\mathcal{F}_{curv}(\mathbf{x}) = H = \left(\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right). \quad (6)$$

The mean curvature is also the normal variation of the surface area (i.e., minimal surface area). There are a variety of options for second-order smoothing terms [41], and the question of efficient, effective higher-order smoothing terms is the subject of on-going research [7, 42, 31]. For the work in this paper, we combine mean curvature with one of the following three terms, weighting it by a factor β , which is tuned to each specific application.

Edges: Conventional edge detectors from the image processing literature produce sets of “edge” voxels that are associated with areas of high con-

trast. For this work we use a gradient magnitude threshold combined with non-maximal suppression, which is a 3D generalization of the method of Canny [16]. The edge operator typically requires a scale parameter and a gradient threshold. For the scale, we use small, Gaussian kernels with standard deviation $\sigma = [0.5, 1.0]$ voxel units. The threshold depends on the contrast of the volume. The distance transform on this edge map produces a volume that has minima at those edges. The gradient of this volume produces a field that attracts the model to these edges. The edges are limited to voxel resolution because of the mechanism by which they are detected. Although this fitting is not sub-voxel accurate, it has the advantage that it can pull models toward edges from significant distances, and thus inaccurate initial estimates can be brought into close alignment with high-contrast regions, i.e. edges, in the input data. If \mathcal{E} is the set of edges, and $D_{\mathcal{E}}(\mathbf{x})$ is the distance transform to those edges, then the movement of the surface model is given by

$$\mathcal{F}_{edge}(\mathbf{x}) = \mathbf{n} \cdot \nabla D_{\mathcal{E}}(\mathbf{x}). \quad (7)$$

Greyscale features—gradient magnitude: Surface models can also be attracted to certain greyscale features in the input data. For instance, the gradient magnitude indicates areas of high contrast in volumes. By following the gradient of such greyscale features, surface models are drawn to minimum or maximum values of that feature. Typically greyscale 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.

If $G(\mathbf{x})$ is some greyscale feature, for instance $G(\mathbf{x}) = |\nabla I(\mathbf{x})|$, where $I(\mathbf{x})$ is the input data (appropriately filtered—we use Gaussian kernels with $\sigma \approx 0.5$), then

$$\mathcal{F}_{grad}(\mathbf{x}) = \mathbf{n} \cdot (\pm \nabla G(\mathbf{x})), \quad (8)$$

where a positive sign moves surfaces towards maxima and the negative sign towards minima.

Isosurface: Surface models can also expand or contract to conform to isosurfaces in the input data. To a first order approximation, the distance from a point $\mathbf{x} \in U$ to the k -level surface of I is given by $(I(\mathbf{x}) - k) / |\nabla I|$. If we let $g(\alpha)$ be a fuzzy threshold, e.g., $g(\alpha) = \alpha / \sqrt{1 + \alpha^2}$, then

$$\mathcal{F}_{iso}(\mathbf{x}) = g\left(\frac{I(\mathbf{x}) - k}{|\nabla I|}\right) \quad (9)$$

causes the surfaces of ϕ to expand or contract to match the k isosurface of I . This term combined with curvature or one of the other fitting terms can create “quasi-isosurfaces” that also include other considerations, such as smoothness or edge strength.

0.3.3 Framework Results

Figure 4 presents one slice from an MRI scan of a mouse embryo, and an iso-surface model of its liver extracted from the unprocessed dataset. Figure 5 presents 3D renderings of the sequence of steps performed on the mouse MRI data to segment the liver. The first step is the initialization, which includes smoothing the input data, thresholding followed by a flood fill to remove isolated holes, and finally applying morphological operators to remove small gaps and protrusions on the surface. The second (surface deformation) step first involves fitting to discrete edges and then to the gradient magnitude. This produces a significant improvement over the result in Figure 4. Figure 8a presents several other structures that were segmented from the mouse embryo dataset. The skin (grey) and the liver (blue) were isolated using computational initialization. The brain ventricles (red) and the eyes (green) were segmented with interactive initialization.

The same set of initialization and surface deformation steps may be combined to extract a model of a spiny dendrite from the transmission electron microscopy (TEM) scan presented in Figure 6a. An iso-surface extracted from the scan is presented in Figure 6b. Figure 7 shows the results of the proposed method compared to the results of a manual segmentation, which took approximately 10 hours of slice-by-slice hand contouring. The manual method suffers from slice-wise artifacts, and, because of the size and complexity of the dataset, the manual segmentation is unable to capture the level of detail that we obtain with the surface-fitting results. Manual segmentation can, however, form connections that are not well supported by the data in order to complete

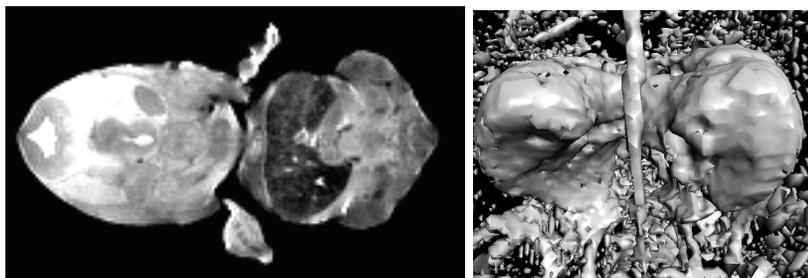


Figure 4: (left) One slice of a $256 \times 128 \times 128$ MR scan of a mouse embryo. The central dark structure is its liver. (right) A dual-threshold surface rendering highlights the segmentation problem.

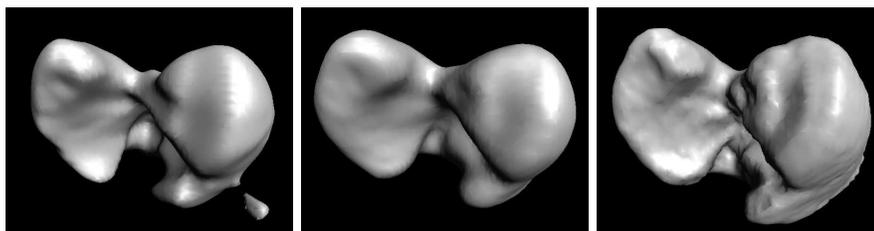


Figure 5: (left) The initialization of a mouse liver dataset using morphology to remove small pieces and holes. (center) Surface fitting to discrete edges. (right) The final fit to maxima of gradient magnitude.

the “spines” that cover this dendrite. These types of “judgments” that humans make when they perform such tasks by hand are a mixed blessing. Humans can use high-level knowledge about the problem to fill in where the data is weak, but the expectations of a trained operator can interfere with seeing unexpected or unusual features in the data.

Figure 8 presents models from four samples of an MR series of a developing frog embryo. The top left image (Hour 9) shows the first evident structure, the blastocoel, in blue, surrounded by the outside casing of the embryo in grey. The top right image (Hour 16) demonstrates the expansion of the blastocoel and the development of the blastoporal lip in red. In the bottom left image (Hour 20) the blastoporal lip has collapsed, the blastocoel has contracted, and the archenteron in green has developed. In the bottom right image (Hour 30) the blastocoel has collapsed and only the archenteron is present. For this

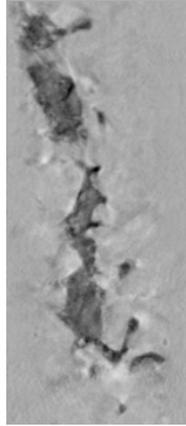


Figure 6a: One slice of a $154 \times 586 \times 270$ TEM scan of a spiny dendrite shows low contrast and high noise content in a relatively complex dataset.



Figure 6b: An isosurface rendering, with prefiltering, shows how noise and inhomogeneities in density interfere with visualizing the 3D structure of the dendrite.

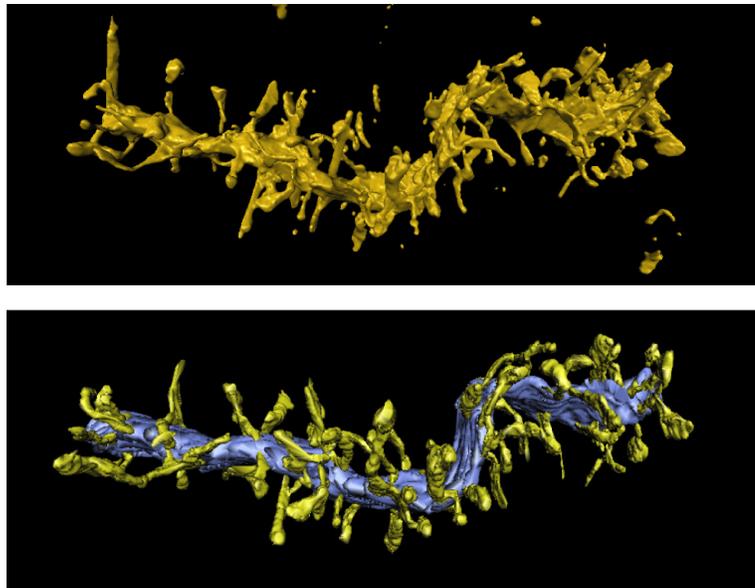


Figure 7: (top) Rendering of a dendrite segmented using our the proposed method. (bottom) Rendering of a manual segmentation of the same dendrite.

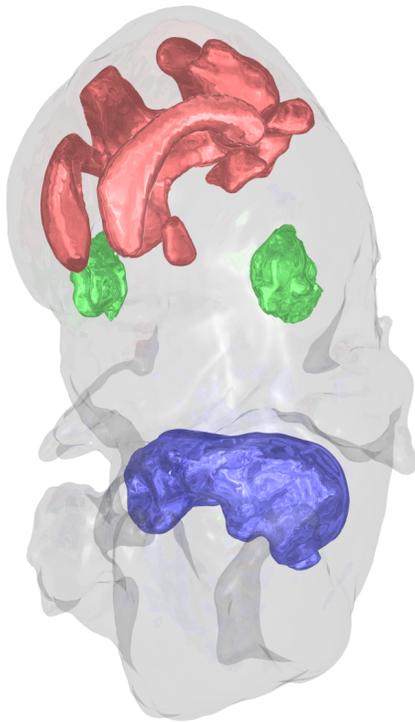


Figure 8a: Final mouse embryo model with skin (grey), liver (blue), brain ventricles (red), and eyes (green).

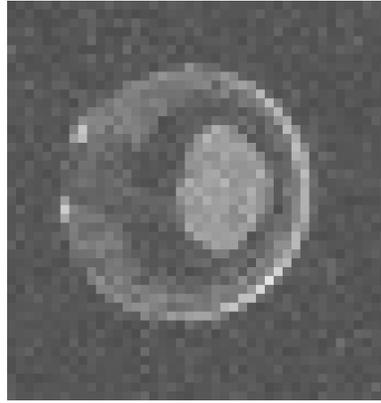


Figure 8b: Hour 16 dataset.

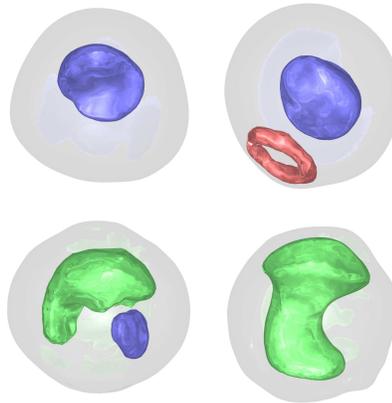


Figure 8c: Geometric structures extracted from MRI scans of a developing frog embryo, with blastocoel (blue), blastoporal lip (red), and archenteron (green). Hour 9 (top left). Hour 16 (top right). Hour 20 (bottom left). Hour 30 (bottom right).

dataset it was difficult to isolate structures only based on their voxel values. We therefore used our interactive techniques to isolate (during initialization) most of the structures in the frog embryo samples.

Table 1 describes for each dataset the specific techniques and parameters we used for the results in this section. These parameters were obtained by first making a sensible guess based on the contrasts and sizes of features in the data and then using trial and error to obtain acceptable results. Each dataset was processed between 4 and 8 times to achieve these results. More tuning could improve things further, and once these parameters are set, they work moderately well for similar modalities with similar subjects. The method is iterative, but the update times are proportional to the surface area. On an SGI 180MHz MIPS 10000 machine, the smaller mouse MR dataset required approximately 10 minutes of CPU time, and the dendrite dataset ran for approximately 45 minutes. Most of this time was spent in the initialization (which requires several complete passes through the data) and in the edge detection. The frog embryo datasets needed only a few minutes of processing time, because they did not require computational initialization and are significantly smaller than the other example datasets.

0.4 Segmentation From Multiple Non-Uniform Volume Datasets

Many of today’s volumetric datasets are generated by medical MR, CT and other scanners. A typical 3-D scan has a relatively high resolution in the scanning $X - Y$ plane, but much lower resolution in the axial Z direction. The difference in resolution between the in-plane and out-of-plane samplings can easily range between a factor of 5 to 10, see Figure 9. This occurs both because of physical constraints on the thickness of the tissue to be excited during scanning (MR), total tissue irradiation (CT), and scanning time restrictions. Even when time is not an issue, most scanners are by design incapable of sampling with high resolution in the out-of-plane direction, producing anisotropic “brick-like” voxels.

The non-uniform sampling of an object or a patient can create certain problems. The inadequate resolution in the Z direction implies that small or thin structures will not be properly sampled, making it difficult to capture them

Dataset	Initialization	Surface Fitting
Dendrite	<ol style="list-style-type: none"> 1. Gaussian blur $\sigma = 0.5$ 2. Threshold: $I < 127$ 3. Fill isolated holes 4. Morphology: $O_{0.5} \circ C_{1.5}$ 	<ol style="list-style-type: none"> 1. Edge fitting: $\sigma = 0.75$, threshold = 6, $\beta = 0.1$ 2. Gradient magnitude fitting: $\sigma = 0.5$, $\beta = 1.0$
Mouse	<ol style="list-style-type: none"> 1. Gaussian blur $\sigma = 0.5$ 2. Threshold: $I > 3, I < 60$ 3. Fill isolated holes 4. Morphology: $O_{2.0} \circ C_{3.0}$ 	<ol style="list-style-type: none"> 1. Edge fitting: $\sigma = 0.75$, threshold = 20, $\beta = 2$ 2. Gradient magnitude fitting: $\sigma = 0.5$, $\beta = 16.0$
Frog	<ol style="list-style-type: none"> 1. Interactive 	<ol style="list-style-type: none"> 1. Gradient magnitude fitting: $\sigma = 1.25$, $\beta = 1.0$

Table 1: Parameters for processing example datasets.

during surface reconstruction and object segmentation. One way to address this problem is to scan the same object from multiple directions, with the hope that the small structures will be adequately sampled in one of the scans. Generating several scans of the same object then raises the question of how to properly combine the information contained in these multiple datasets. Simply merging the individual scans does not necessarily assemble enough samples to produce a high resolution volumetric model. To address this problem we have developed a method for deforming a level set model using velocity information derived from multiple volume datasets with non-uniform resolution in order to produce a single high-resolution 3D model [43]. The method locally approximates the values of the multiple datasets by fitting a distance-weighted polynomial using moving least-squares (MLS) [44, 45]. Directional 3D edge information that may be used during the surface deformation stage is readily derived from MLS, and integrated within our segmentation framework.

The proposed method has several beneficial properties. Instead of merging all of the input volumes by global resampling (interpolation), we locally approximate the derivatives of the intensity values by MLS. This local versus global approach is feasible because the level set surface deformation only requires edge information in a narrow band around the surface. Consequently the MLS calculation is only performed in a small region of the volume, rather than throughout the whole volume, making the computational cost proportional to the object surface area [36]. As opposed to many interpolation schemes the MLS method is stable with respect to noise and imperfect registrations [46]. Our implementation also allows for small intensity attenuation artifacts between the multiple scans thereby providing gain-correction. The distance-based weighting employed in our method ensures that the contributions from each scan is properly merged into the final result. If a slice of data from one scan is closer to a point of interest on the model, the information from this scan will contribute more heavily to determining the location of the point.

To the best of our knowledge there is no previous work on creating deformable models directly from multiple volume datasets. While there has been previous work on 3D level set segmentation and reconstruction [41, 6, 5, 8, 47], it has not been based on multiple volume datasets. However, 3D models have been generated from multiple range maps [48, 49, 36, 29], but the 2D nature of these approaches is significantly different from the 3D problem being addressed

here. The most relevant related projects involve merging multiple volumes to produce a single high-resolution volume dataset [50, 51], and extracting edge information from a single non-uniform volume [52]. Our work does not attempt to produce a high-resolution merging of the input data. Instead, our contribution stands apart from previous work because it deforms a model based on local edge information derived from multiple non-uniform volume datasets.

We have demonstrated the effectiveness of our approach on three multi-scan datasets. The first two examples are derived from a single high resolution volume dataset that has been sub-sampled in the X , Y and Z directions. Since these non-uniform scans are extracted from a single dataset they are therefore perfectly aligned. The first scan is derived from a high resolution MR scan of a 12-day-old mouse embryo, which has already had its outer skin isolated with a previous segmentation process. The second example is generated from a laser scan reconstruction of a figurine. The third example consists of multiple MR scans of a zucchini that have been imperfectly aligned by hand. The first two examples show that our method is able to perform level set segmentation from multiple non-uniform scans of an object, picking up and merging features only found in one of the scans. The second example demonstrates that our method generates satisfactory results, even when there are misalignments in the registration.

0.4.1 Method Description

We have formulated our approach to 3D reconstruction of geometric models from multiple non-uniform volumetric datasets within our level set segmentation framework. Recall that speed function $\mathcal{F}()$ describes the velocity at each point on the evolving surface in the direction of the local surface normal. All of the information needed to deform a surface is encapsulated in the speed function, providing a simple, unified approach to evolving the surface. In this section we define speed functions that allow us to solve the multiple-data segmentation problem. The key to constructing suitable speed terms is 3D directional edge information derived from the multiple datasets. This problem is solved using a moving least-squares scheme that extracts edge information by locally fitting sample points to high-order polynomials.

0.4.1.1 Level Set Speed Function for Segmentation

Many different speed functions have been proposed over the years for segmentation of a single volume dataset [41, 6, 5, 8]. Typically such speed functions consist of a (3D) image-based feature attraction term and a smoothing term which serves as a regularization term that lowers the curvature and suppresses noise in the input data. From computer vision it is well known that features, *i.e.* significant changes in the intensity function, are conveniently described by an edge-detector [53]. There exists a very large body of work devoted to the problem of designing optimal edge detectors for 2D images [14, 16], most of which are readily generalized to 3D. For this project we found it convenient to use speed functions with a 3D directional edge term that moves the level set toward the maximum of the gradient magnitude. This gives a term equivalent to Equation 8,

$$\mathcal{F}_{grad}(\mathbf{x}, \mathbf{n}, \phi) = \alpha \mathbf{n} \cdot \nabla \|\nabla V_g\| \quad (10)$$

where α is a scaling factor for the *image-based* feature attraction term $\nabla \|\nabla V_g\|$ and \mathbf{n} is the normal to the level set surface at \mathbf{x} . V_g symbolizes some global uniform merging of the multiple non-uniform input volumes. This feature term is effectively a 3D directional edge-detector of V_g . However there are two problems associated with using this speed function exclusively. The first is that we cannot expect to compute reliable 3D directional edge information in all regions of space simply because of the nature of the non-uniform input volumes. In other words V_g cannot be interpolated reliably in regions of space where there are no nearby sample points. Hence the level set surface will not experience any image-based forces in these regions. The solution is to use a regularization term that imposes constraints on the mean curvature of the deforming level set surface. We include the smoothing term from Equation 6 and scale it with parameter β , in order to smooth the regions where no edge information exists as well as suppress noise in the remaining regions thereby preventing excessive aliasing.

Normally the feature attraction term, $\nabla \|\nabla V_g\|$, creates only a narrow range of influence. In other words, this feature attraction term will only reliably move the portion of the level set surface that is in close proximity to the actual edges in V_g . Thus, a good initialization of the level set surface is needed before solving Equation 10. A reasonable initialization of the level set surface may be obtained by computing the CSG union of the multiple input volumes, which

are first tri-linearly resampled to give a uniform sampling. However, if the input volumes are strongly non-uniform, i.e. they are severely undersampled in one or more directions, their union produces a poor initial model. To improve the initialization we attract the CSG union surface to the Canny edges [16] computed from V_g using the distance transform produced from those edges. See Equation 7. This approach allows us to move the initial surface from a long range, but only with pixel-level accuracy.

Canny edges are non-directional edges defined from the zero-crossing of the second derivative of the image in the direction of the local normal. In 3D this is

$$\frac{\partial^2}{\partial \mathbf{n}_g^2} V_g = 0 \quad (11)$$

where $\mathbf{n}_g \equiv \nabla V_g / \|\nabla V_g\|$ is the local normal vector of V_g . Using the expression $\partial/\partial \mathbf{n}_g = \mathbf{n}_g \cdot \nabla$ we can rewrite Equation 11 as

$$\frac{\partial^2}{\partial \mathbf{n}_g^2} V_g = \mathbf{n}_g \cdot \nabla [\mathbf{n}_g \cdot \nabla V_g] = \mathbf{n}_g \cdot \nabla \|\nabla V_g\|. \quad (12)$$

The next section focuses on the methods needed to reliably compute the vectors \mathbf{n}_g and $\nabla \|\nabla V_g\|$. In preparation, the latter may be explicitly expressed in terms of the derivatives of the merged volume V_g

$$\nabla \|\nabla V_g\| = \frac{\nabla V_g \hat{H} V_g}{\|\nabla V_g\|} \quad (13)$$

where we have defined the gradient vector and the Hessian matrix,

$$\hat{\nabla} V_g = \left(\frac{\partial V_g}{\partial x}, \frac{\partial V_g}{\partial y}, \frac{\partial V_g}{\partial z} \right) \quad (14a)$$

$$\hat{H} V_g = \begin{pmatrix} \frac{\partial^2 V_g}{\partial x^2} & \frac{\partial^2 V_g}{\partial y \partial x} & \frac{\partial^2 V_g}{\partial z \partial x} \\ \frac{\partial^2 V_g}{\partial x \partial y} & \frac{\partial^2 V_g}{\partial y^2} & \frac{\partial^2 V_g}{\partial z \partial y} \\ \frac{\partial^2 V_g}{\partial x \partial z} & \frac{\partial^2 V_g}{\partial y \partial z} & \frac{\partial^2 V_g}{\partial z^2} \end{pmatrix}. \quad (14b)$$

Thus, in closing we note that the level set propagation needed for segmentation only needs information about the first and second order partial derivatives of the input volumes, not the interpolated intensity values themselves.

0.4.1.2 Computing Partial Derivatives

As outlined above the speed function \mathcal{F} in the level set equation, Equation 4, is based on edge information derived from the input volumes. This requires

estimating first and second order partial derivatives from the multiple non-uniform input volumes. We do this by means of moving least-squares (MLS), which is an effective and well established numerical technique for computing derivatives of functions whose values are known only on irregularly spaced points [44, 45, 46].

Let us assume we are given the input volumes \widehat{V}_d , $d = 1, 2, \dots, D$ which are volumetric samplings of an object on the non-uniform grids $\{\widehat{\mathbf{x}}_d\}$. We shall also assume that the local coordinate frames of $\{\widehat{\mathbf{x}}_d\}$ are scaled, rotated and translated with respect to each other. Hence, we define a world coordinate frame (typically one of the local frames) in which we solve the level set equation. Now, let us define the world sample points $\{\mathbf{x}_d\}$ as

$$\mathbf{x}_d \equiv \mathbf{T}^{(d)}[\widehat{\mathbf{x}}_d] \quad (15)$$

where $\mathbf{T}^{(d)}$ is the coordinate transformation from a local frame d to the world frame. Next we locally approximate the intensity values from the input volumes \widehat{V}_d with a 3D polynomial expansion. Thus, we define the N-order polynomials

$$V_N^{(d)}(\mathbf{x}) = C_{000}^{(d)} + \sum_{i+j+k=1}^N C_{ijk}^{(0)} x^i y^j z^k, \quad d = 1, 2, \dots, D \quad (16)$$

where the C coefficients are unknown. Note that these local approximations to the intensity values share coefficients $C_{ijk}^{(0)}$ of order higher than zero, *i.e.* all of the functions $V_N^{(d)}$, $d = 1, 2, \dots, D$ have the *same edges*. The fact that the zero-order term in Equation 16 is input volume dependent means we allow for local constant offsets between the input volumes \widehat{V}_d . This effectively provides built-in gain-correction in the scheme, since it can handle small intensity attenuation artifacts between the multiple scans.

Moving Least-Squares To solve for the expansion coefficients C in Equation 16 we define the moving least-squares functional

$$E(\mathbf{x}_0) = \sum_{d=1}^D \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[V_N^{(d)}(\mathbf{x}_d - \mathbf{x}_0) - V_d(\mathbf{x}_d) \right]^2 \quad (17)$$

where \mathbf{x}_0 is the expansion point from where we are seeking edge information, $V_d(\mathbf{x}_d) \equiv \widehat{V}_d(\widehat{\mathbf{x}}_d)$ and where

$$w_d(\mathbf{x}) \equiv \begin{cases} 1 - 2(\|\mathbf{x}\|/\Delta)^2 & \text{for } 0 \leq \|\mathbf{x}\| \leq \Delta/2 \\ 2(\|\mathbf{x}\|/\Delta - 1)^2 & \text{for } \Delta/2 < \|\mathbf{x}\| < \Delta \\ 0 & \text{for } \|\mathbf{x}\| \geq \Delta \end{cases} \quad (18)$$

is a ‘‘moving filter’’ that weights the contribution of different sampling points, \mathbf{x}_d , according to their Euclidean distance, $\|\mathbf{x}_d - \mathbf{x}_0\|$, to the expansion point, \mathbf{x}_0 . Other expressions for this weighting function could of course be used, but Equation 18 is fast to compute, has finite support (by the window parameter Δ), and its tangent is zero at the endpoints. After substitution of Equation 16 into Equation 17 we obtain the functional

$$E(\mathbf{x}_0) = \sum_{d=1}^D \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[C_{000}^{(d)} - \widehat{V}_d(\mathbf{x}_d) + \sum_{i+j+k=1}^N C_{ijk}^{(0)} (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \right]^2. \quad (19)$$

The minimization of this moving least-squares functional with respect to the expansion coefficients C requires the partial derivatives to vanish, *i.e.*

$$\frac{\partial \widehat{E}(\mathbf{x}_0)}{\partial C_{000}^{(d)}} = 0 = 2 \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[C_{000}^{(d)} - \widehat{V}_d(\mathbf{x}_d) + \sum_{i+j+k=1}^N C_{ijk}^{(0)} (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \right] \quad (20a)$$

$$\begin{aligned} \frac{\partial \widehat{E}(\mathbf{x}_0)}{\partial C_{lnm}^{(0)}} = 0 = & 2 \sum_{d=1}^D \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[C_{000}^{(d)} - \widehat{V}_d(\mathbf{x}_d) \right. \\ & + \sum_{i+j+k=1}^N C_{ijk}^{(0)} (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \left. \right] \\ & \times (x_d - x_0)^l (y_d - y_0)^m (z_d - z_0)^n. \end{aligned} \quad (20b)$$

This defines a system of linear equations in the expansion coefficients $C_{ijk}^{(r)}$, that can be solved using standard techniques from numerical analysis, see Equation 21 and Equation 23.

Equation 20a and Equation 20b can then be conveniently expressed as

$$\sum_{\mathbf{q}} \mathbf{A}_{\mathbf{p},\mathbf{q}} \mathbf{c}_{\mathbf{q}} = \mathbf{b}_{\mathbf{p}} \quad (21)$$

where \mathbf{A} is a diagonal matrix, and \mathbf{b} , \mathbf{c} are vectors. In this equation we have also introduced the compact index notations $\mathbf{p} \equiv (i, j, k, r)$ and $\mathbf{q} \equiv (l, m, n, s)$ defined as

$$\begin{aligned} \mathbf{p} \in & \{i, j, k, r \in \mathcal{N}^+ \mid i = j = k = 0, 1 \leq r \leq D\} \\ & \cup \{i, j, k, r \in \mathcal{N}^+ \mid 1 \leq i+j+k \leq N, r = 0\} \end{aligned} \quad (22a)$$

$$\begin{aligned} \mathbf{q} \in & \{l, m, n, s \in \mathcal{N}^+ \mid l = m = n = 0, 1 \leq s \leq D\} \\ & \cup \{l, m, n, s \in \mathcal{N}^+ \mid 1 \leq l+m+n \leq N, s = 0\}. \end{aligned} \quad (22b)$$

The diagonal matrix \mathbf{A} , and the vectors \mathbf{b} , \mathbf{c} in Equation 21 are defined as

$$\begin{aligned} \mathbf{A}_{\mathbf{p},\mathbf{q}} \equiv & \sum_d (\delta_{r,d} + \delta_{r,0}) (\delta_{s,d} + \delta_{s,0}) \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \\ & \times (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \end{aligned} \quad (23a)$$

$$\begin{aligned} & \times (x_d - x_0)^l (y_d - y_0)^m (z_d - z_0)^n \\ \mathbf{b}_{\mathbf{p}} \equiv & \sum_d (\delta_{r,d} + \delta_{r,0}) w_d(\mathbf{x}_d - \mathbf{x}_0) \widehat{V}_d(\mathbf{x}_d) \\ & \times (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \end{aligned} \quad (23b)$$

$$\mathbf{c}_{\mathbf{p}} \equiv C_{ijk}^{(r)}. \quad (23c)$$

Next the matrix equation $\mathbf{A}\mathbf{c} = \mathbf{b}$ must be solved for the vector \mathbf{c} of dimension $\binom{N+3}{3} + D - 1$, where N is the order of the expansion in Equation 16 and D is the number of non-uniform input volumes. As is well known for many moving least-square problems it is possible for the condition number of the matrix \mathbf{A} to become very large. Any matrix is singular if its condition number is infinite and can be defined as ill-conditioned if the reciprocal of its condition number approaches the computer's floating-point precision. This can occur if the problem is over-determined (number of sample points, \mathbf{x}_d greater than number of coefficients C) and under-determined (ambiguous combinations of the coefficients C work equally well or equally bad). To avoid such numerical problems, a singular value decomposition (SVD) linear equation solver is recommended for use in combination with the moving least-squares method. The

SVD solver identifies equations in the matrix \mathbf{A} that are, within a specified tolerance, redundant (*i.e.* linear combinations of the remaining equations) and eliminates them thereby improving the condition number of the matrix. We refer the reader to reference [54] for a helpful discussion of SVD pertinent to linear least-squares problems.

Once we have the expansion coefficients \mathbf{c} we can readily express the Hessian matrix and the gradient vector of the combined input volumes as

$$\nabla V = (C_{100}^{(0)}, C_{010}^{(0)}, C_{001}^{(0)}) \quad (24a)$$

$$\mathbf{H}V = \begin{pmatrix} 2C_{200}^{(0)} & C_{110}^{(0)} & C_{101}^{(0)} \\ C_{110}^{(0)} & 2C_{020}^{(0)} & C_{011}^{(0)} \\ C_{101}^{(0)} & C_{011}^{(0)} & 2C_{002}^{(0)} \end{pmatrix} \quad (24b)$$

evaluated at the moving expansion point \mathbf{x}_0 . This in turn is used in Equation 13 to compute the edge information needed to drive the level set surface.

0.4.1.3 Algorithm Overview

Algorithm 1 describes the main steps of our approach. The initialization routine, Algorithm 2, is called for all of the multiple non-uniform input volumes, V_d . Each non-uniform input dataset is uniformly resampled in a common coordinate frame (V_0 's) using tri-linear interpolation. Edge information and the union, V_0 , of the V_d 's is then computed. Algorithm 2 calculates Canny and 3D directional edge information using moving least-squares in a narrow band in each of the resampled input volumes, V_d , and buffers this in V_{edge} and \mathbf{V}_{grad} . Next Algorithm 1 computes the distance transform of the zero-crossings of the Canny edges and takes the gradient of this scalar volume to produce a vector field \mathbf{V}_{edge} , which pulls the initial level set model to the Canny edges. Finally the level set model is attracted to the 3D directional edges of the multiple input volumes, \mathbf{V}_{grad} , and a Marching Cubes mesh is extracted for visualization. The level set solver, described in Algorithm 3, solves Equation 4 using the ‘‘up-wind scheme’’ (not explicitly defined) and the sparse-field narrow-band method of [36], with V_0 as the initialization and \mathbf{V}_{edge} and \mathbf{V}_{grad} as the force field in the speed function.

Algorithm 1: MAIN(V_1, \dots, V_D)

comment: V_1, \dots, V_D are non-uniform samplings of object V

global $V_{edge}, \mathbf{V}_{grad}$

do $\left\{ \begin{array}{l} V_0 \leftarrow \text{uniform sampling of empty space} \\ \text{for } d \leftarrow 1 \text{ to } D \\ \quad \text{do } V_0 \leftarrow V_0 \cup \text{INITIALIZATION}(V_d) \\ V_{edge} \leftarrow \nabla[\text{distance transform}[\text{zero-crossing}[V_{edge}]]] \\ V_0 \leftarrow \text{SOLVELEVELSETEQ}(V_0, \mathbf{V}_{edge}, \alpha, 0) \\ V_0 \leftarrow \text{SOLVELEVELSETEQ}(V_0, \mathbf{V}_{grad}, \alpha, \beta) \end{array} \right.$

return (Marching Cubes mesh of V_0)

Algorithm 2: INITIALIZATION(V_d)

comment: Pre-processing to produce good LS initialization

do $\left\{ \begin{array}{l} V_d \leftarrow \text{Uniform tri-linear resampling of } V_d \\ \Gamma_d \leftarrow \text{Set of voxels in narrow band of iso-surface of } V_d \\ \text{for each } \mathbf{x}_0 \in \Gamma_d \\ \quad \text{do } \left\{ \begin{array}{l} \text{Solve moving least-squares problem at } \mathbf{x}_0 \\ V_{edge}(\mathbf{x}_0) \leftarrow \text{scalar Canny edge, cf. Equation 12} \\ \mathbf{V}_{grad}(\mathbf{x}_0) \leftarrow \text{3D directional edge, cf. Equation 13} \end{array} \right. \end{array} \right.$

return (V_d)

Algorithm 3: SOLVELEVELSETEQ($V_0, \mathbf{V}, \alpha, \beta$)

comment: Solve Equation 4 with initial condition $\phi(t=0) = V_0$

do $\left\{ \begin{array}{l} \phi \leftarrow V_0 \\ \text{repeat} \\ \quad \left\{ \begin{array}{l} \Gamma \leftarrow \text{Set of voxels in narrow band of iso-surface of } \phi \\ \Delta t \leftarrow \gamma / \sup_{\mathbf{x} \in \Gamma} \|\mathbf{V}(\mathbf{x})\|, \gamma \leq 1 \\ \text{for each } \mathbf{x} \in \Gamma \\ \quad \text{do } \left\{ \begin{array}{l} \mathbf{n} \leftarrow \text{upwind scheme}[-\nabla\phi(\mathbf{x})/\|\nabla\phi(\mathbf{x})\|] \\ \dot{\phi}(\mathbf{x}) \leftarrow \|\nabla\phi(\mathbf{x})\|(\alpha\mathbf{V}(\mathbf{x}) \cdot \mathbf{n} + \beta\nabla \cdot \mathbf{n}) \\ \phi(\mathbf{x}) \leftarrow \phi(\mathbf{x}) + \dot{\phi}(\mathbf{x})\Delta t \end{array} \right. \end{array} \right. \\ \text{until } \sup_{\mathbf{x} \in \Gamma} \|\dot{\phi}(\mathbf{x})\| \leq \epsilon \end{array} \right.$

return (ϕ)

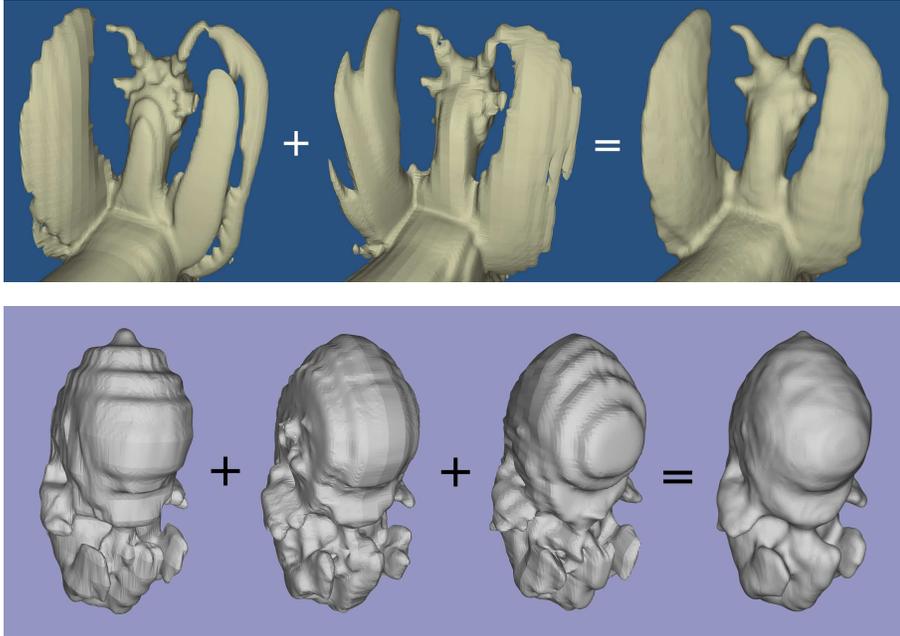


Figure 9: Non-uniform datasets merged to produce high resolution level set models, (top) laser scan of a figurine, (bottom) MR scan of a mouse embryo.

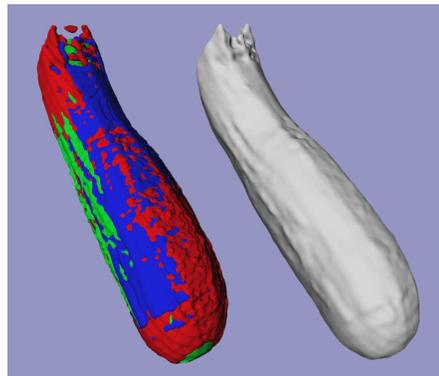


Figure 10: Three low resolution MR scans of a zucchini that have been individually colored and overlaid to demonstrate their imperfect alignment. The level set model on the right is derived from the three low resolution scans.

0.4.2 Multiple Volume Results

We have applied our segmentation method to several multi-scan non-uniform datasets to produce high resolution level set models. The parameters used for these segmentations are listed in Table 2. α and β are weights that the user adjusts to balance attraction to edges with curvature-based smoothing during the level set deformation process.

Table 2: Maximum in-plane to out-of-plane sampling ratios of non-uniform input datasets, and parameters for the two level set speed terms defined in Equation 6 and Equation 10.

Model	Origin	Ratio	α	β
Griffin	Laser scan	6/10:1	1.0	0.5
Mouse	MR scan	10:1	1.0	0.5
Zucchini	MR scan	10:1	1.0	0.5

0.4.2.1 Griffin Dataset

The griffin dataset was created with a volumetric laser scan reconstruction algorithm [49]. This algorithm creates a high resolution volumetric representation of an object by merging multiple depth maps produced via a laser scan. The original griffin dataset has a resolution of $312 \times 294 \times 144$. We have extracted two non-uniform datasets from this high resolution representation by copying every sixth plane of data in the X direction and every tenth plane in the Y direction. The two derived non-uniform griffin datasets have the following resolution: $52 \times 294 \times 144$ and $312 \times 30 \times 144$. Iso-surfaces have been extracted from these datasets, appropriately scaled in the low resolution direction, and are presented in the first two images in Figure 9 (top). Each low resolution scan inadequately captures some important geometric feature of the griffin. We have performed a reconstructions from the undersampled non-uniform scans to produce the result in Figure 9 (top). The method produces a high resolution ($312 \times 294 \times 144$) level set model that contains all of the significant features of the original scan.

0.4.2.2 Mouse Embryo Dataset

The first three scans in Figure 9 (bottom) are derived from a high resolution MR scan of a mouse embryo. They are subsampled versions of a $256 \times 128 \times 128$ volume dataset, and have the following resolutions: $26 \times 128 \times 128$, $256 \times 16 \times 128$ and $256 \times 128 \times 13$. The last image in Figure 9 presents the result produced by our multi-scan segmentation method. The information in the first three scans has been successfully used to create a level set model of the embryo with a resolution of $256 \times 128 \times 130$. The finer features of the mouse embryo, namely its hands and feet, have been reconstructed.

0.4.2.3 Zucchini Dataset

The final dataset consists of three individual MRI scans of an actual zucchini. The separate scans have been registered manually and are presented on the left side of Figure 10, each with a different color. The resolutions of the individual scans are $28 \times 218 \times 188$, $244 \times 25 \times 188$ and $244 \times 218 \times 21$. This image highlights the rough alignment of the scans. The right side of Figure 10 presents the result of our level set segmentation. It demonstrates that our approach is able to extract a reasonable model from multiple datasets that are imperfectly aligned.

0.5 Segmentation of DT-MRI Brain Data

Diffusion tensor magnetic resonance imaging[55, 56] (DT-MRI) is a technique used to measure the diffusion properties of water molecules in tissues. Anisotropic diffusion can be described by the equation

$$\frac{\partial C}{\partial t} = \nabla \cdot (\mathbf{D} \nabla C) \quad (25)$$

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

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix}. \quad (26)$$

Figure 11 presents a “slice” of the diffusion tensor volume data of human brain used in our study. Each sub-image presents the scalar values of the associated diffusion tensor component for one slice of the dataset.

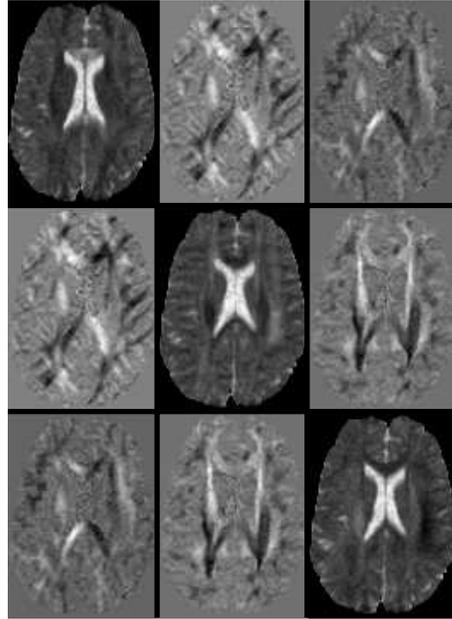


Figure 11: Slice of a tensor volume where every “element” of the image matrix corresponds to one component of the tensor \mathbf{D} .

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 of all, 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, 3D visualization and segmentation techniques available today are predominantly designed for scalar and sometimes vector fields. Thus, there are three fundamental problems in tensor imaging: a) finding an invariant representation of a tensor that is independent of a frame of reference, b) constructing a mapping from the tensor field to a scalar or vector field, and c) visualization and classification of tissue using the derived scalar fields.

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

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.

Using eigenvalues/eigenvectors one can compute different anisotropy measures [55, 57, 58, 59] that map tensor data onto scalars and can be used for further visualization and segmentation. Although eigenvalue/vector computation of the 3x3 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-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[60]. 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 which is rotationally invariant, does not require eigenvalue computations and is stable with respect to noise. Tensor invariants with these characteristics were first proposed by Ulug and Zijl[61]. In Section 0.5.1 we formulate a new anisotropy measure for tensor field based on these invariants.

Visualization and model extraction from the invariant 3D 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[62]. This method was developed for 2D slices and creates visual cluttering when used in 3D. Other standard CFD visualization techniques like tensor-lines do not provide meaningful results for the MRI data due to rapidly changing directions and magnitudes of eigenvector/values and the amount of noise present in the data. Recently Kindlmann[63] developed a volume rendering approach to tensor field visualization using eigenvalue-based anisotropy measures to construct transfer functions and color maps that highlight some brain structures and diffusion patterns.

In our work we perform iso-surfacing on the 3D scalar fields derived from our tensor invariants to visualize and segment the data [64]. An advantage of iso-surfacing over other approaches is that it can provide the shape information

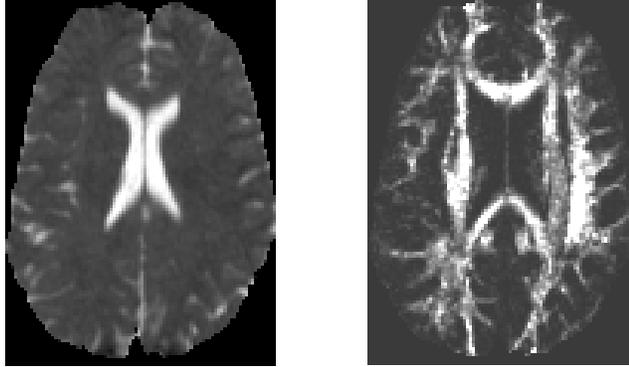


Figure 12: Isotropic C_1 (left) and anisotropic C_a (right) tensor invariants for the tensor slice shown in Figure 11.

needed for constructing geometric models, and computing internal volumes and external surface areas of the extracted regions. There has also been a number of recent publications[65, 66] 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 signal-to-noise ratio than the data used in our study.

0.5.1 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 the diffusion tensor (matrix) D , which are the roots of corresponding characteristic equation

$$\lambda^3 - C_1 \cdot \lambda^2 + C_2 \cdot \lambda - C_3 = 0, \quad (27)$$

with coefficients

$$\begin{aligned} C_1 &= D_{xx} + D_{yy} + D_{zz} \\ C_2 &= D_{xx}D_{yy} - D_{xy}D_{yx} + D_{xx}D_{zz} - D_{xz}D_{zx} + \\ &\quad D_{yy}D_{zz} - D_{yz}D_{zy} \\ C_3 &= D_{xx}(D_{yy}D_{zz} - D_{zy}D_{yz}) \\ &\quad - D_{xy}(D_{yx}D_{zz} - D_{zx}D_{yz}) + D_{xz}(D_{yx}D_{zy} - D_{zx}D_{yy}). \end{aligned} \quad (28)$$

Since the roots of Equation (27) 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

$$\begin{aligned} C_1 &= \lambda_1 + \lambda_2 + \lambda_3 \\ C_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 \\ C_3 &= \lambda_1\lambda_2\lambda_3 \end{aligned} \tag{29}$$

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 the characteristic equation, they are less sensitive to noise, then roots λ_i of the same equation.

Any combination of the above 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

$$\frac{C_1C_2}{C_3} = 3 + \frac{\lambda_2 + \lambda_3}{\lambda_1} + \frac{\lambda_1 + \lambda_3}{\lambda_2} + \frac{\lambda_1 + \lambda_2}{\lambda_3} \tag{30}$$

and we can define a new dimensionless anisotropy measure

$$C_a = \frac{1}{6} \left[\frac{C_1C_2}{C_3} - 3 \right]. \tag{31}$$

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

$$C_a^{limit} \approx \frac{1}{3} \left[1 + \frac{\lambda_1}{\lambda_3} + \frac{\lambda_3}{\lambda_1} \right]. \tag{32}$$

Thus C_a is always $\sim \lambda_{max}/\lambda_{min}$ and measures the magnitude of the diffusion anisotropy. We again want to emphasize that we use the 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 Equations (5) and (31).

0.5.2 Geometric Modeling

Two options are usually available for viewing the scalar volume datasets, direct volume rendering[1, 4] and volume segmentation[67] combined with con-

ventional 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 it is difficult to construct the exact transfer function that highlights the desired structures in the volume dataset[68]. Our approach instead focuses on extracting geometric models of the structures embedded 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 anatomical studies and computational simulation, for example EEG/MEG modeling within the brain[69]. 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. The models may also be used within a surgical planning/simulation/VR environment[70], 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[71]. 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[72]. This technique creates a polygonal model that approximates the iso-surface embedded in a scalar volume dataset for a particular iso-value. 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 the first images in Figures 13 and 16. 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 the tools in our level set framework to smooth the data and remove the noise-related artifacts.

0.5.3 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 dataset from

a human volunteer which contains 60 slices each of 128x128 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 Section 0.5.1. 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 Figure 12 (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 Figure 12 (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 measurable regions with planar diffusion anisotropy. We therefore only need two scalar volumes in order to segment the DT dataset.

We then utilize our level set framework to extract smoothed models from the two derived scalar volumes. First the input data is filtered with a low-pass Gaussian filter ($\sigma \approx 0.5$) to blur the data and thereby reduce noise. Next, the volume voxels are classified for inclusion/exclusion in the initialization 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 grey scale images, such as those used in this paper, the classification is equivalent to high and low thresholding operations. The last initialization step consists of performing a set of topological (e.g. flood fill) operations in order to remove small pieces or holes from objects. This is followed by a level set deformation that pulls the surface toward local maxima of the gradient magnitude and smooths it with a curvature-based motion. This moves the surface toward specific features in the data, while minimizing the influence of noise in the data.

Figures 13 and 14 present two models that we extracted from DT-MRI volume datasets using our techniques. Figure 13 contains segmentations from volume \mathcal{V}_1 , the measure of total diffusivity. The top image shows a Marching Cubes iso-surface using an iso-value of 7.5. In the bottom we have extracted

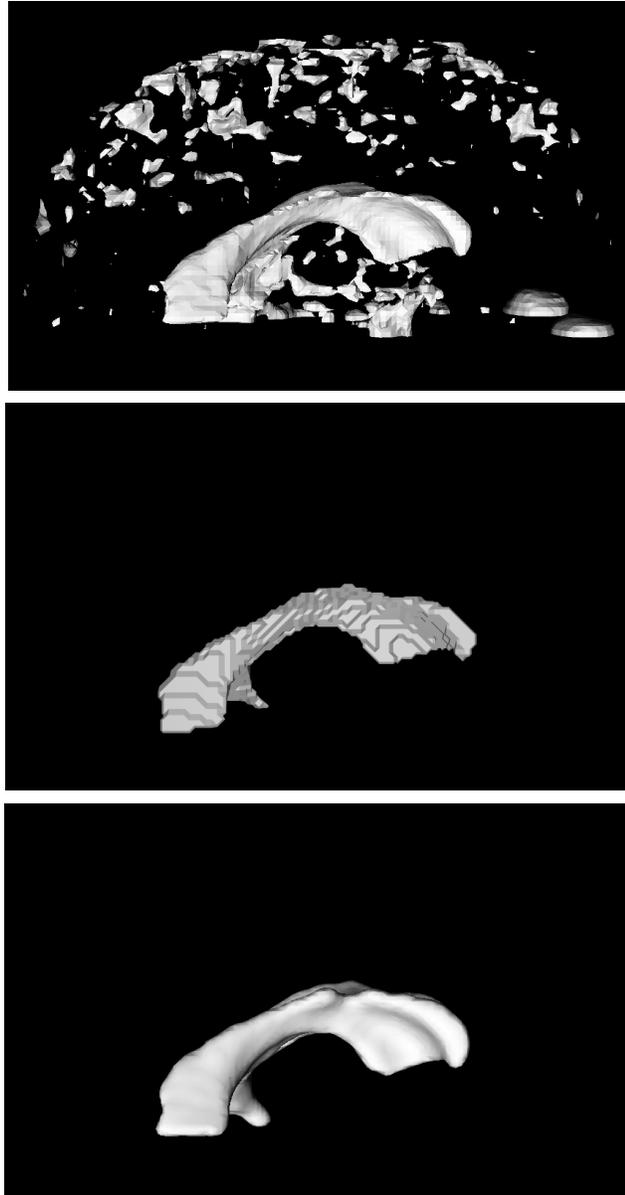


Figure 13: Segmentation from isotropic measure volume \mathcal{V}_1 for the first DT-MRI dataset. The first row is the marching cubes iso-surface with iso-value 7.5. The second row is the result of flood-fill algorithm applied to the same volume and used for initialization. The third row is the final level set model.

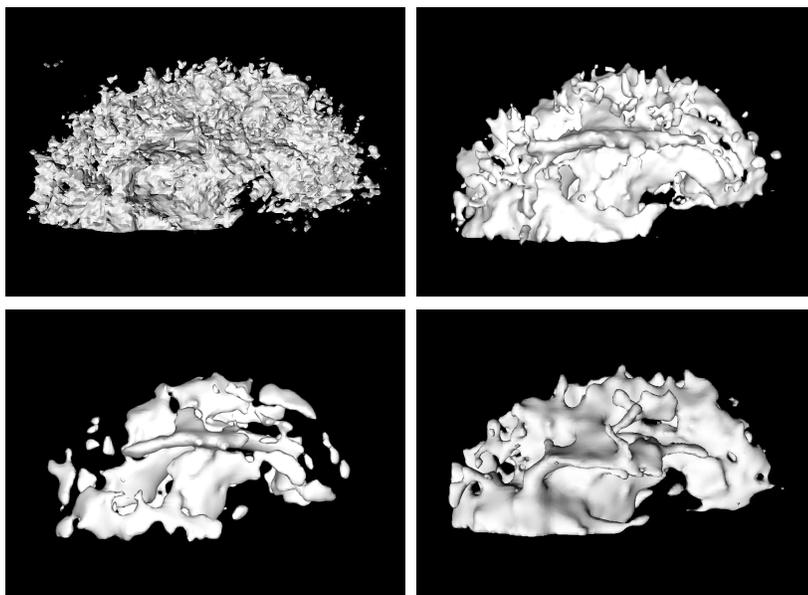


Figure 14: Model segmentation from volume \mathcal{V}_2 . Top left image is an iso-surface of value 1.3, used for initialization of the level set. Clockwise, are the results of level set development with corresponding β values of 0.2, 0.4 and 0.5.

just the ventricles from \mathcal{V}_1 . This is 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 a level set deformation, using a β value of 0.2.

Figure 14 again provides the comparison between direct iso-surfacing and level set modeling, 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 14 contains three models extracted from \mathcal{V}_2 at different values of smoothing parameter β used during

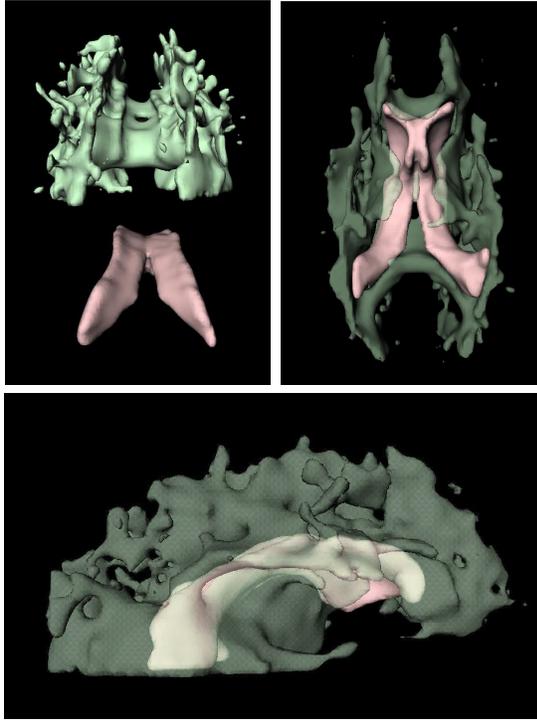


Figure 15: Combined model of ventricles and (semi-transparent) anisotropic regions: rear, exploded view (left), bottom view (right), side view (bottom). Note how model of ventricles extracted from isotropic measure dataset \mathcal{V}_1 fits into model extracted from anisotropic measure dataset \mathcal{V}_2 .

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 Figure 14. The top-right side of this figure presents a model extracted from \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 Figure 14. 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.

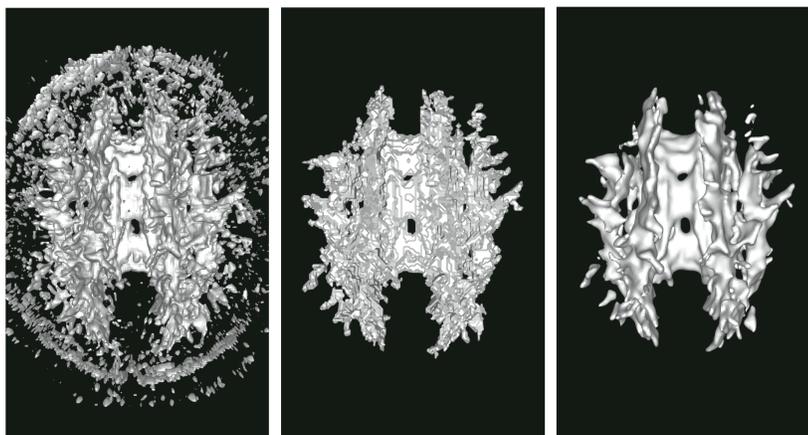


Figure 16: Segmentation using anisotropic measure \mathcal{V}_2 from the second DT-MRI dataset. (left) Marching cubes iso-surface with iso-value 1.3. (middle) Result of flood-fill algorithm applied to the volume and used for initialization. (right) Final level set model.

We can also bring together the two models extracted from datasets \mathcal{V}_1 and \mathcal{V}_2 into a single image. They will have perfect alignment since they are derived from the same DT-MRI dataset. Figure 15 demonstrates that we are able to isolate different structures in the brain from a single DT-MRI scan 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.

Finally, to verify the validity of our approach we applied it to the second dataset from a different volunteer. This dataset has 20 slices of the 256x256 resolution. We generated the anisotropy measure volume \mathcal{V}_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 Figure 16, and demonstrate the generality of our approach.

0.6 Direct Estimation of Surfaces in Tomographic Data

The radon transform is invertible (albeit, marginally so) when the measured data consists of a sufficient number of good quality, properly spaced projections [73]. However, for many applications the number of realizable projections is insufficient, and direct greyscale reconstructions are susceptible to artifacts. We will refer to such problems as *under-constrained* tomographic problems. Cases of under-constrained tomographic problems usually fall into one of two classes. The first class is where the measuring device produces a relatively dense set of projections (i.e. adequately spaced) that do not span a full 180 degrees. In these cases, the sinogram contains regions without measured data. Considering the radon transform in the Fourier domain, these missing regions of the sinogram correspond to a transform with angular wedges (pie slices) that are null, making the transform noninvertible. We assume that these missing regions are large enough to preclude any straight-forward interpolation in the frequency domain. The second class of incomplete tomographic problems are those that consist of an insufficient number of widely spaced projections. We assume that these sparse samples of the sinogram space are well distributed over a wide range of angles. For this discussion the precise spacing is not important. This problem is characterized by very little data in the Fourier domain, and direct inversion approaches produce severe artifacts. Difficulties in reconstructing volumes from such incomplete tomographic datasets are often aggravated by noise in the measurements and misalignments among projections.

Under-constrained problems are typically solved using one or both of two different strategies. The first strategy is to choose from among feasible solutions (those that match the data) by imposing some additional criterion, such as finding the solution that minimizes an energy function. This additional criterion should be designed to capture some desirable property, such as minimum entropy. The second strategy is to parameterize the solution in a way that reduces the number of degrees of freedom. Normally, the model should contain few enough parameters so that the resulting parameter estimation problem is over constrained. In such situations solutions are allowed to differ from the data in a way that accounts for noise in the measurements.

In this section we consider a special class of under constrained tomographic

problems that permits the use of a simplifying model. The class of problems we consider are those in which the imaging process is targeted toward tissues or organs that have been set apart from the other anatomy by some contrast agent. This agent could be an opaque dye, as in the case of transmission tomography, or an emissive metabolite, as in nuclear medicine. We assume that this agent produces relatively homogeneous patches that are bounded by areas of high contrast. This assumption is reasonable, for instance, in subtractive angiography or CT studies of the colon. The proposed approach, therefore, seeks to find the boundaries of different regions in a volume by estimating sets of closed surface models and their associated density parameters directly from the incomplete sinogram data [74]. Thus, the reconstruction problem is converted to a segmentation problem. Of course, we can never expect real tissues to exhibit truly homogeneous densities. However, we assert that when inhomogeneities are somewhat uncorrelated and of low contrast the proposed model is adequate to obtain acceptable reconstructions.

0.6.1 Related Work

Several areas of distinct areas of research in medical imaging, computer vision, and inverse problems impact this work. Numerous tomographic reconstruction methods are described in the literature [75, 76], and the method of choice depends on the quality of projection data. Filtered back projection (FBP), the most widely used approach, works well in the case the fully constrained reconstruction where one is given *enough high-quality projections over 180 degree angular range*. Statistical, iterative approaches such as maximum likelihood (ML) and maximum a posteriori (MAP) estimation have been proven to work well with *noisy* projection data, but do not systematically address the under constrained reconstruction problem and generally rely on complete datasets. An exception is [77], which proposes an iterative algebraic approach that includes some assumptions about the homogeneity of the solution to compute a full greyscale reconstruction. Also, some hybrid approaches [78, 79] are specifically developed to deal with *limited-angle* tomography by extrapolating the missing sinogram data.

Other tomographics reconstruction techniques have been proposed, for example those that utilize discrete tomography strategies [80, 73, 81, 82], and deformable models [83, 84, 85, 86, 87]. The literature also describes many

examples of level sets as curve and surface models for image segmentation [7, 6, 41, 88]. The authors have examined their usefulness for 3D segmentation of TEM *reconstructions* [37]. Several authors have proposed solving *inverse problems* using level sets [89, 90, 91, 92, 93, 94, 95], but are mostly limited to solving 2D problems.

We make several important contributions to this previous body of work; first we give a formal derivation of the motion of deformable surface models as the first variation of an error term that relates the projected model to the noisy tomographic data. This formulation does not assume any specific surface representation, and therefore applies to a wide range of tomographic, surface-fitting problems. Second we present a level set implementation of this formulation that computes incremental changes in the radon transform of the projected model only along the wave front, which makes it practical on large datasets. Third we examine the specific problem of initializing the deformable surface in the absence of complete sinogram data, and demonstrate, using real and synthetic data, the effectiveness of direct surface estimation for a specific class of tomographic problems which are under constrained.

0.6.2 Mathematical Formulation

As an introduction, we begin with the derivation of surface estimation problem in two dimensions. The goal is to simultaneously estimate the interface between two materials and their densities, β_0 and β_1 . Thus we have a background, with density β_0 and collection of solid objects with density β_1 . We denote the (open) set of points in those regions as Ω , the closure of that set, the surface, as \mathcal{S} .

The projection of a 2D signal $f(x, y)$ produces a sinogram given by the radon transform as

$$p(s, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(R_\theta \mathbf{x} - s) d\mathbf{x}, \quad (33)$$

where $R_\theta \mathbf{x} = x \cos(\theta) + y \sin(\theta)$ is a rotation and projection of a point $\mathbf{x} = (x, y)$ onto the imaging plane associated with θ . The 3D formulation is the same, except that the signal $f(x, y, z)$ produces a collection of images. We denote the projection of the model, which includes estimates of the objects and the background, as $\hat{p}(s, \theta)$. For this work we denote the angles associated with a discrete set of projections as $\theta_1, \dots, \theta_N$ and denote the domain of each projection as $S = s_1, \dots, s_M$. Our strategy is to find Ω , β_0 , and β_1 by maximizing the

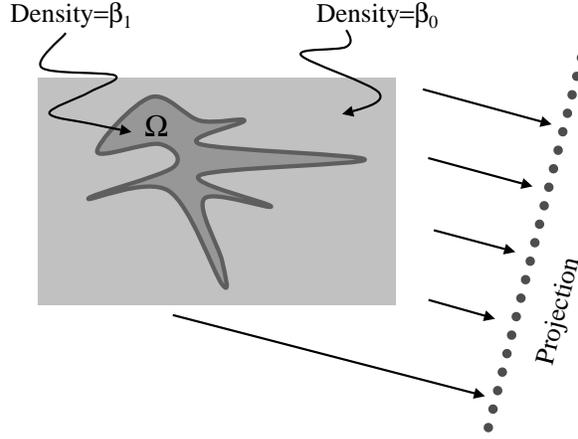


Figure 17: The model is the interface between two densities, which are projected onto the imaging plane to create $\hat{p}(s, \theta_i)$.

likelihood.

If we assume the projection measurements are corrupted by independent noise, the log likelihood of a collection of measurements for a specific shape and density estimate is the probability of those measurements conditional on the model.

$$\ln P(p(s_1, \theta_1), p(s_2, \theta_1), \dots, p(s_M, \theta_N) | \mathcal{S}, \beta_0, \beta_1) = \sum_i \sum_j \ln P(p(s_j, \theta_i) | \mathcal{S}, \beta_0, \beta_1). \quad (34)$$

We call the negative log likelihood the *error* and denote it E_{data} . Normally, the probability density of a measurement is parameterized by the ideal value, which gives

$$E_{\text{data}} = \sum_{i=1}^N \sum_{j=1}^M E(\hat{p}_{ij}, p_{ij}), \quad (35)$$

where $E(\hat{p}_{i,j}, p_{i,j}) = -\ln P(\hat{p}_{i,j}, p_{i,j})$ is the error associated with a particular point in the radon space, and $p_{i,j} = p(s_j, \theta_i)$. In the case of independent Gaussian noise, E is a quadratic, and the log likelihood results in a weighted least squares in the radon space. For all of our results, we use a Gaussian noise model. Next we apply the object model, shown in Figure 17, to the reconstruction of f . If we let $g(x, y)$ be a binary inside-outside function on Ω ,

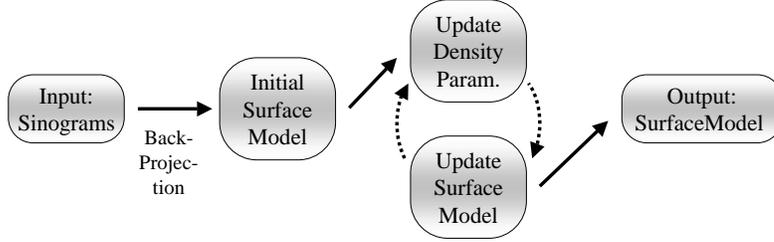


Figure 18: The reconstruction strategy starts with an initial surface estimate and iteratively modifies its shape and the associated density parameters to achieve a good fit to the input data.

then we have the following approximation to $f(x, y)$:

$$f(x, y) \approx \beta_0 + [\beta_1 - \beta_0]g(x, y). \quad (36)$$

Applying the radon transform to the model and substituting for \hat{p} , gives

$$E_{\text{data}} = \sum_{i=1}^N \sum_{j=1}^M E \left(\beta_0 K(s_j, \theta_i) + [\beta_1 - \beta_0] \int_{\Omega} \delta(R_{\theta_i} \mathbf{x} - s_j) d\mathbf{x}, p_{ij} \right), \quad (37)$$

where $K(s_j, \theta_i)$ is the projection of the background— it depends on the geometry of the region over which the data is taken, and is independent of the surface estimate. For some applications we know that $\beta_0 = 0$, and the term $\beta_0 K$ is zero. The integral over Ω results from integrating g over the entire domain.

The proposed strategy is to alternately (i.e. separately) update the shape of the surface model and the density parameters. For the surface shape, a gradient descent minimization approach describes the deformation of the surface, with respect to an evolution parameter t , as it progressively improves its fit to the sinogram data. The incremental change in the likelihood is

$$\frac{dE_{\text{data}}}{dt} = \int_S \sum_{i=1}^N \sum_{j=1}^M \frac{d}{dt} E(\hat{p}_{ij}, p_{i,j}) d\mathbf{x} = \int_S \sum_{i=1}^N \sum_{j=1}^M E'(\hat{p}_{ij}, p_{ij}) \frac{d\hat{p}_{ij}}{dt} d\mathbf{x}, \quad (38)$$

where $E' = \partial E / \partial \hat{p}$, which, for Gaussian noise, is simply the difference between \hat{p} and p . Next we must formulate $d\hat{p}/dt$, which, by the transport equation, is

$$\begin{aligned} \frac{d\hat{p}_{ij}}{dt} &= [\beta_1 - \beta_0] \frac{d}{dt} \int_{\Omega} \delta(R_{\theta_i} \mathbf{x} - s_j) d\mathbf{x} \\ &= [\beta_1 - \beta_0] \int_S \delta(R_{\theta_i} \mathbf{x} - s_j) \mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (39)$$

where \mathbf{n} is an outward pointing surface normal and $\mathbf{v}(\mathbf{x})$ is the velocity of the surface at the point \mathbf{x} . The derivative of E_{data} with respect to surface motion is therefore

$$\frac{dE_{\text{data}}}{dt} = [\beta_1 - \beta_0] \int_{\mathcal{S}} \sum_{i=1}^N \sum_{j=1}^M E'(\hat{p}_{i,j}, p_{ij}) \delta(R_{\theta_i} \mathbf{x} - s_j) \mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x}. \quad (40)$$

Note that the integral over $d\mathbf{x}$ and the δ functional serve merely to associate s_j in the i th scan with the appropriate \mathbf{x} point. If the samples in each projection are sufficiently dense, we can approximate the sum over j as an integral over the image domain, and thus for every \mathbf{x} on the surface there is a mapping back into the i th projection. We denote this point $s_i(\mathbf{x})$. This gives a closed-form expression for the derivative of the derivative of E_{data} in terms of the surface velocity.

$$\frac{dE_{\text{data}}}{dt} = [\beta_1 - \beta_0] \int_{\mathcal{S}} \sum_{i=1}^N e_i(\mathbf{x}) \mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x}, \quad (41)$$

where $e_i(\mathbf{x}) = E'(\hat{p}(s_i(\mathbf{x}), \theta_i), p(s_i(\mathbf{x}), \theta_i))$ is the derivative of the error associated with the point $s_i(\mathbf{x})$ in the i th projection. The result shown in (41) does not make any specific assumptions about the surface shape or its representation. Thus, this equation could be mapped onto any set of shape parameters by inserting the derivative of a surface point with respect to those parameters. Of course one would have to compute the surface integral, and methods for solving such equations on parametric models (in the context of range data) are described in [96].

For this work we are interested in *free-form* deformations, where each point on the surface can move independently from the rest. If we let \mathbf{x}_t represent the velocity of a point on the surface, the gradient descent surface free-form surface motion is

$$\mathbf{x}_t = -\frac{dE_{\text{data}}}{d\mathbf{x}} = (\beta_0 - \beta_1) \sum_{i=1}^N e_i(\mathbf{x}) \mathbf{n}(\mathbf{x}). \quad (42)$$

Thus, at a point $\mathbf{x} \in \mathcal{S}$, the i th projection has the effect of causing the surface to expand or contract according to the difference between the projected model values and the measured data at the point $s_i(\mathbf{x})$, the projection of \mathbf{x} (Figure 19). The surface motion prescribed by a collection of projections is the sum of motions from the individual projections. In the case of continuous set of angles, the surface motion at a point is proportional to the sinusoidal line integral on the *error sinogram*, which is $e(s, \theta)$.

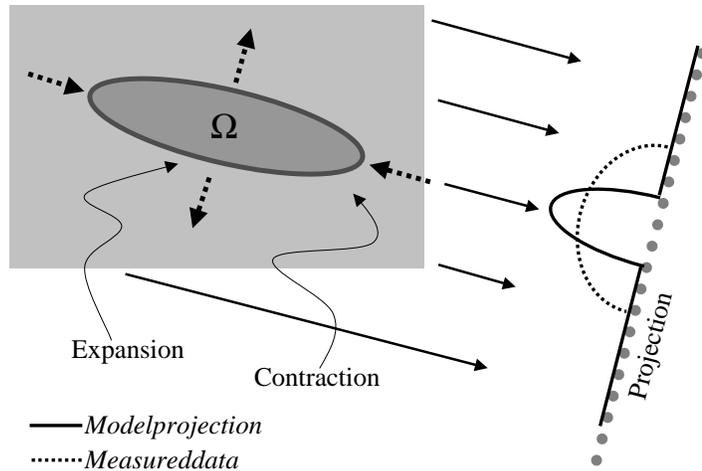


Figure 19: The model expands or contracts based on the difference in the sinograms between the projected model and the measured data.

0.6.2.1 Density Parameter Estimation

The density parameters also affect the error term in equation (37). We propose to update the estimate of the surface model iteratively, and at each iteration we re-estimate the quantities β_0 and β_1 in such a way that the energy, E_{data} is minimized. Treating Ω as fixed, (37) has two unknowns, β_0 and β_1 , which are computed from the following system:

$$\frac{\partial E_{\text{data}}}{\partial \beta_0} = 0, \quad \frac{\partial E_{\text{data}}}{\partial \beta_1} = 0. \quad (43)$$

In the case of a Gaussian noise model (43) is a linear system. Because of variations in instrumentation, the contrast levels of images taken at different angles can vary. In such cases we estimate sets of such parameters, i.e., $\beta_0(\theta_i)$ and $\beta_1(\theta_i)$ for $i = 1 \dots N$.

To extend the domain to higher dimensions, we have $\mathbf{x} \in \mathbb{R}^n$, and $S \subset \mathbb{R}^{n-1}$ and the mapping $\mathbf{s}_i : \mathbb{R}^n \mapsto S$ models the projective geometry of the imaging system (e.g. orthographic, cone beam, or fan beam). Otherwise the formulation is the same as 2D.

One important consideration is to model more complex models of density. If β_0 and β_1 are smooth, scalar functions defined over the space in which the

surface model deforms, and g is a binary function, the density model is:

$$f(\mathbf{x}) = \beta_0(\mathbf{x}) + (\beta_1(\mathbf{x}) - \beta_0(\mathbf{x}))g(x, y). \quad (44)$$

The first variation of the boundary is simply

$$\frac{d\mathbf{x}}{dt} = [\beta_1(\mathbf{x}) - \beta_0(\mathbf{x})] \sum_{i=1}^N e_i(\mathbf{x})\mathbf{n}(\mathbf{x}). \quad (45)$$

Note, this formulation is different from that of Yu and Fessler [95], who address the problem of reconstruction from noisy tomographic using using a single density function f with a smoothing term that interacts with a set of deformable edge models Γ . The edges models are surfaces, represented using level sets. In that case variational framework for deforming Γ requires differentiation of f across the edge, precisely where the proposed model exhibits (intentionally) a discontinuity.

0.6.2.2 Prior

The analysis above maximizes the likelihood. For a full MAP estimation, we include a prior term. Because we are working with the logarithm of the likelihood, the effect of the prior is additive:

$$\mathbf{x}_t = -\frac{dE_{\text{data}}}{d\mathbf{x}} - \frac{dE_{\text{prior}}}{d\mathbf{x}}. \quad (46)$$

Thus in addition to the noise model, we can incorporate some knowledge about the kinds of shapes that give rise to the measurements. With appropriately fashioned priors, we can push the solution toward desirable shapes or density values, or penalize certain shape properties, such as roughness or complexity. The choice of prior is intimately tied to the choice of surface representation and the specific application, but is independent of the formulation that describes the relationship between the estimate and the data, given in (37).

Because the data are noisy and incomplete it is useful to introduce a simple, low-level prior on the surface estimate. We therefore use a prior that penalizes surface area, which introduces a second-order smoothing term in the surface motion. That term introduces a free parameter C , which controls the relative influence of the smoothing term. The general question of how best to smooth surfaces remains an important, open question. However, if we restrict ourselves to curvature-based geometric flows, there are several reasonable options in the

literature [7, 31, 97]. The following subsection, which describes the surface representation used for our application, gives a more precise description of our smoothing methods.

0.6.3 Surface Representation and Prior

Our goal is to build an algorithm that applies to a wide range of potentially complicated shapes with arbitrary topologies—topologies that could change as the shapes deform to fit the data. For this reason, we have implemented the free-form deformation given in (42) with an implicit level set representation.

Substituting the expression for $d\mathbf{x}/dt$ (from Equations 45 and 46) into the $d\mathbf{s}/dt$ term of the level set equation (Equation 4a), and recalling that $\mathbf{n} = \nabla\phi/|\nabla\phi|$, gives

$$\frac{\partial\phi}{\partial t} = -|\nabla\phi| \left(\sum_{i=1}^M e_i(\mathbf{x}) + C\kappa(\mathbf{x}) \right), \quad (47)$$

where κ represents the effect of the prior, which is assumed to be in the normal direction.

The prior is introduced as a curvature-based smoothing on the level set surfaces. Thus, every level set moves according to a weighted combination of the principle curvatures, k_1 and k_2 , at each point. This point-wise motion is in the direction of the surface normal. For instance, the mean curvature, widely used for surface smoothing, is $H = (k_1 + k_2)/2$. Several authors have proposed using Gaussian curvature $K = k_1k_2$ or functions thereof [97]. Recently [98] have proposed using the minimum curvature, $M = \text{AbsMin}(k_1, k_2)$ for preserving thin, tubular structures, which otherwise have a tendency to *pinch off* under mean curvature smoothing.

In previous work [41], the authors have proposed a weighted sum of mean curvatures that emphasizes the minimum curvature, but incorporates a smooth transition between different surface regions, avoiding the discontinuities (in the derivative of motion) associated with a strict minimum. The *weighted curvature* is

$$W = \frac{k_1^2}{k_1^2 + k_2^2} k_2 + \frac{k_2^2}{k_1^2 + k_2^2} k_1 = \frac{2HK}{D^2}, \quad (48)$$

where $D = \sqrt{k_1^2 + k_2^2}$ is the deviation from flatness [99].

For an implicit surface, the shape matrix [100] is the derivative of the normal map projected onto the tangent plane of the surface. If we let the normal map

be $\mathbf{n} = \nabla\phi/|\nabla\phi|$, the derivative of this is the 3×3 matrix

$$N = \begin{pmatrix} \frac{\partial \mathbf{n}}{\partial x} & \frac{\partial \mathbf{n}}{\partial y} & \frac{\partial \mathbf{n}}{\partial z} \end{pmatrix}^T. \quad (49)$$

The projection of this derivative matrix onto the tangent plane gives the shape matrix $B = N(I - \mathbf{n} \otimes \mathbf{n})$, where \otimes is the exterior product and I is the 3×3 identity matrix. The eigenvalues of the matrix B are k_1, k_2 and zero, and the eigenvectors are the principle directions and the normal, respectively. Because the third eigenvalue is zero, we can compute k_1, k_2 and various differential invariants directly from the invariants of B . Thus the weighted-curvature flow is computing from B using the identities $D = \|B\|_2$, $H = \text{Tr}(B)/2$, and $K = 2H^2 - D^2/2$. The choice of numerical methods for computing B is discussed in the following section.

0.6.4 Implementation

The level set equations are solved by finite differences on a discrete grid, i.e. a volume. This raises several important issues in the implementation. These issues are the choice of numerical approximations to the PDE, efficient and accurate schemes for representing the volume, and mechanisms for computing the sinogram-based deformation in (47).

0.6.4.1 Numerical Schemes

Osher and Sethian [30] have proposed an up-wind method for solving equations of the form $\phi_t = \nabla\phi \cdot \mathbf{v}$, of which $\phi_t = |\nabla\phi| \sum_i e_i(\mathbf{x})$, from (47), is an example. The up-wind scheme utilizes one-sided derivatives in the computation of $|\nabla\phi|$, where the direction of the derivative depends, point-by-point, on the sign of the speed term $\sum_i e_i(\mathbf{x})$. With strictly regulated time steps, these scheme avoids overshooting (ringing) and instability.

Under normal circumstances, the curvature term, which is a directional diffusion, does not suffer from overshooting; it can be computed directly from first- and second-order derivatives of ϕ using central difference schemes. However, we have found that central differences does introduce instabilities when computing flows that rely on quantities other than the mean curvature. Therefore we use the method of *differences of normals* [101, 102] in lieu of central differences. The strategy is to compute normalized gradients at staggered grid

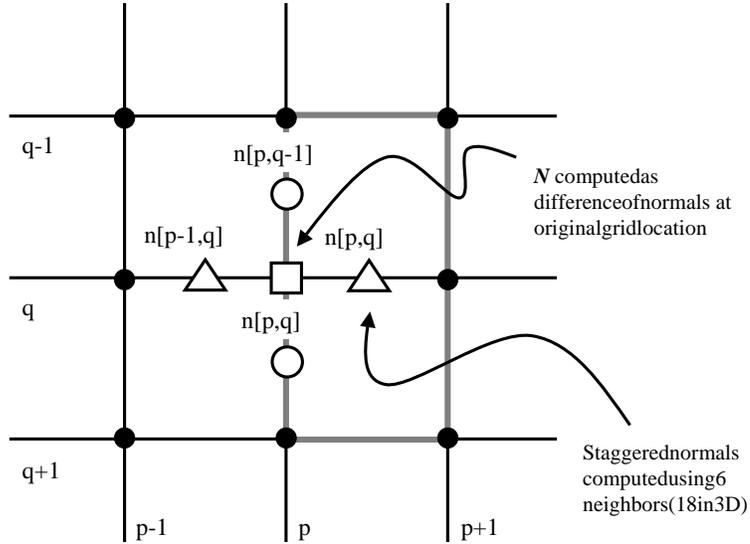


Figure 20: The shape matrix B is computed by using the differences of staggered normals.

points and take the difference of these staggered normals to get centrally located approximations to N (as in Figure 20). The normal projection operator $\mathbf{n} \otimes \mathbf{n}$ is computed with gradient estimates from central differences. The resulting curvatures are treated as speed terms (motion in the normal direction), and the associated gradient magnitude is computed using the up-wind scheme.

0.6.4.2 Sparse-Field Method

The computational burden associated with solving the 3D, second-order, non-linear level set PDE is significant. For this reason several papers [34, 35] have proposed narrow-band methods, which compute solutions only a relatively small set of pixels in the vicinity of k level set. The authors [36] have proposed a sparse-field algorithm, which uses an approximation to the distance transform and makes it feasible to recompute the neighborhood of the level set model at each time step. It computes updates on a band of grid points, called the *active set*, that is one point wide. Several layers around this active set are updated in such a way to maintain a neighborhood in order to calculate

derivatives. The position of the surface model is determined by the set of active points and their values.

0.6.4.3 Incremental Projection Updates

The tomographic surface reconstruction problem entails an additional computational burden, because the measured data must be compared to the projected model at *each iteration*. Specifically, computing \hat{p}_{ij} can be a major bottleneck. Computing this term requires re-computing the sinogram of the surface/object model as it moves. In the worst case, we would re-project the entire model every iteration.

To address this computational concern, we have developed the method of *incremental projection updates* (IPU). Rather than fully recompute \hat{p} at every iteration, we maintain a current running version of \hat{p} and update it to reflect the changes in the model as it deforms. Changes in the model are computed only on a small set of grid points in the volume, and therefore the update time is proportional to the area of the surface, rather than the size of the volume it encloses.

The IPU strategy works with the the sparse-field algorithm as follows. At each iteration, the sparse-field algorithm updates only the active layer (one voxel wide) and modifies the set of active grid points as the surface moves. The incremental projection update strategy takes advantage of this to selectively update the model projection to reflect those changes. At each iteration, the amount of change in an active point's value determines the motion of that particular surface point as well as the percentage of the surrounding voxel that is either inside or outside of the surface. By the linearity of projection, we can map these changes in the object shape, computed at grid points along the surface boundary, back into the sinogram space and thereby *incrementally* update the sinogram. Notice that each 3D grid point has a weighting coefficient (these are precomputed and fixed), which is determined by its geometric mapping of the surrounding voxel back into the sinogram, as in Figure 21. In this way the IPU method maintains sub-voxel accuracy at a relatively low computational cost.

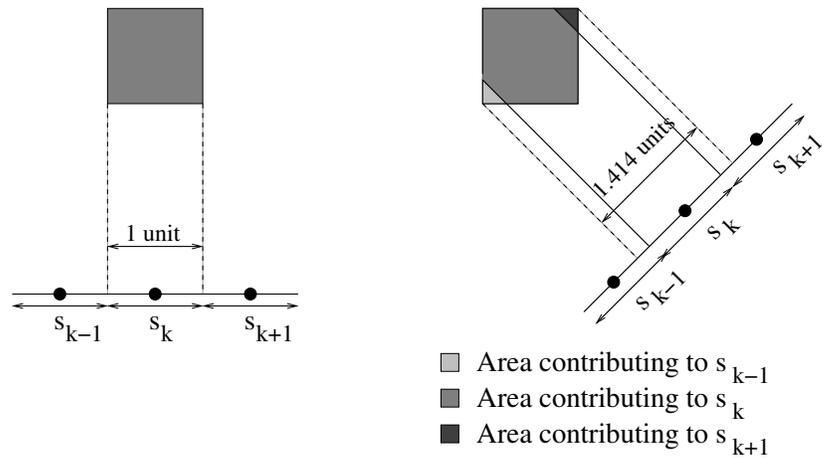


Figure 21: A weighting coefficient for each voxel determines the portions of the discrete sinogram influenced by incremental changes to a grid point.

0.6.4.4 Initialization

The deformable model fitting approach requires an initial model, i.e. $\phi(\mathbf{x}, t = 0)$. This initial model should be obtained using the “best” information available prior to the surface fitting. In some cases this will mean thresholding a greyscale reconstruction, such as FBP, knowing that it has artifacts. In practice the initial surface estimate is impacted by the reconstruction method and the choice of threshold, and because we perform a local minimization, these choices can affect the final result. Fortunately, the proposed formulation is moderately robust with respect to the initial model, and our results show that the method works well under a range of reasonable initialization strategies.

0.6.5 Results

0.6.5.1 Transmission Electron Microscopy

Transmission electron microscopy is the process of using transmission images of electron beams to reveal biological structures on very small dimensions. Typically transmission electron microscopy (TEM) datasets are produced using a dye that highlights regions of interest, e.g. the interior of a microscopic structure, such as a cell (see Figure 22a). There are technical limits to the projection angles from which data can be measured. These limits are due to

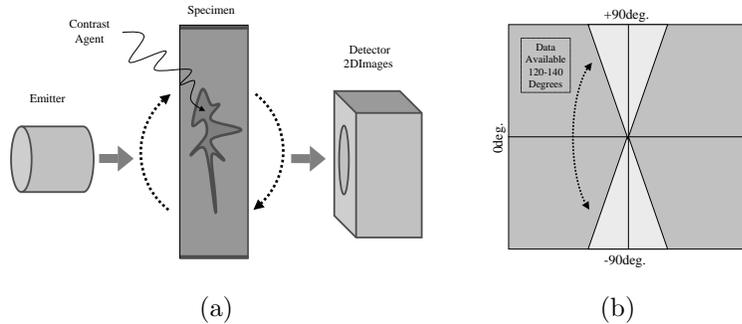


Figure 22: a) Transmission electron microscopy is used to image very small specimens that have been set apart from the substrate by a contrast agent. b) TEM imaging technology provides projections over a limited set of angles.

the mechanical apparatus used to tilt the specimens and the trade off between the destructive effects of electron energy and the effective specimen thickness, which increases with tilt angle. Usually, the maximum tilt angle is restricted to about ± 60 – 70 degrees. Figure 22b shows an illustration of the geometry of this limited-angle scenario. The TEM reconstruction problem is further aggravated by the degree of electron scattering, which results in projection images (sinograms) that are noisy relative to many other modalities, e.g. X-ray CT. Finally, due to the flexible nature of biological objects and the imperfections in the tilting mechanism, the objects undergo some movements while being tilted. Manual alignment procedures used to account for this tend to produce small misregistration errors.

We applied the proposed algorithm to 3D TEM data obtained from a 3 MeV electron microscope. This 3D dataset consists of 67 tilt series images, each corresponding to one view of the projection. Each tilt series image is of size 424×334 . The volume reconstructed by FBP is of size $424 \times 424 \times 334$. Figure 23a and 23b show the sinogram corresponding to a single slice of this dataset and the estimate of the same sinogram created by the method. Figure 23e shows the surface estimate intersecting this slice overlaid on the back projected slice. Some structures not seen in the back projection are introduced in the final estimation, but the orientation of the structures introduced suggests that these are valid features that were lost due to reconstruction artifacts from the FBP. Also, the proposed method captures line-by-line brightness variations in

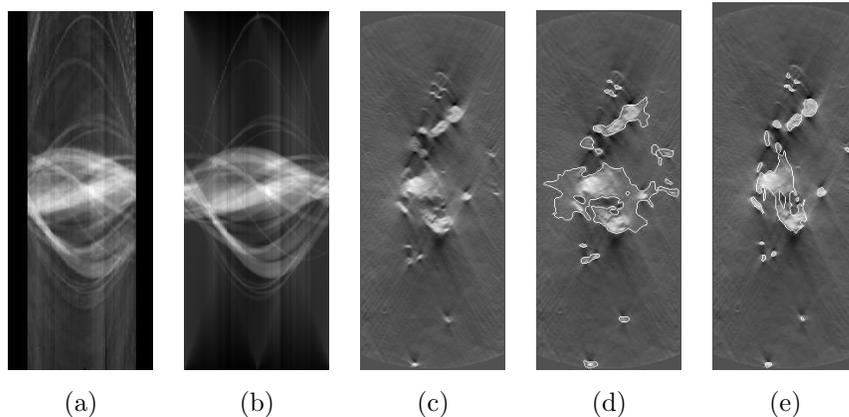


Figure 23: 2D slice of dendrite data: (a) Sinogram of one slice (b) Sinogram estimated by the proposed method (c) Back projection showing artifacts (d) Initial model obtained by thresholding the back projection (white curve overlaid on the back projection) (e) Final surface estimate.

the input sinogram (as explained in section 0.6.2.1). This suggests that the density estimation procedure is correct.

Figure 24 shows the 3D initialization and the final 3D surface estimate. The figure also shows enlarged initial and final versions of a small section of the surface. Computing the surface estimate for the TEM dendrite with 150 iterations took approximately 3 hours on a single 300MHz processor of a Silicon Graphics Onyx2 workstation. We consider these results positive for several reasons. First, the biology is such that one expects the spines (small protrusions) to be connected to the dendrite body. The proposed method clearly establishes those connections, based solely on consistency of the model with the projected data. The second piece of evidence is the shapes of the spines themselves. The reconstructed model shows the recurrence of a characteristic shape—a long thin spine with a cup-like structure on the end. This characteristic structure, which often fails to show up in the FBP reconstruction, does appear quite regularly in hand-segmentations of the same datasets.

0.6.5.2 Sinogram Extrapolation

The fitting of surfaces to this data is a simplification. It is justified in the context of segmentation, but there *are* underlying inhomogeneities in the density of

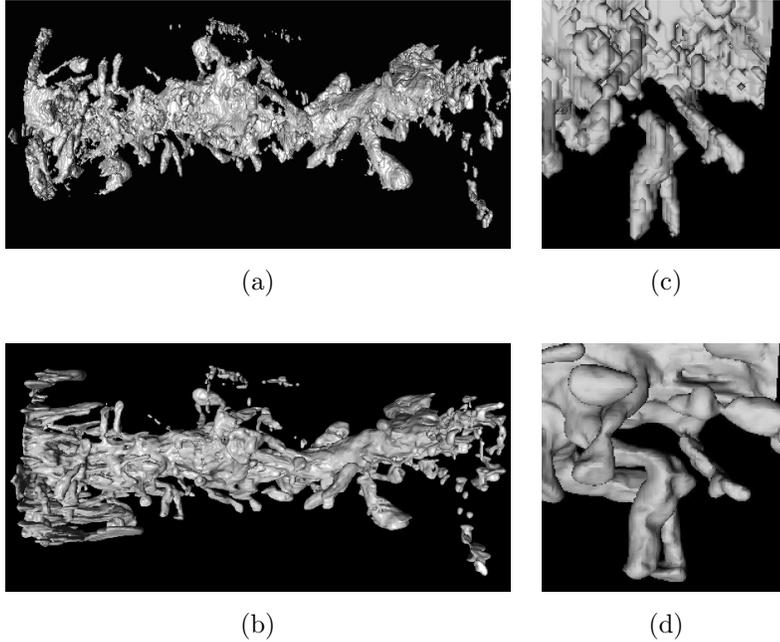


Figure 24: 3D results: (a) Surface initialization (b) Final surface estimated after 150 iterations (c) A portion of the initial surface enlarged (d) The corresponding portion in the final surface

this specimen, which could be indicative of relevant structures. Thus for some applications *direct* visualization of the measured data, by volume rendering, offers advantages over the segmented surfaces. We propose to use the surface estimation algorithm as a mechanism for estimating the missing data in the sinograms.

Figure 25a and 25b show the input sinogram and the sinogram of the estimated model (for one slice) of the TEM dendrite data. The estimated sinogram demonstrates that the surface estimation method recovers the missing information in a reasonable way. Thus, we combine the sinograms from the model with original sinograms to produce a “full” sinogram that still contains *all* of the original, measured data. FBP reconstructions from such augmented sinograms should have fewer limited-angle streak artifacts.

We demonstrate this by comparing volume renderings with and without the augmentation. We create augmented sinograms by using sinogram data from the estimated model only where the data is missing from the measured

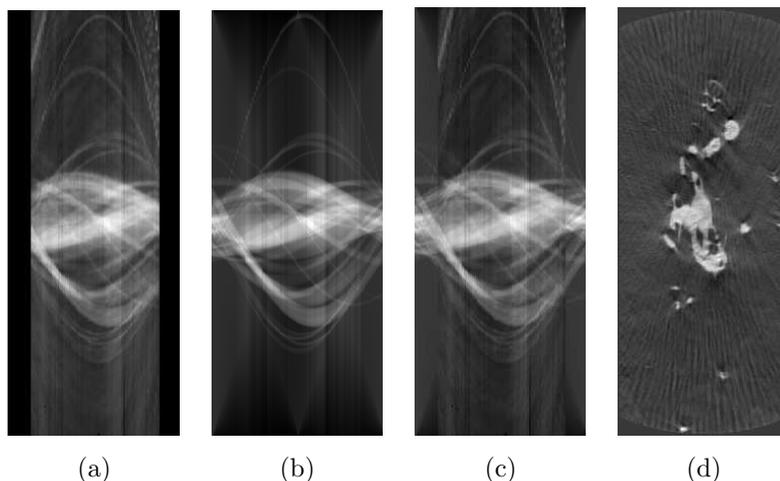


Figure 25: Sinogram extrapolation for slice number 150 of dendrite data: (a) Input sinogram (b) Sinogram estimated by the proposed method (c) Augmented sinogram constructed using original data and estimating missing data from the segmentation (d) FBP reconstruction of the augmented sinogram.

sinograms. The augmented sinogram for a single slice is shown in Figure 25c. The slice reconstructed (FBP) from the augmented sinogram is shown in Figure 25d. Note that this reconstructed slice does not contain the limited-angle artifacts that appear in the slice in Figure 23c. Maximum intensity projection (MIP) volume renderings of the volume created from original sinograms and the volume created from augmented sinograms are compared in Figure 26. The main body of the dendrite, which exhibited a very convoluted and fuzzy boundary, shows better definition. Also, several of the spines which were dangling in the original reconstruction are now connected.

Conclusions

This chapter has described a level set segmentation framework and the pre-processing and data analysis techniques needed for a number of segmentation applications. Several standard volume processing algorithms have been incorporated into the framework in order to segment datasets generated from MRI, CT and TEM scans. A technique based on moving least-squares has been developed for segmenting multiple non-uniform scans of a single object. New

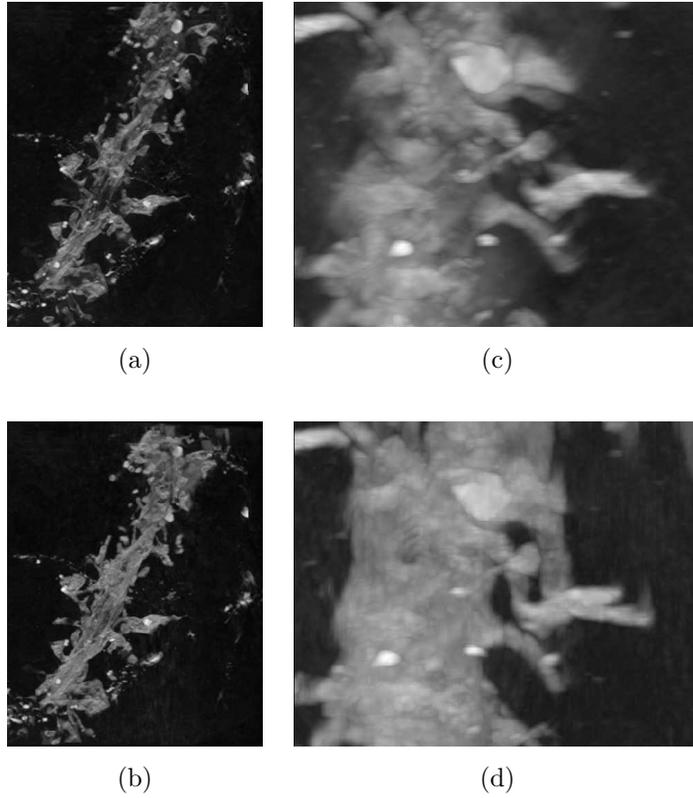


Figure 26: Sinogram extrapolation results: (a) MIP volume rendering of volume reconstructed from original sinograms (b) MIP volume rendering of volume reconstructed from augmented (extrapolated) sinograms (c) A portion of original MIP enlarged (d) The corresponding portion in augmented MIP enlarged

scalar measures have been defined for extracting structures from diffusion tensor MRI scans. Finally, a direct approach to the segmentation of incomplete tomographic data using density parameter estimation is described. These techniques, combined with level set surface deformations, allow us to segment many different types of biological volume datasets.

Acknowledgements

Several people provided valuable technical assistance and support to our work. They are Dr. Alan Barr, Dr. Jason Wood, Dr. John Wood, Dr. Cyrus Papan, Dr. Russ Jacobs, Dr. Scott Fraser, Dr. J. Michael Tyszka, Dr. Miriam Scadeng, Dr. David Dubowitz, Dr. Eric Ahrens, Dr. Mark Ellisman, Dr. Maryanne Martone, Dr. Chris Johnson and Dr. Mark Bastin. Datasets were provided by Caltech Biological Imaging Center (e.g. Figure 8a), National Center for Microscopy and Imaging Research (e.g. Figure 7, funded by NIH grant #P41-RR04050), Caltech Multi-Res Modeling Group (Figure 9 (Top)), Stanford Computer Graphics Laboratory (Figure 9 (Top)), Childrens Hospital - Los Angeles (Figure 10), University of Utah's SCI Institute (e.g. Figure 14), and the University of Edinburgh, UK (Figure 16).

This work was supported by National Science Foundation grants #ASC-89-20219, #ACI-9982273, #ACI-0083287 and #ACI-0089915, the Office of Naval Research Volume Visualization grant #N00014-97-0227, the National Institute on Drug Abuse and the National Institute of Mental Health, as part of the Human Brain Project, the National Library of Medicine "Insight" Project #N01-LM-0-3503, and the Caltech SURF Program.

Bibliography

- [1] Drebin, R., Carpenter, L., and Hanrahan, P., Volume Rendering, In: Proceedings SIGGRAPH 88 Conference, pp. 65–74, 1988.
- [2] Levoy, M., Display of Surfaces from Volume Data, IEEE Computer Graphics and Applications, Vol. 9, no. 3, pp. 245–261, 1990.
- [3] Laur, D. and Hanrahan, P., Hierarchical splatting: A progressive refinement algorithm for volume rendering, In: SIGGRAPH '91 Proceedings, Sederberg, T. W., ed., pp. 285–288, July 1991.
- [4] Parker, S., Parker, M., Livnat, Y., Sloan, P., Hansen, C., and Shirley, P., Interactive Ray Tracing for Volume Visualization, IEEE Transactions on Visualization and Computer Graphics, Vol. 5, no. 3, pp. 238–250, 1999.
- [5] Leventon, M., Faugeras, O., Grimson, W., and W. Wells III, Level Set Based Segmentation with Intensity and Curvature Priors, In: Workshop on Mathematical Methods in Biomedical Image Analysis Proceedings, pp. 4–11, June 2000.
- [6] Malladi, R., Sethian, J., and Vemuri, B., Shape Modeling with Front Propagation: A Level Set Approach, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 17, no. 2, pp. 158–175, 1995.
- [7] Sethian, J., Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, UK, second ed., 1999.
- [8] Staib, L., Zeng, X., Schultz, R., and Duncan, J., Shape Constraints in Deformable Models, In: Handbook of Medical Imaging, Bankman, I., ed., Chapter 9, pp. 147–157, Academic Press, 2000.
- [9] Wu, Z., Chung, H.-W., and Wehrli, F. W., A Bayesian Approach to Subvoxel Tissue Classification in NMR Microscopic Images of Trabecular Bone, Journal of Computer Assisted Tomography, Vol. 12, no. 1, pp. 1–9, 1988.

- [10] Kao, Y.-H., Sorenson, J. A., and Winkler, S. S., MR Image Segmentation Using Vector Decomposition and Probability Techniques: A General Model and Its Application to Dual-Echo images, *Magnetic Resonance in Medicine*, Vol. 35, pp. 114–125, 1996.
- [11] Cline, H. E., Lorensen, W. E., Kikinis, R., and Jolesz, F., Three-Dimensional Segmentation of MR Images of the Head Using Probability and Connectivity, *Journal of Computer Assisted Tomography*, Vol. 14, no. 6, pp. 1037–1045, Nov., Dec. 1990.
- [12] Laidlaw, D. H., Fleischer, K. W., and Barr, A. H., Partial-Volume Bayesian Classification of Material Mixtures in MR Volume Data using Voxel Histograms, *IEEE Transactions on Medical Imaging*, Vol. 17, no. 1, pp. 74–86, feb 1998.
- [13] Johnson, V. E., A Framework for Incorporating Structural Prior Information into the Estimation of Medical Images, In: *Information Processing in Medical Imaging (IPMI'93)*, Barrett, H. H. and Gmitro, A. F., eds., no. 687 In *Lecture Notes in Computer Science*, pp. 307–321, Springer-Verlag, 1993.
- [14] Marr, D. and Hildreth, E., Theory of Edge Detection, *Proceedings of the Royal Society of London*, Vol. B, no. 207, pp. 187–217, 1980.
- [15] Marr, D., *Vision*, Freeman, San Francisco, 1982.
- [16] Canny, J., A computational approach to edge detection, *IEEE Trans. on Pat. Anal. and Mach. Intel.*, Vol. 8, no. 6, pp. 679–698, 1986.
- [17] Cootes, T., Hill, A., Taylor, C., and Haslam, J., The Use of Active Shape Models For Locating Structures in Medical Images, In: *Information Processing in Medical Imaging (IPMI'93)*, Barrett, H. H. and Gmitro, A. F., eds., no. 687 In *Lecture Notes in Computer Science*, pp. 33–47, Springer-Verlag, 1993.
- [18] Stetten, G. and Pizer, S., Medial Node Models to Identify and Measure Objects in Real-Time 3D Echocardiography, *IEEE Transactions on Medical Imaging*, Vol. 18, no. 10, pp. 1025–1034, 1999.

- [19] Wood, Z., Desbrun, M., Schröder, P., and Breen, D., Semi-Regular Mesh Extraction from Volumes, In: Proc. IEEE Visualization 2000, pp. 275–282, 2000.
- [20] Miller, J., Breen, D., Lorensen, W., O’Bara, R., and Wozny, M., Geometrically Deformed Models: A Method for Extracting Closed Geometric Models from Volume Data, In: Proc. SIGGRAPH ’91, pp. 217–226, July 1991.
- [21] Pentland, A. P., Perceptual organization and the representation of natural form, *Artificial Intelligence*, Vol. 28, pp. 293–331, 1986.
- [22] Terzopoulos, D. and Metaxas, D., Dynamic 3D models with local and global deformations: Deformable superquadrics, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, no. 7, pp. 703–714, 1991.
- [23] Gupta, A. and Bajcsy, R., Volumetric Segmentation of Range Images of 3D Objects Using Superquadric Models, *CVGIP: Image Understanding*, Vol. 58, no. 3, pp. 302–326, 1993.
- [24] Muraki, S., Volumetric shape description of range data using “Blobby Model”, In: *SIGGRAPH ’91 Proceedings*, Sederberg, T. W., ed., pp. 227–235, Jul. 1991.
- [25] Szeliski, R., Tonnesen, D., and Terzopoulos, D., Modeling Surfaces of Arbitrary Topology with Dynamic Particles, In: *Proc. Fourth Int. Conf. on Comp. Vision (ICCV’93)*, pp. 82–87, IEEE Computer Society Press, Berlin, Germany, May 1993.
- [26] McInerney, T. and Terzopoulos, D., A Dynamic Finite Element Surface Model for Segmentation and Tracking in Multidimensional Medical Images with Application to Cardiac 4D Image Analysis, *Computerized Medical Imaging and Graphics*, Vol. 19, no. 1, pp. 69–83, 1995.
- [27] Park, J., Metaxas, D., Young, A. A., and Axel, L., Deformable Models with Parameter Functions for Cardiac Motion Analysis from Tagged MRI Data, *IEEE Transactions on Medical Imaging*, Vol. 15, no. 3, pp. 278–289, June 1996.

- [28] DeCarlo, D. and Metaxas, D., Shape Evolution with Structural and Topological Changes using Blending, *IEEE Trans. on Pat. Anal. and Mach. Intel.*, Vol. 20, no. 11, pp. 1186–1205, November 1998.
- [29] Ramamoorthi, R. and Arvo, J., Creating Generative Models from Range Images, In: *SIGGRAPH '99 Proceedings*, pp. 195–204, August 1999.
- [30] Osher, S. and Sethian, J., Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *Journal of Computational Physics*, Vol. 79, pp. 12–49, 1988.
- [31] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, Springer, Berlin, 2002.
- [32] Sethian, J., A Fast Marching Level Set Method for Monotonically Advancing Fronts, In: *Proceedings of the National Academy of Science*, Vol. 93 of 4, pp. 1591–1595, 1996.
- [33] Tsitsiklis, J., Efficient Algorithms for Globally Optimal Trajectories, *IEEE Trans. on Automatic Control*, Vol. 40, no. 9, pp. 1528–1538, 1995.
- [34] Adalsteinsson, D. and Sethian, J. A., A Fast Level Set Method for Propagating Interfaces, *Journal of Computational Physics*, Vol. 118, no. 2, pp. 269–277, may 1995.
- [35] Peng, D., Merriman, B., Osher, S., Zhao, H.-K., and Kang, M., A PDE-Based Fast Local Level Set Method, *Journal of Computational Physics*, Vol. 155, pp. 410–438, 1999.
- [36] Whitaker, R., A Level-Set Approach to 3D Reconstruction From Range Data, *International Journal of Computer Vision*, Vol. 29, no. 3, pp. 203–231, 1998.
- [37] Whitaker, R., Breen, D., Museth, K., and Soni, N., Segmentation of Biological Datasets Using a Level-Set Framework, In: *Volume Graphics 2001*, Chen, M. and Kaufman, A., eds., pp. 249–263, Springer, Vienna, 2001.
- [38] van den Boomgaard, R. and Smeulders, A. W. M., The morphological structure of images, the differential equations of morphological scale-space, *IEEE Trans. on Pat. Anal. and Mach. Intel.*, Vol. 16, no. 11, pp. 1101–1113, 1994.

- [39] Maragos, P., Differential Morphology and Image Processing, *IEEE Trans. on Image Processing*, Vol. 5, no. 6, pp. 922–937, June 1996.
- [40] Requicha, A. and Voelcker, H., Boolean Operations in Solid Modeling: Boundary Evaluation and Merging Algorithms, *Proceedings of the IEEE*, Vol. 73, no. 1, pp. 30–44, 1985.
- [41] Whitaker, R. T., Volumetric Deformable Models: Active Blobs, In: *Visualization In Biomedical Computing*, Robb, R. A., ed., pp. 122–134, SPIE, Mayo Clinic, Rochester, Minnesota, 1994.
- [42] Sapiro, G., *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, Cambridge, UK, 2001.
- [43] Museth, K., Breen, D., Zhukov, L., and Whitaker, R., Level Set Segmentation from Multiple Non-Uniform Volume Datasets, In: *Proc. IEEE Visualization 2002*, pp. 179–186, October 2002.
- [44] Shepard, D., A two-dimensional interpolation function for irregularly spaced points, In: *Proc. ACM Nat. Conf.*, pp. 517–524, 1968.
- [45] Lancaster, P. and Salkauskas, K., Surfaces generated by moving least squares methods, *Math. Comp.*, Vol. 37, pp. 141–159, 1981.
- [46] Farwig, R., Multivariate interpolation of arbitrarily spaced data by moving least-squares methods, *J. of Comp. and App. Math.*, Vol. 16, pp. 79–93, 1986.
- [47] Zhao, H.-K., Osher, S., and Fedkiw, R., Fast Surface Reconstruction using the Level Set Method, In: *Proc. 1st IEEE Workshop on Variational and Level Set Methods*, pp. 194–202, 2001.
- [48] Turk, G. and Levoy, M., Zippered Polygon Meshes from Range Images, In: *Proc. of SIGGRAPH '94*, pp. 311–318, ACM SIGGRAPH, August 1994.
- [49] Curless, B. and Levoy, M., A Volumetric Method for Building Complex Models from Range Images, In: *Proc. SIGGRAPH '96*, pp. 303–312, 1996.

- [50] Tamez-Pena, J., Totterman, S., and Parker, K., MRI Isotropic Resolution Reconstruction from Two Orthogonal Scans, In: Proc. SPIE Medical Imaging, Vol. 4322, pp. 87–97, 2001.
- [51] Goshtasby, A. and Turner, D. A., Fusion of Short-Axis and Long-Axis Cardiac MR Images, In: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 202–211, San Francisco, June 1996.
- [52] Brejl, M. and Sonka, M., Directional 3D Edge Detection in Anisotropic Data: Detector Design and Performance Assessment, Computer Vision and Image Understanding, Vol. 77, pp. 84–110, 2000.
- [53] Haralick, R. M. and Shapiro, L. G., Computer and Robot Vision, Addison-Wesley, 1991.
- [54] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W., Numerical Recipes in C (2nd edition), Cambridge University Press, New York, NY, 1992.
- [55] Basser, P. J., Mattiello, J., and Bihan, D. L., Estimation of the effective self-diffusion tensor from the NMR spin echo, Journal of Magnetic Resonance, Series B, Vol. 103, no. 3, pp. 247–254, 1994.
- [56] Basser, P. J., Mattiello, J., and Bihan, D. L., MR diffusion tensor spectroscopy and imaging, Biophysical Journal, Vol. 66, no. 1, pp. 259–267, 1994.
- [57] Basser, P. J. and Pierpaoli, C., Microstructural and Physiological Features of Tissues Elucidated by Quantitative-Diffusion-Tensor MRI, Journal of Magnetic Resonance, Series B, Vol. 111, no. 3, pp. 209–219, 1996.
- [58] Westin, C.-F., Peled, S., Gudbjartsson, H., Kikinis, R., and Jolesz, F. A., Geometrical Diffusion Measures for MRI from Tensor Basis Analysis, In: Proceedings ISMRM 5th Annual Meeting, p. 1742, 1997.
- [59] Peled, S., Gudbjartsson, H., Westin, C., Kikinis, R., and Jolesz, F., Magnetic Resonance Imaging Shows Orientation and Assymetry in White Matter Fiber Tracts, Brain Research, Vol. 780, pp. 27–33, 1998.
- [60] Basser, P. and Pajevic, S., Statistical Artifacts in Diffusion Tensor MRI Caused by Background Noise, Magnetic Resonance in Medicine, Vol. 44, pp. 41–50, 2000.

- [61] Ulug, A. and van Zijl, P., Orientation-independent diffusion imaging without tensor diagonalization: anisotropy definitions based on physical attributes of the diffusion ellipsoid, *Journal of Magnetic Resonance Imaging*, Vol. 9, pp. 804–813, 1999.
- [62] Laidlaw, D., Ahrens, E., Kremers, D., Avalos, M., Jacobs, R., and Readhead, C., Visualizing Diffusion Tensor Images of the Mouse Spinal Cord, In: *Proc. IEEE Visualization '98*, pp. 127–134, 1998.
- [63] Kindlmann, G. and Weinstein, D., Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields, In: *Proc. IEEE Visualization '99*, pp. 183–189, 1999.
- [64] Zhukov, L., , Museth, K., Breen, D., Whitaker, R., and Barr, A., Level Set Modeling and Segmentation of DT-MRI Brain Data, *Journal of Electronic Imaging*, Vol. 12, no. 1, pp. 125–133, January 2003.
- [65] Basser, P., Pajevic, S., Pierpaoli, C., Duda, J., and Aldroubi, A., In Vivo Fiber Tractography Using DT-MRI Data, *Magnetic Resonance in Medicine*, Vol. 44, pp. 625–632, 2000.
- [66] Poupon, C., Clark, C., Frouin, V., Regis, J., Bloch, I., Bihan, D. L., and Mangin, J.-F., Regularization of Diffusion-Based Direction Maps for the Tracking of Brain White Matter Fascicles, *Neuroimage*, Vol. 12, pp. 184–195, 2000.
- [67] Singh, A., Goldgof, D., and Terzopoulos, D., eds., *Deformable Models in Medical Image Analysis*, IEEE Computer Society Press, Los Alamitos, CA, 1998.
- [68] Kindlmann, G. and Durkin, J., Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering, In: *Proc. IEEE Symposium on Volume Visualization*, pp. 79–86, 1998.
- [69] Zhukov, L., Weinstein, D., and Johnson, C., Independent Component Analysis for EEG Source Localization in Realistic Head Model, *IEEE Engineering in Medicine and Biology*, Vol. 19, pp. 87–96, 2000.
- [70] Gibson, S. et al., Volumetric Object Modeling for Surgical Simulation, *Medical Image Analysis*, Vol. 2, no. 2, pp. 121–132, 1998.

- [71] Bailey, M., Manufacturing Isovolumes, In: Volume Graphics, Chen, M., Kaufman, A., and Yagel, R., eds., pp. 79–83, Springer Verlag, London, 2000.
- [72] Lorensen, W. and Cline, H., Marching Cubes: A High Resolution 3D Surface Construction Algorithm, In: Proc. SIGGRAPH '87, pp. 163–169, Jul. 1987.
- [73] Ramm, A. G. and Katsevich, A. I., The Radon Transform and Local Tomography, CRC Press, Inc., 2000, Corporate Blvd., N.W., Boca Raton, Florida 33431, 1996.
- [74] Elangovan, V. and Whitaker, R., From Sinograms to Surfaces: A Direct Approach to the Segmentation of Tomographic Data, In: Proc. MIC-CAI 2001, Vol. 2208 of Lecture Notes in Computer Science, pp. 213–223, Springer, 2001.
- [75] Herman, G. T., Image Reconstruction From Projections, The Fundamentals of Computerized Tomography, Academic Press, New York, 1980.
- [76] Roerdink, J. B. T. M., Computerized tomography and its applications: a guided tour, Nieuw Archief voor Wiskunde, Vol. 10, no. 3, pp. 277–308, nov 1992.
- [77] Wang, G., Vannier, M., and Cheng, P., Iterative X-ray cone-beam tomography for metal artifact reduction and local region reconstruction, Microscopy and Microanalysis, Vol. 5, pp. 58–65, 1999.
- [78] Inouye, T., Image Reconstruction with Limited Angle Projection Data, IEEE Transactions on Nuclear Science, Vol. NS-26, pp. 2666–2684, 1979.
- [79] Prince, J. L. and Willsky, A. S., Hierarchical Reconstruction Using Geometry and Sinogram Restoration, IEEE Transactions on Image Processing, Vol. 2, no. 3, pp. 401–416, jul 1993.
- [80] Herman, G. T. and Kuba, A., eds., Discrete Tomography: Foundations, Algorithms, and Applications, Birkhauser, Boston, 1999.
- [81] Thirion, J. P., Segmentation of Tomographic Data without Image Reconstruction, IEEE Transactions on Medical Imaging, Vol. 11, pp. 102–110, mar 1992.

- [82] Sullivan, S., Noble, A., and Ponce, J., On Reconstructing Curved Object Boundaries from Sets of X-Ray Images, In: Proceedings of the 1995 Conference on Computer Vision, Virtual Reality, and Robotics in Medicine, Ayache, N., ed., Lecture Notes in Computer Science 905, pp. 385–391, Springer-Verlag, 1995.
- [83] Hanson, K., Cunningham, G., Jr., G. J., and Wolf, D., Tomographic reconstruction based on flexible geometric models, In: IEEE Int. Conf. on Image Processing (ICIP 94), pp. 145–147, 1994.
- [84] Battle, X. L., Cunningham, G. S., and Hanson, K. M., 3D tomographic reconstruction using geometrical models, In: Medical Imaging: Image Processing, Hanson, K. M., ed., Vol. 3034, pp. 346–357, SPIE, 1997.
- [85] Battle, X. L., Bizais, Y. J., Rest, C. L., and Turzo, A., Tomographic reconstruction using free-form deformation models, In: Medical Imaging: Image Processing, Hanson, K. M., ed., Vol. 3661, pp. 356–367, SPIE, 1999.
- [86] Battle, X. L., LeRest, C., Turzo, A., and Bizais, Y., Three-Dimensional Attenuation Map Reconstruction Using Geometrical Models and Free-Form Deformations, IEEE Transactions on Medical Imaging, Vol. 19, no. 5, pp. 404–411, 2000.
- [87] Mohammad-Djafari, A., Sauer, K., Khayi, Y., and Cano, E., Reconstruction of the shape of a compact object from a few number of projections, In: IEEE International Conference on Image Processing (ICIP), Vol. 1, pp. 165–169, 1997.
- [88] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic Active Contours, In: Fifth Int. Conf. on Comp. Vision, pp. 694–699, IEEE, IEEE Computer Society Press, 1995.
- [89] Santosa, F., A level-set approach for inverse problems involving obstacles, European Series in Applied and Industrial Mathematics: Control Optimization and Calculus of Variations, Vol. 1, pp. 17–33, jan 1996.
- [90] Dorn, O., Miller, E. L., and Rappaport, C., A shape reconstruction method for electromagnetic tomography using adjoint fields and level

- sets, Inverse problems: Special issue on Electromagnetic Imaging and Inversion of the Earth's Subsurface (Invited Paper), Vol. 16, pp. 1119–1156, oct 2000.
- [91] Dorn, O., Miller, E. L., and Rappaport, C., Shape reconstruction in 2D from limited-view multi-frequency electromagnetic data, *Contemporary mathematics*, to appear 2001.
- [92] Chan, T. F. and Vese, L. A., A Level Set Algorithm for Minimizing the Mumford-Shah Functional in Image Processing, Tech. Rep. CAM 00-13, UCLA, Department of Mathematics, 2000.
- [93] Tsai, A., Yezzi, A., and Willsky, A., A Curve Evolution Approach to Smoothing and Segmentation Using the Mumford-Shah Functional, In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 119–124, 2000.
- [94] Debreuve, E., Barlaud, M., Aubert, G., and Darcourt, J., Attenuation Map Segmentation Without Reconstruction Using a Level Set Method in Nuclear Medicine Imaging, In: *IEEE International Conference on Image Processing (ICIP)*, Vol. 1, pp. 34–38, 1998.
- [95] Yu, D. and Fessler, J., Edge-preserving tomographic reconstruction with nonlocal regularization, In: *Proceedings of IEEE Intl. Conf. on Image Processing*, pp. 29–33, 1998.
- [96] Whitaker, R. and Gregor, J., A Maximum Likelihood Surface Estimator For Dense Range Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, no. 10, pp. 1372–1387, October 2002.
- [97] Sapiro, G., *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, 2001.
- [98] Lorigo, L., Faugeras, O., Grimson, E., Keriven, R., Kikinis, R., Nabavi, A., and Westin, C.-F., Co-Dimension 2 Geodesic Active Contours for the Segmentation of Tubular Structures, In: *Proceedings of IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 444–452, June 2000.
- [99] Koenderink, J. J., *Solid Shape*, MIT press, Cambridge, Mass, 1990.

- [100] do Carmo, M., *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, 1976.
- [101] Rudin, L., Osher, S., and Fatemi, C., Nonlinear total variation based noise removal algorithms, *Physica D*, Vol. 60, pp. 259–268, 1992.
- [102] Whitaker, R. and Xue, X., Variable-Conductance, Level-Set Curvature for Image Denoising, In: *Proc. IEEE International Conference on Image Processing*, pp. 142–145, October 2001.