An Eulerian PDE Approach for Computing Tissue Thickness

Anthony J. Yezzi, Jr., *Member, IEEE*, and Jerry L. Prince*, *Senior Member, IEEE*

***Abstract*—We outline an Eulerian framework for computing the thickness of tissues between two simply connected boundaries that does not require landmark points or parameterizations of either boundary. Thickness is defined as the length of *correspondence trajectories*, which run from one tissue boundary to the other, and which follow a smooth vector field constructed in the region between the boundaries. A pair of partial differential equations (PDEs) that are guided by this vector field are then solved over this region, and the sum of their solutions yields the thickness of the tissue region. Unlike other approaches, this approach does not require explicit construction of any correspondence trajectories. An efficient, stable, and computationally fast solution to these PDEs is found by careful selection of finite differences according to an upwinding condition. The behavior and performance of our method is demonstrated on two simulations and two magnetic resonance imaging data sets in two and three dimensions. These experiments reveal very good performance and show strong potential for application in tissue thickness visualization and quantification.**

***Index Terms*—Correspondence trajectory, numerical methods, partial differential equations (PDEs), thickness.**

I. Introduction

MEASURING the thickness of anatomical objects is an important objective in medical image analysis for several reasons. For one, the thickness of a particular structure might provide an indication of its functional performance. For example, myocardial thickening during systole is an important indicator of healthy cardiac function [1]. Thickness can also provide an indication of disease. For example, thinning of the gray matter in the brain cortex is thought to be associated with Alzheimer's disease and other neurodegenerative disorders [2]. Thickness might also prove to be the basis for image segmentation. For example, it is well known that the anterior and posterior banks of the central sulcus in the human brain cortex can be distinguished by a difference in thickness alone [3]. Finally, thickness can be used as a basis for efficient characterization of anatomical shape when coupled with a central axis representation [4].

Manuscript received May 24, 2003; revised January 30, 2003. This work was supported in part by the National Institutes of Health (NIH)/National Institute of Neurological Disorders and Stroke (NINDS) under Grant R01-NS37747, in part by the NIH/National Heart, Lung, and Blood Institute (NHLBI) under Grant R01-HL68904, and in part by the National Science Foundation (NSF) under Grant CAREER-CCR0133736. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was J. Duncan. *Asterisk indicates corresponding author.*

A. J. Yezzi, Jr. is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30322 USA (e-mail: ayezzi@ece.gatech.edu).

*J. L. Prince is with the Department of Electrical and Computer Engineering, The Johns Hopkins University, 105 Barton Hall/3400 N. Charles Street, Baltimore, MD 21218 USA (e-mail: prince@jhu.edu).

Digital Object Identifier 10.1109/TMI.2003.817775

In this paper, we present a method for computing the thickness between two surfaces (or curves) that do not have point correspondences defined between them. The method we describe is based on the definition of thickness as the length of *correspondence trajectories* (curved, in general), which run from one surface to the other. While conceptually analogous to the thickness definition used in [5], our definition is more general and our computational approach is fast and stable. Furthermore, the computational procedure outlined in this paper is purely Eulerian in nature, using only the structure of the fixed Cartesian grid. In particular, in our approach, there is no need to explicitly construct or trace any of the correspondence trajectories that form the basis for this notion of thickness.

The thickness computation relies upon the solution of a pair of partial differential equations (PDEs) defined on the tissue region. The sum of their solutions yields thickness values not only along the tissue boundaries, but within the entire tissue region as well. The numerical solutions of these PDEs utilize finite differences, which are selected according to an upwinding condition. Very fast solutions are obtained by using heap-based algorithms similar to the "fast marching method" used for solving the Eikonal equation [6]–[8].

In Section II, we motivate the definition of thickness we have adopted here by discussing some drawbacks associated with a variety of alternative definitions. In Section III, we outline our Eulerian PDE approach and then give its numerical implementation in Section IV. Simulations on two illustrative synthetic examples followed by two examples for segmented cardiac and brain images appear in Section V, demonstrating the algorithm's performance in both two and three dimensions.

II. Thickness Definition

There have been many definitions of anatomical thickness in the literature. Left ventricular myocardial thickness is most often defined within a cross-sectional image, and is assumed to be the distance[1] between the endocardium and epicardium along a line passing through the long axis of the ventricle [9] (thought of as the origin), as shown in Fig. 1(a). This definition does not capture the three-dimensional (3-D) aspect of the heart wall, requires that the positions of the endocardium and epicardium are radial functions, and depends on the (arbitrarily defined) location of the long axis. Most often, the papillary muscles are not included in the segmentation of the endocardium in order to make sure that the endocardium is a radial function. With the increasing resolution of magnetic resonance (MR) images, a more precise, 3-D definition of thickness that is not susceptible to user variation is needed.

[1]The word *distance* means *Euclidean distance* unless otherwise stated.

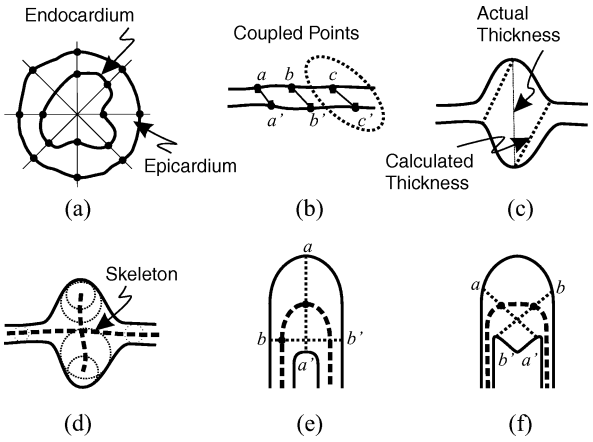

Fig. 1. Problems related to thickness definitions.

Brain cortex thickness has been defined in several ways. Coupled-surface methods, such as that in [10], define thickness as the distance between point pairs uniquely associated between the two surfaces. One problem with this approach is that the thickness measures will be artificially high if the two surfaces are displaced relative to one another, as shown in Fig. 1(b) (solid lines). This problem is addressed in [2], where the thickness is defined as the average of the two distances to the closest points on the opposing surfaces for each of the two paired points, as shown in Fig. 1(b) (dotted lines). This definition, however, loses the idea of unique point association between the two surfaces, and can also yield thickness measures that are too small, as demonstrated in Fig. 1(c).

Uncoupled surface thickness measures do not have *a priori* point associations between the two surfaces. A simple measure of thickness in this case is to measure the distance from each point on a given surface to the closest point on the opposing surface [11]. The most obvious problem with this definition is the lack of symmetry—the thickness is not the same when the surfaces are interchanged. Also, the thickness can be dramatically underestimated using this approach when there is a pronounced bulge in the opposing surface. Both of these problems are demonstrated in Fig. 1(c). It is possible to create point associations between the surfaces by shape matching [12], [13]. However, the standard definitions of thickness for coupled surfaces would now apply, and these suffer from the problems outlined above.

Another class of methods define thickness relative to a central axis or skeleton [4]. Generally, there is no point association between the central axis and the two surfaces, and thickness is typically treated as the diameter of the largest enclosed sphere centered at a given point on the central axis. The problem with this definition is that the skeleton will have to take on an arbitrary topology in order to properly describe highly convoluted objects. An abrupt change in thickness, for example, might require that the skeleton grow a branch in order to completely define the geometry of the two surfaces, as shown in Fig. 1(d). If, on the other hand, the topology of the central axis is restricted—to that of a simple sheet, for example—then the resulting thickness measurement often underestimates the actual thickness when either of the two surfaces is bumpy. Other definitions, such as the distance orthogonal to a central axis and minimal line through an axis, are fraught with difficulties, as is demonstrated in Fig. 1(e) and (f).

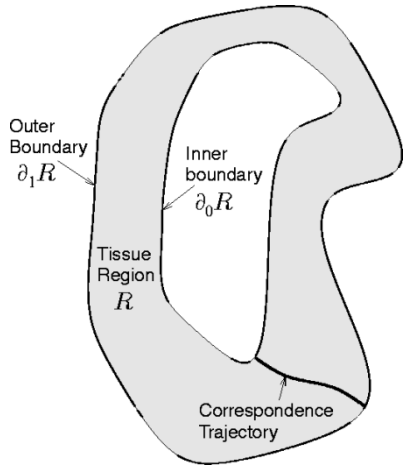

Fig. 2. Inner and outer boundaries, the computational domain, and a correspondence trajectory.

Jones *et al.* [5] proposed a new measure of cortical thickness based on curved lines connecting the two bounding surfaces. They proposed setting the potential of one surface to zero, setting the potential of the other surface to one, and solving Laplace's equation for the potential between the two surfaces (as in an electric field). The lengths of the lines of flow between the two surfaces then defines the thickness. These lines have desirable properties: they are orthogonal to each surface, they do not intersect, and they are nominally parallel. In this paper, we generalize this framework and specifically address the computation of trajectory lengths, which Jones *et al.* implemented by explicitly tracking the correspondence trajectories in a Lagrangian framework. In particular, our approach applies to arbitrary correspondence trajectories, not only those defined by Laplace's equation. Also, our approach is computationally stable and fast, and does not require explicit computation of the correspondence trajectories.

III. Eulerian PDE Approach

In this section, we outline a mathematical approach for measuring the thickness of segmented tissues using a pair of linear PDEs. Specifically, we show how a simple linear PDE can be used to compute trajectory lengths at all points using only a vector field of unit tangents. Although the unit tangent field can be constructed by any (reasonable) process, we will adopt in this paper the method presented by Jones *et al.* [5], which uses the normalized gradient of a harmonic function between the tissue boundaries. Solution of the two PDEs yields thickness directly, thereby eliminating the need to explicitly construct and measure any individual trajectory between the two boundaries.

A. Correspondence Trajectories

We assume that the tissue to be measured occupies a spatial region $R \subset \mathcal{R}^n$, $n = 2,3$, with exactly two simply connected boundaries $\partial_0 R$ and $\partial_1 R$, which we call the inner and outer boundaries, respectively. We define thickness at any point $\mathbf{x} \in R$ as the total arclength of a unique curve, passing through $\mathbf{x}$, which originates on $\partial_0 R$, terminates on $\partial_1 R$, and remains within the region R. This scenario is depicted in Fig. 2. Uniqueness is necessary in this definition in order to avoid ambiguity; it implies that we can construct a family of nonintersecting curves

connecting the boundaries in a bijective fashion. These curves, which we refer to as *correspondence trajectories*, also associate a unique point on each boundary to each point $\mathbf{x}$ in R.

Correspondence trajectories cannot, in general, be established via "closest point" relations between the two boundaries (see Section II). There are, however, an infinite number of vector fields defined on R whose streamlines satisfy the uniqueness requirements. Additional desirable properties narrow this list of suitable candidates. For example, correspondence trajectories should approach a boundary from a normal direction, they should proceed as directly as possible from one boundary to the other, and their speed should never go to zero.

B. Constructing the Tangent Field of the Correspondence Trajectories

It is convenient to construct a unit vector field that coincides with the tangent vectors of the correspondence trajectories. There are many possible choices, and the framework we will outline applies equally well to any particular choice. One possible choice is a normalized gradient vector flow field [14], where the boundaries play the role of edge maps. Another choice is the normalized gradient of the unique harmonic function u over R that interpolates between 0 along $\partial_0 R$ and 1 along $\partial_1 R$. This is the function used by Jones *et al.* [5], and is what we also use in the experiments presented in this paper. We obtain the harmonic function u by solving the Laplace equation over R

$$\Delta u = 0 \tag{1}$$

with the Dirichlet boundary conditions

$$u(\partial_0 R) = 0 \text{ and } u(\partial_1 R) = 1$$

where Δ is the Laplace operator. The corresponding tangent field $\vec{T}$ is given by

$$\vec{T} = \frac{\nabla u}{||\nabla u||} \tag{2}$$

where ∇ is the gradient operator.

C. Computing Thickness Directly From the Tangent Field

Typically, the tissue region R, whose thickness is to be measured, is given as a set of pixels or voxels on a rectangular grid. Thickness is defined at each point in R as the length of its correspondence trajectory. We now show that it is not necessary to explicitly construct the trajectories—e.g., by tracing the flow of particles—in order to calculate these lengths. The key is to devise an Eulerian framework that utilizes the fixed rectangular grid and to exploit the unit tangent field in a particular differential structure.

Let us assume that we are given a unit vector field $\vec{T}(\mathbf{x})$ defined on R such that the orientations of these tangents follow correspondence trajectories directed from the inner boundary $\partial_0 R$ to the outer boundary $\partial_1 R$. It follows that $-\vec{T}(\mathbf{x})$ follows the same correspondence trajectories, but is directed from $\partial_1 R$ to $\partial_0 R$. We define two length functions L_0 and L_1, where $L_0(\mathbf{x})$ gives the arclength of the correspondence trajectory between $\partial_0 R$ and $\mathbf{x}$, and $L_1(\mathbf{x})$ gives the arclength of the trajectory between $\partial_1 R$ and $\mathbf{x}$.

It follows from elementary differential geometry that the length functions L_0 and L_1 must satisfy the following first-order linear PDEs:

$$\nabla L_0 \cdot \vec{T} = 1, \qquad \text{with } L_0(\partial_0 R) = 0 \tag{3}$$

$$-\nabla L_1 \cdot \vec{T} = 1, \qquad \text{with } L_1(\partial_1 R) = 0. \tag{4}$$

We describe an efficient numerical scheme to simultaneously solve these PDEs in Section IV. Since the length functions L_0 and L_1 measure the arclengths starting from opposite endpoints of each correspondence trajectory, the total arclength of the trajectory through any point $\mathbf{x}$ is obtained by adding L_0 and L_1 as follows:

$$W(\mathbf{x}) = L_0(\mathbf{x}) + L_1(\mathbf{x}). \tag{5}$$

We define the quantity $W(\mathbf{x})$ as the *thickness* of the tissue region at $\mathbf{x}$. In this fashion, thickness is computed at every point in R without ever explicitly constructing a correspondence trajectory. We now describe an efficient numerical solution of (3) and (4).

IV. Numerical Implementation

There are many standard numerical methods for solving (1) [15]–[17], any one of which can be used to obtain u, and $\vec{T}$ follows immediately from (2). There are also many alternative ways to define and compute the tangent field $\vec{T}$ without using the Laplace equation. Therefore, we will not go into detail about the numerical computation of $\vec{T}$. Instead, we focus our attention here on the development of a numerical scheme for solving the PDEs (3) and (4) to obtain the length functions L_0 and L_1 assuming we are given the tangent field $\vec{T}$.

We note that the characteristics of the PDEs (3) and (4) are exactly the correspondence trajectories. Therefore, since the correspondence trajectories never intersect, we do not need to worry about shocks. This situation is in contrast to many first-order boundary value PDEs such as the Eikonal equation, which would yield other types of "closest point" correspondences. Due to this, we do not need to be concerned with entropy conditions in the numerical schemes to solve these PDEs. Appropriate consideration of upwinding in the computation of finite differences, however, is crucial.

Here, we will consider only the case of a 3-D rectangular grid with spacing h_x, h_y, and h_z between neighboring grid points (voxels) in the x-, y-, and z-directions, respectively. The two-dimensional (2-D) case is simply a special case of this case. The notation $T_x[i,j,k]$, $T_y[i,j,k]$, and $T_z[i,j,k]$ denotes the components of $\vec{T}$ at the grid point (i, j, k), and backward and forward differences are given by the following standard notation [7], [18], [19]:

$$D_x^- L = \frac{L[i,j,k] - L[i-1,j,k]}{h_x}$$

$$D_x^+ L = \frac{L[i+1,j,k] - L[i,j,k]}{h_x}$$

$$D_y^- L = \frac{L[i,j,k] - L[i,j-1,k]}{h_y}$$

$$D_y^+ L = \frac{L[i,j+1,k] - L[i,j,k]}{h_y}$$

$$D_z^- L = \frac{L[i,j,k] - L[i,j,k-1]}{h_z}$$
$$D_z^+ L = \frac{L[i,j,k+1] - L[i,j,k]}{h_z}.$$

A. Upwind Differencing

We start by considering various combinations of the above first-order differences to approximate ∇L_0 in (3), yielding various linear expressions for $L_0[i,j,k]$ in terms of three of its six neighbors $L_0[i\pm1,j,k]$, $L_0[i,j\pm1,k]$, and $L_0[i,j,k\pm1]$. Equation (3) with the possible choices can be written as follows:

$$\begin{aligned} 1 = {} & T_x[i,j,k]\left(D_x^- L_0[i,j,k] \text{ or } D_x^+ L_0[i,j,k]\right) \\ & + T_y[i,j,k]\left(D_y^- L_0[i,j,k] \text{ or } D_y^+ L_0[i,j,k]\right) \\ & + T_z[i,j,k]\left(D_z^- L_0[i,j,k] \text{ or } D_z^+ L_0[i,j,k]\right). \end{aligned} \quad (6)$$

Since (3) is a first-order PDE with known boundary values, its solution can be constructed by integration along characteristic curves starting from the known boundary. By design, the characteristics of (3) are precisely the correspondence trajectories and, therefore, the tangent vector $\vec{T}[i,j,k]$ determines the direction that the characteristic flows through the grid point $[i,j,k]$.

As stated above, we do not need to worry about shocks and entropy conditions [7], [18]–[20] since the characteristics of the linear PDEs (3) and (4) never intersect. On the other hand, information does flow in the forward direction (downwind) along the characteristics so it is important to choose our differencing scheme (i.e., D_x^- versus $D_x^+, \ldots$) so that the value of $L_0[i,j,k]$ only depends upon values of L_0 in the backward direction (upwind) along the characteristic passing through the grid point $[i,j,k]$. This direction is given by $-\vec{T}[i,j,k]$. Therefore, upwinding dictates the following choice for (6):

$$\begin{aligned} 1 = {} & T_x[i,j,k]\left\{\begin{array}{ll} D_x^- L_0[i,j,k], & -T_x[i,j,k] < 0 \\ D_x^+ L_0[i,j,k], & \text{otherwise} \end{array}\right\} \\ & + T_y[i,j,k]\left\{\begin{array}{ll} D_y^- L_0[i,j,k], & -T_y[i,j,k] < 0 \\ D_y^+ L_0[i,j,k], & \text{otherwise} \end{array}\right\} \\ & + T_z[i,j,k]\left\{\begin{array}{ll} D_z^- L_0[i,j,k], & -T_z[i,j,k] < 0 \\ D_z^+ L_0[i,j,k], & \text{otherwise} \end{array}\right\}. \end{aligned} \quad (7)$$

Solving (7) for $L_0[i,j,k]$ and using an analogous upwind scheme for $L_1[i,j,k]$ (noting that the upwind direction for L_1 is given by $\vec{T}$ rather than $-\vec{T}$) yields the finite-difference approximations shown in (8) and (9) at the bottom of the page, where

$$i \pm 1 = \begin{cases} i+1, & T_x > 0 \\ i-1, & T_x < 0 \end{cases}$$
$$j \pm 1 = \begin{cases} j+1, & T_y > 0 \\ j-1, & T_y < 0 \end{cases}$$
$$k \pm 1 = \begin{cases} k+1, & T_z > 0 \\ k-1, & T_z < 0 \end{cases}$$

and $i \mp 1$, $j \mp 1$, and $k \mp 1$ are defined the same way, except that the signs are reversed. For simplicity, in the above equations, we have assumed that $h_x = h_y = h_z = 1$.

B. Iterative Procedures

We can use (8) and (9) as the core of an iterative procedure to solve for the correspondence trajectory lengths L_0 and L_1. We now describe three possible iterative procedures, and consider their computation times and convergence behaviors.

1) Iterative Relaxation: The simplest procedure is the iterated parallel relaxation method given by the following algorithm:

Algorithm 1 (Iterative Relaxation):
[STEP l] Set $L_0 = L_1 = 0$ *at all grid points (values outside* R *will serve as boundary conditions).*
[STEP 2] Use (8) and (9) to update L_0 *and* L_1 *at points inside* R.
[STEP 3] Repeat Step 2 until the values L_0 *and* L_1 *converge.* ■

It is assumed in the above procedure that the values computed in Step 2 are stored in a buffer and not used until the next iteration, after all values in R have been computed—i.e., a Jacobian procedure. Convergence will be much quicker, however, if during each sweep through the points in R the values of L_0 and L_1 are updated in-place at each point immediately as they are computed. This strategy is a Gauss–Seidel procedure rather than a Jacobian procedure. Assuming that the points are visited in a reasonable order during each iteration, convergence will occur in far fewer steps since the updated values of L_0 and L_1 at each point will be computed using the newly updated values of points visited earlier in the current sweep through R.

Note that, although we initialize L_0 and L_1 to be 0 outside *both boundaries* (even though each length function should have a boundary condition of 0 along only one of the two boundaries), the update equations (8) and (9) are designed to look in opposite directions. Thus, one scheme will be affected by the zero boundary condition only along the inner boundary, while the other will be affected only along the outer boundary. Therefore, it is possible to update L_0 and L_1 simultaneously (as indicated

$$L_0[i,j,k] = \frac{1 + |T_x| L_0[i \mp 1, j, k] + |T_y| L_0[i, j, \mp 1, k] + |T_z| L_0[i,j,k \mp 1]}{|T_x| + |T_y| + |T_z|} \quad (8)$$
$$L_1[i,j,k] = \frac{1 + |T_x| L_1[i \pm 1, j, k] + |T_y| L_1[i, j, \pm 1, k] + |T_z| L_1[i,j,k \pm 1]}{|T_x| + |T_y| + |T_z|} \quad (9)$$

in Step 2), using zero boundary conditions (from Step 1) on both sides, which greatly simplifies the procedure.

2) Ordered Traversal: Extremely fast convergence can be obtained by visiting the points in the order that they are reached by the characteristic curves (the correspondence trajectories) as they flow away from the known boundary. Only one full sweep through the grid points in R would then be required to solve for L_0 followed by (or in parallel with) one other sweep, but in a different order, for L_1. Note that this suggests an algorithm similar to the "fast marching method" used in solving the Eikonal equation (see [6]–[8]). Here, though, the characteristics are known to us in advance, and the equations to be solved at each grid point are linear rather than quadratic. The philosophy, however, is exactly the same as follows:

Algorithm 2 (Ordered Traversal):
[STEP l] Initially tag all points in R as UNVISITED.
[STEP 2] Solve for L_0 at points next to the boundary $\partial_0 R$ (where $L_0 = 0$) and re-tag these points as VISITED.
[STEP 3] Find the grid point, within the current list of VISITED points, with thesmallest value of L_0 computed so far. Remove this point from the list and re-tag it as SOLVED.
[STEP 4] Update the values of L_0 using (8) for whichever neighbors of this grid point are not yet tagged as SOLVED. If any of these neighbors are currently tagged as UNVISITED, re-tag them as VISITED and add them to the current list of VISITED points.
[STEP 5] Stop if all points in R have been tagged SOLVED, else go to Step 3. ■

If a min-heap data structure is used to maintain the active list of visited, but not yet solved grid-points, then Step 3 can be performed without having to search the list, making the algorithm very fast. This procedure has the additional benefit of terminating automatically, eliminating the need to test for convergence. An analogous approach can be used to compute L_1 by first solving for points next to the boundary $\partial_1 R$ in Step 2 instead of $\partial_0 R$.

3) Cyclically Alternating Gauss–Seidel: A simpler scheme that converges reasonably quickly (but avoids the bookkeeping required for the above scheme) is to cyclically alternate the order that the grid points in R are updated during each iteration using simple orderings related to the rectangular grid structure. For example, in the first iteration, grid points $[i, j, k]$ could be visited in order of increasing i, increasing j, and increasing k; while in the next iteration, they could be visited in order of decreasing i, increasing j, and increasing k, and so on (yielding eight combinations for a 3-D grid and four combinations for a 2-D grid). If the correspondence trajectories are not highly convoluted, there will be large subregions of R during each iteration, where the characteristics run approximately along the current i-, j-, and k-directions and, thus, the optimal marching procedure will be closely approximated within these subregions.

4) Computation Times: We implemented the above algorithms in C/C++ on a 800-MHz Pentium III computer running Linux. For the 2-D cardiac example described below, convergence of the cyclic Gauss–Seidel iterative relaxation algorithm was achieved in fewer than ten iterations, and the total computation time was around 1 s. For the same example, the ordered traversal (heap-based) algorithm required only a small fraction of a second. For the 3-D brain example described below, convergence of the cyclic Gauss–Seidel iterative relaxation algorithm required 100 iterations, which took approximately 3 min. (Generally, more iterations are required as the geometry becomes more convoluted.) In contrast, the ordered traversal algorithm required only 12 s for this 3-D example.

In our approach, it is also necessary to solve the Laplace equation in order to provide the correspondence trajectory directions. Our present (nonoptimized) iterative solution of Laplace's equation requires about three times the amount of time as our iterative relaxation thickness computation for both the 2-D and 3-D examples. However, we expect that upon implementation of an optimized Laplace solver—e.g., a preconditioned conjugate gradient algorithm—this time will be cut by approximately a factor of ten. Therefore, we expect that, after further numerical optimization, we will be able to calculate the thickness of the brain cortex in a 3-D volume in approximately 1 min.

V. Experimental Results

In this section, we demonstrate our approach for computing thickness on both synthetic test regions with known values and segmented tissue regions in real images. The first three experiments are on 2-D regions (which can be fully visualized), while the last experiment is in 3-D.

A. Circular Annulus

As a first test of our algorithm, we constructed an annulus between two concentric circles with radii 80 and 160 (all units in pixels), as shown in Fig. 3(a). It is clear that the thickness of this region should be 80 everywhere. Fig. 3(b) shows the harmonic function u, which interpolates between 0–1 along the inner and outer boundaries, and the normalized gradient of u, which comprises the tangent field $\vec{T}$, as shown in Fig. 3(c). The trajectory lengths L_0 and L_1 [Fig. 3(e) and (e)] were computed using (8) and (9), respectively. Their sum, the thickness, is shown in Fig. 3(f), which visually reveals an expected "constant" value (exact experimental values ranged between 79.84–80.30).

B. Elliptic Annulus

As another test, we constructed an annulus between a circle of radius 40 (all units in pixels) and an ellipse with minor and major radii of 80 and 160, as shown in Fig. 4(a). This time, the computed thickness values ranged from 40.000 to 120.036 as we moved from points near the minor axis toward points near the major axis. Note that the correspondence trajectories are straight lines only along these two axes. Away from these axes,

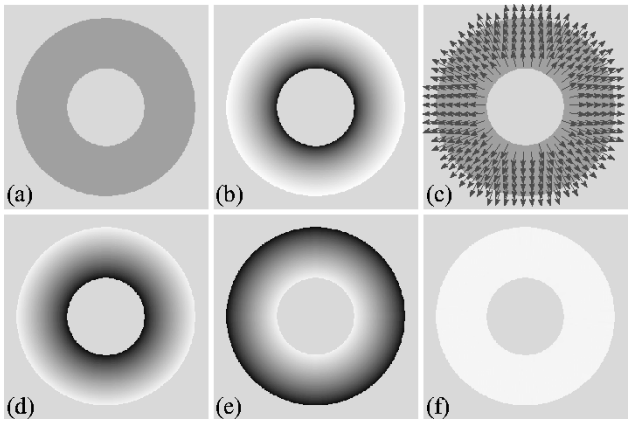

Fig. 3. Thickness computation illustrated for a synthetic annular region between two concentric circles. (a) Circular annulus. (b) Harmonic interpolant. (c) Tangent field $\vec{T}$. (d) Length L_0. (e) Length L_1. (f) Thickness $(L_0 + L_1)$.

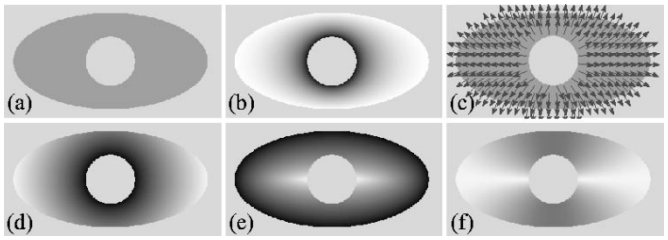

Fig. 4. Thickness computation illustrated for a synthetic annular region between an ellipse and a circle. (a) Elliptic annulus. (b) Harmonic interpolant. (c) Tangent field $\vec{T}$. (d) Length L_0. (e) Length L_1. (f) Thickness $(L_0 + L_1)$.

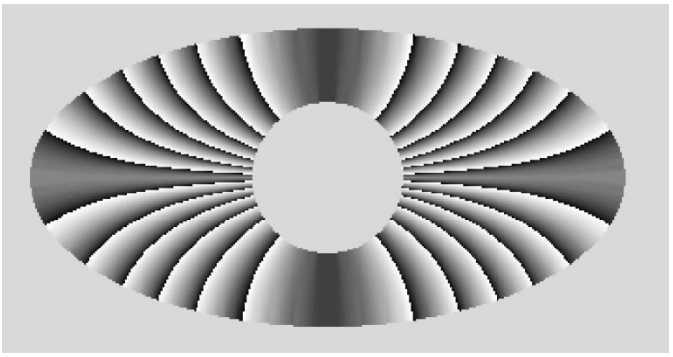

Fig. 5. Level sets of thickness function.

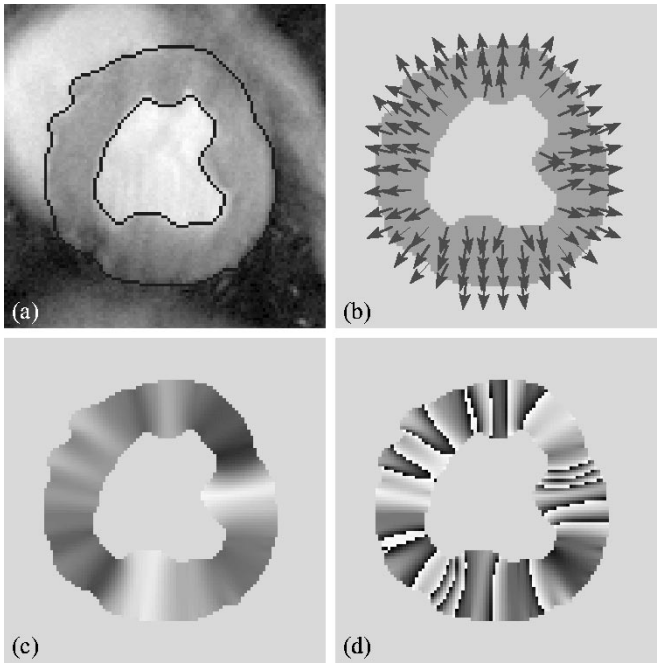

Fig. 6. Myocardial thickness from a short-axis MR image. (a) Endocardial and epicardial contours. (b) Tangent field. (c) Thickness. (d) Level sets of thickness function.

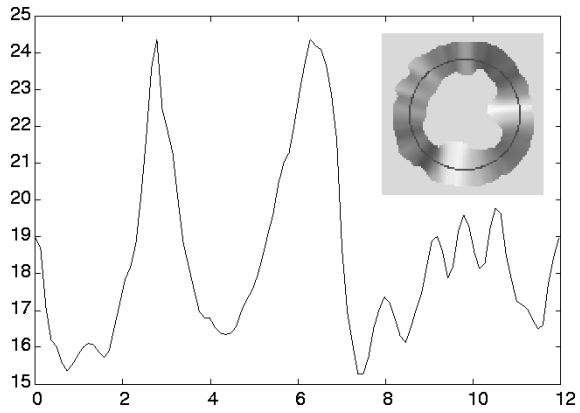

Fig. 7. Plot of computed myocardial thickness values around an inscribed circle (shown in the upper right-hand-side corner). The horizontal scale denotes the location on the circle in "hourly clock units" (1 o'clock, 2 o'clock,. . .), while the vertical scale measures the computed thickness in pixel units.

the trajectories are slightly curved in order to remain perpendicular to both boundaries and to avoid intersection. These trajectories, which were never explicitly computed, can be visualized in Fig. 5, which reveals level sets of the computed thickness function shown in Fig. 4(f).

C. Myocardium

Next, we applied our method to a segmentation of the myocardium obtained from a short-axis MR image of the heart, both of which are shown in Fig. 6(a). A subsample of the tangent vectors computed from solution of Laplace's equation are shown in Fig. 6(b). Evidence of the need to form curved correspondence trajectories is apparent inside the papillary muscle appearing at approximately 3 o'clock on the inside boundary. The calculated thickness is shown in Fig. 6(c), which shows brighter regions where one sees thicker myocardium. An isocontour plot is shown in Fig. 6(d) revealing some of the curved correspondence trajectories over which thickness is (implicitly) computed.

Fig. 7 shows a different visualization of computed thickness, which complements the grayscale presentation in Fig. 6(c). Here, the thickness values are read along an inscribed circle (shown in this figure) and plotted in a clockwise manner starting from 12 o'clock. For convenience and comparison, the grayscale thickness representation from Fig. 6(c) is shown in the upper right-hand-side corner of this figure along with the inscribe circle around which the thickness measurements are plotted. As can be seen from both the two large peaks in the plot and the two bright areas in the grayscale visualization, the thickest portions of this segmented myocardium occur at roughly 3 and 6 o'clock.

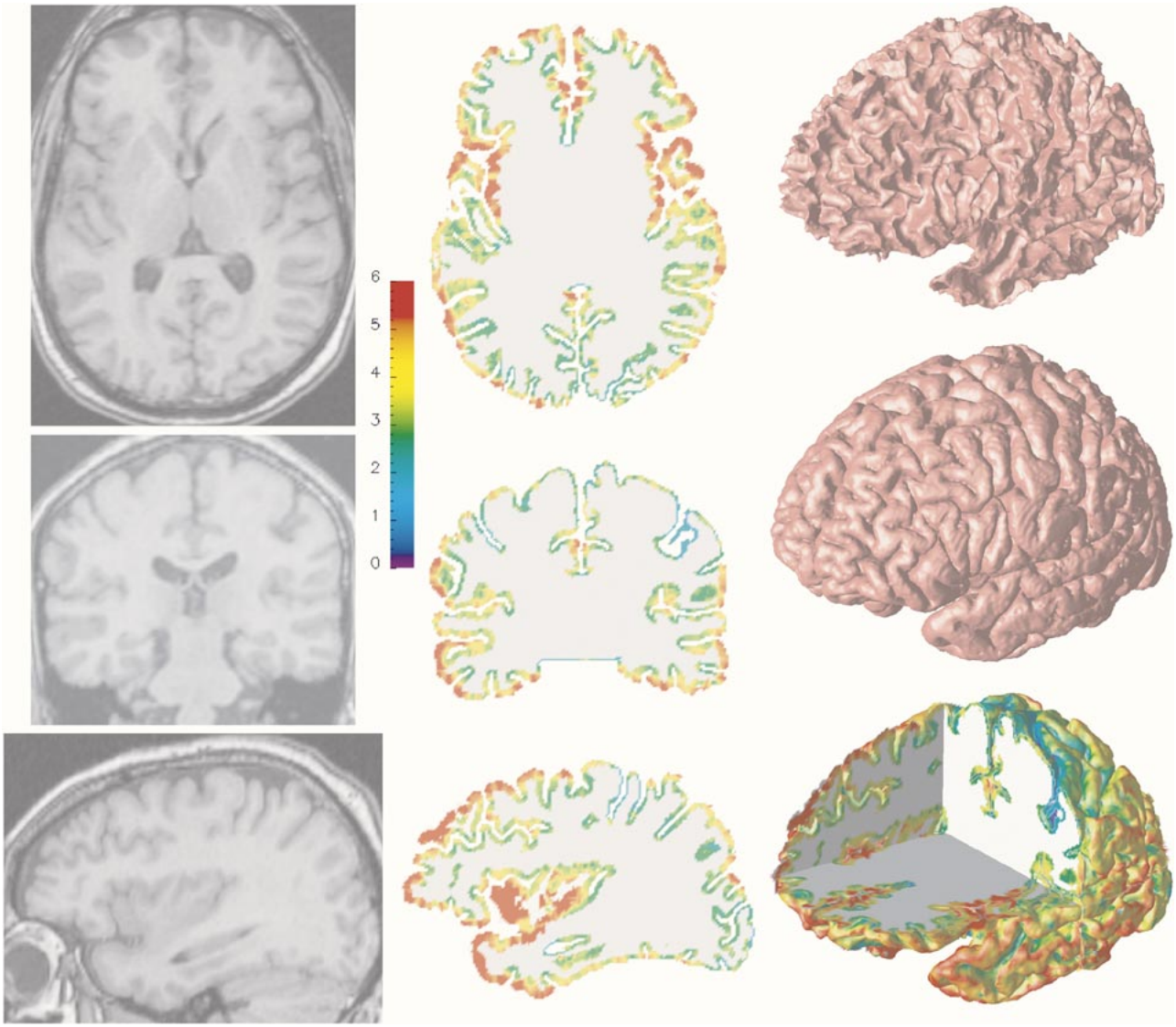

Fig. 8. 3-D cortical thickness calculation. Original slices in left-hand-side column, segmented inner and outer surfaces on upper right-hand side, gray matter thickness in center column, and composite thickness on slices and central cortical surface.

D. Cortical Gray Matter

Next, we applied our method to a 3-D segmentation of the cortex obtained from MR images of a human brain. Representative MR cross sections are shown in the left-hand-side column of Fig. 8. The inner and outer surfaces of the cortical gray matter, segmented from the MR images (cf. [10], [21], and [22]), are shown on the upper right-hand side. Thickness was computed in the region between these two boundaries, and the results for the three MR cross sections shown are shown in the middle column of Fig. 8. The color bar displayed indicates the computed thickness in millimeters. Finally, a composite representation of thickness is shown on the bottom right-hand side of Fig. 8. This figure shows the thickness on a central surface representation of the gray matter, and is also cut away at the cross sections shown in the first two columns of Fig. 8.

E. Tibial Cartilage

Finally, we applied our method to a 3-D segmentation of the two tibial cartilages of the knee [see Fig. 9(a)]. In contrast to previous examples, the inner and outer boundaries—surfaces, in this case—do not comprise nested boundaries, and the region is not an annular region. Instead, we let these boundary surfaces be the "top" and "bottom" surfaces, respectively, which meet around the edges of the cartilages. The bottom surface is defined to be the portion of the cartilage surface that is in common with the tibia, and the top surface is the remaining "exposed" cartilage surface. Thus, defining these surfaces requires segmentation of both the tibia and tibial cartilages. Once defined, however, computation of the thickness using our method proceeds without change.

Two visualizations of the thickness map of the tibial cartilages of a human left knee, computed using our technique with the boundaries defined as described above, are shown in Fig. 9. In Fig. 9(a), we are able to see the shape of the segmented cartilages as well as the computed thickness mapped onto the boundary surfaces. Fig. 9(b) shows the thickness map within the volume of the cartilages, cut away in two cross sections. In both figures, the thickness is given in "voxels," where a "voxels" is approximately 1 mm. It is evident that the thickness is zero

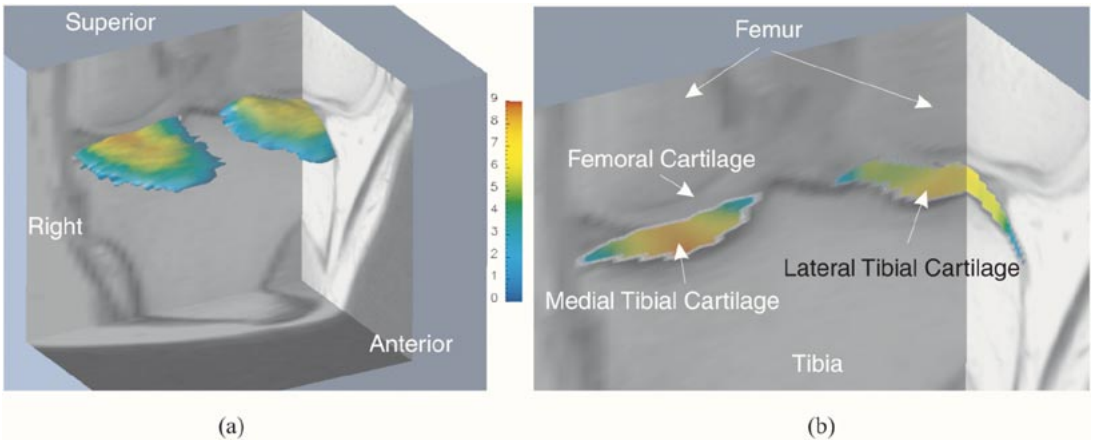

Fig. 9. Computed thickness of the tibial cartilages of the human knee displayed: (a) on the cartilage surfaces and (b) on a cutaway of the MR image volume.

around the edges where the surfaces meet, and grows to a maximum in the interior of each cartilage. This example suggests that with sensible definitions of "inner" and "outer" boundaries, our thickness computation method can be applied to a wide variety of anatomical objects, which need not have annular shapes.

VI. Conclusion

We have presented a fast and accurate method for computing the thickness of segmented objects bounded by two contours or surfaces. Our method was motivated by the need to measure thickness in various tissues seen in medical images. The method uses a two-stage approach in which a unit tangent field is first constructed by appropriate means—e.g., the solution of Laplace's equation within the object—and then the thickness is computed by combining the solution of two linear first-order PDEs. The numerical method is constructed by using appropriate upwinding conditions within an iterative finite-differencing framework. In certain medical-imaging applications, this algorithm might eventually be thought of as a companion to segmentation, producing thickness data carrying significant diagnostic and/or scientific value. As well, we believe that this overall two-stage Eulerian PDE approach has potential for use in other applications besides medical imaging and for other computations besides thickness.

Acknowledgment

The authors thank Dr. C. Xu for many thoughtful conversations on the subject of thickness and to X. Han for help in generating much of the experimental data.

References

[1] M. E. DeBakey and A. M. Gotto, Jr., *The New Living Heart*. Holbrook, MA: Adams Media Corporation, 1997.
[2] B. Fischl and A. M. Dale, "Measuring the thickness of the human cerebral cortex from magnetic resonance images," *Proc. Nat. Acad. Sci.*, vol. 97, no. 20, pp. 11 050–11 055, 2000.
[3] J. Nolte and J. B. Angevine, Jr., *The Human Brain: In Photographs and Diagrams*. St. Louis, MO: Mosby, 1995.
[4] S. Pizer, D. Eberly, D. Fritsch, and B. Morse, "Zoom-invariant vision of figural shape: The mathematics of cores," *Comput. Vision Image Understanding*, vol. 69, no. 1, pp. 55–71, 1998.
[5] S. E. Jones, B. R. Buchbinder, and I. Aharon, "Three-dimensional mapping of the cortical thickness using Laplace's equation," *Hum. Brain Mapping*, vol. 11, pp. 12–32, 2000.
[6] L. Cohen and R. Kimmel, "Global minimum for active contour models: A minimal path approach," *Int. J. Comput. Vis.*, vol. 24, no. 1, pp. 57–78, Aug. 1997.
[7] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge, U.K.: Cambridge Univ. Press, 1996.
[8] J. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1528–1538, Sept. 1995.
[9] W. Grossman, "Assessment of regional myocardial function," *J. Amer. Coll. Cardiol.*, vol. 7, no. 2, pp. 327–328, 1986.
[10] D. MacDonald, N. Kabani, D. Avis, and A. C. Evans, "Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI," *NeuroImage*, vol. 12, pp. 340–356, 2000.
[11] M. I. Miller, A. B. Massie, J. T. Ratnanather, K. N. Botteron, and J. G. Csernansky, "Bayesian construction of geometrically based cortical thickness metrics," *NeuroImage*, vol. 12, pp. 676–687, 2000.
[12] J. C. McEachen, II and J. S. Duncan, "Shape-based tracking of left ventricular wall motion," *IEEE Trans. Med. Imag.*, vol. 16, pp. 270–283, Mar. 1997.
[13] H. D. Tagare, "Deformable 2-D template matching using orthogonal curves," *IEEE Trans. Med. Imag.*, vol. 16, pp. 108–117, Feb. 1997.
[14] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, pp. 359–369, Mar. 1998.
[15] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1988.
[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*. New York: PWS, 1996.
[17] W. Briggs, *A Multigrid Tutorial*. Philadelphia, PA: Soc. Ind. Appl. Math., 1987.
[18] S. Osher, "Riemann solvers, entropy condition, and difference approximations," *SIAM J. Numer. Anal.*, vol. 21, no. 2, pp. 217–235, Apr. 1984.
[19] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.
[20] R. LeVeque, *Numerical Methods for Conservation Laws*. Boston, MA: Birkhäuser, 1992.
[21] C. Xu, D. L. Pham, M. E. Rettmann, D. N. Yu, and J. L. Prince, "Reconstruction of the human cerebral cortex from magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 18, pp. 467–480, June 1999.
[22] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan, "Segmentation and measurement of the cortex from 3-D MR images using coupled-surfaces propagation," *IEEE Trans. Med. Imag.*, vol. 18, pp. 927–937, Oct. 1999.