Estimation of 3-D Left Ventricular Deformation From Medical Images Using Biomechanical Models

Xenophon Papademetris*, Albert J. Sinusas, Donald P. Dione, R. Todd Constable, and James S. Duncan

Abstract—The quantitative estimation of regional cardiac deformation from three-dimensional (3-D) image sequences has important clinical implications for the assessment of viability in the heart wall. We present here a generic methodology for estimating soft tissue deformation which integrates image-derived information with biomechanical models, and apply it to the problem of cardiac deformation estimation. The method is image modality independent. The images are segmented interactively and then initial correspondence is established using a shape-tracking approach. A dense motion field is then estimated using a transversely isotropic, linear-elastic model, which accounts for the muscle fiber directions in the left ventricle. The dense motion field is in turn used to calculate the deformation of the heart wall in terms of strain in cardiac specific directions. The strains obtained using this approach in open-chest dogs before and after coronary occlusion, exhibit a high correlation with strains produced in the same animals using implanted markers. Further, they show good agreement with previously published results in the literature. This proposed method provides quantitative regional 3-D estimates of heart deformation.

Index Terms—Cardiac deformation, left ventricular motion estimation, magnetic resonance imaging, nonrigid motion estimation, validation.

I. INTRODUCTION

ACUTE coronary artery occlusion results in myocardial injury, which will progress from the endocardium to the epicardium of the heart wall in a wavefront fashion. A primary goal in the treatment of patients presenting with acute myocardial infarction is to reestablish coronary flow, and to interrupt the progression of injury, thereby salvaging myocardium. Unfortunately, there are no universally accepted noninvasive imaging approaches for the accurate determination of the extent of injury. Using conventional measures of regional myocardial function, the extent of myocardial infarction is overestimated. This can be attributed to persistent postischemic dysfunction ("stunning"), persistent myocardial hypoperfusion ("hibernation"), or mechanical tethering of normal areas by the adjacent injured myocardium. This tethering can be seen at the lateral margins of an infarct, resulting in a viable although dysfunctional border zone. Motion of the viable epicardium can also be constrained by injury of the underlying endocardial myocardial tissue. The location and ultimate transmural extent of the injury has important implications for long term prognosis of patients following myocardial infarction. Those patients with transmural myocardial infarction are likely to dilate their left ventricles over time, a condition termed left ventricular (LV) "remodeling." The occurrence of postinfarction remodeling carries a much worse long-term prognosis.

A number of laboratories have shown that a comprehensive quantitative analysis of myocardial strain can more accurately identify ischemic injury than can simple analysis of endocardial wall motion or radial thickening [5]. Furthermore, the characterization of segmental strain components has shown great promise for defining the mechanical mechanisms of tethering or remodeling [25], [28]. Experimental animal studies demonstrate that decreased circumferential shortening in myocardial regions adjacent to the infarct zone relative to remote regions is associated with late LV remodeling [24]. At present, quantitative noninvasive measurement of three-dimensional (3-D) strain properties from images has been limited to special forms of magnetic resonance (MR) acquisitions, specifically MR tagging, and to a lesser extent MR phase contrast velocity.

The MR tagging approach to the measurement of myocardial strain was originally developed, and then vigorously pursued further by two groups, one at the University of Pennsylvania [4] and the other at Johns Hopkins [30]. In general, there are three different approaches to estimating displacement data from MR tagging. The first approach involves tagging in multiple intersecting planes at the same time, and using the tag intersections as tokens for tracking (e.g., [1], [23], and [55]). The second approach involves tagging in multiple intersecting planes, one set of parallel planes at a time. Then, each tagging plane is used separately to estimate the normal direction of motion perpendicular to the plane. This generates a set of partial displacements (i.e., the component parallel to the tag lines is missing) to be combined later (e.g., [11] and [18]). The final approach uses a lower resolution modulation technique and attempts to model the tag fading over time using the Bloch equations. The displacements are then extracted using a variable brightness optical flow technique (e.g., [17] and [42]). The reader is also referred to a recently published book [3].

*X. Papademetris is with the Department of Diagnostic Radiology, Yale University, New Haven, CT 06520-8042 USA (e-mail: papad@noodle.med.yale.edu).

A. J. Sinusas is with the Department of Diagnostic Radiology and the Department of Medicine, Yale University, New Haven, CT 06520-8042 USA.

D. P. Dione was with the Department of Medicine, Yale University, New Haven, CT 06520-8042 USA.

R. T. Constable is with the Department of Diagnostic Radiology, Yale University, New Haven, CT 06520-8042 USA.

J. S. Duncan is with the Department of Electrical Engineering and the Department of Diagnostic Radiology, Yale University, New Haven, CT 06520-8042 USA.

As an alternative to MR tagging, several investigators have employed changes in phase due to motion of tissue within a fixed voxel or volume of interest to assist in estimating instantaneous, localized velocities, and ultimately cardiac motion and deformation. While the basic ideas were first suggested by van Dijk [52] and Nayler [33], it was Pelc and his team [39]–[41] that first bridged the technique to conventional cine MR imaging (MRI) and permitted the tracking of myocardial motion throughout the cardiac cycle. This technique basically relies on the fact that a uniform motion of tissue in the presence of a magnetic field gradient produces a change in the MR signal phase that is proportional to velocity. In general, two approaches have emerged to assemble deformation information from phase contrast images: 1) processing the data directly to estimate strain rate (e.g., [40] and [53]) and 2) integrating the velocities over time, via some form of tracking mechanism to estimate displacements (e.g., [10], [19], [32], [56], and [57]).

The use of computer vision-based techniques to estimate displacement is also possible. One approach to establishing correspondence is to track shape-related features on the LV over time as reported by Kambhametu [22], Cohen [9], Amini [2], McEachen [29] and Shi [46]. This is the basis for much of our own work and is expanded later. In general, here, preliminary displacements are estimated by matching local curvatures from segmented surfaces from consecutive time frames and then the estimates are smoothed to produce final displacement values.

Finally, some investigators have used the intensity of the images directly to track local LV regions. Song and Leahy [47] used the intensity in ultrafast CT images to calculate the displacement fields for a beating heart. In addition, other investigators have used local image intensity or intensity-based image texture from echocardiographic image sequences to track local positions over two-dimensional (2-D) image sequences [27], [31]. These efforts, along with some related MR tagging approaches (e.g., [17]) roughly fall into the category of optical flow-based methods. With the exception of methods based on magnetic resonance tagging and to a lesser extent MR phase contrast velocities, none of the other methods is capable of estimating complete 3-D deformation maps of the left ventricle.

In this paper, we present a modality-independent method for estimating 3-D LV deformation from 3-D image sequences. In this current paper, we have applied this methodology to ordinary cine-MR data and 3-D cine-CT data. An earlier version of this methodology was previously applied to 3-D echocardiographic sequences [37]. This paper constitutes the definitive presentation of this methodology, including the complete description of both the geometrical algorithms as well as the integration framework. Moreover, we present results from new 3-D image sequences from both animals and humans. Our algorithm derived strains are validated in the experimental models using surgically implanted markers.

The paper is structured as follows (see also the schematic in Fig. 1). In Section II, we describe the method of reconstructing the left-ventricular bounding surfaces from planar contours. The surfaces are then used as inputs to the shape-based tracking algorithm described in Section III, which is used to generate the initial displacement field. Section IV describes the mesh generation algorithm used to generate the volumetric model for the left ventricle. In Section V, we introduce some concepts from continuum mechanics and describe the model used for the myocardium. In Section VI, we outline the framework used for estimating a complete volumetric displacement field. Experimental results and validation are discussed in Section VII.

II. SURFACE RECONSTRUCTION

The left ventricle is segmented on a slice by slice basis using a custom designed software platform [36]. The segmentation algorithm results in a set of planar contours parameterized using b-splines (as shown in Fig. 2—step 1) which are subsequently sampled to generate a discrete set of points on each plane (see Fig. 2—step 2).

From these contours, we reconstruct the endo- and epicardial surfaces in a two-step procedure as follows: 1) We interpolate between contours to generate in-between contours at the desired sampling distance. This results in an iso-sampled set of points in three dimensions. 2) We construct a surface mesh by forming triangles between the points (as shown in Fig. 2—step 3). This defines the neighborhood relationship between points which is then used in the smoothing and curvature calculation algorithms.

A. Shape-Based Interpolation of Contours

We use a subpixel adaptation of work presented by Herman [20] to interpolate between contours. In [20], the interpolation was done at a pixel level resolution. However, given that the motion we are trying to estimate is of the order of half of a voxel per frame, subpixel resolution is needed.

The first step in the interpolation process is to convert each contour into a gray-value 2-D image known as the *distance map,* where pixel values represent the shortest signed distance of points from the contour, with positive values for pixels inside the contour and negative values outside. The algorithm is initialized by assigning distances to all points that lie within two pixels of the contour using exhaustive search. Then, the complete distance map is calculated by performing two consecutive *chamfering* processes [20] using the template shown in Fig. 3. These templates are scaled versions of the ones used in [20], the scaling being done to improve subpixel resolution, while still remaining within the two-byte integer range. The choices of the original unscaled two 3×3 templates have been justified to be near-optimal [20]. The resulting image represents the chamfer distance map of the given contour.

The second step in the interpolation process is the generation of the output distance map. This is done by combining the input distance maps in the appropriate way. If we label two contours as c_1 and c_2 and their distance maps as $d_m(c_1)$ and $d_m(c_2)$, respectively and we need to find the mean contour c_m, we first generate $d_m(c_m) = (d_m(c_1) + d_m(c_2))/2$. [Note that in the region between contours c_1 and c_2, $d_m(c_1)$ and $d_m(c_2)$ have opposite signs, hence, this results in a zero set.]

The third step is the extraction of c_m from its distance map d_m. We define c_m to be the zero level set in the distance map d_m and we extract it using a border following scheme adapted from the level-set work of Malladi *et al.* [43] (which in turn is
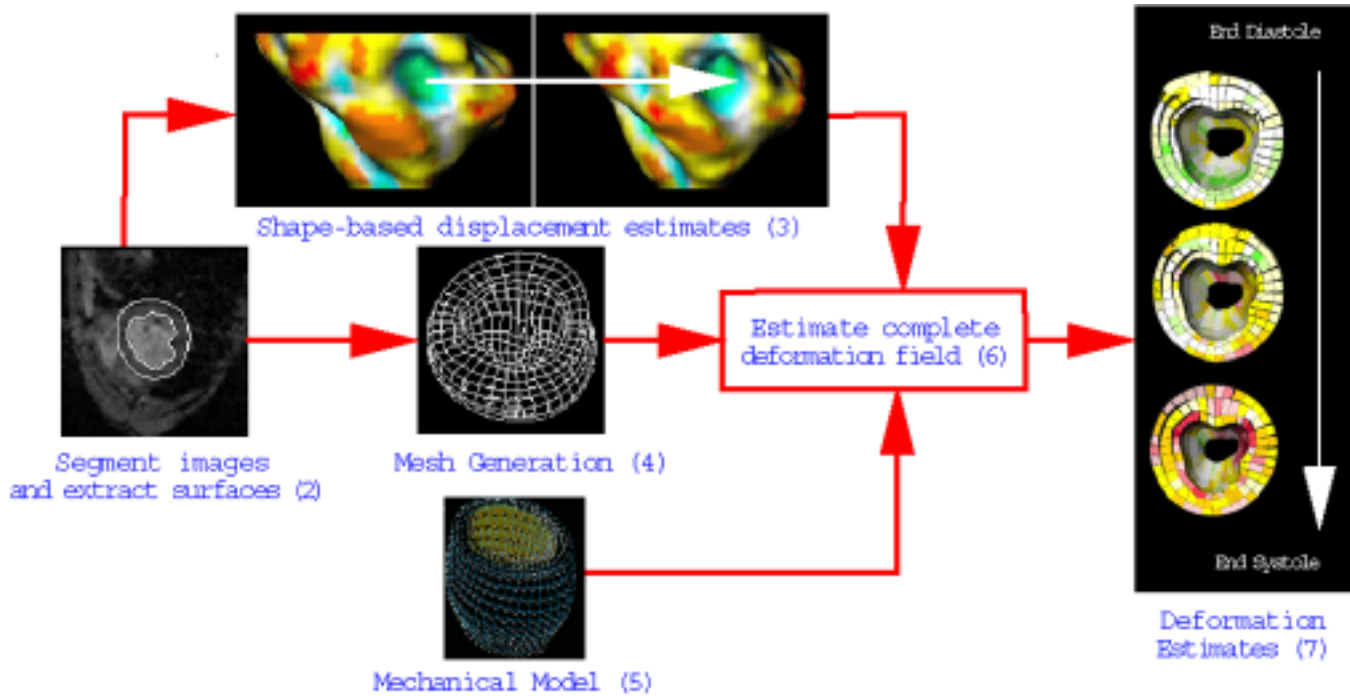
Fig. 1.   Outline of the proposed algorithm. The numbers in parenthesis [e.g., Mesh Generation (IV)] refer to the section in the paper where this component is described, i.e., mesh generation is described in Section IV.
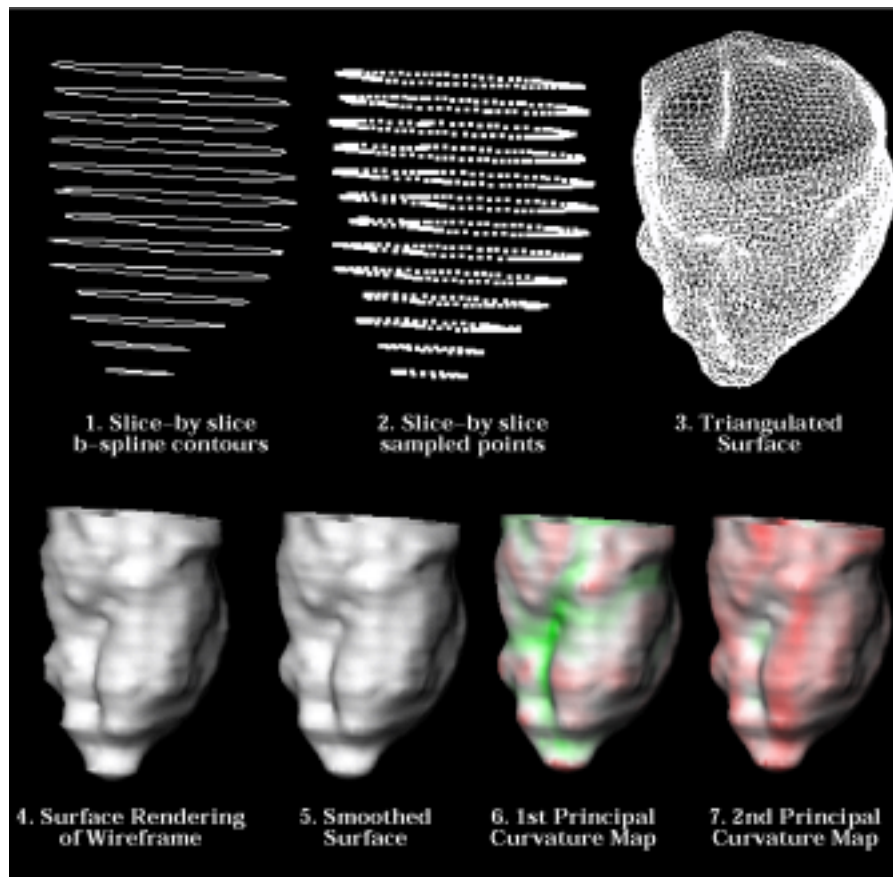

Fig. 2.   Steps involved in moving from slice by slice contours to full surface representation. 1) Slice by slice B-spline parameterized contours as extracted by the segmentation process. 2) Discretized contours as equally-spaced points. 3) Formation of wire-frame by Delaunay triangulation. 4) Surface rendering. 5) Smoothing of surface using nonshrinking smoothing algorithm. 6) and 7) First and second principal curvatures of the surface. Here, green shows negative (i.e., inward) curvature, white shows flat regions and red indicates positive (i.e., outward) curvature.
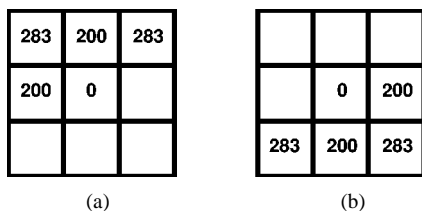
Fig. 3. Chamfer transformation templates. The two templates used by the dual chamfering processes to calculate the distance maps: template (a) for the top-to-bottom, left-to-right chamfering, and template (b) for the bottom-to-top, right-to-left chamfering.
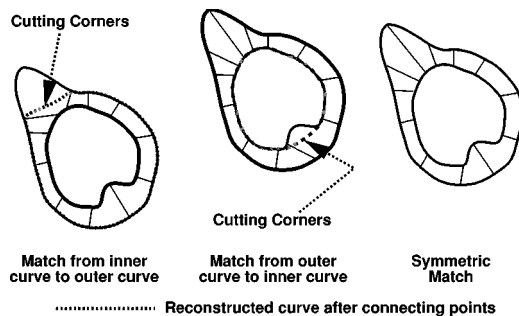

Fig. 4. Illustration of problems with asymmetric nearest neighbor matches. The two examples (left and middle) where the correspondence is driven exclusively in one direction show problems such as "cutting corners" when the two curves are not roughly parallel. In the third case, by using a symmetric nearest neighbor map the problem is avoided.

derived from the marching cube work of Lorenson [26]). It is this last step which gives the method its subpixel resolution compared with the one used in Herman [20]. (See [34], for more details.)

B. Delaunay Triangulation Between Planar Contours

Having created an isotropic set of points on the surface we reconstruct the surface using 2-D-constrained Delaunay triangulation [45]. The algorithm described here takes advantage of the fact that the original points lie on parallel planar contours and are ordered. The resulting triangulation has the smallest total length of triangle sides of all possible triangulations between the two planar contours. This triangulation method is optimal in that *no* flipping of connections can decrease the total length of all the sides of all the triangles.

For the case of constructing a set of triangles between two discretized, anticlockwise oriented, closed planar contours, c_1 and c_2, the procedure is as follows: (a proof of optimality can be found in [34]). First, for a point p_1 on contour c_1 find the nearest point to it p_2 on contour c_2. For this p_2 find the nearest point to it $\hat{p}_1$ on contour c_1. If $p_1 = \hat{p}_1$ label $s_1 = p_1$, $s_2 = p_2$ as the starting pair of points and start creating triangles. Otherwise, if $p_1 \neq \hat{p}_1$ choose another point on contour c_1 and repeat the initialization step. The process fails if there is no point p_1 for which this criterion is satisfied. (This is extremely unlikely.) Once the original seed points s_1 and s_2 are defined, we start the connection process. First, we define two test points t_1 and t_2. t_1 is the next point along c_1 from s_1, and t_2 which is the next point along c_2 from s_2. If $|t_1 - s_2| < |t_2 - s_1|$ label the next point $n_p = t_1$ else $n_p = t_2$, and add triangle s_1, s_2, n_p to the list. Then, if $n_p = t_1$ then set $s_1 = n_p$, else set $s_2 = n_p$. We then proceed to define new test points t_1 and t_2 unless $s_1 = p_1$ and $s_2 = p_2$ in which case the algorithm terminates.

Connectivity Distance: The Delaunay triangulation defines the connectivity of the points on each surface. We further define the distance between the two points to be the order of their connection. A point has a distance of zero with itself, a distance of one with a first-order neighbor (a point with which it is directly connected by the edge of a triangle), a distance of two with a second-order neighbor, and so on. We will call this the connectivity distance d_c.

C. Surface Postprocessing

The constructed surfaces are smoothed using the nonshrinking two-stage Gaussian algorithm proposed by Taubin [50]. (This is compared with the more typical one-stage

Gaussian filtering in [45].) Then, curvatures are computed using the same method as was used in [45] and [46]. At each point on the surface we compute the two principal curvatures k_1 and k_2 as illustrated in Fig. 2.

III. THE SHAPE TRACKING ALGORITHM

In this section, we describe the shape-tracking algorithm, which is an extension of the work by Shi [46]. The key improvement over this previous work is the use of the symmetric nearest neighbor algorithm to initialize the shape-tracking. We first describe the estimation of symmetric nearest neighbor correspondences, first in curves, and then its extension to surfaces. Then, we describe the shape-tracking algorithm itself. We note that the curve version of the symmetric nearest neighbor algorithm is used in the mesh generation algorithm described in Section IV.

A. Symmetric Nearest Neighbor Correspondences in Curves

In most computer-vision applications and in related work [46], [29], the estimation of initial correspondences is done using what we will term an "asymmetric nearest neighbor" technique. In this case, for each point on curve/surface c_1 the nearest point on curve/surface c_2 is found and labeled as the initial point. This has problems when the two curves are locally not parallel as whole regions of one curve map to a single point on the other curve. Also, whole regions on the second curve may not contribute to this map resulting in "cutting corners" as demonstrated in Fig. 4. In this section, we focus on the 2-D case; we present extensions to the full 3-D case in Section III-B.

Motivated by the bimorphism work of Tagare [49] we develop a symmetric technique to estimate initial correspondences without "cutting corners." This is important so as to ensure that, as much as possible, the whole of curve c_1 maps to the whole of curve c_2 and that the map is free from singularities (such as two points mapping to the same point) which are not either permissible or plausible in the areas of application of this algorithm. The symmetric nearest neighbor algorithm has three steps as follows.

Step 1. For all points on curve c_1 find the nearest neighbor on curve c_2. So, for example, for a point p_1 on curve c_1 we have a corresponding point p_2 on curve c_2. Then, for point p_2 estimate its nearest neighbor $\hat{p}_1$
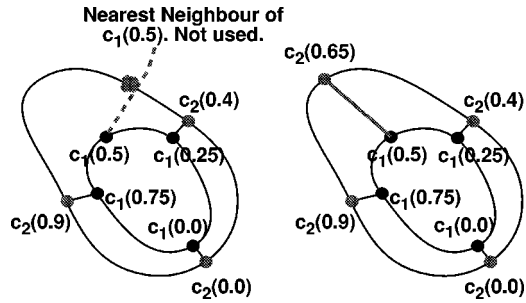
Fig. 5. An example of the 2-D implementation of the symmetric nearest neighbor algorithm. In this case, we try to map the inner curve c_1 to the outer curve c_2. Curve c_1 is defined by four points $(c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75))$, all of which apart from $c_1(0.5)$ have a symmetric nearest neighbor. The nearest neighbor of $c_1(0.5)$ is shown on the left (bad) and the point $c_1(0.5)$ is mapped to by the algorithm is shown on the right [$c_2(0.65)$ good!].
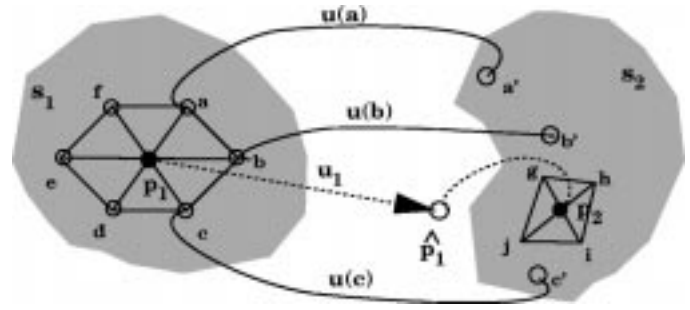

Fig. 6. Symmetric Nearest Neighbor Algorithm in 3-D. A portion of surface s_1 is shown on the left centered on a point p_1 which has first-order neighbors a, b, c, d, e, f. Of these neighbors a, b, c have symmetric nearest neighbors a', b', c' on s_2 shown on the right. p_1 itself does not have a symmetric nearest neighbor on s_2. We generate the first estimate of the position of the corresponding point of p_1, $\hat{p}_1$, by averaging $u(a)$, $u(b)$ and $u(c)$, the displacement vectors of points a, b, c to estimate a vector u_1 and translating p_1 by u_1. Then, $\hat{p}_1$ is mapped to surface s_2 by finding its (asymmetric) nearest point on surface s_2. This is point p_2 which is the corresponding point of point p_1 on surface s_2. We also define $u(p_1)$ (not shown) as $u(p_1) = p_2 - p_1$. We further show the first-order neighbors of p_2 on surface s_2 labeled as g, h, i and j.

on c_1. If $p_1 = \hat{p}_1$ then the points (p_1, p_2) are symmetric nearest neighbors and the match is retained. Otherwise, the match is discarded.

Step 2. For all points on curve c_1 which do not have symmetric nearest neighbors on c_2, find a matching point on c_2 by interpolating between the matching points of its neighbors. We do this until all points on c_1 have a matching point on c_2.

Step 3. Smooth the displacement field slightly to eliminate potential near-singularities.

The interesting part of this algorithm is the interpolation step (step 2). We take advantage of the fact that a curve can be parameterized using its arclength. An example will help to illustrate the point: consider the case that curve c_1 has four points $[(c_1(0.0), c_1(0.25), c_1(0.5), \text{ and } c_1(0.75)]$ which match to different positions on c_2, as illustrated by Fig. 5, and noting that $c_1(s_1)$ represents the point on curve c_1 at arclength of $s = s_1$. In this case, step 1 resulted in three symmetric neighbor pairs and left one point without a match. We can represent the points on c_2 by their arclengths as follows: $[c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75)] \mapsto (c_2(0.0), c_2(0.4), ??, c_2(0.9)]$.

In this case, point $c_1(0.5)$ has no corresponding point after step 1. To generate a match for $c_1(0.5)$ we interpolate between the corresponding points of $c_1(0.25)$ and $c_1(0.75)$ the nearest points to $c_1(0.5)$ on c_1 that do have *symmetric* nearest neighbors. This results in $c_1(0.5) \mapsto c_2(0.65)$.[1] The final result of step 2 is: $[c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75)] \mapsto [c_2(0.0), c_2(0.4), c_2(0.65), c_2(0.9)]$.

Then, in step 3 we smooth the displacements slightly (by convolving the arclengths on c_2 with a small Gaussian kernel) to ensure no near singularities. This could result in a map such as: $[c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75)] \mapsto [c_2(0.05), c_2(0.38), c_2(0.62), c_2(0.88)]$ which tries to equispace the points on c_2 subject to staying close to their original positions. For this approach to work well in practice, where the curves are discretized, c_2 has to be sampled much more finely than c_1 (typically five to eight times more).

B. Symmetric Nearest Neighbor Correspondences in Surfaces

In this section, we extend the work of the previous section to three dimensions. The key step here is to find a way of replacing the arclength parameterization. We do this by using the Euclidean distance. We focus here on steps 2 and 3 of the algorithm; step 1 is identical to the 2-D case.

Before proceeding to the description of the interpolation step we note that if a point p_1 on surface s_1 is mapped to a point p_2 on surface s_2 then we define the displacement vector $u(p_1) = p_2 - p_1$. Any point p_1 on s_1 that has a corresponding point on s_2 also by definition has a displacement vector.

The Interpolation Step: This is the step in which we find corresponding points for all the points on p_1 that do not have a symmetric nearest neighbor. This is done in an iterative fashion. At each iteration, for each point p_1 on surface s_1 that does not have a displacement vector, we average all the displacement vectors of its first-order neighbors (that have a displacement) to generate a displacement vector u_1. If none of the first-order neighbors has a displacement vector we go on to the next point. We then translate p_1 by u_1 to get $\hat{p}_1 = p_1 + u_1$. We then find point p_2 on s_2 which is the closest point on s_2 to $\hat{p}_1$, as shown in Fig. 6. p_2 is the corresponding point of p_1 on s_2 and we define the displacement of p_1, $u(p_1) = p_2 - p_1$. When the iteration over all the points on s_1 is done, we check whether all the points on s_1 have a displacement. If they do, the process terminates, otherwise we execute another iteration.

So long as one point on s_1 has a symmetric nearest neighbor after step 1 this algorithm will generate a set of point pairs. This algorithm is illustrated in Fig. 7.

Smoothing: This is an alternating iterative process and consists of smoothing and mapping steps. Consider a surface s_1. During the *smoothing* step, for all points p_1 on s_1, we find the average displacement vector u_n of all its first-order neighbors. [These would be $u(a)$, $u(b)$, $u(c)$, $u(d)$, $u(e)$ and $u(f)$ of Fig. 6.] We then generate a new displacement vector $u(p_1) \mapsto 0.75u(p_1) + 0.25u_n$. For the *mapping* step, then we translate p_1 by $\hat{u}$ to a point $\hat{p}_1$. We then find the nearest neighbor of point $\hat{p}_1$

---

[1]Note that we in effect place the corresponding point of $c_1(0.5)$ at the centroid of the (shortest) segment of the curve c_2 connecting the corresponding points of its neighbors [$c_2(0.4)$ and $c_2(0.9)$]. This generalization will become useful when we move to 3-D.

A: Step 1   **B: Step 2 (1)**

C: Step 2 (2)   **D: Step 3(2)**

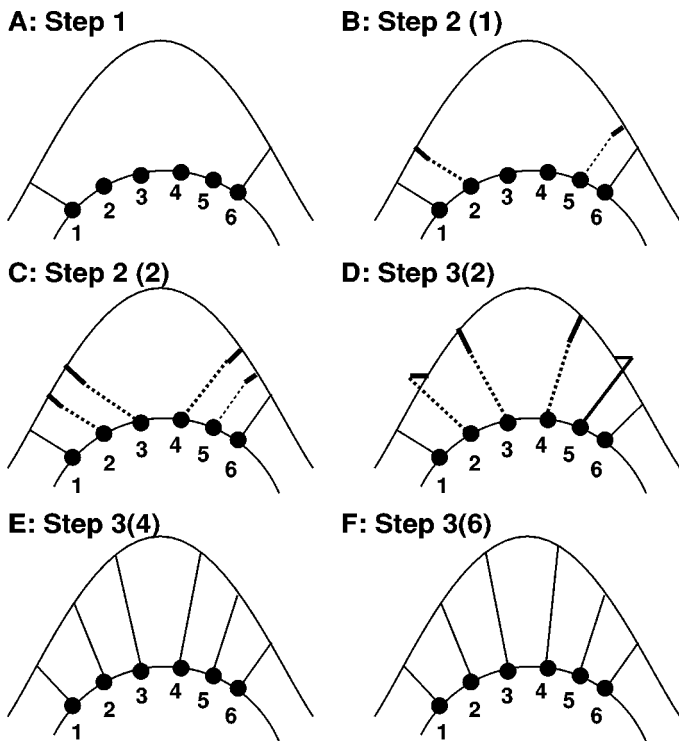**E: Step 3(4)**   **F: Step 3(6)**

Fig. 7. Symmetric 3-D Nearest Neighbor Algorithm. (This is shown in 2-D for simplicity.) (A) Result of step 1, where only points 1 and 6 have corresponding points. (B) (Step 2 iteration 1) Points 2 and 5 also acquire displacements since at least one of their neighbors has a displacement (points 1 and 6, respectively). Note that the displacement vectors of points 2 and 5 have two parts. The first, shown using a dotted line, is the average of the displacements of the neighbors, and the second, shown using a solid line, is the result of mapping this position to the next surface. (C) (Step 2 iteration 2) Points 3 and 4 also have displacements. (D)–(F) Iterations of the smoothing algorithm. Note how the map becomes progressively more regular.

on s_2 and we label this point as p_2. Next, we calculate the displacement vector $u(p_1) = p_2 - \hat{p}_1$. p_2 is also the corresponding point of p_1 on s_2.

C. Shape-Based Matching

The shape-based tracking algorithm tries to follow points on successive surfaces using a shape similarity metric. This distance is based on the difference in principal curvatures. The method was validated using implanted markers [45]. In this paper, we modify the initialization step of this algorithm to take advantage of the symmetric nearest neighbor correspondence finding algorithm previously described in Section III-B.

The first step in this algorithm is to estimate for all points on surface s_1 their symmetric nearest neighbor, as explained in Section III-B. Next, for any given point p_1 on a surface s_1 at time t_1 and which has a corresponding point p_2 on surface s_2 at time t_2 as a result of the symmetric nearest neighbor estimation step we construct a plausible search window W on s_2. This search window W consists of all the points on s_2 which have a connectivity distance less than a threshold t from p_2 on s_2, i.e., $p_w \in W$ iff $d_c(p_2, p_w) < t$.

Next, a search is performed within this plausible region W on the deformed surface s_2 and the point $\hat{p}_2$ which has the local shape properties closest to those of p_1 is selected. The shape properties here are captured in terms of the principal curvatures
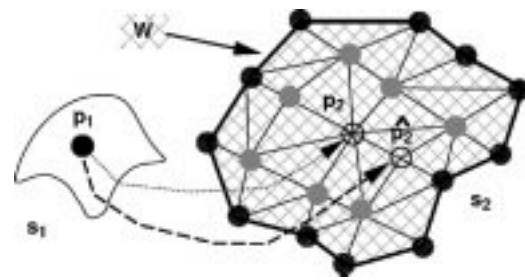

Fig. 8. The shape-tracking algorithm. For a point p_1 on the original surface, a window W of plausible matching points on the final surface is first generated around point p_2 which is the symmetric nearest neighbor of p_1 on the deformed surface. [In this case $\forall p_w \in W$, $d_c(p_2, p_w) < 3$]. Then, the point $\hat{p}_2$ in W which has the most similar shape-properties to p_1 is selected as the candidate match point. The distance function for shape-similarity is based on the principal curvatures.

κ_1 and κ_2. This is illustrated in Fig. 8. The distance measure used is the bending energy required to bend a curved plate or surface patch to a newly deformed state. This is labeled as d_{be} and is defined as (see Shi [45])

$$d_{be}(p1, p2) = \left(\frac{(\kappa_1(p_1) - \kappa_1(p_2))^2 + (\kappa_2(p_1) - \kappa_2(p_2))^2}{2} \right). \tag{1}$$

The displacement estimate vector for each point p_1, u_1^m is given by

$$u_1^m = \hat{p}_2 - p_1, \qquad \hat{p}_2 = \arg\min_{p_2 \in W} [d_{be}(p1, p2)].$$

The underlying assumption in this algorithm is that the curvature at a point does not change much between two time frames. Obviously, strictly speaking, this is not a valid assumption as the left ventricle is a nonrigidly deforming object, but since the change in curvature between any two image time frames is small, this is a reasonable assumption to make.

Probabilistic Modeling of the Shape-Based Displacement Estimates: The bending energy measures for all the points inside the search region W are recorded as the basis to measure the *goodness* and *uniqueness* of the matching choices. These measures are combined to generate a confidence measure in the local match c which takes a high value when a match is both very good and unique and low values otherwise. This is described in more detail in [46].

Given a set of displacement vector measurements u^m and confidence measures c^m we model these estimates probabilistically by assuming that the noise in the individual measurements is normally distributed with zero mean and a variance $\sigma^2 = 1/c^m$. In addition, we assume that the measurements are uncorrelated. Given these assumptions we can write the measurement probability for each point as

$$p(u^m|u) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(-(u-u^m)^2)/2\sigma^2}. \tag{2}$$

IV. MESH GENERATION

We proceed to describe the mesh-generation method used for generating a volumetric model for the left ventricle, in terms of hexahedral elements. The output mesh of this algorithm will
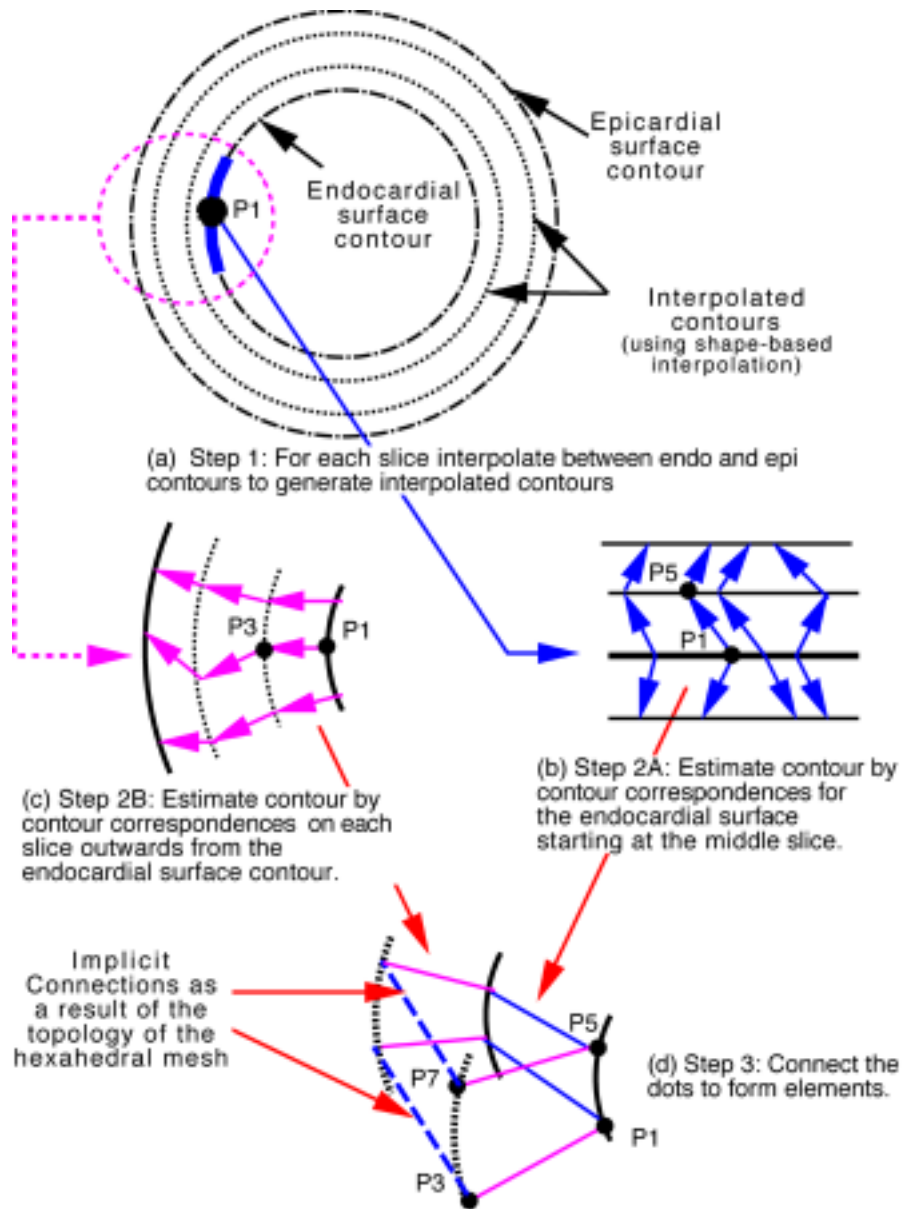
Fig. 9. A schematic of the mesh generation process. First, we interpolate between the endocardial and epicardial surfaces on a contour by contour basis using shape-based interpolation to create the interpolated surfaces. Next, we find correspondences between the contours on the endocardial surface starting at the middle level using the 2-D algorithm described in Section III-A. Next, we find correspondences on each slice starting from the endocardium, using the same algorithm. Finally, we connect the dots to generate the elements.

be used to describe the geometry of the left ventricle as needed for the estimation of the complete deformation field using finite elements.

Mesh generation in three dimensions is a notoriously difficult problem for complicated geometries [6]. Here, we describe an algorithm that takes advantage of the "cylinder-like" geometry of the left ventricle to make the problem easier. The two basic building blocks of the algorithm are the shape-based contour interpolation method of Section II-A and the symmetric nearest neighbor correspondence algorithm described in Section III-A. The algorithm is best described with reference to Fig. 9. It consists of four steps as follows.

Step 1. Interpolate on a contour by contour basis between the endocardial and epicardial surfaces using shape-based interpolation to generate an appro-

priate number of in-between interpolated surfaces (typically three or four). Because of the greater geometrical complexity of the endocardium, we space the interpolated surfaces to be preferentially closer to the endocardium. We then discretize the contour on the middle slice of the endocardium to the desired number of nodes (typically 35–45).

Step 2A. Using the symmetric nearest neighbor algorithm, estimate correspondences between slices on the endocardial surface on a contour-by-contour basis starting in the middle slice. This generates a grid of connected points on the endocardium. These correspondences are shown in blue in Fig. 9(b).

Step 2B. For the points present in the correspondence maps of step 2A, find their correspondences within each

slice starting at the endocardium and moving one level at a time toward the epicardium. This generates a grid of connected points on each slice. These correspondences are shown in purple in Fig. 9(c).

Step 3. We complete the mesh by taking advantage of the grid-like topology of the hexahedral mesh. Consider the following example with reference to Fig. 9(d). A point P1 on slice S10 is mapped to point P5 on slice S11 on the endocardial surface (step 2A), and point P3 on slice S10 on the first midwall surface (step 2B). Further, point P5 on slice S11 on the endocardium corresponds to point P7 on slice S11 of the first midwall surface (step 2B). By virtue of the topology of the mesh, P3 also has to connect to P7. This completes the quadrilateral which forms one face of the element.

V. THE MECHANICAL MODEL

Having described the generation of the image-derived displacements using the shape-based algorithm, we now turn our attention to describing the model used to smooth and interpolate these displacements. We first describe some fundamental concepts in continuum mechanics [48] and next we present the actual model used in this paper.

A. Deformations

Consider a body $B(0)$ which after time t moves and deforms to body $B(t)$. A material particle initially located at some position X on $B(0)$ moves to a new position x on $B(t)$. If we further assume that material cannot appear or disappear there will be an one-to-one correspondence between X and x, so we can always write the path of the particle as: $x = x(X, t)$. We can also define the displacement vector for this particle as $u(t) = x(t) - X$. This relationship is also invertible. Given x and t, we can find X. We consider two neighboring particles located at X and $X + dX$ on $B(0)$. At a new configuration $B(t)$ we can write

$$dx = \frac{\partial x}{\partial X} \, dX. \tag{3}$$

The Jacobian matrix $F(t) = \partial x(t)/\partial X$ is called *the deformation gradient matrix*. We note that by definition $F(0) = I$.

The mapping defined by (3) has two components: a rigid motion component and a change in the shape or deformation of the object. For the purposes of capturing the material behavior, we need to extract from F the component which is a function of the rigid motion and the component which is a function of the deformation.

Small Deformations and Rotations: Since the deformations and the rotations in any one single time-frame interval are small ($<5\%$) we use here the following approximation [48]. $\partial u/\partial x \approx \partial u/\partial X$. Next, we express F as

$$F = RU = (I + \omega)(I + \epsilon). \tag{4}$$

Here, ω is the small rotation tensor and is antisymmetric. ϵ is the small (infinitesimal) strain tensor and is symmetric. These are defined as

$$\omega_{i,j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$$

$$\epsilon_{i,j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{5}$$

Often, taking advantage of the symmetries, these tensors are written in vector form as

$$e = [\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{12}, \epsilon_{13}, \epsilon_{23}]^t$$
$$\theta = [0, 0, 0, \omega_{12}, \omega_{13}, \omega_{23}]^t.$$

This e is the classical definition for strain in infinitesimal linear elasticity [48].

Some Further Properties of the Strain Tensor: Given a strain tensor ϵ_x (a 3×3 matrix) which was computed in a coordinate frame x parameterized by three unit vectors x_1, x_2, x_3 we can transform it to a coordinate frame ξ similarly parameterized by unit vectors ξ_1, ξ_2, ξ_3 as follows. First, construct the 3×3 rotation matrix R. Each component of R, R_{ij} is given by the dot product of x_i and ξ_j, i.e., $R_{ij} = \langle x_i, \xi_j \rangle$. This results in R: $x \mapsto \xi$. Using this matrix R we can write the image of ϵ_x in the ξ coordinate frame ϵ_ξ as: $\epsilon_\xi = R\epsilon_x R^t$.

We also note that the eigenvalues of ϵ are known as the principal strains and the eigenvectors as the principal directions. We use the principal strains in Section VII.

B. Material Models

So far, we have restricted our description to the geometry of the deformation. In this section, we extend this to account for what happens when a material deforms and relate the deformation to the change in the internal structure of the material. Before proceeding to give examples of possible material models we first note that there are some theoretical guidelines which must be observed [13]. The most important ones for this work are: 1) *The axiom of objectivity*—this requires the material model to be invariant with respect to rigid motion or the spatial frame of reference; 2) *The axiom of material invariance*—this implies certain symmetry conditions depending on the type of anisotropy of the material, and implicitly reduces the number of free parameters.

The first axiom can be satisfied by postulating an internal or strain energy function which depends on the gradient deformation matrix F only through the strain tensor E. One way of satisfying this axiom is to model the material using a strain energy function.

Linear Elastic Strain Energy Functions: The simplest useful continuum mechanics model in solid mechanics uses a linear elastic strain energy function W, which takes the form

$$W = \tfrac{1}{2}e^t C e \tag{6}$$

where C is a 6×6 matrix and defines the material properties of the deforming body,[2] as it relates the change in geometry (strain) to the internal energy function W. The simplest model is the isotropic linear elastic model used widely in the image analysis literature [18], [12].

In this paper, the left ventricle of the heart is specifically modeled as a transversely elastic material to account for the preferential stiffness in the fiber direction. This is an extension of the isotropic linear elastic model which allows for one of the three material axes to have a different stiffness from the other two. In this case, the matrix C takes the form

$$C^{-1} = \begin{bmatrix} \dfrac{1}{E_p} & \dfrac{-\nu_p}{E_p} & \dfrac{-\nu_{fp}}{E_f} & 0 & 0 & 0 \\[2mm] \dfrac{-\nu_p}{E_p} & \dfrac{1}{E_p} & \dfrac{-\nu_{fp}}{E_f} & 0 & 0 & 0 \\[2mm] \dfrac{-\nu_{fp}E_f}{E_p} & \dfrac{-\nu_{fp}E_f}{E_p} & \dfrac{1}{E_f} & 0 & 0 & 0 \\[2mm] 0 & 0 & 0 & \dfrac{2(1+\nu_p)}{E_p} & 0 & 0 \\[2mm] 0 & 0 & 0 & 0 & \dfrac{1}{G_f} & 0 \\[2mm] 0 & 0 & 0 & 0 & 0 & \dfrac{1}{G_f} \end{bmatrix} \tag{7}$$

where E_f is the fiber stiffness, E_p is cross-fiber stiffness and ν_{fp}, ν_p are the corresponding Poisson's ratios and G_f is the shear modulus across fibers. [$G_f \approx E_f/(2(1+\nu_{fp}))$.] If $E_f = E_p$ and $\nu_p = \nu_{fp}$ this model reduces to the more common isotropic linear elastic model. The fiber stiffness was set to be 3.5 times greater than the cross-fiber stiffness [16]. The fiber directions used are from [16] and are shown in Fig. 10.

A Probabilistic Formulation of the Mechanical Model: As originally demonstrated by Geman and Geman [15] and previously applied to medical image analysis problems (e.g., [8] and [14]), there is a correspondence between an internal energy function and a Gibbs probability density function. Given an energy function $W(u)$ we can write an equivalent prior probability density function $p(u)$ of the Gibbs form [15] as

$$p(u) = k_1 e^{(-W)} \tag{8}$$

where k_1 is a normalization constant.

C. Limitations of the Transversely Elastic Linear Model

Linear models do not capture the progressive hardening of many materials (especially soft tissue) when it is stretched. In

[2]This class of model is linear as it results in a linear stress-strain relationship, i.e., $\sigma = Ce$. We do not use stresses in this work so we will not express material models explicitly in terms of their stress-strain relationships. Moreover, we deliberately avoid the terms "force," "stress," and "equilibrium." These would be inappropriate since the problem we are trying to solve has no real forces as such. The use of the word "forces" in related work such as Terzopoulos [51] in the context of physics-based vision may have been appropriate since the authors were not trying in any way to use real physics in their methods. In this paper, since we are using *real* mechanical models to model *real* tissue properties, we would only use words such as force to describe *real* forces.
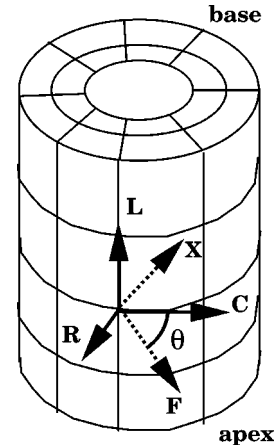

Fig. 10. Coordinate system used to define fiber orientation. The fiber direction (F) lies in the plane defined by the circumferential (C) and longitudinal (L) axes at an angle θ counter-clockwise from the circumferential axis. The fibers are assumed to lie in the plane defined by the local circumferential (C) and longitudinal (L) axes. In the undeformed state, the radial (R) axis points outwards, the circumferential axis (C) is along the circumference of a planar section and the longitudinal axis (L) is vertical. The fiber (F) and cross-fiber axis (X) lie in the plane defined by C and L. The fiber orientation can then be defined by the angle θ as shown in the diagram. The epicardial fiber angle varied between $-43°$ at the base and $-53°$ at the apex, and the endocardial fiber angle varied between $82°$ at the base and $97°$ at the apex. All the other fiber angles can be found by linearly interpolating both along the vertical and the radial directions [16].

the case of linear elastic models, the effective stiffness is a constant with respect to the strain whereas in practice the stiffness increases as the strain increases. Moreover, in the work of Guccione [16] the ratio of fiber-to-transverse stiffness varies with the deformation. We selected a value of 3.5 as a reasonable value for the range of deformation that is observed.

The use of complex nonlinear models is essential if the model by itself tries to capture the whole complex motion of the left ventricle. In this paper, the mechanical model is used as a filter in the signal processing sense to smooth and interpolate displacements in a meaningful way, hence, a simpler model of the type used here produces reasonable results.

VI. ESTIMATING THE COMPLETE DEFORMATION FIELD

Having described the initial displacement estimation process (Section III) and the model used for the myocardium (Section V), we now describe the overall framework for integrating these two forms of available information. We note that this framework is general and can be applied to other forms of soft tissue deformation estimation processes where one is trying to combine image-derived displacements with biomechanical models.

We pose the problem of estimating the complete deformation field as a Bayesian estimation [38] problem as follows: Given 1) a set of initial, noise-corrupted displacements u^m derived from the shape-tracking algorithm, with an associated measurement probability density function $p(u^m|u)$ [see (2)], and 2) a prior probability density function for the true displacement field u, $p(u)$ derived from the mechanical model [see (8)], estimate the optimal value of the displacement field u as the one which maximizes the posterior probability density function $p(u|u^m)$.

Using Bayes' rule we can write the posterior probability at each point as

$$\hat{u} = \arg\max_u \left\{ p(u|u^m) = \frac{p(u, u^m)}{p(u^m)} = \frac{p(u^m|u)p(u)}{p(u^m)} \right\}. \quad (9)$$

First, we note that $p(u^m)$ is a constant once the measurements have been made and can therefore be ignored in the maximization process. We can re-write (9) by taking logarithms to arrive at

$$\hat{u} = \arg\max_u \left(\log p(u) + \log p(u^m|u) \right). \quad (10)$$

Using the volumetric model defined by the mesh generation algorithm (see Section IV), we then proceed to write a finite element formulation [7] for the problem over the whole of the left ventricle. First, we concatenate all the individual displacements at the nodes of the mesh into a long vector U. Then, we can express the model term $\log p(u)$ as the matrix triple product $W = U^t K U$ where K is the global stiffness matrix. Then, we concatenate all the measurements u^m into another long vector U^m and express the data term $\log p(u^m|u)$ as $P = (U^m - U)^t \Sigma^{-1} (U^m - U)$ where Σ^{-1} is the inverse covariance matrix.[3]

Then, we can rewrite (10) in the final form

$$\hat{U} = \arg\min_U \left(U^t K U + (U^m - U)^t \Sigma^{-1} (U^m - U) \right)$$

which when differentiated with respect to U yields the final solution equation

$$\left(K + \Sigma^{-1} \right) U = \Sigma^{-1} U^m. \quad (11)$$

A. The Problem of Unit Reconciliation

There is one fundamental problem with the above framework. This is the problem of "unit reconciliation." This problem arises because the model stiffness is measured in different units from the noise variance, which results in the numbers in the stiffness matrix K having different units from the numbers in the covariance matrix Σ.

To illustrate the effects of this problem, we can rewrite both of these matrices in this general form (using the $n \times n$ matrix M to be either C or Σ^{-1}) as

$$M = \begin{bmatrix} M_{11} & \cdots & M_{1n} \\ \cdots & & \cdots \\ M_{n1} & \cdots & M_{nn} \end{bmatrix} = M_{\max} \left[\overline{M} \right]$$

$$\left[\overline{M} \right] = \begin{bmatrix} \dfrac{M_{11}}{M_{\max}} & \cdots & \dfrac{M_{1n}}{M_{\max}} \\ \cdots & & \cdots \\ \dfrac{M_{n1}}{M_{\max}} & \cdots & \dfrac{M_{nn}}{M_{\max}} \end{bmatrix} \quad (12)$$

where $M_{\max}$ is the maximum value of M. In the case of the stiffness matrix K, $K_{\max}$ would the highest value of the stiffness matrix and would be proportional to the Young's modulus, whereas in the case of the covariance matrix Σ^{-1}, $\Sigma_{\max}^{-1}$ would be the smallest variance, or the highest confidence in any of the

measurements. Note also that the numbers in the matrices $\left[\overline{K} \right]$ and $\left[\overline{\Sigma^{-1}} \right]$ are dimensionless.

We can now rewrite (11) as

$$\left(K_{\max} \left[\overline{K} \right] + \Sigma_{\max}^{-1} \left[\overline{\Sigma^{-1}} \right] \right) U = \Sigma_{\max}^{-1} \left[\overline{\Sigma^{-1}} \right] U^m.$$

Dividing through by $\Sigma_{\max}^{-1}$, we obtain

$$\frac{K_{\max}}{\Sigma_{\max}^{-1}} \underbrace{\left(\left[\overline{K} \right] + \left[\overline{\Sigma^{-1}} \right] \right)}_{\text{Dimensionless}} U = \underbrace{\left[\overline{\Sigma^{-1}} \right]}_{\text{Dimensionless}} U^m. \quad (13)$$

At this point, it is clear that the absolute values of $K_{\max}$ and $\Sigma_{\max}^{-1}$ enter into the functional only through their ratio $K_{\max}/\Sigma_{\max}^{-1}$. Given that the rest of the expressions in (13) are dimensionless[4] to reconcile (13) in terms of dimensionality we need to convert this ratio $K_{\max}/\Sigma_{\max}^{-1}$ in order to also make it dimensionless. This is done by multiplying by a scaling constant k_{sc} of the appropriate units, i.e.,

$$\frac{K_{\max}}{\Sigma_{\max}^{-1}} \mapsto \frac{K_{\max}}{k_{sc} \Sigma_{\max}^{-1}}. \quad (14)$$

From a dimensionality viewpoint, the value of the scaling constant k_{sc} is completely arbitrary. This value can be interpreted as defining in some sense the ratio of the relative confidences in the model *as a whole* and the data *as a whole*. While this is a common problem in many regularization problems such as Horn and Schunk [21], it is especially important to note it clearly in this context where one is trying to estimate deformation of real deformable objects using a mechanical model. The implications of this inconsistency in the units is that the material properties of the solid can be used to set the all the values of the regularization functional (model) up to a scaling constant which is arbitrary. In some previous work in this area, the authors specify the absolute value used for the Young's modulus for the left ventricle, which is meaningless as a result of this problem.

B. The Bias Problem

The estimation framework described so far produces a biased estimate of the deformation. The easiest way to see this is to observe that, since the elastic model penalizes all deformations, any estimation framework which uses it as a prior model or internal energy model as defined in (11) will underestimate the actual deformation. The linear elastic model can be thought of as a prior probability density function on the strain with zero mean and variance proportional to the reciprocal of the Young's modulus. To illustrate this, we rewrite (11) as

$$U = \left(\left(K + \Sigma^{-1} \right)^{-1} \Sigma^{-1} \right) U^m.$$

Taking expectations on both sides gives

$$\mathcal{E}(U) = \left(\left(K + \Sigma^{-1} \right)^{-1} \Sigma^{-1} \right) \mathcal{E}(U^m). \quad (15)$$

Note that for as long as K is nonzero, the expected value of U will be smaller than the expected value of U^m, hence, the deformation will be underestimated. This is a problem in most methods that estimate cardiac deformation—the possible

---

[3]In this case, a diagonal matrix with values c_m on the leading diagonal defined in (2) where there are measurements available and 0 where there are no measurements.

[4]The term "dimensionless" is used to describe a quantity that is a real number with no associated units. A dimensionless quantity will have the same value regardless of the system of units used in its calculation. For example, the ratio of two lengths will the same regardless of whether the lengths are measured in meters or in feet.

exception being those methods that essentially believe the data once, for example, the tag-lines have been extracted (such as [23]).

A number of methods have been proposed to implicitly deal with this problem (see [34] for details). None of these, however, have dealt with the cause of the problem; they are rather, in a sense, trying to limit its effects with varying degrees of success. In this paper, we correct for part of the bias by 1) solving for the cardiac deformation in a frame-by-frame manner, thus keeping the deformations to be estimated small and, hence, closer to zero; 2) by ensuring that points that lie on the outer surfaces of the myocardium at frame t still lie on the outer surface at time $t+1$. This second step eliminates any bias in the directions perpendicular to each outer surface.

C. Numerical Solution

The overall framework described by (11) is assembled and solved using the finite element method [7]. The first step in the finite element method is the division or tessellation of the body of interest into elements; these are commonly tetrahedral or hexahedral in shape. In this paper, we use hexahedral elements generated using a custom mesh-generation algorithm, described in Section IV. In our case, the myocardium is divided into approximately 1500 hexahedral elements.

For each frame between end-systole (ES) and end-diastole (ED), a two step problem is posed: 1) solving (11) normally; 2) adjusting the position of all points on the endocardial and epicardial surfaces so they lie on the endocardial and epicardial surfaces at the next frame using the symmetric nearest-neighbor technique described in Section III-B and solving (11) once more using this added constraint. This ensures that there is a reduction in the bias in the estimation of the deformation.

The value of the weighting constant k_{sc} is set adaptively to be as large as possible (which pushes the optimum toward the data side) subject to solution convergence. In this way, we make the following assumption: the best solution is the one which adheres as much as possible to the initial estimate of the displacement field but still results in a connected solid. Convergence fails when the Jacobian of the deformation field becomes singular.[5] In this case, we decrease the value of k_{sc} to produce a smoother displacement field.

VII. EXPERIMENTAL RESULTS

A. Surgical Preparation/Experimental Protocol

Experiments were performed on fasting adult mongrel dogs with approval of the Yale Animal Care and Use Committee, in compliance with the guiding principles of the American Physiological Society on research animal use. All dogs were anesthetized with 10–12 mg/kg thiopental sodium (Pentothal Abbott, North Chicago, IL) intravenously, intubated and mechanically ventilated on a respirator with a mixture of halothane (0.5%–1.5%), and nitrous oxide and oxygen ($N_2O : O_2 = 3 : 1$).

A femoral vein and both femoral arteries were isolated and cannulated for administration of fluids and drugs, pressure monitoring and arterial sampling. A thoracotomy was performed in

---

[5]For example, when the path of two points on the mesh intersect as a result of a locally bad shape-based displacement estimate.
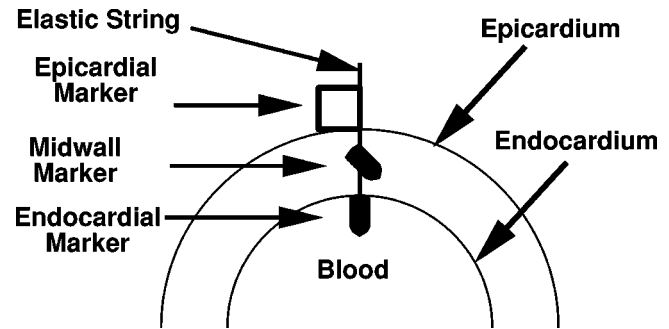

Fig. 11.   Implantation of Image-Opaque Markers. This figure shows the arrangement of markers on the myocardium. First, a small bullet-shaped copper bead attached to an elastic string was inserted into the blood pool through a needle track. Then, the epicardial marker was sutured (stitched) to the myocardium and tied to the elastic string. Finally, the midwall marker was inserted obliquely through a second needle track to a position approximately half-way between the other two markers.

the fifth intercostal space and the heart suspended on a pericardial cradle. MR image opaque markers were implanted for validation of our MR analysis approach, using a marker system previously described by our group [46]. In brief, cubic arrays of three-marker sets were carefully placed in the mid anterior and posterior walls avoiding epicardial surface vessels. First, a small bullet-shaped copper marker attached to an elastic string was inserted through the myocardium via a previously created needle track with the aid of a metal introducer. A specially designed gadolinium-filled capsule was then sutured to the epicardial surface directly above each of the endocardial markers, as illustrated in Fig. 11. The elastic string was withdrawn so the endocardial marker was touching the endocardial surface and fixed to the epicardial capsule with a suture, providing endocardial-epicardial marker pairs. A second copper marker was inserted obliquely so that it would be positioned between each endocardial-epicardial marker pair. The proximal left anterior descending coronary artery was isolated for placement of a snare occluder. This occluder was externalized and the chest closed in layers. After completion of surgical preparation, dogs were positioned in the MR scanner for imaging. An electrocardiogram limb lead was monitored continuously during MRI and used for gating. Resting MR images were completed in one hour. Heart rate (HR) and aortic pressure (AoP) were recorded immediately before and after each complete image acquisition. Dogs subsequently underwent repeat MR imaging following coronary occlusion. Dogs were euthanized with a bolus of potassium chloride after completing all imaging.

B. MR Image Acquisition

MR imaging was performed on a GE Signa 1.5 Tesla scanner with version 4.8 software using the head coil (26 cm diameter) for transmission and reception. Short-axis images through the left ventricle were obtained with the gradient echo cine technique using the following parameters: TE $= 6$ ms, TR $= 40$ ms, flip angle $= 30°$, 16 phases collected, 5-mm slices, matrix 256 $\times$ 256, two averages, field of view $= 40$ cm. A total of 16 contiguous 5-mm-thick slices were collected, by acquiring four sets of staggered short axis slices (four slices/set) with a separation gap of 20-mm and 5-mm offset. This sequence provides images
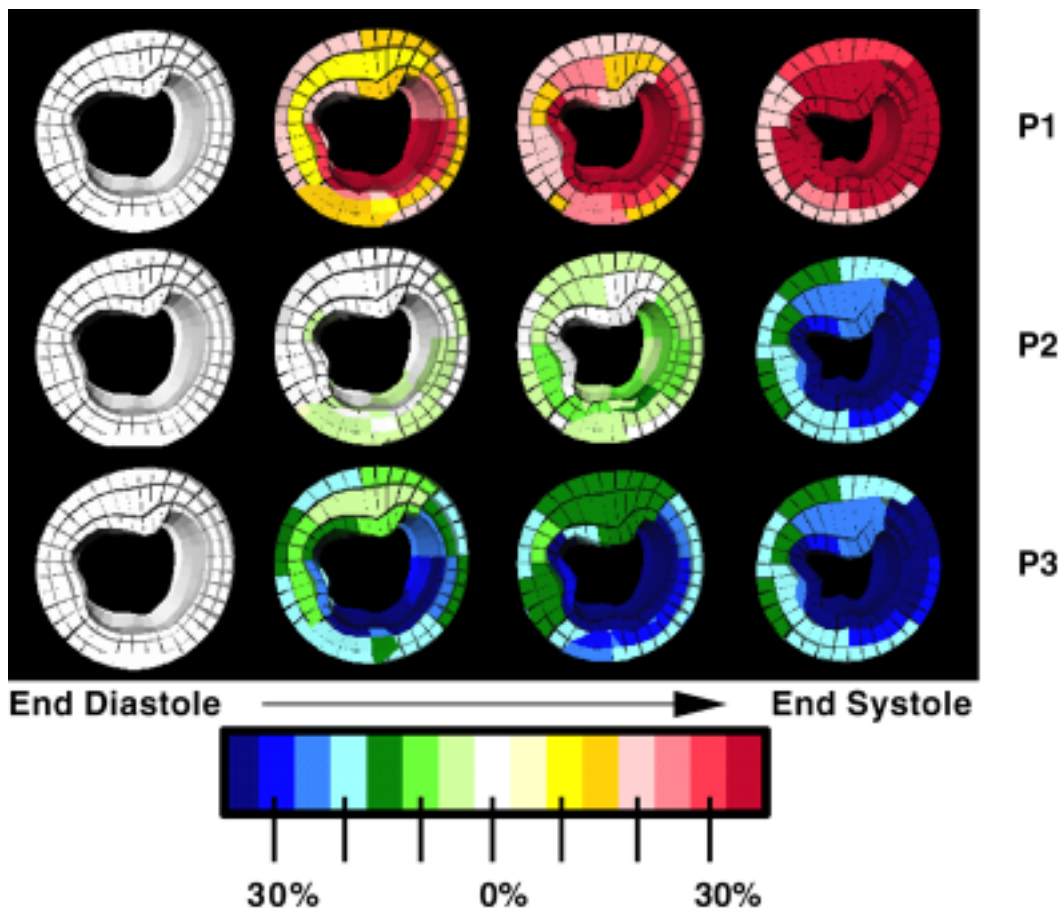
Fig. 12. Canine cine-MRI-derived principal strains using algorithm *f-3*. The horizontal axis represents time from ED to ES (every other frame shown). We display the first (P1) second (P2), and third (P3) principal strains on each of the three rows. The septum is on the left and the anterior wall on the bottom. The markers were implanted in two arrays: one in the anterior wall and another in the lateral wall.

with an in-plane resolution of 1.64 mm × 1.64 mm for a 256 × 256 matrix and a 5-mm resolution perpendicular to the imaging plane. This sequence also provides excellent temporal resolution (16 frames/cardiac cycle, ≈40 ms/frame).

A total of four dogs completed the MR imaging protocol. Hemodynamic parameters and cardiac rhythm remained stable during the MR image acquisition. The HR and systolic AoP immediately before the MR acquisition were not significantly different from that obtained at the completion of the acquisition. All image analyses were performed on all dogs.

C. Strain Computation Using Implanted Markers

The location of each implanted marker is determined in each temporal frame by first manually identifying all pixels which belong to the marker area (because of imaging artifacts the marker "image" extends to more than one voxel) and then computing the 3-D centroid of that cluster of points, weighted by the grey level.[6] This procedure was performed on a total of four animals and both sets of images (baseline and postocclusion). Mid-wall markers were not used since it was difficult to identify them correctly from the images.

Once the positions of the markers were determined, they were used to compute the displacement at each marker between ED and ES. Further, in each of the two regions of the LV where the

[6] In the case of dark markers, the image is first inverted.

TABLE I
PARAMETER SETTINGS

Name	Material Model	Poisson's ratio
f-1	Transversely Isotropic (using fibers)	0.325
f-2	Transversely Isotropic (using fibers)	0.400
f-3	Transversely Isotropic (using fibers)	0.475
i-1	Isotropic (without fiber model)	0.325
i-2	Isotropic (without fiber model)	0.400
i-3	Isotropic (without fiber model)	0.475

Definition of the six different parameter settings used for both the sensitivity analysis and the validation study with implanted markers.

markers were implanted, groups of either six or eight markers (depending on the local geometry) were connected to form either prism or hexahedral elements. Given the computed displacements, we then calculated the strains in these marker regions. In particular, we computed the principal strains at the centroid of each marker array. We labeled the first, second, and third principal strains as P1, P2, and P3, respectively.

D. Sensitivity Analysis

We compare the strains obtained using the implanted markers to the strains computed using our algorithm. In particular, we used six different parameter settings for the algorithm as summarized in Table I, resulting in six different versions of our al-

TABLE II
ACTUAL STRAIN VALUES

	Markers	f-1	f-2	f-3	i-1	i-2	i-3
P1							
μ	27.0	36.0	36.2	39.8	34.4	35.8	38.8
σ	12.4	15.1	14.2	14.8	15.1	15.1	14.9
P2							
μ	-10.4	-6.6	-7.4	-8.3	-6.2	-6.5	-7.3
σ	4.4	1.7	1.5	2.3	1.3	1.7	2.3
P3							
μ	-30.8	-25.8	-26.1	-30.0	-25.8	-27.2	-29.8
σ	6.2	3.9	3.5	4.0	3.9	3.7	5.1

Means (μ) and standard deviations (σ) for both the marker-calculated and the algorithm-derived % strains, in the baseline (preocclusion) studies.

TABLE III
SENSITIVITY ANALAYSIS

P1	f-1	f-2	f-3	i-1	i-2	i-3
f-1	0.0	1.1	4.8	1.6	0.9	4.8
f-2		0.0	4.4	1.9	1.1	4.7
f-3			0.0	6.4	4.7	1.4
i-1				0.0	1.7	6.4
i-2					0.0	4.7
i-3						0.0

P2	f-1	f-2	f-3	i-1	i-2	i-3
f-1	0.0	0.7	2.1	1.2	0.9	1.5
f-2		0.0	1.5	1.7	1.3	1.4
f-3			0.0	2.9	2.3	1.3
i-1				0.0	0.9	2.0
i-2					0.0	1.4
i-3						0.0

P3	f-1	f-2	f-3	i-1	i-2	i-3
f-1	0.0	0.9	4.8	1.3	2.0	5.6
f-2		0.0	4.2	1.5	1.4	5.0
f-3			0.0	4.8	3.5	2.1
i-1				0.0	1.3	5.4
i-2					0.0	4.1
i-3						0.0

Average absolute percentage strain differences for the three principal strains [First, (**P1**), Second, (**P2**), and Third (**P3**)] between outputs of the algorithm with six different parameter settings as defined in Table I. Note that the variations are small even when drastic parameter changes are involved.

gorithm. In three of these versions, we used the transversely isotropic elastic model described by (7) with three different settings of the Poisson's ratio (0.325, 0.4, 0.475), and in the other three versions we used an isotropic elastic model [obtained by setting $E_f = E_p$ in (7)] and similarly varied the Poisson's ratio. An example of the principal strains derived on a baseline heart using algorithm *f-3* is shown in Fig. 12.

The average principal strains obtained in the precoronary occlusion state for all the regions as estimated using the implanted markers as well as our algorithm is tabulated in Table II, in order to give a sense of the magnitude of these strains.

Next, we computed the difference between the outputs of these six different versions of our algorithm in the regions of the implanted marker arrays as a test of the sensitivity of our algorithm to parameter changes. The results are tabulated in Table III. We note that the algorithm is fairly insensitive to

TABLE IV
ACCURACY ANALYSIS

Strain		f-1	f-2	f-3	i-1	i-2	i-3
P1	μ	7.1	6.9	8.2	7.2	7.3	8.4
	σ	6.4	5.8	7.5	5.7	5.9	7.0
	r^2	0.71	0.74	0.77	0.73	0.73	0.78
P2	μ	6.4	6.1	6.8	5.7	6.0	6.3
	σ	4.7	4.7	5.0	4.4	4.7	4.1
	r^2	0.30	0.36	0.21	0.54	0.40	0.41
P3	μ	5.8	5.7	4.5	5.6	5.2	5.2
	σ	4.4	3.7	4.1	4.3	4.0	4.8
	r^2	0.77	0.82	0.83	0.74	0.76	0.72

1) the mean (μ) and standard deviation (σ) of the absolute percentage error between the marker-calculated strains and the strain outputs of six versions of the algorithm, as well as the correlation ratio r^2 between the algorithm generated strains and the marker output.

these changes of settings and also that changing the Poisson's ratio produces a greater change in the strain output than does changing the underlying mechanical model type.

E. Strain Comparison

The image-derived strains were compared with strains derived from implanted markers. These strains were compared with the average image-derived strains in the region of the myocardium contained within each marker array. Comparison results are shown in Table IV for $N = 4$ dogs (two acquisitions per dog, one preocclusion and one postocclusion). We observe a strong correlation of the first and third principal strain values which roughly correspond to the radial and circumferential directions, respectively. The correlation for the second principal strain (roughly corresponding to the longitudinal direction) is lower and could be the result of two factors: 1) The lower image resolution in that direction as a result of using 5-mm-thick slices; 2) Incomplete bias reduction in this direction since it is perpendicular to the epi and endo-cardial surfaces of the ventricles. We further note that the correlation ratio values reported for the third principal strain are close to those reported for validating circumferential shortening derived from MR tagging recently reported in [54]. In particular, they report a correlation ratio of 0.84 for ES.

F. Preliminary Results on Human-MRI and Canine-CT Data

In the future, we plan to apply our algorithm to both human studies and other image modalities. In order to demonstrate the applicability of our algorithm to these cases, we present here some preliminary results obtained on human cine-MRI data (see Fig. 13) and canine cine-CT data (see Fig. 14).

VIII. CONCLUSION

In this paper, we have illustrated the application of our approach to estimating LV deformation from 3-D medical image sequences in both experimental models and human volunteers. The results have been validated *in vivo* using implanted markers.

We note that modality-specific forms of data can be added to this general framework. In the case of magnetic resonance image data, midwall could be derived from MR tagging and/or phase contrast velocities. However, we have tested the method so far using only shape-based displacements as an input.
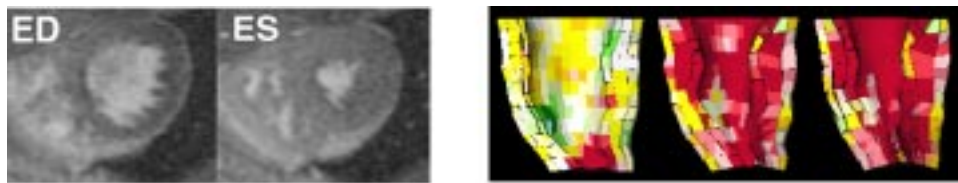
Fig. 13. Human cine-MRI-derived results. Left: Magnitude breath-hold ED and ES images at a single slice level. Right: (see color scale in Fig. 12) radial strains at three long-axis time points between ED and ES. The one difference in the processing of the human cine-MRI data as opposed to the previously presented canine cine-MRI data, was that since, in this case, different 3-D slice levels are acquired at different breath holds, slices at the same time frame can be misaligned along the long axis of the heart. We have corrected for this by manually aligning the data in each frame.

Fig. 14. Algorithm-derived strains from cine-CT dynamic spatial reconstructor (DSR) Images. (left) Example axial slice from baseline dog study at ED and ES. (right) Radial strains at three time points ED to ES. (Again, see Fig. 12 for the colorscale.) The cine-CT canine experiments were performed by Dr. Erik Ritman, at the Mayo Clinic, using the DSR [44]. Note that the values reported are in the same range as strains from our cine-MRI data.

Further research could include the use of an active model [35] to properly handle the bias problems inherent in the passive biomechanical model. An active model could also be used as a means of incorporating a temporal continuity/periodicity constraint.

REFERENCES

[1] A. A. Amini, Y. Chen, R. W. Curwen, V. Manu, and J. Sun, "Coupled B-snake grids and constrained thin-plate splines for analysis of 2-D tissue defomations from tagged MRI," *IEEE Trans. Med. Imag.*, vol. 17, pp. 344–356, June 1998.

[2] A. A. Amini and J. S. Duncan, "Bending and stretching models for LV wall motion analysis from curves and surfaces," *Image Vis. Computing*, vol. 10, no. 6, pp. 418–430, 1992.

[3] A. A. Amini and J. L. Prince, Eds., *Measurement of Cardiac Deformations From MRI: Physical and Mathematical Models*. Norwell, MA: Kluwer Academic, 2001.

[4] L. Axel, "Physics technology of cardiovascular MR imaging," *Cardiol. Clinics*, vol. 16, no. 2, pp. 125–133, 1998.

[5] H. Azhari, J. Weiss, W. Rogers, C. Siu, and E. Shapiro, "A noninvasive comparative study of myocardial strains in ischemic canine hearts using tagged MRI in 3D," *Amer. J. Physiol.*, vol. 268, pp. H1918–1926, 1995.

[6] I. Babuska, J. E. Flaherty, W. D. Henshaw, J. E. Hopcroft, J. E. Oliger, and T. Tezduyar, Eds., *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*. Berlin, Germany: Springer Verlag, 1995.

[7] K. Bathe, *Finite Element Procedures in Engineering Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[8] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "3D brain mapping using deformable neuroanatomy," *Phys. Med. Biol.*, vol. 39, pp. 609–618, 1994.

[9] I. Cohen, N. Ayache, and P. Sulger, *Lecture Notes in Computer Science*. Berlin, Germany: Springer Verlag, 1992, vol. ECCV92, Tracking points on deformable objects using curvature information, pp. 458–466.

[10] T. Constable, K. Rath, A. Sinusas, and J. Gore, "Development and evaluation of tracking algorithms for cardiac wall motion analysis using phase velocity MR imaging," *Magn. Reson. Med.*, vol. 32, pp. 33–42, 1994.

[11] T. S. Denney, Jr. and J. L. Prince, "Reconstruction of 3-D left ventricular motion from planar tagged cardiac MR images: An estimation theoretic approach," *IEEE Trans. Med. Imag.*, vol. 14, pp. 625–635, Dec. 1995.

[12] J. S. Duncan, P. Shi, R. T. Constable, and A. Sinusas, "Physical and geometrical modeling for image-based recovery of left ventricular deformation," *Progr. Biophys. Mol. Biol.*, vol. 69, no. 2–3, pp. 333–351, 1998.

[13] A. C. Eringen, *Mechanics of Continua*. New York: Krieger, 1980.

[14] J. C. Gee, D. R. Haynor, L. Le Briquer, and R. K. Bajcsy, "Advances in elastic matching theory and its implementation," presented at the CVRMed-MRCAS, Grenoble, France, Mar. 1997.

[15] D. Geman and S. Geman, "Stochastic relaxation, Gibbs distribution and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, 1984.

[16] J. M. Guccione and A. D. McCulloch, "Finite element modeling of ventricular mechanics," in *Theory of Heart*, P. J. Hunter, A. McCulloch, and P. Nielsen, Eds. Berlin, Germany: Springer-Verlag, 1991, pp. 122–144.

[17] S. N. Gupta and J. L. Prince, "On variable brightness optical flow for tagged MRI," *Inform. Processing Med. Imag.*, June 1995.

[18] E. Haber, D. N. Metaxas, and L. Axel, "Motion analysis of the right ventricle from MRI images," in *Medical Image Computing and Computer Aided Intervention (MICCAI)*, Cambridge, MA, Oct. 1998, pp. 177–188.

[19] R. Herfkens, N. Pelc, L. Pelc, and J. Sayre, "Right ventricular strain measured by phase contrast MRI," in *Proc. 10th Annu. SMRM*, San Francisco, CA, 1991, p. 163.

[20] G. T. Herman, J. Zheng, and C. A. Bucholtz, "Shape-based interpolation," *IEEE Comput. Graph. Applicat.*, pp. 69–79, May 1992.

[21] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.

[22] C. Kambhamettu and D. Goldof, "Curvature-based approach to point correspondence recovery in conformal nonrigid motion," *CVGIP: Image Understanding*, vol. 60, no. 1, pp. 26–43, July 1994.

[23] W. S. Kerwin and J. L. Prince, "Cardiac material markers from tagged MR images," *Med. Image Anal.*, vol. 2, no. 4, pp. 339–353, 1998.

[24] C. Kramer *et al.*, "Regional differences in function within noninfarcted myocardium during left ventricular remodeling," *Circulation*, vol. 88, pp. 1279–1288, 1993.

[25] C. Kramer, W. Rogers, T. Theobald, T. Power, S. Petruolo, and N. Reichek, "Remote noninfarcted regional dysfunction soon after first anterior myocardial infarction: A magnetic resonance tagging study," *Circulation*, vol. 94, pp. 660–666, 1996.

[26] W. Lorenson and H. Cline, "Marching Cubes: A high resolution 3D surface construction algorithm," in *Proc. SIGGRAPH*, vol. 21, July 1987, pp. 163–169.

[27] G. E. Mailloux, A. Bleau, M. Bertrand, and R. Petitclerc, "Computer analysis of heart motion from two-dimensional echocardiograms," *IEEE Trans. Biomed. Eng.*, vol. BME-34, pp. 356–364, May 1987.

[28] J. T. Marcus, M. Gotte, A. Van Rossum, J. P. A. Kuijer, R. Heethaar, L. Axel, and C. Visser, "Myocardial function in infarcted and remote regions early after infarction in man: Assessment by magnetic resonance tagging and strain analysis," *Magn. Reson. Med.*, vol. 38, pp. 803–810, 1997.

[29] J. McEachen, R. Owen, and J. S. Duncan, "Shape-based tracking of left ventricular wall motion," *IEEE Trans. Med. Imag.*, vol. 16, pp. 270–283, June 1997.

[30] E. R. McVeigh, "Regional myocardial function," *Cardiol. Clinics*, vol. 16, no. 2, pp. 189–206, 1998.

[31] J. Meunier, "Tissue motion assessment from 3D echographic speckle tracking," *Phys. Med. Biol.*, vol. 43, pp. 1241–1254, 1998.

[32] F. G. Meyer, R. T. Constable, A. J. Sinusas, and J. S. Duncan, "Tracking myocardial deformation using phase contrast MR velocity fields: A stochastic approach," *IEEE Trans. Med. Imag.*, vol. 15, Aug. 1996.

[33] G. Nayler, N. Firmin, and D. Longmore, "Blood flow imaging by cine magnetic resonance," *J. Comput. Assist. Tomogr.*, vol. 10, pp. 715–722, 1986.

[34] X. Papademetris, "Estimation of 3D left ventricular deformation from medical images using biomechanical models," Ph.D. dissertation, Yale Univ., New Haven, CT. [Online] Available: http://noodle.med.yale.edu/thesis, May 2000.

[35] X. Papademetris, E. T. Onat, A. J. Sinusas, D. P. Dione, R. T. Constable, and J. S. Duncan, "The active elastic model," in *Inform. Processing Med. Imag.*, Davis, CA, 2001.

[36] X. Papademetris, J. V. Rambo, D. P. Dione, A. J. Sinusas, and J. S. Duncan, "Visually interactive cine-3D segmentation of cardiac MR images," *J. Amer. Coll. Cardiol.*, vol. 31, no. 2 (Suppl. A), Feb. 1998.

[37] X. Papademetris, A. J. Sinusas, D. P. Dione, and J. S. Duncan, "Estimation of 3D left ventricular deformation from echocardiography," *Med. Image Anal.*, vol. 5, no. 1, pp. 17–29, Mar. 2001.

[38] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.

[39] N. Pelc, R. Herfkens, A. Shimakawa, and D. Enzmann, "Phase contrast cine magnetic resonance imaging," *Magn. Reson. Quart.*, vol. 7, no. 4, pp. 229–254, 1991.

[40] N. J. Pelc, "Myocardial motion analysis with phase contrast cine MRI," in *Proc. 10th Annu. SMRM*, San Francisco, CA, 1991, p. 17.

[41] N. J. Pelc, R. Herfkens, and L. Pelc, "3D analysis of myocardial motion and deformation with phase contrast cine MRI," in *Proc. 11th Annu. SMRM*, Berlin, Germany, 1992, p. 18.

[42] J. L. Prince and E. R. McVeigh, "Motion estimation from tagged MR image sequences," *IEEE Trans. Med. Imag.*, vol. 11, pp. 238–249, June 1992.

[43] J. A. Sethian R, J. A. Malladi, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–174, Feb. 1995.

[44] R. A. Robb, "High-speed three-dimensional X-ray computed tomography: The dynamic spatial reconstructor," *Proc. IEEE*, vol. 71, pp. 308–319, 1983.

[45] P. Shi, "Image analysis of 3D cardiac motion using physical and geometrical models," Ph.D. dissertation, Yale Univ., New Haven, CT, May 1996.

[46] P. Shi, A. J. Sinusas, R. T. Constable, E. Ritman, and J. S. Duncan, "Point-tracked quantitative analysis of left ventricular motion from 3-D image sequences," *IEEE Trans. Med. Imag.*, vol. 19, pp. 36–50, Jan. 2000.

[47] S. Song and R. Leahy, "Computation of 3-D velocity fields from 3-D cine CT images," *IEEE Trans. Med. Imag.*, vol. 10, pp. 295–306, Sept. 1991.

[48] A. Spencer, *Continuum Mechanics*. London, U.K.: Longman, 1980.

[49] H. D. Tagare, "Shape-based nonrigid curve correspondence with application to heart motion analysis," *IEEE Trans. Med. Imag.*, vol. 18, pp. 570–579, July 1999.

[50] G. Taubin, "Curve and surface smoothing without shrinkage," in *Proc. 5th Int. Conf. Computer Vision*, 1995, pp. 852–857.

[51] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformation: Deformable superquadrics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 7, July 1991.

[52] P. van Dijk, "Direct cardiac NMR imaging of heart wall and blood flow velocity," *J. Comput. Assist. Tomogr.*, vol. 8, pp. 429–436, 1984.

[53] V. Wedeen, "Magnetic resonance imaging of myocardial kinematics: Technique to detect, localize and quantify the strain rates of active human myocardium," *Magn. Reson. Med.*, vol. 27, pp. 52–67, 1992.

[54] S. B. Yeon, N. Reichek, B. A. Tallant, J. A . C. Lima, L. P. Calhoun, N. R. Clark, E. A. Hoffman, K. L. Ho, and L. Axel, "Validation of *in vivo* myocardial strain measurement by magnetic resonance tagging with sonomicrometry," *J. Amer. Coll. Cardiol.*, vol. 38, no. 2, pp. 555–561, August 2001.

[55] A. A. Young, D. L. Kraitchman, L. Dougherty, and L. Axel, "Tracking and finite element analysis of stripe deformation in magnetic resonance tagging," *IEEE Trans. Med. Imag.*, vol. 14, pp. 413–421, Sept. 1995.

[56] Y. Zhu, M. Drangova, and N. J. Pelc, "Estimation of deformation gradient and strain from cine-PC velocity data," *IEEE Trans. Med. Imag.*, vol. 16, Dec. 1997.

[57] ——, "A spatiotemporal model of cyclic kinematics and its application to analyzing nonrigid motion with mr velocity images," *IEEE Trans. Med. Imag.*, vol. 18, pp. 557–569, July 1999.