LV Shape and Motion: B-Spline Based Deformable Model and Sequential Motion Decomposition

Guo Luo and Pheng Ann Heng

Abstract—In this paper, we extend Park's previous work [1] and propose a uniform framework to reconstruct LV geometry/motion from tagged MR images. In our work, the LV is modeled as a generalized prolate spheroid, and its motion is decomposed into four components—global translation, polar radial/z-axis compression, twisting, and bending. By formulating model parameters as tensor products of B-splines, we develop efficient algorithms to quickly reconstruct LV geometry/motion from extracted boundary contours and tracked planar tags. Experiments on both synthesized and in vivo data are also reported.

Index Terms—left ventricle (LV), B-spline, generalized prolate spheroid, sequential motion decomposition.

I. INTRODUCTION

With its unique ability to non-invasively create trackable markers (tags) in heart wall, tagged MRI [2], [3] has become one of the most promising imaging modalities in modern diagnosis of heart abnormalities [4]. By applying a special radio-frequency (RF) pulse to alter the magnetic property of selective tissues, MR tagging creates dark pattern (tags) on generated images. Since tag stripes are magnetically embedded, they move with the underlying tissue during heart deformation, thus providing unsurpassed information about local heart wall motion [5].

Extensive researches have been conducted in past decades to reconstruct 3D heart geometry and/or 4D heart motion from tagged MR images. In his series of papers [6]–[8], Young developed a finite-element (FE) framework to reconstruct LV geometry/motion based on pre-extracted boundary contours and tag points. In this work, LV was divided into 16 bicubic Hermite finite elements, and LV motion was recovered by a two-pass (backward + forward) displacement fitting algorithm. Later, by realizing that the task of boundary segmentation/tag tracking can be combined with the process of geometric fitting/motion reconstruction, Young and Augenstein improved the FE framework in their papers [9], [10], where the LV model was fitted interactively using guide points [11] and the LV motion was estimated by matching model tag surfaces to image tags. Although this method is promising in providing clinically important information [10], the small number of finite elements (16) used in this model still results in a coarse-resolution estimation to underlying heart motion [12].

In [13], the computed displacement at sparsely tracked tag points was described using a multidimensional stochastic measurement model. A priori information about heart motion was incorporated via a stochastic formulation, and the Fisher estimation framework [14] was employed to reconstruct a dense displacement field on a pre-defined lattice. This model was further improved in [12], [15] by imposing an incompressibility constraint to the displacement field and by restricting the definition of the displacement field to the myocardium. The improved model shows greater accuracy in reconstructing the synthesized displacement field, and application of this model in 3D myocardial strain calculation was also reported in [16]. In [1], [17]–[19], LV was modeled as a generalized deformable ellipsoid and LV motion was decomposed into three components: scaling, twisting and bending. After being partitioned into finite element meshes, the undeformed model was fitted to extracted contour/tag data via a physics-based approach [20], [21]. In [22], initial LV geometry was obtained by adding parametric offsets to a primitive ellipsoid. LV deformation was recovered by fitting a local displacement field and constant volume constraint was imposed to regularize the fitting process. In [23], the reconstructed LV was embedded into a 3D deformable B-solid whose iso-parametric curves—the so-called implicit snakes—correspond to tag lines in different image slices. Tag tracking, displacement fitting and interpolation were accomplished in a single step by fitting the implicit snakes to tag lines. In [24], the implicit snake framework was improved by embedding the beating LV to a 4D B-solid whose knots planes correspond to tag surfaces. A fast algorithm was also proposed to speed up the calculation of B-spline tensor products. In [25], [26], a 4D polar transformation was defined in 3D planispheric coordinates to describe the LV motion. Motion components were expressed using smooth basis functions and efficient algorithms were developed to recover motion parameters from a system of linear equations. In [27], a backward displacement field was first reconstructed on each SA slice. A tensor product of B-splines was then fitted to obtain the 4D forward displacement field of the heart wall. A brief comparison of above methods is given in Table I 123.

1Strictly speaking, the coordinate system used in Park et al.’s work is not the actual “polar” coordinate system. Here we just borrow this term from [28]

2In Young and Augenstein’s work [6]–[10], the geometric fitting was performed in “polar” (prolate spheroid) coordinate system. But the fitted model was then transformed to an equivalent Cartesian coordinate system (see, for example, [8])

3Actually heart boundary contours were used in some of Denney’s work [12], [15], but these contours were only used to delimit myocardium and non-myocardium region (not for shape reconstruction)
for a quantitative comparison of some of these methods, see [28].

Among all methods mentioned above, we find Park’s work [1], [17]–[19] particularly interesting because of its novel use of parameter functions: they are flexible enough to capture arbitrary features of LV shape and motion, yet they are very intuitive to interpret. Furthermore, all motion parameters in [1] are given in “polar” coordinates, making it possible the natural combination of motion and shape description. This model, however, also suffers from following drawbacks: it does not clearly distinguish between shape and motion parameters (due to the use of aspect ratio parameters), and the $x, y$-axis aspect ratios are correlated with the twist parameter—which can actually cause numerical instability. Furthermore, the use of fine finite element meshes in the implementation introduces large number of nodal parameters, making the model unable to describe the LV geometry in a compact manner. Finally, unlike [24]–[27], it only reconstructs LV motion at discrete time frames, and as a result the motion is not continuous in the time dimension.

Following efforts are made in this paper to extend and improve Park’s model. First, we model the LV as a generalized prolate spheroid, whose only parameter—the prolate spherical radius—is used as LV shape parameter. Second, we replace the three aspect ratio parameters by two—polar radial and $z$-axis—scaling parameters, which are uncorrelated with twisting parameter. As inspired by [29] we also incorporate global translation to LV motion formulation. Finally, we represent all model parameter (functions) in terms of B-splines, thus making the model suitable for 4D continuous motion representation.

The rest of this paper is organized as follows. Section 2 gives a brief review of Park’s generalized deformable ellipsoid, and Section 3, 4 details our LV shape/motion formulation. Experimental results on synthesized and in vivo human data are reported in Section 5, and the paper is concluded in Section 6.

II. LV SHAPE: THE GENERALIZED DEFORMABLE ELLIPSOID

The generalized deformable ellipsoid proposed by [1] is given as follows:

$$
e^0 = \alpha_0 w \begin{pmatrix} \alpha_1(u, w) \cos u \cos v \\
\alpha_2(u, w) \cos u \sin v \\
\alpha_3(u, w) \sin u \end{pmatrix}. \quad (1)$$

Here $(u, v, w)$ $(-\pi \leq u \leq \pi, -\pi \leq v \leq \pi)$ is the model coordinate, where $u$ runs along apical-basal direction, $v$ runs along circumferential direction and $w$ runs along trans-mural direction. $\alpha_0 > 0$ is a scaling factor used to control the global model size, and $\alpha_i \in [0, 1]$ are aspect ratios along $x, y$ and $z$-axis, respectively. The range of model coordinate $u$ is defined in such a way that the constructed ellipsoid is open.

Three transformations are defined in [1] to deform the ellipsoid: non-uniform compression along coordinate axes (which has been modeled using aspect ratio parameters $\alpha_i$), twisting about $z$-axis (here $\tau$ is the twisting angle)

$$T : T e = \begin{pmatrix} \cos(\tau(u, w)) & -\sin(\tau(u, w)) & 0 \\
\sin(\tau(u, w)) & \cos(\tau(u, w)) & 0 \\
0 & 0 & 1 \end{pmatrix} e, \quad (2)$$

and bending about $z$ axis (here $\epsilon_1, \epsilon_2$ are $x, y$-axis offset)

$$B : B e = e + \begin{pmatrix} \epsilon_1(u, w) \\
\epsilon_2(u, w) \\
0 \end{pmatrix}. \quad (3)$$

The overall LV deformation is then defined by:

$$e = B T e^0, \quad (4)$$

where $e^0 = (x^0, y^0, z^0)^T$ is a point on undeformed LV (in the initial frame) and $e = (x, y, z)^T$ is its position on deformed LV.

In implementation, the ellipsoid is first partitioned into finite elements meshes. Model kinematics is then formulated by a system of first-order differential equations [20], [21]:

$$D \dot{q} = f_q,$$

where $D$ is the damping matrix, $q$ is the vector of model’s degree of freedom and $f_q$ is the generalized model force. Model parameters $q$ are solved from above equation using Euler integration.

III. THE NEW GEOMETRIC MODEL: THE GENERALIZED PROLATE SPHEROID

A. The Generalized Prolate Spheroid

One potential problem of Park’s model is that it does not clearly distinguish between LV shape and motion parameters: the aspect ratios $\alpha_i$ are incorporated into LV shape formulation (1), but they are also used as deformation parameters. To describe LV shape and motion more clearly, we propose to model LV using a generalized prolate spheroid (Fig.1):
Our general-parameter functions—as proposed by [1], our model is flexible enough to capture formable superquadrics of [20], [21], the superquadrics–Free-Form-Deformation (FFD)–hybridized model of [32], [33] (see [34], [35] for a survey), the superquadrics–hybridized model of [29], [31] and our model coordinate (as in (1)), \( \alpha_0 \) is the focal radius and \( \rho(u, v, w) \) is the prolate spherical radius [10], [30]. The parameter function \( \rho(u, v, w) \) is used here to exclusively describe LV shape and will not appear in LV motion formulation.

Compared with other popular deformable models used in medical image analysis—such as the physics-based deformable superquadrics of [20], [21], the superquadrics–spherical-harmonics–hybridized model of [29], [31] and the superquadrics–Free-Form-Deformation (FFD)–hybridized model of [32], [33] (see [34], [35] for a survey), our generalized deformable prolate spheroid is more suitable for LV shape modeling since 1) it provides an efficient and direct description of the LV shape [10], [30], and 2) it is a true volumetric model while the superquadrics used in above-mentioned methods is only a surface model. In view of the fact that we are mainly interested in the estimation of LV wall motion, this point becomes an important consideration. Furthermore, with its shape parameter (the prolate spherical radius \( \rho \)) replaced by parameter functions—as proposed by [1], our model is flexible enough to capture arbitrary characteristics of LV shape and, as we will see in next section, the use of model formulation (5) can considerably simplify the geometric fitting process [10] (as compared with [1]).

### B. Initial Geometric Fitting

1) Image, World, Local, Prolate Spherical and Model Coordinate Systems: Due to their different natures, almost each of the following processes—image acquisition, information (boundary contours, tag lines, etc.) extraction and shape/motion reconstruction—requires the definition of a particular coordinate system. These different coordinate systems are now detailed as follows:

- **Image coordinate (IC) system.** The IC system is a 2D Cartesian coordinate system whose reference frame (RF) is determined by the image slice. In our application, the origin of the IC system is located at the upper left corner of the image, and the coordinate \((x, y)\) is defined such that \(x\) increases to the right and \(y\) increases downward. Basic image processing tasks such as boundary segmentation and tag tracking are performed in this coordinate system.

- **World coordinate (WC) system.** The WC system is a 3D Cartesian coordinate system whose RF is defined by the MRI scanner (Fig.2). Since slice position and orientation are generally given in WC, it is through this coordinate system that we can switch back and forth between different coordinate systems.

- **Local coordinate (LC) system.** The LC system is a 3D LV-centered Cartesian coordinate system whose RF is set up as follows [1]: its origin is located at the center of LV, its \(z\)-axis is oriented along the central long axis of LV and its \(y\)-axis is pointing toward the right ventricle (RV) (Fig.3). LV shape and local motion can be conveniently described in this coordinate system.

- **Prolate spherical coordinate (PSC) system.** The PSC system is a 3D curvilinear coordinate system in which each point is represented by the tuple \((\rho, u, v)\) [10], [30]. In our application the PSC system is also located at LV center, and the relationship between this system and the LC system is given by equation (5). We will perform geometric fitting in this coordinate system.

- **Model coordinate (MC) system.** The MC system is again a 3D LV-centered curvilinear coordinate system in which each point is represented by the tuple \((u, v, w)\) (as defined in (5)). We will define LV shape and motion parameters in this coordinate system.

2) **Methods for Geometric Fitting:** Suppose LV boundary contours have been extracted in the initial frame. We represent
these points in prolate spherical coordinates \((\rho_i, u_i, v_i)\) \((i = 1, \ldots, N)\) and adopt the algorithm proposed in [10] to perform geometric fitting: first, we fit model inner/outer surface, which represents the endocardium/epicardium, to corresponding data points (Fig.4). Then we use linear interpolation to join these data points (Fig.4). Then we use linear interpolation to join these
data points (Fig.4). Then we use linear interpolation to join these
data points (Fig.4). Then we use linear interpolation to join these

data points (Fig.4). Then we use linear interpolation to join these

data points (Fig.4). Then we use linear interpolation to join these

as:

$$\rho(u, v, w) = \sum_{i=1}^{N_u} \sum_{j=1}^{N_v} \sum_{k=1}^{N_w} B_i(u) B_j(v) B_k(w) P_{ijk},$$

where \(P_{ijk}\) are control points and \(N_u, N_v, N_w\) are number of control points used in \(u, v\) and \(w\) dimension. For a reasonable approximation, we employ cubic B-spline in \(u, v\) dimension and linear B-spline in \(w\) dimension. Substituting (7) into (6) yields a quadratic function (here \(CP^k\) is a vector of control points):

$$E_{geo}^k = \frac{1}{2} (\mathbf{CP}^k)^T \mathbf{H} \cdot \mathbf{CP}^k + (f^k)^T \cdot \mathbf{CP}^k,$$

and minimizing this function is equivalent to solving the associated linear system:

$$\mathbf{H} \cdot \mathbf{CP}^k + f^k = 0.$$

The B-spline approximation (7) has several advantages over the FE formulation: first, it does not require partitioning the geometric model into finite elements, which may be a complex process due to the singularity at model apex. Second, it avoids the use of large number of finite elements and provides a compact yet smooth representation of LV geometry. Furthermore, it is easier to implement, making it computationally attractive.

We also impose following constraints to ensure our model work properly. First, we require \(\rho(u, v, w)\) be periodic along circumferential \((v)\) direction, i.e. the last three control points in \(v\) dimension must be identical to the first three. Precisely, we have

$$P_{i,N_v-2,k} = P_{i,1,k}, \quad P_{i,N_v-1,k} = P_{i,2,k}, \quad P_{i,N_v,k} = P_{i,3,k}$$

each for \(\text{i} \) and \(\text{k}\). Second, to avoid possible wild behavior of the end control points near base \((u = 0)\), we triple these control points so that the fitted surface must pass through them. Precisely, we have

$$P_{N_u,j,k} = P_{N_u-1,j,k} = P_{N_u-2,j,k}.$$

Third, to obtain a smooth model representation we also require \(\rho(u, v, w)\) be \(C^1\) at apex \((u = -\pi)\). All these constraints are formulated as linear equations, and the singular value decomposition (SVD) [36] is employed to solve the whole linear system.

IV. FAST MOTION RECONSTRUCTION: THE ENHANCED SEQUENTIAL MOTION DECOMPOSITION

A. Sequential Motion Decomposition

We combine the work of Chen [29], [31], Park [1] and propose to decompose heart motion into four sequential components: global translation (rigid), polar radial/2-axis compression, twisting and bending (nonrigid). These components are detailed as follows.

Global translation. Under the assumption that the local deformation is small compared with the global translation [29], we trace the model in 3D space by recording the position of its centroid:

$$\text{centroid}(t) = (x(t), y(t), z(t))^T.$$  

Note the use of the time parameter \(t\) gives us the potential ability to reconstruct continuous 4D heart motion in a very natural manner.

Compression. Different from the approach used in [1], we decompose LV compression into two subcomponents—polar
radial compression \( \alpha_r \) and z-axis compression \( \alpha_z \)—and write the deformation operator as:

\[
C^t : C^t e = C^t \left( \begin{array}{c} lx \\ ly \\ lz \\ \end{array} \right) = \left( \begin{array}{c} \alpha_r(u, w, t) \cdot lx \\ \alpha_r(u, w, t) \cdot ly \\ \alpha_z(u, w, t) \cdot (lz - lz_p) + lz_p \\ \end{array} \right).
\] (9)

Note that we have used one polar radial compression parameter \( \alpha_r \) rather than two aspect ratio parameters \( \alpha_1, \alpha_2 \) to represent the compression in \( xy \)-plane. This can actually help us avoid the numerical instability, and to see this point we may consider following scenario (Fig.5):

Suppose \( P \) is an undeformed model point and \( P' \) is the corresponding deformed point. By using Park’s model, we are not able to figure out whether \( P \) is deformed by the twist motion \( \tau \), or by the combination of \( x \)-axis scaling \( \alpha_1 \) and \( y \)-axis scaling \( \alpha_2 \). This means the compression and twisting motion are correlated, and this can cause numerical instability since the resulting fitting problem is ill-posed (the solution is not unique). However, by replacing \( \alpha_1, \alpha_2 \) with \( \alpha_r \) we no longer have this kind of problem. Say, in above example we know \( P \) must be deformed by the twist motion \( \tau \).

We also point out that in above formulation, we have modeled z-axis compression in such a way that the LV will contract toward some point \( (lz_p) \) other than its centroid. It is true that this “strange” modification does not comply with the common rules proposed in previous work [1], [25], [26], [37], but the fact is that in our experiment, we found no such evident contraction pivot for LV apical-basal compression near the LV centroid. Since the incorrect choice of the contraction pivot may cause disastrous results (especially when the pivot is placed within LV), here we select \( lz_p \) in such a way that it lies on the apical-basal axis, near to apex and outside the LV.

Twisting. We simply use the temporal version of the twist formula (2) and write the deformation operator as:

\[
T^t : T^t e = \left( \begin{array}{ccc} \cos(\tau(u, w, t)) & -\sin(\tau(u, w, t)) & 0 \\ \sin(\tau(u, w, t)) & \cos(\tau(u, w, t)) & 0 \\ 0 & 0 & 1 \end{array} \right) e,
\] (10)

where \( \tau \) represents the twisting angle.

Bending. Again, we write the temporal version of the bending operator (3) as:

\[
B^t : B^t e = e + \left( \begin{array}{c} \epsilon_1(u, w, t) \\ \epsilon_2(u, w, t) \\ 0 \end{array} \right),
\] (11)

where \( \epsilon_1, \epsilon_2 \) are \( x, y \)-axis offset.

With the four motion components defined as above, the overall local motion of LV can be formulated as:

\[
e^t = B^t T^t C^t e^0,
\] (12)

where \( e^0 = (lx^0, ly^0, lz^0)^T \) is a point on undeformed LV (in the initial frame) and \( e = (lx, ly, lz)^T \) is its position on deformed LV at time \( t \). Global motion can be combined with (12) to yield a complete description of LV motion.

It is worth noting that we do not incorporate global rotation into our LV motion model (as in [29]). This is because LV is nearly symmetric about its local z-axis (long axis) and so the accurate estimation of LV local \( x, y \)-axis is generally difficult. Since the global rotation of LV is relatively small [37], we expect that neglecting the global rotation will not significantly affect the accuracy of our model.

Compared with [1], [29], our motion model provides a more complete and clearer description of LV motion since 1) we consider both global and local motion and 2) all parameter functions used in this model have very intuitive interpretations. Also, the use of 4D parameter functions in our model makes it suitable for 4D LV motion reconstruction. Furthermore, by decomposing complex heart motion into several components and by utilizing B-splines (as we will discuss in following subsection), we will be able to develop fast algorithms to reconstruct LV motion.

B. Motion Reconstruction

1) Imaging Protocol: Two sets of heart data (HD) are used in our experiment: the first one (HD1) was taken from a normal volunteer, and the second one (HD2) was taken from a patient suffering from myocardial infarction. For each data set, both short-axis (SA) and long-axis (LA) images were acquired in a breath-hold SPAtial Modulation of Magnetization (SPAMM) cine sequence, and the LA imaging planes were arranged to be radially distributed around LV long axis. Grid and parallel tag patterns were created on SA and LA images, respectively (Fig.6).

2) Tag Tracking: We have modified Young’s previous work [8] and developed a snake-based algorithm to track tags on each image slice. Fig.7-8 shows some of the typical tracking results.

3) Methods for Motion Reconstruction: Given fitted LV shape model \( e^0 \) and tracked tag points in frame \( t \), we estimate LV motion in following three steps: rigid motion (global translation) estimation, local backward displacement fitting and non-rigid motion estimation.

Rigid Motion (Global Translation) Estimation. Suppose the tag points tracked in frame \( t \) have been clipped to the model and have been represented in world coordinates...
\[(wx^t_i, wy^t_i, wz^t_i)^T \quad (i = 1, \ldots, N)\]. Let \[I_1 = \{ i : (wx^t_i, wy^t_i, wz^t_i)^T \in \text{SA slices} \},\]
\[I_2 = \{ i : (wx^t_i, wy^t_i, wz^t_i)^T \in \text{LA slices} \},\]
and denote the size of \(I_1/I_2\) by \(N_1/N_2 (N_1 + N_2 = N)\). We estimate model centroid in frame \(t\) in following two steps: first, the data centroid \(dc^t\) is computed and the origin of the local coordinate (LC) system is translated to \(dc^t\). Second, the model apex/base centroid is estimated and the model centroid is computed by (Fig.9):
\[
\text{centroid}(t) = \text{LC2WC} \left( \text{apex}^t + \frac{2}{3} (\text{base}^t - \text{apex}^t) \right),
\]
where \(\text{LC2WC}\) is a routine converting local coordinates to world coordinates. Finally, the LC system is translated to the estimated model centroid and the local coordinates of tag points are computed (which we denote by \(e^t_i = (lx^t_i, ly^t_i, lz^t_i)^T\)).

**Local Backward Displacement Fitting.** As stated in [8], [26], [27], each tracked LA tag point provides a 1D constraint on the local backward displacement field—here we refer the backward displacement as “local” since it is computed in local coordinate system —that is, if we let \(e^0_i = (lx^0_i, ly^0_i, lz^0_i)^T \quad (i \in I_2)\) denote the local coordinate of LA tag points in the initial frame, \(e^t_i = (lx^t_i, ly^t_i, lz^t_i)^T \quad (i \in I_2)\) denote the local coordinate of LA tag points in frame \(t\), \((u^t_i, v^t_i, w^t_i) \quad (i \in I_2)\) denote the model coordinate of LA tag points \(e^t_i\) in frame \(t\), \(e^{00}_i = e^0_i (u^t_i, v^t_i, w^t_i) = (lx^{00}_i, ly^{00}_i, lz^{00}_i)^T \quad (i \in I_2)\) denote the local coordinate of the model point \((u^t_i, v^t_i, w^t_i)\) in the initial frame, we must have
\[
(e_i^{00} - e^t_i) \cdot n_{LA} = (e_i^0 - e^t_i) \cdot n_{LA}, \quad i \in I_2,
\]
where \(n_{LA}\) is the unit normal of corresponding initial tag plane (Fig.10). This constraint also holds for SA tag points, but in our case, each SA tag point provides two 1D constraints on the local backward displacement field since they are tracked to provide 2D planar LV motion. That is, if we use the same notation as above (but \(i \in I_1\)), we must have
\[
(e_i^{00} - e^t_i) \cdot n_{SA_k} = (e_i^0 - e^t_i) \cdot n_{SA_k}, \quad i \in I_1, \quad k = 1, 2
\]
where \(n_{SA_1}\) and \(n_{SA_2}\) are unit normals of corresponding initial tag planes. Although (14) and (15) cannot be directly combined to provide complete backward displacement for arbitrary material point in myocardium, we can construct a continuous backward displacement field by approximating...
each component of the displacement field using a continuous function.

Following the idea of [23], [27], we represent each component of the backward displacement as a B-solid:

\[
bd_i(lx, ly, lz) = \frac{1}{N_l} \sum_{l=1}^{N_l} \frac{1}{N_m} \sum_{m=1}^{N_m} \frac{1}{N_n} \sum_{n=1}^{N_n} B_l(lx)B_m(ly)B_n(lz)(P_{lmn})^i,
\]

(16)

where \(bd_i (i = 1, 2, 3)\) is the 1D displacement along \(n_{SA_1}, n_{SA_2}\) and \(n_{LA}^i = n_{SA_1} \times n_{SA_2}\), \(lx, ly, lz\) are local coordinates, \(B_l, B_m, B_n\) are basis functions and \((P_{lmn})^i\) are corresponding control points. We choose cubic basis function in each dimension, and for each basis function we use periodic uniform knot sequence:

\[
f_0; 0; 0; 0; u_1; \ldots; u_{n-3}; u_1; u_2; u_1; u_1;\]

where \(n\) is the number of control points. To fit the B-solid to given displacement data, we need to minimize following least square errors \((k = 1, 2)\):

\[
E_{bdk} = \frac{1}{N_l} \sum_{i \in I_k} \left( (e_i^k - e_i^k) \cdot n_{SA_k} - bd_k(lx^k_i, ly^k_i, lz^k_i) \right)^2,
\]

(16)

However, it is not hard to see that even for relatively small number of control points (say, a \(5 \times 5 \times 5\) grid), the direct minimization of (17) is very time-consuming due to the large size of the coefficient matrix. To remedy this situation, we take the approach analogous to [24] and design a fast algorithm to perform displacement fitting (see Appendix for details). Roughly speaking, this algorithm works by interpolating motion data to a predefined grid so that the multi-pass fitting procedure is applicable. With the use of this algorithm, we gained at least a ten-fold of speed up in our experiments.

**Nonrigid Motion Estimation.** Given the fitted local backward displacement field, we can compute the initial positions of tag points \(e_i^0\) (see Fig.10) and their model coordinates \((u_i, v_i, w_i)\) \((i = 1, \ldots, N)\). We next estimate the (forward) nonrigid LV motion by following three sequential steps:

1. Compression. The compression parameter \(\alpha_r\) is estimated by minimizing following cost function:

\[
Ec_r = E_{reg}(\alpha_r) + \frac{1}{N} \sum_{i=1}^{N} \mu_i \left( \alpha_r (u_i, v_i, w_i, t) r_i^0 - r_i^t \right)^2,
\]

(18)

where \(E_{reg}(\alpha_r)\) is the regularization term defined in (6) and
Fig. 6. Typical SA (top) and LA (bottom) images. The dashed lines show the arrangement of LA (top) and SA (bottom) imaging planes.

Fig. 9. Estimation of the model centroid.

The $\mu_i$'s are weight parameters (we set $\mu_i = 1/r_i^0$ in our experiments). For $z$-axis compression parameter $\alpha_z$, the estimation is accomplished by minimizing a similar cost function:

$$E_{cz}^t = E_{\text{reg}}(\alpha_z) + \frac{1}{N} \sum_{i=1}^{N} \mu_i \left( \alpha_z (u_i^t, w_i^t, t) \left( l z_i^{t0} - l z_p \right) \right)^2,$$

where we set $\mu_i = 1/|l z_i^{t0} - l z_p|$.  

2. Twisting. Similarly, the twisting angle is estimated by minimizing:

$$E_{\text{twist}}^t = E_{\text{reg}}(\tau) + \frac{1}{N} \sum_{i=1}^{N} \mu_i \left\| T^t C^t \mathbf{e}_i^{t0} - \mathbf{e}_i^t \right\|_{xy}^2,$$

where $\mu_i = 1/t_i^{t0}$ and the norm $\| \cdot \|_{xy}$ is taken with respect to $x$ and $y$ only:

$$\left\| (x, y, z)^T \right\|_{xy}^2 = x^2 + y^2.$$

Note we have used updated compression operator $C^t$ in (20) since we are estimating motion parameters in sequential order.

3. Bending. The estimation of bending parameters follows in a completely similar manner and can be achieved by minimizing following objectives:

$$E_{bx}^t = E_{\text{reg}}(\epsilon_1) + \frac{1}{N} \sum_{i=1}^{N} \left( \left[ B^t T^t C^t \mathbf{e}_i^{t0} \right]_x - l x_i^t \right)^2,$$

$$E_{by}^t = E_{\text{reg}}(\epsilon_2) + \frac{1}{N} \sum_{i=1}^{N} \left( \left[ B^t T^t C^t \mathbf{e}_i^{t0} \right]_y - l y_i^t \right)^2.$$

4) Implementation: As we have done in LV geometry fitting, we approximate local motion parameters ($\alpha_r, \alpha_z$ and $\tau$) using tensor product of B-splines—say, we may approximate the radial compression $\alpha_r(u, w, t)$ by:

$$\alpha_r(u, w, t) = \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} B_i(u) B_j(w) P_{ij}.$$

Thus minimizing (18) is equivalent to solving a linear system. All other local motion parameters can be estimated in the same way except for $\tau(u, w, t)$, the twisting angle, whose minimization requires the solution of a nonlinear system. To simplify the calculation, we write $T^t C^t \mathbf{e}_i^{t0}$ in full as:

$$T^t C^t \mathbf{e}_i^{t0} = \begin{pmatrix} \cos \tau \cdot \alpha_r \cdot l x_i^{t0} - \sin \tau \cdot \alpha_r \cdot l y_i^{t0} \\ \sin \tau \cdot \alpha_r \cdot l x_i^{t0} + \cos \tau \cdot \alpha_r \cdot l y_i^{t0} \end{pmatrix}.$$

By substituting the Taylor expansion of sine and cosine functions

$$\sin \tau \approx \tau, \quad \cos \tau \approx 1 - \frac{\tau^2}{2},$$

Fig. 10. Tracked tag points on LA images provide a 1D constraint on LV motion [15], [27] (see (14))
into above formula, we obtain:
\[
T^iC^i\mathbf{e}_{i0} \approx \left( \frac{1 - \frac{x_i^2}{2}}{\alpha_x} - \frac{y_i^0}{\alpha_y} \right) \cdot \alpha_x \cdot l_x^{i0} + \left( 1 - \frac{x_i^2}{2} \right) \cdot \alpha_y \cdot l_y^{i0}.
\]
Substituting above approximation into (20) and omitting \(O(\tau^3)\) terms then yields the desired quadratic objective. Given the assumption that LV twisting is small during systole (< 10\(^o\)), we expect that this linearization will not significantly affect the accuracy of twisting estimation.

After imposing appropriate constraints to guarantee the smoothness of all motion parameters at the apex \((u = -\frac{\pi}{2})\), we use SVD to solve the whole linear system.

### C. Time Smoothing

After estimating motion parameters in each frame, we perform a 1D time smoothing (as in [27]) and obtain a 4D continuous representation for LV motion parameters.

Compared with other B-spline based motion reconstruction methods such as the B-tag surface of [38], the B-solid of [23], [24], the 4D B-spline motion field of [27] and the Free Form Deformation (FFD) of [32], our approach enjoys several advantages since 1) the backward displacement fitting is very fast. By estimating each displacement component directly instead of constructing the complete displacement field at selected points, we make it possible the employment of the standard multi-pass fitting algorithm used in B-spline based data modeling. Consequently, even for a control point grid as fine as the 13 × 13 × 13 grid used in our experiment, our backward displacement fitting algorithm is able to reconstruct the displacement field for one frame within ten seconds (on a 1.4GHz PC). Our approach is also attractive since 2) the parameters used in our model are meaningful. This makes the tracking results readily interpretable in clinical applications [1].

### V. Experimental Results

#### A. Geometric Fitting

To test our geometric fitting algorithm, we first manually segment LV contours in the initial frame. After stacking all contours in the 3D space, we overlay the prolate spheroid onto the data and apply the geometric fitting algorithm to deform the model. The recovered LVs are shown in Fig. 11.

#### B. Motion Reconstruction

1) Validation on Motion Simulator: We first validate our motion reconstruction algorithm on synthesized data in following three steps. First, the geometric model defined in equation (5) (with specific shape parameters) is deformed by tuning the motion parameters. Second, a set of model points are selected and tracked. Third, the motion reconstruction algorithm is applied and the results are compared with the true motion.

![Fig. 11. Recovered LV geometry (at ED). Left: HD1. Right: HD2](image-url)

<table>
<thead>
<tr>
<th>Simulator</th>
<th>Shape</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>S2</td>
<td>47.6</td>
<td>HD1</td>
</tr>
<tr>
<td>S3</td>
<td>58.0</td>
<td>HD2</td>
</tr>
</tbody>
</table>

Our code is developed in Matlab, which is running on an application server in our experiment. This means that, for all algorithm running times we presented in this paper, the network overhead has also been counted in.
Several observations can be made from the tracking results.

1) Generally speaking, our algorithm is successful in predicting compression and twisting parameters: the maximum errors are always within a few percent even in the presence of reasonably sparse data (say, MD2). The errors can be further reduced when the number of trackable model points is reasonably large (Table III).

2) On the other hand, our algorithm is not very accurate in estimating model centroid—the maximum error for S3 is more than 8%. Nevertheless, this behavior is expected since the approximation (13) is not accurate if a) large deformation occurs during heart contraction, or b) the model points are not uniformly distributed in the 3D space. This is exactly what happened in our experiment.

3) At the first glance, what we have in 1) and 2) seems to be a contradiction: our algorithm does not work very well for the estimation of the model centroid, but it accurately predict the compression and twisting parameters despite the fact that the model centroid is estimated first. A plausible explanation for this apparent paradox is that, although the percentage error of the model centroid is large, the model does not translate very much during the deformation. Thus the absolute error of the model centroid is in effect small. For example, the largest percentage error for the model centroid of S3 is 8.8%, but the largest absolute error is only 0.61 since
4) We note that the maximum error of model centroid and \( z \)-axis compression parameter is relatively insensitive to the number of trackable model points. For all three simulators the maximum error of these two parameters does not change very much. On the other hand, the estimation of polar radial compression parameter and twisting parameter depends strongly on the number of trackable points (in particular, the number of trackable points in \( u, v \) dimension). The maximum error of these two parameters can go crazy if not enough model points are presented.

6) We finally compare our method with some of the previous work and the results are listed in Table IV. We emphasize that above experiments only provide a partial validation of our motion reconstruction algorithm since the steps of tag tracking and local backward displacement fitting have not been tested. Nonetheless, the reconstruction results are promising and they strongly suggest the feasibility of our method.

We also note that both algorithms for geometric fitting and motion reconstruction are fast enough to be conveniently deployed in real life applications. In our experiments, the average time for geometric fitting is about 3.5 seconds and that for motion reconstruction (including local backward displacement fitting) is about 11 seconds per frame. The algorithms are developed in Matlab 6.1, and they are tested on a Pentium IV 1.4G PC.

7) There are other authors who also tested their methods on cardiac simulators [27], [38], but they used absolute errors to measure the accuracy of the displacement fitting, so the results are not directly comparable with ours.

8) One may suspect that the error of model centroid will be further increased once the local backward displacement fitting is included. This is not the case since the estimation of model centroid does not depend on the estimation of the backward displacement field (see equation (13)).

## Table III

<table>
<thead>
<tr>
<th>MD</th>
<th>PT</th>
<th>CTRD</th>
<th>( \alpha_u )</th>
<th>( \alpha_z )</th>
<th>( \tau )</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 x 13 x 4</td>
<td>0.7866</td>
<td>2.3617</td>
<td>1.8523e-25</td>
<td>1.6864</td>
<td>0.0100</td>
</tr>
<tr>
<td>11 x 11 x 4</td>
<td>0.8007</td>
<td>2.3617</td>
<td>1.3159e-26</td>
<td>1.7149</td>
<td>0.0107</td>
</tr>
<tr>
<td>9 x 9 x 4</td>
<td>0.7800</td>
<td>2.3617</td>
<td>2.0844e-25</td>
<td>1.5261</td>
<td>0.0143</td>
</tr>
<tr>
<td>7 x 7 x 4</td>
<td>0.7543</td>
<td>2.3617</td>
<td>1.0416</td>
<td>1.6199</td>
<td>3.2276</td>
</tr>
<tr>
<td>5 x 5 x 4</td>
<td>0.9042</td>
<td>2.3617</td>
<td>5.9378</td>
<td>3.4116</td>
<td>20.510</td>
</tr>
<tr>
<td>13 x 13 x 2</td>
<td>0.7849</td>
<td>2.3617</td>
<td>2.4501e-27</td>
<td>1.6943</td>
<td>0.0106</td>
</tr>
<tr>
<td>11 x 11 x 2</td>
<td>0.7989</td>
<td>2.3617</td>
<td>1.9743e-25</td>
<td>1.7128</td>
<td>0.0113</td>
</tr>
<tr>
<td>9 x 9 x 2</td>
<td>0.7783</td>
<td>2.3617</td>
<td>8.8904e-27</td>
<td>1.5243</td>
<td>0.0156</td>
</tr>
<tr>
<td>7 x 7 x 2</td>
<td>0.7527</td>
<td>2.3617</td>
<td>0.3162</td>
<td>1.6179</td>
<td>1.3285</td>
</tr>
<tr>
<td>5 x 5 x 2</td>
<td>0.9023</td>
<td>2.3617</td>
<td>5.9378</td>
<td>3.4106</td>
<td>20.505</td>
</tr>
</tbody>
</table>

## Table IV

<table>
<thead>
<tr>
<th>Simulator</th>
<th>Model</th>
<th>PTGERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ours</td>
<td>&lt; 0.4</td>
</tr>
<tr>
<td>2</td>
<td>Huang [24], Fig. 15</td>
<td>2.476</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1–2</td>
</tr>
</tbody>
</table>

5) As can be observed from Table III, the estimation of model centroid and \( z \)-axis compression parameter is relatively insensitive to the number of trackable model points. For all three simulators the maximum error of these two parameters does not change very much. On the other hand, the estimation of polar radial compression parameter and twisting parameter depends strongly on the number of trackable points (in particular, the number of trackable points in \( u, v \) dimension). The maximum error of these two parameters can go crazy if not enough model points are presented.

We first show in Fig.15 the motion parameters reconstructed from the normal heart. From these plots, it is not difficult to see that the general properties of (normal) LV motion have been successfully captured. Specifically, we observe that...
1) the polar radial compression parameter varies generally slowly along the apical-basal direction. However, it changes faster and is more significant at the inner wall, where the maximum compression is around 35%, than it does at the outer wall, where the maximum compression is around 13%. 

2) The twisting parameter is increasing along the apical-basal direction and takes different signs near the apex and the base—this indicates that the apex and the base twist in opposite directions. What’s more, the twisting at both walls is generally small (\(<0.2\) radians, i.e. 11.46 degrees), and it is more significant at the inner wall, where \(\tau\) ranges from -0.2 to 0.15 radians, than it is at the outer wall, where \(\tau\) ranges from -0.07 to 0.11 radians. All these results agree with the experiments reported in [1], [37].

On the other hand, we must be careful when interpreting the reconstructed \(z\)-axis compression parameter \(\alpha_z\). Since the value of \(\alpha_z\) depends strongly on the choice of \(lz_p\) (the compression pivot), the plot of \(\alpha_z\) itself can be misleading and should not be used directly. To examine whether \(\alpha_z\) has been estimated correctly, we instead compute the \(z\)-axis displacement of LV:

\[
\begin{align*}
    dz &= lz^{10} - (\alpha_z (lz^{10} - lz_p) + lz_p) \\
    \alpha_z &= \frac{tz^{10} - tz_p}{lz^{10} - lz_p} \\
    \alpha_z &= \frac{tz^{10} - tz_p}{lz^{10} - lz_p}
\end{align*}
\]

and plot the results at both walls (Fig.15). From these plots, we see that there is more compression toward the base (\(u = \frac{7}{6}\)), and, as before, the compression is slightly more significant at the inner wall (with maximum value around 5 mm) than it is at the outer wall (with maximum value around 3.4 mm). This again agrees with the common knowledge about LV motion known to physicians.

Having examined the motion pattern of the normal heart, we next turn to the abnormal heart and plot its motion parameters in Fig.16. From these plots, we immediately observe that 1) the polar radial compression of the abnormal heart is significantly less than that of the normal heart; this is especially true at the inner wall where the maximum compression is around 10%, in contrast to the 35% of the normal heart. We also notice that the abnormal heart tends to stretch near the apex. 2) The abnormal heart twists less at the base but more at the apex, and it appears to twist in the reversed direction at the inner wall. 3) Similar to the twisting motion, the \(z\)-axis compression of the abnormal heart is less significant at the base but more obvious at the apex; moreover the apex tends to move away from the compression pivot. In summary, the motion pattern of the abnormal heart is fundamentally different from that of the normal heart.

The parameter graph is not only useful in providing qualitative information about LV motion—in our case it can even help us locate the infarction region. In fact, with a careful examination of Fig.16—especially the two plots of the polar radial compression parameter, we find that the infarcted tissue is located within the region \(-1.1 \leq u \leq -0.5\). Moreover it is closer to the inner wall than to the outer wall.

In conclusion, our method is successful in capturing general features of LV motion and in differentiating between normal and abnormal cases. Thus it does provide qualitatively good results.

VI. CONCLUSION AND FUTURE WORK

In this paper, we extended Park’s previous work [1] and developed a uniform framework for both LV shape modeling and motion reconstruction. As the main contributions of our work, we 1) carefully distinguished and formulated LV shape/motion parameters, 2) combined the work of [29] and [1] to give a more complete decomposition of LV motion, 3) gave compact,
smooth representations of both LV shape/motion by utilizing B-splines, and 4) developed fast algorithms to estimate LV shape/motion parameters from tagged image data.

Having summarized the main achievements of our work, we also point out the following problems and provide suggestions on possible future work:

1) Although an important step in cardiac MR image analysis, the task of boundary segmentation/tracking was not addressed in our work—actually we only segmented boundary contours in the initial frame manually and used them in LV geometry reconstruction. This is not a big problem at the present stage (since boundary
Fig. 16. Estimated motion parameters from HD2. From left to right, top to bottom: $\alpha_r$ at inner wall, $\alpha_r$ at outer wall, $\tau$ at inner wall, $\tau$ at outer wall, $z$-axis displacement at inner wall, and $z$-axis displacement at outer wall.

Contours were not used in myocardial motion tracking, but as indicated in [39], accurately tracked boundary contours can be a good indicator of LV boundary motion and thus can give us a more complete view of LV motion. What’s more, the shape characteristics derived from tracked LV boundaries may also be useful in heart abnormity diagnosis [39]–[42], so the development of appropriate boundary tracking methods would still be one important consideration in our future work. On the other hand, since boundary segmentation in tagged MR images is particularly difficult due to the intervention of tag lines, preprocessing techniques (such as gray-
scale morphological operations) may be required before boundary contours can be reliably extracted and tracked [43].

2) It is also worth mentioning that in LV motion formulation, we made the assumption that all motion parameters are independent of $v$ (see (9)–(11)) and this in turn was used to simplify the reconstruction process. As pointed out by other researchers, however, this assumption may not be realistic since it is common in diseased cases that LV motions are nonuniform along circumferential ($v$) direction. In spite of this fact, we do notice that positive results have been reported in [17]–[19], and we will address this problem in our future work.

3) The motion simulator we developed in this work does not test tag tracking and local backward displacement fitting. To fully understand the performance of our method, it would be helpful to develop a more powerful simulator which can simulate both tag generation and tag deformation. We will investigate this problem in our future research.

4) With the estimated LV motion parameters, the next step of our work would be to quantitatively analyze these data and derive clinically useful information, such as myocardial strain, to help interpret the tracking results. This will also make it possible the quantitative comparison between our method and that of Park’s. Furthermore, more experiments need to be conducted to investigate the relationship between reconstructed motion parameters and various heart diseases.

APPENDIX

In this appendix we give the details of the fast displacement fitting algorithm used in section IV-B.

Algorithm for Fast Local Backward Displacement Fitting–Displacement Component $bd_1$ and $bd_2$

1. Cover the domain of the B-solid $([0, 1] \times [0, 1] \times [0, 1])$ with a 3D grid. This grid is placed in such a way that along $lx$, $ly$-dimension, the grid is uniform while along $lz$-dimension, the grid is (probably) non-uniform where grid points are placed on each SA slice and on the end planes $lz = 0$, $lz = 1$ (which passes through LV apex and base) (Fig.17). Denote the grid points by $(glx_i, gly_j, glz_k)$ ($i = 1, \ldots, nx$, $j = 1, \ldots, ny$, $k = 1, \ldots, nz$).

2. On each SA slice, interpolate displacement component ($bd_1$ or $bd_2$) to grid points while for the end planes $lz = 0$, $lz = 1$, set displacement at grid points to zero (this actually serves as a regularization term in following minimization). Denote the interpolated displacement at grid points by $gbd_{ijk}$.

3. Compute $P_{ijn}''$ ($i = 1, \ldots, nx$, $j = 1, \ldots, ny$, $n = 1, \ldots, N_n$) by minimizing:

$$\frac{1}{ny} \sum_{j=1}^{ny} \left(P_{ijn}'' - \sum_{m=1}^{N_m} B_m(gly_j)P_{ijn}''ight)^2.$$

4. Compute $P_{imn}''$ ($i = 1, \ldots, nx$, $m = 1, \ldots, N_m$, $n = 1, \ldots, N_n$) by minimizing:

$$\frac{1}{nx} \sum_{n=1}^{nx} \left(P_{imn}'' - \sum_{l=1}^{N_l} B_l(glx_i)P_{imn}''ight)^2.$$

5. Compute $P_{imn}$ ($l = 1, \ldots, N_l$, $m = 1, \ldots, N_m$, $n = 1, \ldots, N_n$) by minimizing:

$$\frac{1}{nx} \sum_{i=1}^{nx} \left(P_{imn}'' - \sum_{l=1}^{N_l} B_l(glx_i)P_{imn}''ight)^2.$$

For the third displacement component $bd_3$, the above algorithm needs to be modified slightly [27]:

Algorithm for Fast Local Backward Displacement Fitting–Displacement Component $bd_3$

1. Cover the domain of the B-solid with the same grid.

2. On each LA slice, project the computed cross-tag displacement (along $n_{LA}$) onto $n_{LA}$ to obtain

![Fig. 17. Construction of the computational grid used in the backward displacement fitting. Top: the 2D grid placed on each SA slice (the small circles are tag points). Bottom: the final 3D grid.](image)
3. On each SA slice, interpolate displacement component $b_{3d}$ to grid points using data coming from SA-LA slice intersections.

4. Compute the control points by minimizing (26), (27) and (28).

ACKNOWLEDGMENT

We would like to thank Prof. Wynnie Lam from The Prince of Wales hospital and Mr. Albert Wang for making the in vivo human heart data available to us. We also thank Mr. Yuanquan Wang, Ms. Yuehong Sun, Mr. Zeming Zhou and Prof. Deshen Xia at The Nanjing University of Science and Technology for their helpful comments on this work.

REFERENCES


Medical Image Analysis, vol. 1, no. 2, 1996.


[37] T. Arts, W. Hunter, A. Douglas, A. Muijtjens, and R. Reneman,
“Description of the Deformation of the Left Ventricle by a Kinematic

[38] A. A. Amini, Y. Chen, M. Elayyadi, and P. Radeva, “Tag Surface
Reconstruction and Tracking of Myocardial Beads from SPAMM-MRI

“Point-Tracking Quantitative Analysis of Left Ventricular Surface Motion

Descriptors on 3-D Deformable Surfaces: Application to the Left-

for Temporal Analysis of Cardiac Motion,” in Proceedings of the IEEE
124–133.

[42] J. C. McEachen and J. S. Duncan, “Shape-Based Tracking of Left