Cardiac Motion Simulator for Tagged MRI

Edo Waks*, Jerry L. Prince*, and Andrew S. Douglas†
* Department of Electrical and Computer Engineering
† Department of Mechanical Engineering
Johns Hopkins University
Baltimore, MD 21218
E-mail: prince@jhu.edu

Abstract

This paper describes a computational simulator for use in cardiac imaging using tagged magnetic resonance imaging. The simulator incorporates a 13-parameter model of left-ventricular motion due to Arts et al. (1992) and applies it to a confocal prolate spherical shell, resembling the shape of the left ventricle. Using parameters determined in other work, our model can be made to assume a configuration representing one of 60 phases in the cardiac cycle. In this paper, we determine the inverse motion map analytically, allowing pointwise correspondences to be made between two points at any two times. Using this mathematical relationship, we simulate the (tagged) magnetic resonance imaging process using a standard (tagged) spin-echo imaging equation. Image sequences can be synthesized at arbitrary orientations at any phase. We currently synthesize a SPAMM tag pattern with arbitrary spatial frequency, but other patterns can be readily incorporated. To accommodate two-dimensional motion estimation algorithms, we have created a two-dimensional simulator which restricts the three-dimensional motion to two dimensions. In either two or three dimensions, a true motion is output so that motion estimation algorithms can be compared against the truth. We conclude with a simple demonstration of the performance of the simulator.

Keywords — motion analysis of biomedical images, cardiac magnetic resonance imaging, representation and description of multidimensional shape.

1 Introduction

Tagged MR imaging [8] can provide important information about the motion of the left ventricle (LV) of the heart. In this technique, a pattern of spatially varying magnetism is embedded in the LV using a special tagging pulse sequence. This intensity deforms along with the LV wall during the heart cycle, and is imaged by standard MRI techniques. Motion estimation methods attempt to determine the motion of the heart from the deformation of the tag pattern within an image sequence [2] [6] [4] [7] [3].

In order to test the effectiveness of a motion estimation algorithm it is highly desirable to obtain a set of images representing a known motion. Ideally, a deformable phantom would be used for this purpose. But such phantoms are difficult to build and control, and *a priori* knowledge of the motion they generate is often not available. Furthermore, such phantoms are usually restricted to simple deformations such as compressions and rotations. Computer simulation is an alternative which does not suffer from many of the problems encountered by phantoms. In this approach, a computer is used to generate a series of synthetic images which display some form of motion. The underlying mathematical transformation responsible for the motion displayed by these images is known, making this a good control experiment. But care must be taken to ensure that these simulations are representative of a true medical imaging experiment.

In this research, our goal was to develop an accurate simulator that could generate a control experiment for motion estimation from tagged cardiac MR images. In order to develop a representative simulation, we first selected a shape that is both similar to the LV, and also easy to work with. Our model is based on a prolate spheroidal geometry, and we implement a mechanical model for LV motion developed by Arts et al. [1]. We generate realistic tag patterns using a tagged MR imaging equation described by Prince and McVeigh [6]. These mathematical models have been used to form an accurate simulator capable of generating complex motion as well as perfectly calculable displacement fields.

This research was supported by NIH grant R29-HL47405, and NSF grant MIP-9350336. Correspond to prince@jhu.edu.

2 Background

2.1 Model Shape

In our approach we begin with a three-dimensional volume, and then take two-dimensional slices in order to simulate the MR imaging process. The basis of our geometric model is the prolate sphere, which is a three-dimensional ellipse-like object, as shown in Figure 1.

A prolate sphere is defined to have a constant radius λ which determines its size. Points which lie on this surface can be identified by the parameter λ, along with the elevation angle η and the azimuthal angle ϕ, shown in Figure 1. A point (λ, η, ϕ) in prolate spheroidal coordinates has the following Cartesian coordinates:

$$\begin{aligned} x &= \delta \sinh\lambda \sin\eta \cos\phi\,, \\ y &= \delta \sinh\lambda \sin\eta \sin\phi\,, \\ z &= \delta \cosh\lambda \cos\eta\,. \end{aligned} \tag{1}$$

The fixed parameter δ is the focal radius (the distance from the origin to either focus). A point (x, y, z) in Cartesian coordinates has the following prolate spheroidal coordinates:

$$\begin{aligned} r_1 &= \sqrt{x^2 + y^2 + (z+\delta)^2}\,, \\ r_2 &= \sqrt{x^2 + y^2 + (z-\delta)^2}\,, \\ \lambda &= \cosh^{-1}\frac{r_1 + r_2}{2\delta} && \lambda > 0\,, \\ \eta &= \cos^{-1}\frac{r_1 - r_2}{2\delta} && 0 \le \eta \le 180\,, \\ \phi &= \tan^{-1}\frac{y}{x} && 0 \le \phi < 360\,. \end{aligned} \tag{2}$$

where r_1 and r_2 are the distances between (x, y, z) and each focus.

The LV itself can be modeled by defining it to lie between a pair of confocal prolate spheres with radii λ_i and λ_o such that $\lambda_i < \lambda_o$, as shown in Figure 2 (note that coordinate axes have been rotated for display purposes). The top of this volume is cut off by restricting η to the range $0 \le \eta \le 120^o$ in order to better approximate the shape of the LV. Thus, a point (x, y, z) is defined to lie inside the *model LV* if its prolate spheroidal coordinates satisfy $\lambda_i \le \lambda \le \lambda_o$ and $0 \le \eta \le 120^o$. This model LV defines the material points for the motion, described next.

2.2 Motion Model

Having selected a geometric model of the LV, it is now necessary to model the motion of the LV. We have chosen to implement the motion model designed by Arts et al. [1] because of the wide variety of intricate motions it can provide. Complex motion is achieved in this model by applying a series of thirteen three-dimensional transformations controlled by thirteen motion parameters, k_1 through k_{13}.

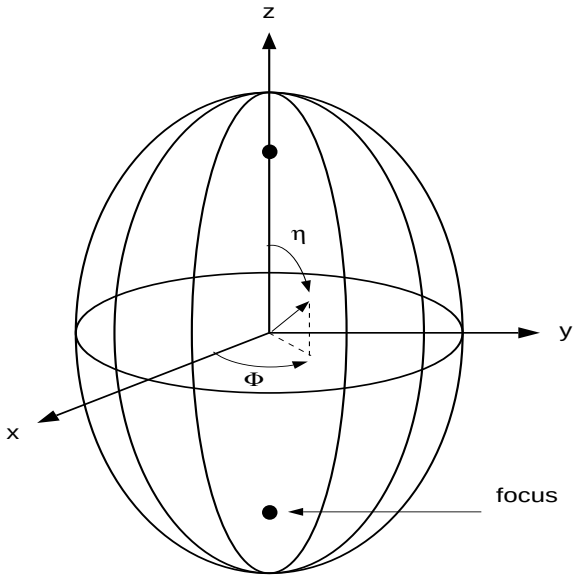

Figure 1. A prolate sphere, centered at the origin and aligned along the z-axis..

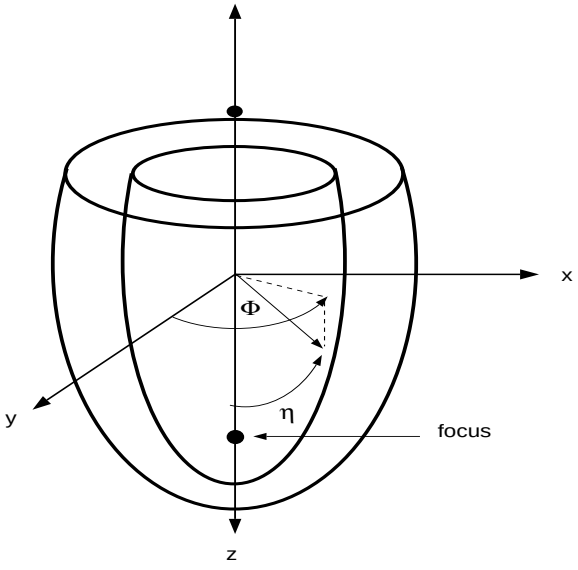

Figure 2. Prolate spheroidal model for the left ventricular myocardium.

Table 1. Description of the thirteen k-parameters.

k_1	Radially dependent compression
k_2	Left ventricular torsion
k_3	Ellipticallization in long axis planes
k_4	Ellipticallization in short axis planes
k_5	Shear in x direction
k_6	Shear in y direction
k_7	Shear in z direction
k_8	Rotation about x-axis
k_9	Rotation about y-axis
k_{10}	Rotation about z-axis
k_{11}	Translation in x direction
k_{12}	Translation in y direction
k_{13}	Translation in z direction

Table 1 lists these parameters, referred to as k-parameters, and the transformations to which they correspond.

All transformations listed in Table 1 are three-dimensional, but we have introduced a fourth dimension which allows us to express the overall transformation as a single matrix map. The fourth dimension is incorporated in order to allow translations to be expressed as matrix transformations. This technique improves computational efficiency and simplifies the overall algorithm. It does not present any theoretical improvement on the motion model itself, but is merely a mathematical convenience.

Motion in the left ventricle is specified through a transformation which maps a *material point* to a corresponding *spatial point* at time t. A material point is a specific location defined on the model LV, and the spatial point is its location in space at a certain time instant. As the LV deforms, the spatial coordinates of each material point change. We express this mathematically as $\mathbf{r} = r(\mathbf{p}, t)$, where the material point $\mathbf{p}$ moves to the spatial point $\mathbf{r}$ at time t. The *reference map* is the inverse transformation $\mathbf{p} = p(\mathbf{r}, t)$.

The overall matrix equation that transforms point $\mathbf{p} = (p_x, p_y, p_z, 1)$, a point inside the model LV, into $\mathbf{r} = (r_x, r_y, r_z, 1)$ is:

$$\mathbf{r} = F_a F_6 F_5 F_4 F_3 F_2 F_1 F_0 \mathbf{p}\,, \tag{3}$$

where

$$F_0 = \begin{bmatrix} a^{1/3} & 0 & 0 & 0 \\ 0 & a^{1/3} & 0 & 0 \\ 0 & 0 & a^{-2/3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F_1 = \begin{bmatrix} \epsilon & 0 & 0 & 0 \\ 0 & \epsilon & 0 & 0 \\ 0 & 0 & \epsilon & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\epsilon = \sqrt[3]{1 + \frac{3k_1 V_w}{4\pi |F_0 \mathbf{p}|^3}}$$

$$F_2 = \begin{bmatrix} \cos ak_2 z_1/|\mathbf{r}_1| & -\sin ak_2 z_1/|\mathbf{r}_1| & 0 & 0 \\ \sin ak_2 z_1/|\mathbf{r}_1| & \cos ak_2 z_1/|\mathbf{r}_1| & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{r}_1 = F_1 F_0 \mathbf{p} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} a^{-1/3} e^{k_4 - (k_3/2)} & 0 & 0 & 0 \\ 0 & a^{-1/3} e^{-k_4 - (k_3/2)} & 0 & 0 \\ 0 & 0 & a^{2/3} e^{k_3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} 1 & k_5 & 0 & 0 \\ k_5 & 1 + {k_5}^2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F_5 = \begin{bmatrix} 1 & 0 & k_6 & 0 \\ 0 & 1 & 0 & 0 \\ k_6 & 0 & 1 + {k_6}^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & k_7 & 0 \\ 0 & k_7 & 1 + {k_7}^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F_a = A_4 A_3 A_2 A_1$$

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos k_8 & -\sin k_8 & 0 \\ 0 & \sin k_8 & \cos k_8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \cos k_9 & 0 & \sin k_9 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin k_9 & 0 & \cos k_9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} \cos k_{10} & -\sin k_{10} & 0 & 0 \\ \sin k_{10} & \cos k_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & k_{11} \\ 0 & 1 & 0 & k_{12} \\ 0 & 0 & 1 & k_{13} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It is possible to deduce what each transformation does by using Table 1, and referring to Reference [1].

Aside from the k-parameters, there are two constants in this model: a and V_w. V_w is the wall volume of the model LV. The constant a is a correctional parameter present in matrix F_0. F_0 transforms a prolate spheroidal shell into a more spherical shape in anticipation of the next transformation, which is a radially dependent compression in spherical coordinates. Ideally, we want all points on the surface of the prolate sphere to compress uniformly, but points in the model LV can have different spherical radii, in general, and nonuniform compression will result. By converting the model LV to a more spherical shape we reduce this undesirable distortion. The effect of F_0 is undone in matrix F_3.

In order for the motion model to be useful one needs the ability to generate realistic cardiac motion. Since a medical imaging sequence involves a discrete set of time frames, the thirteen k-parameters must be determined for each of these frames. This can be characterized as a parameter estimation problem, where the k-parameters are estimated so that the model LV is transformed as close as possible to the actual state of the LV for each time frame. One approach to this problem was presented by Arts et al. [1], where a least squares criterion is minimized using the Levenberg-Marquardt method.

3 A Computational Model

3.1 Overview

The computational model that we have developed first generates a three-dimensional LV model and applies a motion. Next, an imaging plane is selected to intersect the (deformed) LV. An image is generated by selecting points on the imaging plane and assigning them a value which depends on whether they lie inside or outside the LV volume. If they lie outside, they are assigned a value of 0, otherwise they are assigned a value determined by a spatial intensity pattern and an MR imaging equation. Motion estimation algorithms can then be applied to these images.

Testing the effectiveness of a motion estimation algorithm requires two additional pieces of information which must be generated by our computational model: a segmentation, which determines which pixels are inside the LV wall; and a displacement field, which represents the true displacement of the model. The displacement field can be compared with that generated by the motion estimation algorithm, and statistics such as mean square error can then be calculated.

Development of the computational model roughly parallels the steps of generating an actual series of MR images, providing a realistic three-dimensional simulation. But some motion estimation algorithms deal only with two-dimensional motion within the image plane. A three-dimensional simulation would be inappropriate for these algorithms because the LV is moving through the plane, generating an additional displacement component which cannot be accounted for in two-dimensional algorithms. The displacement fields generated by these two methods cannot be compared. Therefore, adjustments have to be made to the simulator in order to create a feasible two-dimensional control experiment. We will first discuss the three-dimensional model, and then outline the adjustments necessary to implement a two-dimensional motion simulator.

3.2 Additional Computations

Several issues have to be resolved before we can begin to develop the computational model. The first issue involves what values to assign the constants V_w and a. A closed form equation for V_w was obtained by integrating in prolate spheroidal coordinates over the boundary $\lambda_i \leq \lambda \leq \lambda_o$, $0 \leq \eta \leq 120, 0 \leq \phi \leq 360$. The result is:

$$V_w = \frac{\pi\delta^3}{4} \times \left(3(\cosh\lambda_o - \cosh\lambda_i) + 4(\cosh^3\lambda_o - \cosh^3\lambda_i)\right) . \quad (4)$$

Assigning a value for a is slightly more complicated. For a point with prolate spheroidal coordinates (λ, η, ϕ) the ideal value for a can be determined to be:

$$a = \frac{\cosh\lambda}{\sinh\lambda} . \quad (5)$$

This value transforms a prolate sphere with radius λ into a sphere. But a dilemma arises when deforming a point inside confocal prolate spheres. Points within this volume have λ values between λ_i and λ_o; therefore, the exact value of a changes within the LV. Our solution is to find the appropriate value of a for the inner and outer shells, and take an average, so that it is approximately correct for all shells within the volume. There may be more accurate methods to select a, but this technique has proven to be quite effective experimentally.

A third issue that needs to be addressed is the calculation of the reference map. When dealing with nonlinear transformations it is often difficult to find the reference map, and one must settle for an approximation using iterative methods or other techniques. In this case, however, an explicit expression can be derived; it is given by

$$\mathbf{p} = R_0 R_1 R_2 R_L \mathbf{r} , \quad (6)$$

where

$$\begin{aligned} R_0 &= {F_0}^{-1} , \\ R_1 &= {F_1}^{-1} , \\ R_2 &= {F_2}^{-1} , \\ R_L &= \left[F_a F_6 F_5 F_4 F_3\right]^{-1} . \end{aligned}$$

Determining R_0 and R_L is straightforward, because these are inverses of constant-valued, non-singular, matrices, and can be calculated numerically. The matrices F_1 and F_2 represent nonlinear transformations, however, so their inverses are a little more difficult to calculate. To find the matrix R_2 it is necessary to notice that the operation of F_2 is a rotation around the z-axis. We define $\mathbf{r}_1$ and $\mathbf{r}_2$ as follows:

$$\begin{aligned} \mathbf{r}_1 &= F_1 F_0 \mathbf{p}\,, \\ \mathbf{r}_2 &= F_2 F_1 F_0 \mathbf{p}\,. \end{aligned}$$

Since these two vectors differ only by a rotation around the z-axis, it follows that $z_1/|\mathbf{r}_1| = z_2/|\mathbf{r}_2|$, where z_1 and z_2 are the z-components of $\mathbf{r}_1$ and $\mathbf{r}_2$ respectively. Therefore,

$$R_2 = \begin{bmatrix} \cos ak_2 z_2/|\mathbf{r}_2| & \sin ak_2 z_2/|\mathbf{r}_2| & 0 & 0 \\ -\sin ak_2 z_2/|\mathbf{r}_2| & \cos ak_2 z_2/|\mathbf{r}_2| & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

Now we define $\mathbf{r}_0$:

$$\mathbf{r}_0 = F_0 \mathbf{p}\,.$$

Since $\mathbf{r}_0$ and $\mathbf{r}_1$ differ only by a scalar multiple in three dimensions, only the magnitudes of the vectors change. We can express this change of magnitude as follows:

$$|\mathbf{r}_1| = |\mathbf{r}_0| \sqrt[3]{1 + \frac{3k_1 V_w}{4\pi |\mathbf{r}_0|^3}}\,.$$

which can be rearranged to get:

$$|\mathbf{r}_0| = |\mathbf{r}_1| \sqrt[3]{1 - \frac{3k_1 V_w}{4\pi |\mathbf{r}_1|^3}}\,.$$

Thus, we can express the matrix R_1 as follows:

$$R_1 = \begin{bmatrix} \hat{\epsilon} & 0 & 0 & 0 \\ 0 & \hat{\epsilon} & 0 & 0 \\ 0 & 0 & \hat{\epsilon} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

where

$$\hat{\epsilon} = \sqrt[3]{1 - \frac{3k_1 V_w}{4\pi |\mathbf{r}_1|^3}}\,.$$

3.3 3-D Simulation

We begin with the three-dimensional simulator because it is simpler than the two-dimensional model. A regular grid of points is first selected on the imaging plane which, ideally, encompasses the intersection of LV with the imaging plane. Next, we define a set of time instants $t_0 < t_1 < \ldots < t_n$. Time t_0 is referred to as the *tag reference time*, and is the time when the tag pattern is initially applied (usually end-diastole). Next, given a spatial point $\mathbf{r}_i$ at time t_i, we must define the following function:

$$\mu = \Upsilon\left(\mathbf{r}_i, t_i\right) \tag{9}$$

where μ is the intensity value corresponding to point $\mathbf{r}_i$ at time t_i.

The value of μ depends on whether $\mathbf{r}_i$ is inside or outside the LV. The procedure for determining this is referred to as *segmentation*. The spatial point $\mathbf{r}_i$ is inside the LV if its corresponding material point $\mathbf{p}_i = p\left(\mathbf{r}_i, t_i\right)$ is inside the model LV. That is, if $\mathbf{p}_i = (\lambda, \eta, \phi)$ where $\lambda_i \leq \lambda \leq \lambda_o$ and $0 \leq \eta \leq 120$, then $\mathbf{r}_i$ is inside the LV. If $\mathbf{r}_i$ is outside the LV, then $\mu = 0$.

In order to determine μ when $\mathbf{r}_i$ is inside the LV, we must know where it came from at time t_0. This relationship is given by

$$\mathbf{r}_0 = r\left(p\left(\mathbf{r}_i, t_i\right), t_0\right) \tag{10}$$

The value of μ is given by the following tagged spin-echo MR imaging equation [6]

$$\begin{aligned} \mu = {} & D_0 e^{-T_E/T_2} \\ & \times \left\{1 + \left[\left(1 - e^{-(T_R - T_d)/T_1}\right) \xi\left(\mathbf{r}_0\right) - 1\right] e^{-T_d/T_1}\right\}, \end{aligned} \tag{11}$$

although other equations are readily incorporated. Here, $T_d = t_i - t_0$, and the constants D_0, T_1, and T_2 (the spin density, longitudinal relaxation time, and transverse relaxation time) are determined by the material properties of the LV being modeled and must be specified. The imaging parameters T_E and T_R (the echo time and the pulse repetition time) must also be specified. The function $\xi(\mathbf{r}_0)$ is the applied spatial tag pattern. In our simulation we used a two-dimensional SPAMM (spatial modulation of magnetization) tag pattern, characterized by the following equation [6]

$$\begin{aligned} \xi\left(\mathbf{r}\right) = {} & \left(\cos^2\theta - \sin^2\theta \cos k_x r_x\right) \\ & \times \left(\cos^2\theta - \sin^2\theta \cos k_y r_y\right)\,. \end{aligned} \tag{12}$$

Here, k_x and k_y are radial spatial frequencies in the x and y directions respectively, and θ is the tag pattern tip angle. Other tag patterns can be easily substituted into the simulation.

Having developed the mathematics to synthesize the images, our next step is to generate a truth displacement field; that is, a a vector field indicating the motion from one frame to the next. Given a point $\mathbf{r}_i$ at time t_i, we want to find the point $\mathbf{r}_{i+1}$ to which point $\mathbf{r}_i$ moves at time t_{i+1}. The expression for this is

$$\mathbf{r}_{i+1} = r\left(p\left(\mathbf{r}_i, t_i\right), t_{i+1}\right)\,. \tag{13}$$

The displacement field is defined as $d_i(\mathbf{r}_i, t_i) = \mathbf{r}_{i+1} - \mathbf{r}_i$.

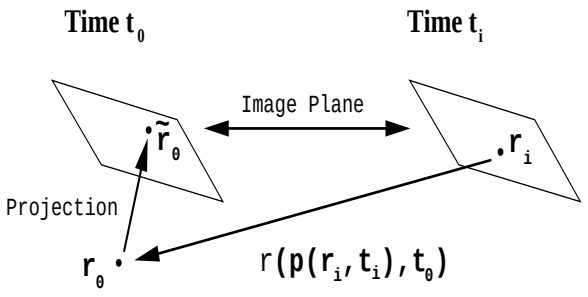

Figure 3. Pattern application in synthetic MR image.

3.4 2-D Model

Using the procedure just outlined, one can generate a full image sequence comprising both short-axis images and long-axis images, and test any three-dimensional motion estimation algorithm. But many algorithms only work in a two-dimensional setting using one imaging plane. These require a control experiment in which the simulated motion is two-dimensional. In other words, we want to generate a two-dimensional object and deform it so that each point on the object remains in the plane.

Again, we must define the value for μ in Equation 9 for all $\mathbf{r}_i$ at all times t_i. If $\mathbf{r}_i$ is outside the LV then $\mu = 0$ still holds. But the procedure for finding μ when $\mathbf{r}_i$ is in the LV must be modified. In the three-dimensional case we found the point $\mathbf{r}_0$, and used it to find the proper intensity using a tag pattern and MR imaging equation. But in general, $\mathbf{r}_0$ will not be located on the imaging plane. Therefore, we replace it with the point $\tilde{\mathbf{r}}_0$, the projection of $\mathbf{r}_0$ onto the imaging plane; Figure 3 illustrates this procedure. Instead of using $\mathbf{r}_0$ in Equation (11), we use $\tilde{\mathbf{r}}_0$, thereby forcing the motion to be two-dimensional.

The segmentation method must also be modified. In the three-dimensional case, we said that $\mathbf{r}_i$ is in the LV if its corresponding material point $\mathbf{p}_i$ is in the model LV. Here, we define $\mathbf{r}_i$ to be in the LV if $\tilde{\mathbf{r}}_0$ is in the LV, which is equivalent to asking whether $\tilde{\mathbf{p}}_0 = p(\tilde{\mathbf{r}}_0, t_0)$ is in the model LV.

The biggest mathematical challenge in this 2-D simulation is in defining the displacement. Given a spatial point $\mathbf{r}_i$ at one time t_i, where does it move to at time t_{i+1}? Chances are that the actual material point corresponding to our spatial point moves off the plane. But we are no longer interested in the actual material point. Any point which, when shifted back to t_0 and projected to the imaging plane, lands on $\tilde{\mathbf{r}}_0$ will carry the exact same intensity as that of $\mathbf{r}_{i+1}$. There are an infinite number of points which project onto $\tilde{\mathbf{r}}_0$, and they all lie on a line which is perpendicular to the imaging plane and passing through point $\mathbf{r}_0$. If we assume that our motion is not pathological, then exactly one of these points will intersect the imaging plane at each time instant. We find this point computationally by doing a line search using the Golden Section Search [5], using $\mathbf{r}_0$ as an initial guess. Our minimization criterion is the magnitude of the distance vector from the estimated $\tilde{\mathbf{r}}_{i+1}$ to the image plane. The resulting 2-D displacement field is given by $d_i(\mathbf{r}_i, t_i) = \tilde{\mathbf{r}}_{i+1} - \mathbf{r}_i$.

4 Simulation Results

The k-parameters underlying our simulation were derived from a bead experiment on a dog heart [1]. They are shown for 60 different times within a single cardiac cycle in Figure 4. This figure shows the k-parameters as a function of the frame number 1 through 60. In a 1 second cardiac cycle, the time frames are equally spaced in steps of 0.017 seconds. All k-parameters are unitless except for the rotational parameters k_8 through k_{10}, which are in units of radians, and the translational parameters k_{11} through k_{13}, which are in units of centimeters. The six columns labeled (a)–(f) indicate key times at which we will show results below. Column (a) is approximately at end-diastole and is also the tag reference frame, while column (e) approximately represents end-systole. Table 2 lists the other key constants used in our simulation and their values.

Figure 5 shows the three-dimensional model at six phases of cardiac deformation. The image plane is also shown. The images generated along this imaging plane are shown in Figure 6. These images are generated for two-dimensional simulation. The tag patterns are seen to clearly deform as time progresses, and the effect of tag fading is also apparent (images get brighter overall).

Finally, the 2-D displacement fields for these times are shown in Figure 7. Each arrow represents the direction of displacement, and the size of the arrow corresponds to the magnitude of the displacements. The arrow lengths have been rescaled for visual enhancement, and do not actually point to the spatial location to which the points move. A larger arrow does however mean that a greater displacement has taken place. As can be seen, the simulator is capable of generating a relatively continuous and intricate displacement field.

5 Discussion

We have presented a LV motion simulator capable of generating tagged MR image studies used in motion estimation. This simulator effectively works in two modes, a three-dimensional and a two-dimensional mode. The mode of operation depends on the type of motion estimation algorithm being tested.

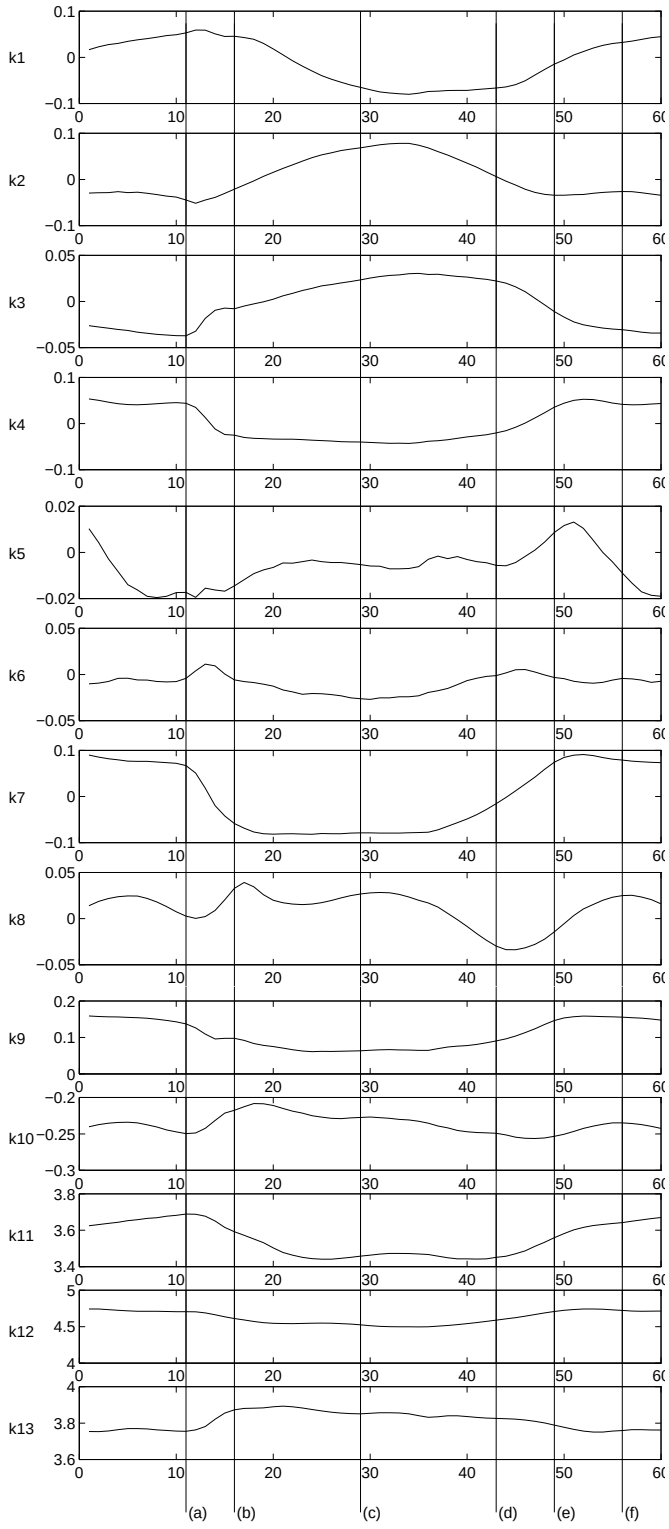

Figure 4. Estimated k-parameters. Vertical lines indicate the time instants used in future plots. Column (a) represents end-diastole and is also the tag reference frame, and column (e) is end-systole.

Table 2. Constant values

Constant	Description	value	units
λ_i	inner radius	0.35	none
λ_0	outer radius	0.55	none
δ	focal radius	4.00	cm
D_0	spin density	300.0	AU *
T_E	echo time	0.03	sec
T_R	pulse repetition time	10.0	sec
T_1	longitudinal relaxation time	0.60	sec
T_2	transverse relaxation time	0.10	sec
k_x	frequency in radial x	8.00	rad/cm
k_y	frequency in radial y	8.00	rad/cm
θ	tip angle of tag pattern	45.0	degrees

* Arbitrary units.

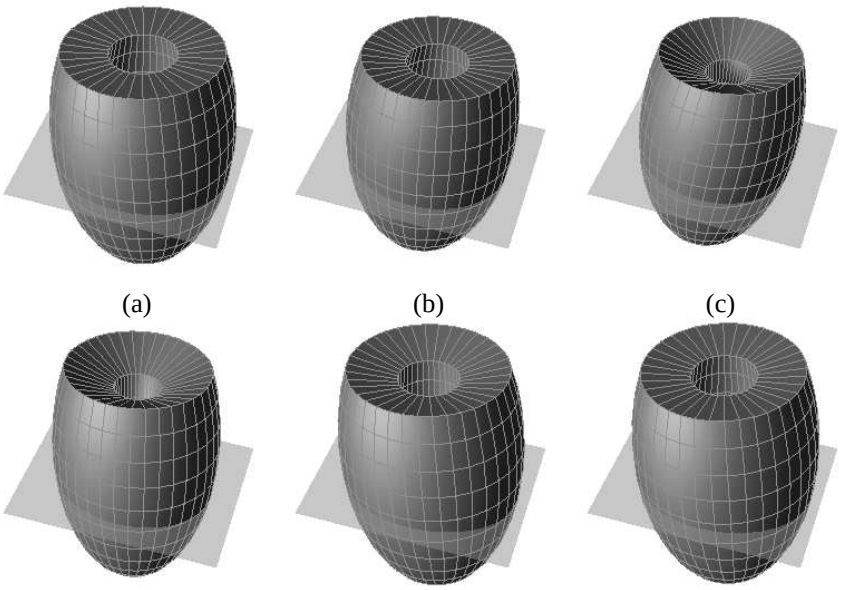

Figure 5. Three dimensional model and imaging plane shown at the time instants depicted in Figure 4

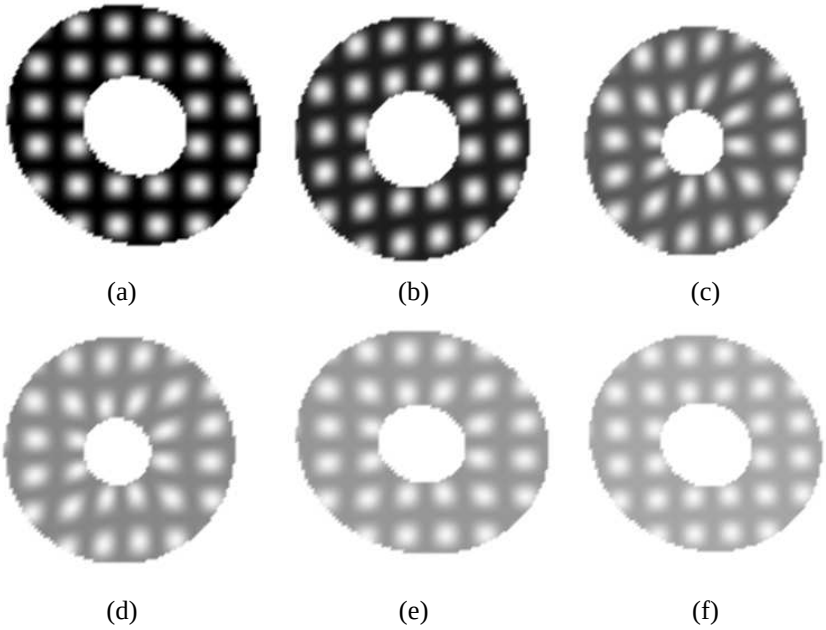

Figure 6. Synthesized MR images for the time instants depicted in Figure 4. The background is shown as white for clarity.

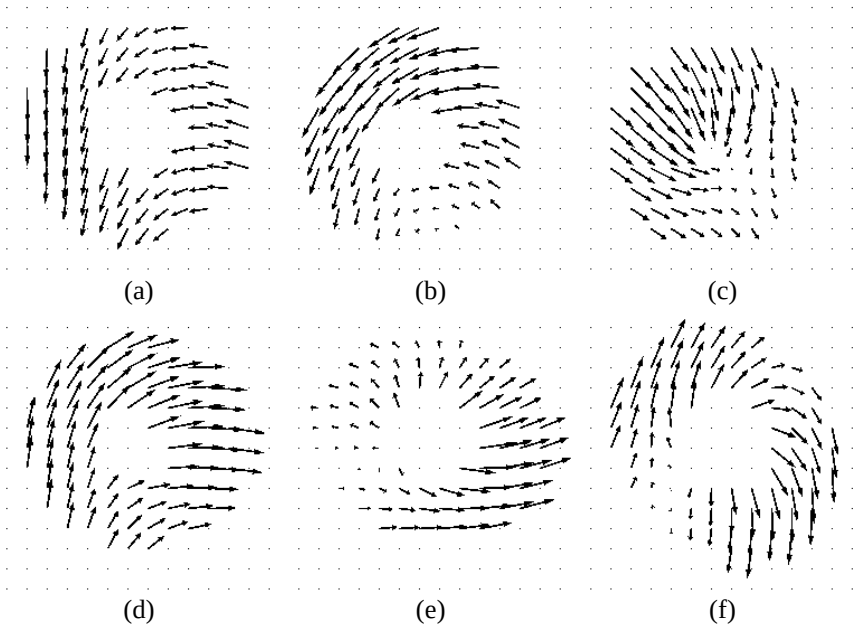

Figure 7. Displacement fields for the different time instants depicted in Figure 4.

Although the simulator functions well in its current stage, several issues remain to be addressed. The issue of parameter estimation has still not been completely resolved. As stated earlier, the simulation parameters were obtained from bead data of a dog heart. It would be very useful to be able to estimate these parameters from cardiac MR images. This approach is non-invasive and would allow us to work with human heart data. One of our next steps will be to try to develop a parameter estimation method using this kind of data.

There are also several additions to the simulator which would improve the overall realism of the output. These include adding a background, noise model, as well as a point spread function. Furthermore, there are countless imaginative improvements which would help generate more representative experiments. With a little ingenuity, we are confident that this simulator can be modified to generate data which can be used to test a wide variety of cardiac motion estimation algorithms.

References

[1] T. Arts, W. Hunter, A. Douglas, A. Muijtjens, and R. Reneman. Description of the deformation of the left ventricle by a kinematic model. *J. Biomechanics*, 25(10):1119–1127, 1992.

[2] T. S. Denney 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.*, 14(4):625–635, December 1996.

[3] F. G. Meyer, R. T. Constable, A. G. Sinusas, and J. Duncan. Tracking myocardial deformation using spatially constrained velocities. In Y. Bizais, C. Barillot, and R. D. Paola, editors, *Information Processing in Medical Imaging*. Kluwer, 1995.

[4] W. G. O'Dell, C. C. Moore, W. Hunter, E. A. Zerhouni, and E. R. McVeigh. Displacement field fitting for calculating 3-d myocardial deformations from tagged MR images. *Radiology*, 1995. submitted.

[5] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[6] J. Prince and E. McVeigh. Motion estimation from tagged MR image sequences. *IEEE Transactions on Medical Imaging*, 11(2):238–249, June 1992.

[7] 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.*, 14(3):413–421, 1995.

[8] E. A. Zerhouni, D. M. Parish, W. J. Rogers, A. Yangand, and E. P. Shapiro. Human heart: tagging with MR imaging — a method for noninvasive assessment of myocardial motion. *Radiology*, 169:59–63, 1988.