DEFORMABLE MESH MODEL FOR CARDIAC MOTION ESTIMATION FROM MRI DATA

by

Alex González Paredes

FINAL RESEARCH PROJECT

Approved by: _____

Jovan G.Brankov,PhD

ABSTRACT

The use of technology in medical applications has expanded tremendously in the last 50 years, playing an important role in diagnosis and treatment of disease. As a consequence of this growth, it has been possible to develop new techniques and processes which allows scientists and physicians to extract potentially life-saving information by peering noninvasively into the human body. One of these techniques, Magnetic Resonance Imaging, has the capacity of providing data which contain information about motion of human tissue. In this project we exploit this characteristic to present a model for estimating left ventricular heart motion through a whole cardiac cycle. The model is based on the utilization of a non-rigid deformable mesh which tracks the intensity variations on the datasets of MRI images. The use of a mesh allows us to interpolate the motion on every point of the myocardial tissue. Although still in his first steps of development, preliminary results here presented demonstrate great potential of our model in terms of efficiency and flexibility versus the other existing models. It constitutes a good alternative for developing full 4D heart models in the future.

ACKNOWLEDGMENTS

First of all, I would like to thank the Illinois Institute of Technology and the Polytechnic University of Catalonia for his strong relation and cooperation in mobility of students, as well as the “Fundación Vodafone España” for their financial support.

I would also like to thank my advisor, **Dr. Jovan Brankov**, for accepting me as a Research Student in MIRC and for giving me the opportunity to learn a lot about this field.

I cannot leave Chicago without mentioning **Sergi Orrit** and **Miguel Lauzurica**. Thank you for sharing with me this great experience and for your warmness.

I would like to thank **my parents** and **my brother**, for their fully support in everything, for turning me into the person I am today and for their unconditional love.

Finally, I would like to thank **Olalla** for believing in me and overcoming this challenge.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
DMM	Deformable Mesh Model
ED	End of Diastole timeframe
ES	End of Systole timeframe
LA	Long Axis
LA2CH	Long Axis 2 Chamber
LA4CH	Long Axis 4 Chamber
LV	Left Ventricle
MF	Motion Field
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
SA	Short Axis

1. Introduction to medical imaging

1.1. Overview

The use of technology in medical applications has expanded tremendously in the last years, playing an increasingly prominent role in the diagnosis and treatment of disease. A key contribution to the growth of medical technology is the application of basic science and engineering. As a consequence, it has been possible to develop a collection of techniques and processes which allows scientists and physicians to extract potentially life-saving information by peering noninvasively into the human body. This new field is known as “medical imaging”.

The role of medical imaging has expanded beyond the simple visualization and inspection of anatomic structures. It has become a tool for surgical planning and simulation, intra-operative navigation, radiotherapy planning, and for tracking the progress of disease.

The medical image analysis community has become preoccupied with the challenging problem of extracting, with the assistance of computers, clinically useful information about anatomic structures imaged through Computed Tomography (CT), Magnetic Resonance (MR), Positron Emission Tomography (PET), and other modalities. Although modern imaging devices provide exceptional views of internal anatomy, the use of computers to quantify and analyze the embedded structures with accuracy and efficiency is limited. Accurate, repeatable, quantitative data must be

efficiently extracted in order to support the spectrum of biomedical investigations and clinical activities from diagnosis, to radiotherapy, to surgery.

1.2. Basics of Magnetic Resonance Imaging (MRI)

Magnetic resonance imaging (MRI) is a relatively new medical imaging technique primarily used in radiology to generate anatomical and functional images of the human body. Unlike other medical imaging techniques, MRI does not use ionizing radiation and therefore is considered safer than many other techniques.

MRI generates images with excellent soft tissue contrast and thus is particularly useful for neurological, musculoskeletal, cardiovascular, and oncological imaging. Although not as sensitive as PET or SPECT, or as fast as CT, MRI is a very versatile technique able to generate great variety of image contrasts for a wide range of clinical and research applications. MRI detects signals predominantly from hydrogen nuclei (i.e., protons) in water or fat molecules. MRI signal acquisition is based on the phenomenon of nuclear magnetic resonance (NMR), which deals with the interactions between nuclear spins and magnetic fields. Signal localization in MRI is achieved by the application of linear gradients of a magnetic field.

Figure 1.1. MRI machine [1]

In the experience with in vivo nuclear magnetic resonance, an intrinsic sensitivity of MRI to motion was early recognized before the development of MRI per se. This motion sensitivity stems primarily from two sources:

- There is persistence (within the limits of the relaxation times) of local alterations in the magnetization of material (such as the heart wall), even in the presence of motion; this allows us to deliberately produce local perturbations of the magnetization of the heart wall that will be visible in the images and will serve as material tags within the heart wall.

- The motion of excited (signal-producing) nuclei in the presence of the magnetic field gradients used during MRI data acquisition has the potential to induce a phase shift of their signal due to the motion of the nuclei, e.g., due to heart wall motion, that is related to the velocity of or distance traveled by the nuclei. While these effects can be seen in the course of the performance of conventional MRI of the heart, we can also design modified MRI pulse sequences to produce

images that specifically exploit these effects, in order to use them to quantitatively assess motion.

Although these MRI motion characterization methods have previously been primarily applied to the study of blood flow, they can also be used to study the motion of the heart wall in what we call “myocardial tagging”.

The principle of myocardial tagging is based on producing a spatial pattern of saturated magnetization within the heart wall, e.g., at end diastole, and then imaging the resulting deformation of the pattern as the heart contracts through the cardiac cycle. These tagged patterns appear initially in darker colors in the tagged MR images. Generally, they will persist for a short period of time in the myocardium (milliseconds) and deform with the underlying tissue during the cardiac cycle in vivo, which provides detailed myocardial motion information.

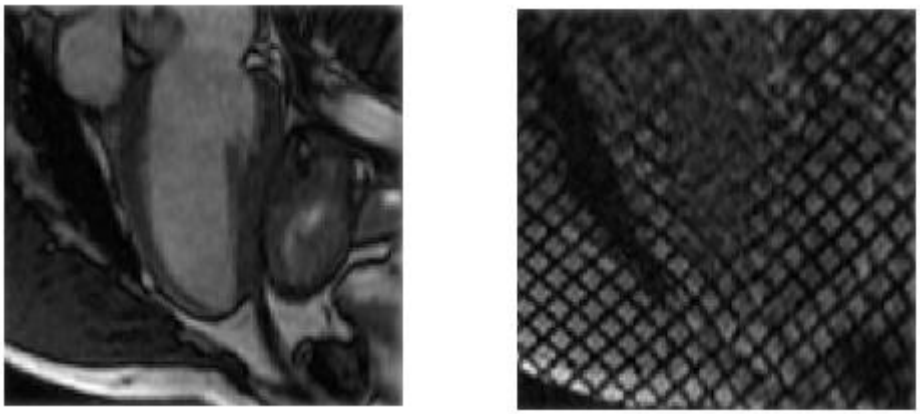

Figure 1.2. Cine (left) and Tagged (right) MRI images

The deformation of parallel tagging planes provides only one-dimensional motion of the myocardium. In order to analyze the entire 3D motion of the heart, three

independent tagging directions must be analyzed and combined. However, in case of utilizing a grid pattern as the one in Figure 1.2., we can obtain 2D motion simultaneously by tracking the tag intersections of that grid.

1.3. Objectives

In this project we will try to develop a model to estimate left ventricular heart motion through a whole cycle, based upon the utilization of a non-rigid deformable mesh which tracks the intensity variations on datasets of MRI images.

For our proposal, three sets of tagged MRI images from a normal subject are used to analyze the heart motion. These sets of data span the left ventricle (LV) and were acquired for twenty timeframes over the cardiac cycle. The data includes one grid-patterned set of short-axis (SA) images and two sets of long-axis (LA) images. The SA image set consists of fifteen SA image planes, from base to apex, as seen in Figure 1.3. The LA image sets were radially oriented on 0º and 90º, thus obtaining the long-axis 4 chamber view (LA4CH) and the long-axis 2 chamber view (LA2CH) respectively. Heart deformation on the x-y plane will be extracted from the SA views. The taglines in the LA image volume were designed to be parallel to the SA image planes, in order to analyze the z deformation. In combination, the taglines of the three sets of image volumes form a regular 3D grid from which to analyze the 3D motion of the heart.

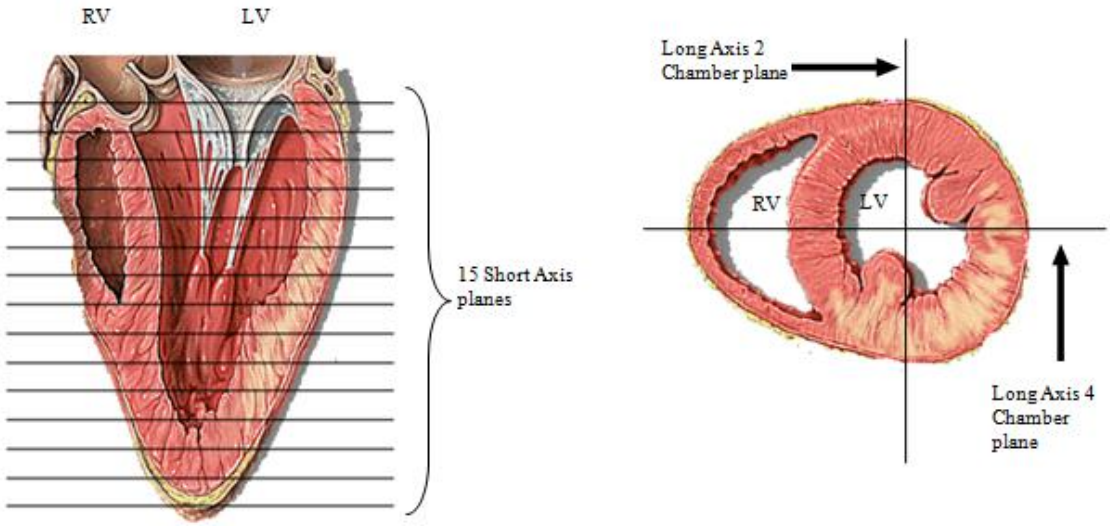

Figure 1.3. Sets of images. Left: Set of 15 short axis planes spanning the left ventricle. Right: Two perpendicular sets for long axis planes. [2]

However, intensities of LV walls on tagged MRI appear strongly masked by the tags overlaying. This causes the mesh motion estimation to be much more sensitive to the displacement of the tags than to the LV walls contraction, thus leading to an incomplete estimation of the heart motion. In order to solve this problem, we will combine the tagged MR images with another set of images, the "cine MRI data". These images contain information about the contraction of the myocardium and will help the deformable mesh to follows the LV walls.

The composition of cine and tagged MRI data consists in a simple addition of the intensities of their pixels (weighted with parameters that can be controlled by the user), obtaining a set of images which incorporates information of both of them.

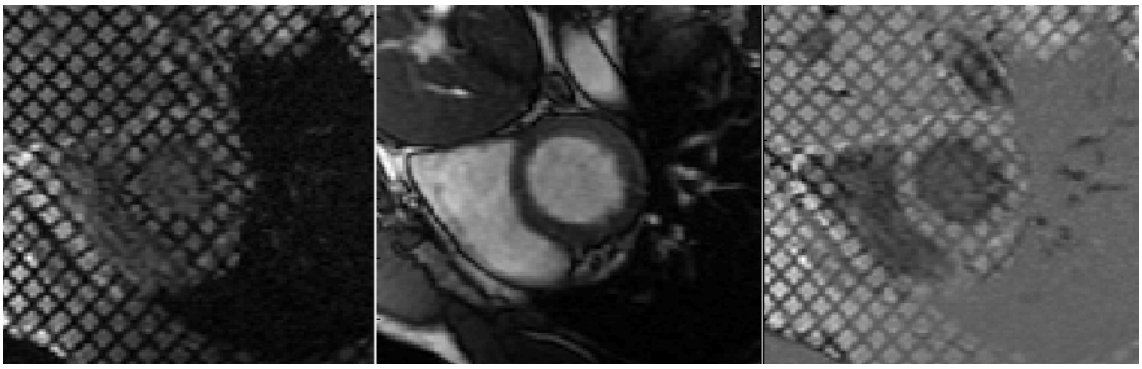

Figure 1.4. Composition of tagged and cine MRI. Left: tagged image. Middle: cine image. Right: Composition of both tagged and cine images.

In the following chapters we will describe carefully the basic principles of the deformable mesh, his implementation and the motion estimation results we had with our method.

2. Deformable Mesh Model (DMM)

2.1. Basic Principles

In this project we propose the use of a non-rigid deformable mesh model (DMM) as an alternative to conventional pixel-based methods of motion estimation.

In a mesh model, the domain is partitioned into non-overlapping polygonal regions, called mesh elements, the vertices of which are called nodes. The aim is to obtain the displacement vector field of the nodes over time, tracking the variation of intensities from frame to frame. As a consequence of this displacement the mesh elements deform describing motion.

2.2. Mathematical Model

2.2.1. Estimation of nodal displacements

We describe a mathematical method for estimating a motion field (MF), $\mathbf{d}_{k\to l}(\mathbf{x})$ by deforming the mesh from the current frame k to the target frame l. Specifically, given the nodal positions $\mathbf{X}_k = \left\{\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_N^{(k)}\right\}$ for frame k, the goal is to determine their corresponding locations $\mathbf{X}_l$ in frame l.

In the proposed method, motion estimation is accomplished by starting with a mesh structure for the current frame k and deforming it to determine a mesh structure

for the target frame l . Let $\mathbf{X}_k = \left\{\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{\mathrm{N}}^{(k)}\right\}$ be the set of locations of the N nodes that constitute the mesh for frame k, and let $\mathbf{X}_l = \left\{\mathbf{x}_1^{(l)}, \ldots, \mathbf{x}_{\mathrm{N}}^{(l)}\right\}$ be the set of corresponding nodal positions for frame l. The nodal values of the mesh representation of the MF are given by $\mathbf{d}_n^{(k,l)} = \mathbf{x}_n^{(l)} - \mathbf{x}_n^{(k)}$, $n = 1, \ldots, N$. This vector represents the displacement of node n from frame k to frame l.

We accomplish this by utilizing a variation of an algorithm proposed by Wang and Lee [2]-[3], in which the deformation of the mesh from the current frame to the target frame is achieved by minimizing a penalized squared error matching criterion to find the target frame nodal positions $\mathbf{X}_l$.

We use the following mesh-element matching criterion:

$$\mathrm{E}_{k\to l} = (1-\lambda)\cdot \sum_{m=1}^{M}\left[\int_{D_m}\left[f_k\left(\mathbf{x}+\mathbf{d}_{k\to l}(\mathbf{x})\right) - f_l(\mathbf{x})\right]^2 d\mathbf{x}\right] + \lambda\cdot \mathrm{ED}_{k\to 0} \tag{2.1}$$

where $\mathbf{d}_{k\to l}(\mathbf{x})$ denotes the displacement vector at pixel **x** describing motion from frame k to frame l, D_m is the domain of the mth mesh element, M is the number of mesh elements. The expression (3.1) is then minimized from frame k to frame $k+1$ by means of an iterative gradient descending algorithm.

Penalization factor ($\lambda\cdot \mathrm{ED}_{k\to 0}$):

$\lambda\cdot \mathrm{ED}_{k\to 0}$ is a penalization factor given as:

$$\mathrm{ED}_{k\to 0} = \sum_{n=1}^{N} \left\| \mathbf{x}_n^{(k)} - \mathbf{x}_n^{(0)} \right\|^2 \tag{2.2}$$

$$\lambda = \left| \frac{k}{K} \right|^{\Upsilon} \tag{2.3}$$

where N is the total number of mesh nodes in the image, k is the current frame being processed, and K is a constant.

It can be clearly seen in Equation (2.2) that $\mathrm{ED}_{k\to 0}$ corresponds to the square error between the current mesh at frame k and the mesh at time zero. Therefore, the penalty term $\mathrm{ED}_{k\to 0}$ enforces similarity of mesh model in the kth frame and the first frame in the sequence. Likewise, Υ and λ are parameters controlling the shape and strength of the penalization through the time sequence.

The factor $\lambda \cdot \mathrm{ED}_{k\to 0}$ is defined in order to achieve two goals:

- To gain control over the noise: The quality of taglines decreases as function of time, becoming difficult to track the intensities through the last frames. The penalty term progressively forces the mesh at the last frames to return to its initial position, therefore making the algorithm more insensitive to the lack of quality of tags.

- To ensure the periodicity of DMM: It is obvious that the mesh motion has to be periodic in the whole time cycle. Therefore, to force the mesh to finish the sequence in the same position as it started is a way to ensure motion continuity and smoothness between the last frame and the first.

2.2.2. Interpolation of nodal displacements

From nodal values $\mathbf{d}_n^{(k,l)} = \mathbf{x}_n^{(l)} - \mathbf{x}_n^{(k)}$, $n = 1, \dots, N$ obtained above, the MF can be obtained for any point $\mathbf{x}$ within mesh element D_m, performing a spatial interpolation as follows:

$$\mathbf{d}_{k\to l}(\mathbf{x}) = \sum_{n=1}^{N_1} \mathbf{d}_n^{(k,l)} \phi_{m,n}^{(k)}(\mathbf{x}), \qquad \mathbf{x} \in D_m \tag{2.4}$$

where $\phi_{m,n}(\mathbf{x})$ is the basis function contributing from node n to element D_m, and N_1 is the number of nodes that are vertices of D_m. In our experiments, we chose the mesh to be triangular, therefore $N_1 = 3$, and we chose the basis interpolation functions to be linear.

A pixel representation of the MF can be obtained as:

$$\mathbf{d}_{k\to l} = \mathbf{\Phi}_k \mathbf{d}^{(k,l)} \tag{2.5}$$

where $\mathbf{d}^{(k,l)} = [\mathbf{d}_1^{(k,l)}, \mathbf{d}_\mathbf{2}^{(k,l)}, \dots, \mathbf{d}_\mathrm{N}^{(k,l)}]^\mathrm{T}$ is a Nx2 matrix representing nodal displacement, is defined as $\mathbf{\Phi}_k = [\phi_{m,1}^k, \phi_{m,2}^k, \dots, \phi_{m,N}^k]$ and $\mathbf{d}_{k\to l}$ is a two-column matrix representing the pixelized MF vectors.

3. Implementation of DMM with MRI data

3.1. Generation of nodes and initial mesh

To initialize the motion estimation algorithm, we need a mesh from which to start for each slice and view. The selection of nodes that determine each initial mesh is an essential factor to have correct motion estimation. The motion algorithm assumes that the structure of this image-intensity mesh is also a good structure for representing a motion field.

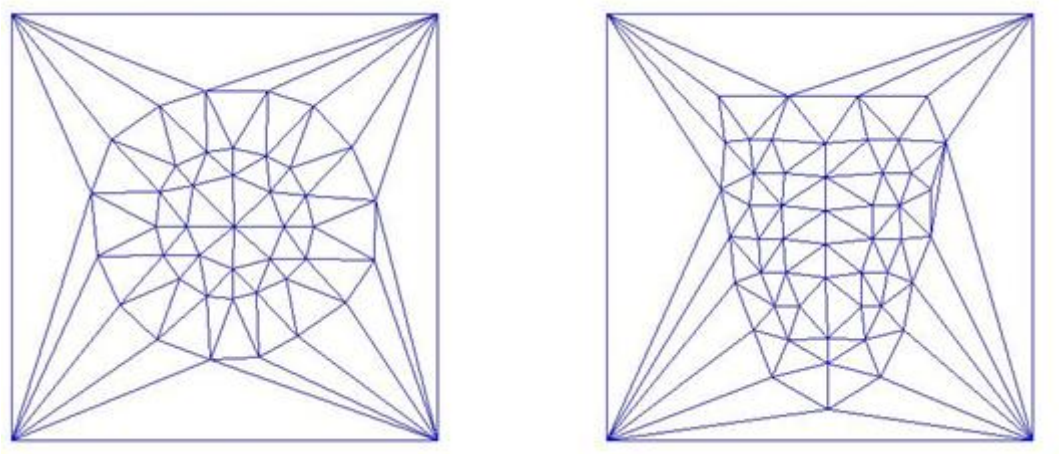

Figure 3.1. Initial mesh to be deformed through time. Left: SA view. Right: LA4CH view.

3.1.1. Contouring of LV walls

The first step is the segmentation of the two walls of the left ventricle: the inner wall or endocardium and the outer wall or epicardium. For this contouring process a semi-automatic program provided by collaborators of the Ginn College of Engineering

at Auburn [1] was used. The software requested to manually click on a set of "seed points" following the boundary of each wall to be contoured. These seed points were converted into a B-Spline curve, which could be controlled and modified to obtain a good approximation of the points which formed the walls. In this project we assumed this step to be done, starting directly with the walls of the first timeframes of each slice segmented in 64 points for each inner and outer walls.

3.1.2. Types of nodes

The initial mesh is composed of three different types of nodes:

Wall nodes: these nodes belong to the LV wall contours. Therefore, we will define inner nodes (endocardium) and outer nodes (epicardium), which assignment has to be done trying to achieve two goals at the same time: a correct position in order to make easier the 3D reconstruction task in the future and also a suitable location to perform the Delaunay's triangulation. For obvious reasons, these nodes are defined as non-stationary (able to move) and will provide us with the information about the displacement through the heart cycle.

Support nodes: these nodes are defined inside the blood pool (inner supporting nodes) and outside the myocardium (outer supporting nodes). Their purpose is to serve as a support structure for the deformable mesh. Our results showed that the best performance is achieved when we define the outer supporting nodes as stationary, and the inner supporting nodes as non-stationary.

Corner nodes: these four nodes are defined at the corners of the image. Their function is to serve as a support for all the rest of nodes and the whole mesh generated from them. They are defined as stationary.

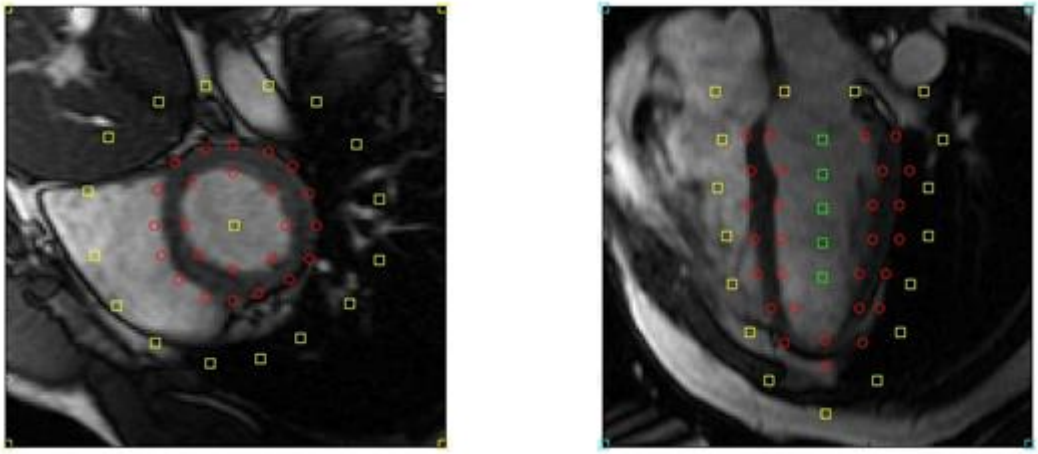

Figure 3.2. Generation of nodes. Red nodes correspond to wall nodes. Green nodes correspond to inner support nodes. Yellow nodes correspond to outer support nodes. Blue nodes correspond to corner nodes.

3.2. Motion estimation

Once we have generated the initial mesh for each slice and view, the next step is running the motion estimation algorithm. This algorithm is based on the mathematical apparatus explained in "2.2. Mathematical Model". We apply the initial mesh to the set of composed MRI images in order to perform iteratively the minimization of Equation (2.1) from each frame k to the next $k+1$. Thus, given a mesh at a frame k (reference mesh), we will obtain the position of the nodes in frame $k+1$ (target mesh).

The minimization of Equation (2.1) is calculated by an iterative gradient descending algorithm. This algorithm performs two different kinds of iterations:

- Local iterations: estimating the nodal displacement from each frame to the next one for a certain number of times
- Global iterations: repeating periodically the estimation of all slices for each timeframe through several cycles.

In the following section we will present the results obtained.

3.3. Results

2D Motion Field

In order to know if our mesh is moving correctly, we first show the motion field calculated from ED to ES for an SA slice given, in this case the 8th slice (see Figure 3.3.). For visual purposes we have represented only the vectors corresponding to the pixels inside the left-ventricle walls.

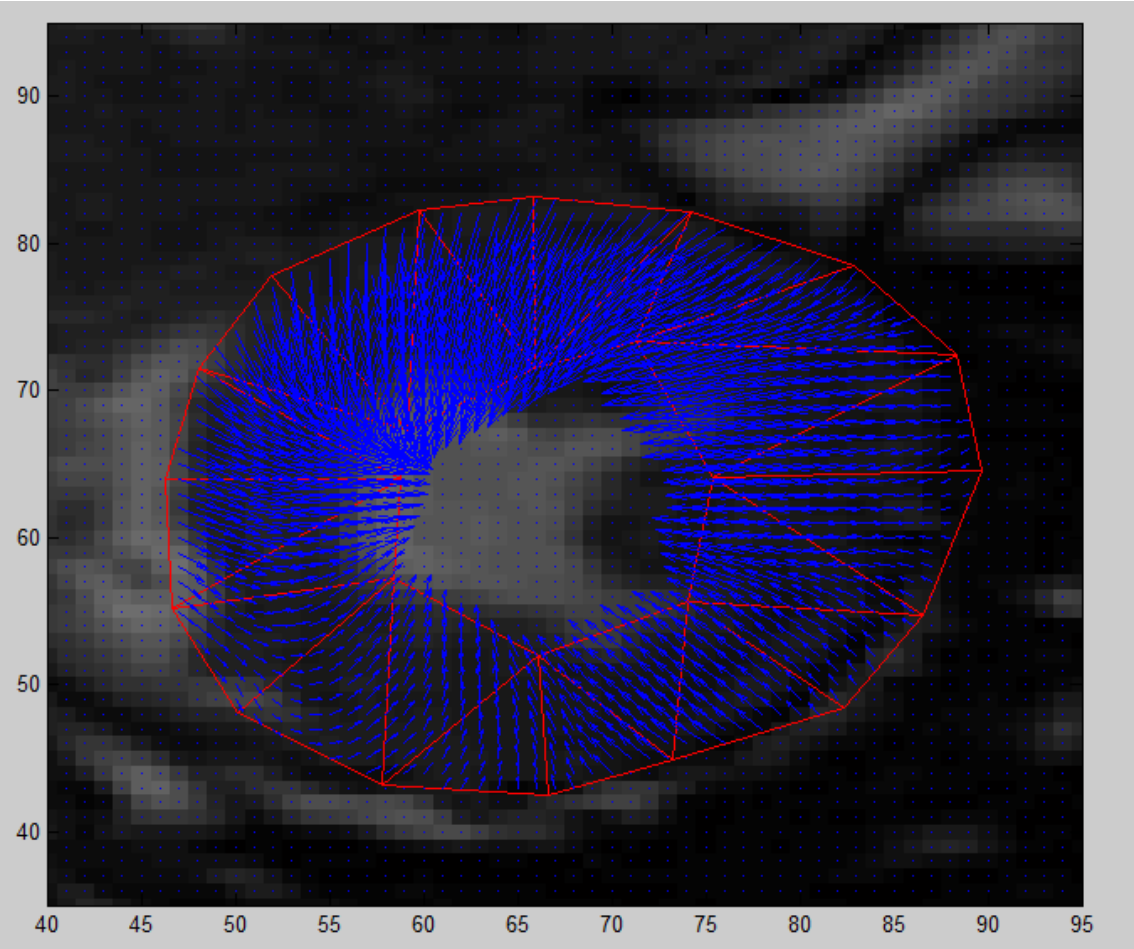

Figure 3.3. 2D Motion Field

The arrows represent the estimated deformation (change of position for each pixel) of the heart. As explained in "2.2. Mathematical Model", we originally have only

an estimation of the movement for each node, and then use interpolation to obtain the movement for the rest of the pixels of the image. As a first impression, we can recognize a logical motion, as the contraction is clearly by watching the arrows directions.

2D DMM sequence

After analyzing the motion field between two timeframes, the next step is to improve the movement through the whole cardiac cycle for each slice (2D). The set of images shown under these lines (Figure 3.4.) represent the evolution of the mesh for one slice given (in this case the 6th slice) for 6 different timeframes (each two timeframes from the first to the 11th).

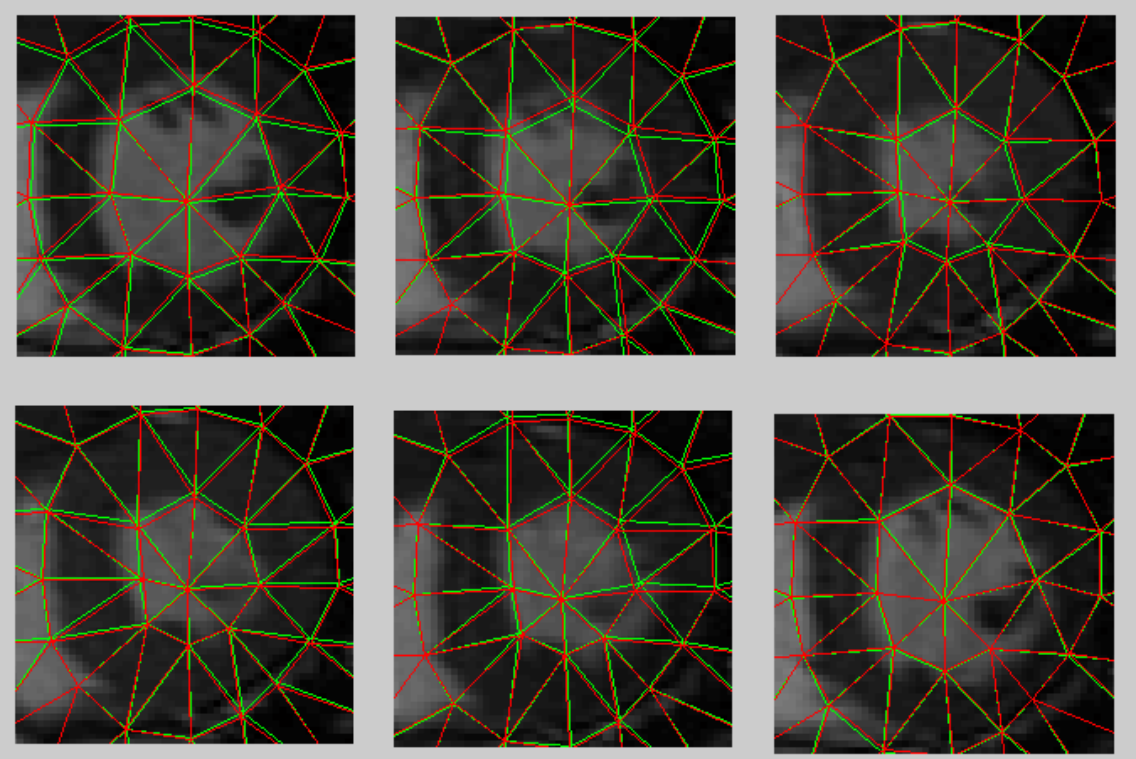

Figure 3.4. 2D DMM sequence

In the images we represent two mesh structures. The red one is the reference mesh, whereas the green one corresponds to the target mesh. The first 3 timeframes (superior row), correspond to the contraction of the left ventricle, as the ES takes place between the 6^{th} and the 7^{th} timeframe. It can be clearly observed, that the green mesh is closer to the center, reflecting contraction. But on the other hand, when the ventricle is stretching (images of the inferior row), the green mesh (target mesh) shows a tendency to leave the center. The red mesh (reference mesh) is now closer to the center.

Apparently we have achieved our goal of estimate the motion, as the mesh is following the contraction and stretching of the left ventricular walls.

3D DMM plot

Finally, we verify the result of the reconstruction in 3D of the left ventricle. It is possible to recognize how all SA slice contract at the same time, showing a correct synchrony during the whole cardiac cycle. In the image bellow (Figure 3.5.), we display the 3D images for the ED and ES timeframes, for several SA slices.

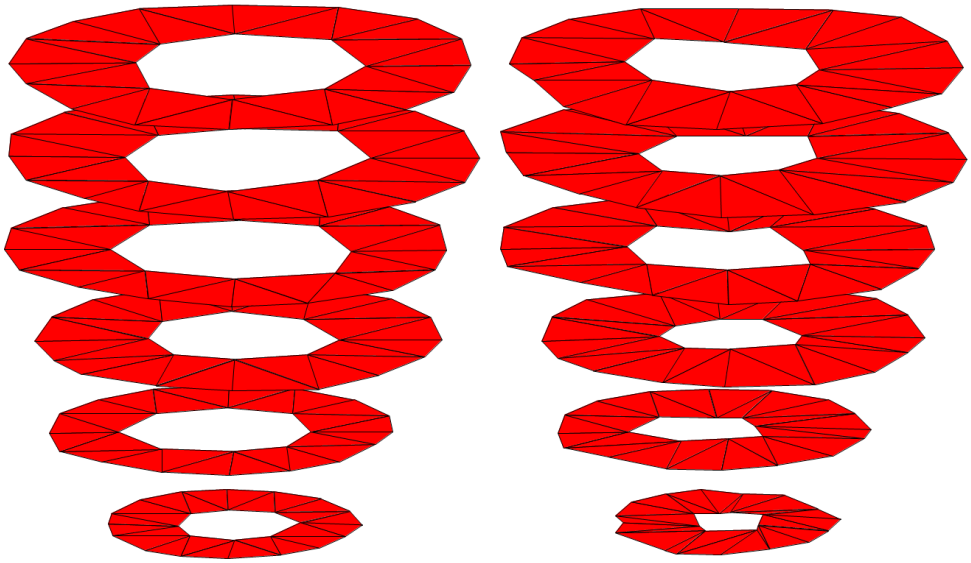

Figure 3.5. 3D DMM plot

4. Myocardial Strain: A measure of deformation

4.1. Introduction to Myocardial Strain

The definition of deformation is the change in distance between any two points within a body from one time to another. If the distance between any pair of points has not changed the body has not been deformed, though it may have been rotated and translated. Thus, rotations and translations does not contribute to the deformation. Strain is the dimensionless ratio between the length in the original state and the length in the deformed state. Today only strain is possible to study and measure when it comes to the heart's muscle tissue.

Although ventricular mass, volume, and ejection fraction (EF) represent the current standard for evaluating cardiac global function, extensive research has shown that regional function measures, such as strain, may be more specific in defining cardiac disease and more sensitive in detecting subclinical markers of LV dysfunction and myocardial diseases.

4.2. Mathematical Principles

To provide a precise description of changes in the shape, size and orientation of a solid body, there are various tensors that describe this deformation. The displacement field $\mathbf{u}(\mathbf{x})$ describes the change in positions, relative to a convenient coordinate system, of all points in the body.

The key lies in describing how each "material point" is displaced, *i.e.*, in associating a displacement **u** , with each material point. The way to identify material points, is by their positions in the reference state, **X**. Thus,

- **X** is position in the reference state,
- **x** is position in the current (deformed) state, and
- **x** is function of **X**, $\mathbf{x} = \mathbf{x}(\mathbf{X})$

Let us define the deformation gradient as the gradient of , **x**(**X**) as follows,

$$\mathbf{F} = \delta\mathbf{x}/\delta\mathbf{X} \tag{4.1}$$

Since the relation in terms of displacement for each point satisfy

$$\mathbf{x} = \mathbf{X} + \mathbf{u} \tag{4.2}$$

we may express the displacement gradient as

$$\frac{\delta\mathbf{u}}{\delta\mathbf{X}} = \frac{\delta\mathbf{x}}{\delta\mathbf{X}} - \mathbf{I} = \mathbf{F} - \mathbf{I} \tag{4.3}$$

Further, let **N** be a unit vector embedded within the body in the reference configuration; the deformation transforms **N** into **n** in the deformed state. The square of the stretch of **N**, $\lambda(\mathbf{N})$**,** can be calculated as,

$$\lambda^2(\mathbf{N}) = (\mathbf{F}\cdot\mathbf{N})(\mathbf{F}\cdot\mathbf{N}) = \mathbf{N}\cdot\mathbf{F}^T\cdot\mathbf{F}\cdot\mathbf{N} \tag{4.4}$$

Define $\mathbf{C} \equiv \mathbf{F}^T \cdot \mathbf{F}$ as the right Cauchy-Green deformation tensor. Then the stretch, squared, is also given by

$$\lambda^2(\mathbf{N}) = \mathbf{N} \cdot \mathbf{C} \cdot \mathbf{N} \tag{4.5}$$

By using the polar decomposition theorem, it is obtained

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U}, \qquad \mathbf{C} = \mathbf{F}^T \cdot \mathbf{F} = \mathbf{U^2}, \qquad \lambda^2(\mathbf{N}) = \mathbf{N} \cdot \mathbf{U^2} \cdot \mathbf{N} \tag{4.6}$$

The tensor $\mathbf{U}$ is called the right stretch tensor. If we now define the strain in the direction $\mathbf{N}$ as

$$\mathrm{E}(\mathbf{N}) = \frac{1}{2}[\lambda^2(\mathbf{N}) - 1] = \mathbf{N} \cdot \mathbf{E} \cdot \mathbf{N} \tag{4.7}$$

we deduce the strain tensor

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) \cdot \tag{4.8}$$

which is known as the Lagrangian strain tensor.

4.3. Strain Calculation

Typically, the following three types of the myocardial normal strain are used, which correspond to the geometry of the heart:

- Circumferential strain (E_{cc})
- Radial strain (E_{rr})
- Longitudinal strain (E_{ll})

Within the framework of our project, we dispose of a motion field, and utilizing Equation (4.3) we can obtain the displacement gradient just adding the identity matrix to the motion field, and consequently we can deduce the Lagrangian strain for each specific direction applying Equation (4.7).

Results

In the Figure 4.1. we can see the radial strain calculation for an SA slice. The motion field used was for the displacement of the pixels between ED and ES, as you can appreciate on the right side of the picture. As the heart is contracting, there appear dark areas near to center. The corners show a stretching of the muscle in the white areas.

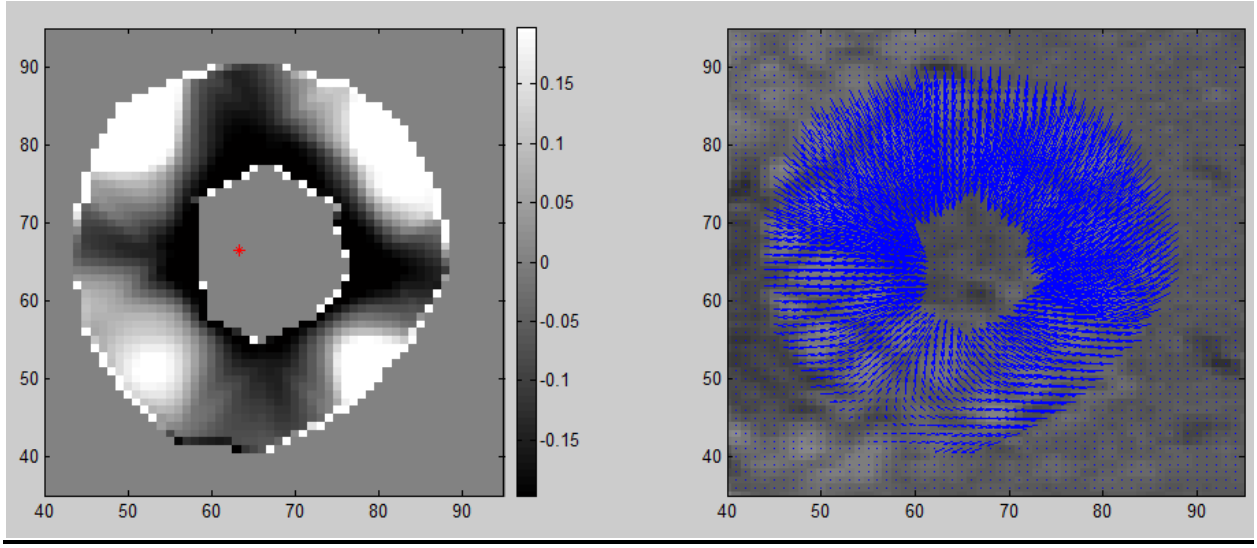

Figure 4.1. Strain Calculation for MF between ED and ES

5. Matlab Guide

5.1. Previous Configuration

1. Download from the CD these next 3 folders and save them in the root directory of your computer (C:/) with the same name:

   tdenney2

   fparages

   SCCOR

   Therefore, in your computer you must have these 3 folders, with the following paths:

   C:\tdenney2

   C:\fparages

   C:\SCCOR

2. Add to the Matlab path these 3 folders **with all their subfolders**.

5.2. Nodes Generation Software

Program "generate_nodes_SA.m"

- **Goal**:
  - to generate the nodes corresponding to the SA slices

- **Directory**:

  'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Nodes_Generation \Main_Programs'

- **Function:**

```
function [nodes_structure_SA]=
generate_nodes_SA(slice_SA_ini,slice_SA_end)
```

- **Inputs:**

  slice_SA_ini → generate nodes **since** this slice
  slice_SA_end → generate nodes **until** this slice

- **Outputs:**

  node_structure_SA → structure of the SA nodes

  Example:

```
>> [nodes_structure_SA]=generate_nodes_SA(3,14)

nodes_structure_SA =

     slice3: [1x1 struct]
     slice4: [1x1 struct]
     slice5: [1x1 struct]
     slice6: [1x1 struct]
     slice7: [1x1 struct]
     slice8: [1x1 struct]
     slice9: [1x1 struct]
    slice10: [1x1 struct]
    slice11: [1x1 struct]
    slice12: [1x1 struct]
    slice13: [1x1 struct]
    slice14: [1x1 struct]
```

```
>> nodes_structure_SA.slice3

ans =

      Node_x: [45x1 double]
      Node_y: [45x1 double]
    Node_val: [45x1 double]
       Nnode: 45
        stac: [45x1 double]
       inner: [45x1 double]
       outer: [45x1 double]
```

- **Calls to other programs**

  nodes_for_one_SA
  (generates the nodes only for one slice)

Program "generate_nodes_LAXCH.m"

This explanation is valid for LA2CH and LA4CH (X can be 2 or 4)

- **Goal**:
  - to generate the nodes corresponding to the LA

- **Directory**:

  'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Nodes_Generation\Main_Programs'

- **Function:**

```
function [nodes_structure_LAXCH]=
generate_nodes_LAXCH(nodes_structure_SA)
```

- **Inputs:**

  nodes_structure_SA → the nodes of SA, so that we can calculate the nodes of LA and they coincide

- **Loaded ".mat" files:**

  ci_groupFlipe_LAXCH;
  ci_cineFlipe_LAXCH;

(these structures are created in the segmentation program and contain the information about the wall contours and the MRI images)

Example:

```
>> load ci_groupFlipe_LA2CH
load ci_cineFlipe_LA2CH

>> ci_group

ci_group =

         nTimeSlots: 20
          timeSlots: [1x20 double]
        nSpaceSlots: 1
         spaceSlots: -25.9675
              slice: [1x1 struct]
           axisType: '2Chamber'
         SliceOrder: []
    ImagingProtocol: 'tagged'
         TagPattern: 'grid45'
     BloodIntensity: []
            TagFWHM: []
           TagAngle: []
         ImageAngle: []
        SArefSeries: []
        TableOffset: []
        LandmarkARV: [1x1 struct]
        LandmarkIRV: [1x1 struct]
         LandmarkMS: [1x1 struct]
       LandmarkAPEX: [1x1 struct]

>> group_struct

group_struct =

         nTimeSlots: 20
          timeSlots: [1x20 double]
        nSpaceSlots: 1
         spaceSlots: -25.9675
              slice: [1x1 struct]
           axisType: '2Chamber'
         SliceOrder: []
    ImagingProtocol: 'cine'
         TagPattern: []
     BloodIntensity: []
            TagFWHM: []
           TagAngle: []
         ImageAngle: []
        SArefSeries: []
        TableOffset: []
        LandmarkARV: []
        LandmarkIRV: []
         LandmarkMS: []
       LandmarkAPEX: [1x1 struct]
```

To obtain information about the image of the first slice and first timeframe:

```
>> ci_group.slice(1).time(1).image

ans =

                        Filename:
'C:\tdenney2\SCCOR\05P1VOL527\BAS\SE9\IM1'
                    Manufacturer: 'GE MEDICAL SYSTEMS'
                         StudyID: '7758'
                       StudyDate: '20051128'
                 StudyDescription: 'SCCOR'
                     SeriesNumber: 9
                SeriesDescription: '2CH TAGGING'
                   SOPInstanceUID:
'1.2.840.113619.2.79.2162876711.9288.1133374724.942'
                AcquisitionNumber: 0
                   InstanceNumber: 1
           CardiacNumberOfImages: 20
                 NominalInterval: 7
                     TriggerTime: 10
                   SliceLocation: -25.9675
            SpacingBetweenSlices: 8
                  SliceThickness: 8
                       HeartRate: 66
             ImagePositionPatient: [3x1 double]
          ImageOrientationPatient: [6x1 double]
                    PixelSpacing: [1.5625 1.5625]
                            Rows: 256
                         Columns: 256
                    WindowCenter: 62
                     WindowWidth: 125
                        TimeSlot: 0
                       SpaceSlot: 0
      RegistrationOffsetScanner: []
   RegistrationRotationScanner: []
       RegistrationInitialCost: []
         RegistrationFinalCost: []
                     ProcVolume: 1
                  ProcCurvature: 1
                    EndoContour: [1x1 struct]
                    EpiContour: [1x1 struct]
                   ApexLandmark: []
                   BaseLandmark: []
                    LandmarkARV: []
                    LandmarkIRV: []
                     LandmarkMS: []
                   LandmarkAPEX: []
                    nTagObjects: 2
                      TagObject: [1x2 struct]
```

- **Outputs:**

  nodes_structure_LAXCH → structure of the LAXCH nodes

  Example:

```
>>[nodes_structure_LA2CH]=generate_nodes_LA2CH(nodes_struc
ture_SA)

nodes_structure_LA2CH =
```

```
    slice0: [1x1 struct]

>> nodes_structure_LA2CH.slice0

ans =

      Node_x: [63x1 double]
      Node_y: [63x1 double]
    Node_val: [63x1 double]
       Nnode: 63
        stac: [63x1 double]
       inner: [63x1 double]
       outer: [63x1 double]
```

- **Calls to other programs**

  points_LAXCH_inner_and_outer
  (from the 64 inner + 64 outer nodes given, selects the nodes in order to have a suitable location)

Program "nodes_for_one_SA.m"

- **Goal**:
  - to generate the nodes corresponding to one SA slice

- **Directory**:

  'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Nodes_Generation\Main_Programs'

- **Function:**

```
function [mesh_node]=nodes_for_one_SA(slice,center_total)
```

- **Inputs:**

  slice → number of SA we want to calculate nodes

  center_total → average of the centers of all SA slices
  (The idea is to create the selection of the nodes, so that the 3D reconstruction can be done. We need to have a reference of center that is valid for all SA slices)

- **Loaded ".mat" files:**

  ci_groupFlipe_SA;
  ci_cineFlipe_SA;
  (these structures are created in the segmentation program and contain the information about the wall contours and the MRI images)

- **Outputs:**

  mesh_node → structure of the SA slices for one slice only!

  Example:

```
>> center_total=[66.9716   64.5760];

>> [mesh_node]=nodes_for_one_SA(5,center_total);

>> mesh_node

ans =

      Node_x: [45x1 double]
      Node_y: [45x1 double]
    Node_val: [45x1 double]
       Nnode: 45
        stac: [45x1 double]
       inner: [45x1 double]
       outer: [45x1 double]
```

- **Calls to other programs**

  points_outer
  points_inner
  (from the 64 inner + 64 outer nodes given, selects the nodes in order to have a suitable location)

5.3. Motion Estimation Software

Main program : "motion_estimation.m"

- **Goal**:

  - to estimate the motion of the nodes generated above, by means of a deformable mesh which tracks composed MRI images

- **Directory**:

  'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Motion_Estimation\Main_Programs'

- **Function:**

```
function [mesh3D]=motion_estimation(slice_SA_end)
```

- **Inputs:**

  slice_SA_end → last SA slice from which to estimate motion

- **Definition of parameters:**

```
NGiter=2;  % number of global iterations(default 2)
niter=5;   % number of local(default 5)
nTF = 20;  % number of timeframes
stat_element_parameter = 2; % number of nodes which define
a stationary  element
gamma_def = 12;  % gamma of the penalization function
(default 12)
scalation =20;   % In short axis 10, in la2ch 30
(default:10 to 30)
slice_LA4CH=1; % LA4CH corresponds to slice number 1
slice_LA2CH=2; % LA4CH corresponds to slice number 2
slice_SA_ini=3; % the first SA corresponds to slice number
3
center_total=[66.9716   64.5760]; % Mean of the centers of
all SA slices
delta=1;
graph=1; % show iterations
text=1;  % show test comments
```

- **Outputs:**

mesh3D → structure with the motion of all slices and for all timeframes

Example of output:

```
mesh3D =

Nnode: 619
Node_x: [619x20 double]
Node_y: [619x20 double]
Node_z: [619x20 double]
Node_val: [619x20 double]
Reg_x: [128x128 double]
Reg_y: [128x128 double]
stac: [619x1 double]
Node_selector: [619x1 double]
tri: [1160x3 double]
T: {1x13 cell}
w: {1x13 cell}
tri_selector: [1160x1 double]
Melement: 1160
N: 128
M: 128
delta: 1
K: 1
inner_wall: [619x1 double]
outer_wall: [619x1 double]
```

Note: the fields Node_selector and tri_selector are **labels.** They indicate us which slice does a node or triangle belong to. The labels are:

- For LA4CH slice → label=1
- For LA2CH slice → label=2
- For SA_slice → labels=3,4,5,etc

- **Calls to other programs**

mesh3D_zero.m
image_preparation.m
D_optimize_up_Alex

Program "mesh3D_zero.m"

- **Goal**:

- to generate the structure which will support all the results of the mesh estimations
- the initial meshes for each slice are defined

- **Directory**:

'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Motion_Estimation\Main_Programs'

- **Function:**

```
function [mesh3D,offset_LA2CH] = mesh3D_zero(slice_end)
```

- **Inputs:**

slice_end → last SA slice from which to estimate motion

- **Definition of parameters:**

```
slice_ini=1;      %% Corresponding to LA4CH
slice_ini_SA=3;   %% Number of the first slice which
belongs to the SA view
size_img=128; % size of the image
slice_dist=5; % distance between SA slices
nTF=20; %number of timeframes
```

- **Loaded ".mat" files:**

SA_nodes
LA2CH_nodes
LA4CH_nodes
(all of them generated in the part Nodes Generation)

- **Outputs:**

mesh3D → structure described in **Goal**
offset_LA2CH → factor to align LA slices in mesh3D

- **Calls to other programs**

initial_mesh
(creates the initial mesh for a slice given)

Program "image_preparation.m"

- **Goal**:

  - loads the tagged and cine images and combines them in order to obtain the composed set of images
  - prepares the images in order to perform the estimation

- **Directory**:

  'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Motion_Estimation\Main_Programs'

- **Function:**

```
function [Sim]=image_preparation(slice)
```

- **Inputs:**

  slice → the number of the slice from which we want to estimate motion

- **Definition of parameters:**

```
tag_and_cine_combined = true;    %% True if we want to
combine cine+tagged image
                                 %% False to track only the
tagged image
contrast_factor_cine = 2000;     %% Contrast variation
applied to the cine image
contrast_factor_tagged =400;    %% Contrast variation
applied to the tagged image
```

- **Loaded ".mat" files:**

  MSI_SA
  MSIcine_SA
  MSI_LA2CH
  MSIcine_LA2CH
  MSI_LA4CH
  MSIcine_LA4CH

  - These sets contain the information of the images we need in this project
  - When loading them, we work with two different variables, Sim and Sim_cine

  Example:

```
>> load MSI_SA
>> whos Sim
  Name      Size        Bytes    Class     Attributes

  Sim       4-D       39321600  double

>> load MSIcine_SA
>> whos Sim_cine
  Name          Size     Bytes     Class     Attributes

  Sim_cine      4-D      39321600  double
```

Note: the variables Sim and Sim_cine have the next dimensioning:

Sim(dimension_x,dimension_y,timeframe,slice)

```
>> dsp(Sim(:,:,1,3))
```

(displays the first timeframe of the third slice)

- **Outputs:**

  Sim → the Sim variable after preparation

Program "plot_mesh3D.m"

- **Goal**:

  - to plot in 3D the mesh3D resulting of the main program "motion_estimation"
  - we have the option of creating an AVI video

- **Directory**:

  'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Motion_Estimation\Main_Programs'

- **Function:**

```
function [] =plot_mesh3D(mesh3D)
```

- **Inputs:**

mesh3D

- **Definition of parameters:**

```
nTF=20; % number of timeframes
rep = 10;          %% Number of repetitions of the movie
slice_ini_SA = 3; %% Initial SA slice to plot
slice_end_SA=13;  %% Last SA slice to plot
step_slice = 2;   %% STEP to view short axis slices (for
instance: if 2, we will see slices 3,5,7,9, etc)
view_long_axis = [1,2]; %% long axis views we want to see
(1 --> view LA4CH; 2--> view LA2CH; [1,2] --> BOTH)
az = 90;  %% view selection (azimut)
el = 15;  %% view selection (elevation)
```

5.4. Strain Calculation Software

Main program "strain_calculation.m"

- **Goal**:

  - Given a specific slice, the program calculates the Lagrangian Strain for the Motion Field obtained between the first slice (ED) and all the rest. The direction of the strain for each pixel is refered to the center of the SA blood pool

- **Directory**:

  'C:\tdenney2\matlab\MF_intensity_2D_Med_phys\Alex\Strain_Calculation\Main_Programs'

- **Function:**

```
function [E]=strain_calculation(slice,direction)
```

- **Inputs:**

  slice → the slice from which to calculate strain
  direction → 1 for radial strain / 2 for circumferential strain

- **Loaded ".mat" files:**

  mesh3D_just_SA (result of motion_estimation program)
  ci_groupFlipe_SA

- **Outputs:**

  E → Lagrangian Strain

- **Calls to other programs**

  image_preparation
  dif_plot_alex2 (obtain displacement fields)
  lagrangian_strain

6. Conclusions

The increasingly important role of medical imaging in the diagnosis and treatment of disease has opened an array of challenging problems centered on the computation of accurate models as the one we are presenting in this project.

Although still in his first steps of development, preliminary results have demonstrated great potential of the deformable mesh model. As it works directly over the images, not depending on human interaction to previously segmentate the tags, it has more flexibility and efficiency tan other models.

The main disadvantage of DMM lays in his high sensitivity to the change of nodal positions and parameter values, which causes an unpredictable behavior when modifying them. Future lines of research have to contemplate this problem, by trying to optimize DMM initial parameters and improve the software of nodes generation.

7. References

[1] Copyright © 2006, Imaging of Door County, LLC. All rights reserved.

[2] Copyright ©1999, Yale University. All rights reserved.

[3] Brankov, J.G.; Yang, Y. & Wernick, M.N. (2005) . "Spatiotemporal processing of gated cardiac SPECT images using deformable mesh modeling". *Medical Physics, Vol. 32, 2839 - 2849*

[4] Y. Wang and O. Lee. (1996). "Use of two-dimensional deformable mesh structures for video coding .I. The synthesis problem: mesh-based function approximation and mapping." *IEEE Trans. Circuits Syst. Video Tech., vol. 6, pp. 636-646.*

[5] Denney Jr.,T.S. (2007) . "SCCOR contouring and Tag Tracking procedure". *Ginn College of Engineering at Auburn University*

[6] Pottumarthi V. Prasad (2006) "Magnetic Resonance Imaging: Methods and Biologic Applications", *Department of Radiology, Evanston and Feinberg School of Medicine at Northwestern University*

[7] Javier Bonet and Richard D. Wood (1997) "Nonlinear continuum mechanics for finite element analysis". *Cambridge University Press*