Open source Framework for cardiac MRI analysis

Supervisor: DR. Ahmed S. Fahmy.

Team members: Ahmed Ibrahim Ahmed      Mohamed Fayez Abd Eltwab
Marwan Mohamed Naguib Mohamed
# TABLE OF CONTENTS

## ASBTRACT

5

## INTRODUCTION

5

### CARDIAC MRI

5

### DICOM FORMAT

6

### PACS SERVER

7

1. PACS ARCHITECTURE

7

2. PACS BENEFITS

7

## BUILDING OPEN SOURCE FRAMEWORK

9

### PACS SERVER

9

1. DCM4CHEE

10

1.1. DCM4CHEE ARCHITECTURE

10

### PACS AND DICOM VIEWER

11

1. MIPAV

12

1.1. MIPAV FEATURE LIST

12

1.2. MIPAV DOCUMENTATION

13

1.3. MIPAV IMAGE SCHEMA

14

## ACTIVE CONTOUR MODEL

15

### APPLICATIONS

15

### ADVANTAGES OVER OTHER EDGE DETECTION TECHNIQUES

16

### ACTIVE CONTOUR (SNAKE) METHOD

16

1. INTERNAL ENERGY

17

1.1. ELASTIC ENERGY

17

1.2. BENDING ENERGY

17

2. EXTERNAL ENERGY

18

3. COMBINING ENERGIES

18

### ACTIVE CONTOUR (SNAKE) ALGORITHM

18

### RESULTS

21

## OPTICAL FLOW

23

### OPTICAL FLOW ESTIMATION

23

### OPTICAL FLOW APPLICATIONS

24

### LUCAS KANADE METHOD

25

### RESULTS

27
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOMATIC SEGMENTATION</td>
<td>28</td>
</tr>
<tr>
<td>ACM GUIDED WITH OPTICAL FLOW ENERGY METHOD</td>
<td>28</td>
</tr>
<tr>
<td>1. ENERGY FUNCTION OF THE OPTICAL FLOW CONSTRAINT</td>
<td>28</td>
</tr>
<tr>
<td>2. CONTOUR CONSTRAINTS</td>
<td>29</td>
</tr>
<tr>
<td>3. CONTOUR-CONTRAINED OPTICAL FLOW ENERGY FUNCTION</td>
<td>31</td>
</tr>
<tr>
<td>RESULTS</td>
<td>31</td>
</tr>
<tr>
<td>CARDIAC PARAMETERS</td>
<td>34</td>
</tr>
<tr>
<td>LEFT VENTRICULAR (L.V.) FUNCTION ANALYSIS</td>
<td>34</td>
</tr>
<tr>
<td>1. GLOBAL L.V. FUNCTIONS</td>
<td>34</td>
</tr>
<tr>
<td>1.1. VOLUME-TIME CURVES (VTC)</td>
<td>34</td>
</tr>
<tr>
<td>1.2. EJECTION FRACTION (EF)</td>
<td>34</td>
</tr>
<tr>
<td>1.3. STROKE VOLUME (SV)</td>
<td>35</td>
</tr>
<tr>
<td>2. REGIONAL MYOCARDIAL THICKENING</td>
<td>35</td>
</tr>
<tr>
<td>RESULTS</td>
<td>36</td>
</tr>
<tr>
<td>4D VISUALIZATION OF L.V. WALLS</td>
<td>39</td>
</tr>
<tr>
<td>JAVA3D</td>
<td>39</td>
</tr>
<tr>
<td>1. WHAT IS JAVA3D?</td>
<td>40</td>
</tr>
<tr>
<td>2. THE SCENE GRAPH</td>
<td>40</td>
</tr>
<tr>
<td>PROCEDURE</td>
<td>42</td>
</tr>
<tr>
<td>1. STEPS OF BUILDING SEGMENT IN THE MODEL</td>
<td>43</td>
</tr>
<tr>
<td>RESULTS</td>
<td>44</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>46</td>
</tr>
</tbody>
</table>
FIGURE LIST

FIGURE (1), MODULAR DCM4CHEE ARCHITECTURE 10
FIGURE (2), MIPAV IMAGE SCHEMA 14
FIGURE (3), DIFFERENT BETWEEN ACM AND EDGE DETECTION METHODS 16
FIGURE (4), GENERAL IDEA IN ANY ACTIVE CONTOUR ALGORITHM 18
FIGURE (5), IMPLEMENTATION FLOW CHART FOR GREEDY ALGORITHM 20
FIGURE (6), SEGMENTATION OF ENDOCARDIUM USING ACM 21
FIGURE (7), SEGMENTATION OF MYOCARDIUM USING ACM 22
FIGURE (8), IMPLEMENTATION FLOW CHART FOR LOCUS KANADE 26
FIGURE (9), RESULTS OF USING LOCUS KANADE TRACKING METHOD 27
FIGURE (10), FLOW CHART FOR ACM GUIDED BY O.F. ENERGY 30
FIGURE (11), CALCULATION OF E_Of FROM TWO CONSECUTIVE TIME FRAMES T-1,T 30
FIGURE (12), RESULTS OF TRACKING L.V. WALLS USING ACM GUIDED BY O.F. ENERGY 31
FIGURE (13), ANOTHER RESULTS OF TRACKING L.V. WALLS USING ACM GUIDED BY O.F. ENERGY 32
FIGURE (14), RESULTS OF TRACKING L.V. EPICARDIUM WALLS USING ACM GUIDED BY O.F. ENERGY 33
FIGURE (15), COLOR MAPPING FOR NORMALIZED THICKNESS VALUES 35
FIGURE (16), VOLUME-TIME CHART FOR BLOOD, EPICARDIUM AND MYOCARDIUM 36
FIGURE (17), REGIONAL MYOCARDIAL THICKENING CHART 37
FIGURE (18), RESULT PDF CONTAINING CHARTS AND CLINICAL PARAMETERS 38
FIGURE (19), SCENE GRAPH OF 4D MODELING PLATFORM 41
FIGURE (20), DIFFERENT VIEWS FOR ENDOCARDIUM 3D MODEL 42
FIGURE (21), SKETCH OF RECONSTRUCTING MYOCARDIUM USING 2D IMAGES 43
FIGURE (22), BUILDING EPICARDIUM SURFACE USING TRIANGULATION THEORY 43
FIGURE (23), SEGMENT OF L.V. MYOCARDIUM 44
FIGURE (24), BUILDING SURFACE BETWEEN EPI- AND ENDO-CARDIUM 44
FIGURE (25), SIDE VIEW OF L.V. 3D MODEL 44
FIGURE (26), TOP VIEW OF UPPER SEGMENT IN L.V. 3D MODEL 45

TABLE LIST

TABLE (1), COMPARISON BETWEEN DIFFERENT PACS SERVERS 9
TABLE (2), COMPARISON BETWEEN DIFFERENT PACS VIEWERS 11
TABLE (3), REGIONAL MYOCARDIAL THICKENING 37
In current clinical settings, there are several technological challenges to perform automated functional analysis from cardiac MRI. In this work, we present a framework to automatically segment the heart anatomy, define segments of the left ventricle, extract myocardial motions for quantitative analysis of cardiac global and regional functions, and visualization of cardiac shape and deformation. This framework makes use of four-dimensional (4-D) processing (three spatial and one temporal) to multiphase multi-slice MRI that are widely available in clinical practice and produce a continuous 4-D model of the myocardial surface deformation. The model is used to measure diagnostically useful parameters, such as wall motion, myocardial thickening, myocardial mass measurements, end diastolic volume (EDV), end systolic volume (ESV), ejection fraction (EF), and cardiac output (CO).
INTRODUCTION

Cardiac MRI (CMR) image analysis is increasingly requested by clinicians worldwide in the workup of patients with suspicion of or established diagnosis of cardiovascular disease. The accuracy and reproducibility of CMR is well known. Quantitative volume, mass and function measurements are often mandatory for making an accurate diagnosis. Quantification of the cardiac ventricles is time-consuming, so it was a mandatory need to produce a System to eliminate this Problem and introduce an acceptable accuracy and performance.

In this work we were seeking for building System Satisfies our needs, which mainly were low cost, open source, high performance, user friendly, high accuracy, and cross platform software for Quantification of the cardiac ventricles.

CMRI quantification System usually consist of PACS server which used to store MRI datasets, DICOM viewer which used to view MR DICOM images, Software algorithms that used to perform the analysis.

CARDIAC MAGNETIC RESONANCE IMAGING (CMRI)

MRI is a safe, noninvasive test that creates detailed pictures of your organs and tissues, uses radio waves, magnets, and a computer to create pictures of your organs and tissues. Unlike other imaging tests, MRI doesn't use ionizing radiation or carry any risk of causing cancer [1].

Cardiac MRI creates both still and moving pictures of your heart and major blood vessels. Doctors use cardiac MRI to get pictures of the beating heart and to look at its structure and function. These pictures can help them decide the best way to treat people who have heart problems.

Cardiac MRI Clinical Applications for example, is the evaluation of congenital heart disease, cardiac masses, the pericardium, right ventricular dysplasia, hibernating myocardium, evaluation of myocardial perfusion and of valveular and ventricular function and coronary artery imaging [2].

MRI diagnoses and assesses many diseases and conditions, including:

- Coronary Heart Disease.
- Damage caused by a Heart Attack.
- Heart Failure.
- Heart Valve Problems
- Congenital heart defects (at birth).
• Pericarditis (a condition in which the membrane around your heart is inflamed).
• Cardiac Tumors.

**DICOM FORMAT IMAGES**

With the introduction of computed tomography (CT) followed by other digital diagnostic imaging modalities in the 1970’s, and the increasing use of computers in clinical applications, the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) recognized the emerging need for a standard method for transferring images and associated information between devices manufactured by various vendors. These devices produce a variety of digital image formats [3].

So, the Digital Imaging and Communications in Medicine (DICOM) standard was created by the National Electrical Manufacturers Association (NEMA) to aid the distribution and viewing of medical images, such as CT scans, MRIs, and ultrasound [4].

A single DICOM file contains both a header (which stores information about the patient's name, the type of scan, image dimensions, etc), as well as all of the image data (which can contain information in three dimensions). This is different from the popular Analyze format, which stores the image data in one file (*.img) and the header data in another file (*.hdr). Another difference between DICOM and Analyze is that the DICOM image data can be compressed (encapsulated) to reduce the image size. Files can be compressed using lossy or lossless variants of the JPEG format, as well as a lossless Run-Length Encoding format (which is identical to the packed-bits compression found in some TIFF format images).

It includes a network communications protocol. The communication protocol is an application protocol that uses TCP/IP to communicate between systems. DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format.

DICOM enables the integration of scanners, servers, workstations, printers, and network hardware from multiple manufacturers into a picture archiving and communication system (PACS). The different devices come with DICOM conformance statements which clearly state which DICOM classes they support. DICOM has been widely adopted by hospitals and is making inroads in smaller applications like dentists’ and doctors’ offices.
PACS SERVER

PACS is a system for digital storage, transmission and retrieval of radiology images. PACS systems have both software and hardware components, which directly interface with imaging modalities and acquire the digital images from the modalities. The images are transferred to a workstation for viewing and reporting.

PACS ARCHITECTURE [5]

- PACS Central Server: the core hardware of the system.
- PACS Workstation: provide interactive display of digital images with PACS viewer for radiologists to interpret the images.
- Database System: responsible for managing the storage of all the information and images in the PACS system.
- DICOM Server: responsible for all the DICOM based communication with imaging modalities (such as CT and MRI), other PACS servers and DICOM workstations.
- Storage System: the physical storage required to store the DICOM images of the PACS system.
- Interfaces to RIS/HIS: aggregates all the data of the patient from different departments and provides an end-to-end patient care workflow.
- Web Server for Remote Access: With this the images and other PACS data can be accessed with internet browsers such as Internet Explorer, Mozilla, etc.

PACS BENEFITS [5]

PACS has multiple advantages over film based reporting.

- Reduces the cost of radiology by eliminating the need to have hardcopy films and the associated storage space required for the same. The cost of digital storage has decreased drastically over the past few years.
- PACS provides a faster and more reliable way of retrieving prior images.
- PACS enables remote access of the images helping the radiologist to manage his/her time effectively.
- PACS provides for easy integration of images into the hospital information systems. This enables an effective way of managing comprehensive patient information providing a single point of access for all patient information.
- As the images and reports are transmitted digitally, when combined with an effective workflow this can deliver a fast report turnaround. This enables more rapid diagnosis and enhances patient care.
• It enables faster peer review of the reports and subspecialty second opinions resulting in a significant improvement in accuracy of diagnosis and patient care.
• Better analysis of accuracy rates, Turn-around-times (TATs) for reports that help to make management decisions to improve the effectiveness and efficiency of the radiology practice.
• Security and HIPAA compliance is facilitated by using PACS systems.
In this project we use DCM4CHEE as a PACS server. The DCM4CHEE DICOM open source frameworks best fits our General and Specific requirements necessary to develop our System.

Before choosing DCM4CHEE there were two more PACS servers, DCMTK and CONQUEST, and we choose based on our needs and following comparison in figure (1).

<table>
<thead>
<tr>
<th></th>
<th>DCMTK</th>
<th>DCM4CHEE</th>
<th>CONQUEST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documentation</strong></td>
<td>Comprehensive</td>
<td>Not-enough</td>
<td>Poor</td>
</tr>
<tr>
<td><strong>Maintainability</strong></td>
<td>High</td>
<td>High</td>
<td>Lower</td>
</tr>
<tr>
<td><strong>Programming language</strong></td>
<td>C/C++</td>
<td>Java, XML</td>
<td>C/C++</td>
</tr>
<tr>
<td><strong>Extendibility</strong></td>
<td>Add new SOP’s, requires modify and rebuild the source code</td>
<td>Add new SOP’s don’t always require modify the source code</td>
<td>Add new SOP’s requires modify and rebuild the source code</td>
</tr>
<tr>
<td><strong>Operating Systems</strong></td>
<td>UNIX, Linux, Windows</td>
<td>Multiplatform</td>
<td>Windows, UNIX</td>
</tr>
<tr>
<td><strong>Client/Server PACS Model</strong></td>
<td>As Server: Modality Worklist and Storage</td>
<td>Provides complete support</td>
<td>As Server: Modality Worklist and Storage</td>
</tr>
<tr>
<td><strong>DICOM Services</strong></td>
<td>No Hanging Protocol support</td>
<td>Hanging Protocol, Storage Commitment, Basic Worklist</td>
<td>No Hanging Protocol support</td>
</tr>
<tr>
<td><strong>DICOM IOD’s (Image Types)</strong></td>
<td>US, CT, MR, SC, DX, XA, VL, RT</td>
<td>US, CT, MR, SC, DX, XA, VL, RT</td>
<td>US, CT, MR, SC, DX, XA</td>
</tr>
</tbody>
</table>

Table (1), Comparison between different PACS servers based on specifications
DCM4CHEE

DCM4CHEE is a collection of open source applications and utilities for the healthcare enterprise. These applications have been developed in the Java programming language for performance and portability, supporting deployment on JDK 1.4 and up [6].

Dcm4chee uses a database to store information from the DICOM headers, index information for locating objects on the file system, and other pertinent system and clinical data:

- PostgreSQL.
- MySQL.
- Oracle.
- SQL Server
- DB2.
- Firebird.
- HSQL.

DCM4CHEE ARCHITECTURE

The DCM4CHEE architecture was designed to be modular allowing for independent use of the archive's services. Each service can be enabled/disabled using the DCM4CHEE JMX console (web interface). This modular design also lightens the load of maintaining old code and adding new features/services, Figure (1)

Figure (1), Modular dcm4chee architecture chart [7]
PACS AND DICOM VIEWER

The PACS viewer is a software that is installed on the workstation to receive, display, and making processing on the radiology images. The images are then archived into storage for retrieval at a later time. The PACS system manages the storage of these radiology images [8].

In this project we use MIPAV as a PACS viewer it was the perfect choice that satisfies our needs. The MIPAV (Medical Image Processing, Analysis, and Visualization) application enables quantitative analysis and visualization of medical images of numerous modalities such as PET, MRI, CT, or microscopy. Using MIPAV’s standard user-interface and analysis tools, researchers at remote sites (via the internet) can easily share research data and analyses, thereby enhancing their ability to research, diagnose, monitor, and treat medical disorders.

Before choosing MIPAV there were two more PACS viewers K_PACS and Weasis and we choose based on our needs and following comparison in figure(2).

<table>
<thead>
<tr>
<th>Displaying images</th>
<th>MIPAV</th>
<th>K_PACS</th>
<th>weasis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default View, Light box View and Cine View.</td>
<td>Regular tools: cine, zoom, magnifier and windowing.</td>
<td>Default View</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VOIs</th>
<th>Point, 2D line, 2D rectangle, 3D rectangle, 2D ellipse, 2D polygon, 3D polygon, and interactive level-set.</th>
<th>Not supported</th>
<th>Read VOI LUT sequence.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Painting</th>
<th>Masks, Morphological operations, Region growing and Statistics.</th>
<th>Measuring distance and density (HE).</th>
<th>Not supported</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Histogram</th>
<th>Yes.</th>
<th>Not supported</th>
<th>Not supported</th>
</tr>
</thead>
</table>

|----------------------------|-----------------------------------------------------------------|---------------|-----------------------------|


Table (2), Comparison between different PACS viewers based on specifications
MIPAV is a Java application and can be run on any Java-enabled platform such as Windows, UNIX, or Macintosh OS X [9].

MIPAV includes a browser that is specifically designed for examining DICOM images. The DICOM browser allows you to view a list of all of the images in a dataset and quickly display each one individually. The browser provides a fast way of locating and selecting specific image slices.

**MIPAV FEATURE LIST [10]**

- Supported formats
  - PSD, PIC, GIF, JPEG, JPG, JFIF, JFI, BMP, DIB, CUR, ICO, ICO, PCX, DCX, PCC, PNG, DCM, RS, RAS, TGA, TIF, TIFF, XBM, XPM

- Histogram and LUT tools
- DICOM Browser
- DICOM Database Query/Receive
- In application help system
- Image overlays and composites
- Visualization
- Tri-planar View
- Animate View
- Movie View
- Lightbox View
- 3-D Surface Views
- Surface Plotter
- Volume Rendering
- Iso-surface rendering
- RFA (radio frequency ablation) simulation
- Stereo rendering
- Virtual endoscopy tool
There are many documents on how to use MIPAV [11]:

- **User Guide**: the complete end-user information on MIPAV, including installation, use and technical conformance.
- **Frequently Asked Questions (FAQ)**: answers to commonly asked questions.
- **How-To**: a quick guide with step-by-step instruction on complex tasks.
- **Presentations**: a collection of presentation slides that the MIPAV group has used in classes and displays.
- **Video Tutorials**.
Figure (2), MIPAV image schema [12]
ACTIVE CONTOUR MODEL

One of the general tasks in image processing is to split the image into several parts like object(s), also called foreground and background set of pixels. This procedure called segmentation. There are two ways to define the segments of the image. The first way, if we locate the surface pixels of the object areas. The second way, if we define the boundaries of the area of interested. Wide spectrum of the different segmentation techniques known from the simple threshold during region growing and edge detection methods until different kind of machine learning techniques, like clustering methods or neural networks segmentation. The Active Contour Models belong to the class of the edge detection methods. The active contour model algorithm or snake was introduced by Michael Kass, Andrew Witkin and Demetri Terzopoulos in 1988 [13].

Snake is an energy minimizing technique guided by external constraint and influenced by image forces. The algorithm deforms a contour to lock onto features of interest on an image. Usually the features are edges, lines and boundaries. It is working in 2D, 3D and even higher dimensions. Its 3D version is often called as deformable models or active surfaces.

According to the literatures of snake two kinds of snake representation are known recently; Parametric and Geometric Contours.

APPLICATIONS

This technique is often used in computer vision (motion tracking) and image analysis (edge/contour detection, subjective contours). It is also used to describe the boundary of an object. For example a snake can be used in medical image analysis to find the outline of an organ. Also it might be used to highlight some part of an electric circuit in manufacture environment and so on.
ADVANTAGES OVER OTHER EDGE DETECTION TECHNIQUES

Several methods are used for edge detection in images. The most known algorithms are “Sobel” and “Prewitt” methods. A default of these methods is to detect every small object but not the complete contour of the general shape.

![Figure (3)](image)

Figure (3). Different results between ACM and other edge detection methods:
(a) Initial image, (b) Sobel filtering, (c) Prewitt filtering,
(d) Canny filtering, (e) Active contour [14]

ACTIVE CONTOUR (SNAKE) METHOD

Snake is an “active” model as it always minimizes its energy function and therefore it has dynamic behavior.

A simple elastic snake is defined as

1. A set of $n$ points.
2. An internal elastic energy term.
3. An external edge based energy term.

A snake has to be initialized near the target and it will iteratively be attracted towards the targeted shape we want to detect. If we define a snake by a collection of $n$ points, $v_i = (x_i, y_i)$; where $i=1...n$, then the energy function will be defined by the following equation:
The internal energy is the part that depends on intrinsic properties of the snake, such as its length or curvature. It's the sum of elastic energy and bending energy.

**INTERNAL ENERGY**

Total internal energy of the snake can be defined as

\[
E_{\text{int}} = E_{\text{elastic}} + E_{\text{bending}} = \frac{1}{2} \int \left( \alpha |v_s|^2 + \beta |v_{ss}|^2 \right) ds
\]

**ELASTIC ENERGY**

- The curve is treated as an elastic rubber.
- It stops stretching by introducing tension.

\[
E_{\text{elastic}} = \frac{1}{2} \int \alpha(s) |v_s|^2 \, ds \\
v_s = \frac{dv(s)}{ds}
\]

- \( \alpha(s) \): allows us to control elastic energy along different parts of the contour, considered to be constant for many applications.
- Responsible for shrinking of the contour.

**BENDING ENERGY**

- It is defined as sum of squared curvature of the contour.

\[
E_{\text{bending}} = \frac{1}{2} \int \beta(s) |v_{ss}|^2 \, ds
\]

- \( \beta(s) \): plays a similar role to \( \alpha(s) \).
**EXTERNAL ENERG**

The external energy (image energy) of a contour is the external forces on the snake. This determines its relationship to the image. It is dependent on the contour location in the image. The energy is minimal when the contour is exactly on the shape.

\[ E_{ext} = \int E_{image}(v(s))ds \]

\( E_{image} \) variation depends on the image gradient, some examples:

- \[ E_{image}(x, y) = -|\nabla I(x, y)|^2 \]
- \[ E_{image}(x, y) = -|\nabla (G_\sigma(x, y)*I(x, y))|^2 \]

**COMBINING ENERGIES**

A snake being used for image analysis attempts to minimize its total energy, which is the sum of the internal and external energies.

**ACTIVE CONTOUR (SNAKE) ALGORITHM**

- The contour \( v(s) \) is represented by a set of control points \( v_0, v_1, \ldots, v_{n-1} \)
- Force equations applied to each control point separately.
- Each control point allowed moving freely under the influence of the forces.
- The energy and force terms are converted to discrete.

The general idea approximately in any active contour algorithm can be defined as

**While Stop Criteria**

For all active contour points

For all proximity points

Energy computation

End - For

Minimization to select the best point

End - For

End - While
Greedy algorithm is a technique for fast iteration of snake active contours, simply starting with discretizing initial curve given by user to n-points and for each point applies this equation:

\[ E_i = \_E_{\text{cont}}(v(s)) + \_E_{\text{curv}}(v(s)) + E_{\text{image}}(v(s)) \]

- the higher \( \alpha \), the distance between points is minimized
- The higher \( \beta \), the angles are maximized.
- High Image Gradient = Low Energy value.
- The higher \( \gamma \), the more important image edges are increased.

Greedy is faster than Amini’s \( O(nm^3) \) algorithm (dynamic programming method), being \( O(nm) \) for a contour having n points which are allowed moving to any point in a neighborhood of size m at each iteration. During each iteration, a neighborhood of each point is examined and the point in the neighborhood giving the smallest value for the energy term is chosen as the new location of the point. After moving the snake during each iteration, an additional step should be processed for corner detection. It is done by setting \( \beta = 0 \) for a particular point if all of the following conditions are satisfied at the end of each iteration:

1) Curvature is larger than that of both previous and next points.
2) Curvature is above a threshold.
3) Edge strength is above a threshold.
Greedy has a main drawback as it’s high sensitive to noise both Gaussian and Salt & Pepper Ones, So before defining initial points we need to apply Gaussian Blur image then apply a Median Filter to give more accurate results.
DISADVANTAGE OF SNAKES

- They can often get stuck in local minima states.
- They often overlook minute features in the process of minimizing the energy over the entire path of their contours.
- Their accuracy is governed by the convergence criteria used in the energy minimization technique; higher accuracies require tighter convergence criteria and hence, longer computation times.

RESULTS

Figure (6), Results of segmentation of endocardium in different slices and time frames using ACM with $\alpha=0.2$, $\beta=1$, $\gamma=1$. (Resolution= 1mm/pixel)
Review on results:

From previous results, although change in resolution and SNA changes but same values of parameters valid for segmentation, so no need for changing them for a wide range of MRI datasets and make it input variable from user.

Figure (7), Results of segmentation of myocardium (both epi- and endo-cardium) in different slices and time frames using ACM with $\alpha=0.2$, $\beta=1$, $\gamma=1$ (resolution $= 1.4\text{mm/pixel}$)
Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. The concept of optical flow was introduced by the American psychologist James J. Gibson in the 1940s to describe the visual stimulus provided to animals moving through the world. James Gibson stressed the importance of optic flow for affordance perception, the ability to discern possibilities for action within the environment. Followers of Gibson and his ecological approach to psychology have further demonstrated the role of the optical flow stimulus for: the perception of movement by the observer in the world; perception of the shape, distance and movement of objects in the world; and the control of locomotion. Recently the term optical flow has been co-opted by roboticists to incorporate related techniques from image processing and control of navigation, such as motion detection, object segmentation, time-to-contact information, focus of expansion calculations, luminance, motion compensated encoding, and stereo disparity measurement [15].

**OPTICAL FLOW ESTIMATION**

Sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements. Fleet and Weiss provide a tutorial introduction to gradient based optical flow. John L. Barron, David J. Fleet, and Steven Beauchemin provide a performance analysis of a number of optical flow techniques. It emphasizes the accuracy and density of measurements.

The optical flow methods try to calculate the motion between two image frames which are taken at times $t$ and $t + \Delta t$ at every voxel position. These methods are called differential since they are based on local Taylor series approximations of the image signal; that is, they use partial derivatives with respect to the spatial and temporal coordinates.

For a 2D+t dimensional case (3D or n-D cases are similar) a voxel at location $(x, y, t)$ with intensity $I(x, y, t)$ will have moved by $\Delta x$, $\Delta y$ and $\Delta t$ between the two image frames, and the following image constraint equation can be given:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

From these equations it follows that:

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0$$
Or
\[
\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \frac{\Delta t}{\Delta t} = 0
\]

Which results in
\[
\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0
\]

where $V_x, V_y$ are the $x$ and $y$ components of the velocity or optical flow of $I(x, y, t)$ and $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the derivatives of the image at $(x, y, t)$ in the corresponding directions. $I_x, I_y$ and $I_t$ can be written for the derivatives in the following.

Thus:
\[
I_x V_x + I_y V_y = -I_t
\]

Or
\[
\nabla I^T \cdot \vec{V} = -I_t
\]

This is an equation in two unknowns and cannot be solved as such. This is known as the aperture problem of the optical flow algorithms. To find the optical flow another set of equations is needed, given by some additional constraint. All optical flow methods introduce additional conditions for estimating the actual flow.

OPTICAL FLOW APPLICATIONS

Motion estimation and video compression have developed as a major aspect of optical flow research. While the optical flow field is superficially similar to a dense motion field derived from the techniques of motion estimation, optical flow is the study of not only the determination of the optical flow field itself, but also of its use in estimating the three-dimensional nature and structure of the scene, as well as the 3D motion of objects and the observer relative to the scene, most of them using the Image Jacobian.

Optical flow was used by robotics researchers in many areas such as: object detection and tracking, image dominant plane extraction, movement detection, robot navigation and visual. Optical flow information has been recognized as being useful for controlling micro air vehicles.

The application of optical flow includes the problem of inferring not only the motion of the observer and objects in the scene, but also the structure of objects and the environment. Since awareness of motion and the generation of
mental maps of the structure of our environment are critical components of animal (and human) vision, the conversion of this innate ability to a computer capability is similarly crucial in the field of machine vision. The optical flow vector of a moving object in a video sequence.

**LUCAS KANADE METHOD [16]**

One of the more popular methods for optical flow computation is Lucas and Kanade’s local differential technique, figure (8). This method involves solving for the optical flow vector by assuming that the vector will be similar to a small neighborhood surrounding the pixel. It uses a weighted least squares method to approximate the optical flow at pixel \((x, y)\).

\[
E_v = \sum_{p \in \Omega} W^2(p)[\nabla I(p) \cdot v + I_t(p)]
\]

Where \(\nabla I(p)\) and \(I_t(p)\) represent the spatial gradient and temporal gradient neighboring pixel \((p)\) respectively. \((v)\) is the optical flow vector for pixel \((x, y)\) and \(W(p)\) is the weight we associate with that neighboring pixel.

For each pixel we find an optical flow vector consistent with the neighboring spatial and temporal gradients. We consider a surrounding neighborhood of size \(n\) where each neighbor is represented as \(p_i\). Ignoring the weights for an instant, this equation simply sums the error of applying the flow vector \(v\) to the spatial and temporal gradients of all the surrounding neighbors using the optical flow equation of Equation 3. If this considered flow vector is inconsistent with the spatial and temporal gradients of some neighbors, the error will be higher. The weights are used to diminish the importance of distant neighbors. The farther the neighbor is from the pixel the smaller the associated weight. In this way, the influence of more distant pixels is reduced. This least squares problem can be solved by:

\[
A^T W^2 A v = A^T W^2 b
\]

\[
A = \begin{bmatrix}
  \nabla I(p_1) \\
  \nabla I(p_2) \\
  \vdots \\
  \nabla I(p_n)
\end{bmatrix} \\
W = \text{diag}[W(p_1), W(p_2), \ldots, W(p_n)] \\
b = \begin{bmatrix}
  -I_t(p_1) \\
  -I_t(p_2) \\
  \vdots \\
  -I_t(p_n)
\end{bmatrix}
\]
where $A$ is a vector of all the spatial gradients of all the $n$ neighbors of the neighborhood $\Omega$, $W$ are the weights for each neighbor and $b$ is a vector of the temporal gradients. Solving for $v$ gives us

$$v = (A^T W^2 A)^{-1} A^T W^2 b$$

where

$$A^T W^2 A = \left[ \begin{array}{cc} \sum W^2(p) I_x^2(p) & \sum W^2(p) I_x(p) I_y(p) \\ \sum W^2(p) I_x(p) I_y(p) & \sum W^2(p) I_y^2(p) \end{array} \right]$$

and can be easily found by summing the spatial derivatives $I_x, I_y$ of the neighborhood surrounding the current pixel along with the associated weights. If this matrix is invertible, $v$ can be found.

Figure (8), Locus kanade algorithm implementation flow chart
RESULTS

Figure(9), Results of using Locus kanade Tracking method.
Frame 1 with initialized contour using ACM,
Frame 2 resultant contour from locus kanade tracking method
AUTOMATIC SEGMENTATION

Any Quantitative analysis of cardiac function depending on 4D CMRI images requires segmentation of the inner and outer walls of L.V in different slices and different time frames to obtain an overall look of cardiac cycle. Although manual segmentation is considered the golden reference for myocardial segmentation, it is a tedious and time-consuming process.

Several automatic (and semi-automatic) techniques for segmentation, like using 2D active contour models [17], region growing [18], or morphological operators [19]. Notwithstanding, these techniques ignore the temporal relationship between the consecutive time frames which can be useful for compatible segmentation of the myocardium.

So we used proposed algorithm by Ahmed S. Fahmy et al [20], the technique can provide a rough estimate of the (initial) contour location at each time frame. This paper formulating a new cost function that include the OF constraint in addition to myocardial contour properties such as smoothness and elasticity.

ACM GUIDED WITH OPTICAL FLOW ENERGY METHOD

ENERGY FUNCTION OF THE OPTICAL FLOW CONSTRAINT

Given an image sequence \( I(x, y; t) \) with \( t=1, 2, .., n \), it is required to estimate the displacement (or equivalently, the velocity) of each image point \( (x,y) \), from one time frame to the other.

At a given time frame, \( t \), and point \( P_i := (x_i, y_i) \) in the optical flow framework, the displacements \( u_i^t, v_i^t \) can be estimated by solving the optical flow constraint equation given by

\[
E_{OF}(u_i^t, v_i^t) = [u_i^t, v_i^t, 1].J(\nabla I)\cdot \begin{bmatrix} u_i^t \\ v_i^t \\ 1 \end{bmatrix}
\]

where \( I_x, I_y \) are the first-order derivative of the image in the x- and y- direction respectively and

\( \nabla I := [I_x \quad I_y \quad I_t] \) and \( J(\nabla I) := (\nabla I) \cdot (\nabla I)^T \).

If we have a set of \( N \) points forming a contour, then the energy term for the contour at a time frame, \( t \), is given by:

\[
E_{OF}^t = \sum_{i=1}^{N} E_{OF}(u_i^t, v_i^t)
\]
CONTOUR CONSTRAINTS

In this proposed algorithm, the property that myocardial contours are smooth and elastic is used as the additional constraints. These additional assumptions can be formulated in a way similar to that of the active contour model (ACM) energy function. That is, at a given time frame, $t$, the following energy term is calculated at each contour point, (see [11] for more details of the ACM energy function),

$$E_{ACM} = E_{Elasticity} + E_{Curvature}$$

where the first- and second-order derivatives in the above equation measure the amount of contour stretching (elasticity energy) and contour bending (curvature energy) respectively. The weighting parameters, $\alpha$ and $\beta$, represent the contour internal properties of elasticity and curvature respectively.

CONTOUR-CONSTRAINED OPTICAL FLOW ENERGY FUNCTION

From previous, an energy function that contains the temporal information from the image sequence in addition to the contour properties. That is, the desired contour that delineates the myocardial border is the one that minimizes the following cost function:

$$E = E_{OF} + E_{ACM}$$

$$E_{ACM}^t = \sum_{i=1}^{N} \left( \alpha \left| \frac{dp_i^t}{ds} \right|^2 + \beta \left| \frac{d^2 p_i^t}{ds^2} \right|^2 \right) + \gamma E_{OF}(u_i^t, v_i^t)$$

A greedy algorithm is used to solve the above minimization problem, where the weighting factors ($\alpha$, $\beta$ and $\gamma$), are preset experimentally and kept fixed for the different runs of the algorithm (i.e. are not changed from dataset to another).

Figure (10) shows a block diagram of the proposed algorithm and Figure 2 shows how the EOF energy term is calculated for the two consecutive time frames $t-1, t$. 


Figure (10), Flow chart for ACM guided by OF energy [20]

Figure (11), Calculation of $E_{OF}$ from two consecutive time frames $t-1$, $t$ [20]
Figure (12), Tracking of L.V walls using ACM guided by OF energy starting with initialized contours in T=1 using ACM. (Resolution=1mm/pixel)
Figure (13), Tracking of L.V walls in another dataset using ACM guided by OF energy starting with initialized contours in T=1 using ACM. (Resolution=1.4mm/pixel)

Note: in some time frames incorrect tracking for some points can be edited by user in runtime
Figure(14), Tracking of L.V epicardium wall in another slice using ACM guided by OF energy starting with initialized contours in T=1 using ACM. (Resolution=1.4mm/pixel)

Note: in some time frames incorrect tracking for some points can be edited by user in runtime
CARDIAC PARAMETERS

Quantitative volume, mass and function measurements are often mandatory for making an accurate diagnosis. Wall motion and Thickening of myocardium help in diagnosis a lot of valves diseases. Thanks to an excellent depiction of both endocardial and epicardial borders, cardiac magnetic resonance imaging (MRI) provides precise information of wall motion (WM) and thickening (WT), suitable for quantitative analysis.

Calculating diagnostically useful parameters, such as wall motion, myocardial thickening, myocardial mass measurements, end diastolic volume (EDV), end systolic volume (ESV), ejection fraction (EF), and cardiac output (CO) start after segmenting LV endocardium and epicardium in slices and time frames in CMRI dataset.

In this section we describe a fully automatic method for regional functional analysis of all temporal frames of the MRI study. The segmentation step operates in short axis.

The quantitative analysis included two steps:
1. Tracing the endocardial contour in end diastolic and end-systolic phase.
2. Tracing the epicardium border in end diastolic and end-systolic phase.

LEFT VENTRICULAR (L.V.) FUNCTION ANALYSIS

A. GLOBAL L.V. FUNCTIONS

1. VOLUME-TIME CURVES (VTC)

VTC were obtained by calculating the volume the area of the contours surfaces and known the thickness of the slices and the gap between, using a simple interpolation method to estimate the gap volume.

2. EJECTION FRACTION (EF)

EF represents the volumetric fraction of blood pumped out of the ventricle (heart) with each heart beat or cardiac cycle [21].

\[ EF = \frac{(EDV - ESV)}{EDV} \]
3. STROKE VOLUME (SV)

SV is the volume of blood pumped from one ventricle of the heart with each beat. SV is calculated using measurements of ventricle volumes from an echocardiogram and subtracting the volume of the blood in the ventricle at the end of a beat (called end-systolic volume) from the volume of blood just prior to the beat (called end-diastolic volume) [22].

\[ SV = EDV - ESV \]

B. REGIONAL MYOCARDIAL THICKENING

Regional myocardial contraction is subject to translational artifacts. It is measured using a myocardial center surface equidistant to the epicardium and endocardial surfaces. Chords are drawn from the endocardial center to their intersection with the epicardium wall. The length of each chord is a measure of the local wall thickness. Then we subdivision LV to 4 segments each segment with average thickness of its chords.

We subdivide the myocardium to 4 sectors the first sector is the left one and the order is clock wise, each sector describe the average thickness of this location in myocardium we took 4 slices one apical, two mid, and one basal. The biggest one is the apical the two in the middle is mid and the smallest one is the apical, then we calculated the thickness for each sector for the four slices threw all time frames.

We divide the range into 4 intervals after normalization of the thickness values range from 0 to 255.

![Color mapping for normalized thickness values](image)
RESULTS

Following figure represents our results for segmented dataset using proposed algorithms.

**Volume Chart**

![Volume Chart](image)

Figure (16), Volume chart showing change in blood, epicardium and myocardium volume w.r.t time.
After calculating the global and local parameters, we compare our results with the typical results.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Typical value</th>
<th>Normal range</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td>end-diastolic volume (EDV)</td>
<td>120 mL</td>
<td>65–240 mL</td>
<td>146.256 mL</td>
</tr>
<tr>
<td>end-systolic volume (ESV)</td>
<td>50 mL</td>
<td>16–143 mL</td>
<td>58.176 mL</td>
</tr>
<tr>
<td>stroke volume (SV)</td>
<td>70 mL</td>
<td>55–100 mL</td>
<td>88.08 mL</td>
</tr>
<tr>
<td>ejection fraction (Ej)</td>
<td>58%</td>
<td>55–70%</td>
<td>60.22317 %</td>
</tr>
<tr>
<td>heart rate (HR)</td>
<td>75 bpm</td>
<td>60–100 bpm</td>
<td>60 bpm</td>
</tr>
<tr>
<td>cardiac output (CO)</td>
<td>5.25 L/minute</td>
<td>4.0–8.0 L/min</td>
<td>5.2848 L/minute</td>
</tr>
</tbody>
</table>

Table (3), Regional myocardial Thickening chart for slices in different time frames [22]
We used iTextPDF Java library to generate dynamic PDF report figure(4) after calculating the parameters. The PDF contains graphs describing the volume of myocardium, blood, and epicardium volumes. It contains also the measured values of EDV, ESV, EF, SV, CO and comparison between the results of calculations, the typical results, and the normal range.

**Cardiac MRI analysis report**

Report generated by Javy, Sat Jul 06 19:39:40 EET 2013

**Patient name:**

**Figure (18), Result PDF containing charts and clinical parameters**
4D VISUALIZATION OF L.V. WALLS

Myocardial motion is a good indicator of coronary artery disease. Abnormal motion of the heart wall during contraction is often caused by a reduction (or occlusion) in blood flow to a region of myocardium or as a result of dead tissue, following an infarct. This phenomenon often occurs in localized areas of the heart. As a result, in both diagnosis relating to coronary artery disease and post infarct assessment, it is important to view and study abnormal wall motion. Using segmented 2-D MRI images with previously implemented tools and algorithms can be stacked to form a 3-D volume, with high soft tissue contrast. Furthermore, the images can be acquired over time. This makes 4-D (3-D + time) analysis possible [23].

Based on 3D reconstruction theories, we can build a 3D model using contours defined from segmentation process of LV inner and outer walls, we used triangulation method in our platform for better and faster modeling.

For our approach of building a complete open source framework, java3d is used as a cross platform scene graph for visualizing built 3D model.

JAVA3D

For writing high-performance and hardware-accelerated software, the use of a powerful low level API is necessary. Today, two widespread APIs exists: OpenGL and Direct3D. Direct3D is included in the DirectX package from Microsoft. The advantage of Direct3D is that it’s well supported by current 3D computer games, which often require the newest DirectX version. The disadvantage is that the user requires a Windows platform. That’s the advantage of OpenGL because it is supported by most operating systems, making it platform independent [24].

For Java two common hardware-accelerated 3D APIs exists, JOGL and Java3D. Using JOGL, the programmer is heavily confronted with the low level rendering techniques, like the rendering pipeline. The advantage is a faster rendering process. Java3D tries to make the internal rendering methods transparent for the programmer. Coding with Java3D gives the programmer the possibility to decide how deep he wants to manipulate the rendering process. Java3D comes with a design concept, which allows to model whole 3D scenes easily.
A. WHAT IS JAVA3D?

Java3D is a standard extension to the Java SDK. It is able to model and render complete 3D scenes using only Java code. Java3D is a wrapper collection for either the OpenGL or the DirectX interface. Instead of manipulating the render pipeline directly, which is a common method in the 3D programming scene; this pipeline is by default completely transparent to the programmer and is handled automatically by Java3D. However, the developer feels free to decide how near the rendering pipeline he wants to program his application. Certainly the programmer gets more control over the rendering process as he goes deeper into the low level programming. Because of the fact that Java3D is just a wrapper API above the two powerful graphic interfaces OpenGL and DirectX, the Java developer gets access to nearly all of their features, with full hardware acceleration and platform independence [24].

Because Java3D is not included in the JRE, a Java3D distribution has to be installed on the system on which the Java3D applications should run. The application has to inform the user to install Java3D on his machine or include a whole Java3D distribution in its own jar file.

B. THE SCENE GRAPH OF JAVA 3D

The most important concept of Java3D is the 3D scene graph. This data structure is able to hold all pieces of information, which are required for a complete 3D scene. A 3D scene graph can represent all possible 3D scenes. The graph is a tree, so every node has just one parent node and any number of child nodes. There is also the restriction that tree paths do not form cycles. So there is only one path from one node to another. The graph has only one root, the virtual universe. Figure (19) shows Scene graph of our 3D modeling.
Figure (19), scene graph for platform of construction and rendering 3D model.
PROCEDURE

After segmentation of 4D (3D + time) MRI dataset, resultant contours of LV, both epicardium and endocardium walls are collected and re-arranged figure (2) to build 3D model of myocardium figure (1). The model is constructed from separated segments of surfaces, each surface is built using triangulation theory figure (3).

Figure (20), a- Top view of epicardium wall showing its texture

B- Side view of epicardium showing side view

C- 3D wired model of epicardium using 6 slices in same time point
Figure (21), Sketch of reconstructing of myocardium using 2D images

Figure (22), Building surface using triangulation theory between epicardium in slice n and slice n+1

### STEPS OF BUILDING SEGMENT OF MODEL

1. Get representing contours of segment.
2. Check if numbers of points for both contours are equal, if not do oversampling for minimum one.
3. Save points in array (TriangleStripArray) as shown in figure(22) and figure(24), point from contour and next one from other contour.
4. Draw a triangle between point and 2 points after.
5. Build a shape3D (ref) object for drawn element and set the appearance.
6. Collect elements in BranchGroup to render together in virtual universe.

Our 3D model consists of (m-1)segments each segment consists of inner, outer wall of L.V., top wall and bottom wall as shown in figure(5).
RESULTS

Following figures showing motion of L.V 3D model w.r.t time from different views.

Figure (23), Segment of LV myocardium, Epicardium in red and myocardium in white.

Figure (24), surface between epicardium and endocardium

Figure (25), Side view of L.V 3D model in different time frames showing motion, epicardium in white and endocardium in red.
Figure (26), Top view of upper segment in L.V 3D model in different time frames showing motion, epicardium in white and endocardium in red
REFERENCES