POLITECNICO DI MILANO

Facoltà di Ingegneria dei Sistemi

Corso di laurea Specialistica in Ingegneria Biomedica

Dipartimento di Bioingegneria

Left Ventricle Segmentation Based on Real-Time MR Image Projections

Relatore:	Prof. Pietro CERVERI
Correlatore:	Prof. Nikolaos V. TSEKOS

Tesi di Laurea di:
Daniele FARRIS
Matr. n. 755118

ANNO ACCADEMICO 2011/2012

Contents

Abstract

Real-time image-guided procedures (manual or robot-assisted) are emerging due to potential improvement in patient management and reduction in the overall cost. To extract and incorporate information on-the-fly from intraoperative images requires new ultra-fast processing techniques. In this context, the work presents a method to process real-time MR images using signal intensity projections and extract boundary points on the tissues surfaces. The method was applied for segmenting blood-pool inside the left ventricle of the beating heart by extracting boundary points representing the endocardium wall. Working with intensity projections means processing a 2D signal (i.e. an image) by means of a one-dimensional signal. This reduces the complexity of the problem and makes everything much faster. The projections drawn across the left ventricle real-time MR images show a signal with two main minima on the sides. The two minima (i.e. the two darkest points) represent the two endocardium cores while what is in between them represents the blood-pool inside the ventricle. An off-line algorithm was implemented to extract any number of projections and process them in order to find two points for every projection that identify the endocardium walls. The algorithm consists in six main stages. The first one regards when the user extracts the projections from the slices they choose. The number of projections is arbitrary. It is true that the more projections, the better the segmentation's accuracy. A first filtering step is performed at the very first stage of the algorithm by automatically averaging a given number of adjacent and parallel projections. The second stage of the algorithm is the actual filtering of the acquired signals. Three types were taken into account: moving average, standard FIR filters, splines approximation. Once the projections are filtered the two minima are to be found. The minima corresponding to the lowest grey level in each half projection are chosen. Totally two minima for each projection are found. The minima are to be displaced towards the center of the projection according to a certain criterium. Three criteria were designed by the author to arrest the displacement at an adequate point and the indexes of the corresponding samples are saved. Each one of these indexes composes a time signal representing the displacement of the endocardium wall over the time. This two time signals are lightly filtered in order to overcome small errors possibly due either to the user's drawing or to the algorithm's parameters selection. At last, the actual segmentation is performed by properly connecting the points found through all the previous steps. Although the cardiac segmentation has been explored and studied under several points of view, this work faces this topic with a novel approach, due to the nature of the images. The projections way of processing fits in the real-time MR imaging world very well. The connection between the two lies on a particular MR acquisition technique used for a different purpose. It is called Navigator Echo and it basically consists in acquiring a linear set of voxels instead of an entire image. Though it has not been possible to program the MR scanner yet, through this technique it is certainly possible to directly acquire the projections and speed up the whole process. Moreover, there exist a lot of techniques to connect the points belonging to a 3D cloud. In this work the segmentation was done in a simple way and it does not deal with the 3D surfaces reconstruction. The robustness of the method was verified by comparing the position of the boundary points with the manual ground-truth. The algorithm was tested on a wide dataset coming from diverse subjects and consisting of different heart views. Although there is a small variability among where on the ventricle the projections are drawn (e.g. apical part, aortic root etc.) and what heart cycle phase is being processed (systole or diastole), the average accuracy of the method was proven to be below 2 mm. The ground-truth was drawn on the images as a manual segmentation. The one that regards the projections was automatically drawn from the segmented images as if it was just another projection with two points identifying the real endocardium walls. The difference between the point identified by the algorithm and the point identified by the ground-truth was computed and therefore represented the error. Since the algorithm included a large set of parameters, the optimal parameters were chosen in a way that minimizes the error. The accuracy of the algorithm was found to be a function of time, side of the projection and heart location on which it is drawn. Particularly this last one gave the start to a sensitivity study regarding the different zones on where the projections are drawn. In order to have one value of accuracy, the optimal parameters minimize a figure of merit that averages the side and the time information, preserving the one relative to the location. For a better understanding of the algorithm, a graphical user interface was designed. The GUI allows the user to chose a dataset, draw as many projections as they want and see their on-line processing. The algorithm was simulated with MATLAB® on a PC

Intel® Core^{TM} i7 CPU 960 @3.20GHz. Every projection takes in average 1.5 ms CPU time to be processed. The processing time depends on the parameters. Depending on what the images acquisition frequency is and how many projections the user desires to draw, the algorithm works in real-time. The real-time processing speed allows the method to be applicable for intraoperative generation of dynamic tissues surfaces for guiding interventions in the beating heart.

Sommario

Le procedure real-time image-guided (assistite da robot o manuali) stanno emergendo dati i potenziali miglioramenti riguardanti la gestione del paziente e la riduzione globale dei costi. Estrarre ed incorporare informazioni on-the-fly da immagini intraoperative richiede nuove tecniche di processamento ultra-veloci. In questo contesto si colloca questo lavoro che presenta un metodo per processare immagini MR in tempo reale tramite l'uso di proiezioni di intensitá, estraendo punti sulla superfice dei tessuti. Il metodo é stato usato per la segmentazione del ventricolo sinistro nel cuore battente tramite l'estrazione di punti che rappresentano la parete dell'endocardio. Lavorare con proiezioni di intensitá significa processare un segnale bidimensionale (i.e. un'immagine) tramite un insieme di segnali monodimensionali. Questo semplifica in maniera sostanziale la risoluzione del problema. Le proiezioni tracciate sulle immagini real-time MR del ventricolo sinistro mostrano un segnale dove sono presenti due minimi principali che solno localizzati approssimativamente sui due lati. Questi due minimi (corrispondenti ai due punti piú scuri nell'immagine) rappresentano i punti piú interni alla parete endocardica, mentre ció che é in mezzo rappresenta la camera cardiaca ripiena di sangue. Un algoritmo off-line é stato implementato per estrarre un qualsiasi numero di proiezioni e processarle in modo da identificare le coordinate dei punti che rappresentano le pareti dell'endocardio. L'algoritmo consiste principalmente in sei fasi. La prima fase consiste nellestrazione, da parte dell'utente, di un numero arbitrario di proiezioni dalle immagini. Resta comunque vero che piú il numero é alto, migliore risulterá l'accuratezza della segmentazione. Un primo filtraggio é eseguito durante l'acquisizione delle proiezioni tramite l'averaging automatico di piú proiezioni parallele e adiacenti. La seconda fase dell'algoritmo é l'effettivo filtraggio delle proiezioni. Tre tipi sono stati presi in considerazione: moving average, filtri FIR standard, approssimazione tramite splines. Una volta che le proiezioni sono state filtrate i due minimi devono essere trovati. I minimi che corrispondono al livello di grigio piú basso in ogni semi-proiezione sono scelti. In totale vengono perció trovati due minimi per ogni proiezione. Questi minimi devono essere spostati verso il centro della proiezione secondo un determinato criterio. Tre criteri sono stati progettati dall'autore per far si che lo spostamento si arresti in un punto appropriato e gli indici dei campioni corrispondenti vengano salvati. Ognuno di questi indici descrive un segnale temporale che rappresenta lo spostamento della parete endocardica nel tempo. I due segnali vengono lievemente filtrati per ovviare a piccoli errori possibilmente dovuti alla selezione dell'utente o ai parametri dell'algoritmo. In fine, la vera segmentazione viene effettuata connettendo appropriamente i punti trovati nelle fasi precedenti. Nonostante la segmentazione cardiaca sia stata esplorata e studata sotto vari punti di vista, questo lavoro affronta l'argomento con un approcio innovativo, per via della natura delle immagini. Processare immagini tramite proiezioni entra perfettamente a far parte del mondo dell'MR imaging. La connessione tra i due consiste in una particalare tecnica di acquisizione per MR originariamente usata per uno scopo differente. Si chiama Navigator Echo e consiste nell'acquisire un insieme di voxel giacenti su una linea invece che un'intera immagine. Sebbene non sia ancora stato possibili programmare direttamente lo scanner MR, tramite questa tecnica é possibile acquisire direttamente le proiezioni e perció velocizzare l'intero processo. Esistono inoltre molte tecniche per connettere i punti appartenenti ad una nuvola 3D. In questo lavoro la segmentazione é stata svolta in una maniera semplice, senza affrontare la ricostruzione di superfici 3D. La robustezza del metodo é stata verificata tramite la comparazione dei punti di confine con la ground-truth manuale. L'algoritmo é stato testato su un vasto dataset proveniente da diversi soggetti e formato da diverse viste del cuore. Nonostante una limitata variabilitá riguardante il dove vengano estratte le proiezioni nel ventricolo (e.g. zona apicale, zona aortica, etc.) e la fase del ciclo cardiaco (sistole o diastole), é stato provato che l'accuratezza media del metodo si mantenga sotto i 2 mm. La ground-truth é stata disegnata sulle immagini come segmentazione manuale. Quella che riguarda le proiezioni é stata automaticamente estratta dalle immagini segmentate come se fosse un'altra proiezione. La differenza tra il punto che viene identificato tramite algoritmo e quello identificato dalla ground-truth rappresenta l'errore. Dal momento che l'algoritmo comprende un vasto insieme di parametri, i parametri ottimi sono stati scelti in modo tale che minimizzino quest'errore. Si è verificato che l'accuratezza dell'algoritmo fosse funzione del tempo, del lato della proiezione e del luogo cardiaco nel quale essa viene disegnata. In particolare questo ha dato il via ad uno studio di sensitivitá riguardante le zone differenti sulle quali le proiezioni sono state disegnate. In modo da avere un solo valore di accuratezza, i parametri ottimi minimizzano una cifra di merito che media l'informazione relativa al lato della proiezione e al tempo, preservando quella relativa alla locazione. Per una comprensione migliore dell'algoritmo, é stata sviluppata un interfeccia grafica. Quest'ultima permette all'utente di selezionare il dataset desiderato, estrarre quante proiezioni si desiderino e osservare il processamento on-line. L'algoritmo é stato simulato con MATLAB® su un PC Intel® Core™ i7 CPU 960 @3.20GHz. Ogni proiezione richiede in media 1.5 ms CPU per essere processata. Il tempo impiegato dipende dalla scelta dei parametri. A seconda di quale sia la frequenza d'acquisizione delle immagini e di quante proiezioni l'utente desideri estrarre, l'algoritmo funziona in tempo reale. La sua velocitá permette che il metodo sia applicabile per la generazione intraoperatoria e dinamica di superfici dei tessuti. Quest'ultima puó essere sfruttata per guidare interventi chirurgici a cuore battente.

Chapter 1

Introduction and state of the art

This thesis project places itself among several different topics. It takes place fundamentally in the medical field, particularly in the field of the image processing. The distinctive feature of it is that it deals with a very basic and renown topic such as the left ventricle segmentation in a quite different way from it has been treated before in the medical field, due both to the type of imaging it uses and the way it extracts information off of it.

1 Cardiac magnetic resonance imaging

Cardiac magnetic resonance imaging (CMRI) is a widely used procedure in the clinical world for many purposes. For this reason we will quickly go over some of its applications and some of the method used during the procedure. Let us start with the type of imaging it concerns, MRI. Magnetic resonance imaging (MRI) is a noninvasive medical tool that helps physicians diagnose and treat medical conditions. It uses a powerful magnetic field, radio frequency pulses, and a computer to produce detailed pictures of organs, soft tissues, bone, and virtually all other internal body structures. The images can then be examined on a computer monitor, transmitted electronically, printed, or copied to a CD. MRI does not use ionizing radiation (x-rays). Detailed MR images allow physicians to better evaluate various parts of the body and determine the presence of certain diseases that may not be assessed adequately with other imaging methods such as x-ray, ultrasound, or computed tomography (also called CT, MDCT, or CAT scanning). A contrast agent, such as gadolinium (gad-oh-LIN-e-um), might be injected into a vein during cardiac MRI. The substance travels to the heart and highlights the heart and blood vessels on the MRI pictures. This contrast agent often is used for people who are allergic to the dyes used in CT scanning. Cardiac MRI imaging is performed to help:

- evaluate the anatomy and function of the heart, valves, major vessels, and surrounding structures (such as the surrounding pericardial sac).
- diagnose a variety of cardiovascular (heart and/or blood vessel) disorders such as tumors, infections, and inflammatory conditions.
- detect and evaluate the effects of coronary artery disease such as limited blood flow to the heart muscle and scarring within the heart muscle after heart attack.
- plan a patient's treatment for cardiovascular disorders.
- monitor a patient's progression over time.
- evaluate the anatomy of the heart and blood vessels in children with congenital cardiovascular disease.

Using cardiac MRI, physicians can:

- examine the size of the heart chambers and the thickness of the heart wall.
- determine the extent of myocardial (heart muscle) damage and effect on heart function caused by a heart attack or progressive heart disease.
- detect the buildup of plaque and blockages in blood vessels.
- assess a patient's recovery following treatment.
- evaluate the heart anatomy function, heart valve function, and vascular blood flow both before and after surgical repair of congenital cardiovascular disease in children and adults.

Cardiac MR scanners require modern electronics. 'Open' magnets are not so good for cardiac scanning as they do not cope with the beating heart very well. There are two magnet strengths mainly in use in CMR - 1.5 tesla and 3 tesla. The 3 tesla can potentially double the amount of information acquired in a scan. It offers particular advantages for perfusion [1]. The downsides of 3 tesla are cost, energy usage requirements, and potentially artifacts degrading the pictures.

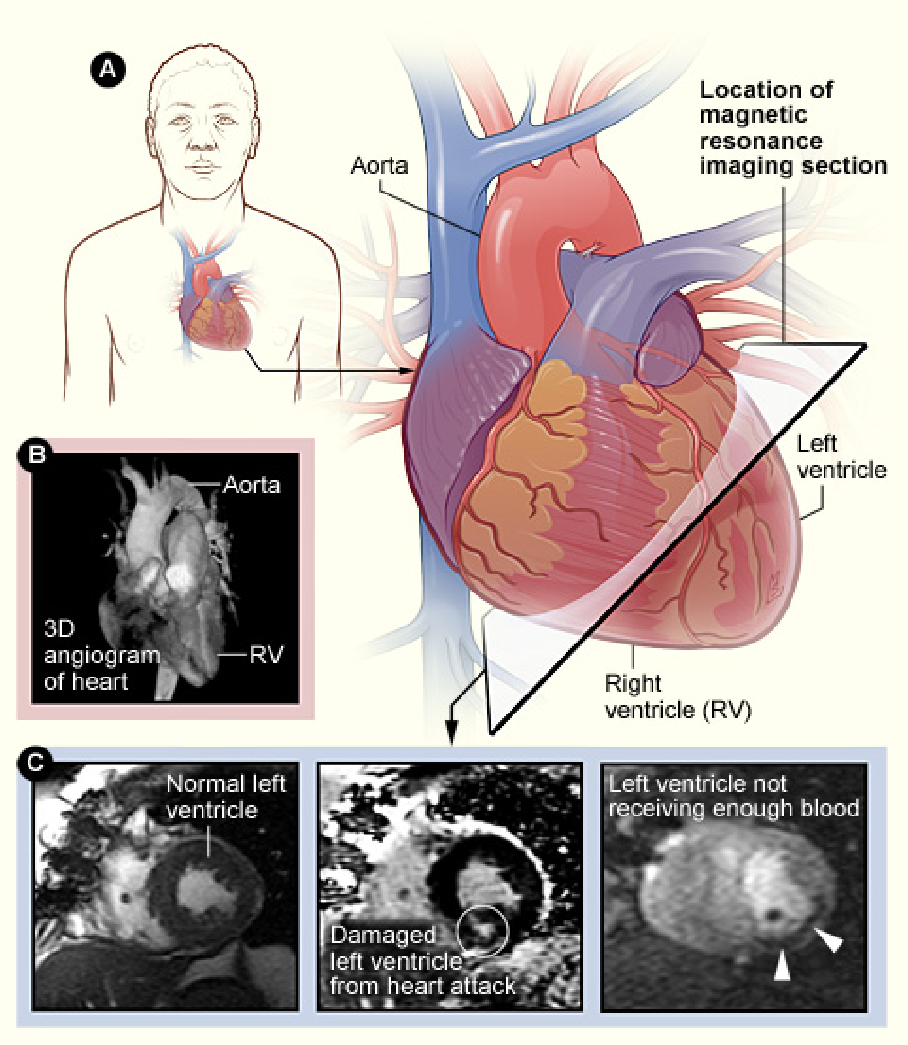
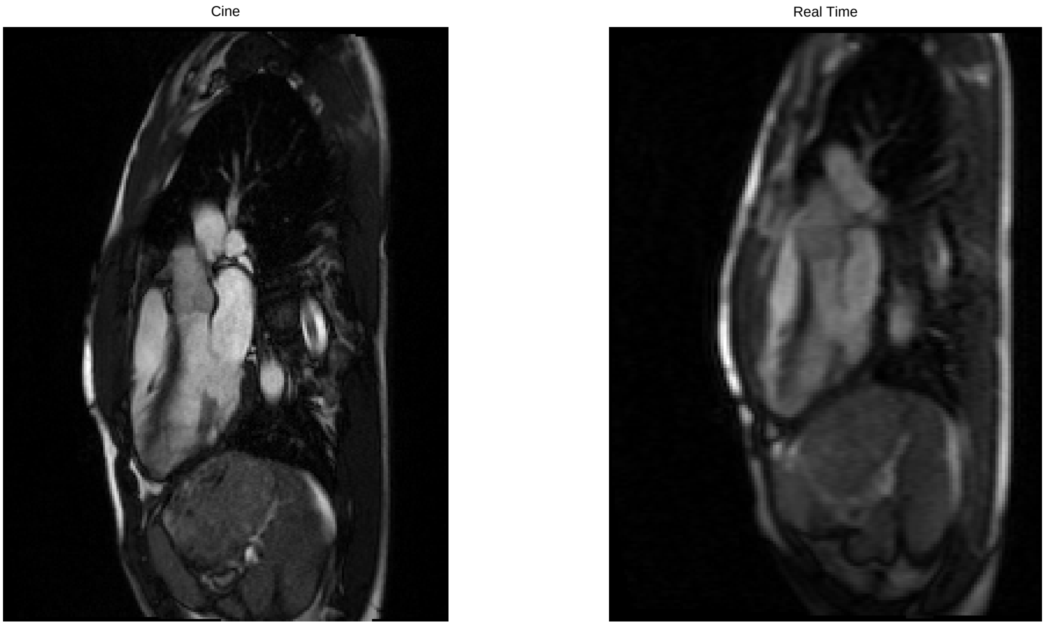

A
Aorta
Location of magnetic resonance imaging section
Left ventricle
Right ventricle (RV)
B
Aorta
3D angiogram of heart
RV
C
Normal left ventricle
Damaged left ventricle from heart attack
Left ventricle not receiving enough blood

Figure 1.1: Figure A shows the heart's position in the body and the location and angle of the MRI images shown in figure C. Figure B is a MRI angiogram, which is sometimes used instead of a standard angiogram. Figure C shows MRI pictures of a normal left ventricle (left image), a left ventricle damaged from a heart attack (middle image), and a left ventricle that isn't getting enough blood from the coronary arteries (right image) [2].

2 Real-time magnetic resonance imaging

Real-time medical image processing is becoming a new challenge due to its potential application for many purposes. Although there is a project of a real-time CT scanner [3] and some near-real-time applications involving the tomography [4], the most often used real-time types of imaging are limited to fluoroscopy and ecography (ecocardiography for cardiac purposes). Fluoroscopy, as well as CT, has the problem of high dose of ionizing radiation issued to the patient. On the other hand, ultrasound images cannot explore the body much deep and they can mostly show only transversal images of the human body (and of the heart for ecocardiography). Through CMR it is possible to obtain very accurate images of the heart. The acquisition sequence is usually triggered by an on-line ECG. Images are acquired at a given ECG phase and then averaged to increase the signal-to-noise ratio. With this method we lose a lot of information regarding the dynamic of the heartbeat. To overcome this problem, in the last couple of years, MR research has come up with real-time magnetic resonance. Since the types of images the designed algorithm is going to deal with are of this kind, we will briefly go through the real-time magnetic resonance functioning and applications.It refers to the continuous monitoring (”filming”) of moving objects in real-time. Because MRI is based on time-consuming scanning of k-space, real-time MRI was possible only with low image quality or low temporal resolution. Using an iterative reconstruction algorithm these limitations have recently been removed: a new method for real-time MRI achieves a temporal resolution of 20 to 30 milliseconds for images with an in-plane resolution of 1.5 to 2.0 mm [5]. Real-time MRI promises to add important information about diseases of the joints and the heart. In many cases MRI examinations may become easier and more comfortable for patients.



Cine

Real Time

Figure 1.2: A long axis view of the heart in Cine mode (left) and in real-time (right). The difference in the quality is appreciable.

The first real-time MR images did not involve the medical field. In fact, in 1982, Peter Mansfields group from Nottingham reported real-time cine MRI of live rabbit heart. Using single shot echo planar imaging (a technique previously described by the same group), they produced cine loop images consisting of 6 frames, each taking 32 ms to acquire (32 32 pixels interpolated to 128 128)[6].While early applications were based on echo planar imaging, which found an important application in real-time functional MRI (rt-fMRI)[7], recent progress is based on iterative reconstruction and FLASH MRI [8] [9]. The real-time imaging method proposed by Uecker and colleagues [5] combines radial FLASH MRI[10], which offers rapid and continuous data acquisition, motion robustness, and tolerance to undersampling, with an iterative image reconstruction method based on the formulation of image reconstruction as a nonlinear inverse problem [11] [12]. By integrating the data from multiple receive coils (i.e. parallel MRI) and exploiting the redundancy in the time series of images with the use of regularization and filtering, this approach enhances the possible degree of data undersampling by one order of magnitude, so that high-quality images may be obtained out of as little as 5 to 10% of the data required for a normal image reconstruction. Because of the very short echo times (e.g. 1 to 2 milliseconds), the method does not suffer from off-resonance effects, so that the images neither exhibit susceptibility artifacts nor rely on fat suppression. While spoiled FLASH sequences offer spin density or T1 contrast, versions with refocused or fully balanced gradients provide access to T1/T2 contrast. The choice of the gradient-echo time (e.g. in-phase vs opposed-phase conditions) further alters the representation of water and fat signals in the images and will allow for separate water/fat movies.

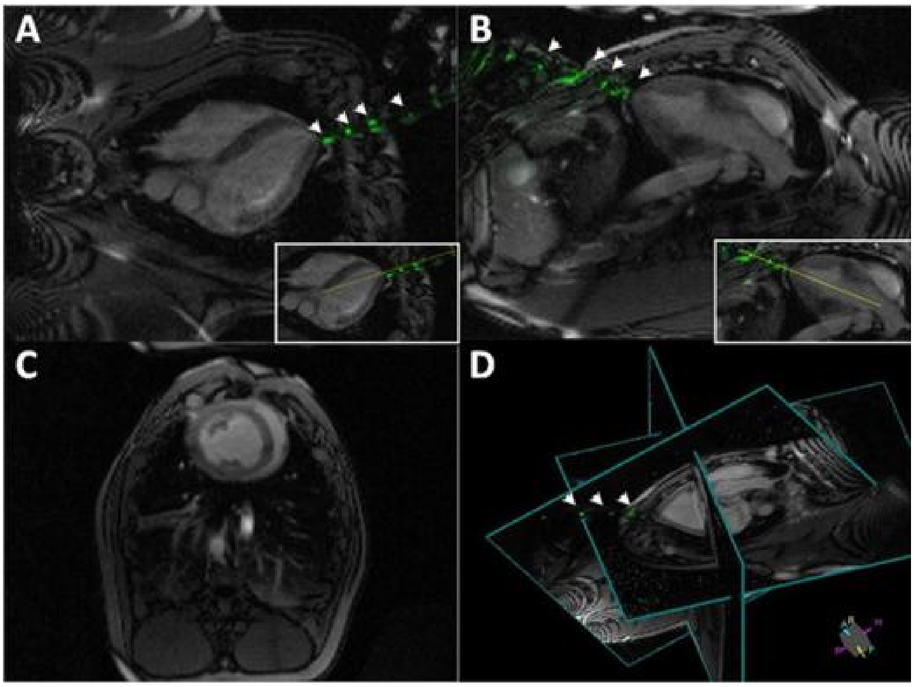


Figure 1.3: Real-time magnetic resonance imaging (MRI)-delineated 3-dimensional trajectory for the active needle by continuously updating 2 perpendicular long-axis imaging planes for needle guidance (A,B), 1 short-axis plane for monitoring left ventricular (LV) function (C), and a 3-dimensional representation of relative positioning for these 3 planes (D). The dotted yellow line in A and B insets indicate a trajectory plan for potential aortic valve intervention [20].

Although applications of real-time MRI cover a broad spectrum ranging from non-medical studies of turbulent flow [13] to the non-invasive monitoring of interventional (surgical) procedures, the most important application making use of the new capabilities is cardiovascular imaging. With the new method it is possible to obtain movies of the beating heart in real-time with up to 50 frames per second during free breathing and without the need for a synchronization to the electrocardiogram.Apart from cardiac MRI other real-time applications deal with functional studies of joint kinetics (e.g., temporomandibular joint [14], knee) or address the coordinated dynamics of the articulators such as lips, tongue, soft palate, and vocal folds during speaking (articulatory phonetics) or swallowing [16]. Applications in interventional MRI, which refers to the monitoring of minimally invasive surgical procedures, are possible by interactively changing parameters such as image position and orientation. An example is the use f RT CMR for catheter ablation [17]. Another example of interventional MRI application is given by the possibility of an aortic valve replacement, already experimented on swines as shown in figure 1.3 [18] [19] [20] [21].

3 Intensity profiles and surface reconstruction

Given the state of the art of the type of imaging this project deals with, the main aim of this work is to find the 3D coordinates of points laying on the endocardium wall with respect to the MR scanner frame. Through the proper connection among these points it is possible to 3D reconstruct the left ventricle. There is actually a huge amount of strategies to reconstruct a surface from a 3D cloud of points. One has just to select the one the most fits in the project. Usually fine algorithms for rendering surfaces are expensive, so one has to find a trade off between processing time and accuracy of the reconstruction. For further information, Fabio published a review paper dealing with all the reconstruction methods [22]. Since the MR images are issued in real-time, it would be challenging for a lot of purposes to reconstruct the ventricle in real-time as well. In fact, there already exist a lot of algorithms for segmenting the left ventricle. For anatomical purposes, no one would ever suggest a real-time CMR test. This is because a CINE (non real-time) test is far more accurate (see figure 1.2) and off-line algorithms allow a fine reconstruction of the left ventricle. In figure 1.4 it is shown a real-time endocardium reconstruction work performed on echocardiographic images [23].

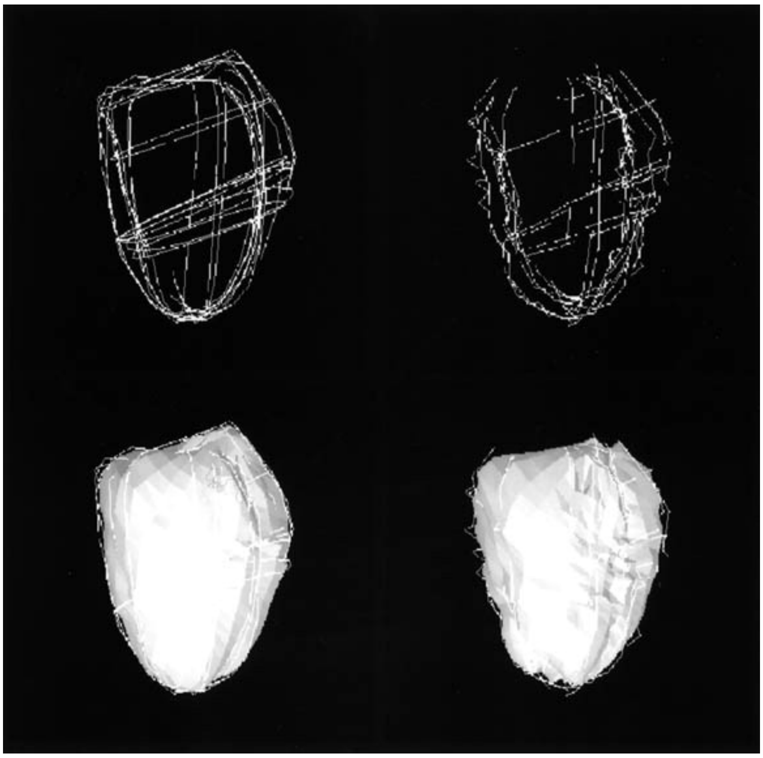


Figure 1.4: Endocardium reconstruction through points selected on echocardiographic images.

The real-time issue has to be contextualized under several points of view and has to meet very strict requirements. It depends on how many parts the algorithm consists of, what kind of platform is used to process everything and so forth. This project does not deal with the contextualization of the algorithm. Though it offers an estimate of how much time it takes to run it on a personal computer. Moreover, all the choices have been taken speed effectively. For this reason, the way the points are identified has to be as fast as possible. For this reason, this thesis embraces both image and signal processing. In fact, it is known that one- dimensional signals are faster to be processed than images (two-dimensional signals). The main idea is to extract a set of one-dimensional signals off of the image that allows us to find the points on the endocardium surface. The one-dimensional signals the algorithm will deal with are nothing but the intensity profiles of the images' grey level. Although we will go deep in this during the following chapter, a brief introduction of what an intensity profile is will be done hereafter. Being an image nothing but a grid whose nodes are associated an integer number (grey level), it is easy to draw a straight line over the image and collect all the grey values this line overlaps on. The following figure 1.5 clearly shows how an intensity profile can be drawn and what form it shows itself.

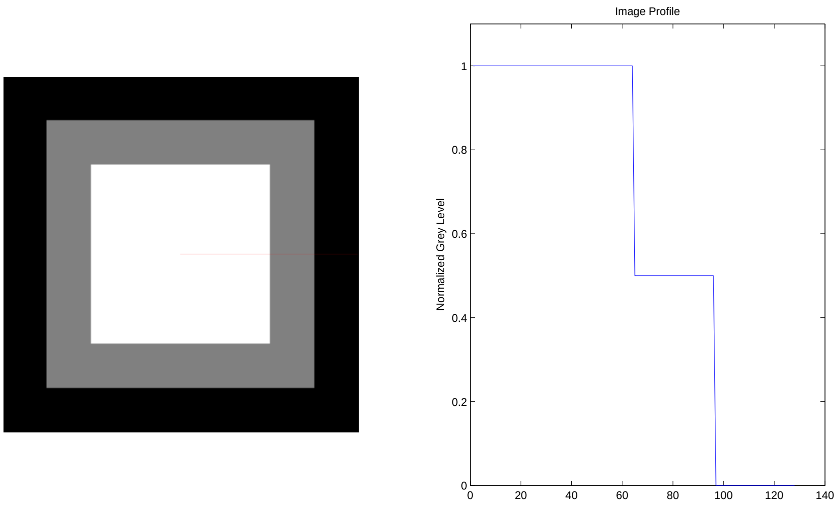


Figure 1.5: Sample image (right) with a drawn profile (red line) and its resulting profile in one dimension (left).

The image profile techniques have been used in the medical field for several purposes. Its capability of processing 2D signals with 1D signals makes any elaboration related to them faster and easier. Nevertheless, quite rarely has it been used as the main method for one given goal [24] [25].

4 Navigator echoes

The intensity profile technique fits in the MRI world for one big reason. With the magnetic resonance scanner it is possible to stimulate and receive signal from an accurately selected set of voxels in the body. Instead of acquiring the whole image and select the straight line from it, it is therefore possible to acquire directly the MR projections, which are the intensity profile acquired directly with a particular MR sequence. The sequence used for this purpose is called navigator echo. Its main use yet concerns the cardiac MR field but in a different sense.This technique measures with an additional quick MR pre-pulse the position, of e.g. the diaphragm [26] before data collecting. Similar respiratory conditions of the patient can be identified and used to synchronize image data acquisition so that respiration-induced image blurring is minimized by either respiratory ordered phase encoding or respiratory gating. The pre-pulse sequence images a small area perpendicular to the structure, which is moving. The contrast of the interface between the diaphragm and the lung should be high to permit easy automatic detection. After data acquisition, the position of the interface is automatically recorded and imaging data are only accepted when the position of the interface falls within a range of prespecified values. This technique has the advantage of greater accuracy than other respiratory gating (therefore used for coronary angiography [27]) and has no need for additional sensing MRI equipment, as the MR system itself provides it. Because of its speed and ease of understanding, navigator echo is a powerful technique suitable for many cardiac MRI purposes [28].

5 MRI Left ventricle segmentation

Numerous methods for segmentation and/or tracking of cardiac MRI in general, and left ventricle in particular, have been proposed in the past two decades. A collection of these techniques are reviewed in [29]. In this section, we present a summary of these methods [30], categorized based on the fundamental computer vision algorithm applied. Higgins et al. [31] designed a semi-automatic method for segmentation of the left ventricle in 3D cardiac CT images. They utilized edge-preserving smoothing filters to reduce random noise while maintaining sharp edges and image features. Sanchez-Ortiz et al. [32] employed a thresholding technique to detect LV boundaries in 2D slices of 3D cardiac ultrasound (US). A fixed threshold value is set based on image intensities while a connected component algorithm is executed on the 3D surface to identify the

LV boundary. Lynch et al. [33] have also utilized intensities in a threshold clustering technique. These methods are efficient and easy to implement, however they are susceptible to intensity variations and low gradients associated with most medical images. At each image point, the gradient is defined as a 2D vector with components representing intensity derivatives in horizontal and vertical directions. Large intensity changes at a boundary result in large gradient magnitudes at the points on the boundary. Having considerable intensity variations at boundaries in an image is one of the main requirements of the edge-based methods, which poses a challenge for CMR, which does not offer great variations in intensity values between various tissue types.Originally formulated by [34], an active contour, also known as snake, is a specific type of deformable model. A snake is a deformable open or closed curve, often represented by a set of control points. A snake glides towards image features while maintaining continuity and smoothness of the curve. Using edges as image features to attract the active contour is a common approach, although the choice of features is application-specific [35]. The energy of the contour expressed as a function of control point indices s is defined in [36].The snake is deformed by minimizing a pre-defined energy function based on internal and external forces. While internal forces control continuity and smoothness of the contour, external forces drive the contour towards image features. Numerous segmentation methods for cardiac MRI have been developed based on active contours. Terzopolouos and McInerney [37] along with Davis [38] provide a detailed report on application of deformable models in medical imaging. In the case of 3D cardiac data, the snake is manually initialized by an expert, using a priori knowledge of the shape constraints at the segmentation problem. A user's interaction may be needed to finely tune the curve in areas of high curvature. A good initial estimate for one slice is used as an initialization for the next slice in the sequence. Therefore, good initialization is required to obtain good segmentation performance. Spreeuwers and Breeuwer [39] propose to estimate position of epicardial and endocardial contours jointly using active contours. A set of concentric radial lines are defined which meet approximately at the center of the LV. The intensity profile along a radial line from the center to the end of the line is derived. The active contour nodes are assigned to points that have significant intensity variation along this profile. The transitional intensity model is initialized for all the radial lines of the first frame. This model is propagated through a time series, and the reference profile is translated and scaled to adjust to new contours at each time frame. Joint-estimation of contours makes this method more robust to low-contrast boundaries. We employ the idea of joint contour estimation, as well as using radial lines in our design.Level-set methods, initially introduced by Osher and Sethian [40], have been utilized in numerous areas of computer vision including medical imaging. Level-sets are implicitly defined contours which evolve in time based on intrinsic geometric measures of the image and some speed function. The curve in 2D is defined as an isocontour of a 3D surface (i.e. a level-set). Due to their dynamic nature, these curves are capable of adapting to different topologies; so connectivity of contour regions is not a limiting factor in curve evolution [41]. Additionally, contour curvature at each point is utilized to impose smoothness constraints on the displacement of each point. These characteristics make level sets a promising approach in contour tracking. Caselles et al. [42] regard the boundary detection problem as finding a curve of minimal weighted length, or a geodesic active contour, in the level set framework. This is an iterative method which converges once the contour reaches a steady state. For high quality images with high-contrast boundaries, this technique yields good results for the LV segmentation problem. However this method is slow with high computation cost, and the convergence criteria may not be met if the boundary is low-contrast or non-existing.One of the shortcomings of deformable models is that they are attracted towards high gradient edges, which causes problems in data frames with a lot of spurious edges. In order to make segmentation more robust to this noisy environment, prior knowledge is used to impose restrictions on the shape of the object. Active Shape Models (ASMs) [43], address the problem of locating objects with known shapes. Hand-annotated segmentation of an aligned set of cardiac data is used as the training set required in this method. Principle Component Analysis (PCA) is applied to represent the training set as a mean shape and common modes of shape variation. A global search is performed on a test image to match the mean shape to image features. This shape is then deformed based on the model's variation modes to better capture local characteristics of the object in the test image. Active Appearance Models (AAMs) [44] follow the same idea, with the exception that appearance characteristics such as intensity values or textures are used to construct the training set. Different research groups [45][46][47][48] have applied either ASMs, AAMs, or a combination of both to the problem at hand. Considering the fact that the curve can only deform according to shape variations in the model, having a comprehensive model is key to the accuracy of these approaches. One of the disadvantages of such models is the manual segmentation required for the development of the training set which, in addition to being subjective, is a cumbersome and labour-intensive task. Additionally, the preprocessing step needed to perform PCA and align the test image to the training set increases the computational time of the algorithm.In order to avoid the overhead introduced by the generation of a training set in model-based methods, probabilistic methods have been proposed to tackle the segmentation problem. Dias et al. [49] derive an estimation for position and thickness of the myocardium in sequences of echocardiographic images. A likelihood function is derived based on radial scan lines as previously seen in [39]. However in this case, the epiand endocardial borders are derived disjointly. Additionally, the assumption is that three distinct regions along each scan line are identified: (a) center point to endocardium, (b) myocardium, (c) outer myocardial boundary to the end of scan line. Maximum a posteriori (MAP) criteria is applied in finding the borders. Although the focus of this work is on echocardiographic images, the algorithm is also applicable to MRI, and we employ some of these ideas in our design. Weng [50] also uses intensity values in learning-based ventricle detection in cardiac MR images. Critical points along each scan line's intensity histogram are identified and used in maximum-likelihood (ML) estimation. In both of the above-mentioned techniques, the assumption is that intensity is roughly constant in each region, which does not necessarily holds true in cardiac data. Recent learning-based approaches like the ones proposed by Sun et al. [51], and Séenéegas et al. [52] are more inclined towards using shape and motion models in a Bayesian framework. Both methods rely on estimations at each frame of a sequence to predict the boundaries in subsequent frames in a sample-based approach; however, they both require constructing an initial training set to be used for the shape model. Therefore, in order to achieve good segmentation results with these techniques, a large number of samples should be used to capture a wide variety of shape deformations, adding to the computational cost of the algorithm.

6 Work and motivations

Real-time image-guided procedures (manual or robot-assisted) are emerging due to potential improvement in patient management and reduction in the overall cost. To extract and incorporate information on-the-fly from intraoperative images requires new ultra-fast processing techniques. In this context, the work presents a method to process real-time MR images using signal intensity projections and extract boundary points on the tissues surfaces. The method was applied for segmenting blood-pool inside the left ventricle of the beating heart by extracting boundary points representing the endocardium wall. The idea is to use the navigator echo principle to extract directly the profiles needed to find the points on the endocardium, without having to go through the actual images. This would incredibly speed up the whole process. Nevertheless, the whole work has been done drawing image profiles on actual RT CMR images, not on projections acquired with the navigator echo technique. Since the topic treated in this work concerns the MRI, the intensity profiles will be often referred as projections, even though they are not extracted directly from the scanner. The work was done by the Medical Robotics Laboratory at the University of Houston where the running project was the design of an MR compatible robot suitable for surgical purposes. The real-time segmentation of the left ventricle's blood-pool was thought to be used as a force generator for the robot end effector. This way the robot would have a virtual corridor to follow while operating in the beating heart (1.6).

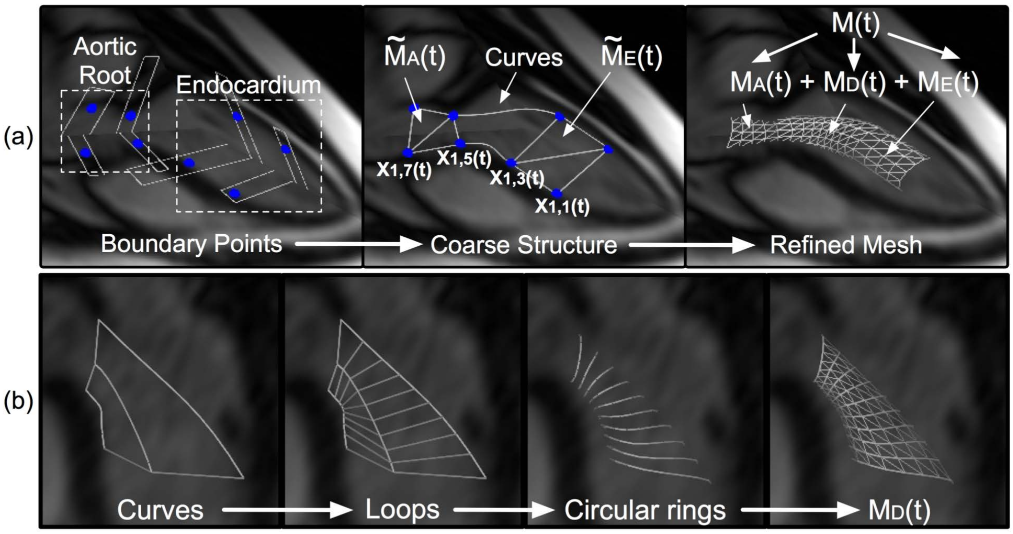


Figure 1.6: Selected points on the endocardium surface, subsequently converted to a coarse structure and eventually to a refined mesh (a). Another way to obtain an access corridor is to pass by loops and circular rings (b) [54].

Though it is very fast and easy to implement, the limit of this approach is that the segmentation's precision strictly depends on how may projections are drawn and on where they are drawn. Moreover, the real-time CMR images are noisy and of low resolution. The challenge is therefore to create a method to obtain a set of 3D points lying on the endocardium walls in real-time through the which it is possible to get a ventricle segmentation. Currently there is no such a thing as a real-time segmentation of the left ventricle on CMR imaging. In order to incorporate more information in the surgical intervention, this work may be innovative and can open a new field of research. Additionally, thanks to the navigator echo technique and to the overall state of the art of all the methods introduced before, the possibility of a real-time heart segmentation becomes real and turns out to be very useful for a lot of applications.

7 Applications

At last, the applications of this method are shown. Real-time image-guided procedures (manual or robot-assisted) are emerging due to potential improvement in patient management and reduction in the overall cost. One of the many applications of a real-time segmentation of the left ventricle is finds place in the field of the visual guidance. During a cardiac intervention the surgeon may have the 3D image of the segmented ventricle either on a screen or projected on the patient (under the hypothesis the scanner is an open one). In addiction to that, it is possible to track any instrument close or inside the heart and visualize it as well. Among the types of image guided cardiac surgery, one in particular has attracted our attention. It is the trans-catheter and trans-apical aortic valve replacement [53]. The type of imaging used in was the fluoroscopy. Fluoroscopy has a very high time resolution but the cardiac images are not very contrasted since X rays are well used for contrasting hard tissues (bones) from the soft ones. The major drawback of this procedure is the amount of ionizing radiations that both the patient and the surgeon have to go through.

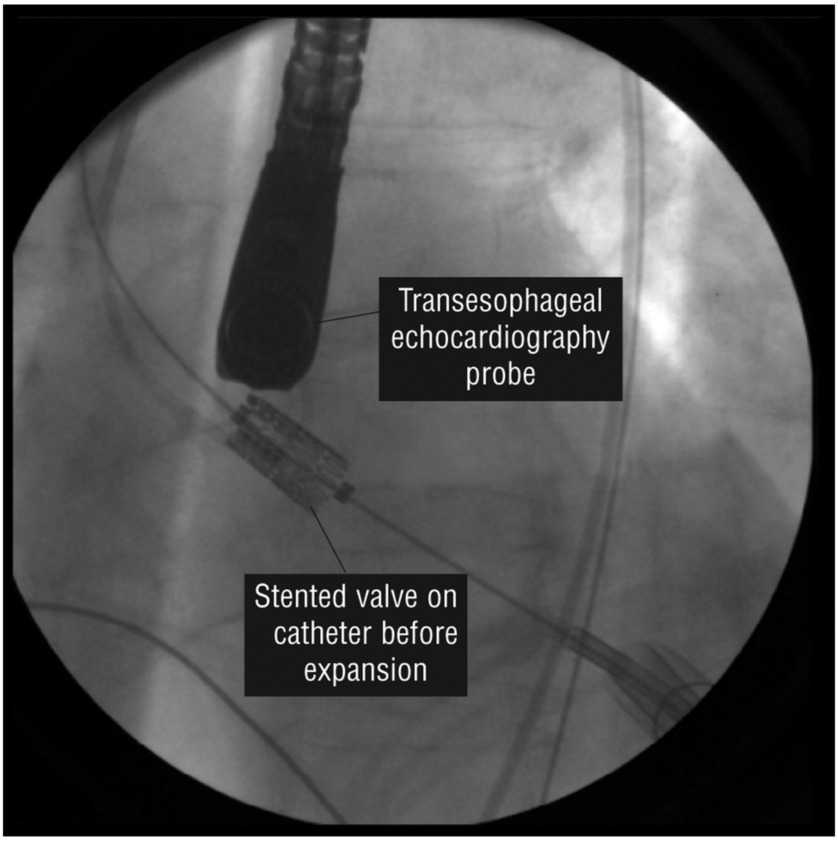


Figure 1.7: Transcatheter and transapical surgery performed with fluoroscopy image guidance.

Another powerful application of this method that somehow relates to the previous one concerns the robot-assisted cardiac surgery and has already been introduced in the previous section. Initially the big drawback of the MRI was the necessity of a complete absence of metal tools in the MR scanner room. This need still exists, while tools made out of MR-compatible materials have already been tested and are in used. MR-compatible robots for this application are a new challenge and they have been designed in the last few years. Not only the real-time segmentation of the left ventricle can show the 3D model of the beating heart, another huge advantage is given to the overall surgery. Being the robot end effector registered with the MR scanner and in turn with the images, a collision between the robot and the heart can be detected in the 3D space. What follows the collision detection is a force feedback to the haptic device of the robot that prevents the endocardium from any kind of damage [54]. Besides the cardiac surgery applications, it can be used for monitoring the heart beat in real-time for whatever clinical aspect such as ventricle functional evaluation [23]. Moreover the mere projections can provide information about the localized heart contraction.

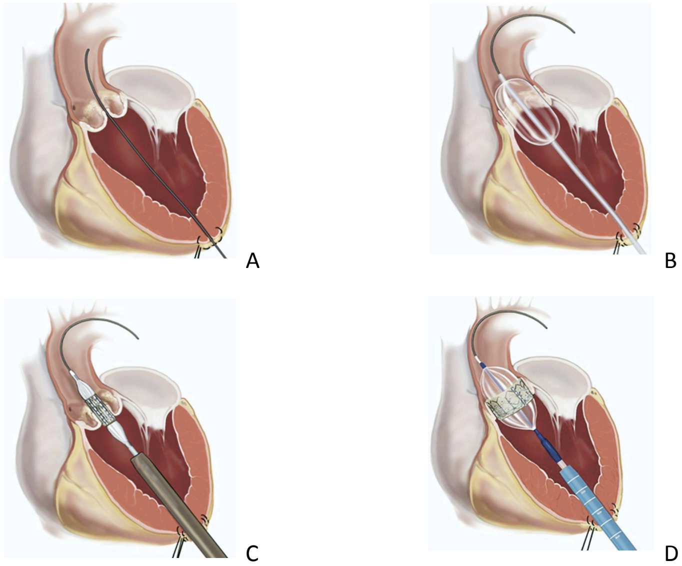


Figure 1.8: All the steps involved in the aortic valve replacement surgery.

Chapter 2

Methods

This chapter illustrates the designed algorithm step by step along with the methods used within it. Afterwards, it deals with important issues regarding the accuracy computation. During the chapter, the reader will run into many basic methods such as filtering, reference frame switching, derivatives computations, etc. Besides those, the other few methods (e.g. walls identification), the whole algorithm structure and its conception for real-time cardiac MRI are novel and designed by the author. The way it deals with the accuracy computation and the adopted figures of merit are designed by the author specially for this thesis purpose. The idea is that, once their utility has been proved in the chapter 3, many of the methods described in this chapter can be used for further improvements, whether they deal with the same topic or not.

1 Algorithm

The aim of this project is to carry out a set of points lying on the endocardium surface. In the images where the aorta is visible a set of points lying on the aorta inner walls will be issued as well. A trivial 2D ventricle reconstruction is performed eventually in order to give a clear visualization of the outcome. The algorithm is semi-automatic. This means that it needs a manual initialization from the user. It can be divided in 6 main parts:

1. Projections drawing
2. Projections filtering
3. Minima searching
4. Walls identification

5. Walls displacement filtering

6. Left ventricle reconstruction

These parts are executed in a sequential way. In order for the algorithm to work properly, it is assumed that:

- The projections are centered with respect to the ventricle (or aorta). The two ventricle walls have to be located one on the left side of the projection and the other one on the right side of it, for each time frame. In order to do this, the surgeon can simply look at one heart cycle before drawing the projections.
- The patient does not move once the projections are drawn.

The algorithm works in two main modalities: on-line and off-line. These differ from each other in how the input is given. In fact:

- In the on-line mode, after the projections drawing, the input consists in a set of N projections belonging to all the views they are drawn from. One projection for time frame is given as input to the second step of the algorithm at time. Therefore each step of the algorithm will be executed as many times as the number of the time frames are, each time processing only one set of projections.
- In the off-line mode, after the projections drawing, the input consists in a set of N sets of projections belonging to all the views they are drawn from. All the time frames of every Nth projection is given as input to the second step. Therefore each step of the algorithm will be executed only once and it will process all the time frames.

The on-line mode is used for the actual simulation and it has to respect the real-time boundaries. The off-line mode is used just to test the algorithm parameters and draw results out of it. Hereafter the online mode will be presented, making a few exceptions every time the two modalities even slightly differ from each other.

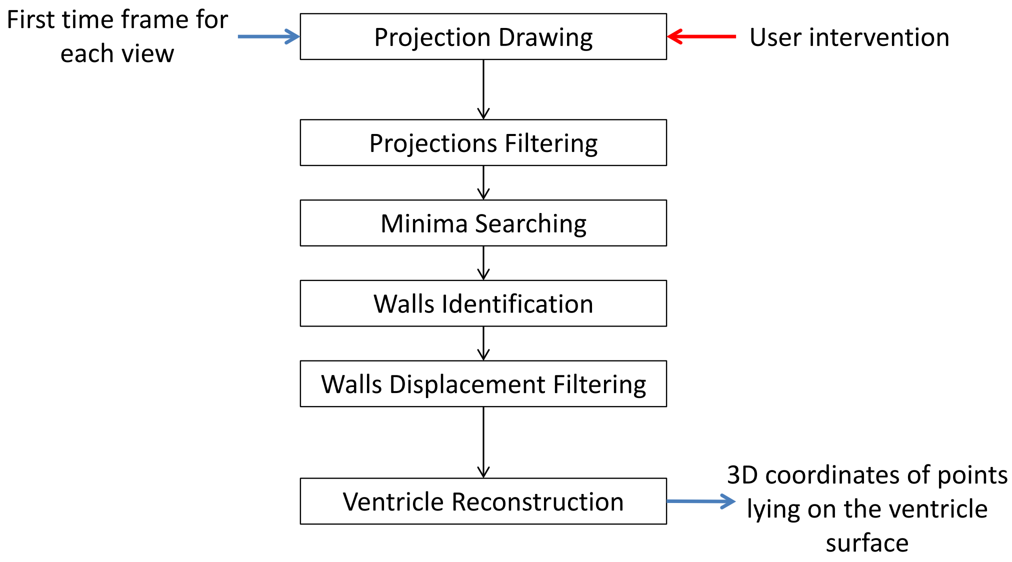


Figure 2.1: Algorithm Scheme.

1.1 Projections drawing

The input to this first part of the algorithm, namely to the whole of it, is a given number of slices from different views (figure 2.2), all of them belonging to the same time frame (figure 2.2). On these slices the user has to draw a number of projections. The number of projections is a user's choice. The more the projections, the more time the algorithm takes to process them and the more accurate the ventricle reconstruction. Before the projections are drawn, the user has to watch at least one heart cycle video, so she/he can meet the first condition mentioned previously. For a rough 2D ventricle reconstruction, the user has to draw the projections in a certain spatial order. The order and the reasons why will be discussed in the last part of the algorithm description (subsection 1.6). If the aim is just finding the points lying on the endocardium surface, the projections can be drawn in any spatial order.

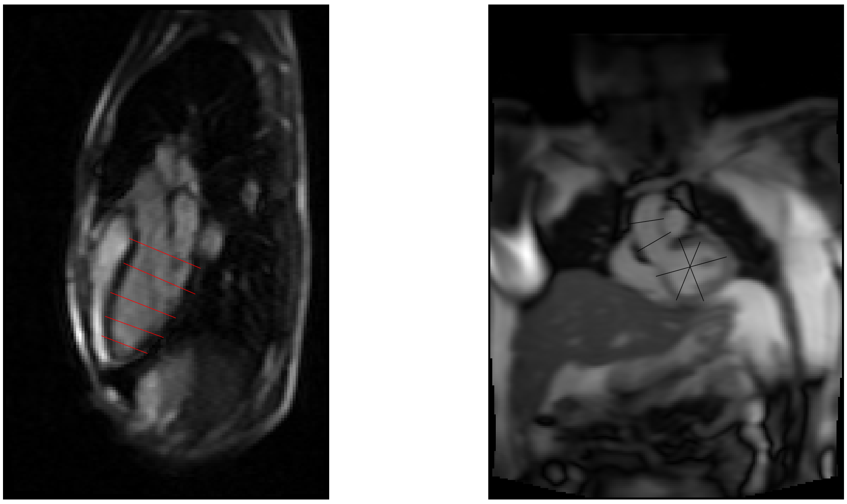


Figure 2.2: Example of five projections drawn on a long axis view (left,red) and on a short axis view (right,black) of the heart.

The projections the user draws are linear. This means that she/he has to define two points in the images that unequivocally identify the beginning and the end of one straight line. Along this straight line and perpendicularly to it, a space sampling is performed. The special frequencies f_l and f_w are chosen by the user, as well as the width of the projection. The sampling is performed considering the closest pixel value in the image to the selected sample that lies either on the straight line drawn by the user (sampling along the length) or on a perpendicular line to it (sampling along the width). The utility of the sampling along the width will be discussed later in this subsection. In order to perform these samplings, a change of frame of reference is needed. In particular, we need to go from the image frame to the projection frame (figure 2.3).

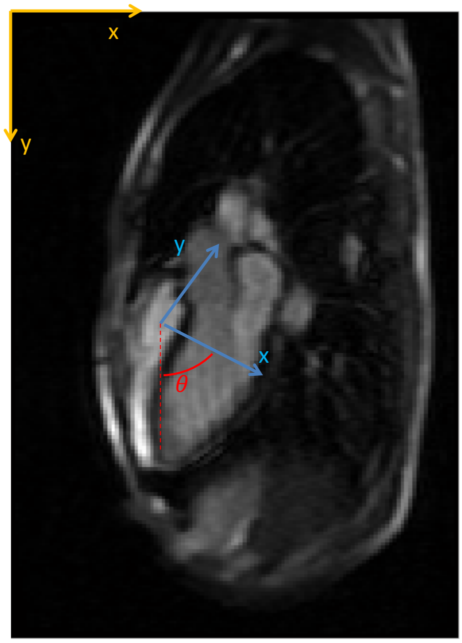


Figure 2.3: The frame of reference is to be changed from the image's one (yellow) to the projection's one (cyan). The angle θ between the two frames is computed as well.

Given the two points in the image frame and the resolution of the image, it is easy to compute the length and the angle of the projection with respect to the image frame. Through the expression 2.1 it is also easy to move back and forth between the two frames (Im stands for image frame and Pr stands for projection frame). θ is the angle between the two frames and it is easily computed through the $\arctan$ operation, by knowing the *x* and *y* components of the projection on the image frame. S_x and S_y are the origin's coordinates of the projection frame in the image frame.

$$\begin{pmatrix} x_{Im} \\ y_{Im} \\ 1 \end{pmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) & S_x \\ sin(\theta) & -cos(\theta) & S_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{pmatrix} x_{Pr} \\ y_{Pr} \\ 1 \end{pmatrix} \tag{2.1}$$

Once the projection is drawn, the matrix is known. The formula 2.1 is used to find every single value of the projection. The first sample of the projection (x_{Pr}=0, y_{Pr}=0) will contain the grey value corresponding to the x_{Im}, y_{Im} coordinates computed through the same formula. All the samples are computed in the same way, just the x_{Pr} and y_{Pr} change. In fact the N-th sample will be computed by putting x_{Pr}=(N-1)*f_l. Since the projection is linear, the projection's X axis frame is chosen to be corresponding to the projection itself. This way, the y_{Pr} value will always be zero. By having a look at the projections drawn this way (figure 2.4), it is noticeable that they show quite sharp variations along their trend.

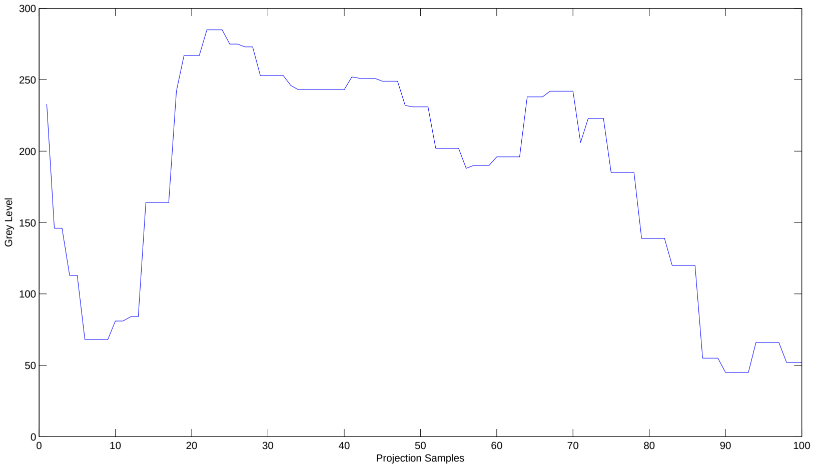


Figure 2.4: Oversampled projection. Sharp variations are appreciable as consecutive constant grey values. This is the roughest projection possible, the only parameter involved is the number of samples (e.g. the sampling frequency).

Furthermore they might show repetitive consecutive values along the projection, making the signal appear like consecutive steps. This is due to the possible space oversampling. The oversampling occurs when the f_l is too high that the interval between two consecutive x_{Pr} is smaller than the resolution with the angle θ. Since the sampling is performed on a squared grid (which is the image) along a straight line with direction given by θ, the resolution along the projection will be depending on the resolution of the grid 0 through the following relationship:

$$\delta(\theta) = \begin{cases} \dfrac{\delta_0}{cos(\theta)}, & \text{if } k\dfrac{\pi}{2} < \theta < (2k+1)\dfrac{\pi}{4} \\ \dfrac{\delta_0}{sin(\theta)}, & \text{if } (2k+1)\dfrac{\pi}{4} < \theta < (k+1)\dfrac{\pi}{2} \end{cases} \tag{2.2}$$

Oversampling is not a problem in terms of information preserving, whereas we have to be sure that:

$$\frac{1}{f_l} \leq \delta(\theta) \tag{2.3}$$

Though it is not a problem, the projections result in a signal full of undesired sharp variations (figure 2.4). Moreover, the occurrence of sudden localized noise might change significantly the shape of it. The need of a first simple smoothing step arises from these reasons. The concept of special averaging is used to perform the smoothing. In fact, two more parameters to be controlled are the width of the projection and the special frequency along the width f_w. Once the user draws a projection, it is like she/he is drawing several parallel projections that will be averaged and will result in one single smoothed projection. For instance, to compute the first sample of the projection, the X_{Pr} stays null and Y_{Pr} varies between $-w/2$ and $w/2$, with steps of $1/f_w$. At every step, the corresponding X_{Im} and Y_{Im} are computed through the formula 2.1 and the grey levels corresponding to those coordinates are extracted. In the end, the first sample of the projection will be the mean value among the value extracted before. This happens for every sample of the projection ($X_{Pr} =$ 0, $1/f_l$, $2/f_l$ . . .). The sampling is done by taking the closest pixel value to the sampled coordinate (figure 2.5).

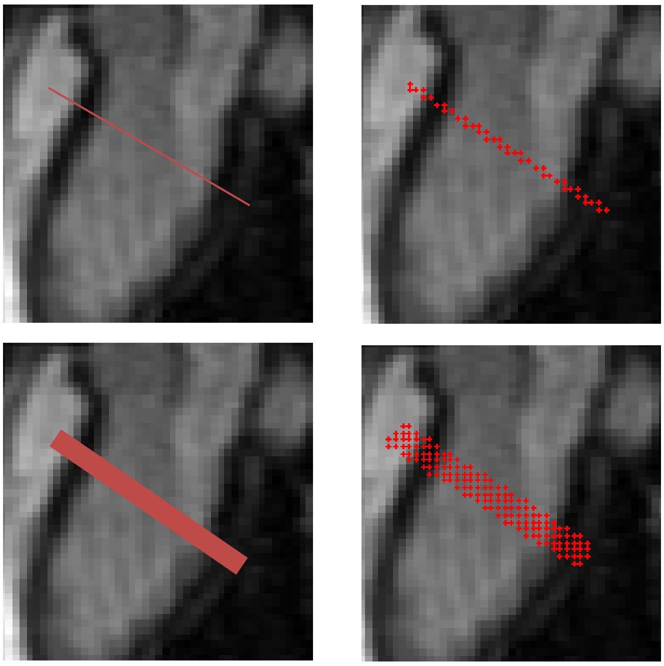


Figure 2.5: The sampling is done by taking the closest pixel value to the sampled coordinate.

The result of this averaging is appreciable on the following images in figure 2.6.

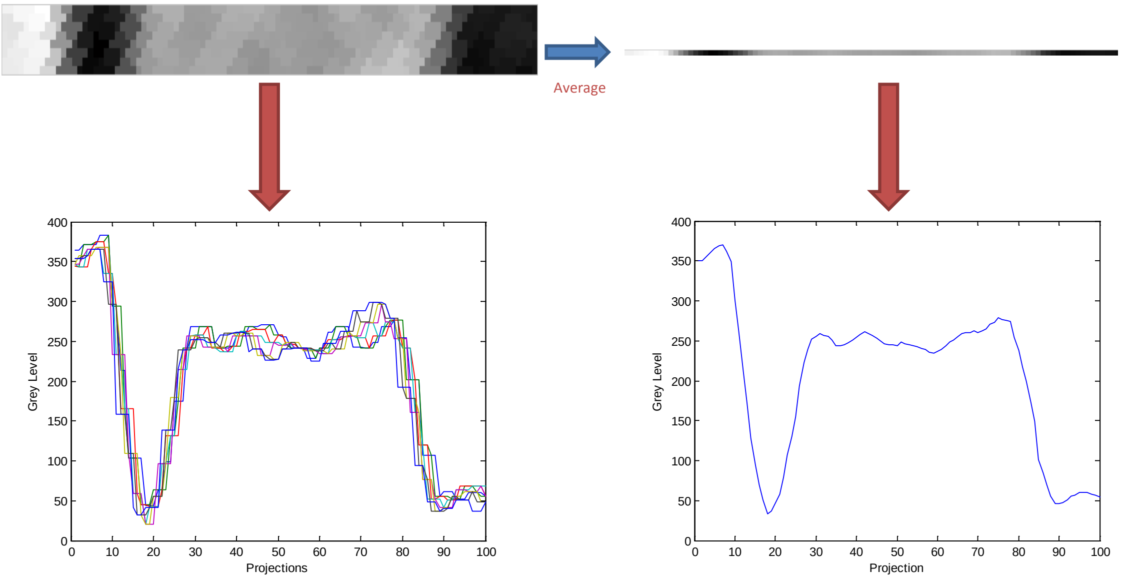


Figure 2.6: Drawing a thick projection corresponds to draw several parallel projections (left). Through the spacial averaging, these projections become one smoothed one (right).

The wider the projection and the higher the special sampling frequency, the more averaging will be performed. Regardless of which the view of the heart is (any acquired slice), the projections have to cross the heart from one side to the opposite. Projections can result in several shapes and trends. This depends in first place on the quality of the image, in last place on the parameters used to draw them. In an ideal world where the blood and the endocardium are perfectly homogeneous and the MR scanner has an ideal impulse response, the projection would result in a constant step from endocardium grey level, to blood grey level, back to endocardium grey level. In this case the ventricle walls would be identified as the first and the last sample of the blood grey level. We have to deal with reality and in reality the shape of the projections are very various. In particular, projections:

- may or may not have two local minima that identify the endocardium core.
- are affected by high and medium frequency noise.
- may or may not show not constant grey level values in the blood zone and/or in the endocardium zone.

- do not show a step from endocardium to blood, but a slope which can be more or less steep.

Besides noise and parameters choice, the shape of the projection strongly depends on what view it is drawn from and where it is drawn in a given view. For instance, projections drawn in certain short axis views show the papillary muscle during the diastole phase (figure 2.8). Projections drawn in certain long axis views, close to the aortic root, show the aortic valve during the diastole phase (figure 2.7). A projection that crosses the papillary muscle will result in an anomalous shape, as well as those which cross the aortic valve. The algorithm has been designed to overcome as much as possible these problems.

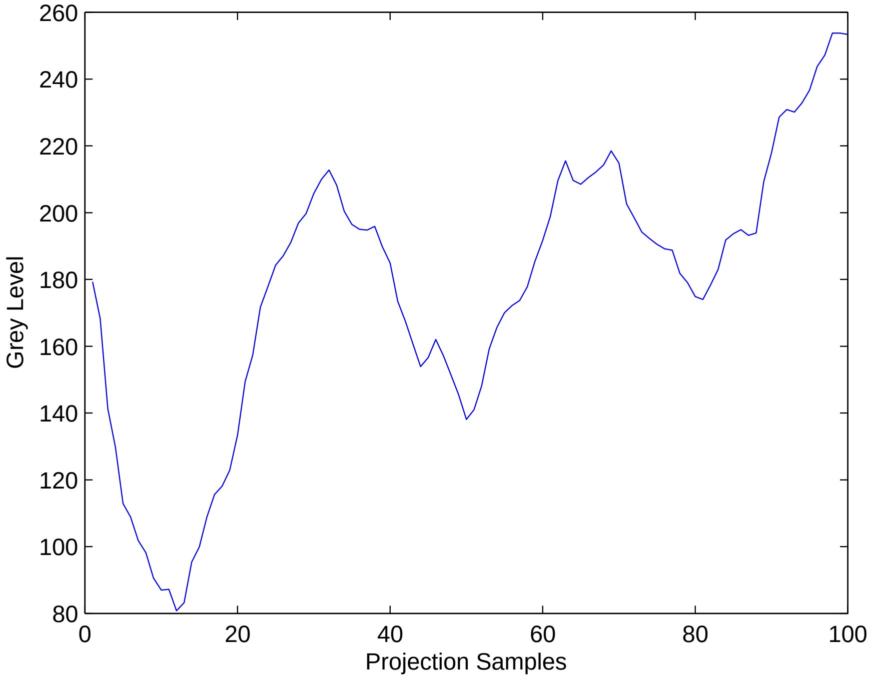


Figure 2.7: Projection drawn across the aortic root on a long axis view. The signal's drop represent the aortic valve, it is visible only during the systole phase.

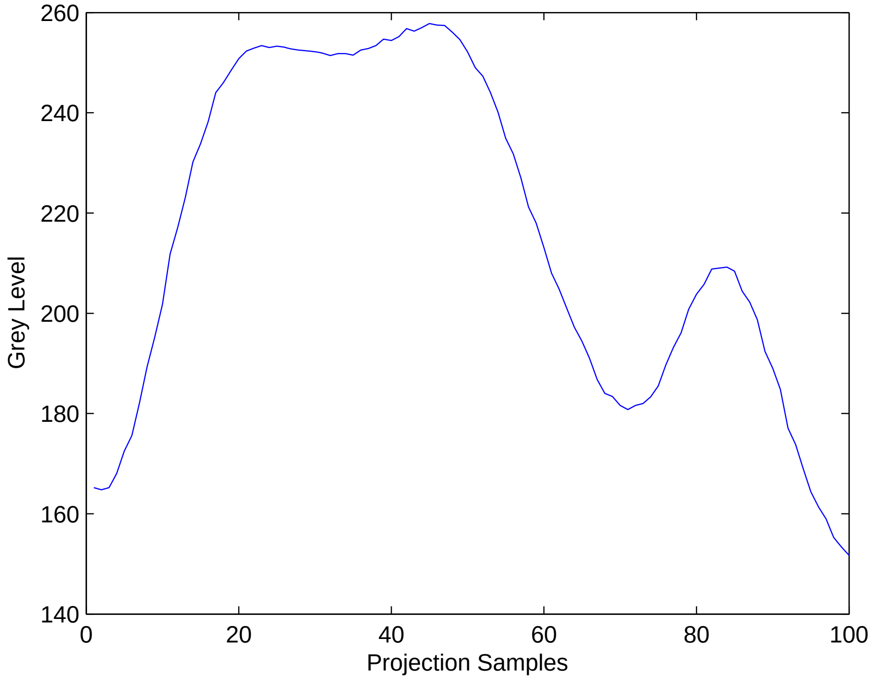


Figure 2.8: Projection drawn across the ventricle on a short axis view. The signal's drop represent the papillary muscle, it is visible only during the diastole phase.

Therefore the identification of the beginning or the end of the endocardium walls in a projection is a problem harder than one would think. This difficulty is due also to the variability among different slices, different patients and along the time in the heart cycle. Since we have assumed that the projections are centered in respect to the heart, we know for sure that there are two points representing the two ventricle walls. In particular, we know these two points are located one within the first half of the projection and one within the second half of it. Therefore we are sure about one thing regarding the projection: it shows high grey levels in proximity of its half and lower grey levels on its sides.

1.2 Projections filtering

The smoothing due to the averaging among parallel projections is useful just to avoid running into anomalous shapes randomly located in the heart, mostly due to noise. Though after the averaging the projection is more smoothed, it is far from showing just two minima where the endocardium is. In fact, a second filtering step is needed. The filtering has to be fast without distorting the main projection's trend. Furthermore, the filtering has to highlight the main trend of the projection, cutting off all the middle frequencies and more. This part of the algorithm takes as input a set of projections from different views but from the same time frame (if on-line, off-line it takes all the time frames). Before explaining all the filtering parameters, it is useful to understand what we actually desired to have after the filtering step. In this way the filter can be properly designed. In fact, what we would need to start searching for the minima (next step) is a smooth signal that shows two minima; whether they are on the very edge or more towards the middle in the projection. This means that we want to extract out of the frequency content of the projection approximately up to the following component in figure 2.9. All the higher components will be more or less smoothly cut off.

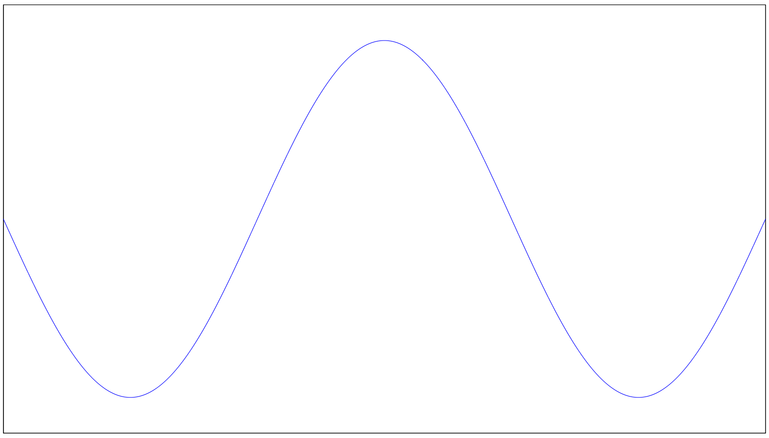


Figure 2.9: Desired projection trend during systole.

This relates to the second condition for which the algorithm works properly. In fact each projection has to show this main trend, since it has to be drawn from one piece of endocardium (first minimum) to another (second minimum), all the way through the blood (middle zone, where the maximum is). So every projection has this frequency component for sure; it is not that the filter is trying to extract a non-existing component out of the noisy projection. Moreover, figure 2.9 shows the highest component the filtered projection should have. During the heart cycle the filtered projection may vary from figure 2.9 to figure 2.10.

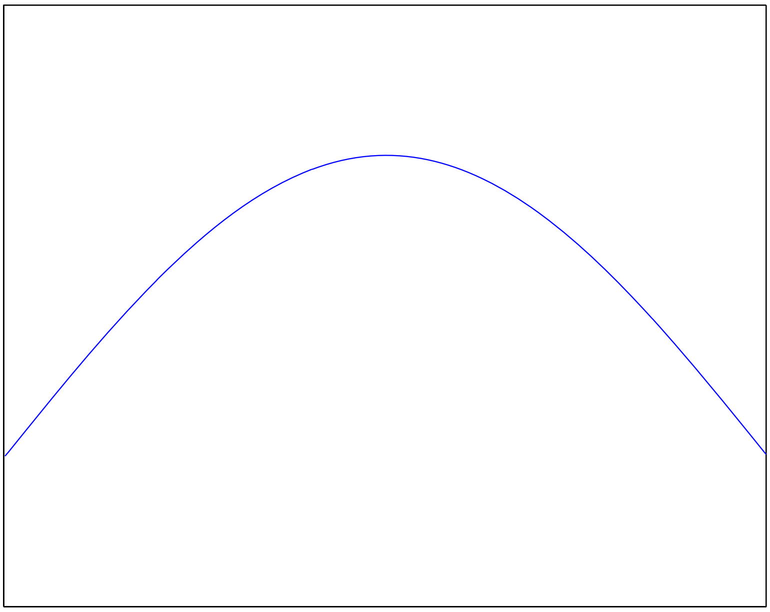


Figure 2.10: Desired projection trend during diastole.

During the systole the heart contracts so the two endocardium walls are closer to each other. During the diastole they are further apart instead. Thus the first graph shows how the filtered projection should be during the systole and the second one during the diastole. So during the systole the signal shows a higher frequency content. For this reason the systole projection represents the worst case and it is taken into account in order to compute the cut-off frequency of the filter. Through this reasoning we can derive what an approximate cut-off frequency for the filter could be, knowing the number of samples N_l. We can easily think in the one-dimensional space domain of the projection. The desired filtered signal will result approximately in 1.5 cycles. Therefore its frequency, and in turn the normalized cut-off frequency of the filter, is easily computed through the following expression:

$$f_c = \frac{2 \cdot 1.5}{N} = \frac{3}{N} \tag{2.4}$$

The cut-off frequency corresponds to the point where the frequency response of the filter hits -3 Db. Once it is defined, the filter has to be designed. Since our priority is to preserve the overall shape of the projection, a linear phase filter is needed, particularly a FIR filter. FIR filters with symmetric coefficients meet this prerequisite, they are always stable and they are very easy to implement; the only drawback is that they have a wide attenuation band and they need a high number of coefficients. In a FIR filter with a fairly low order, the attenuation band is so wide that some frequency components higher than the cut-off frequency might show up after the filtering. Since the filtering has to be quite hard, the cut-off frequency can actually be lower than the one mentioned above.

$$f_c \leq \frac{3}{N} \tag{2.5}$$

Whether the projection is drawn on the apical part of the heart or in correspondence of the aortic root or in the center of the ventricle, it has to show the same trend. This is possible thanks to the length-depending cut-off frequency. Two types of FIR filters have been designed for this purpose:

- The moving average FIR digital filter.
- The classical frequency-windowed FIR digital filter.

Being M the number of coefficients, it is important to keep in mind that for any FIR filter the following relation is valid:

$$FilterOrder = M - 1 \tag{2.6}$$

Moreover, an M coefficients FIR filtering step implies a delay of M/2 samples. This means that the filtering will actually take effect after M/2 samples. For stationary signals, one of the most used expedients is to pad the beginning of the signal with M/2 samples. There are a lot of ways to pad the signal properly in order to spoil as little as possible the first M/2 samples of the output. Another smoothing method was taken into account that is the smoothing through spline approximation. Therefore, the only parameter the filtering part requires is just the cut-off frequency f_c that is in turn function of another parameter (the length of the projection) or the smoothing factor if a smoothing spline is chosen.

Moving average FIR filter

The MA filter is the simplest FIR filter designable. It has only one parameter to be defined, namely its order. In the context of moving average filters, the concept of cut-off frequency is not really taken into account since it is a type of filter designed in the original domain (e.g. space or frequency). Nevertheless, for high filter orders (i.e. greater than 10), expression 2.7 is valid.

$$f_c = 0.886 \cdot \frac{1}{M} \tag{2.7}$$

Since the parameter we want to control in the filtering step is the cut-off frequency, the number of coefficients of the filter is easily computed through the following expression:

$$M = round(\frac{1}{M}) \tag{2.8}$$

In the original domain the moving average filter shows a set of M points all of the same value. In order to normalize its frequency response, the M points have the value of 1/M. Hereafter two examples of moving average filters are shown in their frequency domain.

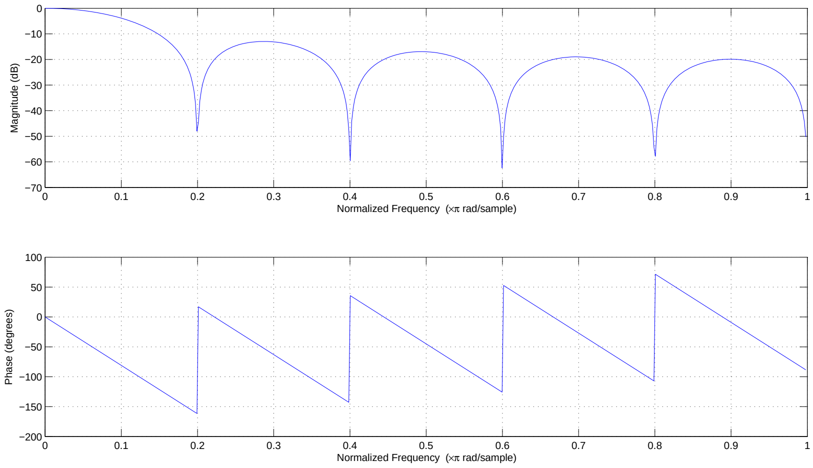


Figure 2.11: Frequency response of a MA filter with 10 coefficients.

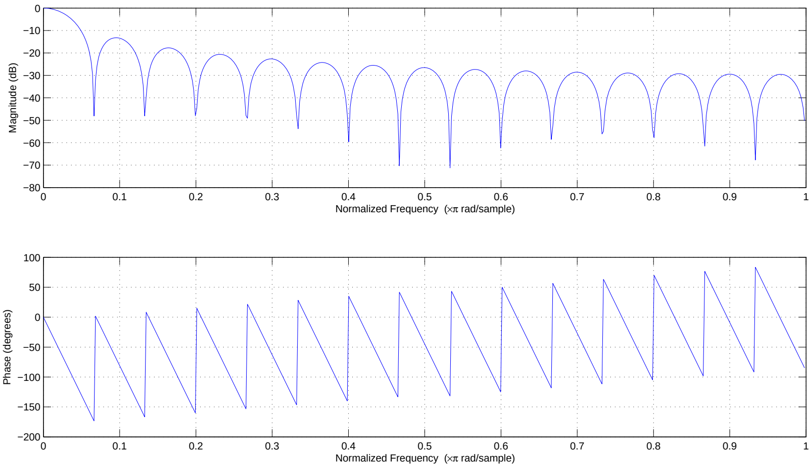


Figure 2.12: Frequency response of a MA filter with 30 coefficients.

Moving average filters need a lot of coefficients to succeed in their task and this slows down the whole filtering. Moreover their frequency response shows quite high ripples after the cut-off frequency. Strong components at the ripple frequencies might show up in the filtered projection.

Classical linear phase FIR filter

The second type of filter is designed using the classical method of FIR filter design [55]. In first place, the filter is designed in the frequency domain as an ideal filter with a given cut-off frequency f_c. The inverse transform of the low pass filter is the ideal impulse response of the filter itself in the original domain $h(n)$. $w(n)$ denotes a chosen window. The windowed digital filter coefficients are given by the following expression, where 1 $\leq$ n$\leq N_f$:

$$b(n) = h(n) \cdot w(n) \tag{2.9}$$

Particularly the used window $w(n)$ is the Hamming window (figure 2.13). The windowing allows us to pass from an ideal filter to a real one and the sampling makes it digital. The wider the window the closer to the ideal filter the digital filter is. The tighter the sampling (i.e. the filter order N_f), the higher the frequency resolution of the filter. On the other hand the filter delay is to be taken into account. In order to filter them properly, the padding is needed. In particular a specular padding is performed in order to alter as little as possible the projection's shape.

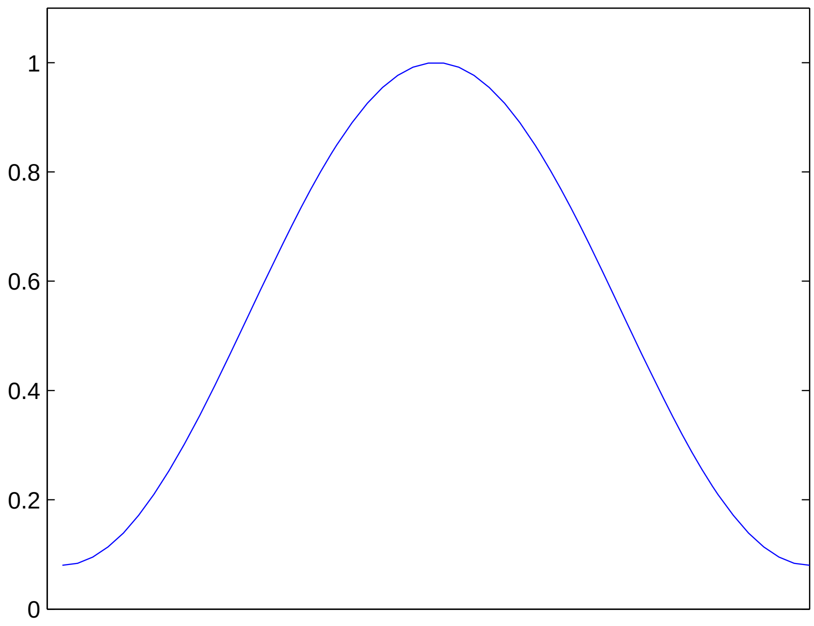


Figure 2.13: Hamming Window.

In this case two parameters are to be controlled: the cut-off frequency and the filter order. It is true that the lower the cut-off frequency is, the higher the resolution has to be and therefore the higher the filter order. By controlling the filter order it is possible to overcome the moving average drawbacks. A lower order filter speeds up the process and this method does not show high ripples after the cut-off frequency in the frequency response. The problem is that the attenuation band in this case is larger than the other one, namely the frequency response is less steep after the cut-off frequency.

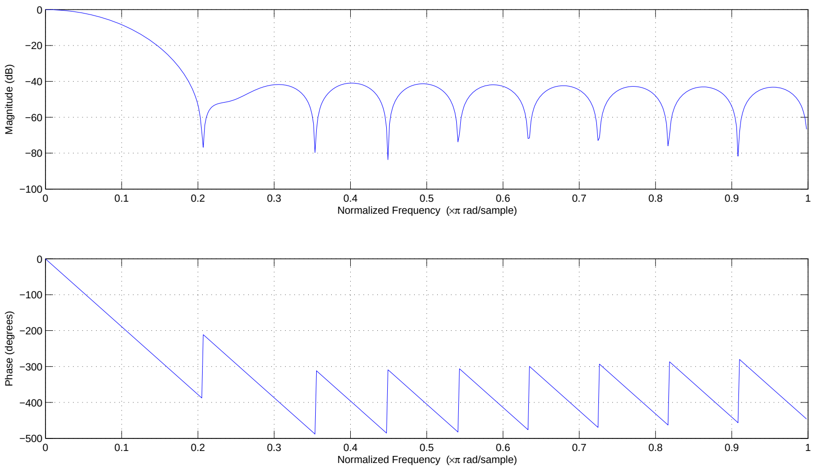


Figure 2.14: Frequency response of a FIR filter with cutoff frequency 0.01 and 20 coefficients.

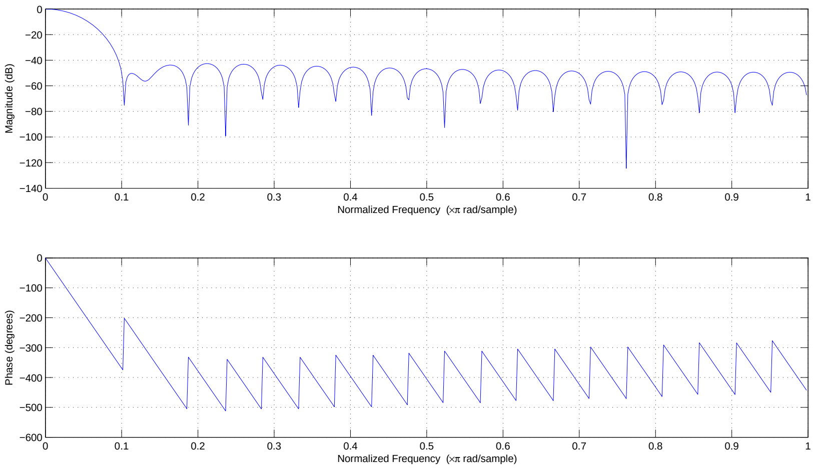


Figure 2.15: Frequency response of a FIR filter with cutoff frequency 0.01 and 40 coefficients.

Smoothing spline

Another type of filter which is not related with digital FIR filters is designed. This method uses the filtering proprieties of the De Boor's cubic spline smoothing parameter p. This parameter determines the relative weight you would like to place on the contradictory demands of having the projection to be smooth against having it to be close to the data. For p=0, the resulting smoothed projection is the least-squares straight line fit to the data, while, at the other extreme (i.e. for p= 1), f is the natural cubic interpolating spline. As p moves from 0 to 1, the smoothing spline changes from one extreme to the other.

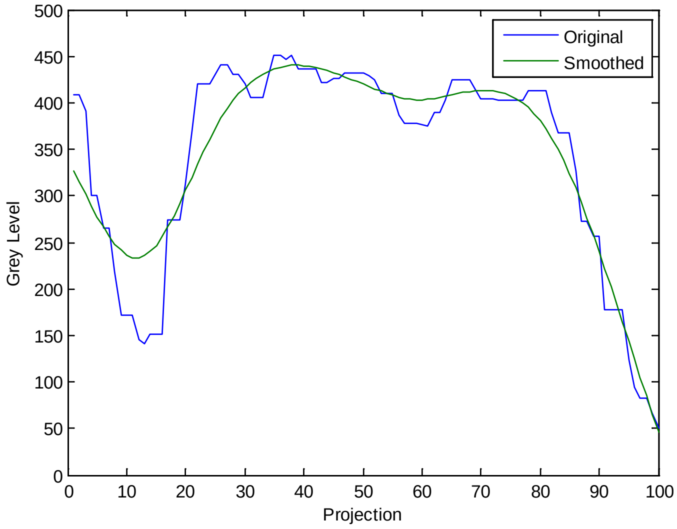


Figure 2.16: Projection smoothed with De Boor's parameter equal to 0.001.

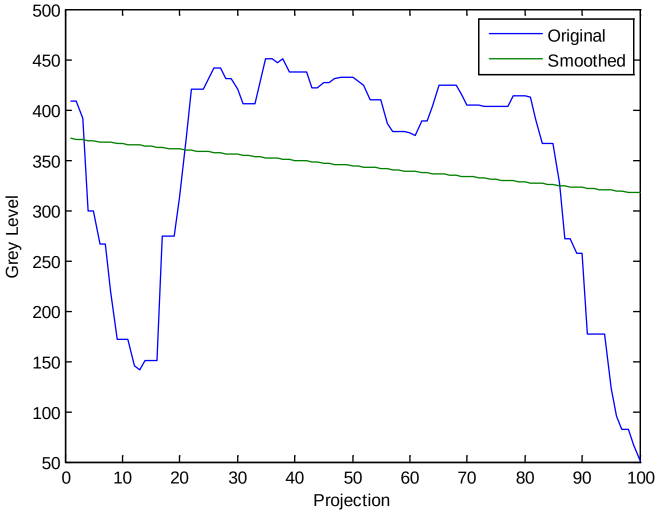


Figure 2.17: Projection smoothed with De Boor's parameter equal to 0. In this case it corresponds to the least-squares straight line fit to the data.

As we can appreciate from the images above, the smoothing performed through this method is quite soft, even though the parameter tends to zero. It is interesting using this kind of smoothing because it does not have anything to do with cut-off frequencies or transfer functions. It is just a numeric approximation of the signal [56].

1.3 Minima searching

This part of the algorithm takes as input the filtered projections, regardless of how they have been filtered. Since this step is completely independent on the previous one, it is important how the input might show itself. In fact, depending on whether the filtering has been hard or soft, the projections will show more or less than two minima. The aim of this section is to give as output two indexes where the two minima are found. The two minima in fact will represent the endocardium. We can divide the types of input (i.e. filtered projections) in two main types:

- Hard filtered.
- Soft filtered.

The minima searching will always be performed in the same way and will issue the same kind of output. In order to point out the minima in the projection, the first thing to do is to compute the first and second derivative. This computation is not always easy. In fact a wide branch of numerical analysis is dedicated to best approximate the derivatives of a given function. In this case we are not interested in the actual shape of the derivative but simply in its change of the sign. This allows us to choose a very easy and fast way to compute them, namely the forward difference method. This method simply operates the difference between two consecutive samples. It can also be seen as a digital filter whose impulse in the z domain is given by the following expression:

$$TF(z) = z - 1 \tag{2.10}$$

We disregard the frequency response of this filter because we do not care about the actual shape of the output. We just know that whenever a minimum occurs in the projection, the derivative changes sign (figure 2.18). The same computation is used on the derived signal, to compute the second derivative of the projection (figure 2.19). Analytically, given the first two derivatives, we can state that a minimum occurs whenever the logic AND between these two conditions is verified:

- The first derivative changes sign.

- The second derivative is positive.

Since we cannot process the digital signal in an analytically way, the first sample that changes sign on the first derivative will correspond to a minimum, only if the same sample in the second derivative is positive. Again we could use numerical analysis methods to exactly find the zero crossing point in the first derivative but it would be useless in terms of efficiency because it will bring a very negligible improvement. Two more cases lead to a minimum identification: when the first sample of the first derivative is positive or when the last sample of the first derivative is negative. This is easy to understand as if the projection starts with a positive slope or ends with a negative one, respectively the first or the last sample can be identified as minima.

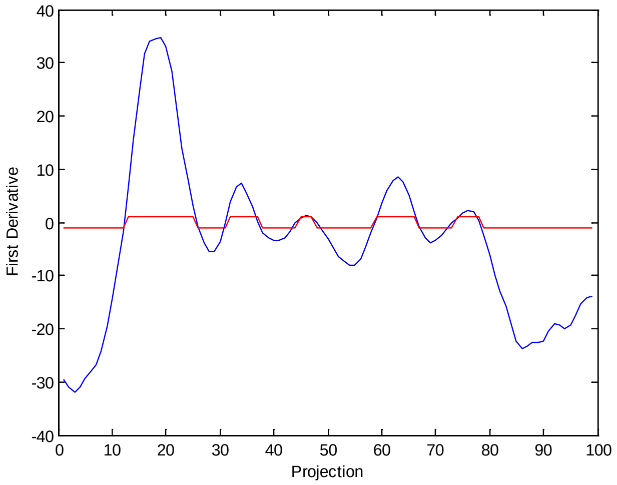


Figure 2.18: First derivative (blue), sign of the first derivative (red). In fact, we are only interested in the sign.

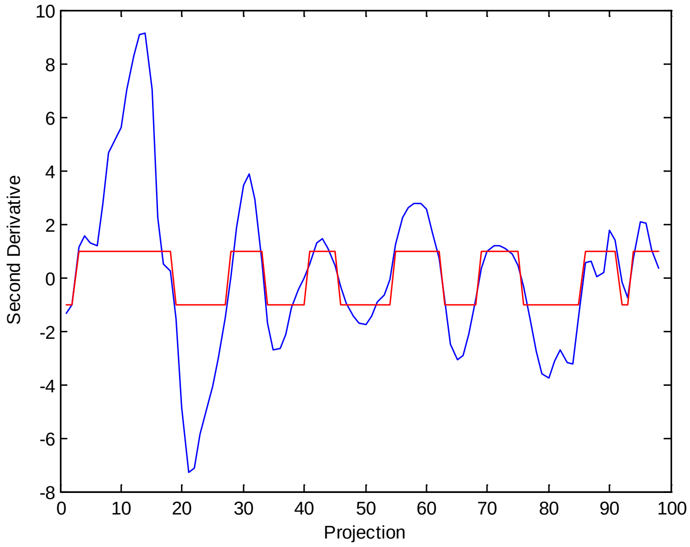


Figure 2.19: Second derivative (blue), sign of the second derivative (red). In fact, we are only interested in the sign.

What it is going to happen applying this trivial operator to the filtered projections will depend on the kind of filtering the projection has undergone in the previous step. If an aggressive filtering had been performed, the projection will show only two minima, most likely corresponding to the edge samples of the projection. If a weak filtering was performed, the projection might show more than two minima. Some minima due to noise might show up between the two endocardium minima. In order to select only the two minima we are interested in a trivial expedient is applied. First the projection is divided in two halves. For each half, only the minimum corresponding to the lowest grey level in the original projection is kept, the others are thrown away. In fact it is true that the two minima we want to search correspond globally to low grey levels. Splitting the signal in two halves just prevents the algorithm to accidentally find two minima on the same side, due perhaps to a drop of the projection on one side close to the actual minimum (figure 2.20).

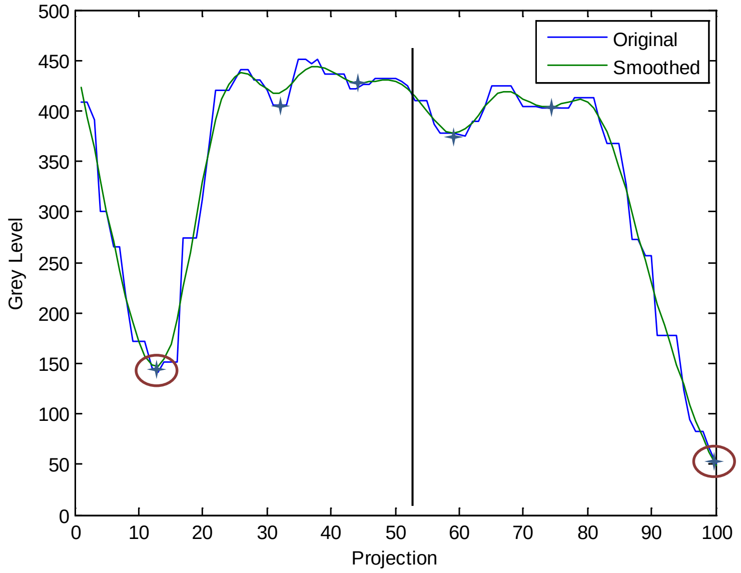


Figure 2.20: Projection smoothed with the spline method. Blue stars represent all the minima, circled stars are the minima corresponding to the lowest grey level in each half-projection.

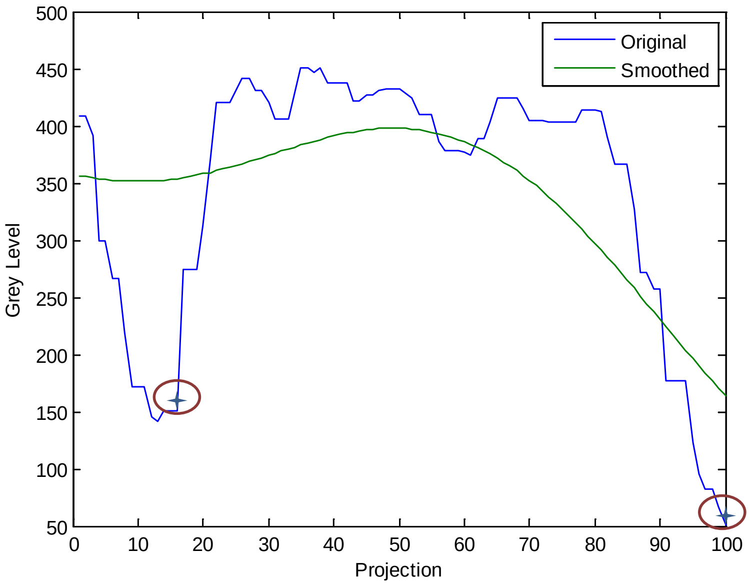


Figure 2.21: Projection smoothed with the classical FIR design method. In this case only two minima are found because of the hard filtering applied.

A combination of a strange shaped projection and hard filtering on it can lead to a very anomalous filtered signal. In particular this signal might show only one minimum in one of the halves. Therefore, if no minimum is found in either one of the halves, the algorithm has to predict where the minimum occurs. The occurrence of this error is very low (it might actually never happen) and when it occurs it affects only one or two time frames every heart cycle. Moreover, it is known that the wall displacement cannot vary too much from one frame to the following one (at a fairly high sampling frequency). For these reasons a very trivial predictor is designed. When no minimum occurs, the predicted minimum corresponds to the minimum in the previous time frame. If the absence of a minimum occurs in the very first time frame, the minimum is assigned to one of the marginal samples. This particular case never occurs, it is just a way to initialize the value. It is important to understand that after the derivatives computations, the filtered projection exhausts its task in the whole algorithm. In fact, we keep track only of the indexes (at which the minima correspond), not of the actual values of the samples. Also in choosing the two minima corresponding to the two lowest grey levels, these values are computed on the original projection, not on the smoothed one.

1.4 Walls identification

Being this section of the algorithm one of the last ones, some rough results will already be issued at the end of it. Therefore before going through it, some details of the last two sections are to be clarified; in particular some errors they intrinsically introduce. By having a look at the original projections, at the smoothed ones and at where the minima are found; it is noticeable that the minima are found in a different position from where they actually are in the original projection. This is because the filtering step, by cutting out some components, introduces a small distortion in the overall shape (figure 2.21). At first this might seem a relevant error whereas it is not. Actually the algorithm could even terminate at this point, if the aim were to find the two minima in a projection. It would issue the two indexes where the minima occur, with a margin of error due to the filtering. We cannot forget that what the whole algorithm does is processing an image. So we have to recall what the two minima in the projection represent in the images. In fact, they correspond to the darkest points in the endocardium. They represent the endocardium core, not the wall. Being the aim of the entire algorithm to find the two points representing the walls, at this step something else must be done. The path between the minima and the walls of the endocardium is nothing but a more or less steep slope towards the middle of the projection. The previous steps of the algorithm gave us just the initial points where to start looking for the walls from. In fact, hereafter we will stop caring about what is beyond the two minima. All the samples that go from the beginning of the signal to the first minimum and from the second minimum to the end of it will no longer take part of any processing. Now, finding the two samples that correspond to the walls is a non-deterministic and ambiguous problem. Even by looking at the images it is hard to tell where exactly the walls are (problem treated in section 2.2). For this reason we can only appeal to what we know for sure, namely that the walls are located in the interval that goes from the minimum to the center of the projection, for each half-projection. Thinking in an image-way, the slope goes from the endocardium core to the blood. This section of the algorithm works on the original projection and simply moves the points from the minima given as input, towards the middle of the projection, until a given grey value is reached. We call this value "stopping value". Three methods to compute the stopping value were designed:

- Constant.
- Mean.
- X-range.

The constant and the X-range methods take as input one parameter, the mean method does not.

Constant

The first method is the simplest one. It takes as input a constant value that will directly correspond to the stopping value. This method to a first approximation corresponds to the trivial segmentation of the image through thresholding (figure 2.22). The only difference between the mere thresholding and this method is the filtering. In fact if any event occurred causing a sudden drop of the signal (e.g. strong noise, presence of papillary muscle or aortic valve, figure 2.8), the sole thresholding would fail. On the one hand, the choice of a constant (i.e. of a threshold) is very simple, fast and intuitive. On the other hand it is not always reliable. The biggest disadvantage in choosing this method is that it is image dependent and thus needs a priori knowledge on the image. Depending on the sequence used by the MR scanner to acquire the images, the grey levels of the image itself will vary. Another a priori piece of knowledge that has to be taken into account is the bit depth of the image. For these reasons this method seems not so suitable, though it has been tested anyway.

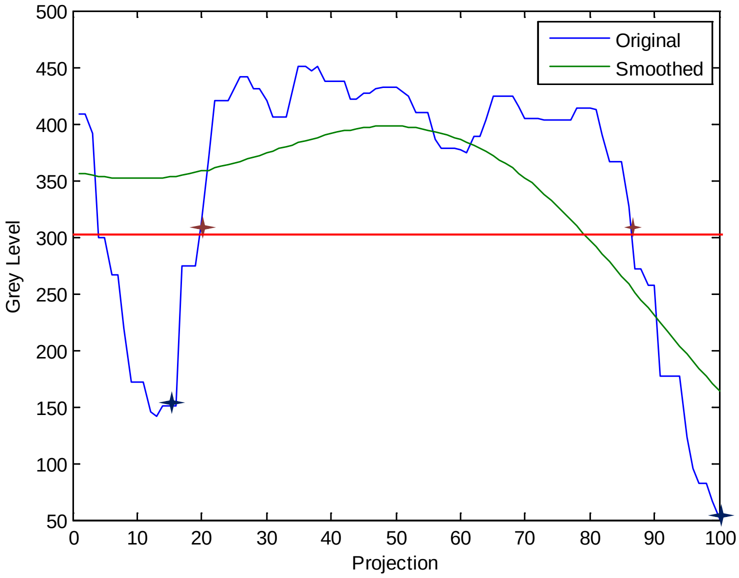


Figure 2.22: The blue stars represent the two minima found before, the constant is represented with the red line (in this case it is 300), the red stars represent the two points that identify the endocardium surfaces.

Mean

According to this method, the stopping value is computed through the mean among the samples between the two minima (figure 2.23). There is another sub-method that first divides the projection in two halves and then computes the mean among the samples between one of the minima and the middle of the projection. This way there will be two different stopping values for the two sides of the projection that identify the walls (figure 2.24). This is just one more option to control the efficacy of the algorithm. Two very important advantages of this method are its independence from the grey levels values and its absence of any parameter as input. This method turned out to be very projection's shape dependent. In particular it is slope-dependent. Let us say that we have a very steep slope that goes from the minima to the high values in the middle of the projection. This implies that, within the interval between the minima, there will be a lot of high values and very a few low values. The mean of this distribution will result in a very high value, close to the values of the blood. On the contrary if the slope is very light, the stopping value will result close to the minima. This might not be fit for the purpose of the algorithm, though it has been tested anyway.

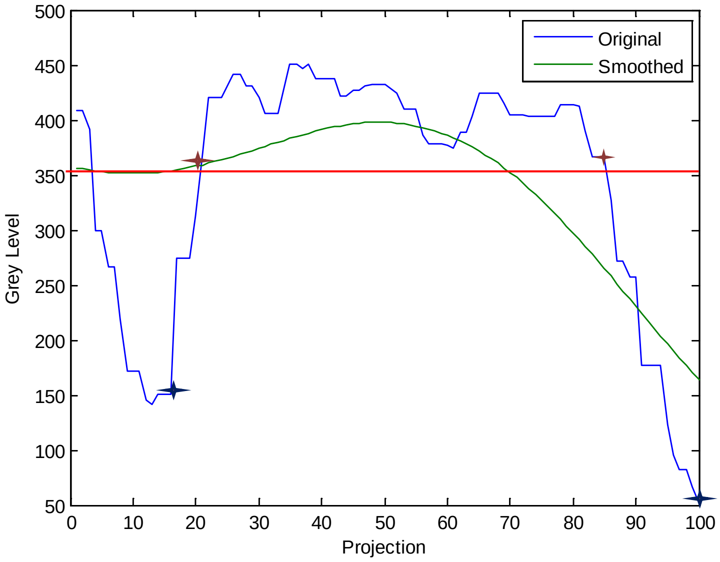


Figure 2.23: The blue stars represent the two minima found before, the mean value is represented with the red line (in this case it is 355), the red stars represent the two points that identify the endocardium surfaces.

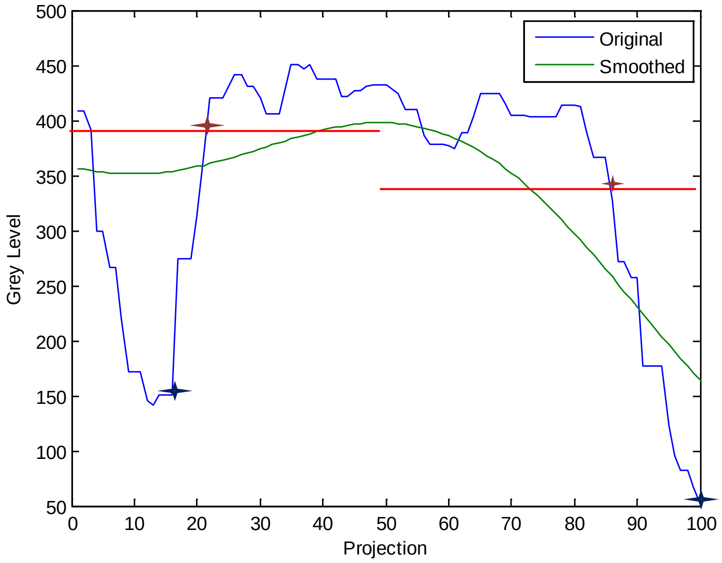


Figure 2.24: The blue stars represent the two minima found before, the red stars represent the two points that identify the endocardium surfaces. In this case two mean values are shown for each half-projection.

X-range

The last method takes as input a ratio between 0 and 1. The stopping value is computed through the product between the input and the difference between the lowest grey value (the lowest one between the ones corresponding to the minima) and the maximum in the projection, computed on the interval between the two minima (figure 2.25). If the input is 0.5, this value corresponds to the mid-range of the distribution of the grey values between the two minima (for this reason the method is called X-range). A higher input will simply move the stopping point more toward the middle of the projection. A null input will let the minima and the stopping value to coincide, namely it will not move the two points. Also for this method a variant that divides the projection in two halves was designed. Instead of picking the maximum value among all the samples between the two minima, the maximum value is selected between one minima and the middle of the projection. Therefore, though the method has only one input, the stopping value will be different for each half-projection (figure 2.26). This method is sensitive only to the difference between minimum and maximum in the projection. It does not depend either on the projection's shape, or on the bit depth or generally on its grey values. It has one parameter to be controlled but it is not as hard to control as it is in the threshold method. In fact the parameter can vary only between 0 and 1 and in reality it will vary between 0.2 and 0.8 because the extreme values are useless for our purpose. These reasons make this method the most trusty amongst the three presented.

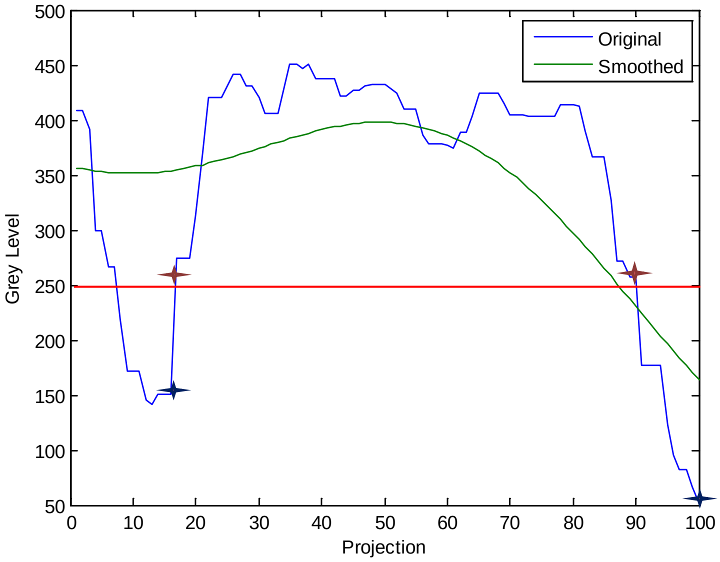


Figure 2.25: The blue stars represent the two minima found before, the X-range value is represented with the red line (in this case it is 250), the red stars represent the two points that identify the endocardium surfaces. The value of X in this case is 0.4.

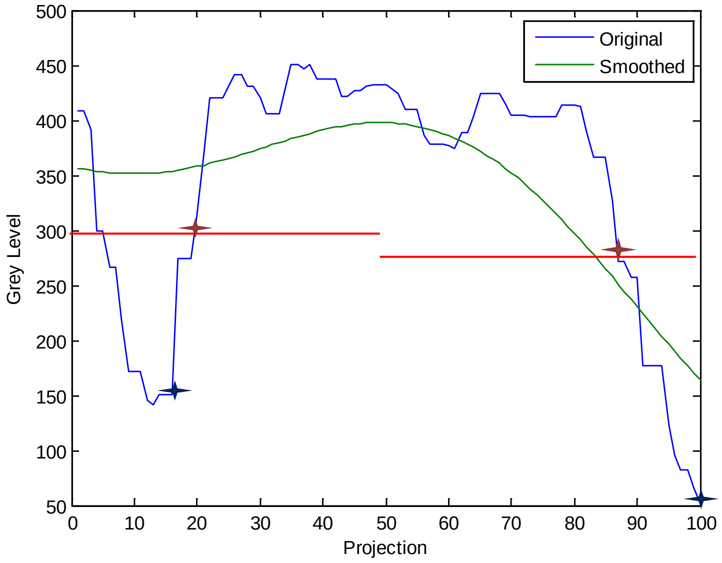


Figure 2.26: The blue stars represent the two minima found before, the red stars represent the two points that identify the endocardium surfaces. In this case two X-range values are shown for each half-projection. The value of X in this case is 0.4.

Through one of the three methods presented above, this section of the algorithm can output the indexes where the endocardium surfaces occur in the projection. This is the first significant outcome since the beginning steps of the algorithm.

1.5 Walls displacement filtering

Depending on what mode the algorithm is running on, the input to this section is either nothing but a set of two-time signals for each projection or two numbers representing the two indexes the endocardium surfaces occur at for each projection. In the offline mode we can appreciate the whole time series of how the two walls move. During the online series, at a current time T we can see just how the walls have moved until that time. So both if the algorithm is running offline or if online mode after a couple of heart cycles, we can clearly see the heart phases like systole and diastole. In particular, when the two points get closer to each other the maximum heart contraction is detected (systole), whereas when they get further apart from each other it is a clear sign of heart relaxation (diastole). A lot of information can actually be drawn out of these signals but we will go over it later. This section of the algorithm is made just in order to correct previous errors that might have occurred in the earlier steps. Errors can be due to either bad setting of the parameters or to any algorithm flaw. The algorithm treats the projections for every time frame as isolated units. At time T, it does not have saved any information about the previous projections (i.e. the previous walls indexes). We know that the beat of a healthy heart is a smooth movement. In spite of this, the first rough walls displacement signals might look a bit jerky. This is due to little imprecisions of the algorithm. To overcome this defect, a trivial but very useful trick was adopted. The signal representing the wall index displacement along the time is low pass filtered. This way we are actually saving some information from the past projections and using it to modify a possible error in the current projection. The aim is to smooth a possible jerky movement, namely to cut off only the highest frequencies in the signal. For this reason a very light filtering is performed (figure 2.27). In fact, a simple moving average filter with three coefficients was chosen.

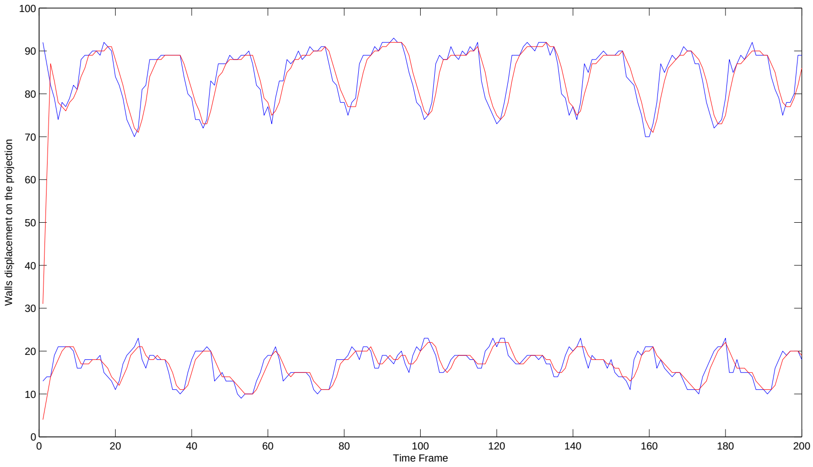


Figure 2.27: The time progression of the two walls lying on a projection, rough (blue) and filtered (red).

Thanks to this filter, the output results smoother and more realistic. This way we get rid of all the small imperfections that might have occurred during the previous steps. It is also true that this filtering introduces a delay of 2 samples. This is not a problem; simply the first two samples of the program execution will be meaningless. The output of this section has the same meaning of the output of the previous one, it has just been refined.

1.6 Left ventricle reconstruction

From the second section of the algorithm to the previous one we have dealt with image intensity projections. This last section brings the whole algorithm back to the image domain, where it was actually created. Actually what is to do now is very similar to what was done at the beginning. What this step has as input is a set of two indexes for every projection that was drawn in the first step. There is a straight correspondence between the indexes belonging to a projection and the x-y coordinates in the image where the walls are actually identified. This relation is given by the same matrix used in the first step (formula 2.1). By back projecting the walls indexes into the original image reference frame, the ventricle can easily be reconstructed. Reconstruction algorithms have been mentioned in the first chapter [22] and are not part of this study. Nevertheless, in order to have a rough idea on how the reconstructed ventricle may look like, a trivial 2D reconstruction method has been implemented. It actually just connects two consecutive points with a straight line (figures 2.28 and 2.29).

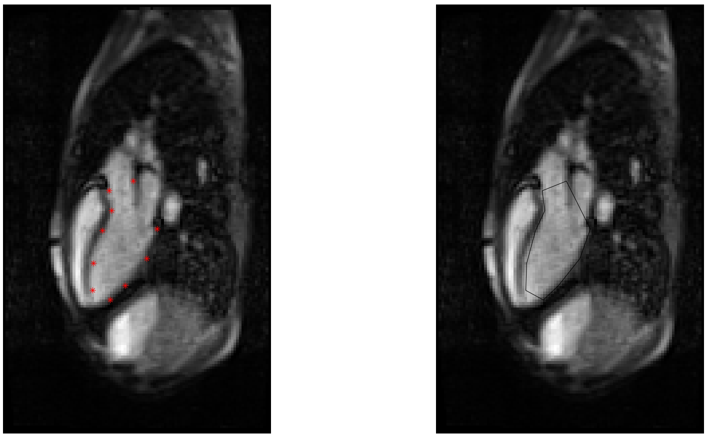


Figure 2.28: Points lying on the ventricle surface (left, red). Reconstructed ventricle with 5 projections (right,black).

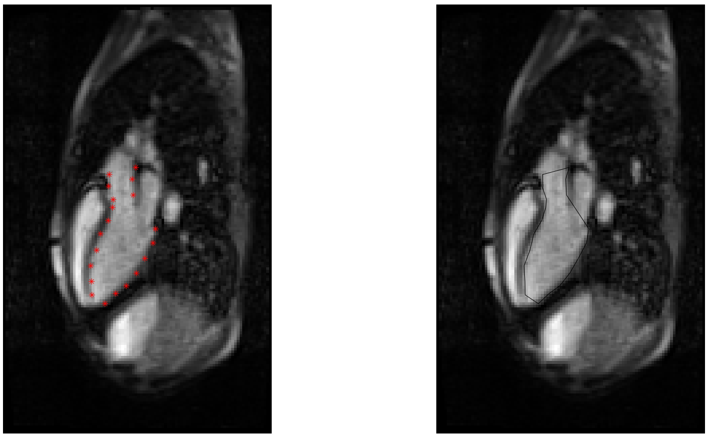


Figure 2.29: Points lying on the ventricle surface (left, red). Reconstructed ventricle with 10 projections (right,black).

The drawback of this method is not that it does not reconstruct the ventricle accurately. Since the future purpose of the algorithm is to stay within the real-time boundaries, if any fancy reconstruction takes precious time to the overall algorithm, they are not going to be performed in favor of a rough reconstruction. Moreover, the number of the projections will strongly influence the final outcome in terms of accuracy on the image. The main drawback of this reconstruction method is that the projections have to be drawn in a particular spatial order and with the same direction. The order can be either from the aorta towards the ventricle or the other way around. The direction must be the same as the first drawn projection's one, for example: from left to right (in the long axis view) or from down to up (in the short axis view). If the user meets these initial requirements, the ventricle is reconstructed properly.

2 Accuracy

This section regards the way the results have been collected, the form they show themselves and the dataset used. Therefore it will be divided in the following subsections:

1. Types of images.
2. Ground truth.
3. Accuracy computation.
4. Figures of merit.

Unlike in the previous section, these subsections are time independent from each other. This order was chosen for a better understanding of the problems.

2.1 Types of images

The cardiac MR images can be divided in views depending on where is locate the slice of acquisition in respect to the heart. Fundamentally there are two views, showed in 2.30. The views are:

- Long Axis View (LAV).
- Short Axis View (SAV).

If we simplify the left ventricle as a cylinder, a long axis view will be any slice parallel to the length of the cylinder whereas a short axis view will be any slice transversal to the cylinder. In the LAV the ventricle will have a stretched shape while in the SAV it will appear as a circle. The LAV will show two endocardium walls (left and right) and the aortic root as well as the aortic arch. Some particular LAV might show the papillary muscle as well. The SAV will show the endocardium walls all around the section and the papillary muscle in the middle of it. Some SAVs show the papillary muscle only in the diastolic phase and might show the aorta as well.

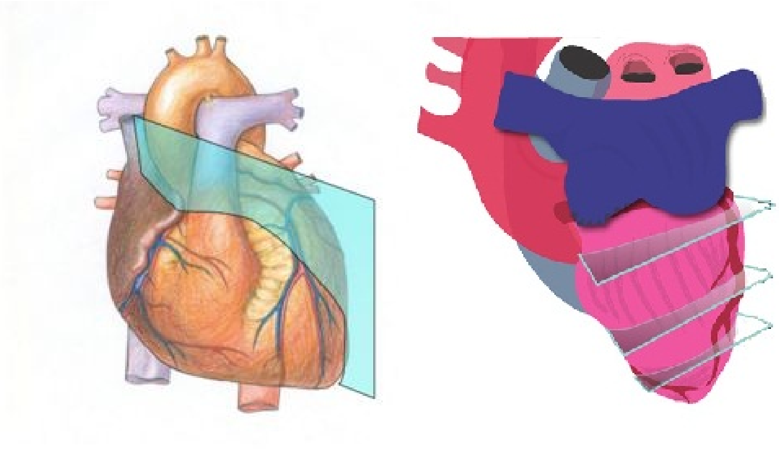


Figure 2.30: Plane where a long axis view lies (left). Plane where a short axis view lies (right).

The SAV reduces the dimension of the ventricle as closer as it is acquired towards the apical part of the heart. The LAV lies on the middle of the left ventricle so it is visible at its largest size.

2.2 Ground truth

The first issue regarding the ground truth is what the ground truth actually is. In order to test the algorithm we have to compare its output with something that represents what the algorithm tries to do. Since its output is basically a set of two indexes in a one-dimensional signal, the ground truth has to have the same form. By just looking at the image projections, one cannot say where exactly the walls are so a ground truth cannot be drawn in this way. In fact, its computation has to start from the images by segmenting them. As we said before, real-time MR images are of a fairly bad quality (figure 1.2). It has been tried to automatically segment the images through a split-merge algorithm, already successfully tested with CINE images in previous studies. Though it worked well with Cine images, it did not work with real-time ones. For this reason the only way to segment real-time images and be sure they are as much accurately segmented as possible is to manually do it. Hence, all the dataset has been segmented manually. Every time a projection is drawn from point p_1 and point p_2, the two points are projected on the segmented image and a projection is automatically drawn on the ground truth as well.

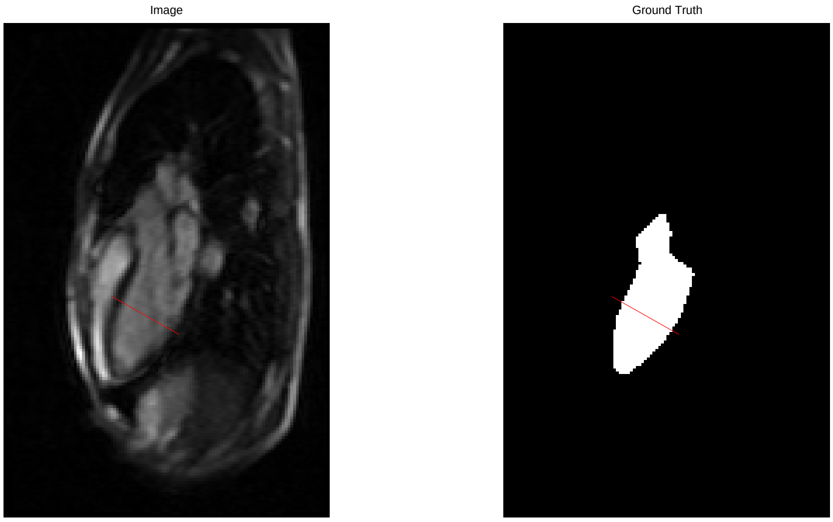


Figure 2.31: Projection drawn on an image (left), same projection drawn on the corresponding segmented image (right).

Although the ground truth was drawn in the image domain, it is meant to be transposed on the projection domain. In fact, it is possible to associate a segmented projection to every projection. This is computed by just extracting the binary values on the segmented image in parallel with the grey levels on the original one. Every time a projection is draw from point p_1 and point p_2, the two points are projected on the segmented image and a projection is automatically drawn on the ground truth. Therefore, the segmented projection will contain ones where it crosses the segmented ventricle, zeros elsewhere (figure 2.31). Yet the ground truth does not have the desired form though. In fact, we want it to be a set of two points per each time frame and each projection representing the ground truth walls displacement, like the output of the algorithm. In order to obtain this, the segmented projection is processed in the following way. Only the first and last samples equals to one are kept. All the other samples of the segmented part are forced down to zero. This way the segmented projection has the desired form (figure 2.32)and can be easily compared to the algorithm output.

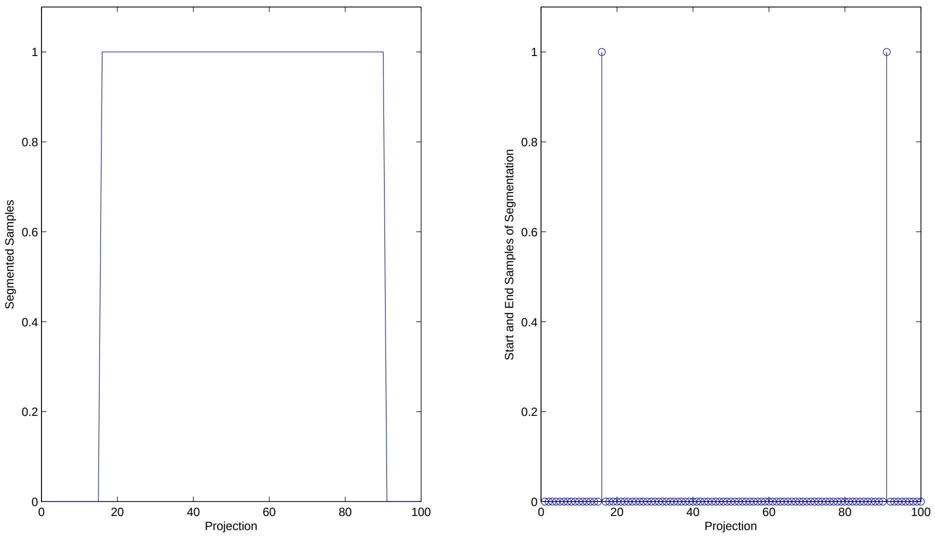


Figure 2.32: Resulting projection out of the segmented image (left), actual ground truth (right).

In presence of the papillary muscle (i.e. all the SAVs) the segmentation becomes a problem. We have to find a point lying on the endocardium wall but it happens that the papillary muscle is extremely close to it, sometimes it even results attached. For this reason it is segmented, just considering it attached to the closest endocardium wall (figure 2.33). This introduces one restriction to the ground truth extraction method. The projection has to cross only one segmented part of the image.

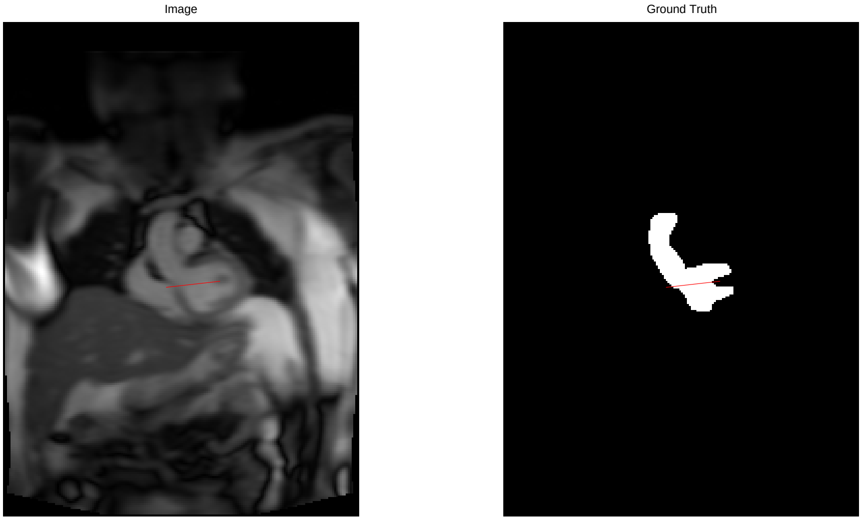


Figure 2.33: Projection drawn on an image (left), same projection drawn on the corresponding segmented image (right).

The last expedient adopted to draw a correct ground truth is the same as adopted in the subsection 1.5. In fact, manually segmenting the images is not easy. One can easily run into small errors due to the poor resolution of the images (figure 2.34). These small errors are reflected on the segmented walls displacement signals as sharp variations. Knowing that the real walls cannot have that kind of sharp variations, the same filtering introduced in the subsection 1.5 is adopted. This way we get rid of small errors the user who segmented the images might have accidentally done. A small segmentation error can still occur, that is a small offset in the segmentation. We do not care about this small error because, if it occurs, it is very small. This has to be taken into account in the results analysis though.

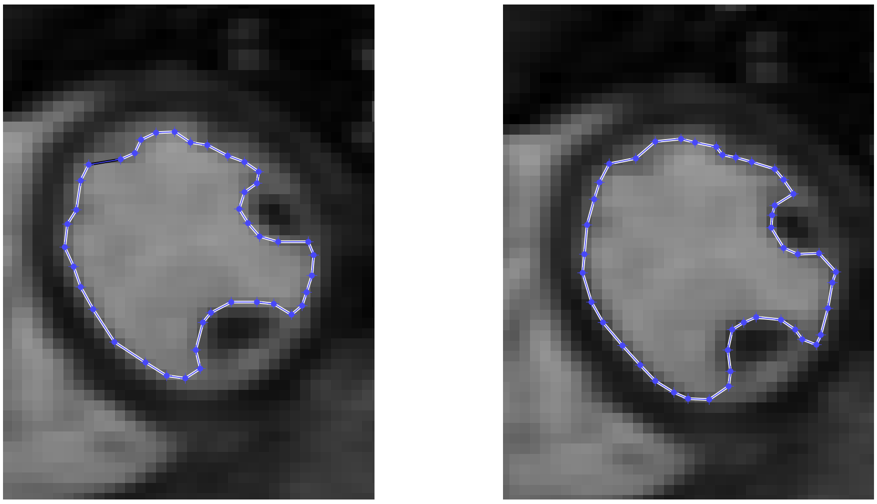


Figure 2.34: Segmentation applied on the same image. It is noticeable that the segmentation is performed in two different slightly ways, none of them is objectively right or wrong. It just depends on whether we want the segmentation closer to the walls or a bit further. Moreover, both figures show how the papillary muscle is segmented.

2.3 Accuracy computation

Once ground truth (GT) and algorithm's output are in the same form, the accuracy computation might seem trivial. Accuracy is in general defined as the difference between the measured value and the ground truth, in absolute value (formula 2.11).

$$ACC = |MeasuredValue - GroundTruthValue| \tag{2.11}$$

The algorithm and the ground truth do not consist of a single value though. In fact, they consist of N_{pr} sets of two-time signals, where N_{pr} is the number of projections. So to select one single ground truth value out of a generic simulation, three variables are to be fixed. One of them fixes the set of two signals (i.e. a chosen projection), one fixes which one of the two signals (left or right value on the projection) and, once we have one only signal left, one last variable has to pick out a single value out of the time sequence. These three variables are embedded with the algorithm simulation and with the images; they are not modifiable by the user. They are just variables the ground truth is a function of.

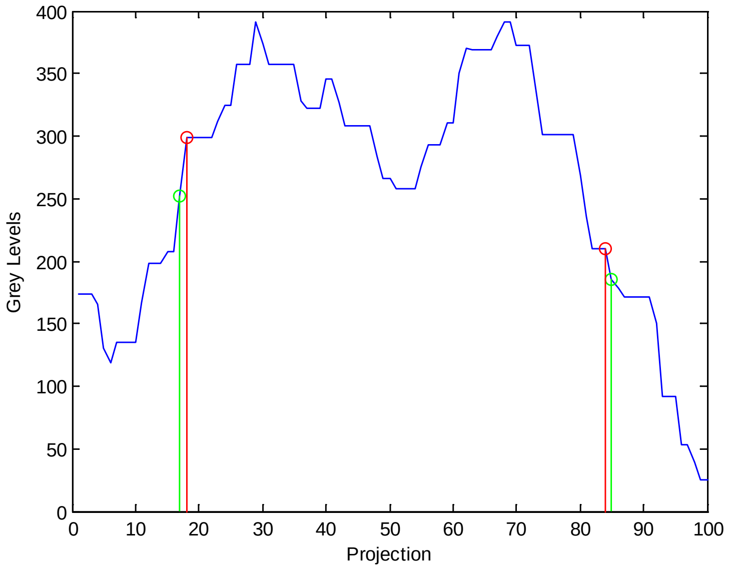


Figure 2.35: Projection drawn on an image (left), same projection drawn on the corresponding segmented image (right).

Of course the algorithm output (OP) is also function of these three variables, but it is also function of a set of parameters the user can choose and modify (e.g. cutoff frequencies, spatial frequencies etc.). Hence, we can define the accuracy with the following expression:

$$ACC(t, pr, s, P) = |MeasuredValue(t, pr, s, P) - GroundTruthValue(t, pr, s)| \tag{2.12}$$

where:

- t represents the time frame.
- pr represents where the projection is drawn.
- s represents the side (left or right).
- P is the set of parameters of the algorithm.

The first distinction is done between qualitative and quantitative variables. In fact, there is no need to discretize a qualitative variable. They are defined by the simulation's intrinsic features. Quantitative parameters are defined by numbers and they can be continuous or discrete. Their domains strictly depend on the nature of the variable. The t variable is quantitative and discrete; the s variable is qualitative and discrete. In particular it can assume only two values: left (L) or right (R). It is useful to distinguish between the first wall index and the second one in a projection, namely between the two signals in the algorithm output. Whereas there is no problem in describing the time frames and the side variables as they are discrete by nature, some issues arise from the pr variable representation and the set of parameters P. The parameters that P includes are the same introduced in the section 1, namely:

- Number of samples along the length.
- Width.
- Number of samples along the width..
- Smoothing mode.
- Cut-off frequency.
- Stopping value method.

In turn, the last parameter (Stopping value method) is function of other parameters:

- Stopping value method.
  - Mode.
    - Variant.
  - Parameter.

In first place, we can distinguish between quantitative and qualitative parameters just as we did for the variables previously. Qualitative parameters are those that are defined by a method or a state configuration (e.g. Smoothing mode=MA or Stopping value mode=Constant). They imply different pieces of code to be executed every time they are changed. Therefore there is no need to discretize a qualitative parameter. They are defined by the algorithm's configuration. Quantitative parameters are defined by numbers. Their domains strictly depend on the nature of the parameter. For example, the cut-off frequency parameter is defined between 0 and 1. The discretization of these parameters will be dealt in the next chapter. Regarding the pr parameter, it is difficult to represent where the projection is drawn as a variable. Theoretically it is basically described by two points, namely four values (two X and two Y). We somehow want to discretize these variables and find what their domain is in order to make the accuracy computation feasible. Using the four values as variables to represent the pr variable would lead to a very complex domain for these values because it will depend on where the ventricle is located on every time frame. It would also lead to an extension from one to four variables to represent one single concept. Furthermore, the variation of the accuracy with respect to the point coordinates of the projection is a very intricate issue. We do not expect an appreciable variation between two projections drawn in the middle of the ventricle (LAV) very close to each other. For these reasons, representing the pr parameter by a pair of coordinates is useless. A study of sensitivity has been performed to see where the accuracy of the algorithm changes significantly. For each piece of the dataset the accuracy was computed, considering several time frames, both the sides left and right and many configurations of parameters. It was proven that the accuracy changes significantly among two to three main locations, depending on the view:

- Long Axis View.
    1. Across the apical zone.
    2. Across the middle zone.
    3. Across the aortic zone.
- Short Axis View.
    1. Across the ventricle without papillary muscle.
    2. Across the ventricle with papillary muscle.
    3. Across the aorta (if visible).

Within the zones the accuracy does not change more than the ± 0.5 mm. This will be probably more clearly explained on the subsection 2.4 This way the pr variable, from being a quantitative continuous (described by the two points coordinates), turned out to be a qualitative variable, described by zones. In conclusion, given a set of parameters as input, the quality of the output may be influenced by the position of the projections, by the fact the output regards one side of the ventricle or the other one, or by the time frames the algorithm is processing. This last issue regarding the time frames is better translatable, instead of a mere time frame issue, in a heart cycle one. The accuracy of the algorithm is a function of three variables and many parameters. For this reason the accuracy is not expressed in a meaningful form. It contains all the information we need, we have just to extract this information and pose it in a good and significant shape. Since our knowledge about where or with what configuration of parameters the algorithm might work better does not have any fundamentals; a figure of merit was designed. In particular, we want to vary in the parameters space P and draw conclusions on what are the optimal parameters the algorithm works better with. Some parameters could even turn out to be non-influential on the algorithm. Thanks to the minimization of the figure of merit, function of P, with respect to the parameters themselves; we both get the minimum value of accuracy possible and the optimal parameters.

2.4 Figures of merit

As the aim is to have a single value that represents the overall quality of the algorithm in terms of accuracy, we have to apply some kind of computation to the equation 2.12 because it is function of too many parameters and variables. The main problem still regards the variables t, pr, and s because they are not treatable as the parameters. In fact, while we can modify the parameters until we find a good result, the variables are intrinsic in the algorithm's concept. Now, we want to better define what is a good result in this case. Fixing a set of parameters $\hat{P}$, the error value (i.e. the accuracy) might stay very low along the time and for both sides only for certain projections, for instance the ones in the middle zone. For projections drawn in different zones it might show different time trends, higher than the previous one in average.

- Accuracy on the projections location.
- Accuracy on the heart phase.
- Accuracy on the endocardium side.

In order to have this, we need to have a set of fixed parameters $\hat{P}$ as well. We want these parameters to be optimal, namely to minimize the figure of merit. The figure of merit will be:

$$ACC(pr, P) = \frac{1}{2 \cdot N_t} \sum_{t=1}^{N_t} \sum_{s=L}^{R} ACC(pr, t, s, P) \tag{2.13}$$

We preserve the information about one variable by averaging the other two. Now, in order to find the best accuracy with the optimal parameters P_o, we need to minimize the figure of merit with respect to the parameters set P.

$$P_o(pr) = \underset{P}{\operatorname{argmin}}\{ACC(pr, P)\} \tag{2.14}$$

$$ACC(pr) = min\{ACC(pr, P)\} = ACC(pr, P_o) \tag{2.15}$$

This way we can evaluate the quality of the algorithm in terms of accuracy regarding how the For example, let us consider an LAV. We draw 2 projections on the apical zone, 4 projections on the middle zone and 3 projections on the aortic root zone. We start the simulation and we collect the accuracy results in the form of $ACC(t, pr, s, P)$. Now we want to know how it has been accurate in the three zones ones. We simply average the accuracy on the t and s variables in order to get it in the form of $ACC(pr, P)$. Now we minimize with respect to P to obtain P_o and the minimum accuracy corresponding to the optimal parameters. This leads to $ACC(pr)$. We know how each projection has been accurate in average. Last step is average the value corresponding to the same zone. In fact, we will notice that all the projections drawn in the same zone have more or less the same accuracy. The sensitivity study mentioned in the subsection 2.3 was performed exactly in this way. The fact that we distinguish in three main zones is a consequence deriving from the fact that within these three zones the accuracy doesn't change significantly. It is clearer for us to have an accuracy function of as few variables as possible.

Chapter 3

Results

The methods explained in the previous chapter have been implemented and tested on a dataset. This chapter shows what the dataset consists of and its acquisition features; what kind of discretization was applied to the continuous parameters and finally the accuracy value of the algorithm. Moreover, a GUI was designed in order for the user to draw the projections, set the parameters and see the algorithm running. The whole implementation, results drawing, and GUI were done in MATLAB. Its functions included in the Signal Processing Toolkit were used when necessary (e.g. filtering).

1 Dataset and parameters discretization

The scanner that was used to collect the images is a Siemens Magneto Avanto 1.5 Tesla. The scanner is located at the Methodist Hospital (Texas Medical Center, Houston) in the cardiovascular MRI department. All the subjects that underwent the test were healthy. The whole dataset comes from the same magnetic resonance scanner. It basically consists of 5 sets of data. All the sets contain 200 time frames and have the same sampling frequency of one image every 50 ms (20 Hz). The resolution varies among the sets.

I. Long axis view1 from subject 1, resolution 2.15 mm
II. Long axis view1 from subject 2, resolution 2.5 mm
III. Long axis view2 from subject 3, resolution 1.97 mm
IV. Short axis view1 from subject 2, resolution 1.95 mm
V. Short axis view2 from subject 3, resolution 1.97 mm

Hereafter the first time frame of each set is shown.

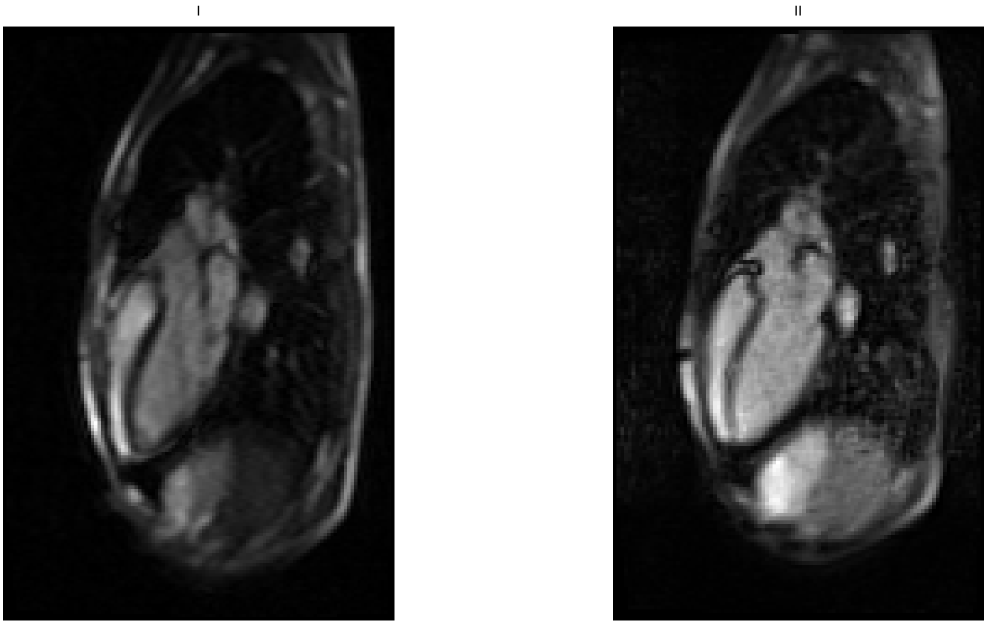


Figure 3.1: First time frame of dataset I e II.

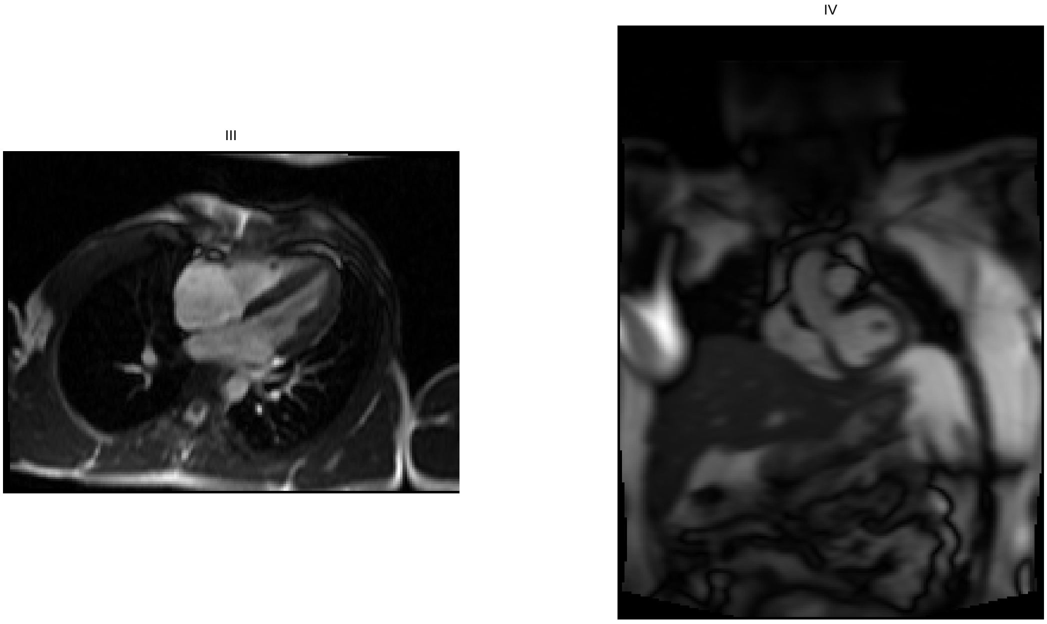


Figure 3.2: First time frame of dataset III e IV.

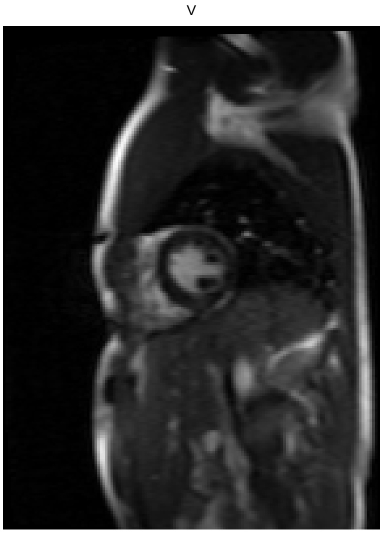


Figure 3.3: First time frame of dataset V.

In order to explore the space of the algorithm's parameters we need to discretize them. As we introduced in chapter 2, some of the parameters are qualitative and do not need to be discretize; some of them are quantitative and need to. We will go over the discretization of all the parameters in order to make this clear. The values of the parameters were chosen accurately, taking into account the features of the images. The parameters were discretize as follows:

- Number of samples along the length *N* = 50,70,100
- Width *W* = 1 mm, 2 mm, 4 mm
- Number of samples along the width *M* = 1,3,6
- Smoothing mode *Smooth* = Moving Average *M*, Classical FIR filter *F*, Spline *S*
  - FIR filters (MA, Classical) Cutoff frequency $f_c = 0.6 f_c$(N) (hard 1), $0.8 f_c$(N) (medium 2), f_c(N) (soft 3)
  - Spline De Boor's parameter = 0.001 (hard 1), 0.01 (medium 2) , 0.1 (soft 3)
- Stopping value method *StopVal* = Constant *C*, Mean *M*, X-Range*X*
  - Value(Constant) *Val*: Threshold = Depends on the image
  - Value(X-Range) *Val*: X = 0.2, 0.4, 0.6
  - Value(Mean) *Val*: none
  - Variants (only for Mean, X-Range) *Var*= Equal *E*, Diverse *D*

The choice of these values is mostly justified in the chapter 2. As we explained in subsection 1.2, the cutoff frequency is a function of the number of samples along the length. For this reason, the cutoff frequencies are all smaller than the one computed with the formula 2.5. The words in *italic* are the tags for each parameter. They will be used in the tables for showing the results.

2 Accuracy of the algorithm

The algorithm has to be initialized. A set of projections was drawn for each set of data:

- I, II
    - 6 projections on the apical zone
    - 10 projections on the middle zone
    - 6 projections on the aortic zone
- III
    - 6 projections on the apical zone
    - 6 projections on the middle zone (papillary muscle)
    - 5 projections on the base zone
- IV
    - 4 projections across the ventricle
    - 4 projections across the ventricle (papillary muscle)
    - 6 projections on the aorta
- V
    - 4 projections across the ventricle
    - 4 projections across the ventricle (papillary muscle)

The number of projections in each zone depends on the view. The long axis views offer a wider image of the heart and a higher number of projections can be drawn. Moreover, the contraction of the heart does not modify the range the projections are working on. In the short axis view (particularly on the dataset V) there is physically less space to draw projections and the heart contraction strongly modify the heart dimensions. This number of projections is the maximum one can draw. For this reason, the accuracy computation differs from the running algorithm. For the accuracy computation we want to have the biggest statistics as possible. The total number of projections processed is 16600 (83x200). For the running algorithm, we have to select the number of projections in order to have a more or less accurate ventricle reconstruction and mind the real-time issue as well.

2.1 Results tables

For each set of data, one table will be shown. In order to understand the tables, the same tags used in section 1 are used. Figure of merit 2.13 is used. The table for this figure of merit will be divided in the zones shown previously (for instance: dataset I, zones=middle, apical, aortic). Each zone has a fixed number of projections and every projection has its least value of accuracy minimized with a certain set of parameters P_o. For each zone. The mode of each parameter is computed. The accuracy for each projection is recomputed with the new set of parameters consisting of the modes for each parameter among the projections in a certain zone. If the new accuracy differs from the first one computed of more than 1 mm, then the original one is kept; otherwise the new accuracy computed with the modes of the parameters is. This relates to what introduced in section 2.3. The main aim is to find a set of parameters suitable in terms of accuracy for at least one zone. Although the minimum is given by a set of parameters P_o, it is possible that moving a little bit (towards the mode of the distribution of that parameter among all the projections) a couple of the parameters the accuracy does not change significantly (1mm). In that case a new set of parameters $\tilde{P}$ is chosen. In the tables, after each zone description, there will be a new set of parameters and a new value of accuracy. This value will be the mean among the accuracies of each projection in that zone, computed with the same set of parameters $\tilde{P}$.

Table 3.1: Dataset I, 6 projections on the aortic zone, 10 on the middle zone, 6 on the apical zone.

Proj	Acc[mm]	StD[mm]	Smooth	f_c	StopVal	Var	Val	W[mm]	N	M
1	1.66	1.7	S	3	X	E	.4	2	70	3
2	1.30	2.2	S	3	X	D	.2	2	100	6
3	2.03	1.5	S	3	X	D	.4	4	100	6
4	0.90	2.1	M	3	X	D	.4	2	100	6
5	1.23	2.2	S	2	X	D	.2	2	70	3
6	1.59	1.9	M	3	M	E	-	2	100	6
$\tilde{P}$	1.81	2.1	S	3	X	D	.4	2	100	6
1	0.83	1.9	S	3	X	E	.4	2	70	3
2	1.30	1.5	F	3	X	D	.2	2	100	3
3	1.51	1.7	F	3	X	D	.4	4	100	3
4	0.73	1.0	F	3	X	D	.4	2	100	6
5	1.43	1.9	S	3	X	D	.2	2	70	3
6	1.29	1.9	M	3	M	E	-	2	100	6
7	0.83	1.9	F	2	X	E	.4	2	100	3
8	1.30	1.5	F	2	X	D	.2	2	100	3
9	0.91	1.7	S	3	X	D	.4	4	100	3
10	1.41	1.1	F	3	X	D	.4	2	100	6
$\tilde{P}$	1.31	1.3	F	3	X	D	.4	2	100	3
1	1.83	2.9	S	3	X	E	.4	2	100	6
2	1.40	1.5	S	3	X	D	.2	2	100	6
3	1.41	1.7	F	3	X	D	.4	4	100	3
4	1.56	2.1	M	3	X	D	.4	2	100	6
5	2.43	1.9	S	3	M	D	-	2	70	3
6	1.59	2.9	M	3	X	E	.4	2	100	6
$\tilde{P}$	1.87	1.8	S	3	X	D	.4	2	100	6

Table 3.2: Dataset II, 6 projections on the aortic zone, 10 on the middle zone, 6 on the apical zone.

Proj	Acc[mm]	StD[mm]	Smooth	f_c	StopVal	Var	Val	W[mm]	N	M
1	1.46	1.7	S	3	X	E	.4	2	70	3
2	1.30	2.5	S	3	X	D	.2	2	100	3
3	1.33	1.9	S	3	X	D	.4	4	100	3
4	1.81	2.1	M	3	X	D	.4	2	100	6
5	1.43	2.0	S	3	X	D	.2	2	70	3
6	1.59	1.9	M	3	X	E	.4	2	100	6
$\tilde{P}$	1.91	2.5	S	3	X	D	.4	2	100	6
1	0.83	1.9	F	3	X	E	.4	2	70	3
2	0.30	1.5	S	3	X	D	.2	2	100	3
3	1.51	1.7	S	3	X	D	.4	4	100	3
4	0.98	2.1	F	3	M	D	-	2	100	6
5	0.99	1.9	F	3	M	D	-	4	70	3
6	1.39	1.9	M	2	X	E	.4	4	100	6
7	0.85	1.9	S	1	X	D	.4	4	70	3
8	1.30	1.5	F	3	X	D	.2	2	100	3
9	1.51	1.7	F	2	X	D	.4	4	100	3
10	1.81	2.1	M	3	X	E	.4	2	100	6
$\tilde{P}$	1.25	1.6	F	3	X	D	.4	4	100	3
1	2.10	1.9	S	3	X	E	.4	2	70	3
2	2.30	1.5	S	3	X	D	.2	2	100	3
3	1.51	1.7	S	3	X	D	.4	4	100	3
4	1.81	2.1	M	3	X	D	.4	2	100	6
5	2.43	1.8	S	3	X	D	.2	2	70	3
6	1.69	1.9	M	3	X	E	.4	2	100	6
$\tilde{P}$	2.05	1.8	S	3	X	D	.4	2	100	6

Table 3.3: Dataset III, 6 projections on the apical zone, 6 on the middle zone with papillary muscle and 5 on the base zone.

Proj	Acc[mm]	StD[mm]	Smooth	f_c	StopVal	Var	Val	W[mm]	N	M
1	1.83	1.9	S	3	X	E	.4	2	70	3
2	1.30	2.5	S	3	X	D	.2	2	100	6
3	2.51	2.7	F	3	M	D	-	4	100	3
4	1.81	2.1	M	3	X	D	.4	2	100	6
5	1.43	1.9	S	3	M	D	-	2	100	6
6	2.59	1.4	M	3	X	D	.4	2	100	6
$\tilde{P}$	2.22	2.1	S	3	X	D	.4	2	100	6
1	3.03	2.2	S	3	X	E	.4	2	100	6
2	2.30	2.8	F	3	M	D	-	2	100	6
3	1.73	1.5	F	3	X	D	.4	4	100	3
4	1.81	2.1	M	2	X	D	.4	2	70	6
5	2.43	2.5	S	3	X	D	.2	2	70	3
6	2.59	1.2	M	3	M	E	-	2	100	6
$\tilde{P}$	2.51	2.1	M	3	X	D	.4	2	100	6
1	0.83	1.2	S	3	X	E	.4	2	70	3
2	1.30	1.5	S	3	X	D	.2	2	100	3
3	0.73	1.0	S	3	X	D	.4	4	100	3
4	0.81	2.0	M	3	M	D	-	2	100	6
5	1.59	1.3	M	3	X	E	.4	2	100	6
$\tilde{P}$	1.71	1.5	S	3	X	D	.4	2	100	3

Table 3.4: Dataset IV, 4 projections across the ventricle, 4 across the papillary muscle, 6 on the aorta.

Proj	Acc[mm]	StD[mm]	Smooth	f_c	StopVal	Var	Val	W[mm]	N	M
1	2.30	2.5	S	3	X	D	.2	2	100	6
2	1.71	1.7	S	3	X	D	.4	4	100	6
3	2.13	1.9	S	3	M	D	-	2	70	3
4	1.89	2.3	M	3	X	E	.4	2	100	6
$\tilde{P}$	2.01	2.8	M	3	X	D	.4	2	100	6
1	2.51	1.7	S	3	X	D	.4	4	100	3
2	0.85	1.3	M	3	X	D	.4	2	100	6
3	2.43	1.9	S	3	X	D	.2	2	70	6
4	1.93	2.9	S	3	X	E	.4	2	70	3
$\tilde{P}$	2.41	2.1	S	3	X	D	.4	2	100	6
1	0.99	1.2	S	3	X	E	.4	2	100	3
2	1.30	1.5	S	3	X	D	.2	2	70	3
3	1.51	1.7	S	3	M	D	-	4	100	3
4	1.41	2.1	M	3	X	D	.4	2	100	6
5	0.88	1.1	S	3	X	D	.2	2	70	3
6	1.59	1.9	M	3	X	E	.4	2	100	6
$\tilde{P}$	1.75	2.1	S	3	X	D	.4	2	100	3

Table 3.5: Dataset V, 4 projections across the ventricle, 4 across the papillary muscle.

Proj	Acc[mm]	StD[mm]	Smooth	f_c	StopVal	Var	Val	W[mm]	N	M
1	2.30	1.5	S	3	X	D	.2	2	100	3
2	2.51	1.7	F	3	X	D	.4	4	100	3
3	1.81	2.1	F	3	X	D	.4	2	100	6
4	2.59	2.9	M	3	M	E	-	2	100	6
$\tilde{P}$	2.51	2.7	F	3	X	D	.4	2	100	6
1	1.93	1.9	S	3	X	E	.4	2	70	3
2	2.30	2.5	F	2	X	D	.2	2	100	3
3	2.51	2.7	F	2	X	D	.4	4	100	3
4	2.43	1.9	F	2	X	D	.2	2	70	3
$\tilde{P}$	2.91	2.9	F	2	X	D	.4	2	100	3

2.2 Results conclusions

The values of the accuracy vary mostly due to the presence of the papillary muscle. Not only it is a problem to decide whether to include the papillary muscle in the segmentation or to treat it like a piece of endocardium, the breathing motion in certain views displaces the heart and therefore the projections are not acquired properly. Since the projections are static, when the heart moves up and down, some projections result overlapped on a completely different structure from the one they were drawn on. This causes the occurrence of anomalous projections. The algorithm can handle them as we explained in the previous chapter. Nevertheless a drop of the accuracy is expected in these cases. The accuracy in absence of the papillary muscle is around 1.5 mm. Given that the real-time images are difficult to segment even manually and that there might be a very small error in the ground truth, this accuracy completely satisfy the purpose of this work. In presence of the papillary muscle the error increases up to 3 mm, which is still a good result. Since the ventricle width in diastole is about 4 cm, if we want to express the error with respect to the heart width it turns out to be about 6% in presence of the papillary muscle and around 2% elsewhere. Even though the accuracy does not vary with the time frame, the relative error expressed as before does not have much sense because the accuracy is not relative to the endocardium width. The standard deviation stays around 2-3 mm. It is the standard deviation of the error, not of the accuracy. We recall the definition of accuracy in formula 2.11. Even if the simulation has been performed with MATLAB, the average time to process one projection has been computed. On a PC Intel$^{®}$ Core^{TM} i7 CPU 960 @3.20GHz the average time is 1.5 ms. This is just to give a quantitative idea of how long the processing can take. Since the images are acquired every 50 ms, the algorithm already works in real-time even for a high number of projections each time frame in the context it was located. When ported to an embedded platform it will for sure go much faster and leave a lot of time to process other algorithms (e.g. surface reconstruction). Some significant figures will be shown hereafter to give an idea of how good the algorithm's accuracy is.

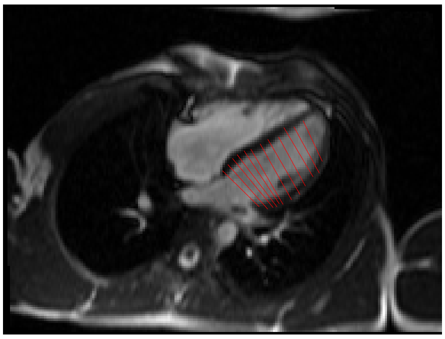


Figure 3.4: Initialization of the simulation for the dataset III.

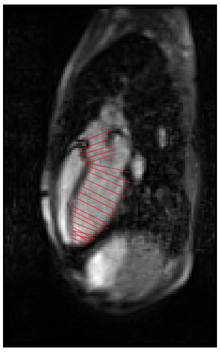


Figure 3.5: Initialization of the simulation for the dataset II.

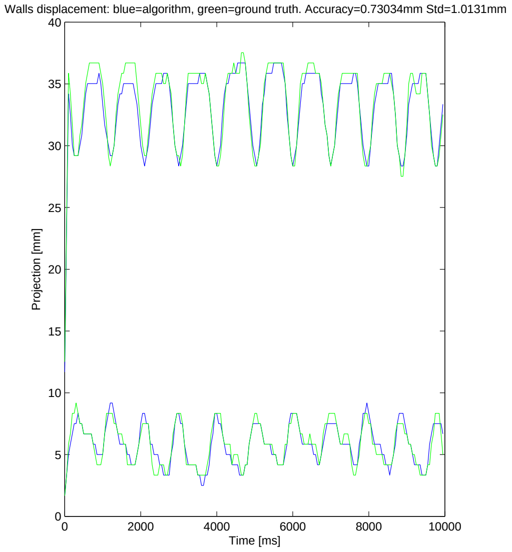


Figure 3.6: Results concerning one middle projection the dataset I. Walls displacement on the left and contraction profile computed as the difference between the two walls on the right.

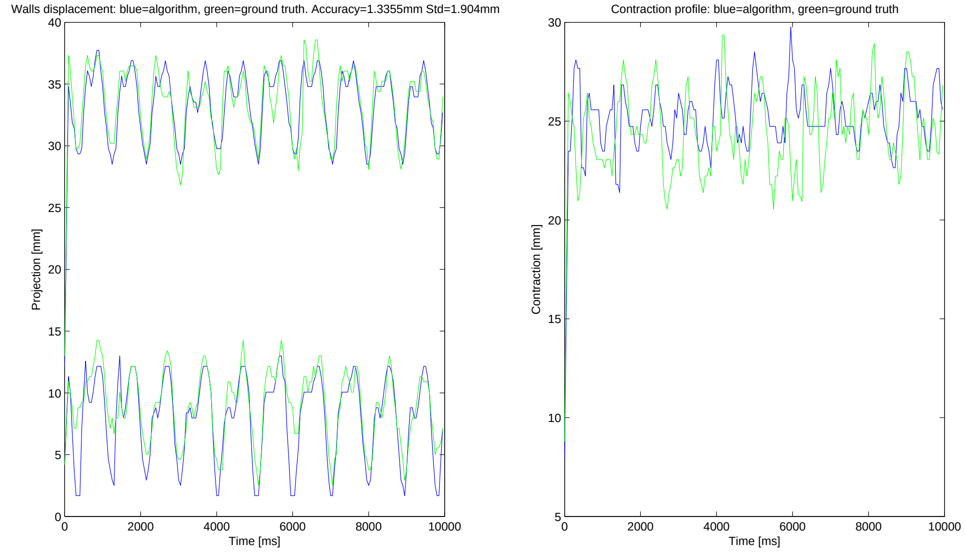


Figure 3.7: Results concerning one aortic projection of the dataset II. Walls displacement on the left and contraction profile computed as the difference between the two walls on the right. The typical wave shape of the aortic pressure is appreciable. We notice moreover that the aorta does not contract as much as the heart.

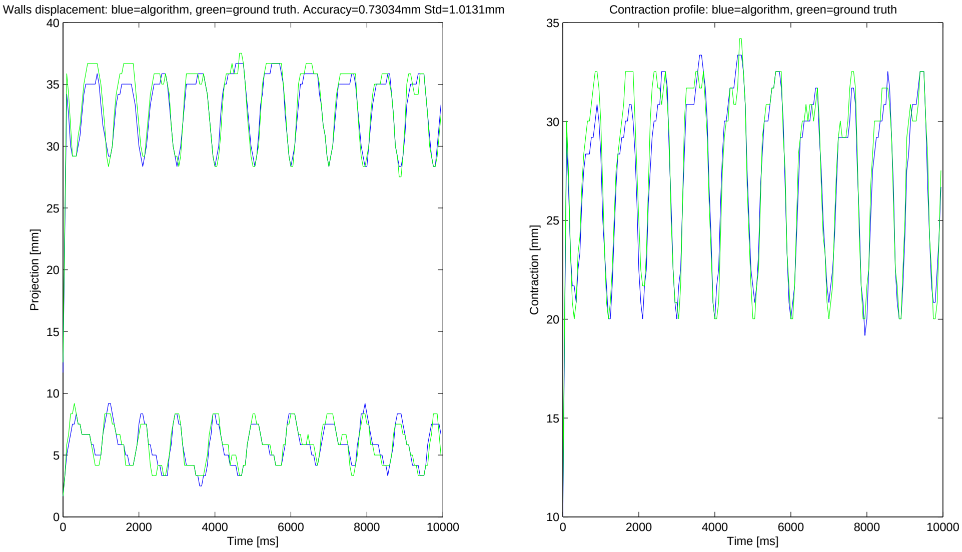


Figure 3.8: Results concerning one base projection of the dataset III. Walls displacement on the left and contraction profile computed as the difference between the two walls on the right.

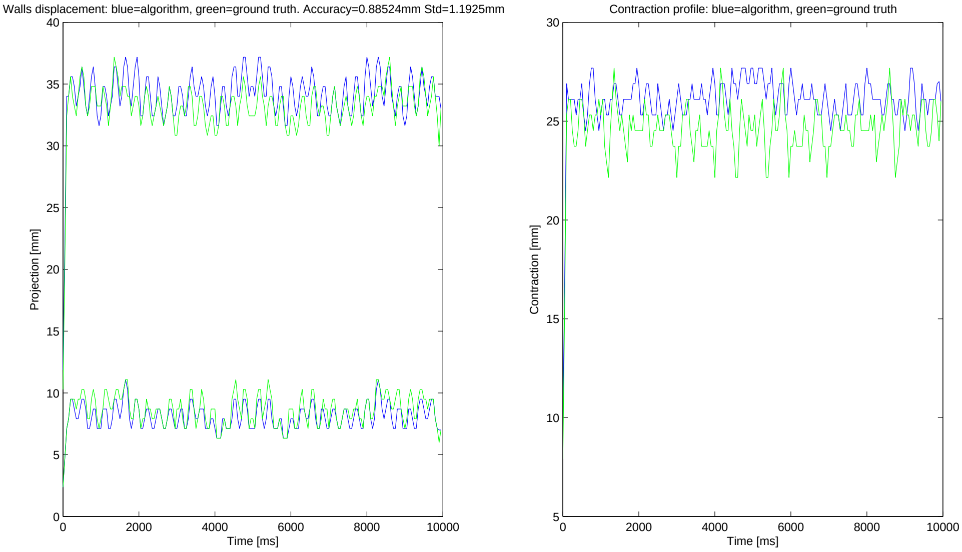


Figure 3.9: Results concerning one aortic projection of the dataset IV. Walls displacement on the left and contraction profile computed as the difference between the two walls on the right.

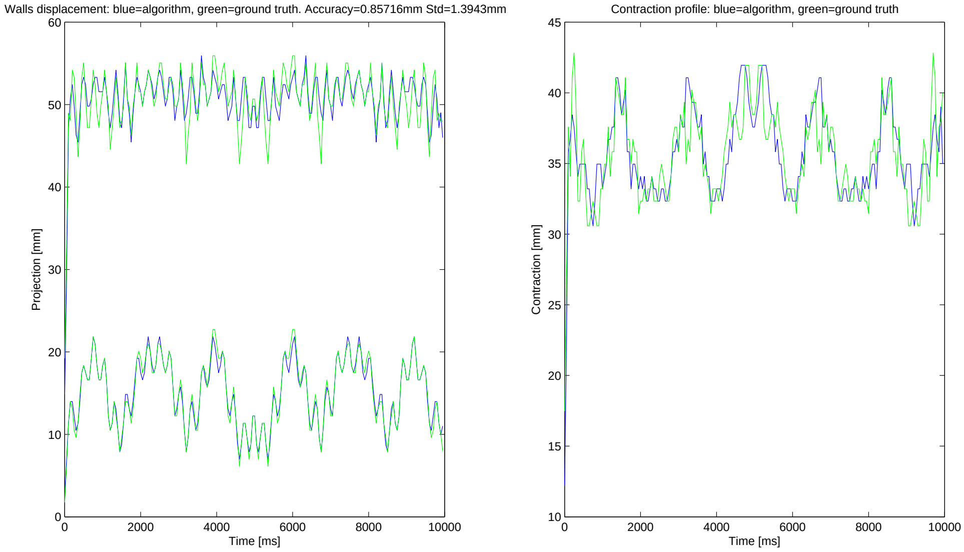
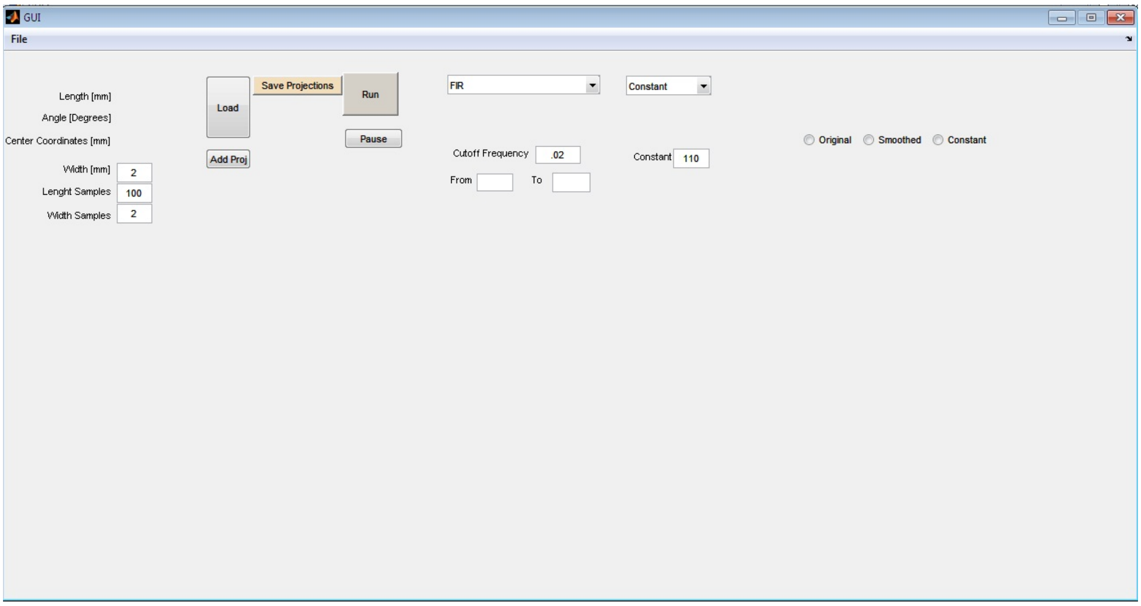


Figure 3.10: Results concerning one papillary projection of the dataset IV. Walls displacement on the left and contraction profile computed as the difference between the two walls on the right.

3 Graphical User Interface (GUI)

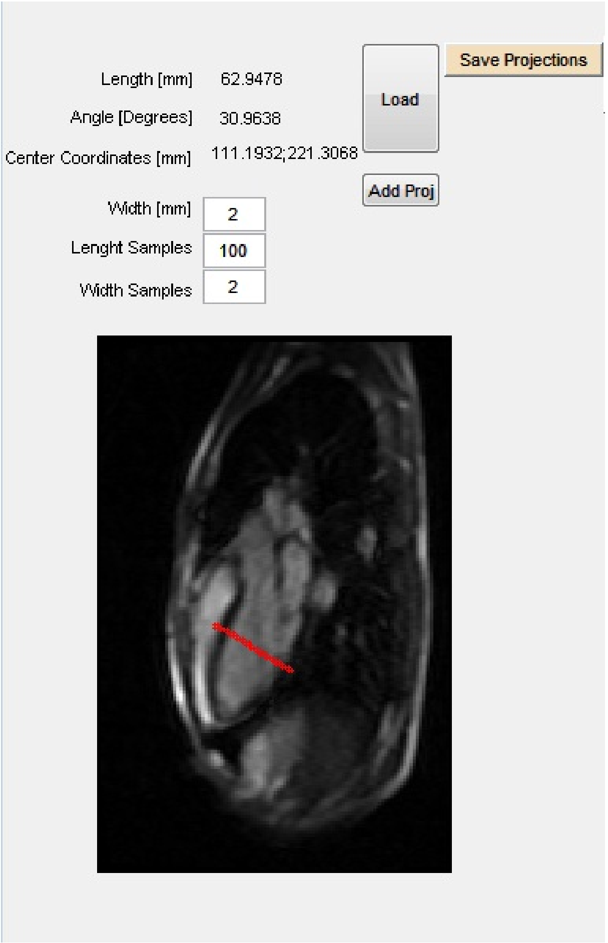
For a more practical use a GUI was designed. The GUI was designed in MATLAB as the rest of the algorithm. It does not want to be a definitive user interface. The purpose of it is just to give the users a friendly environment in which they can try the algorithm functionalities. Moreover it is a nice start for a more complex software on which the clinical operator will have to act.



Width [mm] 2
Lenght Samples 100
Width Samples 2
FIR
Cutoff Frequency .02
Constant
Constant 110

Figure 3.11: The graphical user interface.

As we can appreciate in figure 3.11, the GUI has a lot of parameters to set. We will go through all the steps that allows the user to perform a simulation of the algorithm. It is important to say that the GUI works halfway between online and offline mode. The user begins with hitting on the "Load" button. A window that allows the user to chose a dataset pops up. The dataset can be either segmented (.mat file) or just a folder containing a set of dicom images. The first frame of the dataset will show up below. Once the dataset is selected the user can chose the parameters of the next projection to be drawn (width, samples along the width and along the length). By pressing the "AddProj" button the user can draw one projection on the image. The projection will be drawn with the parameters selected earlier, parameters such as the length of the projection will appear after the user draws it. By clicking again on the "AddProj" button the user can draw as many projections as s/he wants. Once the number of desired projections is reached, the "Save Projections" button has to be hit. This way the projections will be drawn for each time frame of the dataset. The output of this section is a set of N projections times M time frames. In this sense, this part of the algorithm works in offline mode.



Length [mm] 62.9478

Angle [Degrees] 30.9638

Center Coordinates [mm] 111.1932;221.3068

Width [mm] 2

Lenght Samples 100

Width Samples 2

Figure 3.12: The projections drawing interface.

After the projections drawing section, the user has to set all the other parameters. By hitting on the popup windows one can select respectively the filtering modality and the walls identification parameter. Depending on this one, a different set of parameters to set will show up below the popup window.

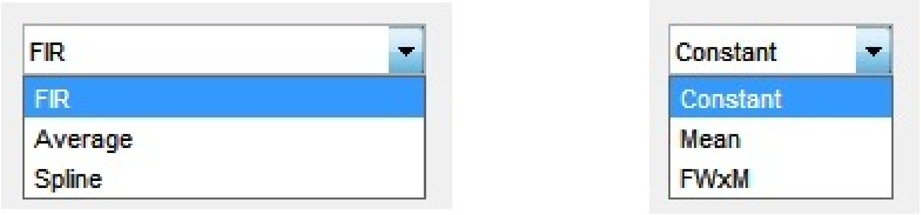
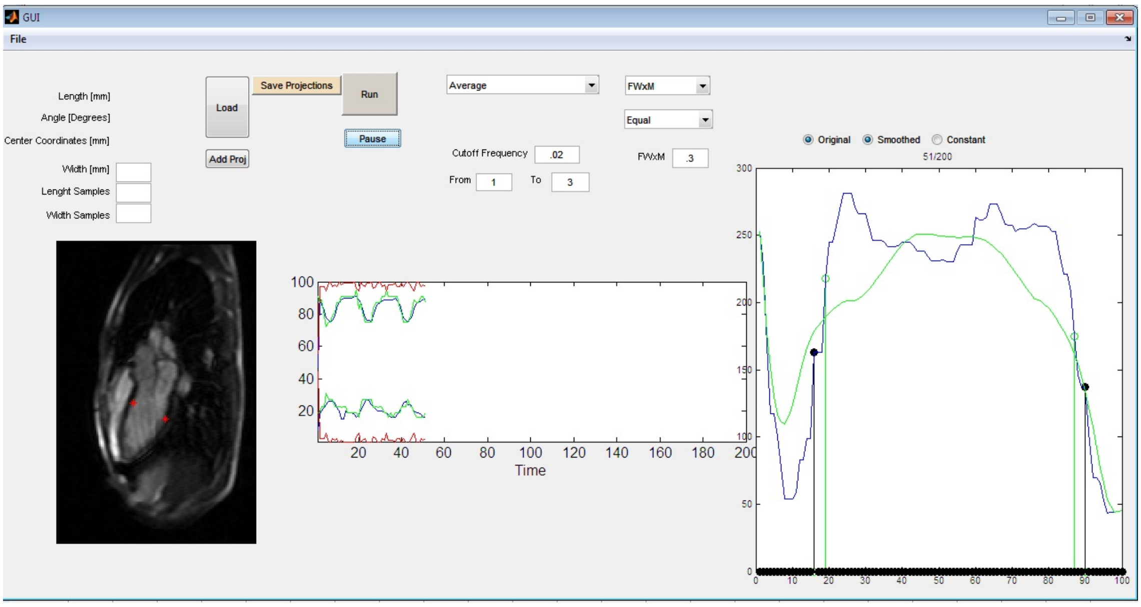


Figure 3.13: The two popup windows. Filtering method selection (left) and walls identification method selection (right).

The two text edit spaces "From" and "To" will respectively contain by default 1 and the number of projections drawn. One can select whether to visualize all the projections or just a part of it by modifying the values inside the boxes. Once all the parameters are set the simulation can start by hitting the "Run" button. From here after the algorithm will work online processing one projection at a time. On the left image the two dots representing the walls will show up. The graph in the middle will show the online walls displacement, both the ground truth and the algorithm output, together with the accuracy for each side. The graph on the right is optional. The radiobutton on top of it can enable the various options. The "Original" activates the processed projection, the GT and the output of the algorithm, the "Smoothed" overlaps the smoothed projection on it, the "Constant" overlaps the stopping value (either Equal or Diverse). Here is a screenshot of the running GUI.



Length [mm]
Angle [Degrees]
Center Coordinates [mm]
Width [mm]
Lenght Samples
Width Samples

Average
FWxM
Equal
Cutoff Frequency .02
From 1 To 3
FWxM .3
● Original ● Smoothed ○ Constant
51/200

Figure 3.14: The running GUI. The middle graph shows the algorithm's output (blue), the ground truth (green), the accuracy for each side (red). The right graph shows the processed projection (blue) and the filtered one (green). The filled black stems represent the algorithm output whereas the empty green ones represent the ground truth.

Chapter 4

General conclusions and future improvements

Real-time medical imaging processing is a very complex topic mostly due to acquisition time issues. In addition, the problem that usually occurs is that there is no time left for any further fine processing. Before the advent of real-time MRI, only two types of medical imaging managed to be within the real-time boundaries: fluoroscopy and ecography. Each of them has their limitations, such as the big amount of ionizing radiations issue of the fluoroscopy or the difficulty in analyzing a deep structure of the ultrasound. The strength points of the algorithm developed within this thesis project are its simpleness and its novel approach to a new field, that is the cardiac real-time MR imaging. The real-time segmentation of the left ventricle has been performed only through ultrasound-based imaging. The 4D heart segmentation is a new challenge that can improve surgical practices such as interventions or visual guidance. The combination of MR imaging and a method based on intensity projections brings the whole field one step up, thanks to the navigator echo technique. A method to find the 3D coordinates of points lying on the endocardium wall has been designed and simulated with MATLAB® on a PC Intel® Core$^{\text{TM}}$ i7 CPU 960 @3.20GHz. The actual segmentation accuracy will depend on how many projections the user chooses to draw and the type of surface reconstruction method used. The accuracy of the presented method has been computed on the single projections basing on manually segmented images turned out to be around 2.5 mm for the worst cases, whereas the average accuracy is around 2 mm. These results make the algorithm suitable and competitive for many purposes. In spite of its good results, it is meant to be just the first part of the overall segmentation algorithm. Many strategies can be adopted for three-dimensional surface reconstruction connecting the points found by the algorithm. These strategies were not treated in this project though are part of the future improvements. The very next goal is testing the optimized algorithm in an embedded platform (DSP or microcontroller) to assert the average time it takes to process one projection. Since the algorithm has been profiled only on a personal computer in the MATLAB environment; it is obvious that once it is ported on an embedded platform its speed performance will strongly increase. Each projection takes in average 1.5 ms to be processed. As the processing time gets shorter, there is more time left in the time slot for more processing. The on-line test on an MR scanner will require to program it for this specific task. In particular the idea is that there will not be the need to acquire the whole image, but just the signal from the voxels part of the projections which will speed the whole process up as well. The use of the algorithm for image-guided surgery is another future goal. The robot-assisted cardiac surgery is particularly related to this work. In fact, one of the purposes of the real-time ventricle segmentation is to give a force feedback to a robot end-effector anytime it hits or gets too close to the segmented wall. In this case a further study has to be done regarding how the projections change their shape in presence of a surgical tool. Besides surgical purposes, the real-time segmentation can be a good method to evaluate cardiac function in real time during patients monitoring. As last; the projections method, associated with the navigator echo technique, can be very useful to segment any type of organ in a fast way.

Bibliography

[1] SM Shea, BE Schirf, X Bi, R Tang, B Lu, RA Omary, D Li (2004). Myocardial perfusion evaluation with T2-prepared gradient echo blood oxygen level dependent imaging at 3 Tesla. Computers in Cardiology, 31:145-148.

[2] (http://www.cardiachealth.org/heart-disease-diagnosis/cardiac-mri). Cardiac Health (July 31, 2012).

[3] (http://www.utsouthwestern.edu/newsroom/news-releases/year-2010/ut-southwestern-unveils-ct-scanner-that-views-whole-organs-in-a-heartbeat-cutting-exposure-time.html) UT Southwestern Medical Center. Next Generation CT Scanner Views Whole Organs in a Heartbeat. ScienceDaily (June 7, 2010).

[4] N Keat, MSc, MIPEM (2001). Real-time CT and CT fluoroscopy. British Journal of Radiology, 74:1088-1090

[5] M Uecker, S Zhang, D Voit, A Karaus, KD Merboldt, J Frahm (2010). Real-time MRI at a resolution of 20 ms. NMR Biomed 23:986-994.

[6] Ordidge RJ, Mansfield P, Doyle M, Coupland RE (1982). Real-time movie images by NMR. Br J Radiol 55:729-733.

[7] Cohen MS (2001). Real-time functional magnetic resonance imaging. Methods 25 (2):201-220.

[8] J Frahm, A Haase, W Hnicke, KD Merboldt, D Matthaei (1985). Hochfrequenz-Impuls und Gradienten-Impuls-Verfahren zur Aufnahme von schnellen NMR-Tomogrammen unter Benutzung von Gradientenechos. German Patent Application P 35 04 734.8, February 12, 1985.

[9] J Frahm, A Haase, D Matthaei (1986). Rapid NMR imaging of dynamic processes using the FLASH technique. Magn Reson Med 3:321-327.

[10] S Zhang, KT Block KT, J Frahm (2010b). Magnetic resonance imaging in real-time: Advances using radial FLASH. J Magn Reson Imag 31:101-109.

[11] M Uecker, T Hohage, KT Block, J Frahm (2008). Image reconstruction by regularized nonlinear inversion Joint estimation of coil sensitivities and image content. Magn Reson Med 60:674-682.

[12] M Uecker, S Zhang, J Frahm (2010b). Nonlinear inverse reconstruction for real-time MRI of the human heart using undersampled radial FLASH. Magn Reson Med 63:1456-1462.

[13] V Wedeen, A Crawley, R Weisskoff, G Holmvang and MS Cohen (1990). Real-time MR imaging of structured fluid flow. Society for Magnetic Resonance in Medicine, 1:164.

[14] S Zhang, N Gersdorff, J Frahm (2011). Real-time magnetic resonance imaging of temporomandibular joint dynamics. The Open Medical Imaging Journal, 5:1-7.

[15] CE Draper, JM Santos, LC Kourtis, TF Besier, M Fredericson, GS Beaupre, GE Gold, SL Delp (2008).Feasibility of using real-time MRI to measure joint kinematics in 1.5T and open-bore 0.5T systems. J Magn Reson Imaging, 28(1):158-66.

[16] Y Suto, M Kamba, T Kato (1995). Dynamic analysis of the pharynx during swallowing using Turbo-FLASH magnetic resonance imaging combined with an oral positive contrast agenta preliminary study. British Journal of Radiology, 68:1099-1102.

[17] P Nordbeck, M Beer, H Kstler, ME Ladd, HH Quick, WR Bauer, O Ritter (2012). Cardiac catheter ablation under real-time magnetic resonance guidance. European Heart Journal Advance Access, doi:10.1093/eurheartj/ehs139.

[18] AN Raval, JD Telep, MA Guttman, C Ozturk, M Jones, RB Thompson, R DeSilva (2005). Real-Time magnetic resonance imaging-guided stenting of aortic coarctation with commercially available catheter devices in swine. circulationaha. 105.542647.

[19] P Kahlert, N Parohl, J Albert, L Schfer, R Reinhardt, GM. Kaiser, I McDougall, B Decker, B Plicht, R Erbel, H Eggebrecht, ME Ladd, HH Quick (2012). Real-time magnetic resonance imaging-guided transarterial aortic valve implantation. J Am Coll Cardiol. 2012;59(2):192-193. doi:10.1016/j.jacc.2011.09.046.

[20] IM Barbash,CE Saikus, AZ Faranesh, K Ratnayaka, O Kocaturk, MY Chen, JA Bell, R Virmani, WH Schenke, MS Hansen, MC Slack, RJ Lederman (2011). Direct percutaneous left ventricular access and port closure. J Am Coll Cardiol Intv.; 4(12):1318-1325.

[21] KA Horvath, M Guttman, M Li, RL Lederman, D Mazilu, O Kocaturk, PV Karmarkar, ER McVeigh (2007). Transapical aortic valve replacement using real-time MRI guidance. Journal of Heart Valve Desease, 39(6):822828.

[22] F Remondino (2003). From point cloud to surface: The modeling and visualization problem. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV-5/W10.

[23] Chuang, Michael L. (1999). Impact of on-line endocardial border detection on determination of left ventricular volume and ejection fraction by transthoracic 3-dimensional echocardiography. Journal of the American Society of Echocardiography, 12:551-558.

[24] HP Burmeistera, T Bitterb, PM Heilerc, A Irintchevb, R Frberd, M Dietzela, PA Baltzera, LR Schadc, JR Reichenbache, H Gudziolb, O Guntinas-Lichiusb, WA Kaisera (2012). Imaging of lamination patterns of the adult human olfactory bulb and tract: In vitro comparison of standard- and high-resolution 3 T MRI, and MR microscopy at 9.4 T. NeuroImage, 60(3):1662:1670.

[25] R de Nijs, S Holm, G Thomsen, M Ziebell, C Svarer (2010). Experimental determination of the weighting factor for the energy window subtraction-based downscatter correction for I-123 in brain SPECT studies. Journal of Medical Phisics Year, 35(4):215-222.

[26] YP Du, M Saranathan, TKF Foo (2004). An accurate, robust, and computationally efficient navigator algorithm for measuring diaphragm positions. Journal Of Cardiovascular Magnetic Resonance, 6(2):483-490.

[27] M Stuber, RM Botnar, PG Danias, KV Kissinger, WJ Manning (1990). Impact of navigator timing on free-breathing submillimeter 3D coronarv magnetic resonance angiography. Radiology, 212:579-587.

[28] D Firmin, J Keegan (2001). Navigator echoes in cardiac magnetic resonance. Journal of Cardiovascular Magnetic Resonance, 3(3), 183-193.

[29] JS Suri (2000). Computer vision, pattern recognition and image processing in left ventricle segmentation: the last 50 years. Pattern Analysis and Applications, 3(3):209242.

[30] L Imanirad (2006). Segmentation and Tracking of the Left Ventricle in Cardiac MRI. A thesis submitted in conformity with the requirements for the degree of Master of Applied Science Graduate Department of The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto.

[31] WE Higgins, EL Ritman (1990). 3D Image-Enhancement Technique for Volumetric Cardiac Images. In AC. Bovik and WE Higgins, editors, SPIE, Biomedical Image Processing, 1245:159170.

[32] GI Sanchez-Ortiz, JA Noble, GJT Wright, J Feldmar, M Mulet-Parada (1999). Automated LV Motion Analysis from 3D Echocardiography. In Medical Image Understanding and Analysis, pages 8588.

[33] M Lynch, O Ghita, PF Whelan (2006). Automatic Segmentation of the Left Ventricle Cavity and Myocardium in MRI Data. Elsevier Computers in Biology and Medicine, 34(4):389407.

[34] M Kass, AP Witkin, D Terzopoulos (1987). Snakes: Active Contour Models. International Journal of Computer Vision, 1(4):321331.

[35] A Blake, M. Isard (1998). Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion. Springer-Verlag, Secaucus, NJ, USA.

[36] DR Cheung (2004). Motion Segmentation Incorporating Active Contours for Spatial Coherence. Masters thesis, University of Toronto.

[37] T McInerney, D Terzopoulos (1996). Deformable Models in Medical Image Analysis: a Survey. Medical Image Analysis, 1(2):91108.

[38] DN Davis (1995). The Application of Active Contour Models to MR and CT Images. Technical report, Medical Vision Group, University of Birmingham, Edgbaston, Birmingham, UK.

[39] L Spreeuwers, M Breeuwer. Detection of Left Ventricular Epi- and Endocardian Borders Using Coupled Active Contours. International Congress Series, 1256:11471152.

[40] SOsher, JA Sethian (1988). Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. Journal of Computational Physics, 79:1249, 1988.

[41] S Osher, R Fedkiw (2003). Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag, Secaucus, NJ, USA.

[42] V Caselles, R Kimmel, G Sapiro (1997). Geodesic Active Contours. International Journal of Computer Vision, 22(1):6179.

[43] TF Cootes, CJ Taylor, DH Cooper, J Graham (1995). Active Shape ModelsTheir Training and Application. Computer Vision and Image Understanding, 61(1):3859.

[44] TF Cootes, GJ Edwards, CJ Taylor (1998). Active Appearance Models. In H. Burkhardt and B. Neumann, editors, Proceedings of European Conference on Computer Vision, 2:484498.

[45] S Ordas, L Biosrobert, M Huguet, AF Frangi (2003). Active Shape Models with Invariant Optimal Features (iof-asm) Application to Cardiac MRI Segmentation. Computers in Cardiology, 30:633636.

[46] A Tsai, A Yezzi, W Wells, C Tempany, D Tucker, A Fan, E Grimson, A Willsky (2003). A Shape-Based Approach to the Segmentation of Medical Imagery Using Level Sets. IEEE Transactions on Medical Imaging, 22(2):137154.

[47] RM Lapp, M Lorenzo-Valdes, D Rueckert (2004). 3D/4D Cardiac Segmentation Using Active Appearance Models, Nonrigid Registration, and the Insight Toolkit. In C Barillot, DR Haynor, P Hellier, editors, Proceedings of the seventh International Conference on Medical Image Computing and Computer-Assisted Intervention, 1:419426.

[48] X Huang, Z Li, D Metaxas (2004). Learning Coupled Prior Shape and Appearance Models for Segmentation. In Proceedings of the Seventh International Conference on Medical Image Computing and Computer-Assisted Intervention, 6069.

[49] JMB Dias, JMN Leitao (1996). Wall Position and Thickness Estimation from Sequences of Echocardiographic Images. IEEE Transactions on Medical Imaging, 15(1):2538.

[50] J Weng, A Singh, MY Chiu (1997). Learning-Based Ventricle Detection from Cardiac MR and CT Images. IEEE Transactions on Medical Imaging, 16(4):378391.

[51] W Sun, M Etin, R Chan, V Reddy, G Holmvang, V Chandar, A Willsky (2005). Segmenting and Tracking the Left Ventricle by Learning the Dynamics in Cardiac Images. Information Processing in Medical Imaging, number 3565 in Lecture Notes in Computer Science, 553565.

[52] J Sénégas, T Netsch, CA Cocosco, G Lund, A Stork (2004). Segmentation of Medical Images with a Shape and Motion Model: A Bayesian Perspective. Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, number 3117 in Lecture Notes in Computer Science, 157168.

[53] ME McRae, M Rodger, BA Bailey (2009). Transcatheter and transapical aortic valve replacement. Crit Care Nurse, 29(1):22-37.

[54] NV Navkar, E Yeniaras, DJ Shah, NV Tsekos, Z Deng (2011). Generation of 4D access corridors from real-time multislice MRI for guiding transapical aortic valvuloplasties. Med Image Comput Comput Assist Intervention, 14(Pt 1):251-258.

[55] Programs for Digital Signal Processing, IEEE Press, New York, 1979. Algorithm 5.2.

[56] C De Boor (1998). Calculation of the Smoothing Spline With Weighted Roughness Measure. Mathematical Models and Methods in Applied Sciences, 11(1):33-41