Automated Segmentation of the Pericardium Using a Feature Based Multi-atlas Approach

Alexander Norlén

Master's thesis
2014:E53

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

CENTRUM SCIENTIARUM MATHEMATICARUM

Abstract

Multi-atlas segmentation is a widely used method that has proved to work well for the problem of segmenting organs in medical images. But standard methods are time consuming and the amount of data quickly grows to a point making use of these methods intractable. In this work we present a fully automatic method for segmentation of the pericardium in 3D CTA-images. We use a multi-atlas approach based on feature based registration (SURF) and use RANSAC to handle the large amount of outliers. The multi-atlas votes are fused by incorporating them into an MRF together with the intensity information of the target image and the optimal segmentation is found efficiently using graph cuts. We evaluate our method on a set of 10 CTA-volumes with manual expert delineation of the pericardium and we show that our method provides comparable results to a standard multi-atlas algorithm but at a large gain in computational efficiency.

Keywords: computer vision, medical image analysis, multi-atlas segmentation, feature based registration, Markov Random Fields, pericardium segmentation.

Acknowledgments

I would like to thank my supervisors Prof. Fredrik Kahl and Ass. Prof. Olof Enqvist for their valuable comments and insights regarding this project and the report. I would also like to take the opportunity to thank those who might be relatively tired of hearing me rambling about hearts and registration, but still for some reason find the energy to stay and listen.

Contents

List of Figures

Chapter 1

Introduction

According to the World Health Organization cardiovascular diseases are the number one cause of death worldwide [24]. Visceral adipose tissue, which is fat surrounding internal organs, may be a marker for greater risk of different metabolic and cardiovascular diseases. Epicardial fat is the visceral fat depot enclosed by the pericardial sac. In other words it is the fat located around the heart but inside of pericardial sac that surrounds the heart. In Figure 1.1 we see a 2D slice of a CT volume with the pericardium and the epicardial fat highlighted. In recent years, several studies have shown a relationship between increased volume of epicardial fat and coronary artery disease, coronary plaque, adverse cardiovascular events, myocardial ischemia and atrial fibrillation. Because of this scientific evidence there is a need for further investigation concerning the prognostic importance of epicardial fat, see [6].

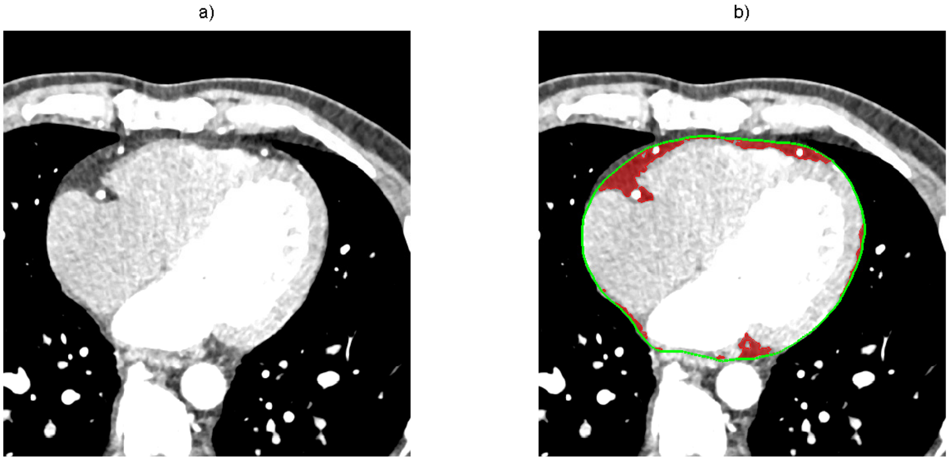

Figure 1.1: a) A slice of a CT-volume of the heart. b) The same slice as in a) but with a manual delineation done by an expert of the pericardium (green) and the epicardial fat highlighted (red).

The Swedish CArdioPulmonary bioImage Study (SCAPIS) is a unique research project that started in 2012 in a collaboration between Sahlgrenska University Hospital, the University of Gothenburg and the Swedish Heart-Lung

Foundation. It is a large scale studie which aims at collecting CT, MR and ultrasound images form 30 000 men and women between the ages of 50-65 years. This database, which will be the largest of its kind, will then serve as a national knowledge base for the study of identifying risk factors that show predisposition towards heart, lung and cardiovascular diseases [15].

This database is an opportunity for investigating the prognostic importance of epicardial fat. However, measuring the epicardial fat manually is very time consuming and especially doing it on 30000 patients. Therefore there is a great need for a fully automatic method for measuring epicardial fat. The pericardium is a barely visible thin line in CT-scans. In many parts of the image an expert needs to rely on knowledge on which other anatomical structures must be inside or outside the pericardium to be able to guess where the pericardium is located. This makes delineating the pericardium a non-trivial problem.

1.1 Problem Formulation

An image I is regarded as a set of pixels/voxels $\mathcal{P}$ where each voxel $p \in \mathcal{P}$ has an intensity i_p. In other words the image is the set of intensities $I = \{i_p \mid p \in \mathcal{P}\}$.

Given a CT-volume I (this volume will often be referred to as an image but keep in mind that it is a 3D image) we want to label each voxel p (a voxel will interchangeably throughout this work be referred to as a pixel in which case of course we mean a 3D pixel) with a label $l_p \in \{0, 1\}$ that should correspond to if the pixel is either belonging to the region *inside of the pericardium* (denoted $l_p = 1$) or not belonging to this region, i.e. *background* (denoted $l_p = 0$). A segmentation, i.e. a set of labeled voxels $\mathcal{L} = \{l_p \mid p \in \mathcal{P}\}$, will be represented by a binary volume with the same size as I with ones representing voxels labeled as object and zeros representing background. The boundary of this mask should correspond spatially to the pericardium.

Given one of these CT-volumes I, we want to estimate a labeling $\mathcal{L}^*$ that maximizes the *Jaccard index* between the estimated labeling and the manual labeling $\mathcal{L}$ (the gold standard).

The Jaccard index measures the similarity between labelings A and B (or more generally two sets A and B) and is defined as the size of the intersection between the two sets divided by size the union, i.e.

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{1.1}$$

The Jaccard index takes values $0 \leq \text{Jaccard}(A, B) \leq 1$ where $\text{Jaccard}(A, B) = 0$ means that there is no overlap between A and B and $\text{Jaccard}(A, B) = 1$ would mean a perfect segmentation. Hence, we want to estimate a labeling that is as similar as possible to the labeling done by the expert.

When $\mathcal{L}^*$ is estimated the epicardial fat volume is easily measured by thresholding. In CT-images the intensities can be directly related to a physical unit called the Hounsfield Unit (HU) [23] which measures radiodensity. Different types of tissue have different radiodensity and in this work HU between -192 and -30 are considered to be fat. This means that when we have segmented the pericardium we simply count the voxels inside this area that have intensities between -192 and -30 and multiply by the volume of each voxel.

1.2 Data Set

In this work we are given 10 CT-volumes of the heart with corresponding manual labelings of the pericardium done by an expert. CT, or Computed Tomography is a method where beams of x-rays are passed from a rotating device through an area of interest. These x rays are computer processed to a series of consecutive tomographic images (slices). This stack of 2D images is then used to generate a 3D image of the object. The patients were given contrast material which means that the blood and tissue especially around the left atrium and ventricle become more detailed (see Figure 1.1). Some contrast material also enters the pericardium itself making it easier to distinguish than if no contrast was used.

The images have resolutions ranging between $512 \times 512 \times 342$ and $512 \times 512 \times 458$ voxels all with voxel dimensions $0.3906 \times 0.3906 \times 0.3000$. That means that for each image there are more than 10^8 voxels that we need to classify.

The manual labels where done by an expert. The delineation was done on every 10th slice. But not only in one viewing direction. The same thing was done in all three viewing directions of each volume meaning that we had a stack of 2D delineations in each view, namely axial, coronal and sagittal. These where interpolated into a final volume that was approved by the same expert. We refer to the manual labelings as the Gold Standard.

1.3 Proposed Solution

In this section we will describe how it is that we propose to solve the problem and it should be viewed as an overview of this entire work. Everything mentioned in this proposition will be explained and motivated more deeply in the following chapters.

We propose to solve the problem of segmenting the inside of the pericardium by a combination of multi-atlas segmentation based on feature based registration and integrating this information into a probabilistic framework that is globally optimized through graph cuts.

Multi-atlas segmentation is based on image registration. An image and its corresponding manual labeling is referred to as an *atlas*. By registering the image of the atlas onto an unlabeled *target image* we obtain a transformation that in some sense aligns the image onto the target image. If we apply the same transformation on the labels of the atlas we align the labels onto the unlabeled image. This is called label propagation and results in a guess of where the region of interest is in the target image. Of course, the better the registration problem is solved, the better the guess.

By doing this for a set of atlases we obtain a good initial guess of where the pericardium is spatially (see Figure 1.2). These registrations will be done in two parts. Firstly, an affine (rigid) transformation will be estimated. Secondly, that affine transformation will be used as an initialization for a non-rigid transformation based on B-splines.

Since image registrations can be very time consuming we propose to base the multi-atlas segmentation on feature based registration (specifically SURF, Speeded Up Robust Features [1]). Feature based registration is a lot faster than more widely used (in medical image analysis) intensity based registration methods. It is not used much since medical images produce a lot of outliers.

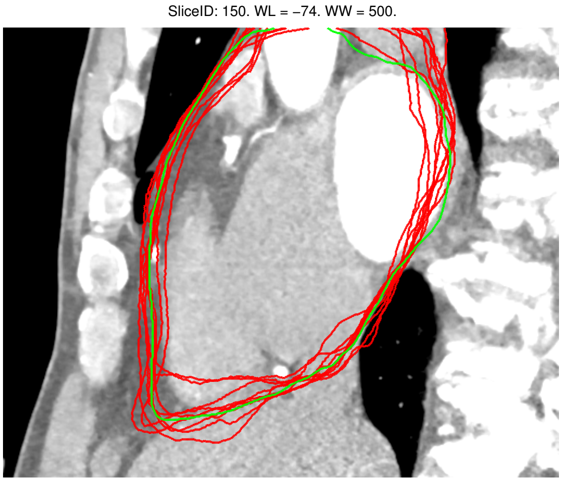

Figure 1.2: A sagittal slice of one of the atlases. By aligning the labels of several atlases (by estimating a transformation through registration) onto the target image we get a good estimate of the segmentation. Each red line in this figure represents the boundary of an aligned labeling using feature based non-linear registration. The green line corresponds to the boundary of the gold standard (the manual labeling).

But we propose to handle the outliers mainly using Random Sample Consesus (RANSAC), [9]. To evaluate this method we compare the results with Niftyreg which is an intensity based method used widely for medical registration.

The problem remains on how to decide on a final segmentation given the guesses (votes) from the atlas registrations. The most straight forward way is to include all voxels which half or more of the atlases label as inside the pericardium. This is called majority voting (or decision fusion). Sometimes the feature based registration makes mistakes that are obvious when looking at the specific intensities of the voxels. For example, this can happen if lung cavity is included in the segmentation In Figure 1.2 you can see this effect where some of the red lines pass through the lung meaning that some of the atlases estimate the lung to be inside of the pericardium. These types of errors are easily corrected by incorporating the votes into a Markov Random Field (MRF) (see e.g. [4]) together with the intensity information. The MRF has two main advantages. It is very flexible in the sense that it is easy to incorporate different information and it can be optimized in polynomial time by representing the MRF as a graph and finding the minimal cut through the graph using maxflow algorithms.

The MRF in its most used formulation only regularizes the boundary making the segmentation surface smooth. We propose an extended formulation that makes the cost of the boundary data dependent, actively pushing the boundary towards a probable path. In this work we only use pixel intensity and the number of multi-atlas votes as observations on which we build the MRF but the framework is easily expandable to more complex features and classifiers.

1.4 Contributions and Related Work

Recently a few methods have been developed for fully automated pericardium segmentation. In [19] Shahzad et al. use multi-atlas segmentation with majority voting. Practically the same method as was used by Kirisly et al. [11] for cardiac segmentation. Both algorithms were based on intensity based registration (Elastix). Dey et al. [7] used another intensity based registration algorithm (Demons) and proposed to speed up the the segmentation time by co-registering the atlases before hand and given an unlabeled image only performing one atlas registration. By measuring the difference between each atlas and the target image a weight was calculated measuring the importance of the atlas for the decision fusion.

The contributions made by this work is partly that feature based registration works excellent for registration of medical images and at huge gain in computational efficiency. Especially we show that feature based registration proves considerably more accurate and robust for initialization of the nonrigid registrations compared to the computationally more demanding intensity based algorithm Niftyreg. We also propose an efficient algorithm for pericardium segmentation based on feature based multi-atlas segmentation that shows significantly improved results over previous state-of-the-art-methods.

1.5 Structure of the Report

In Chapter 2 we explain the theory that is needed to follow the what is covered in this report. The problem of image registration (Section 2.1) Markov Random Fields (Section 2.2) and graph cuts (Section 2.3) are explained. Everything in this chapter is frequently used notions in image analysis and computer vision and can be skipped if the reader already is familiar with these terms. The method is divided into two parts. In Chapter 3 we begin by presenting multi-atlas segmentation in general. We continue with presenting two different registration methods that we evaluate as basis for the multi-atlas segmentation. One feature based using SURF, Lowe matching and RANSAC (Section 3.1) and one intensity based using Niftyreg (Section 3.2). In Chapter 4 we present our method for fusing the votes from the multi-atlas registrations into a final segmentation by optimizing an MRF through graph cuts. In Chapter 5 we cover all the details of the implementation that is needed for reproducibility (e.g. different settings and parameters). Thereafter (in Chapter 6) we present the results where we evaluate the different steps of the algorithm. The results are discussed in Chapter 7 and lastly we present the conclusion in Chapter 8.

Chapter 2

Theory

In this chapter we will cover some theory that the reader needs to be familiar with to be able to follow the rest of this report. Firstly, the problem of image registration and two types of widely used transformations that we will use in this work (affine transformation and non-rigid transformation based on B-splines) will be explained. Using our knowledge of registration we can then proceed to explain multi-atlas segmentation. After that the theory of Markov Random fields will be presented and a way to formulate the problem of finding the maximum a posteriori probability of this Markov Random Field by finding the minimal cut, or equivalently maximal flow, through a graph (often referred to as graph cuts).

2.1 Image Registration

This section covers the basics of image registration. For further information see for example [10, 25, 20].

2.1.1 Problem Formulation

Image registration is the process of spatially aligning two images, i.e. finding a one-to-one map between one image and another so that the corresponding points in the images refer to the same point in the object they are depicting. Formally this can be formulated as:

Given a *target image* I_t (also in the literature referred to as reference or static image) and a *source image* I_s (floating or moving image), find the values of the parameters $\boldsymbol{\theta}$ of the transformation (mapping function) $\mathbf{T}(\boldsymbol{\theta})$ that minimizes a cost function ρ, i.e.

$$\arg\min_{\boldsymbol{\theta}} \left(\rho(I_t, \mathbf{T}(\boldsymbol{\theta}) \circ I_s)\right) \tag{2.1}$$

where the cost function ρ is a measure of the accuracy of the registration, i.e. the similarity between the target image and the transformed source image.

2.1.2 Transformations

There are many ways to define the mapping function $\mathbf{T}$. You want to have a transformation model that describes the real expected transformation between the source and target images as accurately as possible. At the same time you do not want the model to be too complex. If the model has a lot of parameters not only will the optimization be computationally more demanding, it will also be harder to optimize.

To try and avoid the problem of finding local minima the problem is usually split into two parts. The first part is a rough registration with a simpler *affine transformation*. Since the affine transformation is easier to optimize we use that as an initialization for the second part, which is a more complex free-form deformation which allows for local deformations in the image.

Affine Transformation

The affine transformation (here denoted $\mathbf{T}_{\text{aff}}$) is a composition of a translation and a linear map and can be represented by a 4×4-matrix with 12 parameters ($\boldsymbol{\theta} = \theta_1, \ldots, \theta_{12}$) and can be described by

$$\mathbf{T}_{\text{aff}}(\boldsymbol{\theta}) = \begin{bmatrix} \theta_1 & \theta_4 & \theta_7 & \theta_{10} \\ \theta_2 & \theta_5 & \theta_8 & \theta_{11} \\ \theta_3 & \theta_6 & \theta_9 & \theta_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

for each voxel the transformation is then defined by

$$\begin{bmatrix} \boldsymbol{y} \\ 1 \end{bmatrix} = \mathbf{T}_{\text{aff}}(\boldsymbol{\theta}) \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \tag{2.3}$$

where $\boldsymbol{x}$ is the voxel coordinates in the source image and $\boldsymbol{y}$ is the new transformed coordinates.

The intensity based registration method that we evaluate estimate an affine transformation. But the feature based method for efficiency estimates a *rigid transformation*. A rigid transformation is a special case of an affine transformation which only allows for rotation and translation and the number of degrees of freedom are reduced to 6.

Free-Form Deformation

The affine transformation is a global transformation but the anatomical structure of the hearts that we will try to register will display more complex relations. This means that we need a transformation that also can handle local deformations. In this work we will be using two methods which both use a free-form deformation (FFD) model based on cubic B-splines. The method is by Rueckert et al. from 1999 [18]. FFD is a method which has got wide acceptance in the medical image analysis community [20].

A grid of control points is superimposed on the image and the basic idea is that the image will be influenced by manipulating the control points. The control points control the B-splines and the resulting deformation will be a smooth C^2 continuous transformation. Another advantage of B-splines is that they have local support, i.e. a control point will only influence a local region around that control point which makes the transformation easier to optimize and compute.

To mathematically formulate the FFD (following the formulation in [18]) we define the domain of the (in our case 3D) image as

$$\Omega = \{(x, y, z) \mid 0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z\}$$

where X, Y and Z define the size of the image. An $n_x \times n_y \times n_z$ grid of uniformly spaced control points $\boldsymbol{\theta}_{i,j,k}$ is superimposed on the image. By moving the control points the image is transformed which means that the set of parameters is the position of these points. The transformation of a point $\boldsymbol{x} = (x, y, z)$ is defined as

$$\mathbf{T}_{\text{ffd}}(\boldsymbol{\theta}) \circ \boldsymbol{x} = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(u) B_m(v) B_n(w) \boldsymbol{\theta}_{i+l,j+m,k+n} \tag{2.4}$$

where $i = \lfloor x/n_x \rfloor - 1$, $j = \lfloor y/n_y \rfloor - 1$, $k = \lfloor z/n_z \rfloor - 1$, $u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$, $w = z/n_z - \lfloor z/n_z \rfloor$ and B_l represents the l:th basis function of the B-spline, i.e.

$$\begin{aligned}
B_0(u) &= (1-u)^3/6 \\
B_1(u) &= (3u^3 - 6u^2 + 4)/6 \\
B_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6 \\
B_3(u) &= u^3/6.
\end{aligned}$$

If you have a 10 × 10 × 10 grid of control points there are 3000 parameters that need to be optimized but as is clear from (2.4) the transformation of $\boldsymbol{x}$ is only dependent of the location of the 27 closest control points. This means that if you move any of the other control points it will not affect this part of the image and hence the support for the control points is local which makes the parameters easier to optimize.

For a deeper read about B-Splines in general see for example [5]. For a comparison with other methods see [20].

2.2 Markov Random Fields

Probabilistic graphical models combine knowledge from probability theory and graph theory and are a powerful formalism for a wide range of problems in various scientific fields. In computer vision (and other fields) one especially powerful method is regarding the image as originating from a Markov Random Field (MRF), partially because it can model the underlying quantity to be smooth (which is often the case in computer vision) and because it is very adaptive and can handle a wide variety of priors. If the MRF is formulated in a sensible way it can be formulated as a graph and the maximum a posteriori distribution can be found using graph cuts, i.e. it can be globally optimized in polynomial time.

Following the notation in [4], you have a set $\mathcal{P} = \{1, \ldots, m\}$ of sites p (pixels/voxels). You have a neighborhood system $\mathcal{N} = \{\mathcal{N}_p \mid p \in \mathcal{P}\}$ where $\mathcal{N}_p$ is the set of pixels that are considered neighbors to p and a field (set) of random variables $F = \{F_p \mid p \in \mathcal{P}\}$ where each random variable F_p can take a value f_p in some set of labels. A joint event $\{F_p = f_p \mid p \in \mathcal{P}\}$ is abbreviated $F = f$ where $f = \{f_p \mid p \in \mathcal{P}\}$ is a realization (or configuration) of the random field.

In the context of segmentation the optimal configuration will be used directly as labels describing the final segmentation (i.e. $f_p = l_p$). For now we choose to separate the notation between f_p (meaning a realization of the random variable F_p) and l_p (representing the label of a voxel in an image). We will abbreviate $\Pr(F = f)$ as $\Pr(f)$ and $\Pr(F_p = f_p)$ as $\Pr(f_p)$. An MRF is a field F with the property (known as local Markov property) that each random variable F_p only depends on its neighbors

$$\Pr(f_p | f_{\mathcal{P}\setminus\{p\}}) = \Pr(f_p | f_{\mathcal{N}_p}), \quad \forall p \in \mathcal{P}. \tag{2.5}$$

According to the Hammersley-Clifford theorem any distribution that obeys the Markov property (2.5) can be written as

$$\Pr(f) \propto \prod_{c \in \mathcal{C}} \exp\left(-V_c(f_c)\right) = \exp\left(-\sum_{c \in \mathcal{C}} V_c(f_c)\right) \tag{2.6}$$

where $\mathcal{C}$ is the set of all maximal *cliques* of the MRF. A clique is a subset of variables that are all connected to each other. V_c is called the *potential function* of the clique c and is a positive real-valued function on the possible configurations f_c of the clique. In this work we will only consider the *pairwise MRF* in which the probability in (2.6) is factorized into potential functions defined on cliques of size strictly less than three. Although the MRF can always be defined by potential functions on the maximal cliques $\mathcal{C}$ usually the potential function is split onto set of *unary potentials* V_p which are defined on single variables and a set of *pairwise potentials* $V_{p,q}$ which are defined on pairs of variables. The probability of a configuration can then be written

$$\Pr(f) \propto exp\left(-\sum_{p \in \mathcal{P}} V_p(f_p) - \sum_{\{p,q\} \in \mathcal{N}} V_{\{p,q\}}(f_p, f_q)\right). \tag{2.7}$$

The realization of the field f is generally not observed directly so it needs to be estimated through the joint event $O = \{O_p = o_p \mid p \in \mathcal{P}\}$ referred to as the observation. In computer vision the realization o_p can for example be the observed intensity at the pixel p. In this work the observation will be a combination of the intensities in the image and vote map from the multi-atlas segmentation. The probability that we are interested in is the posterior probability $\Pr(f|O)$ which can be defined straight forwardly as

$$\Pr(f|O) \propto \exp\left(-\sum_{p \in \mathcal{P}} V_p(f_p; o_p) - \sum_{\{p,p\} \in \mathcal{N}} V_{\{p,q\}}(f_p, f_q; o_p, o_q)\right). \tag{2.8}$$

We are interested in finding the configuration f that maximizes the posterior probability (2.8). Since the logarithm is a monotonically decreasing function maximizing the posterior probability is the same as minimizing the *posterior energy function*

$$\begin{aligned} E(f) &= -\ln(P(f|O)) \\ &= \sum_{p \in \mathcal{P}} V_p(f_p; o_p) + \sum_{(p,q) \in \mathcal{N}} V_{\{p,q\}}(f_p, f_q; o_p, o_q) \end{aligned} \tag{2.9}$$

Of course this energy can be rewritten as functions on the maximal cliques

$$E(f) = \sum_{c \in \mathcal{C}} V_c(f_p, f_q; o_p, o_q). \tag{2.10}$$

For more on MRFs see for example [22] and [21].

2.3 Graph Cuts

2.3.1 Problem Formulation

Consider a weighted graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ consisting of a set of n nodes $\mathcal{V}$ and a set of m edges $\mathcal{E}$ where each edge $e \in \mathcal{E}$ in the graph has a nonnegative weight (or cost) w_e. Define two special nodes, usually referred to as *terminal nodes* or source s and sink t nodes (there can be more than two terminal nodes but in this work we will consider the binary problem). A *cut* through the graph is as a subset of edges $C \subset \mathcal{E}$ such that in the graph $\langle \mathcal{V}, \mathcal{E} \backslash C \rangle$ the terminal nodes are separated, i.e. there are no path from s to t when the edges C are removed from the graph. The graph is partitioned into two completely separated graphs. The graph cut problem (also called min-cut/max-flow problem for reasons that will be clear in a moment) is to find the minimal cut, i.e. find the cut C through the graph $\mathcal{G}$ such that the sum of the cost of the edges that are cut is minimal

$$\arg\min_{C \subset \mathcal{E}} \sum_{e \in C} w_e. \tag{2.11}$$

According to one of the fundamental theorems of combinatorial optimization by Ford-Fulkerson the min-cut problem is dual to the max-flow problem. The max-flow problem is easily understood by considering the edges of the graph as pipes and the costs of the edges is the capacities of the pipes (i.e. the amount of flow they can facilitate) the problem is now stated as the maximum flow (of e.g. water) that can be pushed from the source to the sink. When the maximum flow is found some pipes will be saturated and the set of saturated pipes will be equal to the minimum cut. The realization that the min-cut problem and the max-flow problem are dual is important since the max-flow problem can be solved in polynomial time by different algorithms that iteratively push flow through the edges until the saturated state is achieved. For reference see for example [3].

2.3.2 Graph Cuts in Computer Vision

To reformulate an image as a graph $\mathcal{G}$, let each pixel/voxel p be a node. A neighborhood system $\mathcal{N}$ is defined by putting an edge (p, q) between each pair of neighboring voxels p and q. Define two extra nodes s and t and connect edges from s to each of p and from each p to t. The edges between the pixels are usually referred to as *n-links* (for neighborhood) and the edges between the pixels and the terminals s and t are called *t-links*. All of the edges are assigned weights w_e. For a small 3×3 example image the graph will look like Figure 2.1.

To build some intuition about this graph we consider a cut through the graph separating s and t. We define the set $S \subset \mathcal{P}$ as the set of nodes connected to s and the set $T \subset \mathcal{P}$ as the set of nodes connected to t. If a node p belongs

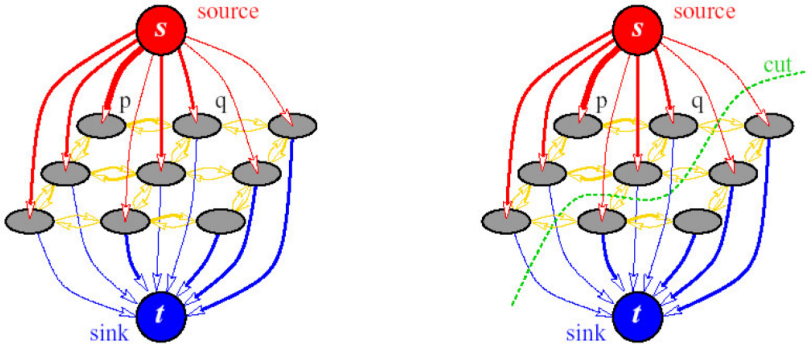

Figure 2.1: The grey nodes represent a 3 × 3 image. the nodes are connected to its neighbors and all pixels are connected both to the source node s and the sink node t. All edges have costs w_e. The cut parts the graph so that no path exists between s and t and the pixels are separated into two sets: the pixels that are connected to s and the pixels that are connected to t. The minimal cut is the cut that separates t and s and has the minimal cost. Figure is found in [3].

to the set S it will be labeled as $f_p = 1$ (foreground) and $f_p = 0$ (background) if it belongs to T. In fact we view the graph as describing a second degree pseudo-boolean function, i.e. a function from the boolean configurations of the nodes ($f_p \in \{0, 1\}$) to $\mathbb{R}$ that can be factorized into

$$E(f) = \sum_{\forall \{p,q\}} a(f_p, f_q). \tag{2.12}$$

To understand this we look at a small example 2.2. Each edge in the two-node graph has a weight. We want to cut the graph into two parts where the nodes connected to 0 will be labeled 0 and the nodes connected to 1 will be labeled 1. We have four possible configurations of this graph. If we want to label $f_p = 1$ and $f_q = 0$ the edges $w_{0,q}$, $w_{p,1}$ and $w_{p,q}$ are cut and the cost of this configuration hence is $a(1, 0) = w_{0,q} + w_{p,1} + w_{p,q}$. In the same way we define costs for the other configuration $a(0, 0)$, $a(0.1)$, $a(1, 1)$.

Now propose that se defined the costs a of the different configurations and we want to set the weights w so that the graph describes these costs. One formulation is

$$\begin{aligned} w_{p,0} &= a(1,0) - a(0,0) \\ w_{1,p} &= a(1,0) - a(1,1) \\ w_{0,p} &= a(1,0) + a(0,1) - a(0,0) - a(1,1) \end{aligned}$$

and setting the rest of the weights to 0. By this convention we can formulate a graph that describes any quadratic pseudo-boolean function where we have defined the cost of a configuration of each clique $\{p, q\}$ as $a_{\{p,q\}}(f_p, f_q)$. If we want the function to be easily optimized there is the criterion that all weights w in the graph must be non-negative. This is achieved if the following criterion on a holds (referred to as submodularity):

$$a_{\{p,q\}}(0,0) + a_{\{p,q\}}(1,1) \leq a_{\{p,q\}}(0,1)a_{\{p,q\}}(1,0), \quad \forall \{p,q\} \in \mathcal{C}. \tag{2.13}$$

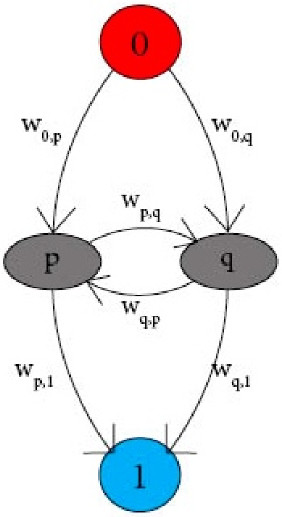

Figure 2.2: The grey nodes represent a small two node graph. Both nodes are connected to each other and the source (denoted as 0) and the sink (denoted as 1). All edges have defined weights w_e.

Now we note that the energy function describing the MRF (2.11) is a quadratic pseudo-boolean function. This of course means that the energy function can be represented by a graph and as long as it is submodular (which it turns out to be in our formulation) it can be formulated as graph with non-negative weights and the minimal cut of the graph corresponds to the configuration f that maximizes the posterior probability (2.8) of the MRF. For further information see [2].

Chapter 3

Method: Multi-atlas Segmentation

Multi-atlas segmentation is a method that has become very popular for many different medical applications, including segmenting brain and its internal structures, lungs, hearts and other internal abdominal organs [8].

An anatomical *atlas* is an image $I = \{i_p \mid p \in \mathcal{P}\}$ with corresponding manual labeling $\mathcal{L} = \{l_p \mid p \in \mathcal{P}\}$. In our case we have a CT-volume of a heart and a binary volume of equal size with the labeling done by an expert. If a voxel p in the binary volume has value 1 it means that $l_p = 1$ and i_p describes a part of the image that is inside the pericardium (foreground). If $l_p = 0$, i_p describes a part of the image that is outside of the pericardium (background). I and $\mathcal{L}$ together will be referred to as an atlas. An example slice of one of our atlases can be seen in Figure 3.1.

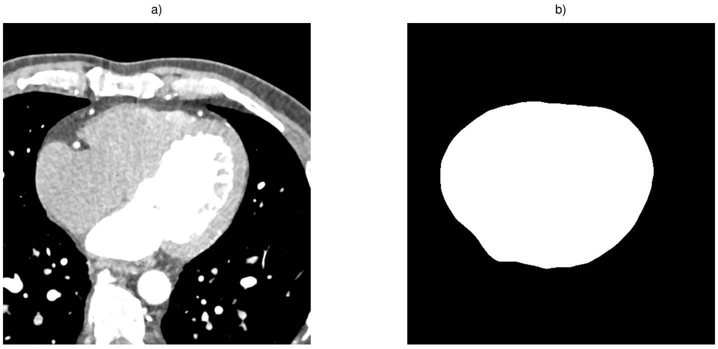

Figure 3.1: a) One slice of an CT-volume. b) Corresponding slice of the manual labeling. Together a) and b) form an atlas.

When given a new unlabeled target image I_t an atlas image I_s is registered to the target image and a transformation $\mathbf{T}$ is estimated. This transformation should align the source image onto the target image. By applying the same transformation on the labeled volume $\mathcal{L}_s$, the labels should also align onto the target image. This is called label propagation and can be used as a labeling (segmentation) of the voxels of the target image. The process of registering an atlas to a target image and segmenting the image through label propagation is called single-atlas segmentation.

Multi-atlas segmentation registers multiple atlases to the target image and then combines their segmentation labels. Because of the natural variation e.g. between different hearts, one single atlas might fail to accurately classify the target image. Using multiple independent classifiers and fusing their results might produce better results.

There are several ways of combining the labels into a final segmentation. The most straight forward is majority voting (also referred to as decision fusion or label voting). Each atlas labeling is considered a vote. The label of a voxel in the target image is selected as the label that a majority of the atlas segmentations agree on. Majority voting will be used to evaluate the multi-atlas segmentation based on the different registration techniques. But the final segmentation will be found by incorporating the multi-atlas votes into an MRF field and finding the maximum a posteriori configuration through graph cuts (explained in Chapter 4).

Multi-atlas segmentation is at its core multiple image registrations. There are a lot of proposed methods for solving the registration problem described in Section 2.1. In this work we choose to evaluate the performance of two different methods. One *feature based* and one *intensity based* method. The feature based methods in general differ from the intensity based methods in that they, instead of directly working with the intensity levels of the pixel/voxel, extract features that represent the information of the image on a higher level. The problem reduces to registering the extracted features in the images to each other.

Feature based registration methods are usually recommended when the images contain enough distinctive and easily detectable objects as is often the case in many areas of computer vision. In medical images on the other hand, the objects are often not as rich in detail. Features can be extracted but a large amount of them will be noise or mismatched between the images and hence result in a lot of outliers when the transformation between the images is estimated. Because of this drawback feature based methods are rarely used in medical applications, see [25].

However, since we are interested in making multiple registrations for each new patient for the multi-atlas segmentation and since ultimately this work is supposed to be used on a very large data set (in connection with the SCAPIS project), it is of value if we can make it less computationally demanding and feature based registration is in general much faster to compute. Therefore we evaluate two methods. One feature based method based on SURF and one intensity based method named Niftyreg.

For more about registration and the different methods see for example [25, 20].

3.1 Feature Based Registration

The feature based registration method that we use is based on SURF [1]. SURF has shown to outperform comparable methods not only in speed but also in robustness, i.e. it is less sensitive to noise [1]. This is valuable to us since we are dealing with a lot of outliers. To further try to cope with the outlier problem we use RANSAC, which is very robust to outliers, to estimate the transformation between the features obtained from SURF.

The registration uses SURF for feature detection and feature description, the matching criterion by Lowe to find correspondences between the detected features and RANSAC to estimate a rigid transformation between the two feature sets. Finally, a B-spline based transformation is used to map the correspondences even closer to each other.

3.1.1 SURF

Speeded-Up Robust Features (SURF) is a method for feature detection and description developed by Herbert Bay et al. in 2006 [1]. It was originally developed for images in two dimensions (and so will this explanation be) but the concept is easily expandable to 3D. It consists of two parts: interest point detection and interest point description.

Interest Point Detection

Interest points are points in the image that seem more like the center of a blob than points in its neighborhood. The detection of these points is based on approximation of the determinant of the Hessian matrix which describes the curvature of the image. The Hessian matrix $\mathcal{H}(\boldsymbol{x}, \sigma)$ of a point $\boldsymbol{x}$ at scale σ is given by

$$\mathcal{H}(\boldsymbol{x}, \sigma) = \left[\begin{array}{cc} L_{xx}(\boldsymbol{x}, \sigma) & L_{xy}(\boldsymbol{x}, \sigma) \\ L_{xy}(\boldsymbol{x}, \sigma) & L_{yy}(\boldsymbol{x}, \sigma) \end{array} \right] \tag{3.1}$$

where $L_{xx}(\boldsymbol{x}, \sigma)$ is the convolution of the Gaussian second order partial derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I at $\boldsymbol{x}$ and similarly for L_{xy} and L_{yy}. If the determinant is large, the blob response is high and if the blob response is higher than the blob response in a neighborhood (local maxima) the point is detected as an interest point. This detection is done for different scales σ with different sizes of the convolution filters to be able to detect blobs of different sizes. The scale space is divided into octaves (representing an scaling factor of 2) and each octave is subdivided into a constant number of scales.

The Hessian has been used before as a way to detect interest points but Bay et al. proposed to speed up these calculations by approximating the convolution filters L with simple box filters (see fig 3.2) which can be calculated very efficiently with use of the integral image. The element at $\boldsymbol{x} = (x, y)^\intercal$ in the integral image I_Σ relates to the original image I as

$$I_\Sigma(\boldsymbol{x}) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i, j). \tag{3.2}$$

This conversion allows for efficient computation of the sum of all pixel intensities in any upright rectangular area since, regardless of size, it only needs to evaluate four points in I_Σ (see fig 3.3).

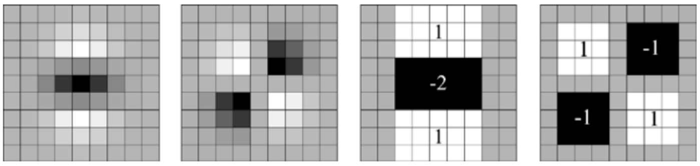

Figure 3.2: From left to right: the discretized second order partial derivative in y- (L_{yy}) and xy-direction (L_{xy}), respectively. And the box approximation of these filters.

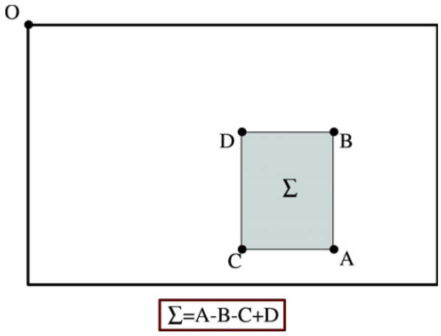

Figure 3.3: To evaluate the sum of all pixels Σ inside a box you only need to evaluate and sum four elements in the integral image I_Σ

Interest Point Description

To be able to successfully match an interest point in one image to the corresponding feature in another image, each interest point is described with a vector (a descriptor). The descriptor should be invariant to the transformations that are expected between one image and the other. If that is the case the descriptors of corresponding points in the different images would be similar and the euclidean distance between these descriptors would be small.

The description of the features uses information from the interest points neighborhood. Firstly, in order for the descriptor to be rotation invariant, a reproducible orientation for the image point is identified. This is done by calculating the Haar wavelet (see Figure 3.4) responses in x and y direction in a circular neighborhood around the point. Both the radius of the neighborhood and the size of the Haar wavelets are scaled according to in which scale the interest point was detected to make the descriptor scale invariant. The response from the wavelets in the x and y direction are then used to find a dominant orientation of the neighborhood.

To extract the descriptor, a square region centered at the interest point and oriented along the dominant orientation is constructed. The size of this region is again dependent of the scale. This region is divided into 4×4 subregions and for each of these subregions the x and y Haar wavelet responses are evaluated at the points of a 5×5 grid inside the subregion. The responses in the horisontal direction (perpendicular to the dominant direction) are here denoted d_x and the the responses in the vertical direction are denoted d_y, the

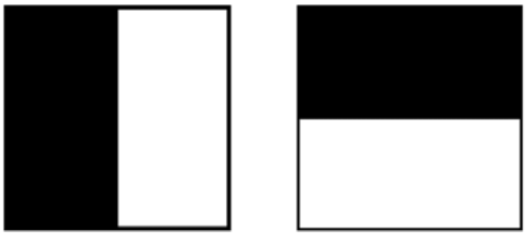

Figure 3.4: Haar wavelet filters in the x- and y directions. Dark areas has weight -1 and white has weight $+1$.

intensity structure of that subregion is described by a 4D-description vector $\boldsymbol{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Concatenating these from all subregions of the interest point you get a final feature descriptor of length 64.

3.1.2 Matching

For each image (the target and the source) a set of features are extracted. The next step is determining which features in the source set that are corresponding to which features in the target set. The most straight forward way would be to find the feature that is closest (in the descriptor space). However only choosing correspondences by which features have the most similar descriptors will result in a lot of bad matches. A more effective method (as shown by Lowe [13]) is to compare the closest neighbor to its second closest neighbor. Two features are matched if the ratio between the distance between a features closest neighbor and its second closest is below a certain threshold. I.e. if the descriptor vector is denoted $\boldsymbol{d}$, $\boldsymbol{d}_1$ is the closest feature to $\boldsymbol{d}$ and $\boldsymbol{d}_2$ is the second closest, a match is found if

$$\frac{|0\boldsymbol{d}-\boldsymbol{d}_1|0}{|0\boldsymbol{d}-\boldsymbol{d}_2|0} \leq \rho \tag{3.3}$$

where ρ is a parameter. In other words a match is found if a feature has a match that is considerably closer than any other match. The matches are in this work found by exhaustive search.

As mentioned earlier, medical images are not very rich in detail which results in there not being many distinct features. If the the matching threshold ρ is set to low we might end up with too few matches to be able to estimate a reliable transformation. Instead we set ρ high, resulting in a larger amount of matches but also more outliers.

3.1.3 RANSAC

Given the correspondences found from the matching in Section 3.1.2 we want to find a transformation that transforms the points in the source image onto the points in the target image. This is done using RANSAC, a method which is very insensitive to outliers.

Random Sample Consesus (or RANSAC) is a method developed in 1981 by Fischler and Bolles [9] for fitting a model to experimental data. Fischler and Bolles recognized that there are two types of errors present in experimental data, classification and measurement errors. Classification errors are outliers that are not captured by the model (e.g. when the detector incorrectly identifies a feature). Measurement errors occur when the detector correctly identifies a feature but slightly miscalculates one of its parameters (usually modeled by a normal distribution). Standard fitting techniques like least squares will fail since the gross classification errors will have a significantly larger effect than the measurement errors and the errors will not balance out.

RANSAC handles this by, instead of using all of the data for parameter estimation, randomly selects a minimum set of features and uses these to estimate the parameters of the transformation. Given these estimated parameters, find the set of features (the consensus set) that are within some error threshold of this model and therefore can be considered measurement errors. Iteratively redo these calculations for new randomly chosen initialization sets and choose the parameters of the model with the largest consensus set (i.e. the model with the most inliers).

We were trying to estimate a rigid transformation $\mathbf{T}(\boldsymbol{\theta})$ which has 6 parameters. The minimal random set hence consists of 6 correspondences (6 points $\boldsymbol{x} = (x_1, \dots, x_6)$ in the source image and their corresponding points $\boldsymbol{y} = (y_1, \dots, y_6)$ in the target image). The transformation relating these points to each other are easily computed explicitly so that

$$\boldsymbol{y} = \mathbf{T}(\boldsymbol{\theta}) \circ \boldsymbol{x}. \tag{3.4}$$

Now map all points in the source image with this transformation and count the number of points that are transformed within a threshold distance from its corresponding point in the target image (i.e. the number of correspondences in the consensus set).

3.1.4 B-spline registration

The correspondences found by RANSAC are mapped to each other non-rigidly using a package developed by Dirk-Jan Kroon available freely for MATLAB called B-spline Grid, Image and Point based Registration. It contains a point based B-spline registration based on the paper by Lee et al. [12]. It estimates a B-spline based free form deformation that maps corresponding points onto each other analogously with the intensity based free form deformation in the Niftyreg package that will be explained in Section 3.2.

3.2 Intensity Based Registration

Niftyreg is a freely available medical image registration package. It contains two programs for image registration that we are using in this work, reg aladin and Reg f3d. Reg aladin estimates an affine/rigid transformation and is based on block-matching. Reg f3d computes a free form transformation based on B-splines.

Reg aladin

The algorithm, on which Reg aladin is based, was presented in [16] and [17] by Ourselin et al. Basically, the source image is divided into blocks $\mathscr{B}'$ and each of these blocks is compared to blocks in the target image $\mathscr{B}$. The similarity between a block $\mathscr{B}'_{ab}$ centered at (a, b) in the source image and a block $\mathscr{B}_{uv}$ centered at (u, v) in the target image is measured using the normalized cross correlation of the blocks

$$C\left(\mathscr{B}_{ab}, \mathscr{B}_{uv}\right) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{\left(I_s(a+i, b+j) - \mu_{s_{(a,b)}}\right)\left(I_t(u+i, v+j) - \mu_{s_{(u,v)}}\right)}{\sigma_{s_{(a,b)}} \sigma_{t_{(u,v)}}} \tag{3.5}$$

where $\mu_{s_{(a,b)}}$ and $\sigma_{s_{(a,b)}}$ is the mean and standard deviation of the block $\mathscr{B}_{ab}$ in the source image and $\mu_{t_{(u,v)}}$ and $\sigma_{t_{(u,v)}}$ is mean and standard deviation of the block $\mathscr{B}_{uv}$ in the target image. To reduce computational time only the blocks with high variability ,which correspond to high contrast regions, are considered.

The block matching provides a set of corresponding points (the centers of the best matching blocks). Ourselin et al. estimates that 20% of the obtained matches are due to outliers and notes that the least squares approach to estimating the parameters $\boldsymbol{\theta}$ of the affine transformation $\mathbf{T}_{\text{aff}}$ is unsatisfactory. Instead they propose to use the L_1-estimator. That is find the transformation that minimizes the sum of the L_1 norm of the residuals $r_k = y_k - \mathbf{T}$, in other words, minimizes $\sum_k |r_k|$. Since it is a slower growing function than the L_2 norm it is less sensitive to outliers. Ultimately they ended up using the manhattan distance of the residual instead of the euclidean distance. Although the manhattan distance is dependent of the particular coordinate system Ourselin et al. observe that it yields slightly better results compared to the Euclidean distance.

The source image is transformed using the estimated transform and then the block matching starts over, iteratively updating the transform until the variation between the new transform and the previous one gets small enough or a set number of iterations. Because of the complexity of the algorithm it uses a multi-scale implementation where it first finds a rough transform at a coarse scale. When the a transform is found it continues but on a refined scale. The algorithm stops when the block size becomes so small that the information of the block is considered insufficient or after a set number of levels.

Reg f3d

Reg f3d is based on an algorithm presented by Rueckert et al. [18] and the implementation is by Modat et al. and described in [14]. The algorithm seeks to minimize a cost function $\mathcal{C}$ associated with the control points Φ of the B-spline grid (which were covered in Section 2.1.2)

$$\mathcal{C}(\Phi) = -\mathcal{C}_{similarity}(I_t, \mathbf{T}(I_s)) + \lambda \mathcal{C}_{smooth}(\mathbf{T}). \tag{3.6}$$

The first term measures the similarity between the target image I_t and the transformed source image I_s using normalized mutual information (NMI). Mutual information is a concept from information theory and measures how much of the information in one image A is present in the other image B. More specifically

$$\mathcal{C}_{similarity}(A, B) = H(A) + H(B) - H(A, B) \tag{3.7}$$

where $H(A)$ and $H(B)$ are the marginal entropy of A and B and $H(A, B)$ is the joint entropy. These are calculated by computing estimations of the histograms of the voxels in the images.

The second term describes the smoothness of the transform. Since, in general, we know that the transformation will be smooth we can penalize the transformation for bending to much by adding the bending energy of the transformation to the cost function

$$\begin{aligned}\mathcal{C}_{smooth} = \frac{1}{V}\int_0^X\int_0^Y\int_0^Z \Bigg[\left(\frac{\partial^2 \mathbf{T}}{\partial x^2}\right) + \left(\frac{\partial^2 \mathbf{T}}{\partial y^2}\right) + \left(\frac{\partial^2 \mathbf{T}}{\partial z^2}\right) + \\ + \left(\frac{\partial^2 \mathbf{T}}{\partial xy}\right) + \left(\frac{\partial^2 \mathbf{T}}{\partial xz}\right) + \left(\frac{\partial^2 \mathbf{T}}{\partial yz}\right)\Bigg] dxdydz\end{aligned} \tag{3.8}$$

The transformation is optimized with a multi-scale approach where the spacing between the control points initially are large (i.e. there are fewer parameters for the transformation and the possible transformations are on a more global level). To find the optimal position for the control points the conjugate gradient descent method is used where the gradient of the cost function is estimated and iteratively a step proportional to the length of the gradient vector is taken downwards. When a local optima is found (the length of the gradient gets below a threshold) more control points are added which allows for a more local transformation and the optimization starts over.

Chapter 4

Method: MRF Segmentation

There are many different segmentation techniques. This work is focused around multi-atlas segmentation which is based on multiple image registrations (which was covered in chapter 3)

However, the multi-atlas segmentation will never be perfect. Especially the feature based method, which works with the images on a higher level (registering extracted features to each other) and with data with a lot of outliers, will sometimes produce errors which are apparent when taking into consideration the intensity information (e.g. when lung tissue is classified as inside the pericardium). Therefore we constructed an MRF into which we could incorporate both the atlas registrations and the intensity information. Another advantage of the MRF formulation (apart from being very flexible) is that it can be globally optimized in polynomial time using graph cuts.

4.1 MRF Formulation

What is usually done (for e.g. in [4]) is rewriting the posterior probability using Bayes' Theorem as

$$\Pr(f|\mathcal{O}) = \frac{\Pr(\mathcal{O}|f)\Pr(f)}{\Pr(\mathcal{O})} \propto \Pr(\mathcal{O}|f)\Pr(f) \tag{4.1}$$

where the likelihood $\Pr(\mathcal{O}|f)$ is assumed to factorize as

$$\Pr(\mathcal{O}|f) = \prod_{p\in\mathcal{P}} \Pr(o_p|f_p) \tag{4.2}$$

i.e. each $\Pr(o_p|f_p)$ is independent of the rest of the field. Here f on the other hand is assumed to be a Markov field and according to Hammersley-Clifford (see Section 2.2) it factorizes to

$$\Pr(f) \propto \exp\left(-\sum_{\{p,q\}\in\mathcal{C}} V_{p,q}(f_p, f_q)\right). \tag{4.3}$$

The *pairwise clique potential* $V_{\{p,q\}}$ is usually set to

$$V_{p,q}(f_p, f_q) = \begin{cases} \lambda, & f_p \neq f_q \\ 0, & f_p = f_q \end{cases} \tag{4.4}$$

where λ is a constant which regularizes the segmentation by penalizing making the boundary of the segmentation too long.

Of course, saying that f is a Markov random field, and that the likelihood $\Pr(\mathcal{O}|f)$ is not, is only a trick to divided the problem into parts that are easier to understand intuitively. We can rewrite $\Pr(f|\mathcal{O})$ into a product of functions $a_{\{p,q\}}(f_p, f_q)$ on the maximal cliques $\{p,q\}$ by collecting (4.2), (4.3) and (4.4) into

$$\Pr(f|\mathcal{O}) \propto \exp\left(-\sum_{\{p,q\}\in\mathcal{C}} a_{\{p,q\}}(f_p, f_q)\right), \tag{4.5}$$

where

$$a_{\{p,q\}}(f_p, f_q) = \begin{cases} -\ln(\Pr(o_p, o_q|f_p, f_q)), & f_p = f_q \\ -\ln(\Pr(o_p, o_q|f_p, f_q)) + \lambda, & f_p \neq f_q \end{cases}. \tag{4.6}$$

(4.5) and (4.6) represent the same MRF as above (apart from a constant that is dependent on the number of neighbors to each pixel p) and both are maximized by the same configuration f. This formulation will be referred to as the *original* formulation.

4.1.1 Expanding the formulation

Formulating the probability as in the original formulation has two main advantages: it is an easy formulation and it regularizes the segmentation in a predictable way which means that you have some control over the shape of the segmentation. The main disadvantage is that, for it to work satisfactory, the boundary cannot pass through an area where there is an overlap between the density functions $\Pr(o_p|F_p = 0)$ and $\Pr(o_p|F_p = 1)$. If it is, the location of the boundary will be ambiguous and the regularization will fall back to the shortest path.

Segmenting the pericardium suffers from this drawback. If the pericardium is located directly between the lung cavity and the myocardium the classification is straight forward since the air is clearly not inside and the probability of the soft tissue that constitutes the myocardium is rather likely to be inside. In theses areas the segmentation using the MRF will show a clear improvement over the multi-atlas segmentation. But the pericardium often passes through fat tissue. Fat is likely to be found both outside and inside of the pericardium and in these areas the multi-atlas segmentation is not likely to be improved by the MRF.

But there are some indicators of where the boundary should be since the pericardium is not completely invisible. We want to exploit this fact by making the cost of the boundary dependent on these observations and we propose to find a simple and intuitive formulation that can guide the boundary in areas of ambiguity. We have been experimenting with a lot of different functions for the cost of the boundary which all have in common that they are a function of the negative logarithm of some probability that is dependent on the observations. We tried a large set of formulations but found the following to be not to complicated and at the same time work well on our training sets.

The Hammersley-Clifford Theorem (see Section 2.2) states that

$$\Pr(f|\mathcal{O}) \propto \exp\left(-\sum_{\{p,q\}\in\mathcal{C}} V_{\{p,q\}}(f_p, f_q)\right).$$

Since the probability factorizes over the cliques we can split the formulation onto the sets $f^{=} = \{\{p,q\} \in \mathcal{N} \mid f_p = f_q\}$ and $f^{\neq} = \{\{p,q\} \in \mathcal{N} \mid f_p \neq f_q\}$ and we get

$$\begin{aligned}\Pr(f|\mathcal{O}) &\propto \exp\left(-\sum_{\{p,q\}\in f^{=}} V_{\{p,q\}}(f_p, f_q)\right) \exp\left(-\sum_{\{p,q\}\in f^{\neq}} V_{\{p,q\}}(f_p, f_q)\right) \\ &\propto \Pr(f^{=}|\mathcal{O}) \Pr(f^{\neq}|\mathcal{O}). \end{aligned} \tag{4.7}$$

$\Pr(f^{=}|\mathcal{O})$ will be written straight forwardly as

$$\Pr(f^{=}|\mathcal{O}) \propto \prod_{\{p,q\}\in f^{=}} \Pr(f_p, f_q|o_p, o_q) \tag{4.8}$$

which corresponds to a potential function

$$V_{\{p,q\}}(f_p, f_q) = -\ln(\Pr(f_p, f_q|o_p, o_q)), \quad f_p = f_q.$$

Here it is worth noting that all probabilities regarding specific pixels are considered independent of all other pixels. For example we have

$$\Pr(f_p, f_q|o_p, o_q) = \Pr(f_p|o_p)\Pr(f_p|o_p) = \frac{\Pr(o_p|f_p)\Pr(f_p)}{\Pr(o_p)} \frac{\Pr(o_q|f_q)\Pr(f_q)}{\Pr(o_q)}. \tag{4.9}$$

$\Pr(f^{\neq}|\mathcal{O})$ describes the probability of the boundary and we put a little more thought into formulating this probability. We want to construct a modified probability distribution $g(f_p, f_q; o_p, o_q)$ that gives us some control over the cost of the boundary.

In the original formulation g was simply formulated as

$$g(f_p, f_q; o_p, o_q) = \Pr(f_p, f_q|o_p, o_q).$$

But now we see that g should also reflect the probability of f_p and f_q belonging to $f^{\neq}$. We let $\mathcal{B}$ be the set of pixels that are on the boundary, i.e. $\mathcal{B} = \{p \in \mathcal{P} \mid p \in f^{\neq}\}$. A more natural formulation of g then is

$$g(f_p, f_q; o_p, o_q) = \Pr(f_p, p \in (B)|o_p)\Pr(f_q, q \in \mathcal{B}|o_p) \tag{4.10}$$

and we can rewrite this as

$$g(f_p, f_q; o_p, o_q) = \Pr(p \in \mathcal{B}|f_p, o_p)\Pr(q \in \mathcal{B}|f_q, o_p)\Pr(f_p, f_q|o_p, o_q).$$

We denote $g_b(f_p, f_q; o_p, o_q) = \Pr(p \in (B)|f_p, o_p)\Pr(q \in \mathcal{B}|f_q, o_p)$ which gives us

$$\begin{aligned}\Pr(f^{=}|\mathcal{O}) &\propto \exp\left(-\sum_{\{p,q\}\in f^{\neq}} -\ln(g(f_p, f_q; o_p, o_q))\right) \\ &= \exp\left(-\sum_{\{p,q\}\in f^{\neq}} (-\ln(\Pr(f_p, f_q|o_p, o_q)) - \ln(g_b(f_p, f_q; o_p, o_q)))\right). \end{aligned} \tag{4.11}$$

Now we can collect (4.8) and (4.11) into (4.7) and get

$$\begin{aligned}\Pr(f|\mathcal{O}) &\propto \Pr(f^{=}|\mathcal{O})\Pr(f^{\neq}|\mathcal{O}) \\ &= \exp\left(\sum_{\{p,q\}\in\mathcal{C}} -\ln(\Pr(f_p, f_q|o_p, o_q))\right)\Pr(\mathcal{B}|f,\mathcal{O}) \qquad (4.12)\end{aligned}$$

where

$$\Pr(\mathcal{B}|f,\mathcal{O}) = \exp\left(\sum_{\{p,q\}\in f^{\neq}} -\ln(g_b(f_p, f_q; o_p, o_q))\right)$$

.

We note that if g_b equals a constant we have the original formulation which simply regularizes the segmentation. If we estimate the probabilities in g_b we get a cost of the boundary that in a natural way reflects the probability of the boundary given the observations.

We further introduce two parameters λ_1 and λ_2 to control the effect the boundary cost will have on the segmentation

$$\Pr(\mathcal{B}|f,\mathcal{O}) = \exp\left(\sum_{\{p,q\}\in f^{\neq}} -\lambda_1\ln(g_b(f_p, f_q; o_p, o_q)) + \lambda_2\right) \qquad (4.13)$$

Finally, we set

$$-\lambda_1\ln(g_b(f_p, f_q; o_p, o_q) + \lambda_2 = \lambda_3, \quad \forall o_p, o_q \notin \text{fat} \qquad (4.14)$$

where λ_3 is some constant. This is done to isolate the peak in g_b which actually corresponds to the intensities found on the pericardium when it is visible. If the observations tell us that the current p and q are fat we will look for the most probable path through the fat. If p or q are not fat, g_b is set to a constant and we are instead falling back to regular regularization. This is motivated by the fact that it is mostly in fatty areas that this effect is needed. This also removes some peaks in g_b that only serve to confuse the segmentation. In practice we define this fatty peak as intensities between -120 HU and 40 HU. These intensities correspond to intensities that are a little lighter than dark fat and a little darker that soft tissue. The intensity of the pericardium when it is actually visible is somewhere around -35 HU which means that it is in the center of this window.

Summarizing all this we formulate the MRF as

$$\Pr(f|\mathcal{O}) \propto \exp\left(-\sum_{\{p,q\}\in\mathcal{N}} V_{\{p,q\}}(f_p, f_q)\right) \qquad (4.15)$$

where

$$V_{\{p,q\}}(f_p, f_q) = \begin{cases} -\ln(\Pr(f_p, f_q|o_p, o_q)), & f_p = f_q \\ -\ln(\Pr(f_p, f_q|o_p, o_q)) - \lambda_1\ln(g_b(f_p, f_q; o_p, o_q)) + \lambda_2, & f_p \neq f_q \end{cases}$$

with $g_b(f_p, f_q; o_p, o_q) = \Pr(p \in \mathcal{B}|f_p, o_p)\Pr(q \in \mathcal{B}|f_q, o_q)$ and putting

$$-\lambda_1\ln(g_b(f_p, f_q; o_p, o_q) + \lambda_2 = \lambda_3, \quad \forall o_p, o_q \notin \text{fat}.$$

The configuration f that maximizes the posterior probability (4.15) is the same as the configuration that minimizes the energy function

$$E(f) = \sum_{\{p,q\}\in\mathcal{C}} V_{\{p,q\}}(f_p, f_q). \tag{4.16}$$

4.2 Maxflow

By constructing a graph as explained in Section 2.3.2, minimizing the energy function (4.16) is the same as finding the minimum cut (or the maximum flow) through the graph. We used the maxflow algorithm by Boykov and Kolmogorov [3] which is a widely used method in computer vision. The algorithm is based on *augmenting paths* meaning that it pushes flow through the graph iteratively until no more flow can be pushed. Some of the edges will be saturated and the set of saturated edges will be the same as the set of edges in the minimal cut. Boykov and Kolmogorov's algorithm is a variant of this algorithm which is developed to be more efficient on the grid like graphs that often arise in computer vision.

4.3 Training

The algorithm was evaluated by cross validation since we only had 10 labeled volumes. This means that we had to train 10 instances of our method. One volume was chosen as validation set V and the other 9 as the training set T. For each volume in the training set all 8 remaining volumes where registered to this volume resulting in a vote map $\mathcal{M} = \{m_p \mid p \in \mathcal{P}\}$ where each voxel p has a number of votes $m_p \in \{0, \dots, 8\}$.

The features for the observations o_p are chosen simply as the pixel intensity at the given pixel i_p and the number of votes m_p from the multi-atlas registrations. Given the vote map $\mathcal{M}$ the labeling $\mathcal{L}$ and the intensities I from all the 9 volumes in the training set the probabilities $\Pr(i_p, m_p|f_p)$, $\Pr(i_p, m_p|p \in \mathcal{B}$, $\Pr(f_p)$ and $\Pr(f_p, \mathcal{B})$ are easily estimated and we have

$$\Pr(f_p|o_p) = \Pr(f_p|i_p, m_p) = \frac{\Pr(i_p, m_p|f_p)\Pr(f_p)}{\Pr(i_p, m_p)} \tag{4.17}$$

and

$$\Pr(p \in \mathcal{B}|f_p, o_p) = \Pr(p \in \mathcal{B}|f_p, i_p, m_p) = \frac{\Pr(i_p, m_p|f_p, p \in \mathcal{B})\Pr(f_p, \mathcal{B})}{\Pr(i_p, m_p|f_p)\Pr(f_p)}. \tag{4.18}$$

In practice we need to put a threshold on $\Pr(i_p, m_p|f_p)$ and $\Pr(i_p, m_p|f_p, p \in \mathcal{B})$ not allowing the probabilities to be zero. In other words we put a probability on unseen observations.

The probabilities were trained on vote maps consisting on votes from 8 separate registrations where m_p can take one of 9 values between 0 and 8, However, when the method will be validated on the validation set (the one atlas that is not in the training set) 9 atlases will have been registered to it. The probability of a voxel p with $m_p = 9$ is zero from the training phase since there only was 8 atlases voting. This is solved by interpolating the probability distribution to 10 linearly spaced points between 0 and 8 using linear interpolation.

It remained to estimate the parameters of the model $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$. The parameters were estimated as the parameters that maximized the mean Jaccard index when using the model on the training set

$$\arg\min_{\boldsymbol{\lambda}} \left(\frac{1}{9} \sum_{i \in T}^{|T|} \text{Jaccard}(\mathcal{L}_i^*(\boldsymbol{\lambda}), \mathcal{L}_i) \right). \tag{4.19}$$

We proposed to to optimize the parameters using a simple greedy approach where each λ_i was optimized separately using golden section search. This was not as straight forward as expected. Instead we found a reasonable initialization for λ_1 which was then kept fixed. Then λ_2 and λ_3 was optimized using the above method. The impression was that (4.19) varied slowly in the parameter space and was modestly dependent on which test set we used which meant that we could have kept λ_2 and λ_3 constant as well.

Chapter 5

Implementation and Evaluation Details

5.1 Registration

The registration methods were evaluated by registering all atlases to each other and computing the Jaccard index between the target atlas and the registered atlas. Since there were 10 atlases we obtained 90 measurements for each registering method (the registration methods were not symmetric) from which we calculated the mean and standard deviation.

5.1.1 Affine Initialization

The affine registration is used as an initialization for the non-rigid free form registration. For the affine initialization we evaluated two methods, the feature based using SURF and RANSAC and the intensity based using Reg aladin from the Niftyreg package.

Feature Based Registration

The features where extracted using 3 different octaves and 3 scales per octave from the target and the source image. Since the source image is part of an atlas a region of interest in the source image is known. We only used the interest points in the source image that are within a distance of 20 mm from the boundary of the manual labeling of the atlas, i.e. the pericardium. The mask corresponding to this region in the source image will be referred to as the *source mask*. The points where matched between the images using Lowe's matching criterion with the matching parameter ρ set to 0.95. With the best correspondences found between the interest points present inside of the source mask and all the interest points of the target image the optimal rigid affine transformation between the images were estimated using RANSAC. The definition of an inlier was set to an euclidian threshold of 10 mm, i.e. if an interest point $\boldsymbol{x}$ in the source image is mapped by the estimated transformation $\mathbf{T}$ to within 10 mm of its corresponding interest point $\boldsymbol{y}$ in the target image it is considered an inlier. The transformation estimation using 6 randomly sampled points and the computation of number of inliers of that transformation was iterated one million times and the transformation with the largest amount of inliers was chosen.

Intensity Based Method

Reg aladin was used to estimate an affine transformation between the source and the target image. It was run with default values of the parameters, i.e. 5 iterations for the transformation estimation on each level and 3 levels. Only the 50% of the blocks that have highest variability are used and when estimating the transformation 50% of the matches are considered outliers. In the same manner as for the feature registration a mask is used for the source image so that only the region within 20 mm of the pericardium is registered to the target image.

5.1.2 Free Form Registration

Since the feature based rigid registration proved to be more robust than the intensity based method we used that registration as an initialization for the nonrigid deformations. For each target image, the affine transformation was already estimated from all the other 9 atlases and by label propagation we had 9 guesses of where the pericardium was in the target image. Using these guesses we constructed a region of interest in the target image, i.e. a region in which the boundary of the mask most likely was present, referred to as a *target mask*. It was constructed by normalizing the vote map (dividing by $\#votes$) and blurring with a gaussian kernel with $\sigma = \sqrt{5}\,mm$. The target mask was then defined as voxels with values between 10^{-7} and $1 - 10^{-7}$ (see figure 5.1).

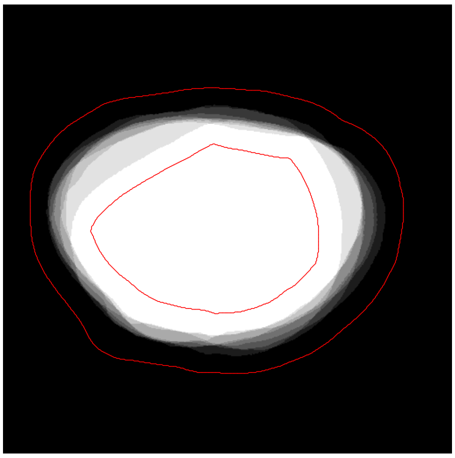

Figure 5.1: A slice of a vote map constructed from 9 rigid feature based atlas registrations. The vote map was blurred with a Guassian kernel to extract a target mask (the boundary of which are shown in red).

Feature Based Registration

All interest points in the target image that were not in the target mask were removed and the points were rematched between the points inside of the source and the target masks. By this procedure a lot of outlier matches were removed. Again RANSAC was used to find an inlier set that was consistent with a rigid transformation. Assuming that the nonrigid deformations were small enough these inliers were considered inliers also for the free form deformation and a transformation was estimated using the MATLAB package mentioned in Section 3.1.4 mapping the interest points onto each other using a B-spline based transfomation.

Intensity Based Registration

For the intensity based method the affine transform obtained earlier was used as an initialization which meant that the source image was transformed using the affine transformation before the images was registered using the free from registration. `Reg f3d` has the ability of specifying a target mask which means that we tested two methods. Firstly, the image specified by the source mask was registered to the whole target image and secondly the same thing was done but also specifying the target mask. The registration without a specified target mask worked a little better which meant that we only only evaluated theses registrations. Default values where used for the parameters: 64 bins for the joint histogram, a final grid spacing of 5 voxels, the weight of the bending energy penalty term was 0.005 and the optimization was done over 3 levels with a maximum of 300 iterations per level.

5.2 Multi-atlas Segmentation

To each target image (in the set of 10 atlases) all remaining 9 atlases was registered and the labels where combined using majority voting. The final segmentation was evaluated by measuring the Jaccard index between the segmentation and the manual labeling of the target image. This was done for the multi-atlas segmentation obtained using registrations based on Niftyreg and based on features.

5.3 MRF Segmentation

5.3.1 Graph Cuts

To make the inference of the minimal cut less computationally demanding we extracted the part of the target image in which the boundary (the cut) most likely would be present. This was done in the same manner as when constructing the target mask for the non rigid registrations. The vote map constructed from the atlas registrations was normalized (divided by $\#voters$) and blurred with a gaussian kernel with $\sigma = \sqrt{5}\,mm$. The target area was then defined as all voxels that have intensities between 10^{-7} and $1 - 10^{-7}$ (again see Figure 5.1).

2D histograms describing the occurrences of intensities and number of votes were computed from this target area and used to compute the prior probabilities of the MRF. The possible intensities in a CT-volume are $i_p \in [-1024, 3072]$. We divided these into 1024 bins and since we had 9 different numbers of votes to each image in the training set, the histograms was of size 1024 × 9. A typical histogram can be seen in Figure 5.2

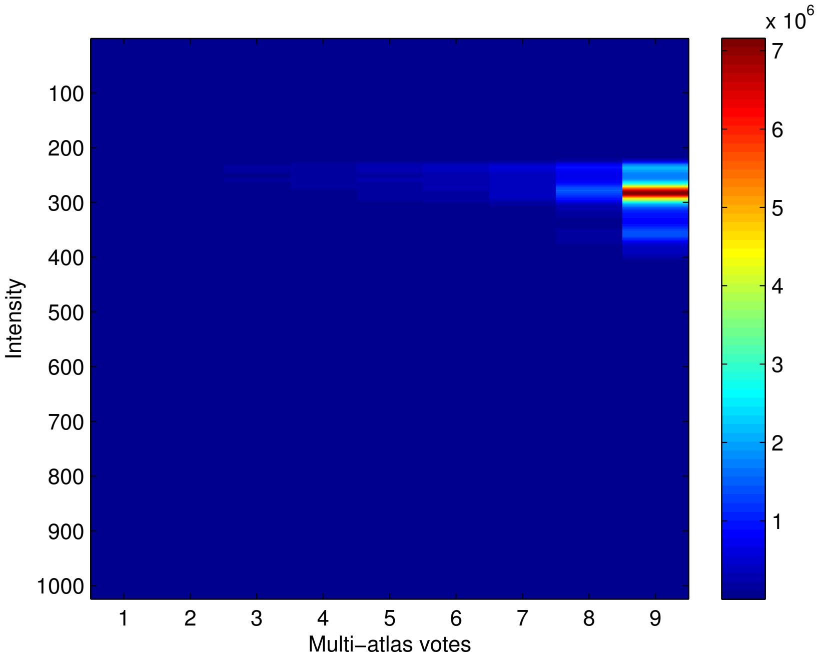

Figure 5.2: A typical histogram showing number of voxels in the training set with number of multi-atlas votes on the x-axis and intensity bins on the y-axis.

Parameters

For both the feature based and the intensity based MRF an initialization of $\lambda_1 = 20$ proved reasonable. The parameter λ_2 was then optimized on the interval $[-70, -40]$ and λ_3 on [100, 140]. The optimizations were done using golden section search assuming that the optimal value was found after 6 iterations.

Chapter 6

Results

The results will be presented in four sections. Firstly, the results covering the different registration methods are presented together with the multi-atlas segmentations. Thereafter we present the results we obtain by instead of majority voting incorporating the multi-atlas vote map into the MRF. Results are presented as MEAN±STD Jaccard index unless otherwise stated. We show the runtime of our algorithm and lastly we show how well these segmentations measure the actual fat inside of the pericardium.

6.1 Registration and Multi-atlas Segmentation

All evaluations of the different registration were evaluated by the Jaccard index of the single atlas segmentation. The label of the atlas was propagated onto the target image and the Jaccard index between the propagated segmentation and the gold standard was measured.

Firstly the rigid affine registrations using features (denoted F) and the affine registrations using Niftyreg (denoted as N) where considered. After registering all atlases to each other we found that the registration accuracy of Niftyreg was 0.75 ± 0.11 whereas the accuracy of the feature based method was 0.83 ± 0.04. On the basis of these results we chose to use the feature based registration as initialization for all the nonrigid registrations.

The non-rigid registrations are denoted NF (for nonrigid registration using Niftyreg and feature based initialization) and FF (for feature based nonrigid registration using feature based initialization). The nonrigid registrations were then combined into a multi-atlas segmentation through majority voting. The multi-atlas method based on NF is denoted multNF and the one based on FF multFF. The results of all of these methods are shown in Table 6.1.

6.2 MRF Segmentation

To improve on the atlas segmentation where there was ambiguity we proposed to incorporate the votes into an MRF together with the intensity information. We evaluate both the original formulation which only has regularization (denoted regMRF) and the extended formulation with a data dependent boundary cost (denoted extMRF). Table 6.2 shows the results of the MRF methods applied to the multi-atlas votes based on FF registrations and NF registrations.

Method	Jaccard index (MEAN±STD)
F	0.8149 ± 0.0384
N	0.7403 ± 0.1090
FF	0.8788 ± 0.0313
NF	0.9061 ± 0.0239
multFF	0.9241 ± 0.0180
multNF	0.9398 ± 0.0165

Table 6.1: The mean and standard deviation of the single-atlas segmentations for method N, F, NF, FF and of the multi-atlas segmentation for method multNF and multFF after the results of registering all 10 atlases to each other were collected.

Method	FF	NF
mult only	0.9241 ± 0.0180	0.9398 ± 0.0165
mult+regMRF	0.9321 ± 0.0181	0.9419 ± 0.0182
mult+extMRF	0.9379 ± 0.0186	0.9440 ± 0.0186

Table 6.2: The mean and standard deviation of the MRF methods applied to the multi-atlas votes constructed from the FF and NF registrations respectively.

6.3 Runtimes

The runtime of the different parts of the algorithm are presented in Table 6.3. The runtimes were evaluated with an Intel Core i7-43930k @3.40GHz with 6 cores and 12 threads. The feature detector and the Niftyreg methods were implemented to make use of all threads. The maxflow algorithm used 6 threads. The feature matching algorithm and the feature registration algorithms only utilized 1 thread in their current implementations.

Method	Time [s]
Feature detection	8.7
Feature matching	5.4
Feature rigid registration	36.3
Feature nonrigid registration	27.4
Maxflow	114.7
Niftyreg affine registration	678.8
Niftyreg nonrigid registration	1253.1

Table 6.3: Mean runtimes for the the different parts of our algorithm and for the Niftyreg based methods.

For constructing a final segmentation (using 10 atlases), feature detection and maxflow were run once on the target image and the rest of the methods were run 10 times. This collects into a total running time of about 13.6 minutes for multFF+extMRF compared to about 5.4 hours if using Niftyreg for both affine initialization and nonrigid registration. When using feature based initialization this reduces to 3.6 hours (multNF).

6.4 Fat Estimation

Using the final segmentations from multFF+extMRF and multNF+extMRF we measured the fat volume through thresholding (as explained in Section 1.1). The results are presented in Table 6.4 and 6.5.

	FF	
Method	Fat difference [ml]	Fat difference [%]
mult only	3.6130 ± 19.1113	6.17 ± 20.96
mult+regMRF	-0.2344 ± 20.0538	2.47 ± 20.17
mult+extMRF	-6.9403 ± 15.6680	-5.77 ± 14.24

Table 6.4: The mean and standard deviation of the MRF methods applied to the multi-atlas votes constructed from the FF registrations.

	NF	
Method	Fat difference [ml]	Fat difference [%]
mult only	-6.1516 ± 13.4166	-6.04 ± 12.25
mult+regMRF	-4.7326 ± 14.4616	-4.13 ± 12.62
mult+extMRF	-8.1224 ± 11.6586	-8.23 ± 8.85

Table 6.5: The mean and standard deviation of the MRF methods applied to the multi-atlas votes constructed from the NF registrations

Chapter 7

Discussion

7.1 Registrations

When evaluating the different parts of the registration we found that the feature based initializations were considerably more robust than the Niftyreg affine registrations. This might partly be explained by the fact the we only allow for rigid transformation when using this method which prevents it from making outrageous estimations of the transformation. But the huge disparity (0.8149 ± 0.0384 for the feature based compared to 0.7403 ± 0.1090) leads us to the conclusion that the feature based method is the superior method for initialization of the registration.

For the nonrigid registration we used B-spline interpolation for both Niftyreg and SURF. But the feature based method (FF) used an simple algorithm where the correspondences in the consensus set from RANSAC where mapped directly on top of each other. No regard was taken to the fact that many of these correspondences should be considered as outliers for the estimation of the B-spline transformation. Still this method considerably improves the rigid initialization (again see Table 6.1) but as expected the algorithm used by Niftyreg (NF) produces better results. We based all of the nonrigid registrations on the feature based registrations.

This allows for a fair comparison between the nonrigid feature based method and nonrigid intensity based method where the intensity based proved superior. What is not clear is what results we would have gotten if we used the intensity based affine registration as initialization for the nonrigid intensity based registration method. Since the intensity based initialization was less robust than the one we used this would probably have resulted in worse results for the method based on Niftyreg. This would have been a fairer comparison between our method and the intensity based method and would be a test for future work.

After combining the single-atlas registrations into a multi-atlas segmentation through majority voting (multFF and multNF) we considerably improve the robustness and accuracy of the algorithm. The Jaccard index was 0.9241 ± 0.0180 for the feature based and 0.9398 ± 0.0165 for Niftyreg with feature based initialization (Table 6.1).

7.2 MRF

The MRF further improves the accuracy of the segmentation as is clear from Table 6.2. The extended formulation of the MRF (MRFext) does provide slightly better results than the MRF formulation that only uses regularization. But since this improvement mostly is in fatty areas we also see that although it is a small improvement measured in Jaccard index, it has a nice effect on the robustness of the fat volume estimation. The standard deviation is lowered by 6.7 percentage points for the FF fat estimation and 3.4 percentage points for the NF fat estimation (see Tables 6.4 and 6.5). We also note that with the MRFext formulation the algorithm consistently underestimates the fat volume. This is an effect inherent to graph cuts since putting a cost on the boundary means that a short boundary is a cheap boundary. The algorithms we used for fat estimation where optimized for Jaccard index. By instead optimizing for fat estimation we would potentially achieve better results.

Especially we note that our method, with feature based multi-atlas segmentation coupled with a data dependent MRF, produces comparable results to using standard multi-atlas based segmentation based on Niftyreg (0.9379 ± 0.0186 compared to 0.9398 ± 0.0165) and at a significantly lower computational cost.

7.3 Computational Efficiency

Where the multi-atlas segmentation using Niftyreg would take 5.4 hours to compute using a standard computer, our method takes 13.6 minutes. This is of course a huge gain if this algorithm is to be used on a database consisting of thousands of images where the intensity based method in-fact would prove untractable.

We have not been experimenting much with the parameters regarding the the feature registration. For now 1 000 000 iterations were used by RANSAC but especially for the affine initialization this many iterations might be a lot more than needed. We tried a couple of registrations using 500 000 iterations which did not seem to affect the success of the registrations and of course cuts the running times in half. More work would be needed to evaluate and find the optimal way to handle the number of iterations for speed and accuracy.

In this work the registration algorithm only uses one processor thread. If it was implemented in parallel (as the Niftyreg baseline) it would potentially have been 12 times faster reducing the registration time to a few seconds. Also, the runtime of the maxflow algorithm would be greatly reduced by constructing a much smaller target mask (as explained in Section 5.3.1) and thereby a much smaller graph. This is motivated by the fact that the segmentation boundary is unlikely and therefore never present where the number of votes are too low or too high. This is especially true where there are no votes and where all the atlases agree which constitutes a large part of the graph in the implementation used in this work.

7.4 Comparison to Related Work

The methods proposed by [19] and [7] use different data sets with CT images without contrast. This is arguably a slightly harder problem since the different anatomical regions of the heart are not as easy to differentiate. They report a Dice similarity index for their segmentations of 0.89 ± 0.03 and 0.89 ± 0.01 respectively. The Dice similarity index is similar to Jaccard but it can be shown that

$$\mathrm{Dice}(A, B) \leq \mathrm{Jaccard}(A, B)$$

Measured in Dice similarity the result of our algorithm (multFF+MRFext) is 0.9678±0.0100. The conclusion we can draw from this fact is that CTA (i.e. contrast enhanced CT imaging) together with our method provides the best results for fully automatic pericardium segmentation known to the author. But since not all patients can undergo CTA-scans, future work would include evaluating this method on non-contrast CT images.

7.5 Review of Algorithm

Figure 7.1 shows an example of a successful segmentation (in fact a slice of a successful 3D segmentation). The MRF has the advantage that it directly corrects not allowed segmentations (as in Figure 7.2). In areas where the pericardium is visible it also has the ability to fit the boundary onto the true pericardium (Figure 7.3). When the pericardium is not visible or if the multi-atlas estimation is not good enough this does not work and the segmentation will fail to represent the boundary accurately (Figure 7.4).

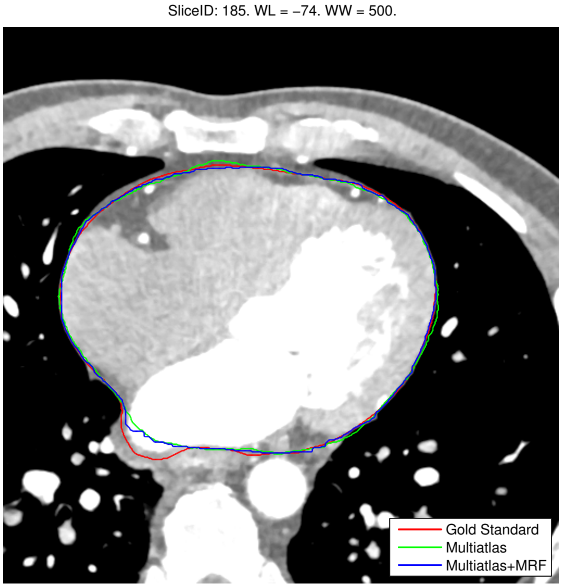

Figure 7.1: An example of a part of a good segmentation. The red boundary is the boundary of the gold standard. The green line represents the boundary of the segmentation estimated by the feature based multi-atlas segmentation. The blue line corresponds to the same feature based multi-atlas segmentation incorporated into the MRF.

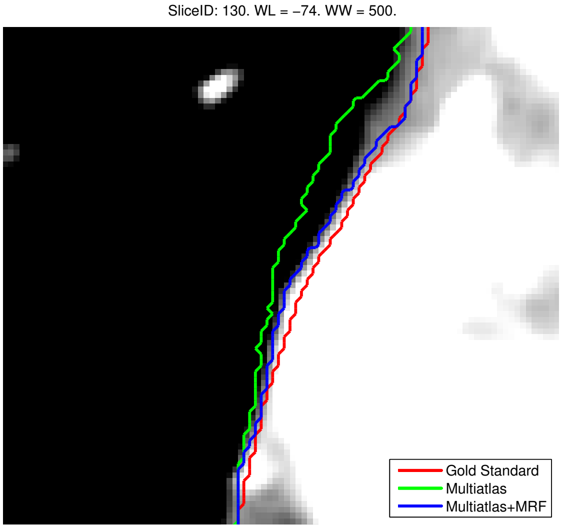

Figure 7.2: Here the MRF corrects the segmentation that is wrongly estimated to being in the lung cavity. Red is gold standard, green multi atlas segmentation and blue is the segmentation also using the MRF.

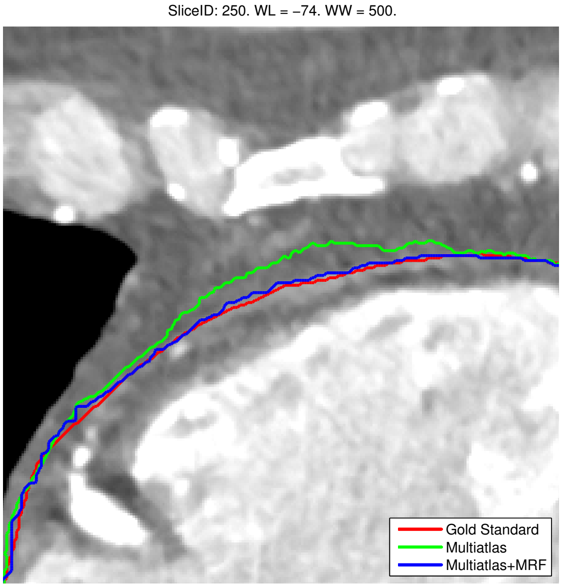

Figure 7.3: In this example where the pericardium is visible (the intensities are slightly brighter under the red line representing the boundary of the gold standard) we can clearly see the effect of the intensity dependent boundary cost of the MRF.

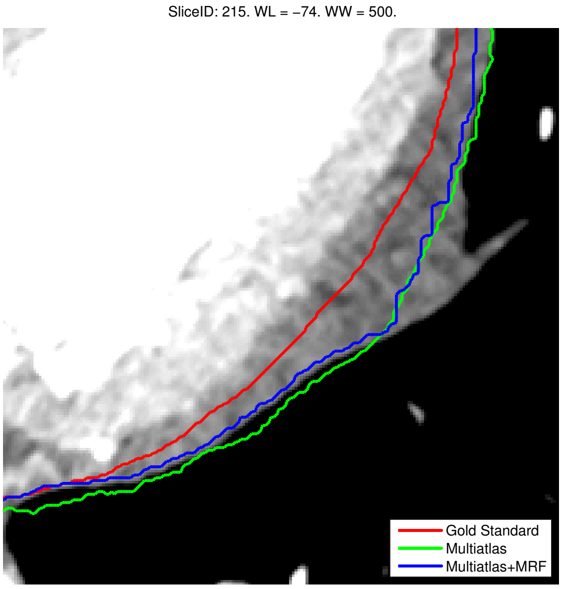

Figure 7.4: If the multi-atlas segmentation is not satisfactory and the pericardium is not clearly visible the segmentation will not represent the boundary accurately.

In this work we have been optimizing the algorithm against the Jaccard index. This is sometimes a problem since the pericardium is not in fact a closed object. We have the pulmonary veins and arteries, the aorta and inferior and superior vena cava that all are considerable objects that pass through the pericardium. When the expert made the Gold Standard labelings he more or less arbitrary cut through these objects. This is in general handled by our algorithm by the fact that the atlas registrations also provides a label propagated guess of how these areas should be closed. But we see one unexpected effect where this can fail.

The images series are acquired in a consistent manner. But we see slight variations. Figure 7.5 shows a slice of a volume where the image series starts unusually high (i.e. more of the arteries and veins in the top of the image is visible). Since most of the images of the atlases are cut just above where the manual labeling starts there is little information in the atlas set of what the heart looks like *above* the labeling. We can see this effect by the atlases being stretched far above the gold standard resulting in a miss aligned segmentation.

Our algorithm can actually be summarized well by Figure 7.5. It shows a slice of the image on which our algorithm performed the poorest results (Jaccard index 0.9002) and where most of the effects discussed are visible.

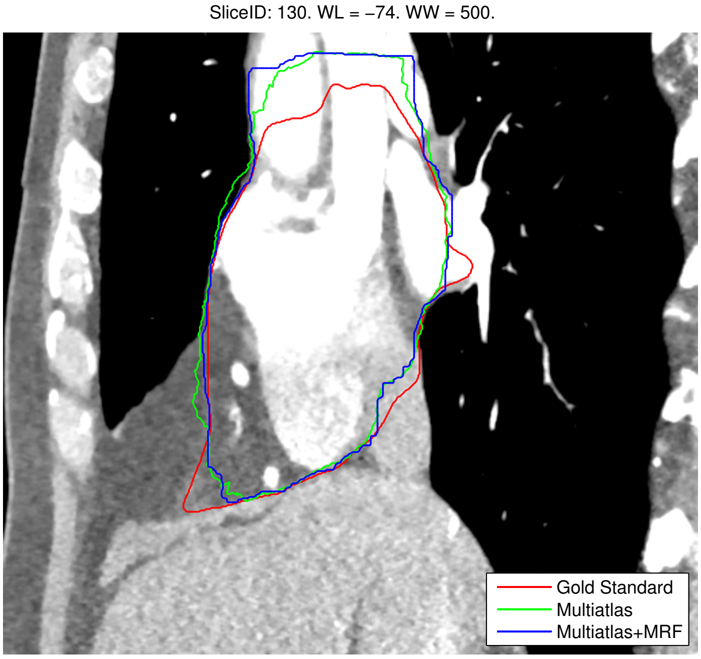

Figure 7.5: This figure shows a slice of the image on which our algorithm performed poorest. In a way it summarizes the advantages and drawbacks of our algorithm. The segmentation fails in the top of this slice due to the image stretching further above the pericardium relative to the other atlases. This has a large effect on the Jaccard index but arguably not as profound on the fat measurements. To the right and to the bottom left the pulmonary veins and the inferior vena cava pass through the pericardium and we can see that our algorithm struggle to close these areas in a way consistent with the the experts delineation. On the bottom left the multi-atlas segmentation was not so accurate. The effect of the MRF is clear where it successfully corrects the segmentation that passes through the lung and half of the fatty area to the left but not on the bottom left where the multi-atlas registrations were too far from the correct boundary.

7.6 Future Work

Although our algorithm is considerably faster than the same method based on Niftyreg there still is potential and a demand for an even faster method. We are currently working on a way to combine all the atlases into an *überatlas* based on features. One advantage of an *überatlas* would be to make it possible to construct a multi-atlas segmentation (or at least a good initialization) by only doing one registration. This registration could also produce information about which atlases that are similar to the target image (and therefore should be carefully registered) and which that should not. This would potentially not only reduce computational time but also allow for a having a much larger atlas set and more accurate segmentations.

Other improvements that would drastically reduce the time required for computations include to make our RANSAC implementation parallel and to use an efficient approximating algorithm for finding matching features (instead of brute force which has been used in this work).

One of the big challenges when working with extracted features in these types of images is handling the large amount of outliers. In this work the approach has been to use RANSAC and finding the largest consensus set. With this approach all correspondences within the consensus set are considered perfect matches (belonging to the model) and all that are not in the consensus set are given an equal cost. This approach for error modeling is very robust to outliers but future work would include research on a different way of modeling these errors.

The nonrigid feature based registration uses a straight forward approach which simply maps the corresponding features found by an affine initialization on top of each other using B-splines. By developing a more sophisticated method for non-rigid 3D registration based on features this algorithm would be significantly improved.

Chapter 8

Conclusion

We developed a fully automatic method for segmenting the pericardium in CTA-images that used feature based multi-atlas segmentation. The final segmentation was found by incorporating the votes into an MRF conditioned on the intensities of the target image. Further, the MRF was formulated in way that made use of two simple classifiers. One that classified the probability of pixels either belonging to the inside of the pericardium or background and another that classified the probableity of pixels being either on the boundary of the segmentation or not on the boundary.

We showed that our algorithm produced results that are better than state-of-the-art algorithms for pericardium segmentation and that we could do it in a computational efficient manner through the use of feature based registration. The Jaccard index of the final segmentation was 0.9379 ± 0.0186 after cross validation (Dice index: 0.9678 ± 0.0100). The running time of our algorithm was 13.6 minutes compared to about 5.4 hours using Niftyreg. But we note that there are room for considerable improvements of the running time only through more efficient implementation of the algorithm.

We showed that the feature based rigid initialization proved considerably more robust and accurate compared to the affine intensity based method.

The multi-atlas segmentation using feature based registration and simple majority voting did not provide as good results as the same segmentation based on the way more computationally demanding Niftyreg (in fact compared to the nonrigid Niftyreg algorithm based on the more robust feature based initialization). The main reason for this being that the feature based registration does not work with the image intensities directly and therefore can make errors that are apparent if the intensity information of the target image is taken into consideration. We showed that by incorporating the multi-atlas votes into an MRF that is conditioned on this intensity information (and that can be optimized efficiently) we obtained comparable results to the multi-atlas segmentation based on Niftyreg.

Bibliography

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110:346–359, 2008.

[2] Andrew Blake and Pushmeet Kohli. Introduction to Markov Random Fields. In *Markov Random Fields for Vision and Image Processing*, pages 1–15. 2011.

[3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1124–1137, 2004.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[5] Carl de Boor. *A Practical Guide to Splines.* Springer-Verlag, Berlin, 1978.

[6] Damini Dey, Ryo Nakazato, Debiao Li, and Daniel S Berman. Epicardial and thoracic fat - Noninvasive measurement and clinical implications. *Cardiovascular diagnosis and therapy*, 2:85–93, 2012.

[7] Damini Dey, Amit Ramesh, Piotr J Slomka, Ryo Nakazato, Victor Y Cheng, Guido Germano, and Daniel S Berman. Automated algorithm for atlas-based segmentation of the heart and pericardium from non-contrast CT. *Proceedings of SPIE*, 7623:762337, 2010.

[8] Ahmed Elnakib, Georgy Gimelfarb, Jasjit S. Suri, and Ayman El-Baz. Medical Image Segmentation: A Brief Survey. In *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, pages 1–37. Springer Science+Business Media, 2011.

[9] Martin A Fischler, Robert C Bolles, and J D Foley. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[10] Fahmi Khalifa, GarthM. Beache, Georgy Gimelfarb, JasjitS. Suri, and AymanS. El-Baz. State-of-the-Art Medical Image Registration Methodologies: A Survey. In *Multi Modality State-of-the-Art Medical Image Segmentation*

and Registration Methodologies, pages 235–280. Springer Science+Business Media, 2011.

[11] HA Kirisli and Michiel Schaap. Fully automatic cardiac segmentation from 3D CTA data: a multi-atlas based approach. *Proc. of SPIE*, 7623:762305, 2010.

[12] Seungyong Lee, George Wolberg, and Sung Yong Shin. Scattered data interpolation with multilevel b-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3:228–244, 1997.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[14] Marc Modat, Gerard R. Ridgway, Zeike A. Taylor, Manja Lehmann, Josephine Barnes, David J. Hawkes, Nick C. Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer Methods and Programs in Biomedicine*, 98:278–284, 2010.

[15] Hjart och Lungfonden. SCAPIS - en världsunik nationell kunskapskälla. `http://www.hjart-lungfonden.se/scapis`, 2014. [Online; accessed 30-Sept-2014].

[16] S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache. Reconstructing a 3D structure from serial histological sections. *Image and Vision Computing*, 19:25–31, 2001.

[17] Sébastien Ourselin, Radu Stefanescu, and Xavier Pennec. Robust Registration of Multi-modal Images : Towards Real-Time Clinical Applications. *MICCAI 2002 LNCS*, 2489:140–147, 2002.

[18] D Rueckert, L I Sonoda, C Hayes, D L Hill, M O Leach, and D J Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE transactions on medical imaging*, 18:712–721, 1999.

[19] Rahil Shahzad, Daniel Bos, Coert Metz, Alexia Rossi, Hortense Kirisli, Aad van der Lugt, Stefan Klein, Jacqueline Witteman, Pim de Feyter, Wiro Niessen, Lucas van Vliet, and Theo van Walsum. Automatic quantification of epicardial fat volume on non-enhanced cardiac CT scans using a multi-atlas segmentation approach. *Medical physics*, 40(9), September 2013.

[20] Aristeidis Sotiras, Christos Davatzikos, and Nikos Paragios. Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190, 2013.

[21] Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields. *Rapport technique MSCIS0421 Department of Computer and Information Science University of Pennsylvania*, 50:90, 2010.

[22] Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov Random Field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117:1610–1627, 2013.

[23] Richard Wendt. The Mathematics of Medical Imaging: A Beginner's Guide. *Journal of Nuclear Medicine*, 51(12):1987, December 2010.

[24] WHO. Global status report on noncommunicable diseases. Technical report, 2010.

[25] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

Master’s Theses in Mathematical Sciences 2014:E53
ISSN 1404-6342

LUTFMA-3267-2014

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/