Überatlas: Robust Speed-Up of Feature-Based Registration and Multi-Atlas Based Segmentation

Jennifer Alvén

January 2015

Abstract

Registration is a key component in multi-atlas approaches to medical image segmentation. Current state of the art uses intensity-based registration methods, but such methods tend to be slow and sensitive to large amount of noise and anatomical abnormalities present in medical images. In this master thesis, a novel feature-based registration method is presented and compared to two baseline methods; an intensity-based and a feature-based. The registration method is implemented with the purpose to handle outliers in a robust way and be faster than the two baselines. The algorithm performs a multi-atlas based segmentation by first co-registering the atlases and clustering the feature points in an überatlas, and then registering the feature clusters to a target image. The method is evaluated on 20 CT images of the heart and 30 MR images of the brain, with corresponding gold standard. The method produces comparable segmentation results to the two baseline methods and reduces the runtime significantly.

Keywords: computer vision, medical image analysis, feature-based registration, segmentation of pericardium, segmentation of brain, multi-atlas based segmentation

To John,
whom I will always love and who inspired me to dedicate my time to medicine.

Acknowledgments

I would like to take the opportunity to give thanks to my fantastic, inspiring and patient supervisors Fredrik Kahl and Olof Enqvist. I would also like to thank my tireless collaborator, former classmate and partner in crime, Alexander Norlén.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Segmentation, the process of automatically finding and outlining structures of images, is one of the most fundamental problems in medical image analysis. Segmentation may be used in order to locate tumors, measure tissue volumes, study of anatomical structure, surgery planning, virtual surgery simulation, intra-surgery navigation et cetera [17].

The value of automatic delineation of organs and other anatomical structures is huge, since manual delineation is time-consuming and sensitive to the skill of the expert. Moreover, it is important that the automatic segmentation is robust, that is, insensitive to anatomical abnormalities, noise, and other measurement errors, as well as fast, in order to be useful in clinical care. Moreover, the segmentation algorithm should of course also produce results comparable or better than the skill level of experts.

Several segmentation methods involves image registration, that is aligning two images into a common coordinate frame. In general, this is a difficult task due to the size of medical images, that may consist of hundreds of millions data values, as well as the large amount of noise, inter-subject variations et cetera. There are two different approaches that are considered as state of the art [9]; feature-based and intensity-based registration. Intensity-based registration methods have the capacity of producing accurate registrations, but are time-consuming and sensible to initialization. On the other hand, feature-based methods can be faster, but are sensitive to measurement errors and may risk failing due to the difficulty in establishing correct point-to-point correspondences between the images [21].

In this report, a novel feature-based registration method that takes advantage of co-registration of the data set is presented; überatlas registration. Furthermore, this approach is compared to two competing methods; referred to as the two *baselines*. The first baseline is an intensity-based registration method based on the program package `Niftyreg`, and the second is a standard feature-based method. The purpose of this report is to compare the three different methods with respect to robustness, speed and segmentation results.

1.2 Problem formulation

Given a set of 3D pixels $p \in \mathcal{P}$, an *image* refers to a medical 3D image consisting of a set of intensities $\mathcal{I} = \{i_p : p \in \mathcal{P}\}$. A 3D pixel is referred to as a *voxel* and the voxel coordinates are denoted as $\mathbf{x}_p = (x_p, y_p, z_p) \in \Omega \subset \mathbb{R}^3$. An image may therefore be regarded as a function that maps the coordinates of all the voxels in the image to an intensity, $\mathcal{I} : \Omega \subset \mathbb{R}^3 \to \mathbb{R}$.

A *label* of a voxel is a non-negative, discrete-valued number $l_p = 0, \ldots, L$ and a *labeling* is a set of labels for each voxels, $\mathcal{L} = \{l_p : p \in \mathcal{P}\}$. Therefore, a labeling may be regarded as a function that maps the coordinates of all voxels to a specific label, $\mathcal{L} : \Omega \subset \mathbb{R}^3 \to \mathbb{N}$. The value of the labels corresponds to which anatomical region a voxel belongs to. If two different voxels have the same label they belong to the same anatomical structure, and if not they do not.

A labeling performed manually by an expert, typically a physician, is defined as the *gold standard*, $\mathcal{L}$. A *segmentation* of an image is an estimated labeling of the image and denoted as $\hat{\mathcal{L}}$. In order to find the best possible segmentation, the *Jaccard index* between the gold standard and the segmentation should be approximately equal to 1. The Jaccard index, J, is a number between 0 and 1 that measures the similarity between two labelings. It is defined as

$$J(\mathcal{L}, \hat{\mathcal{L}}) := \frac{\left|\mathcal{L} \cap \hat{\mathcal{L}}\right|}{\left|\mathcal{L} \cup \hat{\mathcal{L}}\right|}. \tag{1.1}$$

The main problem of this thesis is to find a segmentation algorithm that maximizes the Jaccard index between the gold standard and the estimated labeling.

1.3 Proposed solution

The idea of introducing an überatlas that contains the information about pairwise correspondences between all available images in the form of point clusters was proposed by Norlén [14]. The general idea is to co-register all images as exactly as possible, followed by clustering of points that are considered close enough according to the co-registration. The purpose is to be able to register the point clusters to a new image, instead of register all images separately, and in that manner speed up the segmentation. Also, the überatlas will hopefully contain only the best fitted points for matching, which will hopefully reduce the amount of classification errors and improve the segmentation.

1.4 Contributions

The main contribution of this thesis is the proposition of how to speed-up a feature-based multi-atlas segmentation by co-registering atlases in order to create a mean atlas that contains information about feature points in all atlases. Also, this master thesis shows that using the truncated l_1-norm instead of using the truncated l_2-norm in RANSAC when estimating the affine transformation improves robustness as well as computational efficiency.

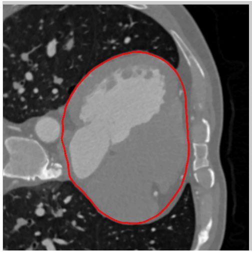

Figure 1.1: *A slice of a CT image of the heart and the expert delineation of the pericardium.*

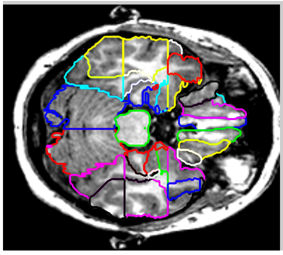

Figure 1.2: *A slice of a MR image of the brain and the expert delineation of some of the 83 regions present in the data set.*

1.5 Data sets

In this work, two different data sets of medical 3D images and manual delineations are used, see Figure 1.1 and 1.2.

The first data set consists of 20 CT images of the heart. CT (Computed Tomography) is a method where computer-processed X-ray images from different angles are used in order to produce virtual slices of the scanned object. The data set includes delineation of the pericardium (heart sack) for each image. The delineations are drawn for every 10th slice in all three viewing directions (also called axial, coronal and sagittal view) by an expert involved in the SCAPIS project. The SCAPIS project is an ongoing project and collaboration between Swedish hospitals and universities that aims to collect CT, MR and ultrasound images from 30 000 healthy subjects in order to detect early bio-markers for heart disease. The delineations are interpolated into a complete 3D delineation and approved by the expert in order to be considered as the gold standard of the data set. The resolution of the image is between $512 \times 512 \times 342$ and $512 \times 512 \times 458$ voxels. The size of the voxels is between $0.3320 \times 0.3320 \times 0.3000$ and $0.4297 \times 0.4297 \times 0.3000$ mm^3.

The second data set consists of 30 MR images of the brain of young adults. MR (Magnetic Resonance) produces slices of the scanned object with help of strong magnetic fields and radio waves. The data set also includes delineation of 83 regions of the brain, which are manually drawn according to the protocol in Hammers et al. [7]. These delineations represent the gold standard. The data set is available online on http://www.brain-development.org. The resolution of the images is between $165 \times 198 \times 155$ and $195 \times 199 \times 175$ voxels. The size of the voxels is between $0.9350 \times 0.9350 \times 0.9350$ and $0.9375 \times 0.9375 \times 0.9375$ mm^3.

1.6 Related work

An intensity-based multi-atlas approach has been used by Heckemann et al. [8] in order to segment the brain MR set, see Section 1.5. In their work, they succeed to perform single-atlas based segmentations with Jaccard index equal to 0.502 ± 0.032 and 0.605 ± 0.020 for affine and non-rigid transformations respectively, and multi-atlas based segmentations with Jaccard index equal to 0.616 ± 0.030 and 0.718 ± 0.013 for affine and non-rigid transformations respectively.

A feature-based multi-atlas approach similar to the method referred to as Baseline 2 was used by Norlén [14] in order to perform segmentations of a subset of the presented data set of the CT images of the heart. The values of the Jaccard index of the segmentations are similar to the values of the Jaccard index presented for Baseline 2. It should be noted, that Norlén used an MRF model to improve the results slightly.

The idea of co-registering and creating a mean atlas of the atlases is not novel. For instance, Dey et al. [4] proposed to co-register a set of atlases of the heart with an intensity-based affine and non-rigid transformation and then register a new target image to one of these atlases and let the other atlases vote indirectly. Gill et al. [6] proposed to create a mean atlas including feature points and mean shapes of a set of atlases of the lung, and use this mean atlas to initialize an active shape model.

1.7 Structure of report

In Section 2, the general image registration problem is presented. Also, atlas-based segmentation is explained and the multi-atlas approach is discussed. In Section 3 two methods are presented: intensity-based and feature-based registration. These methods will be used as a comparison to the performance of the invented and main registration method of this report; the überatlas registration presented in Section 4. In Section 6, the three registration methods will be compared with respect to i) Jaccard index, ii) detected inliers/outliers and iii) runtime. The implementation details and parameter choices that were used in order to produce the results are presented in Section 5. Finally, Section 7 and 8 provides a discussion around the obtained results.

Chapter 2

Theory

2.1 Image registration

The *image registration problem* refers to the problem of aligning one image with another by applying a geometric transformation. More specificly, the task is to find the transformation between a *target image* $\mathcal{I}_t$, also referred to as the fixed or reference image, and a *source image*, $\mathcal{I}_s$, also referred to as the moving or floating image that maximizes the similarity between the images. The transformation is denoted as $\mathbf{T}(\boldsymbol{\theta})$, and the parameter $\boldsymbol{\theta}$ is found by minimizing a cost, $\mathcal{C}$, (maximizing a similarity) between the target image and the transformed source image, that is, solving the problem

$$\underset{\theta}{\operatorname{argmin}} \quad \mathcal{C}(\mathcal{I}_t, \mathbf{T}(\boldsymbol{\theta}) \circ \mathcal{I}_s). \tag{2.1}$$

In this thesis, the transformation is a composition of an affine transformation and a non-rigid transformation, where the affine transformation is estimated first and then the non-rigid transformation. The affine transformation is a global transformation that describes the overall motion between the images, whereas the non-rigid transformation performs a local, more general deformation. Though, there are other possible choices of transformations.

Two different image registration methods are used in order to find the optimal transformation between the target and the source image; *intensity-based* and *feature-based* registration. Intensity-based registration aims to estimate the transformation between images by optimizing the intensity similarity, while feature-based registration aims to find point correspondences between features in the images such as blobs and corners, and estimate the transformation by maximizing the similarity between the coordinates of these points. Given N correspondences $(\mathbf{x}_n, \mathbf{y}_n)$, one usually solves the problem

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \sum_{n=1}^{N} \mathcal{C}(\mathbf{x}_n, \mathbf{T}(\boldsymbol{\theta}) \circ \mathbf{y}_n), \tag{2.2}$$

where $\mathcal{C}$ is a cost function.

Nevertheless, the differences between the methods are mainly how the parameters are estimated, while the formulas for the transformations remain the same. For more information about the general registration problem and transformations, see for example [9].

2.1.1 Affine transformation

The intention of the affine transformation is to affinely convert the voxels in the source image to the coordinate system of the target image. Given a voxel $p \in \mathcal{P}_s$ in the source image, where $\mathbf{x}_p$ are the coordinates of the voxel, the new coordinates $\mathbf{x}'_p$ in the target image are given by

$$\begin{pmatrix} \mathbf{x}'_p \\ 1 \end{pmatrix} = \mathbf{T}_{\text{aff}} \begin{pmatrix} \mathbf{x}_p \\ 1 \end{pmatrix}, \tag{2.3}$$

where the affine transformation has 12 degrees of freedom and is given by

$$\mathbf{T}_{\text{aff}} = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}. \tag{2.4}$$

2.1.2 Non-rigid transformation

The non-rigid transformation is a free-form deformation equal to a 3D tensor product on 1-D cubic B-splines proposed by Rueckert et al. [19]. The transformation is applied after the affine transformation.

The purpose of the non-rigid transformation is to allow more general transformations compared to the affine transformation. This is done by finding the displacement field that optimizes the similarity between images by capturing more arbitrary movements of the voxels than the affine counterpart. Though, in medical images only certain displacements are probable to occur. By using B-splines, the optimal displacement field is interpolated in a way that increases the smoothness of the transformation.

For simplicity, the non-rigid transformation is explained for coordinates in a target space given by $\Omega_t = \{\mathbf{x} = (x, y, z) : 0 \leq x \leq X, 0 \leq y \leq Y, 0 \leq z \leq Z\}$. The parameter of the transformation is a $N_x \times N_y \times N_z$ mesh denoted as $\Phi = \{\phi_i, \phi_j, \phi_k\}$ with uniform spacing, and the non-rigid transformation equals

$$\mathbf{T}_{\text{ffd}}(\mathbf{x}) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(u) B_m(v) B_n(w) \begin{pmatrix} \phi_{i+l} \\ \phi_{j+m} \\ \phi_{k+n} \end{pmatrix}, \tag{2.5}$$

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 is the lth basis function of the B-spline given by

$$\begin{aligned} B_0(u) &= (u-1)^3/6, \\ B_1(u) &= (3u^3 - 6u^2 + 4)/6, \\ B_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6, \\ B_3(u) &= u^3/6. \end{aligned} \tag{2.6}$$

2.1.3 Image warping

Given an estimated coordinate transformation $\mathbf{T}(\boldsymbol{\theta})$, the transformed image may be described as the interpolated set of old intensities given by $\mathbf{T}(\boldsymbol{\theta}) \circ \mathcal{I}_s = \{i_q = i_{p_q} : q \in \mathcal{P}_t\}$, where p_q is defined as

$$p_q = \underset{p}{\operatorname{argmin}} \quad |\mathbf{x}_q - \mathbf{T}(\boldsymbol{\theta}) \circ \mathbf{x}_p|, \quad p \in \mathcal{P}_s, \tag{2.7}$$

that is, a nearest neighbor interpolation is used.

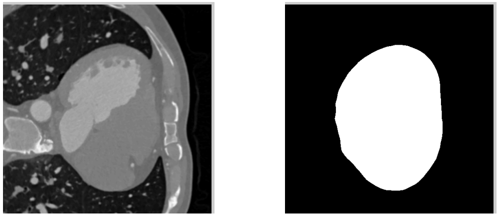

Figure 2.1: *A slice of a CT image of the heart and the corresponding labeling of the pericardium. Together, they form an atlas.*

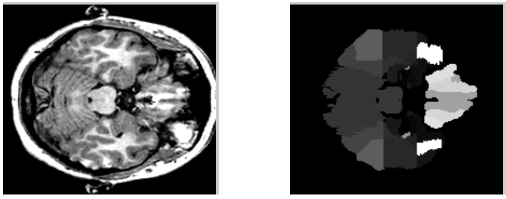

Figure 2.2: *A slice of a MR image of the brain and the corresponding labeling of some of the 83 regions present in the data set. Together, they form an atlas.*

2.2 Segmentation

2.2.1 Atlas-based segmentation

Together, an image and a labeling represents an *atlas*, $\mathcal{A} = \{\mathcal{I}, \mathcal{L}\}$, see Figure 2.1 and 2.2. By finding the optimal transformation $\mathbf{T}(\boldsymbol{\theta})$ between a source image $\mathcal{I}_s$ and a target image $\mathcal{I}_t$, a segmentation may be computed with the method *atlas-based segmentation.* This is done by transforming the labeling of the source atlas with the estimated transformation. Thus, the segmentation is given by

$$\hat{\mathcal{L}}_t = \mathbf{T}(\boldsymbol{\theta}) \circ \mathcal{L}_s. \tag{2.8}$$

2.2.2 Multi-atlas based segmentation

Given a set of atlases $\{\mathcal{A}_s : s = 1, \dots, M\}$, the segmentation of a target image can be estimated with an approach called *multi-atlas segmentation*, see [10, 8], by incorporating several atlas-based segmentation. Firstly, all individual labelings are transformed independently into the coordinate system of the target image by atlas-based segmentation. Thus, a set of segmentations is given by $\{\hat{\mathcal{L}}_t(s) = \mathbf{T}(\boldsymbol{\theta_s}) \circ \mathcal{L}_s : s = 1, \dots, M\}$. For each voxel in the target image, $p \in \mathcal{P}_t$, its label l_p is estimated by letting all estimated labels in the set given by $\{\hat{l}_p(s) : s = 1, \dots, M\}$ cast a weighted vote, that is,

$$\hat{l}_p = \underset{l}{\operatorname{argmax}} \quad \sum_{s=1}^{M} \omega_s \mathbb{1}_{\hat{l}_p(s)=l}, \quad l = 0, \dots, L, \tag{2.9}$$

where the weight ω_s determines the importance of the individual atlases. The final segmentation is thereafter given by $\hat{\mathcal{L}}_t = \{\hat{l}_p : p \in \mathcal{P}\}$. In this work, all atlases are given the same weight, which is a method called *majority voting.* See Figure 2.3 and 2.4.

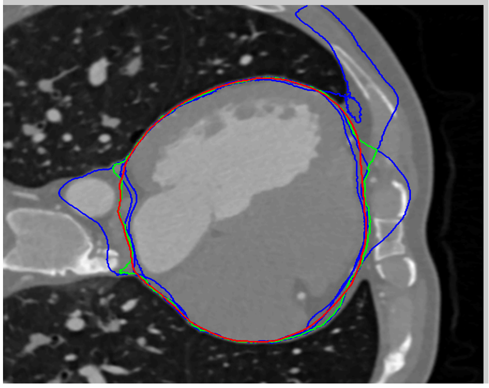

Figure 2.3: *A slice of a CT image of a heart and examples of segmentations. The gold standard is marked as red, the atlas-based segmentations as blue and the multi-atlas segmentation as green.*

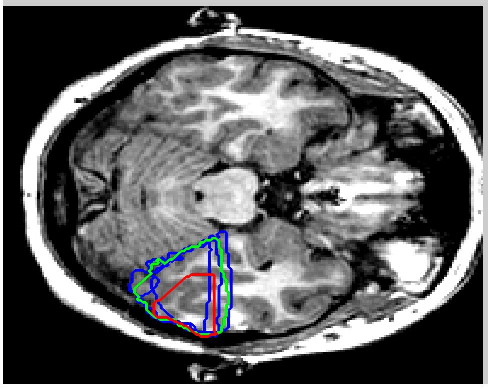

Figure 2.4: *A slice of a MR image of the brain and examples of segmentations of one of the regions in the brain. The gold standard is marked as red, the atlas-based segmentations as blue and the multi-atlas segmentation as green.*

Chapter 3

Baseline methods

3.1 Intensity-based registration

Intensity-based registration is frequently used in medical imaging. There are many different approaches based on different similarity measures and estimation methods. For this master thesis, the intensity-based program package `NiftyReg` is used as one of the baselines. This toolkit is available for everybody at no cost.

When using `NiftyReg` in order to register two images, the affine transformation is performed by the program `Reg_aladin` which is based on a block matching strategy proposed by and implemented by Ourselin et al. [16, 15]. The non-rigid transformations are performed by the program `Reg_f3d`, which is based on the free-form deformation explained in Section 2.1.2, and implemented by Modat et al. [13].

The affine transformation is used as an initialization for the non-rigid transformation. That is, when using `Reg_f3d`, both the parameters of the affine transformation and the grid of the free-form deformation are estimated.

Finally, when the transformations are estimated, the source image is transformed to the target space with help of the program `reg_resample`, where nearest neighbor interpolation is used, see Section 2.1.3.

3.1.1 Affine transformation

A block $\mathcal{B}(\mathbf{x})$ is defined as a $N \times N \times N$ patch centered at the position $\mathbf{x} + (N/2, N/2, N/2)$. The mean intensity of the block is denoted as $\mu(\mathbf{x})$ and the standard deviation of the intensities as $\sigma(\mathbf{x})$. Given a block in the target image, $\mathcal{B}_t(\mathbf{x})$, and a corresponding block in the source image, $\mathcal{B}_s(\mathbf{y})$, the similarity between the two blocks is computed as the correlation coefficient of the voxel intensities,

$$C(\mathcal{B}_t(\mathbf{x}), \mathcal{B}_s(\mathbf{y})) = \frac{1}{N^3} \sum_{i_1=0}^{N-1} \sum_{i_2=0}^{N-1} \sum_{i_3=0}^{N-1} \frac{(\mathcal{I}_t(\mathbf{x}+\mathbf{i}) - \mu_t(\mathbf{x}))(\mathcal{I}_s(\mathbf{y}+\mathbf{i}) - \mu_s(\mathbf{y}))}{\sigma_t(\mathbf{x})\sigma_s(\mathbf{y})}, \tag{3.1}$$

where $\mathbf{i} = (i_1, i_2, i_3)$.

By finding the block position in the target image that maximizes the similarity, that is, solving

$$\underset{\mathbf{x}_p}{\operatorname{argmax}}\, C(\mathcal{B}_t(\mathbf{x}_p), \mathcal{B}_s(\mathbf{y})), \quad p \in \mathcal{P}_t \tag{3.2}$$

for every block in the source image, a list of corresponding 3D points is obtained. These points may be used in order to estimate an affine transformation between the images, by minimizing the distance between the correspondences using trimmed least squares [18].

In order to find the corresponding blocks, the blocks in the source image are moved in its neighborhood and compared to blocks with a similar position in the target image. In order to improve computational efficiency, the algorithm is implemented in a multi-scale iterative scheme. The algorithm starts estimating a rigid transformation at a coarse level with large blocks and neighborhoods, and progressively refines the scale while estimating affine transformations until a predefined number of refinements is reached.

3.1.2 Non-rigid transformation

In the non-rigid transformation, the parameters (the parameters of the affine transformation and the mesh Φ) are computed by minimizing a cost function consisting of a similarity term and a smoothness term given by

$$\mathcal{C} = -\mathcal{C}_{\text{similarity}}(\mathcal{I}_t, \mathbf{T} \circ \mathcal{I}_s) + \lambda \mathcal{C}_{\text{smoothness}}(\mathbf{T}). \tag{3.3}$$

The similarity term is a measure of alignment of the images and equals the *normalized mutual information*, which is a concept from information theory that describes the amount of information one image contains about another, and is given by

$$\mathcal{C}_{\text{similarity}}(\mathcal{I}_t, \mathbf{T} \circ \mathcal{I}_s) = \frac{H(\mathcal{I}_t) + H(\mathbf{T} \circ \mathcal{I}_s)}{H(\mathcal{I}_t, \mathbf{T} \circ \mathcal{I}_s)}, \tag{3.4}$$

where $H(\cdot)$ denotes the entropy of a image and $H(\cdot, \cdot)$ is the joint entropy of two images. The entropy is computed by histograms of the images.

The smoothness term aims to regularize the non-rigid transformation in a way that constrains the transformation to be smooth. The smoothness term only penalizes non-rigid transformations, that is, it is zero for affine transformations. The parameter λ decides the relationship between the similarity term and the smoothness term.

The smoothness term is given by the equation

$$\begin{aligned} \mathcal{C}_{\text{smoothness}} =& \frac{1}{|\Omega_t|} \int_0^X \int_0^Y \int_0^Z (\frac{\partial^2 \mathbf{T}}{\partial x^2} + \frac{\partial^2 \mathbf{T}}{\partial y^2} + \frac{\partial^2 \mathbf{T}}{\partial z^2} + \\ & 2\frac{\partial^2 \mathbf{T}}{\partial x \partial y} + 2\frac{\partial^2 \mathbf{T}}{\partial y \partial z} + 2\frac{\partial^2 \mathbf{T}}{\partial z \partial x}) dx dy dz. \end{aligned} \tag{3.5}$$

Firstly, the optimal affine transformation is estimated by maximizing (3.4). Then, the non-rigid transformation is estimated by iteratively refining the grid of the control points. In each iteration, a local minimum of the cost function in (3.3) is found by the method of steepest descent followed by a refinement of the grid of control points. The algorithm is continued until a predefined termination criterion.

3.2 Feature-based registration

Feature-based registration aims to find and compare discrete correspondences between two images, unlike most of the intensity based methods that measures intensity similarities between complete images and/or patches. Examples of features are lines, corners and blobs (blob-like structures).

Generally, feature-based registration consists of three initial steps; detection, description and matching. Firstly, detection is the step where feature points are extracted from an image. Secondly, description is the step where the properties of the neighborhood of each feature point is estimated by a descriptor. Finally, matching refers to when descriptors are compared in order to find correspondences between images. Thereafter, a transformation of the source image to the target space may be estimated by using corresponding features in the images.

In this work this, the detection of features is done with the method SIFT (Scale Invariant Feature Transform) proposed by Lowe [12]. The description of the feature points is based on SURF (Speeded Up Robust Features) proposed by Bay et al. [1]. Finally, matching is performed by a symmetric nearest neighbor approach.

3.2.1 Correspondences

Feature point detection

The SIFT feature point detector is based on a scale pyramid approach such that the feature points are searched for at different scales of the image. Given an image, smoothing at different scales σ is performed by applying a convolution of a 3D Gaussian filter G to the image, given by

$$G(\mathbf{x}, \sigma) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_c|^2}{2\sigma^2}\right), \tag{3.6}$$

where $\mathbf{x}_c$ is the center of the image.

Convolutions of the image are performed iteratively from the finest level, corresponding to $\sigma_{\min}$, to the most coarse level, corresponding to $\sigma_{\max}$. That is, the image at the most coarse level is computed by smoothing the image at the next most coarse level et cetera. Every time the scale has doubled, a new *octave* (the next pyramid level) is entered and the image is downsampled by a factor 2 in order to speed up the computations.

For each octave, the Difference of Gaussian $\Gamma(\sigma)$ (DoG) is computed for each scale. If $\sigma_i < \sigma_{i-1}$, DoG at scale σ_i is given by

$$\Gamma(\sigma_i) = \mathcal{I} * G(\mathbf{x}, \sigma_i) - \mathcal{I} * G(\mathbf{x}, \sigma_{i-1}). \tag{3.7}$$

By finding local optima, $(\mathbf{x}, \sigma)$, to the DoG, feature points are detected. A feature point is considered a local maximum (or minimum) if $\Gamma(\sigma_i)$ has a strictly larger (or smaller) value than the closest pixels horizontally, vertically and diagonally (both spatially and in the scale-dimension). That is, there are in total $27 + 27 + 26$ neighbors. Moreover, in order to be considered a feature point, the points are investigated with respect to skewness (it should correspond to a corner or blob, and not an edge) and contrast. Feature points that do not fulfill predefined thresholds are discarded. See Figure 3.1 and 3.2 for detected feature points in a slice of a heart and a brain, respectively.

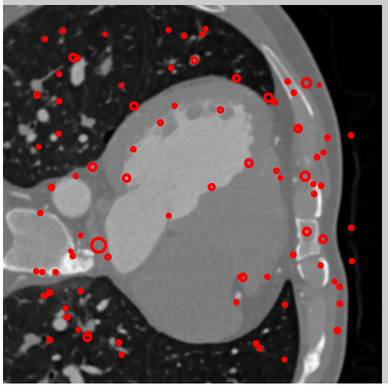

Figure 3.1: *A slice of a CT image of the heart and detected feature points in the neighborhood. The radius of the circles are proportional to the scale where the feature points were detected.*

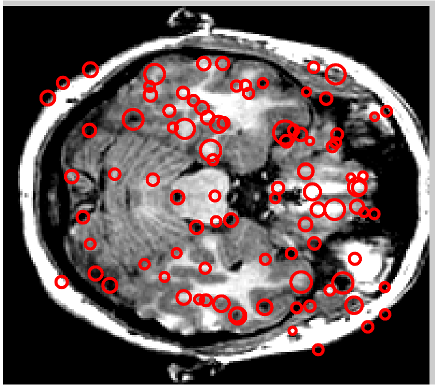

Figure 3.2: *A slice of a MR image of the brain and detected feature points in the neighborhood. The radius of the circles are proportional to the scale where the feature points were detected.*

Feature point description

The purpose of the feature point descriptor is to, in a compact but representable way, describe the neighborhood of each feature point. In this work, the SURF approach is used, which describes properties of the image in the neighborhood of the feature point. The standard SURF descriptor is rotation invariant, though in this work, the upright, rotation variant descriptor (U-SURF) is used.

The SURF descriptor is computed by considering a patch for every feature point, formed as a cube with center equal to the coordinates of the feature point and with edges proportional to the scale. Every patch is divided into $4 \times 4 \times 4 = 64$ regions, and for each region i the gradients ∇_i are computed for every point in a grid (and for some points outside, but close to the regions). All gradients are weighted with a Gaussian filter with center at the feature point, and for each region the gradients are smoothed with an additional Gaussian filter with center in the middle of the region.

Finally, the descriptor $\mathbf{d}(\mathbf{x})$ is defined as

$$\begin{aligned}\mathbf{d}(\mathbf{x}) = \Bigg(& \sum \left(\frac{\partial}{\partial x}\right)_1, \sum \left(\frac{\partial}{\partial y}\right)_2, \sum \left(\frac{\partial}{\partial z}\right)_3, \\ & \sum \left|\left(\frac{\partial}{\partial x}\right)_1\right|, \sum \left|\left(\frac{\partial}{\partial y}\right)_2\right|, \sum \left|\left(\frac{\partial}{\partial z}\right)_3\right|, \ldots, \sum \left|\left(\frac{\partial}{\partial z}\right)_{64}\right| \Bigg),\end{aligned} \tag{3.8}$$

that is, the descriptor is a vector with 384 elements. Finally, the descriptor is normalized in order to be invariant to intensity scaling.

Feature point matching

In contrast to the approach used in both SIFT and SURF, this master thesis includes a matching step where the matching between the descriptors is performed symmetrically. Given a set of M feature points in the target image and a set of N feature points in the source image a pair of feature points $(\mathbf{x}, \mathbf{y})$ is considered as a correspondence if the following criterion is fulfilled

$$\begin{aligned}\mathbf{x} &= \underset{\mathbf{x}_m}{\operatorname{argmax}} \quad \mathbf{d}(\mathbf{y}) \cdot \mathbf{d}(\mathbf{x}_m), \qquad m = 1, \ldots, M, \\ \mathbf{y} &= \underset{\mathbf{y}_n}{\operatorname{argmax}} \quad \mathbf{d}(\mathbf{y}_n) \cdot \mathbf{d}(\mathbf{x}), \qquad n = 1, \ldots, N.\end{aligned} \tag{3.9}$$

3.2.2 Inliers and outliers

When estimating the transformation between the correspondences, there are two different kinds of errors that need to be taken into account. Firstly, the measurement error (the noise) and secondly, classification errors (improperly matched correspondences). Feature points that are incorrectly classified as correspondences are referred to as *outliers*, while correctly matched correspondences are referred to as *inliers*.

One disadvantage of working with features points is the large amount of outliers introduced to the problem. When estimating the transformations based on correspondences, a common cost function such as the l_2-norm is able to deal with the noise, but may fail to produce good results due to outliers. This problem is solved in two different manners; by introducing a more robust cost function, the truncated l_2-norm, and by using RANSAC (Random Sample Consensus), see [5].

3.2.3 Affine transformation

Given a set of N correspondences, the affine transformation $\mathbf{T}_{\text{aff}}$ is found by minimizing the cost function chosen as the truncated l_2-norm, proposed by Blake et al. [2]. When minimizing the truncated l_2-norm the transformation is found by solving the problem

$$\underset{\mathbf{A},\mathbf{t}}{\operatorname{argmin}} \quad \sum_{n=1}^{N} \min\left(|\mathbf{A}\mathbf{y}_n + \mathbf{t} - \mathbf{x}_n|^2, \epsilon^2\right), \tag{3.10}$$

where ϵ is a parameter. The advantage of introducing a truncation is to assign inliers a cost corresponding to the squared euclidean error while the outliers are assigned a constant cost ϵ^2. Here, ϵ should correspond to the euclidean threshold between inliers and outliers and should be approximately equal to the noise level.

The estimation of the affine transformation is computed by using RANSAC, proposed by Fischler and Bolles [5]. The purpose of RANSAC is to minimize the cost function (or maximize the amount of inliers) by iteratively computing an affine transformation with a minimal set of correspondences sampled randomly. In order to estimate an affine transformation, the minimal set consist of 4 correspondences since an affine transformation has 12 degrees of freedom. In each iteration, the current cost function is evaluated and saved if it corresponds to a new minimum. Finally, when a predefined amount of iterations are run, the final affine transformation is estimated with the maximum amount of inliers with linear least squares.

3.2.4 Non-rigid transformation

After an affine transformation is estimated, a non-rigid transformation is found by using the free-form deformation explained in Section 2.1.2. The parameter of the transformation that is optimized is the control grid Φ. Unlike the intensity-based case, the cost function only depends on corresponding points in the target image and the source image, and does not take the intensities of the image into account. In this work, this is performed by a program implemented by D. Kroon based on a proposition of Lee et al. [11]. As an input, it needs a set of points in the target image, $\{\mathbf{x}_n : n = 1, \ldots, N\}$, and a set of affinely transformed points from the source image, $\{\mathbf{T}_{\text{aff}} \circ \mathbf{y}_n : n = 1, \ldots, N\}$. In order to get a good non-rigid transformation, the sets are cleared from the correspondences marked as outliers after the final affine transformation.

Chapter 4

The überatlas

The idea of introducing an überatlas that contains the information about pairwise correspondences between all available atlases in the form of feature clusters was proposed by Norlén [14]. The general idea is to co-register all atlases as exactly as possible, followed by clustering of all extracted feature points that are considered close enough according to the co-registration. The purpose is to be able to register the feature clusters to a new target image, instead of registering all atlases separately, and in that manner speed up the segmentation. Also, the überatlas will hopefully contain only the best fitted feature points for matching, which will hopefully reduce the amount of outliers and improve the segmentation.

First, some definitions will be introduced followed by the general registration algorithm and finally information about how to construct an überatlas with help of co-registration and agglomerative clustering.

4.1 Definitions

Given a total number of N detected feature points in M atlases, the jth feature point is denoted as $\mathbf{f}_j$, see Definition 4.1.

Definition 4.1 *A feature point* $\mathbf{f}$ *is uniquely determined by the index of the atlas from which it is extracted,* i*, the coordinates of the feature point in the coordinate frame of the atlas,* $\mathbf{x}$*, and the corresponding descriptor,* $\mathbf{d}$*, that is,*

$$\mathbf{f} = (i, \mathbf{x}, \mathbf{d}). \tag{4.1}$$

The *mean atlas* is a arbitrarily chosen atlas with index $\hat{\imath}$ out of the M atlases. The jth *transformed feature point* is denoted as $\hat{\mathbf{f}}_j$ and corresponds to the jth feature point, though its coordinates are transformed into the coordinate frame of the mean atlas, see Definition 4.2.

Definition 4.2 *A transformed feature point* $\hat{\mathbf{f}}$ *is uniquely determined by the index of the atlas from which it is extracted,* i*, the coordinates of the feature point transformed into the coordinate frame of the mean atlas,* $\hat{\mathbf{x}}$*, and the corresponding descriptor* $\mathbf{d}$*, that is,*

$$\hat{\mathbf{f}} = (i, \hat{\mathbf{x}}, \mathbf{d}). \tag{4.2}$$

The transformed coordinates of the jth feature point are given by $\hat{\mathbf{x}}_j = \hat{\mathbf{T}}_{i_j,\hat{i}} \circ \mathbf{x}_j$, where the transformation $\hat{\mathbf{T}}_{i_j,\hat{i}}$ is a non-rigid transformation that should describe the mapping between the atlas indexed as i_j and the mean atlas $\hat{i}$ as exactly as possible, see Definition 4.3. From now on, $\hat{\mathbf{x}}_j$ will be referred to as the *mean atlas coordinates* of jth feature point.

Definition 4.3 *The silver standard transformation, $\hat{\mathbf{T}}_{i,\hat{i}}$ is the optimal non-rigid transformation between the atlas given by index i and the mean atlas $\hat{i}$, and should solve the following cost function optimization problem*

$$\hat{\mathbf{T}}_{i,\hat{i}} = \underset{\mathbf{T}_{i,\hat{i}}}{\operatorname{argmin}} \quad \mathcal{C}_I(\mathcal{I}_{\hat{i}}, \mathbf{T}_{i,\hat{i}} \circ \mathcal{I}_i) + \mathcal{C}_L(\mathcal{L}_{\hat{i}}, \mathbf{T}_{i,\hat{i}} \circ \mathcal{L}_i), \tag{4.3}$$

which means that the transformation should both take the similarity of the images and the similarity of the labelings into account.

A *feature cluster* $\mathcal{F}$ is a set of transformed features from different atlases with similar mean atlas coordinates. Moreover a feature cluster contains more than 3 transformed features, see Definition 4.4.

Definition 4.4 *A feature cluster $\mathcal{F}$ is a set of transformed features $\mathcal{F} = \{\hat{\mathbf{f}}_k : k \in \mathcal{K}\}$ such that*

$$|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k'}| < \epsilon_s \quad \forall k, k' \in \mathcal{K}, \tag{4.4}$$

$$i_k \neq i_{k'} \quad \forall k, k' \in \mathcal{K}, \tag{4.5}$$

$$3 \leq |\mathcal{K}| \leq M, \tag{4.6}$$

where ϵ_s is the spatial threshold.

The *mean atlas descriptor* of the uth feature cluster is denoted as $\hat{\mathbf{d}}_u$, and the definition is given in Definition 4.5.

Definition 4.5 *The mean atlas descriptor $\hat{\mathbf{d}}$ of the cluster $\mathcal{F}$ is given by the mean of all descriptors in the cluster,*

$$\hat{\mathbf{d}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathbf{d}_k. \tag{4.7}$$

Finally, an *überatlas* is a set of non-overlapping feature clusters, see Definition 4.6.

Definition 4.6 *An überatlas is a set of clusters $\{\mathcal{F}_u : u \in \mathcal{U}\}$, such that*

$$\mathcal{F}_u \cap \mathcal{F}_{u'} = \emptyset \quad \forall u, u' \in \mathcal{U}. \tag{4.8}$$

To sum up, the überatlas contains information about corresponding features in all atlases, as well as an approximate way of describing all the feature clusters. Therefore, an überatlas is a set of clusters that describes feature points that are found in several images that can be matched with help of the mean atlas descriptors, as well as the original descriptors, to a new target image.

4.2 Überatlas registration

4.2.1 Correspondences

Given a new target image, feature points are detected and descriptors are computed as explained in Section 3.2.1. Thereafter, the feature points are matched to the clusters in the überatlas by matching the descriptors of the feature points detected in the target image to the mean atlas descriptors.

Firstly, each feature cluster in the überatlas are matched to its nearest neighbor. Given the total amount of T detected feature points in the target image and a specific feature cluster, $\mathcal{F}_u$, the index t corresponds to the feature point that should be matched to the feature cluster. The index t is computed by solving

$$t = \underset{t'}{\operatorname{argmax}}\, \mathbf{d}_{t'} \cdot \hat{\mathbf{d}}_u, \quad t' = 1, \ldots, T, \tag{4.9}$$

which means that the total number of correspondences in the first matching step is equal to the total number of feature clusters.

Secondly, it is not certain that a cluster correspondence $(\mathbf{f}_t, \mathcal{F}_u)$ is an actual correspondence when it comes to registration of a specific source atlas to the target image, since all source atlases may not be represented in the specific feature cluster. That is, only the correspondences where the feature cluster contains a transformed feature from the specific source atlas may be used. Given a source atlas indexed as i_s, the set of correspondences are therefore given by $\{(\mathbf{f}_t, \mathbf{f}_s) : i_t = i_s, \hat{\mathbf{f}}_s \in \mathcal{F}_u\}$.

Moreover, in order to eliminate outliers, a restriction on the distance between the descriptors of the correspondences are introduced. Given the correspondence $(\mathbf{f}_t, \mathbf{f}_s)$, it is considered as an inlier if the following criterion is fulfilled

$$|\mathbf{d}_t - \mathbf{d}_s| < \epsilon_d \mathbf{D}_u, \tag{4.10}$$

where ϵ_d is a parameter and $\mathbf{D}_u$ is the maximum euclidean distance between the descriptors within the cluster $\mathcal{F}_u$ and given by

$$\mathbf{D}_u = \max\{|\mathbf{d}_i - \mathbf{d}_j| : \hat{\mathbf{f}}_i, \hat{\mathbf{f}}_j \in \mathcal{F}_u\}. \tag{4.11}$$

4.2.2 Affine transformation

As described in Section 3.2.2, the problem of using feature-based registration is the great amount of outliers produced. Even though the amount of outliers hopefully is reduced thanks to the construction of the überatlas, there is no reason to believe that the problem is eliminated. In the method presented in Section 3.2.3, the problem was solved with the RANSAC algorithm based on the truncated l_2-norm.

In order to reduce the sensitivity of the choice of truncation threshold and to enable larger thresholds without losing robustness, the affine transformation is estimated by optimizing a hopefully more robust cost function. This cost function is the truncated l_1-norm, that is, the truncated l_1-norm is assumed to have the potential to be more robust than the truncated l_2-norm for larger truncation thresholds. The assumption is based on the fact that the l_2-norm is suitable when the noise is Gaussian, and otherwise the l_1-norm may be a better

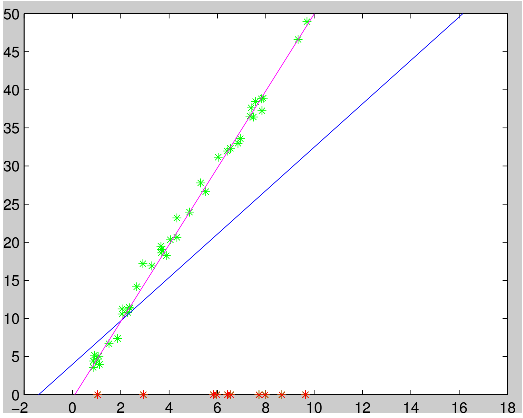

Figure 4.1: *A set of inliers (marked as green) given by* $\{(x_i, y_i) : y_i = x_i + \epsilon, \epsilon \in \mathcal{N}(0, 50)\}$*, and a set of outliers (marked as red) given by* $\{(x_i, y_i) : y_i = 0\}$*. Blue line corresponds to the* l_2 *solution, and the purple line to the* l_1 *solution.*

choice, see Figure 4.1. By increasing the value of the spatial threshold, there is a risk of including more outliers when estimating the affine transformation, which may result in that the assumption of normally distributed residuals is incorrect. Therefore, the affine transformation is estimated by optimizing the truncated l_1-norm, which is hopefully is more robust than the truncated l_2-norm.

Iteratively reweighted least squares

Given a set of N local correspondences, where the coordinates of the correspondences are given by $\{(\mathbf{x}_n, \mathbf{y}_n) : n = 1, \ldots, N\}$, the affine transformation $\mathbf{T}_{\text{aff}}$ is estimated by minimizing the cost function chosen as the truncated l_1-norm of the residuals. When using the truncated l_1-norm, the transformation is computed by solving the problem

$$(\hat{\mathbf{A}}, \hat{\mathbf{t}}) = \underset{\mathbf{A},\mathbf{t}}{\operatorname{argmin}} \sum_{n=1}^{N} \min\left(|\mathbf{A}\mathbf{y}_n + \mathbf{t} - \mathbf{x}_n|, \epsilon\right), \tag{4.12}$$

where ϵ is a parameter.

Unlike the l_2-norm, there is no analytical expression that provides the optimal estimation of $\mathbf{T}_{\text{aff}}$. However, the affine transformation may still be obtained by solving the problem approximately with a method called *iteratively reweighted least squares*, which is a algorithm that converges linearly for l_1-problems, see Chartrand and Yin [3]. In order to enable a faster algorithm RANSAC is skipped and the affine transformation is estimated by executing iteratively reweighted least squares once.

Instead of optimizing the equation given by (4.12), an approximate minimization problem is solved iteratively with the l_2-norm as a cost function. In each iteration i, the approximate minimization problem is given by

$$(\hat{\mathbf{A}}_i, \hat{\mathbf{t}}_i) = \arg\min_{\mathbf{A}_i,\mathbf{t}_i} \sum_{n=1}^{N} \omega_{i,n} |\mathbf{A}_i\mathbf{y}_n + \mathbf{t}_i - \mathbf{x}_n|^2, \tag{4.13}$$

where the weight $\omega_{i,n}$ should be a measure of the errors in the previous iteration in order to approximate truncated l_1-norm.

In order to avoid dividing with zero, a regularization, $|\delta| \ll 1$, is introduced. Moreover, in order to take the truncation into account, the weights are set to zero if the residuals reside outside the outlier threshold. Thereby, the weights are given by

$$\omega_{i,n} = \begin{cases} 1/\max\{|r_{i-1,n}|, |\delta|\}, & |r_{i-1,n}| < \epsilon, \\ 0, & |r_{i-1,n}| \geq \epsilon, \end{cases} \tag{4.14}$$

where the residuals are given by $r_{i,n} = \mathbf{A}_i\mathbf{y}_n + \mathbf{t}_i - \mathbf{x}_n$.

4.2.3 Non-rigid transformation

The non-rigid transformation is the same as described in Section 3.2.4.

4.3 Training

4.3.1 Silver standard

In Section 4.1, a non-rigid transformation, the silver standard transformation, was used in order transform feature coordinates to the coordinate frame of the mean atlas. In order to transform features as correctly as possible, a set of transformations between the atlases and a mean atlas need to be estimated as exactly as possible. A combination of the feature-based registration method explained in Section 3.2 and the intensity-based registration method explained in Section 3.1 is used in order to perform pairwise registrations between the atlases and the mean atlas. While the explained methods only use the information in the images, the fact that the atlases also hold information about the correct labeling, the gold standard, is used in this section.

Firstly, each atlas $\mathcal{A}_i$ is concatenated into a $N_{x_i} \times N_{y_i} \times N_{z_i} \times 2$ 4D image where the first element in the fourth dimension corresponds to the image $\mathcal{I}_i$ and the second element corresponds to the weighted labeling $\alpha_i\mathcal{L}_i$.

A feature-based estimation of the affine transformation is used as an initialization to the non-rigid transformation. There are several reasons why the feature-based registration is used; it is faster and previous work has showed that the affine registration works better than the corresponding intensity-based method used by `NiftyReg` for hearts [14]. Also, it is just an initialization. The non-rigid registration is performed by `NiftyReg`, due to its capability of register 4D images to each other using normalized mutual information as described in Section 3.1.2. Though, in the 4D case the similarity measure is based on a bivariate distribution. Therefore, the weighting of the labeling by the parameter α is needed to make sure that the labeling contributes with desired impact.

The standard output of `reg_ffd` gives a control point grid transformation. Though, with help of an additional function in `NiftyReg` called `reg_transform`, the control point grid parametrization is easily converted into a displacement field, $\mathbf{T}_{\text{disp}} : \Omega_i \to \mathbb{R}^3$. That is, the silver standard transformation for the ith atlas is given by

$$\hat{\mathbf{x}} = \mathbf{T}_{i,\hat{i}} \circ \mathbf{x} = \mathbf{x} + \mathbf{T}_{\text{disp}}(\mathbf{x}). \tag{4.15}$$

4.3.2 Feature clusters

In order to construct the feature clusters of the überatlas, the method of agglomerative clustering is used. Each transformed feature point will start in a cluster with size 1 and pairs of clusters are merged as the clustering proceeds. Before clustering starts, a distance matrix based on the distances between all descriptors are constructed and sorted. The algorithm tries to merge feature points with the smallest descriptor distances first. For every potential merging of clusters, the criteria given by (4.4), (4.5) and (4.6) are checked. Finally, feature clusters that do not fulfill the criterion given by (4.6) are discarded.

Chapter 5

Implementation

5.1 Evaluation

Pair-wise registration of 20 images of the heart and 30 images of the brain were performed with intensity-based registration (Baseline 1), feature-based registration (Baseline 2) and überatlas registration. Since all the affine registration methods and the non-rigid transformations are non-symmetric, 180 pairwise registrations of the heart and 870 pairwise registrations of the brain were computed. For these registrations, the mean and standard deviation of the Jaccard index defined in Section 1.2 were calculated. Also, multi-atlas segmentation for each image was performed with the remaining atlases; 20 multi-atlas segmentations of the heart (with 19 atlases) and 30 multi-atlas segmentations of the brain (with 29 atlases). The multi-atlas segmentations were evaluated by computing the Jaccard index.

The mean value and standard deviation of the amount of correspondences found in the registration were saved for the features-based registration and the überatlas registration. Also, the mean value and standard deviation of the amount of inliers and outliers according the registration were saved. Finally, all correspondences were evaluated with respect to the silver standard for these two methods as well. A correspondence was set to a *silver standard inlier* (SSI) if the spatial distance between the transformed feature points were less than 10 mm, otherwise set to a *silver standard outlier* (SSO). Therefore, the percentage of correspondences correctly/incorrectly marked as an inlier (or an outlier) could be estimated.

5.2 Implementation

5.2.1 Baseline 1: Intensity-based registration

The programs `reg_aladin` and `reg_f3d` were used in order perform a pair-wise registration of 20 images of the heart and 30 images of the brain. The programs were run with default values.

5.2.2 Baseline 2: Feature-based registration

Feature points were extracted by a feature detection and description program `ff3d.m`, implemented by Svärm et al. [20]. The program was run with default values and for rotation variant features. The images of the brains were upsampled with a factor 2. The affine registration was performed with RANSAC, that was run 10000 iterations. The spatial threshold for inliers, ϵ, was tested to be 10 and 50 mm. The non-rigid transformation was run with default values, though the number of grid refinements was set to 5.

5.2.3 Überatlas registration

Feature points were extracted by the feature detection and description program `ff3d.m`. The program was run with default values and for rotation variant features. The images of the brains were upsampled with a factor 2. The affine registration was performed with iteratively reweighted squares with 10 iterations. The spatial threshold for inliers, ϵ, was tested for different values, where 10 and 50 mm are presented in this report. The descriptor threshold, ϵ_d, was set to 1 for both hearts and brains. The non-rigid transformation was run with default values, though the number of grid refinements was set to 3.

5.3 Überatlas training

5.3.1 Silver standard

Affine transformation

Feature points were extracted by the feature detection and description program `ff3d.m`. The program was run with default values and for rotation variant features. The images of the brains were upsampled with a factor 2. The labeling weighting factor α_i was set to $\max \mathcal{I}_i - \min \mathcal{I}_i$ (the intensity span of the image). The affine registration was performed with RANSAC, with truncated l_2-norm as a loss function, that was run 1 million iterations. The spatial threshold for inliers was set to 20 mm.

Non-rigid transformation

The non-rigid registrations were run with `reg_f3d` with default values. The precision of the transformations was evaluated by computing the Jaccard index of the pairwise registrations.

5.3.2 Feature clusters

An agglomerative clustering algorithm based on the complete-linkage criteria ϵ_s was implemented. The spatial threshold was set to 10 mm for both hearts and brains. An überatlas based on all remaining atlases where constructed for each of the images. The mean atlas was randomly picked out of the remaining atlases. A sorted distance matrix with respect to the descriptor distance was constructed and sorted. Only the $3 \times M$ nearest neighbors to each of the transformed feature points were taken into account, where M is the total amount of atlases. This restriction was done in order to speed up the training.

Data set	**Silver standard**
Hearts	0.9715 ± 0.0562
Brains	0.7612 ± 0.0095

Table 6.1: *The table contains the pairwise Jaccard index of the silver standard transformations of the atlases (Mean ± Standard deviation).*

Data set	**Feature clusters**	**Features total**
Hearts	19706 ± 491	341640 ± 4136
Brains	30777 ± 2672	189050 ± 880

Table 6.2: *The table contains the amount of feature clusters of the atlases and the total amount of features used in order to create the feature clusters (Mean ± Standard deviation).*

Chapter 6

Results

6.1 Training

In order to be able to evaluate the performance of the training of the algorithm, the accuracy of the transformations that represent the silver standard is presented. In Table 6.1 the mean and standard deviation of the Jaccard index of all registrations performed according to Section 4.3.1 are given account for.

In Table 6.2 the mean and standard deviation of the amount of feature clusters produced by the algorithm explained in Section 4.3.2 is compared to the total amount of features.

Figure 6.1 and 6.2 depict an example of six different, randomly chosen feature clusters for the heart and brain respectively, depicted in the mean atlas with the mean atlas coordinates. The corresponding features are also plotted in eight other atlas images with its original coordinates. The eight other atlases contain all or a subset of the feature points present in the six clusters. The images are chosen in order to show the atlases that contain the most feature points present in the clusters.

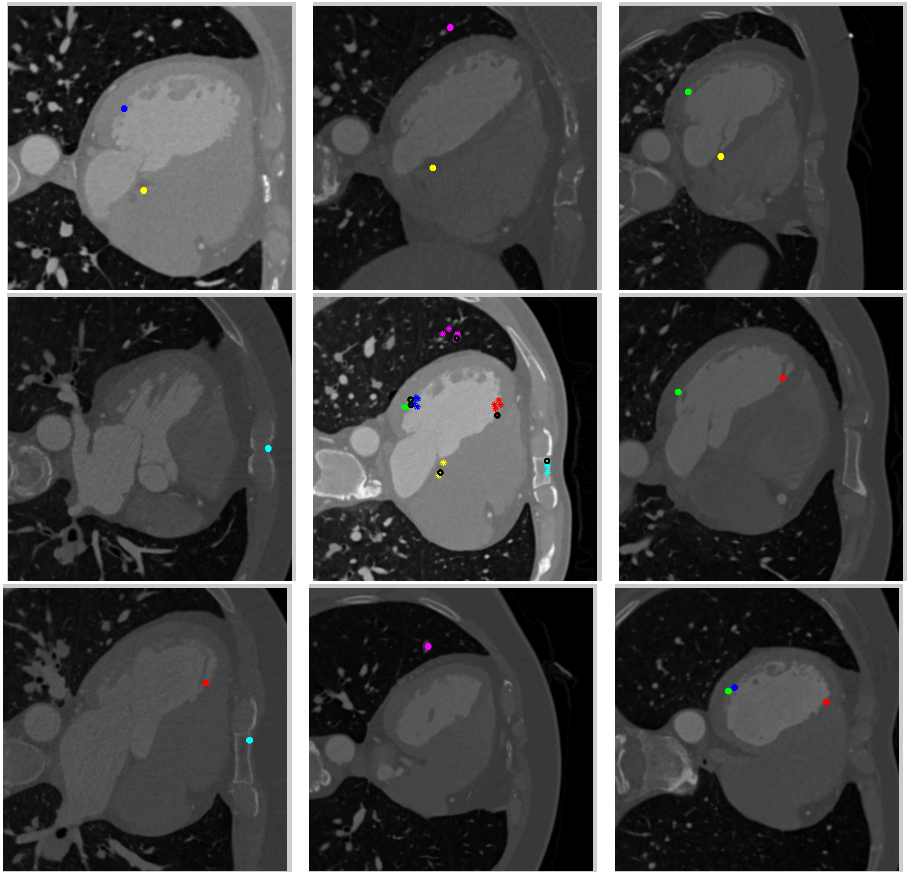

Figure 6.1: *Six different feature clusters (marked with six different colors). The heart in the middle is the mean atlas. In the mean atlas, the mean atlas coordinates of the feature points belonging to the same cluster are marked with the same color. The black circles correspond to the coordinates of the feature points extracted from the mean atlas. The eight surrounding figures depict a sample of the original images represented in the clusters and the feature points belonging to the clusters with its original coordinates. If a color is missing, it means that the corresponding cluster does not contain a feature extracted from the corresponding image.*

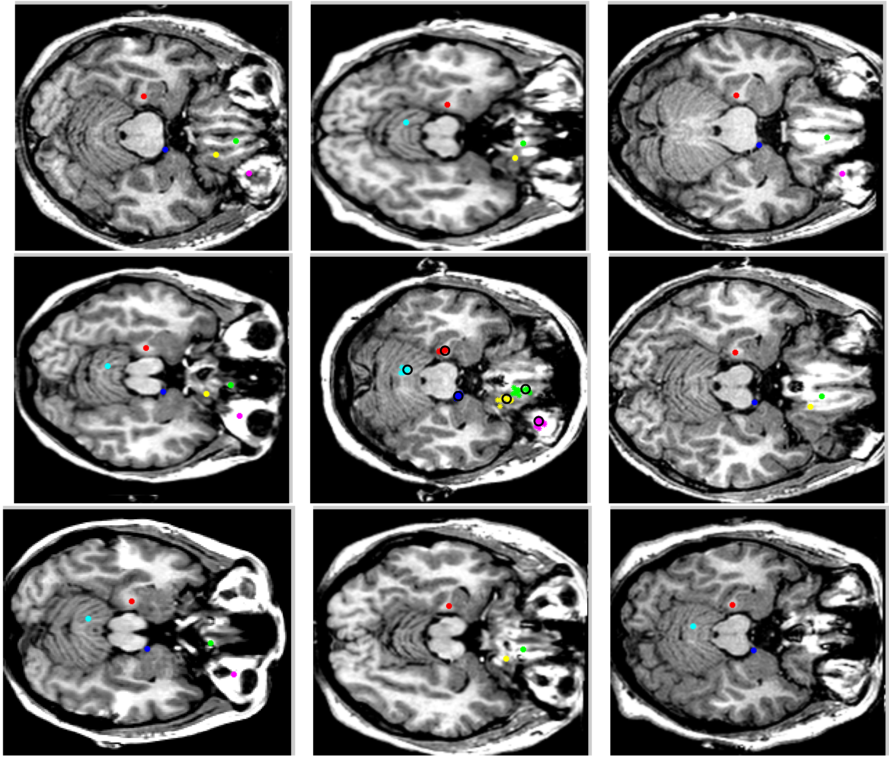

Figure 6.2: *Six different feature clusters (marked with six different colors). The brain in the middle is the mean atlas. In the mean atlas, the mean atlas coordinates of the feature points belonging to the same cluster are marked with the same color. The black circles correspond to the coordinates of the feature points extracted from the mean atlas. The eight surrounding figures depict a sample of the original images represented in the clusters and the feature points belonging to the clusters with its original coordinates. If a color is missing, it means that the corresponding cluster does not contain a feature extracted from the corresponding image.*

Method	**Affine**	**Non-rigid**
Intensity	0.6209 ± 0.1528	0.7737 ± 0.1388
$\epsilon =$ **10 mm**		
Features	0.7746 ± 0.0840	0.8314 ± 0.0832
Überatlas	0.7817 ± 0.0917	0.8320 ± 0.0936
$\epsilon =$ **50 mm**		
Features	0.6984 ± 0.1054	0.8291 ± 0.0449
Überatlas	0.8006 ± 0.0577	0.8150 ± 0.0577

Table 6.3: *The table contains the Jaccard index of the pairwise registrations of the heart (Mean ± Standard deviation).*

Method	**Affine**	**Non-rigid**
Intensity	0.4673 ± 0.0258	0.5654 ± 0.0193
$\epsilon =$ **10 mm**		
Features	0.4710 ± 0.0246	0.5022 ± 0.0167
Überatlas	0.4737 ± 0.0242	0.5212 ± 0.0171
$\epsilon =$ **50 mm**		
Features	0.4604 ± 0.0270	0.4749 ± 0.0189
Überatlas	0.4688 ± 0.0253	0.5069 ± 0.0186

Table 6.4: *The table contains the Jaccard index of the pairwise registrations of the brain (Mean ± Standard deviation).*

Method	**Affine**	**Non-rigid**
Intensity	0.7206 ± 0.1044	0.8779 ± 0.0800
$\epsilon =$ **10 mm**		
Features	0.8531 ± 0.0349	0.9104 ± 0.0347
Überatlas	0.8519 ± 0.0362	0.9075 ± 0.0329
$\epsilon =$ **50 mm**		
Features	0.7197 ± 0.0533	0.9115 ± 0.0254
Überatlas	0.8458 ± 0.0277	0.8814 ± 0.0440

Table 6.5: *The table contains the Jaccard index of the multi-atlas based segmentations of the heart (Mean ± Standard deviation).*

Method	**Affine**	**Non-rigid**
Intensity	0.5705 ± 0.0246	0.6739 ± 0.0141
$\epsilon =$ **10 mm**		
Features	0.5721 ± 0.0224	0.6362 ± 0.0151
Überatlas	0.5724 ± 0.0221	0.6246 ± 0.0171
$\epsilon =$ **50 mm**		
Features	0.5572 ± 0.0246	0.6268 ± 0.0154
Überatlas	0.5646 ± 0.0231	0.6118 ± 0.0183

Table 6.6: *The table contains the Jaccard index of the multi-atlas based segmentations of the brain (Mean ± Standard deviation).*

Method	**Correspondences**	**SSI**	**SSO**
Features	3434 ± 541	151 ± 77	3283 ± 515
Überatlas	3489 ± 954	183 ± 88	3307 ± 909

Table 6.7: *The table contains the amount of detected correspondences of the pair-wise registrations of the heart. Also, it presents the amount of inliers and outliers according to the silver standard (Mean ± Standard deviation).*

Method	**Correspondences**	**SSI**	**SSO**
Features	2275 ± 181	536 ± 74	1739 ± 157
Überatlas	5559 ± 585	706 ± 75	4852 ± 576

Table 6.8: *The table contains the amount of detected correspondences of the pair-wise registrations of the brain. Also, it presents the amount of inliers and outliers according to the silver standard (Mean ± Standard deviation).*

6.2 Segmentations

In order to be able to evaluate the performance of the algorithm compared to Baseline 1 and 2, the accuracy of the segmentations of each of the three methods is presented. In Table 6.3 and 6.4 the mean and standard deviation of the Jaccard index of all pair-wise registrations are given account for. The mean and standard deviation of the Jaccard index of the multi-atlas segmentations are presented in Table 6.5 and 6.6.

6.3 Inliers and outliers

In order to compare the feature-based registration method (Baseline 2) and the überatlas registration, the amount of detected inliers and outliers in pairwise registration is presented in Table 6.9 and 6.10 for the heart and brain, respectively.

By computing the euclidean distance between the mean atlas coordinates of the detected correspondences, inliers and outliers in the registration, one may evaluate whether they correspond to an inlier or an outlier according to the silver standard. A silver standard inlier has a distance less than 10 mm between the mean atlas coordinates, otherwise it is marked as an outlier. Table 6.7 and 6.8 present the amount of silver standard inliers and outliers among the detected correspondences. Finally, Table 6.11 and 6.12 present the percentage of the correspondences correctly and incorrectly marked as inliers/outliers.

Method	**Correspondences**	**Detected inliers**	**Detected outliers**
$\epsilon =$ **10 mm**			
Features	3434 ± 541	664 ± 239	2770 ± 479
Überatlas	3489 ± 954	584 ± 254	2905 ± 833
$\epsilon =$ **50 mm**			
Features	3434 ± 541	2287 ± 453	1147 ± 294
Überatlas	3489 ± 954	1293 ± 431	2196 ± 697

Table 6.9: *The table contains the amount of detected correspondences of the pair-wise registrations of the heart. Also, it presents the amount of inliers and outliers according to the registration (Mean ± Standard deviation).*

Method	**Correspondences**	**Detected inliers**	**Detected outliers**
$\epsilon =$ **10 mm**			
Features	2275 ± 181	1809 ± 159	466 ± 93
Überatlas	5559 ± 585	2707 ± 536	2851 ± 473
$\epsilon =$ **50 mm**			
Features	2275 ± 181	2244 ± 180	31 ± 11
Überatlas	5559 ± 585	3599 ± 869	1960 ± 649

Table 6.10: *The table contains the amount of detected correspondences of the pair-wise registrations of the brain. Also, it presents the amount of inliers and outliers according to the registration (Mean ± Standard deviation).*

	Inliers		Outliers	
Method	**Correct**	**Incorrect**	**Correct**	**Incorrect**
$\epsilon =$ **10 mm**				
Features	0.83 ± 0.13	0.17 ± 0.13	0.84 ± 0.05	0.16 ± 0.05
Überatlas	0.58 ± 0.13	0.42 ± 0.13	0.86 ± 0.05	0.14 ± 0.05
$\epsilon =$ **50 mm**				
Features	1.00 ± 0.00	0.00 ± 0.00	0.35 ± 0.07	0.65 ± 0.07
Überatlas	0.66 ± 0.09	0.34 ± 0.09	0.64 ± 0.08	0.36 ± 0.08

Table 6.11: *The table contains the percentage of detected inliers and outliers in registration of the heart correctly/incorrectly labeled according to the silver standard (Mean ± Standard deviation).*

	Inliers		Outliers	
Method	**Correct**	**Incorrect**	**Correct**	**Incorrect**
$\epsilon =$ **10 mm**				
Features	0.99 ± 0.01	0.01 ± 0.01	0.26 ± 0.04	0.74 ± 0.04
Überatlas	0.87 ± 0.05	0.13 ± 0.05	0.57 ± 0.07	0.43 ± 0.07
$\epsilon =$ **50 mm**				
Features	1.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.01	0.98 ± 0.01
Überatlas	0.88 ± 0.05	0.12 ± 0.05	0.39 ± 0.12	0.61 ± 0.12

Table 6.12: *The table contains the percentage of detected inliers and outliers in registration of the brain correctly/incorrectly labeled according to the silver standard (Mean ± Standard deviation).*

Sub-method	**Intensity**	**Features**	**Überatlas**
Feature detection	-	40.85	40.85
Feature matching	-	128.00	8.05
Affine transformation	20026.00	53.45	0.15
Non-rigid transformation	36955.00	141.25	120.90
Total	56981.00	363.55	169.95

Table 6.13: *Runtime for the sub-steps of the online algorithm, and the total runtime, for an average multi-atlas based segmentation of the heart (using 19 atlases). Sub-steps as resampling of the images and multi-atlas voting are excluded. (Time in seconds)*

Sub-method	**Intensity**	**Features**	**Überatlas**
Feature detection	-	18.77	18.77
Feature matching	-	24.40	5.23
Affine transformation	1827.00	163.03	0.63
Non-rigid transformation	2900.00	27.53	23.47
Total	4727.00	233.73	48.1

Table 6.14: *Runtime for the sub-steps of the online algorithm, and the total runtime, for an average multi-atlas based segmentation of the brain (using 29 atlases) Sub-steps as resampling of the images and multi-atlas voting are excluded. (Time in seconds)*

6.4 Runtime

Finally, the runtimes of an average multi-atlas segmentation are presented in Table 6.13 and 6.14. Though, parts of the algorithm such as resampling the image and computing the multi-atlas segmentations out of the pair-wise segmentations are not presented since these sub-steps are the same for all the three methods. Also, only online substeps are presented. That is, feature detection and matching refer to the detection and matching of feature points in the target image to already computed feature points/feature clusters. The feature points/feature clusters in the source images are computed offline.

Chapter 7

Discussion

7.1 Training

7.1.1 Silver standard

By studying Table 6.1, one may conclude that the proposed method of registering atlases to each other in order to produce a silver standard works well for the heart, and worse for the brains.

One could argue that a Jaccard index of 0.9715 should be high enough in order to map features to each other with a large precision. Though, the standard deviation is quite large, 0.0562, which implies that there are some transformations that are not good enough in order to represent an approximation of the "ground truth". Though, the result could hopefully be improved by optimizing the choice of weighting of the labeling, α, with respect to the Jaccard index. Another aspect that could be taken into account is to reconsider the choice of cost function. When training the silver standard, one could argue that the chosen similarity measure, a bi-variate normalized mutual information, is non-intuitive. Firstly, one would probably want to use a cost function that treats the image and the labeling independently. Secondly, one should probably consider comparing with other functions such as least squares. Both independent normalized mutual information and least squares are implemented by `Niftyreg` and could easily be tested.

The silver standard training of the brains is more disappointing. It is uncertain that a Jaccard index equal to 0.7612 could be considered high enough in order to represent a silver standard at all. Apart from the propositions of improvements above considering the hearts, there are additional improvements that be considered due to the multi-labelings in the brain atlases. In the training, the image and the labeling compose a 4D image where the image represents the first layer and the labeling the second layer. When representing all different labels as one layer, an optimization of a cost function for all different labels at once is performed. A more intuitive approach would be to let all different labels represent one layer each. That is, there would be 85 layers in total (1 image + 1 background label + 83 region labels). Though, this approach would take a huge amount of time and may not be practical in reality.

7.1.2 Feature clusters

As can been studied in Table 6.2, as well as in Figure 7.1 and 7.2, the clustering algorithm produces a significant greater amount of clusters in the brains than in the hearts, even though the initial amount of feature points are larger in the hearts. This may be due to the fact that there are relative few good feature points that could be used as correspondences in the heart images compared to the brain images. Though, it could also mean that it is harder to distinguish between correspondences when it comes to the brains. Moreover, even though the original feature points are spatially distributed more or less uniformly, the feature clusters do not exhibit the same behaviour. This is most evident for the hearts, see Figure 7.1.

Both these issues may partly be explained as a result of the insufficient silver standard registration discussed in the previous section, which limits was most evident for the brain. Though, in order to evaluate whether a cluster is formed correctly or not, one would need to manually evaluate all feature points and feature clusters, which would be a tiresome work and unrealistic in practice. Furthermore, in order to do a manual evaluation properly, one would probably need to consult a physician. A better suggestion could be to evaluate the feature clusters by introducing a statistical model that describes the spatial distributions of the feature points, and how that changes with clustering algorithm.

An attempt to evaluate whether the überatlas registration produces more or less than ordinary feature registration (Baseline 2) is presented in Table 6.7 and 6.8. Here, all correspondences found in the registrations respectively are evaluated with help of the silver standard transformations. According to this, the überatlas method succeeds to include more inliers, though it produces more outliers as well. This is most evident for the brains, where an increase of barely 200 inliers is probably not a justification of an increase of more than 3000 outliers. The situation is more positive for the hearts.

In order to rely less on the silver standard transformations when forming the clusters, one could introduce other criteria than the spatial distance between the mean atlas coordinates. For instance, one approach was tested where the clustering was based on the distance between the descriptors between the feature points. More specificly, the complete-linkage criterion was based on a comparison between the maximum descriptor distance within the potential cluster and the minimum descriptor distance between the cluster and the remaining features. This approach was tested both in combination with a spatial criterion as well as without. The purpose was to construct a simple statistical model of the descriptor space. Though, clustering based on descriptor distance did produce very few clusters, which is unfortunate when it comes to registering images. The approach could be investigated further, since it would eliminate the dependence of the silver standard transformation that takes a great amount of time to produce and is poor at least for the brains in its current implementation.

Finally, it should be mentioned that clustering was tested with different spatial thresholds (different linkage-criteria), and this parameter could probably benefit from further tuning.

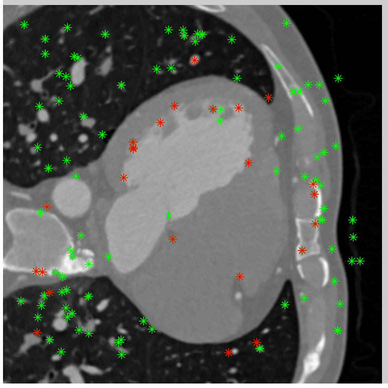

Figure 7.1: *A slice of a CT image of the heart, all detected feature points in the corresponding region marked as green and all feature points included in a feature cluster marked as red.*

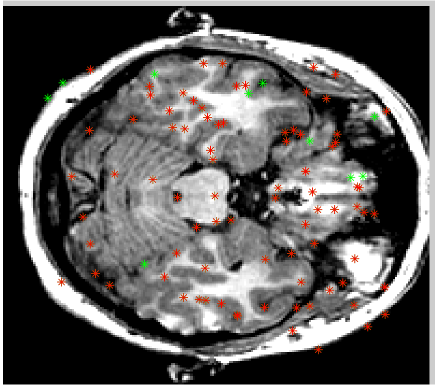

Figure 7.2: *A slice of a MR image of the brain, all detected feature points in the corresponding region marked as green and all feature points included in a feature cluster marked as red.*

7.2 Affine transformation

The Jaccard indexes of the pair-wise registrations in Table 6.3 and 6.4 do exhibit some positive tendencies considering the affine transformations. For both hearts and brains, the slowest intensity-based method performs the worst and the fastest überatlas registration performs the best.

Moreover, the überatlas registration seems to be more insensitive to the spatial threshold defining inliers and outliers. This is positive due to the decreased need of tuning parameters, and may possible be explained by the introduction of the more robust l_1-norm which is less dependant of choice of truncation threshold. By studying Table 6.11 and 6.12. this becomes even clearer. In this table, the percentage of the inliers and outliers marked correctly/incorrectly according to the silver standard may be studied. On the on hand, the original feature-based method do find more correct inliers, but on the other hand the überatlas approach has a much larger capacity of weed out the outliers, especially for the higher thresholds. This is an attractive trait of a robust registration method, and implies that the state-of-the-art method RANSAC is dispensable. Though, it should be noticed that the überatlas method creates more outliers in the first place, as mentioned in the previous section. If these outliers are clearly mismatches, they are of course easier to separate as well.

7.3 Non-rigid transformation

The Jaccard indexes for the pair-wise registrations in Table 6.3 and 6.4 are not as easily analyzed as the affine counterparts. For the hearts, the feature-based methods still perform better than the intensity-based approach, which is not true for the brains. This is probably due to the already mentioned problem of correctly separating detected correspondences as inliers and outliers. This especially evident when studying Table 6.11 and 6.12 and comparing how many outliers that are incorrectly marked as inliers for hearts and brains respectively. The reason why the incorrectly labeled correspondences cause an impairment of performance of the non-rigid segmentation probably depends on the program used in the algorithm, which is not very robust.

Another aspect, maybe even more important, is the opposite of incorrectly labeled outliers, namely the amount of inliers. This aspect could namely be one explanation to why the non-rigid transformation behaves differently for hearts and brains. The non-rigid point-based program demands a great amount of correspondences in order tor work properly, which is probably the reason why the two feature-based methods perform differently for hearts and brains. Like Table 6.9 and 6.10 show, the original feature-based method detects more inliers in the hearts and has a greater improvement of Jaccard index compared to the affine transformation. On the other hand, the überatlas registration produces more final inliers in the brains, which gives a higher relative increase of Jaccard index when applying the non-rigid transformation.

7.4 Multi-atlas segmentations

The Jaccard indexes for the mulit-atlas segmentation in Table 6.5 and 6.6 do not follow the result of the pair-wise registrations entirely. There is not much to say about the intensity-based multi-atlas segmentation; the Jaccard index is the best for the brains and the worst for the hearts like the pair-wise registrations.

More interesting, the Jaccard index of the feature-based multi-atlas segmentations are better than überatlas for both affine and non-rigid transformations for both hearts and brain. That is highly notably, since there is only one out of eight examples of the pair-wise registrations where the feature-based method performs better. This could be due to the already mentioned fact that the clustering algorithm has a tendency to create more feature clusters in some anatomical regions than other, while the original features are more uniformly distributed. This could cause the überatlas registration to estimate consistently good estimations of the segmentation in some anatomical regions, but on the other hand consistently fail for other regions.

Another explanation could be that the clustering algorithm is biased with respect to the choice of mean atlas, which could cause a systematic error. If the silver standard transformation is systematically incorrectly estimated for one mean atlas, this could cause a bias in the resulting überatlas. The original feature-based method does not have this bias towards on of the atlases, which may explain why it performs better in the multi-atlas step.

7.5 Runtime

As can be seen in Table 6.13 and 6.14 there is a huge speed-up between the intensity-based method and the feature-based methods.

The überatlas algorithm succeeds to speed up the registration even more. Most interestingly is of course the enormous speed-up of the affine transformation. It should noticed that both methods are implemented in C++, and therefore comparable. Also, the feature matching time is decreased. It could be decreased even further, since both presented methods are currently implemented in `MATLAB`. When it comes to feature detection there is no change due the use of the same algorithm, though it should be noticed that there is a current implementation that is faster than the presented runtimes. That is, even though it may seem like this could be a substep that needs to be speeded up, the current implementation allows faster detection if OpenMP is installed on the computer. The bottleneck of the algorithm is evidently the non-rigid transformation, which also fails to produce as big improvements as its intensity-based counterpart. In Section 7.6, this will be discussed more thoroughly.

Finally, one should mention that there are other time-consuming parts of the algorithm that is not mentioned. For instance, the resampling program that is used by all methods in order to resample the images are unnecessary slow. This is an issue that needs to be solved in order to make the algorithm practical in clinical care, though outside the scope of this master thesis.

7.6 Further work

As discussed in the previous section, the current method of estimating the non-rigid transformation is insufficient in more than one sense; it takes a lot of runtime, it is highly dependent of a large amount of feature points spread across the whole image and the performance is mediocre at its best. Even though the approach of basing the non-rigid transformation on b-splines seems to work well in the intensity-based case, it takes a huge amount of time.

Since the affine transformation presented in this work performs great, is robust and fast, one would like to use this advantage when estimating the non-rigid transformation. Therefore, it could be of great interest to implement a piecewise affine transformation between the images and let this represent the non-rigid deformation. This could be done by dividing the image into patches, and compute a local affine transformation between all patches. Then, the patches could vote independently of each other when applying the multi-atlas method. This would hopefully be more robust and improve the performance and runtime.

Another approach that could be of interest, is to let the feature points vote for labels directly instead of indirectly using them for transforming the whole image. In that way, one would only need to detect and match feature points, and then let the correspondences decide the labels. This could for example be implemented by letting all feature points decide the label of the image in a predefined neighborhood of the point.

Also, a combination of these methods could be used by constructing and matching clusters of voxels, so called *supervoxels*, and base local affine transformations on correspondences of such items.

Chapter 8

Conclusion

The implemented überatlas registration is a novel, feature-based method that uses co-registration of atlases, clustering of feature points and robust affine transformations with the l_1-norm. The registration method is faster than the two methods used as a comparison, one multi-atlas based registration of the heart (using 19 atlases) takes 2.8 minutes compared to 6.1 and 949.7 (15.8 hours!) minutes respectively. The speed-up of the registration compared to the original feature-based method is even more evident when it comes to brains, 0.8 minutes compared to 3.9 minutes. Even though the speed-up may seem small, the überatlas registration reduces the runtime with a half or even more, which could be of great use in for instance clinical care.

Moreover, the affine registration performed by the implemented algorithm is more robust and detects a greater percentage of outliers. Even more, it succeeds to produce better segmentation results with respect to Jaccard index and has a lower standard deviation. Also, it is not as sensitive to parameters. The non-rigid registration do not cause a significant change or improvement of the registration results, and needs to be improved or changed totally in order to produce relevant results, especially when it comes to the brains. For instance, the b-splines could be replaced by a piece-wise affine method, since the current affine transformation has proved to work excellent so far.

Th training of the algorithm seems to work sufficient good for the hearts, and worse for the brains. The agglomerative clustering algorithm is standard, and could easily be improved and tuned. The silver standard transformations needed for the clustering is sufficient for the hearts, but poor for the brains. Other methods such as adding more layers to the 4D atlas image, changing cost function or method entirely could probably improve the accuracy. More specific, the distribution of the feature clusters compared to the original feature points needs to be investigated in order to achieve a more uniform distributed überatlas. This would probably improve the result of the multi-atlas segmentation as well, since the segmentation accuracy of different anatomical regions would be more consistent.

Bibliography

[1] H. Bay, T. Tuytelaars, and L. Van Gool. "SURF: Speeded up robust features". In: *Computer Vision and Image Understanding* 110 (2008), pp. 346 – 359.

[2] A. Blake and A. Zisserman. "Visual reconstruction". In: *MIT Press* 225 (1987).

[3] R. Chartrand and W. Yin. "Iteratively reweighted algorithms for compressive sensing". In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (2008).

[4] D. Dey et al. "Automated algorithm for atlas-based segmentation of the heart and pericardium from non-contrast CT". In: *Proc SPIE.* 7623:762337 (2010).

[5] M.A. Fischler and R.C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24 (1981), pp. 381 – 395.

[6] G. Gill, M. Toews, and R.R. Beichel. "Robust initialization of active shape models for lung segmentation in CT scans: A feature-based atlas approach". In: *International Journal of Biomedical Imaging* 2014 (2014).

[7] A. Hammers et al. "Three dimensional maximum probability atlas of the human brain, with particular reference to the temporal lobe". In: *Hum. Brain Mapp.* 19 (2003), pp. 224 – 247.

[8] R.A. Heckemann et al. "Automatic anatomical brain MRI segmentation combining label propagation and decision fusion". In: *NeuroImage* 33 (2006), pp. 115 – 126.

[9] F. Khalifa et al. "State-of-the-Art Medical Image Registration Methodologies: A Survey". In: *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies.* Springer Science+Business Media, 2011, pp. 235 – 280.

[10] H. A. Kirisli et al. "Fully automatic cardiac segmentation from 3D CTA data: a multi-atlas based approach". In: *Proc. of SPIE* 7623:762305 (2010).

[11] S. Lee, G. Wolberg, and S.Y. Shin. "Scattered data interpolation with multilevel B-splines". In: *IEEE Transaction on visualization and computer graphics* 3 (1997).

[12] D.G. Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of International Conference on Computer Vision* (1999), pp. 1150 – 1157.

[13] M. Modat et al. "Fast free-form deformation using graphics processing units". In: *Computer methods and programs in biomedicin* 98 (2009), pp. 278 – 284.

[14] A. Norlen. "Automated segmentation of the pericardium using a fetaure-based multi-atlas approach". MA thesis. Faculty of Engineering, Lund University, 2014.

[15] S. Ourselin, R. Stefanescu, and X. Pennec. "Robust registration of multi-modal images: Towards real-time clinical applications". In: *MICCAI 2002, LNCS* 2489 (2002), pp. 140 – 147.

[16] S. Ourselin et al. "Reconstructing a 3D structure from serial histological sections". In: *Image and vision computing* 19 (2001), pp. 25 – 31.

[17] D.L. Pham, C. Xu, and J.L. Prince. "Current methods in medical image segmentation". In: *Annual Review of Biomedical Engineering* 2 (2000).

[18] P. J. Rousseeuw. "Least Median of Squares Regression". In: *Journal of the American Statistical Association* 79 (1984).

[19] D. Rueckert et al. "Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images". In: *IEEE transactions on medical imaging* 18 (1999), pp. 712–721.

[20] L. Svarm et al. "Improving robustness for inter-subject medical image registration using a feature-based approach". In: *Submitted to International Symposium on Biomedical Imaging* (2015).

[21] B. Zitova and J. Flusser. "Image registration methods: a survey". In: *Image and vision computing* 21 (2003), pp. 977–1000.