Automatic segmentation of hippocampal substructures

Santiago Flores, G.

Award date:
2012

Thesis Report

Automatic Segmentation of Hippocampal Substructures

Gerardo Santiago Flores
email: g.santiago.flores@student.tue.nl
Student ID 0757324

Technische Universiteit Eindhoven

Prof. Dr. Gerard de Haan

Technische Universiteit Eindhoven

Philips Research Eindhoven

Dr. Radu Jasinschi and Dr. Octavian Soldea

Philips Research Eindhoven

July 2, 2012

Abstract

Segmentation of brain structures is an important component in the field of medical imaging because it provides support for diagnosis and treatment as well as therapy evaluation guidance. Modern technology allows acquisition of MR images with very high resolution up to 0.3 mm of spacing between voxels. In this context, we propose a fully automatic segmentation method focused on substructures of the Hippocampal Formation. For this purpose, we present the development and implementation of three relevant contributions. First, we introduce a fast registration method based on moments. Second, we present a multi-level initialization scheme for obtaining initial contours required for segmentation; this procedure is fully automatic and it requires a training set of images with manual annotations of the substructures of interest. Third, we introduce a new segmentation algorithm, which is based on active contours driven by moments prior. We minimize an energy cost function in order to get optimal segmentation employing signatures based on moments. We compared our results to state-of-the-art tools and show significantly improvement in time performance. In addition, we tested our method with patients with Alzheimer Disease.

Keywords: Substructures segmentation, Hippocampal formation, Moments prior, Active Contours Initialization

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my company supervisors, Dr. Radu Jasinschi and Dr. Octavian Soldea, whose guidance and tutelage considerably help me in the development of my research work. I have learnt plenty of things from them. In the technical side, their expertise in the field of medical imaging were fundamental to achieve the goals of this project and in the personal side, I received valuable advice from them that made me considered both as my friends.

I would also like to thank Prof. Dr. Gerard de Haan, not only for giving me the opportunity to work at Philips Research, but also for all the insightful contributions and suggestions that he made to improve my work.

My special thanks to all my fellow interns for all the great time that we spent in the office, it was a great experience to work in such a nice environment. Very special thanks to my best friend Nitejat, she has always been there for me when I need her. Her advice and patience have assisted me for many years.

Finally, I would like to thank my parents and my brother for all the support that I have received from them in my entire life. Everything I have done has been thanks to my family. All the achievements that I have made are dedicated to them.

Contents

List of Figures

CHAPTER 1

Introduction

Segmentation of brain structures from MR images is an important component in the field of medical image analysis due to the fact that it is used in plenty of applications such as diagnosis, surgical guidance and therapy evaluation [2]. For example, segmentation of hippocampal formation is motivated by the fact that hippocampal atrophy is known to occur at early stages of Alzheimer disease (AD) [3] [4] [5]. Hence, the analysis of hippocampal structure provides important information at the diagnosis phase. Figure 1.1 is an example of brain segmentation. In Figure 1.1 different colors represent different structures, left-hippocampus is marked with blue, left-lateral-ventricle is green, left-putamen is shown in red.

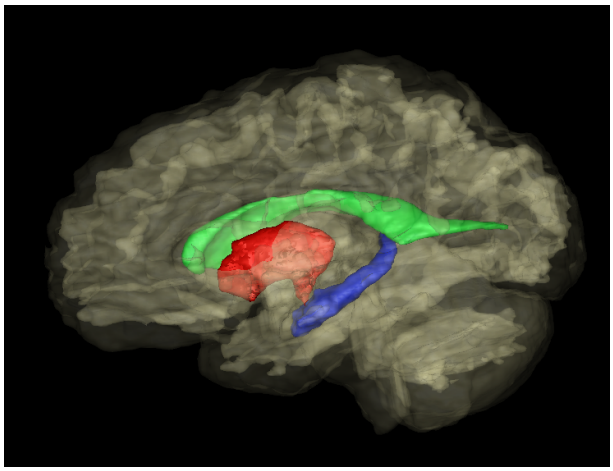


Figure 1.1: Example of brain segmentation

Although segmentation can be performed manually, it is a tedious and time-consuming task which does not provide entirely replicable results [2]. Moreover, continuous use of manual agility and hand-eye coordination makes the expert rater quickly fatigued [3]. Thus segmentation of a large number of MR images by an expert is not practical. Therefore, automatic and semiautomatic segmentation tools have begun to emerge [2], [6], [7], [8]. On one hand, these methods are based on training sets from ground truth annotations made by experts and their results are still not comparable to manual segmentation. On the other hand, the speed provided by (semi)automatic methods offers the opportunity to evaluate a large number of patients. This

information can be used to create statistics over trends of diseases' evolution from which medical diagnosis and treatment benefit.

Automatic segmentation results are strongly influenced by the quality of the images. On one hand, low contrast areas and different acquisition parameters have a negative impact on annotations results from automatic methods. On the other hand, the modern resolution of MR images, with physical spacing between voxels of 0.3 mm. , provide extremely good level of detail [1]; this allows for segmentation of substructures. In this context, we propose to segment the hippocampal formation at the subfield level.

1.1 Problem Description

In this thesis, we treat the problem of fully automatic segmentation of Hippocampal Formation (HF). The final goal is to develop and implement a tool that is able to automatically segment hippocampal substructures. This problem implies major challenges: First, due to the very small size of the analyzed substructures, information may be lost during the procedure that we employ. Second, we aim at providing a robust method in terms of performance and quality of the results. In addition, we also consider computational resources limitations in order to make our tool optimized. Furthermore, an efficient solution is required in terms of speed and precision, these parameters will be compared to existing methods

1.2 Related Work

In this section, we discuss some methods previously proposed in order to solve the problem of segmentation of HF at the substructure level. To the best of our knowledge, there are two publications that were proposed to solve substructure segmentation of **HR!**. We discuss them briefly.

1. **Nearly automatic segmentation of hippocampal subfields in *in vivo* focal T2-weighted MRI [8]**

   In this publication, the authors proposed a segmentation method which is based on a combination of multi-atlas segmentation, similarity-weighted voting and a learning-based bias correction technique. The method is not fully automatic due to the fact that the authors manually divided MRI slices of the HF into three main partitions, namely Head, Tail and Body. After this stage the fully automatic method comprises four steps.

   In the first step, the authors create an initial segmentation of the hippocampal subfields by using a multi-atlas segmentation technique. Given a set of training MR Images with manual annotations of HF subfields, the method produces a candidate segmentation for each of the patients in the training set. In order to combine these candidate segmentation into a single segmentation, the authors used a voting scheme, where the candidate with higher similarity metric influences more in the final consensus segmentation image. In this stage, the authors make use of both types of weighted images T1 and T2, since, according to them, using both types of images provides complementary information. The authors named this step as Multi-atlas segmentation and voting (MASV).

   In the second step, the authors use the initial segmentation produced in step 1 in order to train AdaBoost classifiers. The goal of these classifiers is to detect and correct mislabeled voxels. The classifiers use image texture, initial segmentation results and spatial location

features. In the third step, the authors used MASV to produce initial segmentation for each of the patients in the testing set. Finally in the fourth step, the authors used the classifiers trained in step 2 in order to improve the results from step 3.

One important point from this method is that the ROI that surrounds the HF is fixed. The authors proposed to use different reference spaces for each of the HF namely Left and Right. Each of these reference spaces has dimensions of 40 x 55 x 40 mm^3, this reference space is used along the entire procedure.

The results in terms of DC are around 0.873 for DG, 0.875 for CA1, 0.770 for Sub and 0.787 for EC. We consider that the manual intervention plays an important role in getting this high value results. The fastest computation time reported in this publication is 19 hours for the training time and 8 hours for the testing time for a training set of 20 subjects and a testing set of 10, using a cluster of 8 CPUs with 8 cores each.

2. **Automated Segmentation of Hippocampal Subfields From Ultra-High Resolution In Vivo MRI [1]**

In this publication, the authors proposed a fully automatic method to solve the HF subfields segmentation problem. This method is based on a Bayesian modeling approach. The authors employed a *likelihood* distribution to assign a single neuroanatomical to each of the voxels in the image.

In order to build the likelihood distribution, the authors proposed a tetrahedrical mesh which covers the ROI. This mesh is deformed from its reference position by a Markov random field model. The authors find the ROI by using affine registration. Once a ROI is found, they employed a tissue classification algorithm to discard all voxels which do not belong to any of the subfields. Finally they employ the likelihood model to the voxels that remain available. Similar to [8], in [1] the authors proposed a fixed ROI which has dimensions 94 x 66 x 144 voxels.

The results obtained in terms of DC are: 0.68 for DG, 0.62 for CA1 and 0.74 for Sub. Note that the authors employed a different protocol than [9], therefore no result for EC is reported. For each patient, the computation time of the algorithm was about 5 hours.

1.3 Report Organization

The organization of this report is as follows: In Chapter 2, we introduce some background required to defined the method proposed. We briefly explain the structure of interest as well as the mathematical definition of *moments*, *registration* and the metric that we use to validate our results. In Chapter 3, we describe the proposed method to solve substructures segmentation. In Chapter 4, we present the results obtained with the proposed method, as well as a comparison with existing methods. Finally, in Chapter 5 we present conclusions and future work.

CHAPTER 2

Background

In this chapter, we explain the background required for the segmentation method that we propose. The background explain here consists of Hippocampal Formation, moments and pose information, registration and one validation metric.

2.1 Hippocampal Formation

In this section we describe the motivation behind the study of the HF as well as a brief description of its anatomy.

2.1.1 Motivation

The analysis of the HF is important due to the fact that this structure takes part in many functions of the brain: following [5], the hippocampus is involved in propositional memory, and neuronal and synaptic plasticity. In addition, due to the pathological processes of conditions such as epilepsy and Alzheimer's disease, that occur in this formation, its analysis offers the opportunity to provide novel diagnosis.

2.1.2 Hippocampal Anatomy

The hippocampal formation is a structure from the brain which is formed by the following substructures: the dentate gyrus (DG), hippocampus, subiculum(Sub), presubiculum(Pre), parasubiculum(Para), and entorhinal cortex (EC) [5]. The hippocampus has three subdivisions: CA1, CA2, and CA3, (CA means cornu ammonis). Figure 2.1 shows the hippocampal formation.

In this work, we focused on four substructures, namely DG, CA, EC and Sub. We merge CA1, CA2, CA3 into a single substructure CA for validation purposes. In Figure 2.2, we show left and right HF from coronal view.

2.2 Moments and Pose Information

In this section we describe the concepts of *moments* and *pose information* which are relevant concepts for the segmentation method that we propose.

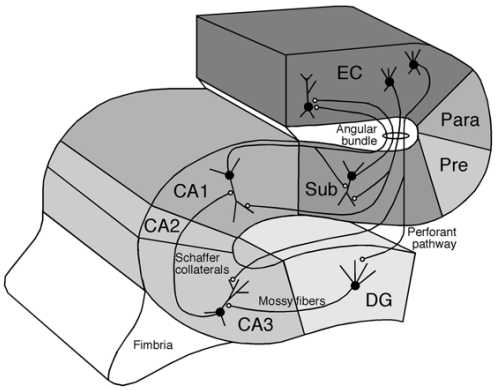


Figure 2.1: Hippocampal Formation. Taken from [5]

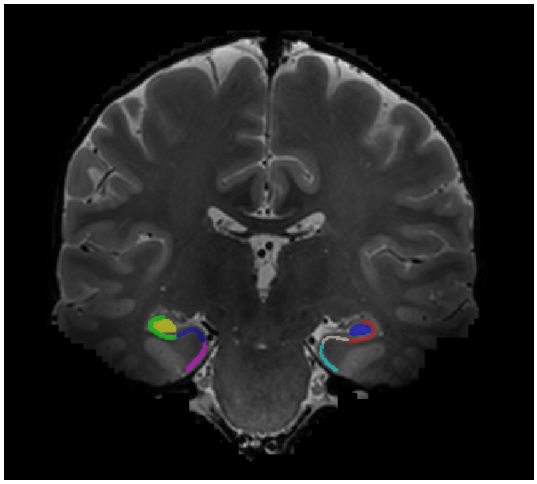


Figure 2.2: Manual Annotation of Hippocampal Formation

2.2.1 Definition of Moments

Following [10], moments are a measurement that expresses how values of a given set are distributed around a reference. For example, moments are used in statistics to represent the distribution of random variables or in mechanics where moments represent the physical distribution of the mass of a body. Considering the grey level of the image as a density distribution function, moments theory can be applied to obtain relevant information from the image, such as its pose information. The moment of order n of a 3D image is defined as:

$$m_{p,q,r} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q z^r f\,(x,y,z)\,dxdydz \tag{2.1}$$

where $n = p + q + r$ defines the order of the moment and $f(x, y, z)$ is an implicit function that defines the intensity value in grey scale at point (x, y, z).

2.2.2 Definition of Pose Information

The pose information of an object is a parameter that indicates the position of it. Following [2], pose of an object is determinate by the location of its center of mass, its size(volume), and the orientation of its principal axis. Formally, we define the pose parameter p of an object I as:

$$p_I = [V, c_x, c_y, c_z, \theta] \tag{2.2}$$

where V is the volume, $c_x\, c_y\, c_z$ are the coordinates of the center of mass and Θ is the orientation of the object.

We use moments computation in order to find the pose information of an object. The volume is defined by:

$$V = m_{0,0,0} \tag{2.3}$$

The center of mass of an object is defined by:

$$c_x = \frac{m_{1,0,0}}{m_{0,0,0}} \quad c_y = \frac{m_{0,1,0}}{m_{0,0,0}} \quad c_z = \frac{m_{0,0,1}}{m_{0,0,0}} \tag{2.4}$$

Second-order moments are used to define object's orientation. We use second-order moments to find the inertia moments which are required to find the inertia tensor. The eigenvectors of the inertia tensor are the principal axes of the object. By knowing the principal axes of the object we can define its orientation. Following [11], inertia moments are expressed as:

$$\begin{aligned}
I^{xx} &= m_{020} + m_{002} - \frac{m_{010}^2}{m_{000}} - \frac{m_{001}^2}{m_{000}} \\
I^{yy} &= m_{200} + m_{002} - \frac{m_{100}^2}{m_{000}} - \frac{m_{001}^2}{m_{000}} \\
I^{zz} &= m_{200} + m_{020} - \frac{m_{100}^2}{m_{000}} - \frac{m_{010}^2}{m_{000}} \\
I^{xy} &= I^{yx} = m_{110} - \frac{m_{100} m_{010}}{m_{000}} \\
I^{xz} &= I^{zx} = m_{101} - \frac{m_{100} m_{001}}{m_{000}} \\
I^{yz} &= I^{zy} = m_{011} - \frac{m_{010} m_{001}}{m_{000}}
\end{aligned} \tag{2.5}$$

The inertia tensor is:

$$\bar{I} = \begin{bmatrix} I^{xx} & -I^{xy} & -I^{xz} \\ -I^{yx} & I^{yy} & -I^{yz} \\ -I^{zx} & -I^{zy} & I^{zz} \end{bmatrix} \tag{2.6}$$

By finding the eigenvectors of the inertia tensor, we obtain the principal axes of the object. The orientation matrix θ of an object I is the matrix formed by the eigenvectors of the inertia tensor $\bar{I}$.
The eigenvalues λ of the matrix $\bar{I}$ are given by:

$$det\,(\bar{I} - \lambda I) = 0 \tag{2.7}$$

The eigenvectors $\bar{v}$ are found with:

$$\bar{I}\bar{v} = \lambda\bar{v} \tag{2.8}$$

Then, we define the orientation matrix θ of the object I as:

$$\theta = \begin{bmatrix} \bar{v_1} & \bar{v_2} & \bar{v_3} \end{bmatrix} \tag{2.9}$$

2.2.3 Discrete Implementation of Moments

In order to compute the moments of an image, we employ a discrete version of equation 2.1, namely:

$$m_{pqr} = \sum_{z=0}^{L-1} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q z^r g(x, y, z) \tag{2.10}$$

where m_{pqr} is the moment of order $n = p + q + r$, L, M and N are the number of sample voxels in each direction, and $g(x, y, z)$ is the intensity value of the voxel located in x, y, z.

The precision of moments computation is proportional to the resolution of the image. Furthermore, in the case of medical imaging, the available discrete data is given by the acquisition quality of the scans. Moreover, we can modify the amount of discrete data by changing the amount of voxels in the image. In this work, we control the amount of voxels used to compute moments by increasing or decreasing the physical space between them. On one hand, by decreasing the physical spacing between voxels, the number of voxels is increased, this means that computation of moments is more precise, however, this also implies more computation time. On the other hand, increasing the physical spacing between voxels decreases the total number of voxels used for moments computation and, as a consequence, the precision of moments. Figure 2.3 illustrates the effect of modifying the spacing between sample voxels.

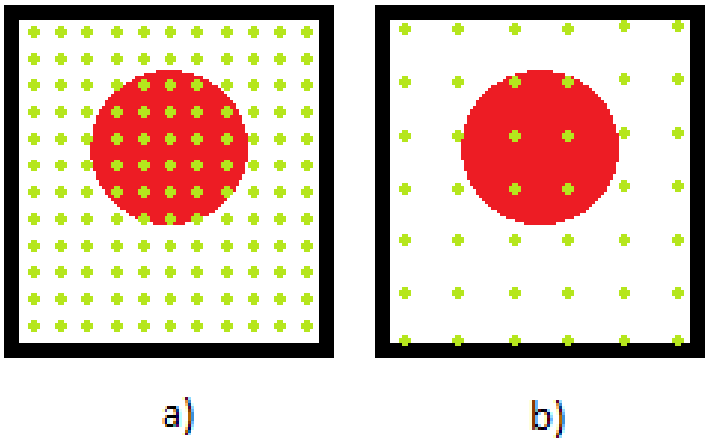


Figure 2.3: Discrete implementation of moments. a) shows discrete image with a certain spacing (X) between sample voxels. b) shows the same image but with a spacing of 2X. In both cases the green points are the points considered for moments calculation

2.3 Registration

In this section, we describe the concept of *registration*, which is an important concept due to the fact that we use registration at the *initialization* phase as well as in the *segmentation* algorithm.

Image registration aims at finding the best transformation by which we align one image (moving) to a reference image (fixed). Aligning images is a required preprocessing step before developing any further image treatment [12]. In this thesis, we employ registration in two main

parts. First, we use registration for aligning new brain images to the training set. By doing this, we obtain the initial contours required for the segmentation algorithm, see Section 3.4. Second, we use registration at the substructure level, in order to obtain relevant information about how a substructure is distributed around the ensemble of substructures, this information is needed for the Segmentation algorithm, see Section 3.5. Figure 2.4 depicts the registration concept.

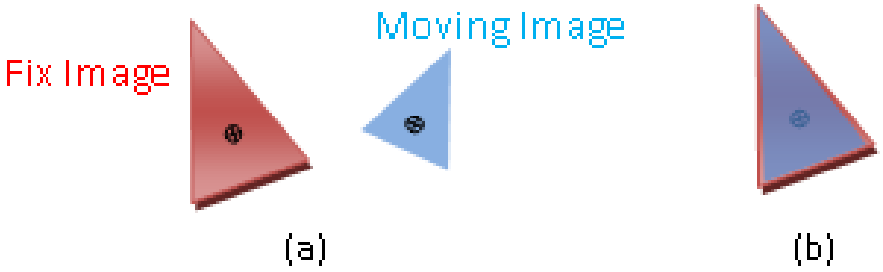


Figure 2.4: Registration Procedure. In (a) images are not aligned. Part (b) shows the result after registration.

Usually, the problem of finding the best aligning transformation is treated as an optimization. For example, in Elastix tool [13], the authors follow the optimization scheme which shown in Figure 2.5. At each iteration a transformation is obtained, this transformation is applied to the moving image. The interpolator is needed in order to obtain intensity values at non-grid positions, then the quality of alignment is obtained by making use of a similarity metric. Finally the optimizer is responsible for optimizing the transformation. In the following sections we give a brief description of some possible types of *similarity metrics* and *transformations*.

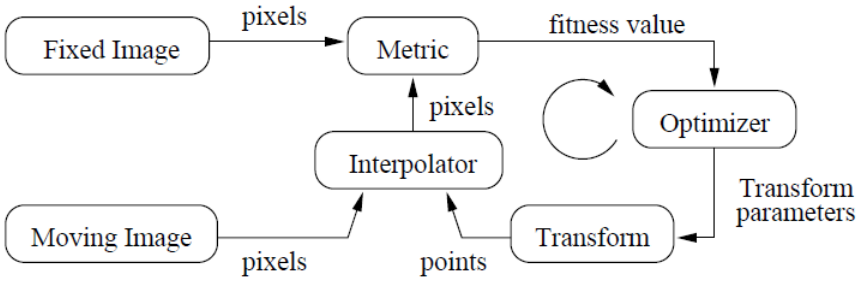


Figure 2.5: Registration Framework. Taken from [14].

2.3.1 Similarity Metrics

As mentioned in Section 2.3, a similarity metric is required in order to evaluate the quality of the alignment. Some possible metrics of this type are:

Sum of Square Differences SDD
SDD is defined as:

$$SDD(I_F, I_M) = \frac{1}{|\Omega_F|} \sum_{x_i \epsilon \Omega_F} \left(I_F\left(x_i\right) - I_M\left(x_i\right)\right)^2 \tag{2.11}$$

where Ω_F is the domain of the fixed image I_F, $|\Omega_F|$ the number of voxels and $I_F(x_i)$ and $I_M(x_i)$ the voxel values of the fixed and moving image respectively.

Normalized Correlation Coefficient NCC
NCC is defined as:

$$NCC(I_F, I_M) = \frac{\sum_{x_i \epsilon \Omega_F} (I_F(x_i) - \bar{I_F})(I_M(x_i) - \bar{I_M})}{\sqrt{\sum_{x_i \epsilon \Omega_F} (I_F(x_i) - \bar{I_F})^2 \sum_{x_i \epsilon \Omega_F} (I_M(x_i) - \bar{I_M})^2}} \tag{2.12}$$

where $\bar{I_F} = \frac{1}{\Omega_F} \sum_{x_i \epsilon \Omega_F} I_F(x_i)$ is the average intensity value of the fixed image and $\bar{I_M} = \frac{1}{\Omega_F} \sum_{x_i \epsilon \Omega_F} I_M(x_i)$ is the intensity average of the moving image.

Mutual Information
Following [14] and [15], MI is defined as:

$$MI(I_F, I_M) = \sum_{m \epsilon L_M} \sum_{f \epsilon L_F} p(f, m) log_2 \left(\frac{p(f, m)}{p_F(f) p_M(m)} \right) \tag{2.13}$$

where L_F and L_M are sets of regularly spaced intensity bin centers, p is the discrete joint probability, and p_F and p_M are the marginal discrete probabilities of the fixed and moving images, obtained by summing p over m and f, respectively. The joint probabilities are estimated using B-Spline Parzen windows:

$$p(f, m) = \frac{1}{\Omega_F} \sum_{x_i \epsilon \Omega_F} w_F(f/\sigma_F - I_F(x_i)/\sigma_F) \times w_M(m/\sigma_M - I_M(x_i)/\sigma_M) \tag{2.14}$$

where w_F and w_M represent the fixed and moving Bspline Parzen windows. The scaling constants σ_F and σ_M must equal the intensity bin widths defined by L_F and L_M.

2.3.2 Transformation Types

There exist a number of possible transformation in order to map the moving image into the fixed one. In this thesis we use the following transformation types:

Rigid transformations Rigid transformations use the same transformation for mapping each of the voxels in the moving image.

Similarity
Similarity transformation is defined as:

$$T(x) = sR(x - c) + t + c \tag{2.15}$$

where t is the translation vector, s is the scaling factor, c is the center of the transformation and R is a rotation matrix. In this type of transformation the image is treated as a rigid body, it can be translated, rotated and scaled. However, the scaling factor is the same in all directions (isotropic).

Affine

Affine transformation is defined as:

$$T(x) = A(x - c) + t + c \tag{2.16}$$

The main difference between the similarity transformation and Affine Transformation is that matrix A has no restrictions. This means that besides rotation we can apply anisotropic scaling factors.

Non-Rigid Transformations

The characteristic of the Non-Rigid Transformations is that different transformations are used for different parts of the image; this allows local deformations.

BSpline

A possible parametrization of a non-rigid transformations is by means of Bsplines, following [16]:

$$T(x) = x + \sum_{x_k \epsilon N_x} p_k \beta^3 \left(\frac{x - x_k}{\sigma} \right) \tag{2.17}$$

where x_k are control points, $\beta^3(x)$ is a cubic multidimensional Bspline polynomial, p_k are the Bspline coefficient vectors, σ is the spacing between control points and N_x is the set of control points support of the Bspline at point x.

The control points x_k form a regular grid with resolution σ. The resolution of the grid influences heavily on the transformation. On one hand, a finer control points grid will focus on small localized transformation. On the other hand, a coarser grid will allow larger displacements in the transformation.

2.4 Validation Metric

In order to evaluate the result of registration and segmentation we use Dice Coefficient DC, which is a common metric for validating segmentation algorithms. DC is defined as:

$$DC(A, B) = \frac{2\,|A \cap B|}{|A| + |B|} \tag{2.18}$$

where A and B are two objects to be compared, and $|X|$ is the volume of the object X.

DC measures the overlap between two sets, in our case volumes. A value of 0 represents no overlap, a value of 1 represents a perfect overlap. The goal of a segmentation algorithm is to obtain DC close to 1, when the results are compared to a manual annotation segmentation.

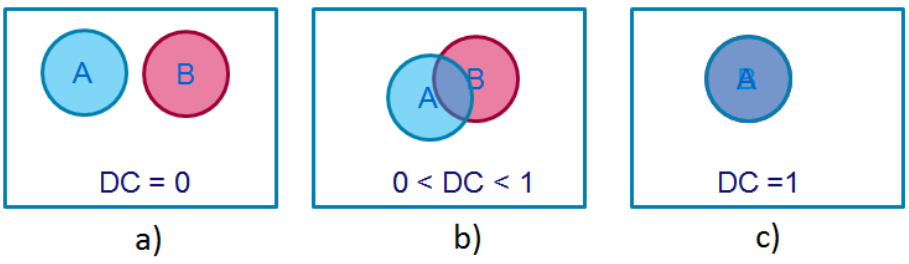


Figure 2.6: Dice Coefficient. a) No overlapping. b) Partial overlapping. c) Complete overlapping

CHAPTER 3

Method proposed

In this section we describe the algorithms that we propose in order to achieve fully automatic segmentation of HF Substructures. We make three relevant contributions. First, we propose a fast *Registration* method which is based on *Moments*. Second, we propose a *Initialization scheme*. Finally we use a *Segmentation Algorithm*, which is based on *Pose Information Prior*.

3.1 Overview of the method

The work flow of the Segmentation Method that we propose is depicted in Figure 3.1. As input, we use an MR volumetric image of the brain. We first apply a preprocessing step in which we remove the skull and non-brain tissue from the image obtaining only the parenchyma, see Figure 3.2. For this preprocessing step we use the tool Brain Extraction Tool BET from Fmrib Software Library FSL.

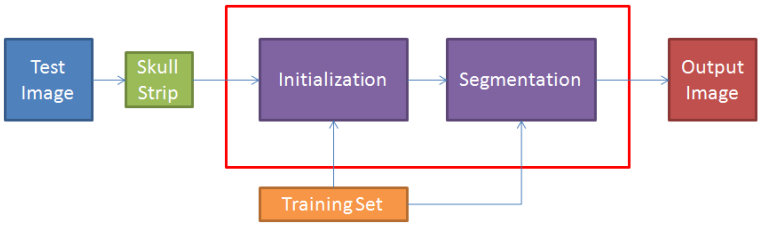


Figure 3.1: Segmentation Method Overview

After the preprocessing, we start with the segmentation algorithm which consists of two main stages, namely Initialization and Segmentation. In both stages we use a Training Set, see 3.2. At the Initialization phase, we create initial contours. We predict the location of these initial contours by using the contours from the Training Set. At the Segmentation stage, we apply the algorithm proposed in order to deform the initial contours in such a way that the segmentation is optimized. The result of this procedure is a volumetric image which contains the segmentation of the substructures of interest.

We implemented the algorithms in C++, using ITK and VTK libraries. For the Affine and BSpline registrations required in the Initialization stage, we use Elastix [13].

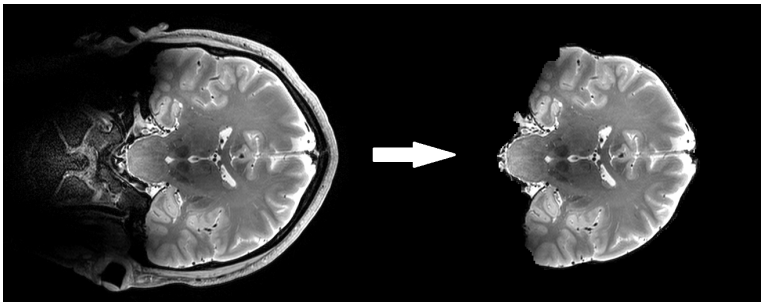


Figure 3.2: Skull Strip

3.2 Training Set

The training set that we employed is a set of MR images of healthy brains with manual annotations of the substructures of the HF, see Figure 3.3. The images were provided by Universitair Medisch Centrum Utrecht (UMC) and Leids Universitair Medisch Centrum (LUMC) These manual annotations were performed by experts following the protocol in described in [9]. Moreover, the training set contains substructure signatures based on moments which describe how each of the substructure is distributed around the ensemble [1] of all substructures combined. These signatures are used in the segmentation method proposed and they are computed using *Moments* of each substructure as related to the ensemble.

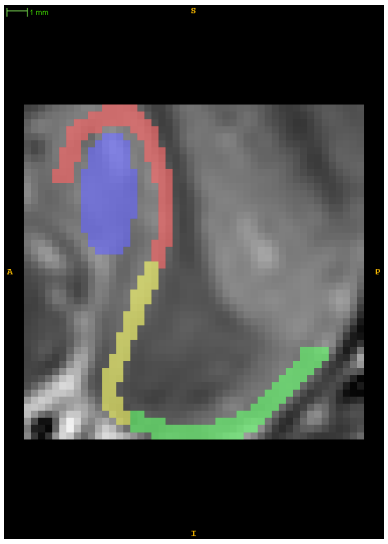


Figure 3.3: Example of training image. Coronal View. Each of the colors represent a different substructure: CA is red; DG is blue; Sub is yellow and EC is green

3.3 Fast Moments Based Registration

As mentioned in Section 2.3, the goal of registration is to find the best transform T to align a moving object to a fixed one. In this context, we propose a fast method to find the parameters of an *affine transformation*, see Section 2.3.2. The method that we propose uses the *pose information* of moving and fixed object, see Section 2.2.2. Because we employ registration at many

[1] We consider the ensemble as a structure which is formed by all the substructures of the HF .

steps of the method proposed, we require a fast registration method to decrease computation time. We use this fast registration method in the Segmentation algorithm, see Section 3.5.

3.3.1 Formal Definition of Registration based on Moments

We define the registration transformation T as:

$$p_M T = p_F \tag{3.1}$$

where p_M is the pose information, see 2.2.2, of the moving object and p_F is the pose information of the fixed object.

The transformation that we use is a concatenation of three parts: translation, rotation and scaling. We obtain the transformation parameters to build the transformation matrix as follows.

3.3.1.1 Translation

Translation is required to align both objects to a common center of mass. Since both centers of mass are known by its pose information parameter, we can define the vector for the translation as:

$$\bar{T} = [c_{Mx} - c_{Fx}, c_{My} - c_{Fy}, c_{Mz} - c_{Fz}] \tag{3.2}$$

where c_M is the center of mass of the moving image and c_F is the center of mass of the fixed image.

3.3.1.2 Rotation

We require a rotation transformation in order to align the orientations of the objects. The orientation of both objects, fixed and moving, is known by their pose parameters: θ_F, θ_M, hence the following equation needs to be solved:

$$\theta_F = R\theta_M \tag{3.3}$$

Rotation transformation is then defined as:

$$R = \theta_F \theta_M^{-1} \tag{3.4}$$

3.3.1.3 Scaling

Scaling is required to obtain images of similar dimensions. We describe two methods to find the scaling factors.
A simple way to obtain the difference of scale is to work with the volumes of the images ($m_{0,0,0}$). Therefore the scale required to register the moving image to the fixed one is given by:

$$s = \sqrt[3]{\frac{m_{000}^M}{m_{000}^F}} \tag{3.5}$$

In Equation 3.5, we consider that the scaling factor is isotropic, i.e., the same in the three axes. However, the scale required in each of the axes can be different; therefore different scale

factors are needed among the different axes. In order to find these different scaling factors, we employ second order moments. Following [10], a scaling change in an image modifies the value of its moments as follows.

Consider $f(x, y, z)$ as the original intensity function of a 3D image, the function that results after scaling by a factor of α on X axis, β on Y axis, and γ on Z axis is given by:

$$f'(x, y, z) = f(\frac{x}{\alpha}, \frac{y}{\beta}, \frac{z}{\gamma}) \tag{3.6}$$

The relation between the moment values of the original and transformed functions is expressed as:

$$m'_{pqr} = \alpha^{1+p}\beta^{1+q}\gamma^{1+r}m_{pqr} \tag{3.7}$$

By considering the second-order moments of both, fixed and moving objects, the following equations are derived:

$$\begin{aligned} m'_{002} &= \alpha\beta\gamma^3 m_{002}, \\ m'_{020} &= \alpha\beta^3\gamma m_{020}, \\ m'_{200} &= \alpha^3\beta\gamma m_{200} \end{aligned} \tag{3.8}$$

where m'_{pqr} are the moments of the fixed image and m_{pqr} are the moments of the moving image.

Thus, from Equation 3.8 α, β and γ can be expressed as a function the second-order moments as follows:

$$\begin{aligned} \alpha &= \left(\frac{(m'_{200})\,(m'_{002})^{\frac{1}{4}}(m'_{020})^{\frac{1}{4}}}{(m_{200})\,(m_{002})^{\frac{1}{4}}(m_{020})^{\frac{1}{4}}} \right)^{\frac{2}{5}}, \\ \beta &= \sqrt[3]{\frac{m'_{020}}{\alpha\gamma m_{020}}}, \\ \gamma &= \sqrt[8]{\frac{m_{020}(m'_{002})^3}{m'_{020}(m_{002})^3\alpha^2}} \end{aligned} \tag{3.9}$$

Finally, we express the parameters for the transformation in terms of 4x4 matrixes, this allows us to concatenate transformation easily.

$$Translation\ Matrix = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rotation\ Matrix = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & 0 \\ \theta_{21} & \theta_{22} & \theta_{21} & 0 \\ \theta_{31} & \theta_{32} & \theta_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Scaling\ Matrix = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

The final transformation matrix is built by multiplying the three previous matrixes, resulting in:

$$T = \begin{bmatrix} \alpha\theta_{11} & \alpha\theta_{12} & \alpha\theta_{13} & T_x \\ \beta\theta_{21} & \beta\theta_{22} & \beta\theta_{21} & T_y \\ \gamma\theta_{31} & \gamma\theta_{32} & \gamma\theta_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

We describe the outline of the moments based registration scheme in Algorithm 1.

Algorithm 1 Moments Based Registration Function

Input:
F - a 3D image, which is the Moving Image
M - a 3D image, which is the Fixed Image
Output:
$T[\mathbf{p}]$ - a function $T[\mathbf{p}] : R^3 \rightarrow R^3$ that represents the function that aligns M to F

1: Compute the Pose Information of F and M: p_F and p_M, see Equation 2.2
2: Compute the translation vector $\bar{T}$, by using Equation 3.2
3: Compute the rotation matrix R, by using Equation 3.4
4: Compute the scale factors S, by using Equation 3.5 (isotropic) or 3.9 (anisotropic)
5: Form the Affine transformation matrix T using Equation 3.11

3.4 Initialization

In this section we describe the procedure to obtain the initial contours which are required to start the segmentation algorithm properly.

Initial contours are essential components in the proposed segmentation method since they indicate the position in which the segmenting algorithm begins. A good positioning of the initial contours is needed due to fact that the proposed Segmentation Method is very sensitive to the initial contours. Initializing the algorithm with a contour distant from the target structure may yield into unsatisfactory results because of the existence of sub-optimal local minima. In order to find the initial contours closest to the target structure, we employed a *multi-level registration scheme* in which we used both, global and local transformations. In addition, we explore the substructure level in order to find best contours for all substructures.

The algorithm that we propose is fully automatic, it requires a training set as explained in 3.2. For the implementation of the registration steps (Affine and BSpline) we used Elastix [13].

3.4.1 Relation between Registration and Initialization.

As mentioned in Section 2.3, registration aims at finding the best transformation that aligns a moving image to a fixed image. By using this transformation we predict the position of the structures to be segmented. This procedure is explained as follows.

Consider P_T a target MRI volumetric image to be segmented. Furthermore, consider a set P_i of training volumes , where $i = 1..N$. Each one of the i volumes has j manual annotated structures, we call S_i^j the binary volume representing the manual annotation of structure j of volume i. The objective of the initialization phase is to obtain the set of binary volumes S_T^j which are the annotations of the j structures of the target volume T.

By registering a patient P_i from the training set P_i (moving image) to the target patient P_T (fixed image), we obtain a transformation T_i. We apply the transformation T_i to the set of manual annotations S_i^j in order to obtain $S_{T_i}^j$, which is the set of j structures of volume T, that are predicted by using manual annotated structures of training volume i..

The problem to solve is to find T_i that predicts the best position of $S_{T_i}^j$. We explain the method that we employed to find T_i in the following sections.

3.4.2 Multi-level Registration Scheme

The motivation of a multi-level registration scheme is that using a single (coarse) step of registration is not sufficient in order to find a proper transformation T_i for initializing the structures of the HF. We validate this statement in Section 3.4.2.1. For this reason, we developed a scheme that consists of 4 steps. At each stage, we apply a new registration operation taking into account the result of the previous step. This means that the final T_i is a concatenation of 4 intermediate transformations.

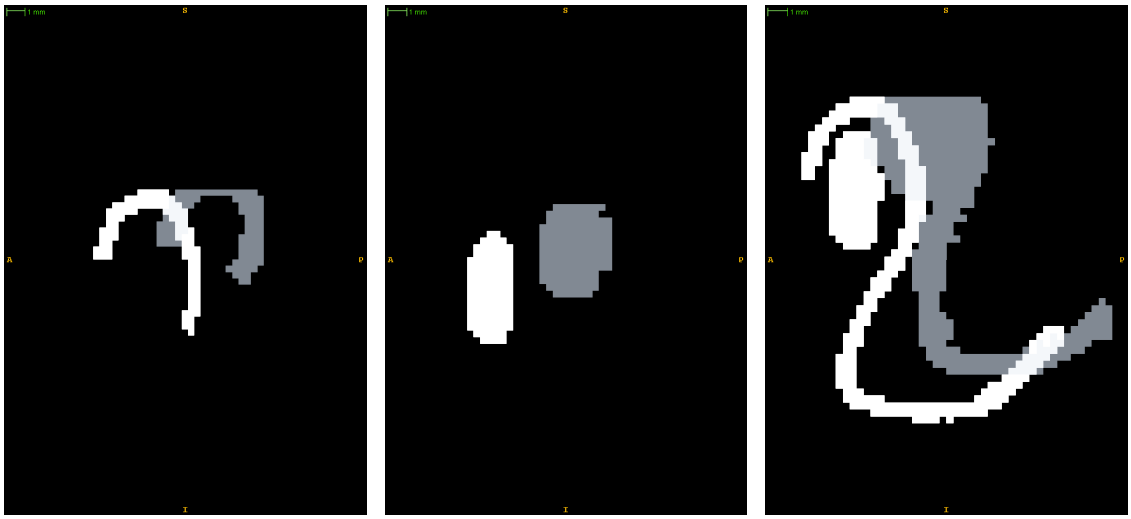
3.4.2.1 Step 1. Affine Registration on Global Data

The fist step of the scheme is used to cope with large differences between volumes. For doing this, we register full brain volumes (only skull was removed from original MR scans) using affine transformation.

Figure 3.4 shows typical results after applying the transformation found in Step 1, to a training contour. From Figure 3.4, it can be seen that the predicted structures are close to the target ones but the overlapping is minimum, as in Figure 3.4a; or in some cases none, as in Figure 3.4b. By looking at Figure 3.4, it is clear that the contours predicted by a global affine registration are not close enough to the target contours. For this reason, we employed a second registration step.

3.4.2.2 Step 2. BSpline Registration on Global Data

In the second step, we used BSpline registration to bring the training contours closer to the target ones. BSpline registration depends heavily on the resolution of the control points mesh, as explained in [16], large spacing between control points allows global deformation, while small spacing allows local deformations. The decision between a coarse or a fine mesh depends on the expected amount of displacement. By looking at the results from Step 1, we observe that the transformed structures are still far (3 to 7 millimeters) from the target, this means that large displacements are still required. Moreover using a finer mesh implies more computation time and since we are working on high resolution images this tasks becomes not feasible. For these



(a) CA structure (b) DG structure (c) 4 structures

Figure 3.4: Initialization results step 1. Coronal slice of Step 1 result. The target structures are shown in white. The transformed structures from one patient of the training set are shown in grey.

reasons, at this step we employ a relative coarse control points mesh (spacing between control points of 8 millimeters) using the full brain images.

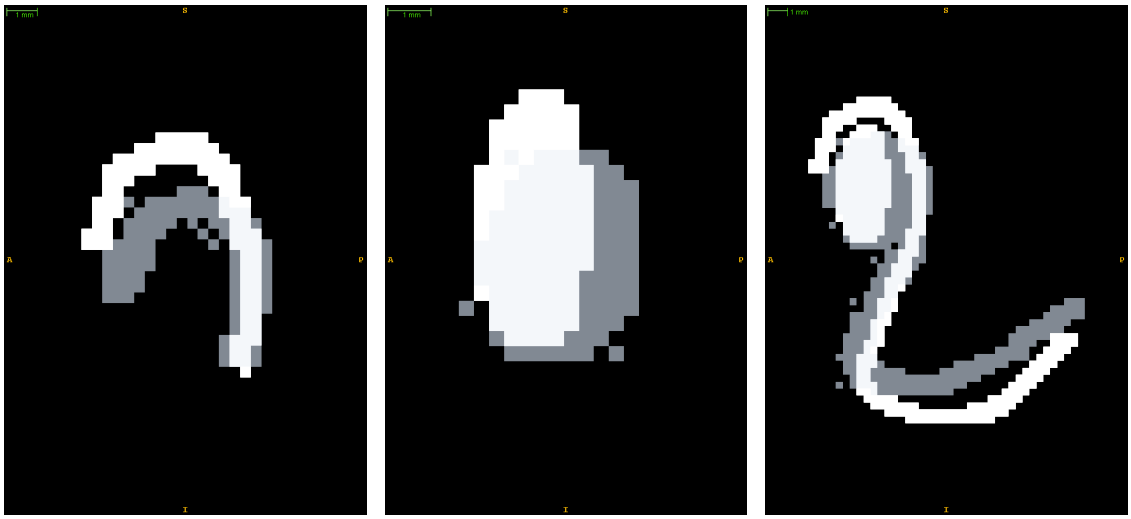
Figure 3.5 shows typical results after applying the transformation found in Step 2. We observe a major improvement when we compare Figure 3.5 with the results after Step 1 in Figure 3.4. In most of the cases, all structures overlap to the target. However, these results are still not entirely satisfactory because the overlapping between the target and the predicted structure is low in terms of DC. For this reason we employ a third registration step in which we use only the volume of interest surrounding the HF.

3.4.2.3 Selection of Volume of Interest

After Step 2, the predicted structures generally overlap the target ones, thus the displacements needed to improve this overlapping are smaller than the ones required in previous steps and they depend heavily on local data i.e. data from the close surrounding of the structures. For this reasons, we selected the surrounding area of the structures of interest and apply registration on this small selected volume of interest data.

In order to select the data of interest we employed the following procedure: From the training set we know the spacial location of the manual annotations, we built a Volume of Interest (VOI) around these manual annotations by locating the set of points P_i that surround the manual annotation. Then we apply the transformation T_i, which was previously obtained in Step 1 and Step 2, to the set of points P_i obtaining the set of points P_i' that surround the new VOI. This procedure is depicted in Figure 3.6.

Once we know the location of the VOI, we extracted the data surrounded by the VOI from both, the Fixed and Moving full volumes. The result are small images that only contain the structure of interest, this means that we can use these images to perform finer registration



(a) CA structure (b) DG structure (c) 4 structures

Figure 3.5: Initialization results step 2. Coronal slice of Step 2 result. The target structures are shown in white. The transformed structures from one patient of the training set are shown in grey.

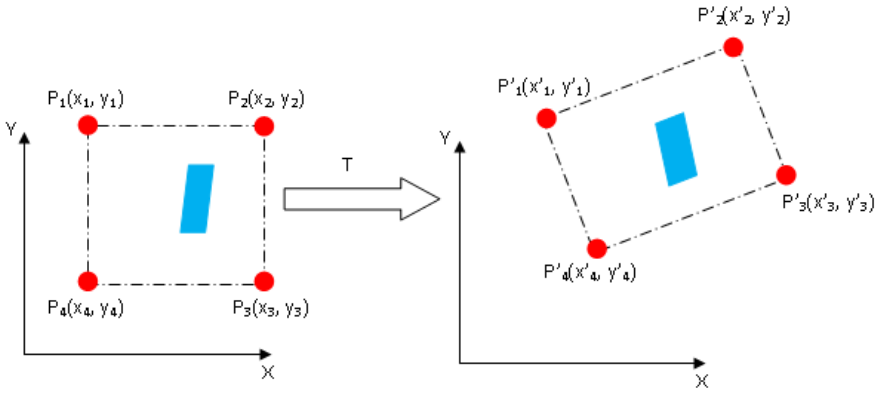


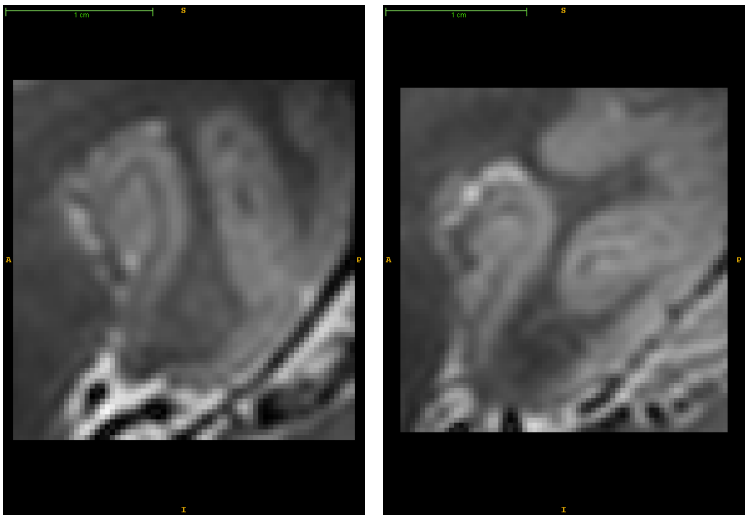
Figure 3.6: VOI selection

without spending much computation time. Figure 3.7 shows typical results of the VOI that we obtained.

3.4.2.4 Step 3. Affine Registration on Local Data

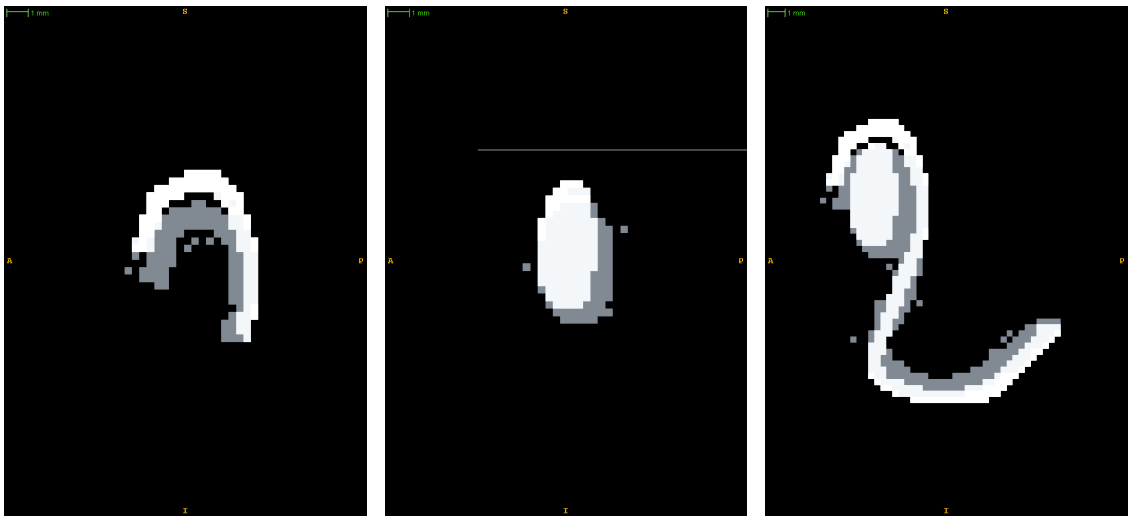
With the images obtained from the procedure of Section 3.4.2.3 we continue the registration scheme. Since now we are entirely focused only on the volume of interest, we can apply finer registration procedures without requiring a large amount of computation time. We apply again affine transformation but this time the transformation T depends only on local data, i.e., data surrounding the structures of interest.

Figure 3.8 shows typical results of the predicted contours obtained after Step 3. By looking at Figure 3.8 we see an improvement compared with the results obtained after Step 2. The



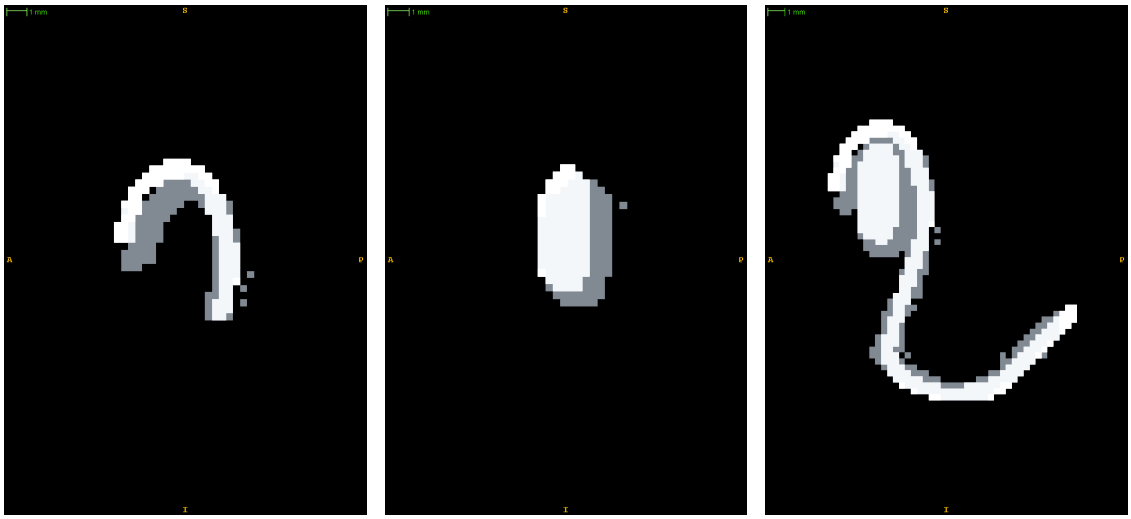
(a) Fix data (b) Moving data

Figure 3.7: VOI for the from fixed (a) and the moving (b) images.



(a) CA structure (b) DG structure (c) 4 structures

Figure 3.8: Initialization results step 3. Coronal slice of Step 3 result. The target structures are shown in white. The transformed structures from one patient of the training set are shown in grey.



(a) CA structure (b) DG structure (c) 4 structures

Figure 3.9: Initialization results step 4. Coronal slice of Step 4 result. The target structures are shown in white. The transformed structures from one patient of the training set are shown in grey.

predicted contours clearly overlap to the target ones. However, small modifications are still required in order to obtain better overlapping. As Figure 3.8 shows, this modifications cannot be solved with rigid transformations, thus we employ BSpline registration in the next step.

3.4.2.5 Step 4. BSpline Registration on Local Data

The final Step in this scheme involves a BSpline registration on local data. By doing this we expect to obtain the closest predicted contours to the target structures. Since we are working on small localized data, we can use high resolution mesh of the control points for the BSpline registration. We used a spacing between control points of 4 mm, with this parameter we expect only small deformations.

Typical results after Step 4 are depicted in Figure 3.9. From Figure 3.9 we observe that the predicted contours are strongly overlapping with the target structures, nevertheless this overlapping is still not complete. The active contours algorithm is proposed to solve the differences between initial contours and the target structures.

3.4.3 Implementation Details of Initialization

We used Elastix [13] to implement each of the registration steps. We present the input files used for Elastix in appendix B. Elastix is an Image Registration Tool, based on ITK. We made use of this tool because it allows us to define precisely the type of registration that we define, this means that we are able to choose the Transformation Type and the Similarity Metric among other parameters. Depending on the step in which the registration is applied, we used different registration types. The main difference is the *Transformation Type*, for steps 1 and 3 we used *Affine Transformation* and for steps 2 and 4 we used *BSpline Transformation*. Another important difference is the resolution of the grid of control point for BSpline registration. At step 2,

we used a coarse grid (spacing = 32 voxels) and at step 4 we used a finer grid (spacing = 8 voxels). After a number of experiments, we chose these resolutions of the grid since they produce best results. In addition, we made experiments on different similarity metrics to optimize registration, we finally decided to use advanced mattes mutual information, see Section 2.3.1, since this metric produced better results; this can be explained due to the fact that mutual information can cope better with differences in intensity values produced by the different acquisition parameters of the MR scans. Once we found the transformation, we apply that transformation to the contours of the training set by using Transformix, which is a tool provided by Elastix.

We describe the outline of the Multi-Level Initialization scheme in Algorithm 2.

Algorithm 2 Multi-Level Initialization scheme

Input:

P_T - a 3D image, which is the test image

P_i - a 3D image, which is the training image of patient i from the training set

S_i^j - a 3D image, which is image that contains the labels of the manual annotations of the j substructures of patient i

Output:

$S_{T_i}^j$ - a 3D image, which is image that contains the predicted contours of the j substructures for P_T obtained from patient i

1: Find T_{i_1} by computing affine registration on complete images, considering P_T as fixed image and P_i as moving image. Apply transformation T_{i_1} to S_i^j, the result is S_{T_1}, see Section 3.4.2.1

2: Find T_{i_2} by computing bspline registration on complete images, considering P_T as fixed image and P_i as moving image. Apply transformation T_{i_2} to S_{T_1}, the result is S_{T_2}, see Section 3.4.2.2

3: Find the VOI surrounding $S_{i_2}^j$, obtain the VOI from P_T and P_i, see Section 3.4.2.3

4: Find T_{i_3} by computing affine registration on VOI images, considering VOI of P_T as fixed image and VOI of P_i as moving image. Apply transformation T_{i_3} to S_{T_2}, the result is S_{T_3}, see Section 3.4.2.4

5: Find T_{i_4} by computing bspline registration on VOI images, considering VOI of P_T as fixed image and VOI of P_i as moving image. Apply transformation T_{i_4} to S_{T_3}, the result is $S_{T_i}^j$, see Section 3.4.2.5

3.5 Segmentation Algorithm

The final part of the method proposed is the segmentation algorithm. The goal of this stage is to improve the segmentation given by the initialization. The segmentation algorithm that we use is based on active contours [17], following a moments-based prior as described in [6]. This means that the segmentation process involves the minimization of an energy cost functional. In order to minimize the energy cost we employed an iterative algorithm. Similar to [2] and [6], we define the energy cost functional in a maximum a posteriori estimation framework as

$$E(C) = -\log P(C, data) \tag{3.12}$$

where C is a set of evolving contours which represent the image segmentation of the structures of interest and *data* is the intensity image in gray scale. The main difference between [2] and

[6] is that our energy functional combines the data term and the prior in one single term. The main idea of the method is to evolve the contours C in order to minimize the cost function which is equivalent to maximize a probability density function.

We define the joint kernel density estimate of m shapes as:

$$P(C, data) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{M} k_i^j \left(d\left(\phi^j, \phi^{ji}\right), \sigma_j \right) \tag{3.13}$$

where N is the number of training patients, M is the number of substructures and $k(\cdot, \sigma_j)$ is a Gaussian kernel with standard deviation σ. We use Signed Distance Function SDF (ϕ) to represent the active contour at each iteration. In Equation 3.13 ϕ^j is the SDF of the jth object and ϕ_i^j is the SDF of the jth object from the ith member of the training set.

We make use of a distance function $d(\cdot, \cdot)$ which is defined as:

$$d\left(\phi^j, \phi^{ji}\right) = \sqrt{\sum_{r+s+t \leq L} w_{r,s,t} \left(m'_{r,s,t}(\phi^j) - m'_{r,s,t}(\phi^{ji}) \right)^2} \tag{3.14}$$

where $m'_{r,s,t}(\phi^j)$ is the value of moments of structure j taken into account the intensity value in gray scale, this is:

$$m'_{r,s,t} = \int_{\Omega} x^r y^s z^t g(x, y, z) \left(H\left(-\phi\left(x, y, z\right)\right)\right) dxdydz \tag{3.15}$$

where $g(x, y, z)$ is the intensity value in gray scale and $H(\cdot)$ is the Heaviside function

We implement the distance function in Equation 3.14 by using the relative pose information, see 2.2.2. This means that we consider $d\left(\phi^j, \phi^{ji}\right) = d\left(p^j, p^{ji}\right)$, where p^j is the relative pose of substructure jth and p^{ji} is the relative pose of the jth structure from the ith member of the training set. In order to compute the relative pose for each substructure, we register that substructure to the ensemble of all substructures of interest and then compute moments following Equation 3.15. At this stage, we use the fast registration method proposed in Section 3.3.

We express the Gaussian kernel function in Equation 3.13 as:

$$k_i^j = \frac{1}{\sqrt{2\pi {\sigma_j}^2}} \exp\left\{ -\frac{1}{2{\sigma_j}^2} \left[\sum_{r+s+t \leq L} w_{r,s,t}^j \left(m_{r,s,t}^{'j} - m_{r,s,t}^{'ji} \right)^2 \right] \right\} \tag{3.16}$$

where $w_{r,s,t}^j$ are weights. These weights sum up to one and are defined in such a way that the structures from the training set that are more similar to the test image have more relevance in the summation.

3.5.1 Gradient Flow

In order to minimize Equation (3.12), we define a Gradient Flow for the joint kernel in Equation (3.13). Following [6], we use the fact that $\log P(C, data) = \log \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{M} k_i^j$, then

$$\frac{\partial}{\partial t}\log P\left(C, data\right) = \frac{1}{N}\frac{\sum_{i=1}^{N}\left\{\sum_{j=1}^{m}\frac{\partial}{\partial t}k_i^j\prod_{l=1,l\neq j}^{m}k_i^l\right\}}{P\left(C, data\right)} \tag{3.17}$$

Using Equation (3.16) and the definition of moments given in Equation (3.15), we obtain:

$$\frac{\partial}{\partial t}k_i^j = \frac{k_i^j}{{\sigma_j}^2}\int_{R^3}\sum_{m'^j_{r,s,t}\in\mathcal{M}_L} w_{r,s,t}\; g(x,y,z)\; x^r y^s z^t\left[\left(m'^j_{r,s,t} - m'^{ji}_{r,s,t}\right)\delta_\epsilon\left(\phi_{C^j}\right)\right]\phi'_{C^j}dxdydz. \tag{3.18}$$

Inserting Equation (3.18) into Equation (3.17):

$$\frac{\partial\phi_{\bar{C}^j}}{\partial t} = \frac{\sum_{i=1}^{N}\prod_{l=1}^{m}k_i^l MPF\left(j,i\right)\delta_\epsilon\left(\phi_{\widetilde{\mathbf{C}}^j}\right)}{{\sigma_j}^2 P\left(\widetilde{\mathbf{C}}\right)\cdot N}\left|\nabla\phi_{\widetilde{\mathbf{C}}^j}\right|, \tag{3.19}$$

where $MPF(j,i) = \sum_{r+s+t\leq L} w^j_{r,s,t}x^r y^s z^t g(x,y,z)\left(m'^j_{r,s,t} - m'^{ji}_{r,s,t}\right)$ for each $j\in\{1,\ldots,m\}$.

3.5.2 Update Step

At each iteration the gradient given in Equation (3.19) is computed. As a result an update term for each substructure is obtained with:

$$\phi_{\bar{C}} = \frac{\partial\phi_{\bar{C}^j}}{\partial t}\cdot\Delta_t \tag{3.20}$$

$\phi_{\bar{C}^j}$ is added to the result of the previous iteration yielding in:

$$\phi_{i+1} = \phi_i + \phi_{\bar{C}} \tag{3.21}$$

We describe the segmentation algorithm in Algorithm 3.

Algorithm 3 The segmentation algorithm.

Input:
$t := 0$; $\mathbf{C}_{t=0}$;Initial contours give by Algorithm 2
$\mathbf{C}_i^j$;training set of manual annotated contours, where $i=\{1,\ldots,N\}$ and $j=\{1,\ldots,M\}$,N is the number of patients and M is the number of substructures of interest.

Δ_t; $\bar{\varepsilon}$ – a threshold for steady state
Output:
$\mathbf{C}_t$ a set of segmenting contours

1: **repeat**
2: **for all** j={1,...,m}, (i.e. for each structure of interest) **do**
3: Compute shape and relative pose according to Equation (3.19) :

$$\frac{\partial\phi_{\bar{C}^j}}{\partial t} = \frac{\sum_{i=1}^{N}\prod_{l=1}^{m}k_i^l MPF\left(j,i\right)\delta_\epsilon\left(\phi_{\widetilde{\mathbf{C}}^j}\right)}{{\sigma_j}^2 P\left(\widetilde{\mathbf{C}}\right)\cdot N}\left|\nabla\phi_{\widetilde{\mathbf{C}}^j}\right|$$

4: Update the signed distance functions of each structure

$$\phi_{i+1} = \phi_i + \frac{\partial\phi_{\bar{C}^j}}{\partial t}\cdot\Delta_t$$

5: **end for**
6: **until** $\left(\text{steady state is achieved, i.e. } d\left(\phi_t,\phi_{t+1}\right) < \bar{\varepsilon}, \text{for all } j\in\{1,\ldots,m\}\right)$

CHAPTER 4

Results

In this chapter we present the results obtained for each of the methods proposed in Chapter 3. We start by validating the registration based on moments from Section 3.3. Then, we present the results obtained for the initialization scheme of Section 3.4. Finally, we show the results of the segmentation algorithm of Section 3.5. At each step, we first describe the experiment used to validate the results.

All the experiments were performed using a 3.33 GHz Intel Xeon Processor (64-bits). We employed Python scripts for running the algorithms through the patients in the training set.

4.1 Results of Registration Based on Moments

In this section, we present the results for the validation of our registration method which is based on *moments*. For this purpose we make use of two data sets namely the Internet Brain Segmentation Repository IBSR [18] and 7T data provided by Leiden University (ADAB).

IBSR is a open data set composed by 18 subjects; each subject contains normalized T1-weighted volumetric images and segmentation ground truths for a number of brain structures. The set is composed by 14 men and 4 women, with ages from 8 to 71 years. The resolution of the images is 1.5 mm. ADAB is a set of T1-weighted volumetric images, it comprises 18 scans, 7 women and 11 men. The resolution of the images is 0.35mm. As a preprocessing step we removed non-brain tissue using BET.

4.1.1 Experiment Description

For both data sets the following process was done. One brain was considered as the fixed or reference one, then all the others were registered to it. DC was used as metric to evaluate the performance of the method proposed, see Section 2.4. Scaling was made isotropic and anisotropic, see Section 3.3.1. Better DC values are expected from anisotropic scaling because anisotropic scaling considers the difference of scaling among axes. In addition the method that we proposed is compared to a state-of-art tool called Linear Image Registration Tool FLIRT [19].

Moreover, as an optimization, we compute the parameters of the transformation by calculating moments using different resolution on the images, this means that we took different number of sample voxels from the images in order to compute moments when using equation 2.10. By doing this procedure, the computation time was reduced, due to the fact that less operations where required to calculate moments since less voxels were considered.

4.1.2 IBSR Results

Figure 4.1 shows the result of global moments-based registration applied to IBSR data set. We obtained the results of isotropic and anisotropic methods by discretizing the image to a spacing of 1mm, this means sample voxel was taken every millimeter. Each column in Figure 4.1, represents the average DC obtained from the registration to that specific patient, applying the different methods previously described (isotropic, anisotropic and FLIRT).

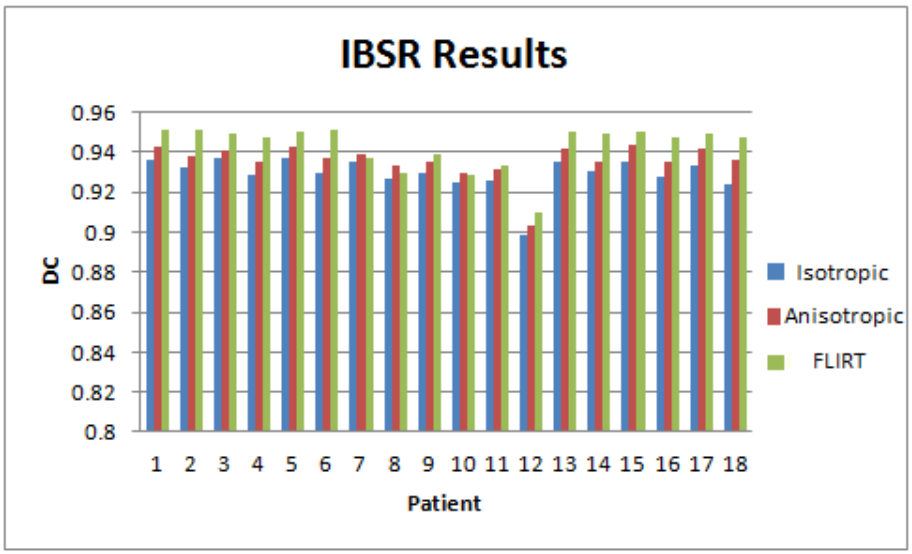


Figure 4.1: Registration results DC from IBSR data set

From Figure 4.1 it can be observed that, according to the metric that we employed, the anisotropic scheme produces slightly better results than the isotropic scheme. The anisotropic DC average is 0.93552 and the isotropic DC average is 0.92911. This means an improvement of 0.68%. From Figure 4.1, it can also be observed that FLIRT method yields in slightly better results than the method that we proposed in both scaling schemes (isotropic and anisotropic). The average DC obtained with FLIRT in all patients is 0.94284, this is 0.78% better than DC obtained with anisotropic scheme.

Figure 4.2 shows the average computation time that the registration procedure took for each patient.

From Figure 4.2 it can be observed that the computation time that FLIRT spends is much higher than the one of the method proposed. The average time used by FLIRT is 118.50 seconds, while our proposed method spends only 31.03 seconds with isotropic scheme and 31.09 with anisotropic scheme. Thus the execution time that our method needs is only 26.15% of the time that FLIRT requires. Moreover our method can be even faster when a fewer number

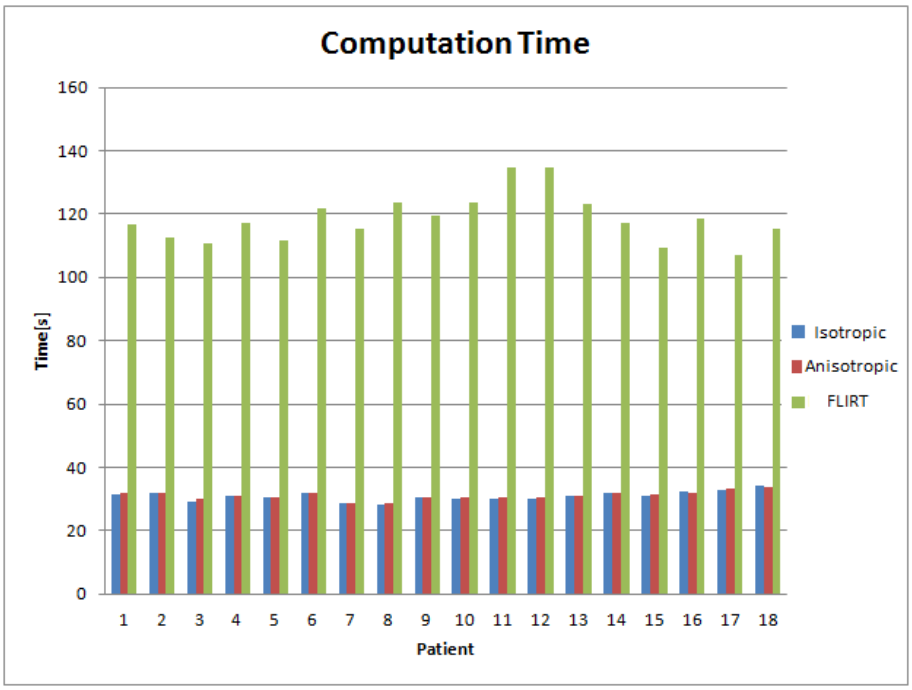


Figure 4.2: Registration results Computation Time from IBSR data set

of samples voxels are considered in moments computation, this results are shown in Section 4.1.4.

4.1.3 ADAB Results

Figure 4.3 depicts the results obtained for ADAB data set. Similar to Figure 4.1, in Figure 4.3 each column represents the average DC obtained from the registration to that specific patient, applying the different methods previously described (isotropic, anisotropic and FLIRT).

Similar to IBSR results, in ADAB data set a small improvement can be seen from isotropic to anisotropic scaling. The average DC of isotropic scaling is 0.87844 and for anisotropic is 0.88286. Our method does not perform as good as FLIRT, the average DC of FLIRT is 0.90089, this represent a difference of 2.04% in respect to anisotropic scaling. Moreover there is a general decrease in performance of all methods when compared to IBSR data set. A possible reason is that IBSR data set contains standardized test images while ADAB images are scans without any preprocessing and with different scan parameters.

Figure 4.4 shows the average computation time that the registration procedure took for each patient in ADAB data set.

From Figure 4.4, it is clear that the computation time required by our method is much smaller than the one that FLIRT consumes. The average time that FLIRT utilized for this data set is 220 seconds and out method spent only 24.35 in isotropic scheme and 25.05 seconds in the anisotropic one. This means that out method proposed is 8.82 times faster.

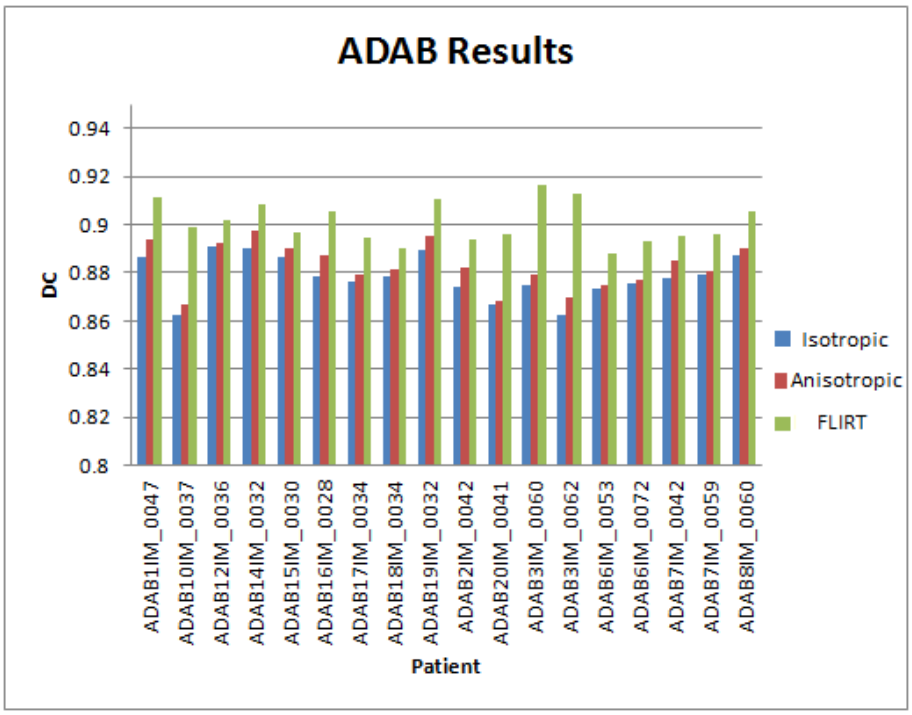


Figure 4.3: Registration results DC from ADAB data set

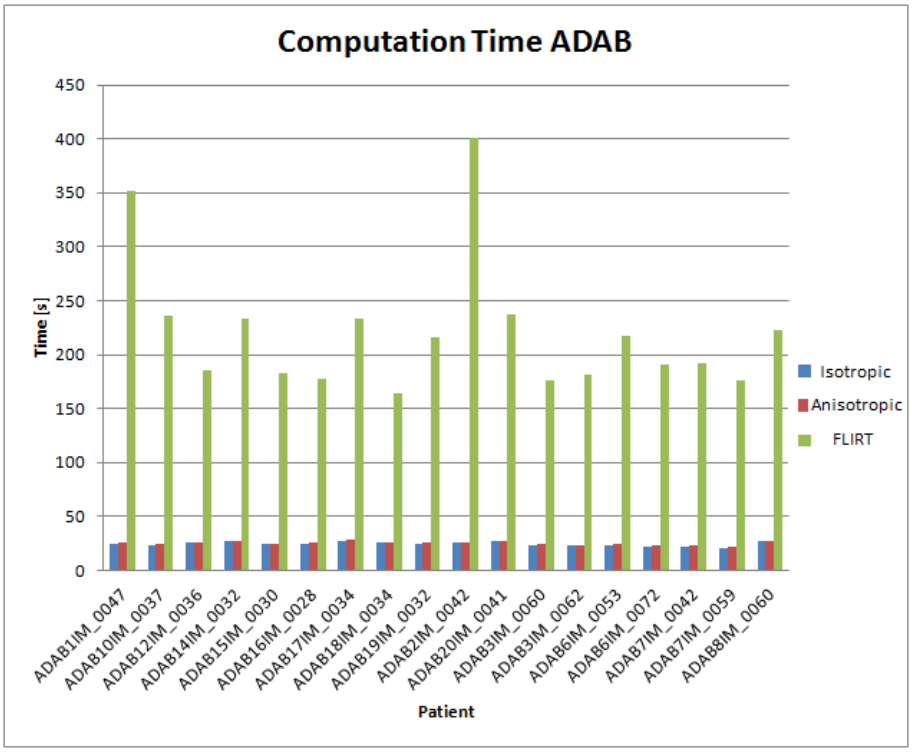


Figure 4.4: Registration results Computation Time from ADAB data set

4.1.4 Optimizations

In this section, we show the results obtained when a fewer number of sample voxels were used in order to compute moments. The idea behind this optimization is that the computation time decreases when a fewer number of voxels are used to calculate moments, however the precision of the computation also decreases. This reduction is obtained by increasing the spacing in which the sample voxels are taken, see Section 2.2.3.

Figure 4.5 shows the results of DC and computation time that we obtained by using different spacing sizes for computing moments.

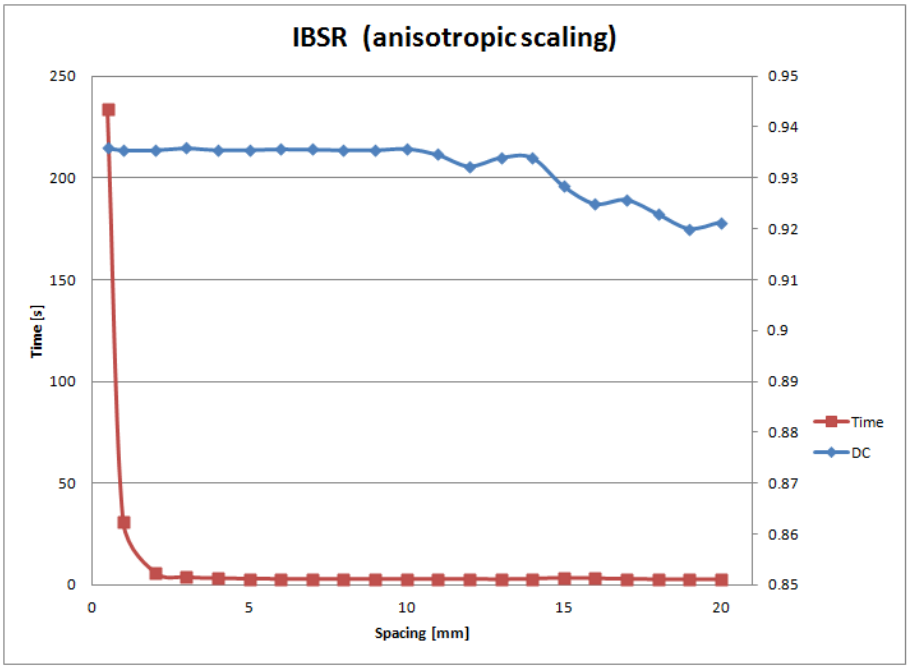


Figure 4.5: Optimizations on IBSR data set

In Figure 4.5 it can be observed that decreasing the samples voxels used for moments calculation decreases drastically the computation time. Using a spacing of 0.5 mm requires in average 233 seconds, using 1 mm as spacing takes 31 seconds in average and using a spacing of 3 mm takes only 3.66 seconds. This reduction in precision on moments computation does not decrease drastically the final result of registration. In the three previously mentioned cases the DC scored is in average 0.935. Even with a very coarse grid with a spacing of 20 mm, the DC does not decreases too much. The result obtained with this spacing is a DC of 0.9211 which is 1.57% less than the score obtained with 0.5 mm of spacing.

Moments computation is only a part of the complete registration method. This means that the optimizations done on moments computations do not cover the entire computation time that the registration method spends. For this reason, there is a limit in which increasing the spacing between sample voxels does not decrease computation time any more. This can be seen in Figure 4.5 where after spacing of 4 mm a steady state of the computation time is reached.

Figure 4.6 shows the results of DC and computation time that we obtained by using different spacing sizes in ADAB data set. Similar to the IBSR data set, in ADAB the reduction of precision in moments calculation does not modify severely the DC results keeping a value higher than 0.88 in spacing from 0.5 mm to 5 mm. However, the time consumed by moments

computation changes significantly when modifying the spacing of the sample voxels, from 96 seconds using a spacing of 0.5 mm to 5.7 seconds when using a spacing of 4.

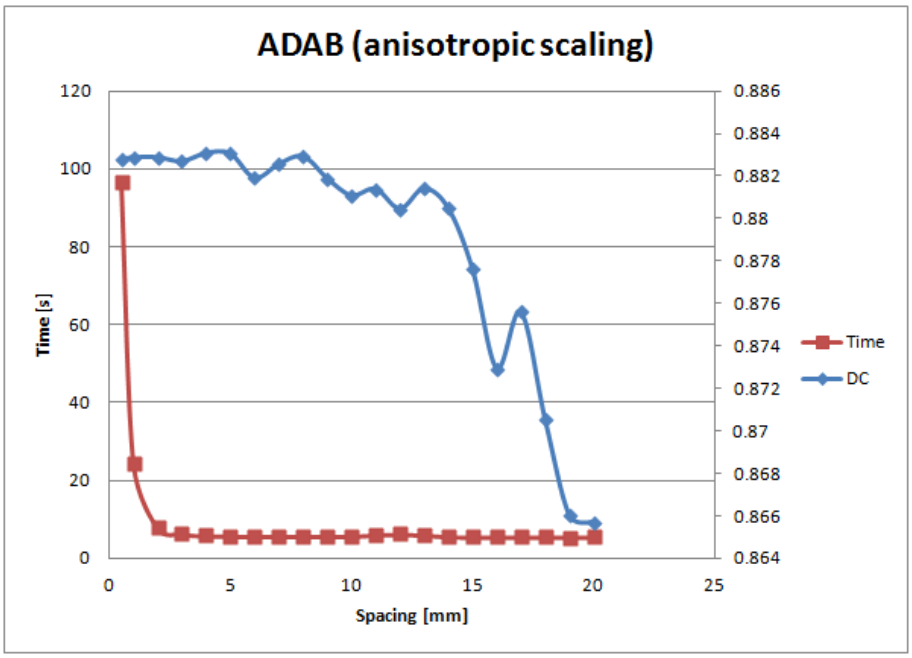


Figure 4.6: Optimizations on ADAB data set

In both cases it can be observed that sometimes a decrease in precision of moments does not decrease the DC values. This is explained by the fact that moments computation is used to obtain the transformation matrix but after that, interpolation is required to apply the transformation previously found, this means that there is still room to errors due to interpolation which modify the DC result.

4.2 Results of Initialization

In this section we describe the results to validate our initialization scheme. First we describe the input data that we used. Second, we describe the experiment. Then, we present the results in terms of DC. Finally we show performance indicators.

4.2.1 Input Data Description

In order to detect the substructures of the HF we require images with very high resolution, for this reason we employ images obtained with MR scans with strength of 7 Tesla. The resolution of the images that we use is commonly 720 x 720 x 543 pixels, with a spacing in millimeters of 0.35 x 0.35 x 0.35. Due to this very high resolution of images, the normal memory size of each image is in the range of 500 MB to 1GB.

4.2.2 Experiments Description

In order to validate the scheme proposed, we used leave-one-out cross validation. Starting from a training set of 3 patients, see 3.2, with manual annotation of 4 structures of the HF, namely DG, CA, Sub and EC, from both left and right side, we applied the Multi-level Registration Scheme proposed, taking one patient as Test image and the other two as Training set. We predict the contours for the Test image by using the manual annotations of the Training set. Then, we obtained average DC of the overlapping between the predicted substructures and the target ones.

4.2.3 Results

We present the average DC obtained for each structure at each step of the method, for both sides left and right.

Figure 4.7 shows the results obtained for the left side HF.

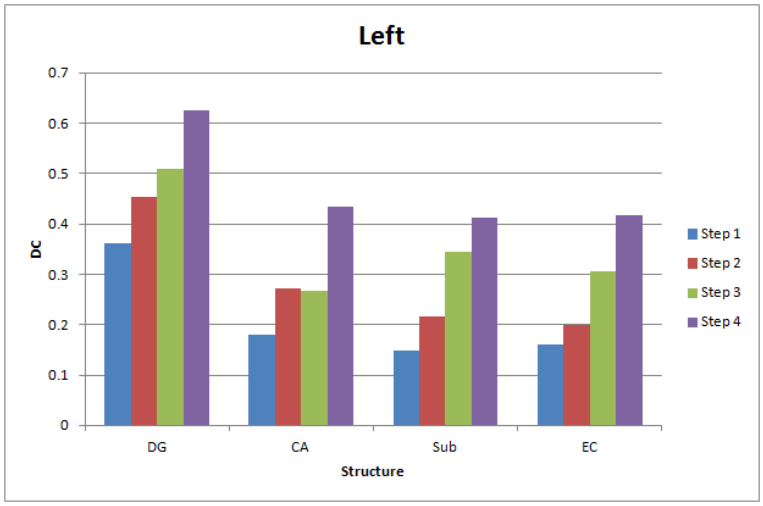


Figure 4.7: Initialization Results Left Side

From Figure 4.7, we can observed how the overlapping metrics increases at each step of initialization. It is clear that the registration scheme proposed increases the overlapping between the predicted structure and the ground truth.

Figure 4.8 shows the results obtained for the right side. By looking at Figure 4.8 we observe a decrease of DC in the DG substructure. The explanation for this situation is that while doing registration, we consider volume of interest that includes all the substructures. This means that the transformation obtained is focused on finding the best alignment for the entire volume, as a consequence for some substructures, in this case DG, the overlapping decreases but the overall average increases.

The complete results (average of left and right) are shown in Figure 4.9. Figure 4.9 depicts the evolution of overlapping between the predicted and target structures over the steps of the procedure proposed. It is clear that the overlapping increases accordingly to the steps taken. The largest improvement occurs between Step 1 and Step 2, where BSpline registration is employed for the first time. For example, after Step 1, the overlapping of DG structure is only 0.27 DC and after Step 2 the overlapping is 0.61 DC.

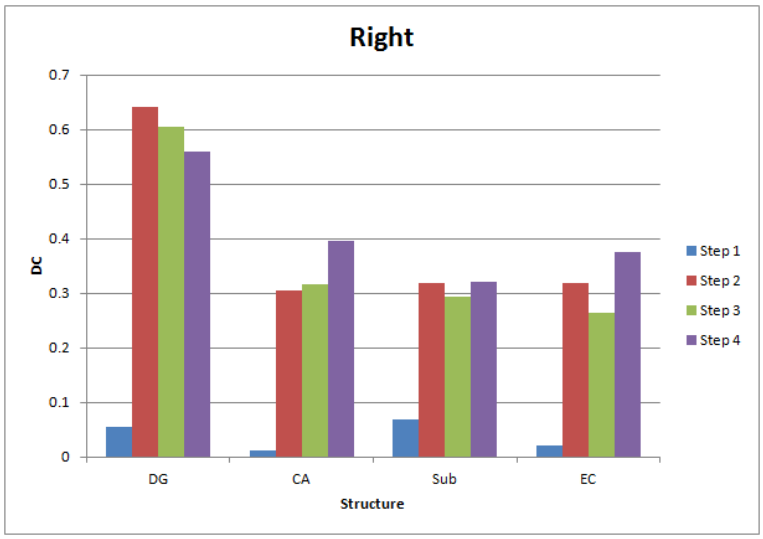


Figure 4.8: Initialization Results Right Side

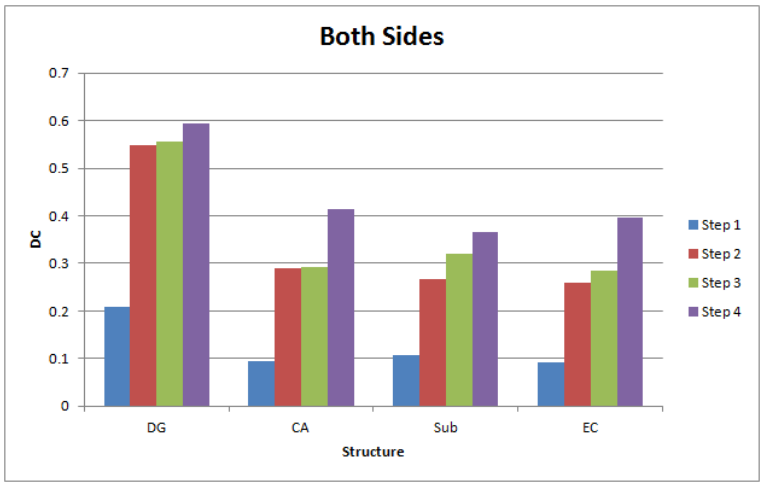


Figure 4.9: Initialization results both sides

Figure 4.10 shows the evolution of the average DC for all patients for substructure DG as well as the standard deviation. From Figure 4.10 we can see that there is an increase of DC along the steps and also, the standard deviation becomes smaller, this means that after step 4, the initial result is good no matter which training patient is used to predict it.

A qualitative analysis can be done by looking at Figure 4.11. From Figure 4.11 it can be seen that there exists overlapping between the substructures, however, the overlapping is not the same accurate for all of them. It is clear that DG and CA have better quality of overlapping when compared with Sub and EC. We consider that the reason of this discrepancy is that there exists high anatomical differences, among the patients in the training set, in substructures Sub and EC. For this reason, it is more difficult to find a correct transformation to predict the contours of those substructures.

Table 4.1: Initialization Results Left

	Left Side			
Structure	Step 1	Step 2	Step 3	Step 4
DG	0.362	0.453	0.509	0.626
CA	0.179	0.272	0.266	0.434
Sub	0.147	0.216	0.345	0.412
EC	0.162	0.199	0.306	0.417

Table 4.2: Initialization Results Right

	Right Side			
Structure	Step 1	Step 2	Step 3	Step 4
DG	0.055	0.642	0.605	0.561
CA	0.012	0.305	0.316	0.396
Sub	0.069	0.319	0.293	0.320
EC	0.022	0.319	0.265	0.376

Table 4.3: Initialization Results Both Sides

	Both Sides			
Structure	Step 1	Step 2	Step 3	Step 4
DG	0.209	0.547	0.557	0.593
CA	0.096	0.289	0.291	0.415
Sub	0.108	0.268	0.319	0.366
EC	0.092	0.259	0.285	0.397

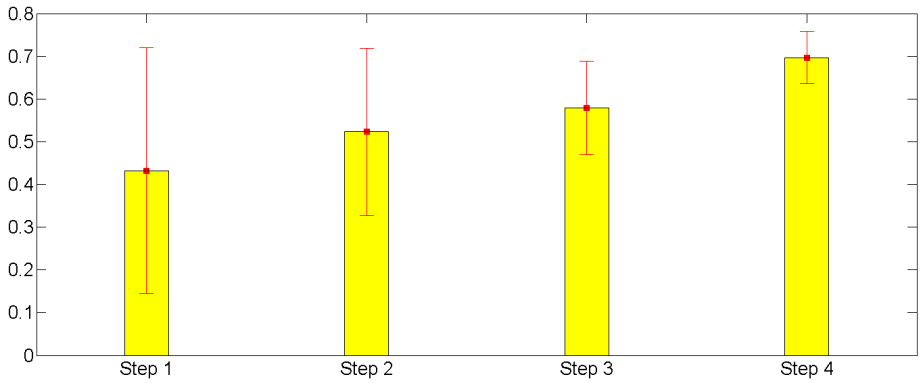
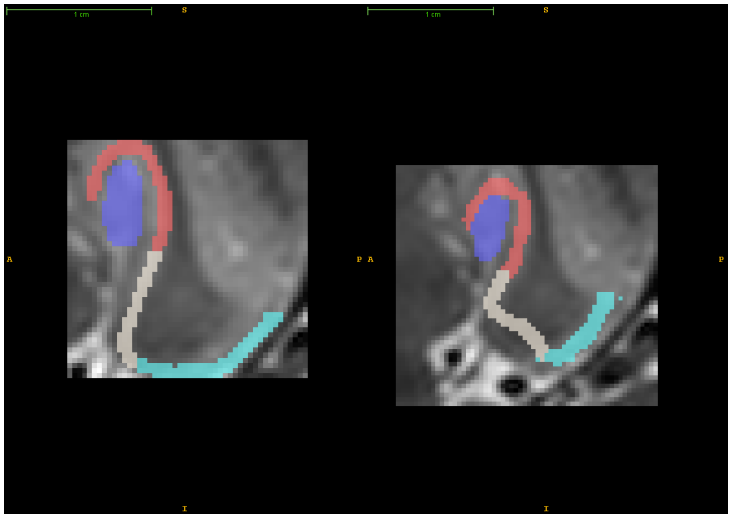


Figure 4.10: Average DC with standard deviation of DG substructure (Initialization Stage)



(a) Ground Truth (b) Predicted Contour

Figure 4.11: Qualitative Analysis Initialization

4.2.4 Performance Metrics

The initialization took 50 minutes per subject. The experiment was done using a 6 cores Intel Xeon Processor with 6GB RAM.

4.3 Results of Segmentation

By using the initials contours obtained from 3.4, we apply the algorithm described in 3.5. We present results for two sets, healthy patients and patients with Alzheimer Disease. For validating our tool on healthy patients, we make use of the training set and validate the method by means of cross validation and DC. We also tested our tool in patients with Alzheimer Disease, however, we do not have ground truths for those patients, for this reason we employ visual validation to evaluate the results in them.

In both cases, we made use of different modalities of prior, taking into account moments up to order 2. We present the best results obtained for each of the sides and substructures.

4.3.1 Evaluation on Healthy Patients

In this experiment we made cross validation from the training set, see 3.2. Figure 4.12 shows the DC with standard deviation for the segmentation results of Left HF.

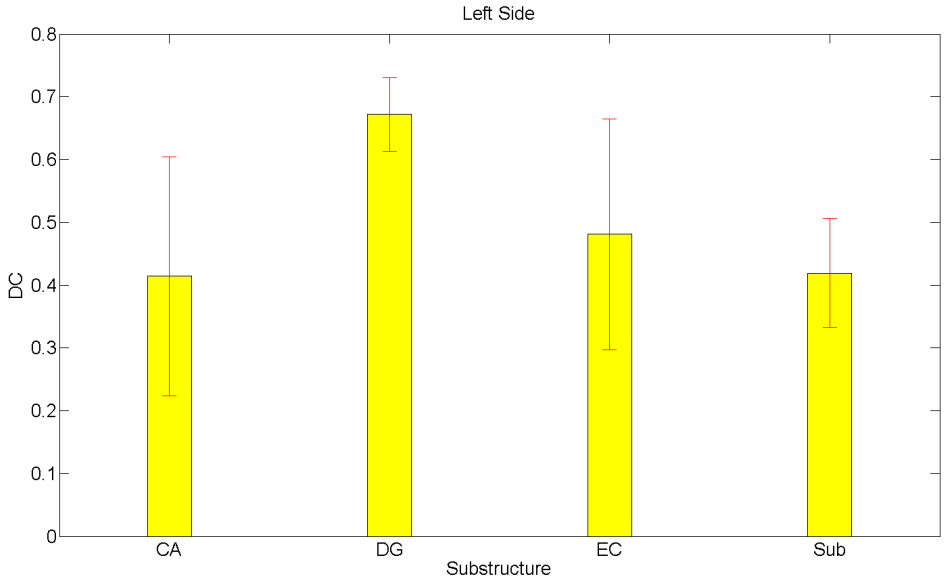


Figure 4.12: Left Side Segmentation Results

Figure 4.13 shows the DC with standard deviation for the segmentation results of Right HF.

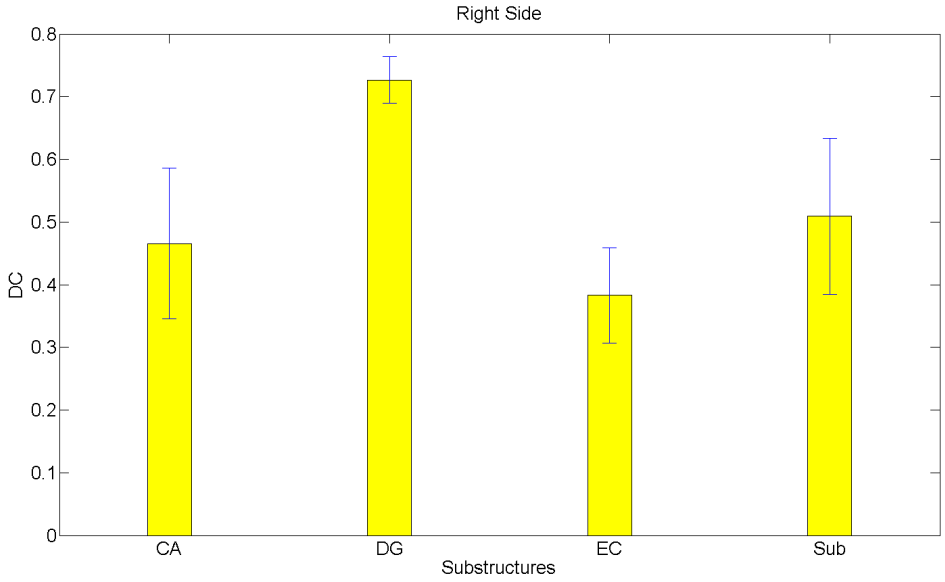
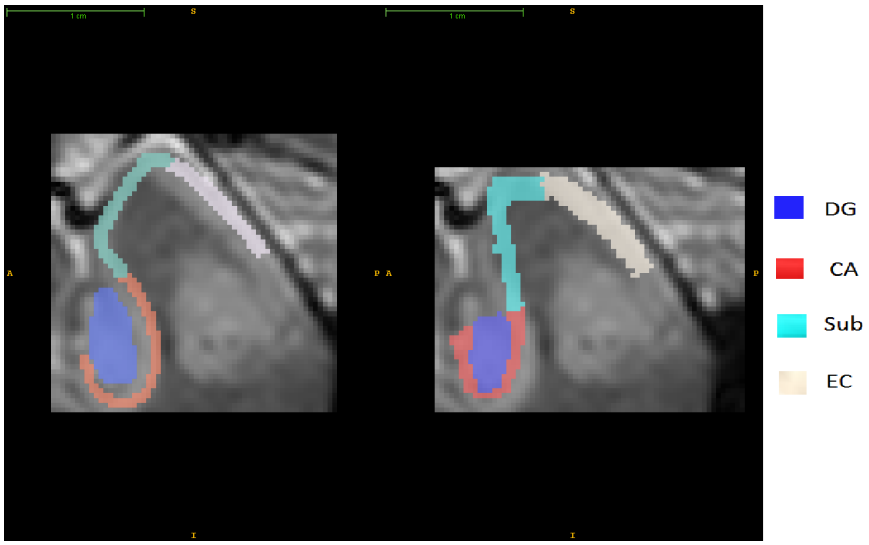


Figure 4.13: Right Side Segmentation Results

By looking at Figure 4.12 and 4.13, it becomes clear that the best results from our method



(a) Target Ground Truth (b) Segmentation Result

Figure 4.14: Qualitative Analysis Segmentation Result

are the ones for substructure DG in both sides. In the other structure the variation is bigger. For example, in the Left side EC results are better than CA but on the Right side CA produces best results than EC. One of the factors that impacts the results is the precision of the manual annotation. As mentioned in [8], there exist no general consensus in structures segmentation, for different experts, different voxels may be considered inside or outside from a specific structure.

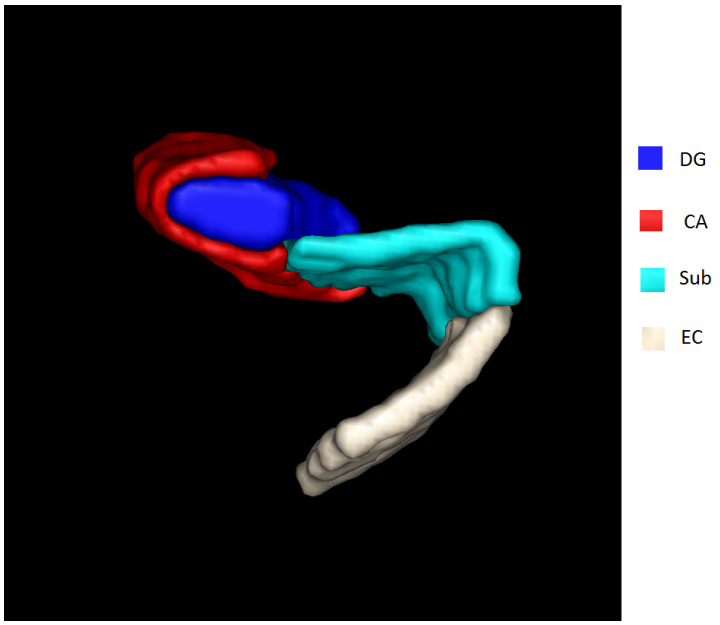


Figure 4.15: Segmentation Result 3D Visualization

Despite the variations in the results among substructures and sides, it is clear that the results after the Segmentation Algorithm are better than the ones produced by the Initialization Algorithm. By comparing Table 4.3 and Table 4.4, we can see that DC increases constantly by

using the Segmentation Algorithm.

Moreover, we consider that the quality of the results obtained are affected by the reduced Training Set that we employed. We used only 3 patients, however, having more patients in the Training Set increases the probability of finding a patient similar to the test one.

Table 4.4: Add caption

Structure	Left	Right	Both
DG	0.672	0.727	0.699
CA	0.414	0.466	0.440
Sub	0.419	0.509	0.464
EC	0.481	0.383	0.432

4.3.2 Evaluation on Patients with Alzheimer Disease

In this section we present the results obtained when using images of patients with Alzheimer Disease.

Figure 4.16 shows the result on Right HF in a patient that presents atrophy. We can observe that the results are still not optimal. The label of the substructures are not correctly located over the real ones. Another important fact is that the size of the labels are almost the same as the original from the training set, the shrinking caused by AD is not entirely coped with the method proposed. The reason behind the not optimal results is that the brain that we tested, present large anatomical differences with the training set, probably because of atrophy caused by AD. This makes extremely difficult to find a correct transformation from training to the test patient.

4.3.3 Performance Metrics

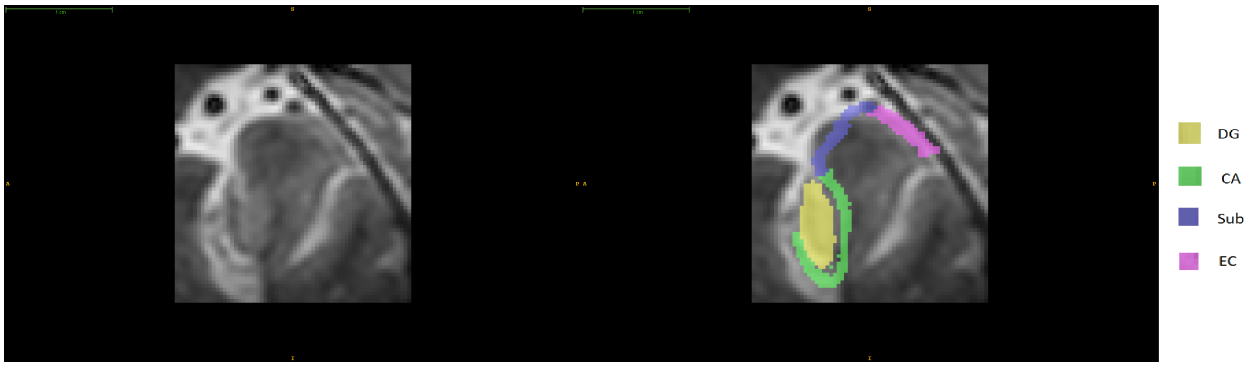
The iterative process of the Segmentation Algorithm took in average 5 minutes. The experiment was done using a 6 cores Intel Xeon Processor with 6GB RAM.

4.3.4 Comparison with Existing Method

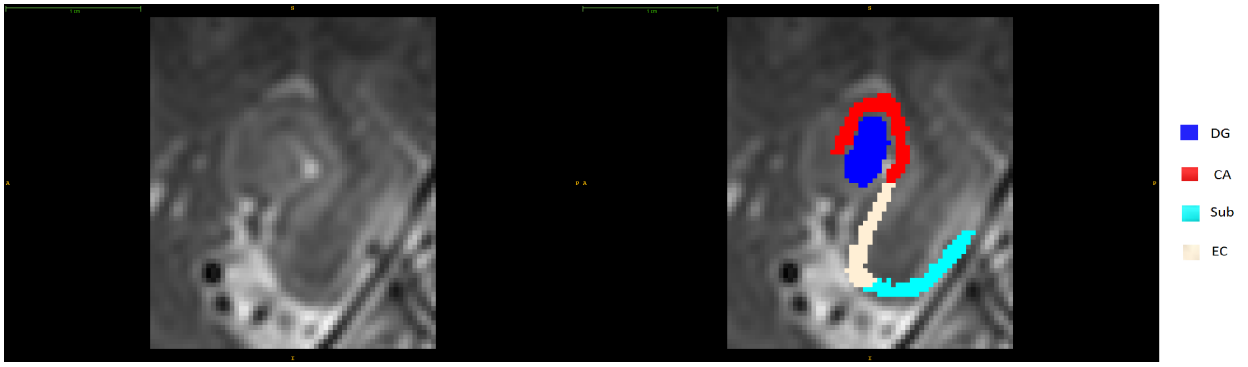
Figure 4.17 shows the comparison in terms of DC with the fully automatic method for segmentation described in [1].

By looking at Figure 4.17, we can observe that in two substructures (CA and Sub), the method described in [1] achieves better DC than our method. However, for substructure DG our method performs slightly better. In [1], the authors employed a different anatomical definition of HF, for this reason no comparison is possible for EC substructure.

When we analyze the computation time that each of the method requires, we can see a big difference between [1] and our method. While in [1], the procedure takes 5 hours for one patient, in our method the procedure takes, in average, only 55 minutes (50 in the Initialization and 5 in the Segmentation). This means that our method is 5.4 times faster that the one proposed in [1].



(a) Only Data (b) Data + Segmentation



(c) Only Data (d) Data + Segmentation

Figure 4.16: Segmentation Results on Patients with AD, (a) and (b) show Right HF and (c) and (d) show Left HF

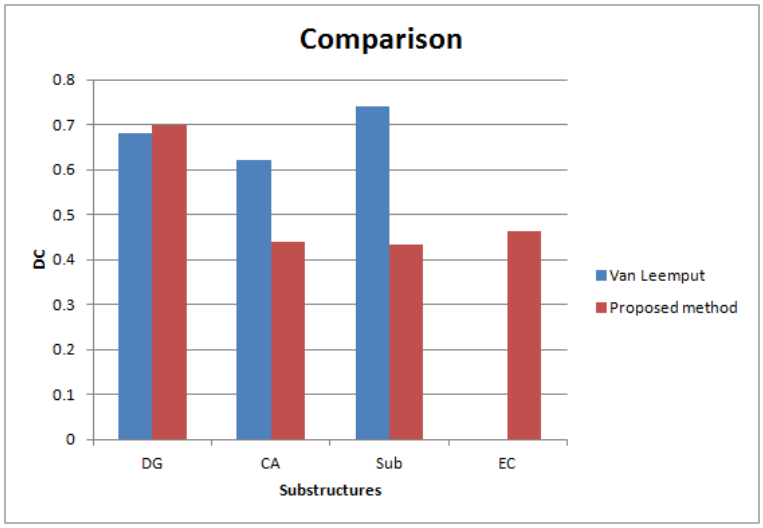


Figure 4.17: Comparison with existing method [1]

CHAPTER 5

Conclusions and Future Work

We developed and implemented a Fully Automatic Segmentation Tool for Brain Substructures focusing on the HF. In order to solve this problem, we made three relevant contributions:

- First, we proposed a fast registration method based on moments. We validate our results by means of DC and compared our results to a state-of-the-art tool called FLIRT. By using two different data sets, namely IBSR and ADAB, we showed that our proposed method achieves similar results to the ones from FLIRT but spending much less computation time.

- Second, we introduced an Initialization scheme which is based on a Multi-Level Registration approach. At this stage, we made use of different types of transformations and different resolutions. We implemented this framework using Elastix. Our proposed method predicts the contours of the substructures of interest from a set of training contours, the overlapping coefficient shows that this initial prediction is good enough to start the segmentation algorithm.

- Third, we proposed a Segmentation Algorithm based on relative Pose Information, similar to [6] and [2]. The relevant contribution that we made is that the intensity term is merged with the prior based on Moments into a single term. We obtained results using leave-one-out cross validation experiments. We compared our results to the fully automatic method proposed in [1]. We showed that for a specific substructure from the HF out tool performs better in terms of DC. Although our results in the rest of the substructures are not as good as the ones from [1], our tool spends much less computation time, being 5 times faster than the one proposed in [1]. In addition, we validate our tool using volumetric images of patients with Alzheimer Disease that present atrophy in the HF. We showed that the results produced by our tool come close to the real substructures even when some level of atrophy is present. However, we cannot conclude that out tool is robust enough to cope with high atrophy levels.

- For the validating our segmentation tool, we employed 7T images T2-weighted. For the anatomical description of the HF, we followed the protocol described in [9].

5.1 Future work

One of the aspects that can be improved is the Registration base on Moments. In the method proposed, we employed moments up to order 2, a possible improvement can be achieved by using higher order moments which give us more insight about the objects of an image.

One open question from our tool is the stopping criteria in the iterative process of the segmentation algorithm. So far, we fixed a certain number of iterations for it, by doing this we are still not assuring optimum results for this algorithm.

Another important missing component is the optimization for coping with images of patients with Alzheimer Disease. In this specific cases the substructures present shrinking, so our segmentation algorithm should be able to detect the real volume of each substructure.

Relative to performance, we can improve the computation time by making use of a Graphics Processing Unit (GPU). The algorithms proposed are highly parallalizable because most of the operations are computed voxel by voxel, this means that using a GPU for accelerating certain parts of the tool would be beneficial.

At the Initialization stage, we used each of the patients from the training set in order to obtain different predicted contours for a test patient, based on the overlapping with the target contours we selected the best element from the training set to continue with the segmentation algorithm . However, we do have yet a method to select the most appropriate candidate a priori. Finding such method becomes relevant when the training set is very large and it is not possible to analyze all possible candidates.

APPENDIX A

Acronyms

AD Alzheimer's Disease

BET Brain Extraction Tool

CA Cornu Ammonis

CA1 Cornu Ammonis 1

CA2 Cornu Ammonis 2

CA3 Cornu Ammonis 3

DC Dice Coefficient

DG Dentate Gyrus

EC Entorhinal Cortex

FLIRT Linear Image Registration Tool

FSL Fmrib Software Library

HF Hippocampal Formation

IBSR Internet Brain Segmentation Repository

LUMC Leids Universitair Medisch Centrum

MASV Multi-atlas segmentation and voting

MRI Magnetic Resonance Image

NCC Normalized Cross Correlation

ROI Region Of Interest

Sub Subiculum

SDF Signed Distance Function

UMC Universitair Medisch Centrum Utrecht

APPENDIX B

Parameter File Elastix

B.1 Affine Registration Elastix Parameter

```
(FixedInternalImagePixelType "float")
(MovingInternalImagePixelType "float")
(FixedImageDimension 3)
(MovingImageDimension 3)
(UseDirectionCosines "true")
(Registration "MultiResolutionRegistration")
(Interpolator "BSplineInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(FixedImagePyramid "FixedRecursiveImagePyramid")
(MovingImagePyramid "MovingRecursiveImagePyramid")
(Optimizer "AdaptiveStochasticGradientDescent")
(Transform "AffineTransform")
(Metric "AdvancedMattesMutualInformation")
(AutomaticScalesEstimation "true")
(AutomaticTransformInitialization "true")
(HowToCombineTransforms "Compose")
(NumberOfHistogramBins 32)
(ErodeMask "false")
(NumberOfResolutions 1)
(MaximumNumberOfIterations 250)
(NumberOfSpatialSamples 2048)
(NewSamplesEveryIteration "true")
(ImageSampler "Random")
(BSplineInterpolationOrder 1)
(FinalBSplineInterpolationOrder 3)
(DefaultPixelValue 0)
(WriteResultImage "true")
(ResultImagePixelType "short")
(ResultImageFormat "nii")
```

B.2 BSpline Registration Elastix Parameter

```
(FixedInternalImagePixelType "float")
(MovingInternalImagePixelType "float")
(FixedImageDimension 3)
(MovingImageDimension 3)
(UseDirectionCosines "true")
(Registration "MultiResolutionRegistration")
(Interpolator "BSplineInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(FixedImagePyramid "FixedRecursiveImagePyramid")
(MovingImagePyramid "MovingRecursiveImagePyramid")
(Optimizer "AdaptiveStochasticGradientDescent")
(Transform "BSplineTransform")
(Metric "AdvancedMattesMutualInformation")
(FinalGridSpacingInVoxels 32)
(HowToCombineTransforms "Compose")
(NumberOfHistogramBins 32)
(ErodeMask "false")
(NumberOfResolutions 1)
(MaximumNumberOfIterations 500)
(NumberOfSpatialSamples 2048)
(NewSamplesEveryIteration "true")
(ImageSampler "Random")
(BSplineInterpolationOrder 1)
(FinalBSplineInterpolationOrder 3)
(DefaultPixelValue 0)
(WriteResultImage "true")
(ResultImagePixelType "short")
(ResultImageFormat "nii")
```

Bibliography

[1] K. Van Leemput, A. Bakkour, T. Benner, G. Wiggins, L. L. Wald, J. Augustinack, B. C. Dickerson, P. Golland, and B. Fischl, "Automated segmentation of hippocampal subfields from ultra-high resolution in vivo mri," *Hippocampus*, vol. 19, no. 6, pp. 549–557, 2009. [Online]. Available: http://dx.doi.org/10.1002/hipo.20615

[2] M. Uzunbas, O. Soldea, D. Unay, M. Cetin, G. Unal, A. Ercil, and A. Ekin, "Coupled nonparametric shape and moment-based intershape pose priors for multiple basal ganglia structure segmentation," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 12, pp. 1959 –1978, dec. 2010.

[3] O. T. Carmichael, H. A. Aizenstein, S. W. Davis, J. T. Becker, P. M. Thompson, C. C. Meltzer, and Y. Liu, "Atlas-based hippocampus segmentation in alzheimers disease and mild cognitive impairment," *NeuroImage*, vol. 27, pp. 979–990, 2005.

[4] L. Wang, J. S. Swank, I. E. Glick, M. H. Gado, M. I. Miller, J. C. Morris, and J. G. Csernansky, "Changes in hippocampal volume and shape across time distinguish dementia of the alzheimer type from healthy aging," *NeuroImage*, vol. 20, no. 2, pp. 667 – 682, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811903003616

[5] P. Andersen, R. Morris, D. Amaral, T. Bliss, and J. O'Keefe, *The Hippocampus Book*, 1st ed., ser. Oxford Neuroscience. Oxford University, 2006.

[6] O. Soldea, N. Doan, A. Webb, M. Van Buchem, and R. Milles, Jasinschi, "Simultaneous brain structures segmentation combining shape and pose factors," vol. 1, Jun. 2005, pp. 886–893 vol. 1. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[7] K. Van Leemput, "Encoding probabilistic brain atlases using bayesian inference," *Medical Imaging, IEEE Transactions on*, vol. 28, no. 6, pp. 822 –837, june 2009.

[8] P. A. Yushkevich, H. Wang, J. Pluta, S. R. Das, C. Craige, B. B. Avants, M. W. Weiner, and S. Mueller, "Nearly automatic segmentation of hippocampal subfields in in vivo focal t2-weighted mri." *NeuroImage*, vol. 53, no. 4, pp. 1208–1224, 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20600984

[9] J. Pluta, S. Mueller, C. Craige, and P. Yushkevich, *Hippocampal Subfield Segmentation Protocol at 4T*.

[10] J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," in *CVGIP: Graphical Models and Image Processing*, 1992, pp. 438–460.

[11] J. Meriam and K. L.G., *Engineering Mechanics Dynamics 5th Edition.* John Wiley & Sons, Incorporated, 2003.

[12] S. Strother, "Evaluating fmri preprocessing pipelines," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 25, no. 2, pp. 27 –41, march-april 2006.

[13] S. Klein, M. Staring, K. Murphy, M. Viergever, and J. Pluim, "elastix: A toolbox for intensity-based medical image registration," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 1, pp. 196 –205, jan. 2010.

[14] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*, 2nd ed., Kitware, Inc. ISBN 1-930934-15-7, http://www.itk.org/ItkSoftwareGuide.pdf, 2005.

[15] P. Thevenaz and M. Unser, "Optimization of mutual information for multiresolution image registration," *Image Processing, IEEE Transactions on*, vol. 9, no. 12, pp. 2083 – 2099, dec 2000.

[16] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes, "Nonrigid registration using free-form deformations: application to breast mr images," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 8, pp. 712 –721, aug. 1999.

[17] T. Chan and L. Vese, "Active contours without edges," *Image Processing, IEEE Transactions on*, vol. 10, no. 2, pp. 266 –277, feb 2001.

[18] The internet brain segmentation repository ibsr. [Online]. Available: http://www.cma.mgh.harvard.edu/ibsr/

[19] Flirt- fmrib's linear image registration tool. [Online]. Available: http://fsl.fmrib.ox.ac.uk/fsl/flirt/

[20] H. Wang, S. Das, J. Pluta, C. Craige, M. Altinay, M. Weiner, S. Mueller, and P. Yushkevich, "Shape-based semi-automatic hippocampal subfield segmentation with learning-based bias removal," in *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, april 2010, pp. 424 –427.

[21] P. Viola and W. M. Wells, III, "Alignment by maximization of mutual information," *Int. J. Comput. Vision*, vol. 24, pp. 137–154, September 1997. [Online]. Available: http://dl.acm.org/citation.cfm?id=263008.263015

[22] Y. Chen, S. Thiruvenkadam, F. Huang, K. Gopinath, and R. Brigg, "Simultaneous segmentation and registration for functional mr images," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 1, 2002, pp. 747 – 750 vol.1.

[23] J. Flusser and T. Suk, "A moment-based approach to registration of images with affine geometric distortion," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 32, no. 2, pp. 382 –387, mar 1994.

[24] K. Van Leemput, "Probabilistic brain atlas encoding using bayesian inference," in *Proceedings of the 9th international conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I*, ser. MICCAI'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 704–711. [Online]. Available: http://dx.doi.org/10.1007/11866565_86

[25] A. Akselrod-Ballin, M. Galun, J. M. Gomori, A. Brandt, and R. Basri, "Prior knowledge driven multiscale segmentation of brain mri," in *Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention*, ser. MICCAI'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 118–126. [Online]. Available: http://dl.acm.org/citation.cfm?id=1775835.1775853

[26] T. Rohlfing, D. Russakoff, and C. Maurer, "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 8, pp. 983 –994, aug. 2004.

[27] Brain extraction tool. [Online]. Available: http://www.fmrib.ox.ac.uk/fsl/bet2/index.html

[28] Fmrib software library. [Online]. Available: http://www.fmrib.ox.ac.uk/fsl/

[29] Insight segmentation and registration toolkit. [Online]. Available: http://www.itk.org/

[30] Visualization toolkit. [Online]. Available: http://www.vtk.org/