

Master Thesis



Czech  
Technical  
University  
in Prague

**F3**

Faculty of Electrical Engineering  
Department of Cybernetics

## Mapping the changes of the human brain in neurodegenerative diseases

Bc. Milan Němý

Supervisor: Ing. Lenka Vysloužilová, Ph.D.

Field of study: Biomedical Engineering and Informatics

Subfield: Biomedical Engineering

December 2016



## DIPLOMA THESIS ASSIGNMENT

**Student:** Bc. Milan N ě m ý

**Study programme:** Biomedical Engineering and Informatics

**Specialisation:** Biomedical Engineering

**Title of Diploma Thesis:** Mapping the Changes of the Human Brain in Neurodegenerative Diseases

### Guidelines:

1. Study libraries for processing of MRI images, particularly the ANTS library.
2. Design and implement a series of transformations that allow the measurement of atrophy among given population of individuals.
3. Test the whole process of transformation.
4. Evaluate statistical differences in brain changes in given population.

### Bibliography/Sources:

- [1] AVANTS, Brian B.; TUSTISON, Nick; SONG, Gang. Advanced normalization tools (ANTS). Insight J, 2009, 2: 1-35.
- [2] HUA, Xue, et al. Tensor-based morphometry as a neuroimaging biomarker for Alzheimer's disease: an MRI study of 676 AD, MCI, and normal subjects. Neuroimage, 2008, 43.3: 458-469.
- [3] FRACKOWIAK, Richard SJ, et al. Human brain function. San Diego, CA, 2004.

**Diploma Thesis Supervisor:** Ing. Lenka Vysloužilová, Ph.D.

**Valid until:** the end of the winter semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic  
**Head of Department**

prof. Ing. Pavel Ripka, CSc.  
**Dean**

Prague, May 23, 2016



## Acknowledgements

I would like to express my gratitude to my supervisor Ing. Lenka Vysloužilová, Ph.D. for the useful comments, remarks and engagement through the process of creating this master thesis.

My gratitude extends to MUDr. Zuzana Nedělská for her willingness to provide both data and clinical expertise.

Furthermore, I would like to thank Mgr. Josef Urban, Ph.D. for providing me with a high-level computation capacity for performing key calculations.

## Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 15. December 2016

## Abstract

This master thesis deals with the MR image processing of patients suffering from neurodegenerative disorders, in particular with the Alzheimer's disease. The aim was to explore various tools for image processing and subsequent analysis and to construct a transformation pipeline for brain atrophy evaluation.

The thesis presents morphometric techniques based on a non-linear image registration, especially the tensor-based morphometry with the SyN registration method from the ANTs bundle.

The proposed solution allows the evaluation of atrophic rate of individuals, but also to statistically assess atrophic brain locations across the whole population. Based on this, several algorithms for automatic classification of atrophied brains were proposed.

**Keywords:** magnetic resonance imaging, MRI, Alzheimer's disease, digital image processing, registration, tensor-based morphometry, SyN, ANTs, atrophy, classification

**Supervisor:** Ing. Lenka Vysloužilová, Ph.D.

## Abstrakt

Tato diplomová práce řeší problematiku zpracování obrazu z magnetické rezonance pacientů s neurodegenerativními onemocněními, zejména pak s Alzheimerovou chorobou. Cílem byl průzkum prostředků pro zpracování obrazu a následnou analýzu a sestavení řetězu vhodných transformací, který by umožňoval hodnocení mozkové atrofie.

Práce zejména představuje morfometrické metody postavené na nelineární registraci MR snímků, především pak morfometrii založenou na tenzorech za použití registrační metody SyN z balíčku ANTs.

Vytvořené řešení umožňuje jednak vyhodnotit míru atrofie v mozku jednotlivců, tak i statisticky stanovit atrofická místa v celé populaci nemocných. Na základě toho bylo navrženo několik algoritmů pro automatickou klasifikaci mezi nemocnými a zdravými pacienty.

**Klíčová slova:** magnetická rezonance, MRI, Alzheimerova choroba, zpracování digitálního obrazu, registrace, morfometrie založená na tenzorech, SyN, ANTs, atrofie, klasifikace

**Překlad názvu:** Mapování změn lidského mozku u neurodegenerativních onemocnění

# Contents

<b>1 Introduction</b>	<b>1</b>	3.4 Jacobian Determinant Image ...	50
<b>2 Background knowledge</b>	<b>3</b>	3.5 Statistical studies .....	54
2.1 Magnetic Resonance Imaging .....	3	3.5.1 Statistical Parametric Mapping	54
2.1.1 Brief History .....	3	3.5.2 Classification .....	54
2.1.2 Tomographic Imaging .....	4	<b>4 Conclusion</b>	<b>61</b>
2.1.3 Spin .....	4	<b>A Bibliography</b>	<b>63</b>
2.1.4 Basic Principles of NMR .....	4		
2.1.5 Spatial encoding of MRI .....	6		
2.1.6 Tomographic reconstruction ..	6		
2.2 Alzheimer's Disease .....	6		
2.2.1 The Hallmarks of AD [Rod08]	7		
2.2.2 The Changing Brain in AD...	7		
2.2.3 Use of MRI in Alzheimer's			
disease .....	8		
2.3 Brain Morphometry .....	8		
2.3.1 Voxel-Based Morphometry			
(VBM) .....	9		
2.3.2 Deformation-Based			
Morphometry (DBM) .....	9		
2.3.3 Tensor-Based Morphometry			
(TBM) .....	9		
2.4 TBM Pipeline .....	10		
2.4.1 Image Pre-Processing Pipeline	12		
2.4.2 Average Group Template			
(MDT) .....	13		
2.4.3 Warping .....	13		
2.4.4 Cross-Sectional TBM .....	13		
2.4.5 Longitudinal TBM .....	13		
2.4.6 Morphometric measures .....	15		
2.4.7 Classification .....	15		
2.5 Image Registration .....	17		
2.5.1 Point registration .....	17		
2.5.2 Surface matching .....	18		
2.5.3 Registration with intensities .	18		
2.5.4 Interpolation .....	19		
2.5.5 Transformation models .....	20		
2.6 Analysis tools for MRI .....	24		
2.6.1 MRI data formats .....	24		
2.6.2 MRICron .....	25		
2.6.3 FSL .....	26		
2.6.4 ANTs .....	32		
2.6.5 ANTsR .....	40		
<b>3 Pipeline and Data Analysis</b>	<b>43</b>		
3.1 Data Overview .....	43		
3.2 Image Pre-Processing Pipeline ..	44		
3.3 Average Template .....	46		

## Figures

2.1 Axial T1-weighted, proton-density weighted, T2-weighted, and FLAIR images (from left to right) [BFBLS11]	5	3.14 Decision tree diagram	60
2.2 Mild to moderate AD [Rod08]	8		
2.3 MR Image Pre-processing Pipeline	11		
2.4 Intensity inhomogeneity in MR brain image. [VPL07]	12		
2.5 TBM	14		
2.6 Comparison of different transformation models [ATS09]	24		
2.7 Example of input and outputs of BET	27		
2.8 Example outputs of FAST (partial volumes (PV) and binary segmentation (BS))	29		
2.9 Comparison of unregistered and FLIRT registered volumes (moving image – green, fixed image – grey)	31		
2.10 Process of creating an average group template	41		
3.1 Original head MRI scan before pre-processing	44		
3.2 Head MR image after bias correction	45		
3.3 Skull stripped head MR image (orthogonal slices)	47		
3.4 Skull stripped head MR image (surface rendering)	47		
3.5 Temporary average brains in the process of building an Average Template	49		
3.6 Average Template	50		
3.7 Histogram and clipping of a log jacobian determinant image	51		
3.8 Suggested overlay colormap	52		
3.9 Jacobian determinant map (in the template space)	52		
3.10 Jacobian determinant map (in the individual space)	53		
3.11 p-value map	55		
3.12 Manual segmentation of cerebellum (red), temporal lobes (green) and lateral ventricles (blue)	56		
3.13 Automatic segmentation of white matter (blue), grey matter (green) and cerebrospinal fluid (red)	57		



## Tables

2.1 Contingency table for classifier evaluation .....	16
2.2 Transformations and a subset of the similarity metrics available in ANTs [ATS09] .....	33
3.1 Classification attributes .....	58
3.2 Evaluation of classification .....	59





# Chapter 1

## Introduction

Magnetic resonance imaging (MRI) is an imaging technique based on nuclear magnetic resonance (NMR), a method to obtain spatial and structural information about molecules. [MMGP07] Initially, this modality served for tomographic imaging of the human body. With the advance of technology, not only were thin slices produced, but advanced volume imaging techniques were also introduced. These days it is a common practise to analyse slice images, volume and tensor data among others.

In spite of the cost of an MRI examination, it is becoming increasingly popular. The reasons might be the following: MRI does not expose the patient to a harmful radiation as opposed to CT or conventional X-Ray. MRI scanners do not emit any ionizing radiation and thus patients can undergo MRI examination repeatedly without increased risk. Secondly, MRI provides much better contrast for soft tissue examination such as brain tissue, spinal chord, ligaments, tendons, etc. Because of this property, MRI is widely used for capturing brain structures. It can especially help diagnose conditions such as brain tumors, causes of epilepsy, cortical and hippocampal atrophy and multiple sclerosis.

Various post-processing methods can be applied to observe valuable underlying information. Brain morphometry is one of the most popular one as it concerns with measurement of brain structures and their development with the help of image data. [FFF<sup>+</sup>04] The most significant advantage of this family of methods is the ability to compare corresponding brain structures of different subjects even though their brains can seem to differ considerably in size and proportions.

This theses focuses on utilization of MRI in evaluation of patients with Alzheimer's disease, in particular on brain morphometry. Special attention will be paid to MRI data formats, description and evaluation of commonly used brain MRI processing software bundles. Especially, ANTs library will be thoroughly examined as it provides a set of powerful normalization methods suitable for morphometry design.

Using mainly the ANTs library, a transformation pipeline for evaluation of brain atrophy among given population of individuals will be designed and implemented. Consequently, statistical differences in brain changes will be evaluated.

The aim is to find the brain locations which are liable to atrophic changes even before the disease passes on to a clinical stage and thus to provide a powerful tool for neurologists. This could possibly lead to an early diagnosis and with the help of correct medication, patients could live longer and with less associated unpleasant symptoms.

This thesis is organized into several chapters. First, the introduction presents the main theme and sets up desired aims and objectives. In the second chapter, basic information about magnetic resonance imaging, Alzheimer's disease and brain morphometry is provided as well as a brief description of several MRI processing toolboxes. The third chapter shows step by step the creation of image processing pipeline for brain morphometry and presents the reader with its numeric outcomes for provided data. In the last chapter, results of the thesis are evaluated and compared to the aims set in the introduction. Additionally, the future prospects and possible continuation of the project are mentioned.

## Chapter 2

### Background knowledge

In the following chapter, a brief account on magnetic resonance imaging, Alzheimer's disease and brain morphometry is given as well as a short description of a couple of MRI processing toolboxes which are utilized in the thesis.

#### 2.1 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is an imaging technique used foremost in medical settings to produce high-quality images of the inside of the human body. This method is based on principles of nuclear magnetic resonance (NMR), spectrographic method used to obtain chemical and physical information about molecules. The medical modality was, however, called magnetic resonance rather than nuclear magnetic resonance (NMRI) because of some negative connotations of the word *nuclear* in the late 1970s.

Originally, MRI started out as a tomographic method, i.e., for capturing of NMR signal in thin slices across the human body. By these days MRI has evolved from tomography to a volume imaging technique.

In this theses, head MRI scans are used as an input for a subsequent analysis of brain atrophy.

##### 2.1.1 Brief History

In 1946, Felix Bloch and Edward Purcell independently discovered the magnetic resonance phenomenon, for which they were awarded with the Nobel Prize in 1952. Between 1950 and 1970, NMR was massively developed and used for chemical and physical analysis.

In 1971, Raymond Damadian showed that magnetic resonance relaxation times of healthy tissues and tumors significantly differ and thus he motivated scientists to exploit magnetic resonance for disease detection. [Dam71]

Magnetic resonance imaging was demonstrated for the first time on a small test tube by Paul Lauterbur in 1973. A back projection technique similar to that one in CT was employed. [Lau73] In 1975, Richard Ernst proposed MRI based on frequency and phase coding and Fourier transformation. That forms the basis of the present MRI. [KWE75]

### ■ 2.1.2 Tomographic Imaging

MRI started out as a tomographic modality, i.e., producing NMR slice images across the human body. Each such slice has a certain thickness and it consists of many volume units, so-called *voxels*. The volume of a 1.5 T MRI voxel can be approximately  $2 \text{ mm}^3$ . Resulting image generally consists of many image points, which are called *pixels*. The intensity of a pixel is proportional to the intensity of NMR signal from the corresponding volume element.

MRI is based on absorption and emission of energy in a radio-frequency domain of electromagnetic field. From the attenuation spectrum of the human body it is clear why the conventional X-Ray was commonly used for imaging. However, why it took so long before imaging with the help of radio waves was developed? Many scientists used to believe that it is impossible to depict objects smaller than the wavelength of the energy, which is used for imaging. MRI got around this limitation by spatial variation in the phase and frequency of radio frequency energy, which the imaged object absorbs and emits. [MMGP07]

### ■ 2.1.3 Spin

The human body consists above all of water and fat that are composed of a huge number of hydrogen atoms which in fact makes the human body approximately 63 % hydrogen atoms. Nuclei of hydrogen are able to produce NMR signal. On that account MRI shows predominantly NMR signal from hydrogen nuclei. A single voxel of the image of the human body contains one or more tissues. Each of them contains living cells, in which water molecules are found. Each water molecule is compounded by one oxygen atom and two hydrogen atoms. Each hydrogen atom has a nucleus consisting of a single proton possessing a property called *spin*: [MMGP07]

- Spin can be modelled as a very tiny magnetic field.
- Spin causes the nucleus to create the NMR signal.
- Not all nuclei possess non-zero spin.

### ■ 2.1.4 Basic Principles of NMR

To make things easier, let's assume the classical (non-quantum) model in which one can interpret spin as a small magnet.

Without any external magnetic field, the orientation of spins can be described as random and thus the net magnetization  $\mathbf{M}$  equals zero. However, a tissue in a MRI scanner is exposed to a very strong magnetic field ( $B_0=1.5 \text{ T}$ ). It is energetically favorable for spins to align parallel to the  $B_0$  field, let call this direction  $z$ , by which means an external magnetization  $\mathbf{M}$  arises. Nevertheless, the net magnetization is only about 5 ppm of the maximal possible one because of the heat motion. This phenomenon can be described by the Maxwell-Boltzmann statistics. [Kyb]

According to the quantum mechanics, the spin is somewhat deflected from the field  $B_0$  and thus it *precess* around  $z$  axis. Precession frequency is called *Larmor frequency* and it equals  $f = \gamma B$ , where  $\gamma$  is *gyromagnetic constant*.  $\gamma = 42.58 \text{ MHz/T}$  for hydrogen. [Kyb]

A rotating spin can absorb energy from an radio-frequency pulse with the frequency  $f$ . It causes the magnetization axis to deflect from  $z$  axis. When an pulse with an appropriate energy is applied, the spin will be rotating in the  $xy$ -plane ( $90^\circ$ -pulse). All spins in the tissue will be rotating with the same phase, therefore a measurable net magnetization arises which will be rotating with frequency  $f$  in the  $xy$ -plane and which one can detect with a suitable receive coil. This signal is usually called *echo*.

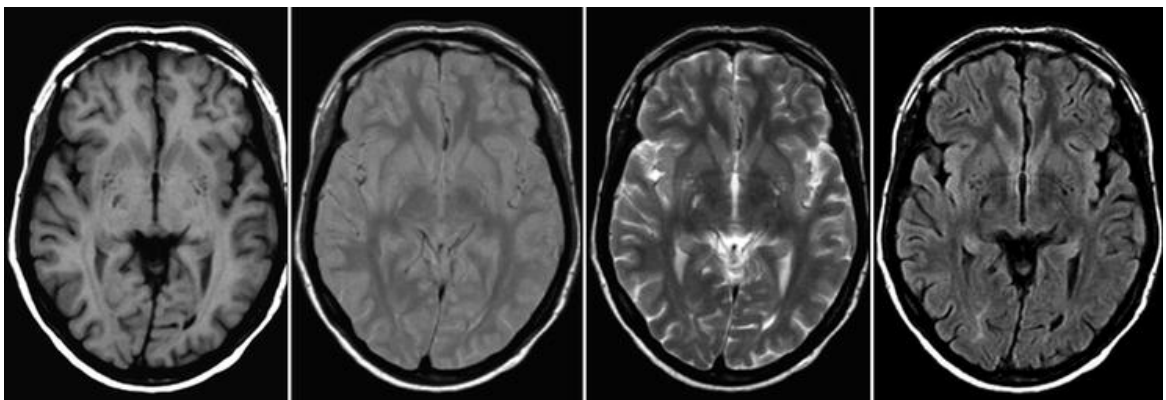
After the end of the pulse the magnetization is falling back to its equilibrium (relaxation) with a time constant  $T_1$  (spin-lattice relaxation time). Another reason for the signal intensity attenuation is the spin desynchronization due to mutual interactions and magnetic field non-homogeneity – "spin-spin" relaxation with a transversal relaxation time  $T_2$ .  $T_1$  and  $T_2$  relaxation times are specific for different tissues.

Only (rotating) transversal magnetization induces signal into reception coils. Detection of longitudinal magnetization  $M_z$  requires renewed flipping of recovered longitudinal magnetization into  $xy$ -plane by  $90^\circ$ -pulse. This sequence is called *saturation recovery* – recovery of longitudinal magnetization after initial  $90^\circ$ -pulse by repeated  $90^\circ$ -pulse at time  $T_R$  (repetition time). If  $T_R$  is set short enough, the resulting signal is  *$T_1$ -weighted*.

*Spin-echo sequence* is another frequently used sequence. Its goal is the compensation of dephasing by sequence of appropriate transversal field pulses at time  $T_E/2$  (echo time). If the echo time is set long enough, the resulting signal is  *$T_2$ -weighted*. [Suc15]

For a long  $T_R$  time and short  $T_E$  time, the intensity of received signal hardly depends on the time constants of the tissue. Instead, the intensity is proportional to the proton density and thus we call this contrast *PD (proton density) contrast*.

There are also another more complex sequences such as *FLAIR*, *TSE*, etc.



**Figure 2.1:** Axial T1-weighted, proton-density weighted, T2-weighted, and FLAIR images (from left to right) [BFBLS11]

### 2.1.5 Spatial encoding of MRI

Reception coils acquire vector sum (or integral) over induced signal. Unambiguous discrimination of position along  $x$ -,  $y$ - and  $z$ -coordinates is achieved by gradients of  $B_z$ -field.

A gradient  $G_z$  is applied during RF pulse which makes the magnetic field intensity  $B_z$  and resonance frequency dependent on a position. Only spins in a thin slice are flipped by transversal field pulse – *slice selection*. Bandwidth of the excitation impulse determines the thickness of the slice.

Encoding of  $x$ -coordinate is carried out by gradient  $G_x$  during readout. It causes the precession frequency to vary with  $x$ -coordinate – *frequency coding*.

Finally, encoding of  $y$ -coordinate is accomplished by gradient  $G_y$  applied for a fixed interval  $T_y$  between RF pulse and readout. Now spins emit electromagnetic waves with phase shift  $\varphi_y(y)$  dependent on  $y$ -position – *phase coding*.

### 2.1.6 Tomographic reconstruction

The signal  $S(f_x, f_y, z_0)$  received by the coils is the integral over transversal magnetization phasor in the entire slice  $z = z_0$ . This signal is demodulated in a quadrature modulator. It holds that the demodulated signal is the *Fourier transform* of the transversal magnetization at slice  $z_0$ . Frequencies  $f_x, f_y$  are often called wave numbers  $k_x, k_y$ . Therefore the MRf slice acquisition proceeds by "sampling in  $k$ -space".  $k_x$  is sampled by sampling of time with sampling interval  $\Delta t_x$ .  $k_y$  by sampling of gradient  $G_y$  in steps  $\Delta G_y$ . [Suc15]

Once the  $k$ -space is sampled dense enough, the (weighted-)magnetization image is recovered by inverse 2D Fourier transform.

## 2.2 Alzheimer's Disease

Alzheimer's disease (AD) is the most common form of dementia, affecting over 40 million people worldwide. [SBB<sup>+</sup>16] Although the exact time course is unknown, AD-related pathogenesis is believed to begin decades before clinical symptoms, such as memory impairment, can be detected. As AD develops, patients suffer from progressive decline in executive function, language, and other cognitive and behavioral domains.

MRI is widely used in AD studies as it can non-invasively quantify gray and white matter integrity with high reproducibility. MRI-based measures have been used in recent clinical trials and they have been shown to correlate with pathologically confirmed neuronal loss and with the molecular hallmarks of AD. [HLP<sup>+</sup>08]

For these reasons, we attempt to measure and quantify brain changes in AD on the basis of MRI data. On top of that, brain atrophy among given population of individuals will be evaluated.



### ■ 2.2.1 The Hallmarks of AD [Rod08]

The brains of people with AD have an abundance of two abnormal structures – *amyloid plaques* and *neurofibrillary tangles* – that are made of misfolded proteins. This is especially true in certain regions of the brain that are important in memory. The third main feature of AD is the loss of connections between cells. This leads to diminished cell function and cell death.

*Amyloid plaques* are found in the spaces between the brain's nerve cells. They consist of largely insoluble deposits of an apparently toxic protein peptide, or fragment, called *beta-amyloid*. We now know that some people develop some plaques in their brain tissue as they age. However, the AD brain has many more plaques in particular brain regions.

The second hallmark of AD is *neurofibrillary tangles*. Tangles are abnormal collections of twisted protein threads found inside nerve cells. The chief component of tangles is a protein called *tau*. Healthy neurons are internally supported in part by structures called microtubules, which help transport nutrients and other cellular components, such as neurotransmitter-containing vesicles, from the cell body down the axon. Tau, which usually has a certain number of phosphate molecules attached to it, binds to microtubules and appears to stabilize them. In AD, an abnormally large number of additional phosphate molecules attach to tau. As a result of hyperphosphorylation, tau disengages from the microtubules and begins to come together with other tau threads. These tau threads form structures called paired helical filaments, which can become enmeshed with one another, forming tangles within the cell. The microtubules can disintegrate in the process, collapsing the neuron's internal transport network. This collapse damages the ability of neurons to communicate with each other.

The third major feature of AD is the gradual *loss of connections between neurons*. Neurons live to communicate with each other, and this vital function takes place at the synapse. The AD process not only inhibits communication between neurons but can also damage neurons to the point that they cannot function properly and eventually die. As neurons die throughout the brain, affected regions begin to shrink in a process called brain atrophy. By the final stage of AD, damage is widespread, and brain tissue has shrunk significantly.

### ■ 2.2.2 The Changing Brain in AD

AD begins deep in the brain, in the entorhinal cortex, a brain region that is near the hippocampus and has direct connections to it. Healthy neurons in this region begin to work less efficiently, lose their ability to communicate, and ultimately die. This process gradually spreads to the hippocampus, the brain region that plays a major role in learning and is involved in converting short-term memories to long-term memories. Affected regions begin to atrophy. Ventricles, the fluid-filled spaces inside the brain, begin to enlarge as the process continues, as shown in Fig. 2.2. [Rod08]

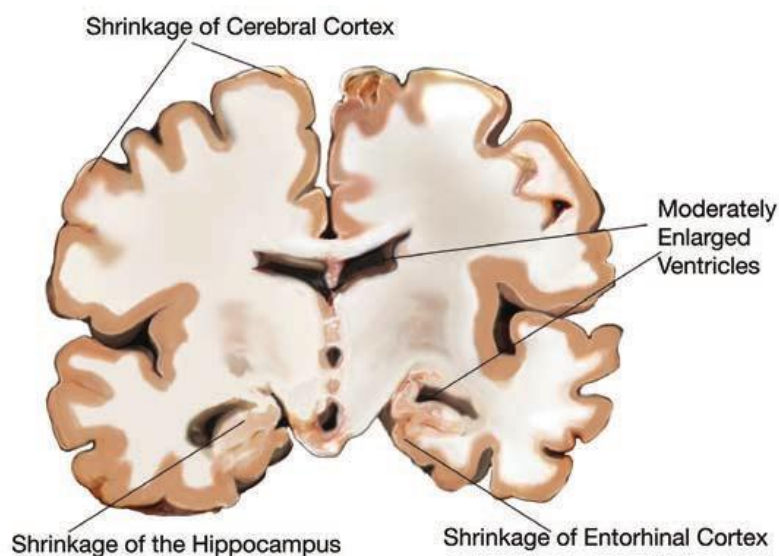


Figure 2.2: Mild to moderate AD [Rod08]

### 2.2.3 Use of MRI in Alzheimer's disease

MRI-based measures of atrophy are regarded as valid markers of disease state and progression for several reasons. Atrophy seems to be an inevitable, inexorably progressive concomitant of neurodegeneration. The topography of brain tissue loss correlates well with cognitive deficits, both cross-sectionally and longitudinally.

Rates of change in several structural measures, including whole-brain, entorhinal cortex, hippocampus and temporal lobe volumes, as well as ventricular enlargement, correlate closely with changes in cognitive performance, supporting their validity as markers of disease progression. [FFJ<sup>+</sup>10]

## 2.3 Brain Morphometry

Morphometric methods are a process of numerically identifying and characterizing *structural differences* among populations, or for finding relation between brain shape and, for example, disease severity.

Morphometric methods have a number of different aims. They can be used for localising significant structural differences among populations, or for showing that overall brain structure is related to some effect of interest. When testing the overall brain structure, multivariate statistical methods are used to analyse groups of parameters for the whole brain (e.g., the deformation-based morphometry, see below). The result of the forms of morphometry that localise structural differences would typically be a statistical parametric map of regional differences. Statistical parametric maps (SPMs) can be derived from uni-variate data where there is a single variable at each voxel (e.g., the

voxel-based morphometry, see below), or from multi-variate data, where there are several different variables at each voxel (e.g., tensor-based morphometry, see below).

Another use for morphometric methods is for characterising essential differences, or for producing some form of classification. Linear methods such as correlation analysis, or nonlinear classification methods can be used for these purposes. [AF01]

In our case, a method of tensor-based morphometry is utilized in order to obtain volumetric differences between corresponding locations in different brains. Thus, one might be able to localize excessively expanding or shrinking brain structures.

### ■ 2.3.1 Voxel-Based Morphometry (VBM)

VBM is a technique for producing SPMs of volumetric differences. It performs a voxel-wise comparison of the local concentration of grey matter between two group of subjects.

Voxel-based morphometry of MRI data involves spatially normalising all the subjects' images to the same stereotactic space, extracting the grey matter from the normalised images, smoothing, and finally performing a statistical analysis to localise, and make inferences about group differences.

The output from the method is an SPM showing regions where grey matter concentration differs significantly among the groups. [AF01]

### ■ 2.3.2 Deformation-Based Morphometry (DBM)

Deformation-based morphometry is a characterisation of the differences in the vector fields that describe global or gross differences in brain shape. These vector fields are the deformation fields used to effect nonlinear variants of spatial normalisation, when one of the images is a template that conforms to some standard anatomical space. [AF01]

### ■ 2.3.3 Tensor-Based Morphometry (TBM)

The objective of TBM is to localize regions of shape differences among groups of brains, based on deformation fields that map points in a template  $(x_1, x_2, x_3)$  to equivalent points in individual source images  $(y_1, y_2, y_3)$ . In principle, the Jacobian matrices of the deformations (a 2nd order tensor field relating to the spatial derivatives of the transformation) should be more reliable indicators of local brain shape than absolute deformations. Absolute deformations represent positions of brain structures, rather than local shape, and need to be quantified relative to some arbitrary reference position.

A Jacobian matrix contains information about local stretching, shearing and rotation involved in the deformation, and is defined at each point by:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} \\ \frac{\partial y_3}{\partial x_1} & \frac{\partial y_3}{\partial x_2} & \frac{\partial y_3}{\partial x_3} \end{bmatrix} \quad (2.1)$$

A simple form of TBM involves comparing relative volumes of different brain structures, where the volumes are derived from *Jacobian determinants* at each point. Simple univariate statistics (t- or F- tests) can then be used to make inferences about regional volume differences among populations. This type of morphometry is useful for studies that have specific questions about whether growth or volume loss has occurred.

Spatial normalization of a series of source images involves determining a mapping from each point in the template image to corresponding points in the source image. To compare image shapes, it is necessary to derive measures of shape within the coordinate system of the template image, rather than within the different coordinate systems of the individual source images.

Within a TBM framework, multivariate statistics would be applied to the elements of a strain tensor in order to localize volume, area and length differences.

Rather than basing the tests either on information at each pixel, or on information from the whole brain, it is easy to see that forms of morphometry could be based on regional analysis. For example, multivariate TBM could be applied such that the tests include information from strain tensors in regions of neighboring lattice locations. The regions could be based on pre-defined structures on the template image. [FFF<sup>+</sup>04]

## ■ Types of Studies

There are two types of TBM studies: cross-sectional and longitudinal.

*Cross-sectional TBM* is based on an average brain template of healthy subjects (control group). Then a source image, i.e., the one we want to quantify, is registered with a highly non-linear transformation to the average brain template.

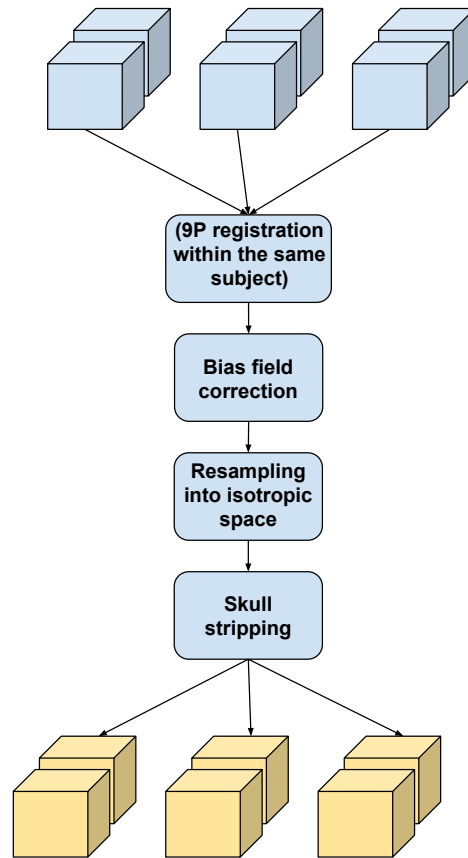
The advantage of this approach is that it does not need follow-up images from the same patient. Only a group of healthy subjects for the creation of the brain template is required. However, cross-sectional TBM does not yield such sensitive results as in case of longitudinal studies.

*Longitudinal TBM* relies on the comparison of follow-up MR brain images from the same person. It usually outputs more reliable results than the cross-sectional TBM but it often faces the problem of lack of follow-up examinations which are usually quite costly and not always available.

## ■ 2.4 TBM Pipeline

The following chapter elaborates more on the topic of the *tensor-based morphometry* as it was chosen as the ideal technique for a precise localization

## Image Pre-processing



**Figure 2.3:** MR Image Pre-processing Pipeline

and measurement of atrophied structures in Alzheimer’s disease. The TBM pipeline consists of several steps. First, the data need to be preprocessed (see Sec. 2.4.1). Then either cross-sectional TBM or longitudinal TBM is performed. The cross-sectional TBM involves creating an average group template and afterward, registering all individual brain images to this template. On the other hand, the longitudinal TBM is applied in case of having several follow-up scans from the same patient available because this method involves aligning a brain scan not to a template but to an older brain image from the same subject. The outcome of both of these approaches is a Jacobian matrix field which tells which parts of the brain are expanding or shrinking and to what extent. Usually a statistical analysis based on this field follows (see Sec. 2.4.6). The whole process is also illustrated in Fig. 2.5.

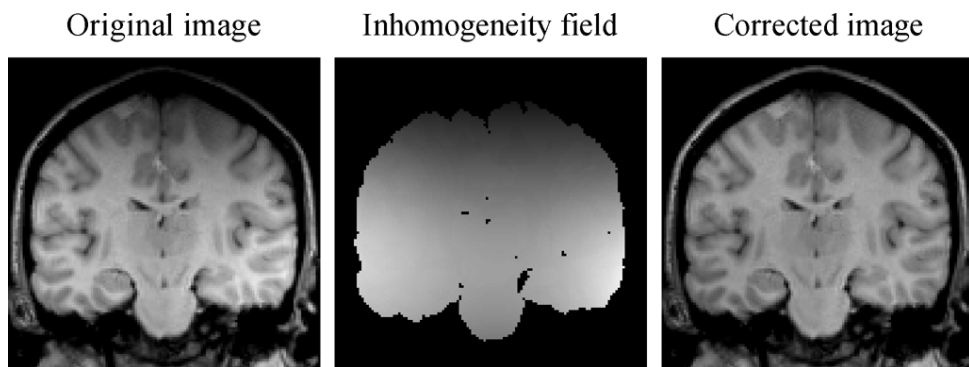
### 2.4.1 Image Pre-Processing Pipeline

It is a good practice to pre-process the input MR images before processing further in the TBM pipeline (see Fig. 2.3).

- (for longitudinal TBM only) linear registration of follow-up scans to adjust for linear drifts in head position and scale within the same subject.
- An input image is *resampled* into isotropic space, e.g., with a final voxel volume of  $1 \text{ mm}^3$ . One may also down-sample the image to speed up the entire pipeline, however, at the expenses of resolution and information loss. Resampling also requires a sensible choice of interpolator.
- MRI suffers from an imaging artifact commonly referred to as *intensity inhomogeneity* or *bias field*, which appears as low-frequency multiplicative noise in the images. It rises from the imperfections of the image acquisition process and manifests itself as a smooth intensity variation across the image, see Fig. 2.4. Because of this phenomenon, the intensity of the same tissue varies with the location of the tissue within the image. Although intensity inhomogeneity is usually hardly noticeable to a human observer, many medical image analysis methods, such as segmentation and registration, are highly sensitive to the spurious variations of image intensities. This is why a number of methods for intensity inhomogeneity correction of MR images have been proposed. [VPL07]

A well-known intensity inhomogeneity correction method is known as the *N3* [LIVL14]. It is a histogram-based method with high-frequency maximization. Its popularity is due to its high performance, automatic image handling and no required prior information about the MRI input.

A variant of N3 algorithm, called *N4ITK*, has been presented in [TAC<sup>+</sup>10]. This variation couples a robust B-spline approximation algorithm with a modified optimization strategy which includes a multiresolution option to capture a range of bias modulation.



**Figure 2.4:** Intensity inhomogeneity in MR brain image. [VPL07]

- *Skull stripping* may be required to improve the robustness of TBM. There are several different approaches to this problem, such as employing

deformable model operating with virtual forces or template-based method. For particular examples, see Section 2.6.3.

### ■ 2.4.2 Average Group Template (MDT)

To construct an MDT, an initial affine average template is created by taking a voxel-wise average of globally aligned scans after intensity normalization. Next, a non-linear average template is built after warping individual brain scans to the affine template. The above steps are repeated until a full-resolution image registration is achieved. Lastly, the MDT is generated by applying inverse geometric centering of the displacement fields to the non-linear average. [HX11] See a detailed process diagram in Fig.2.5a.

### ■ 2.4.3 Warping

The above mentioned non-linear alignment of the individual images to the MDT is often called a *warping*.

Warping is a highly non-linear registration process which locally deforms source to the target image. The quality of TBM extremely depends on the warping algorithm used. Quite popular one is the *non-linear inverse consistent elastic intensity-based registration algorithm*, which optimizes a joint cost function based on mutual information and the elastic energy of the deformation. [HX11]

In recent years a new method called *Symmetric Diffeomorphic Image Normalization method (Syn)* [ATS09] was developed by a team led by B. B. Avants. It is supposed to yield better results and it is implemented in a software bundle ANTs which is heavily used in this thesis.

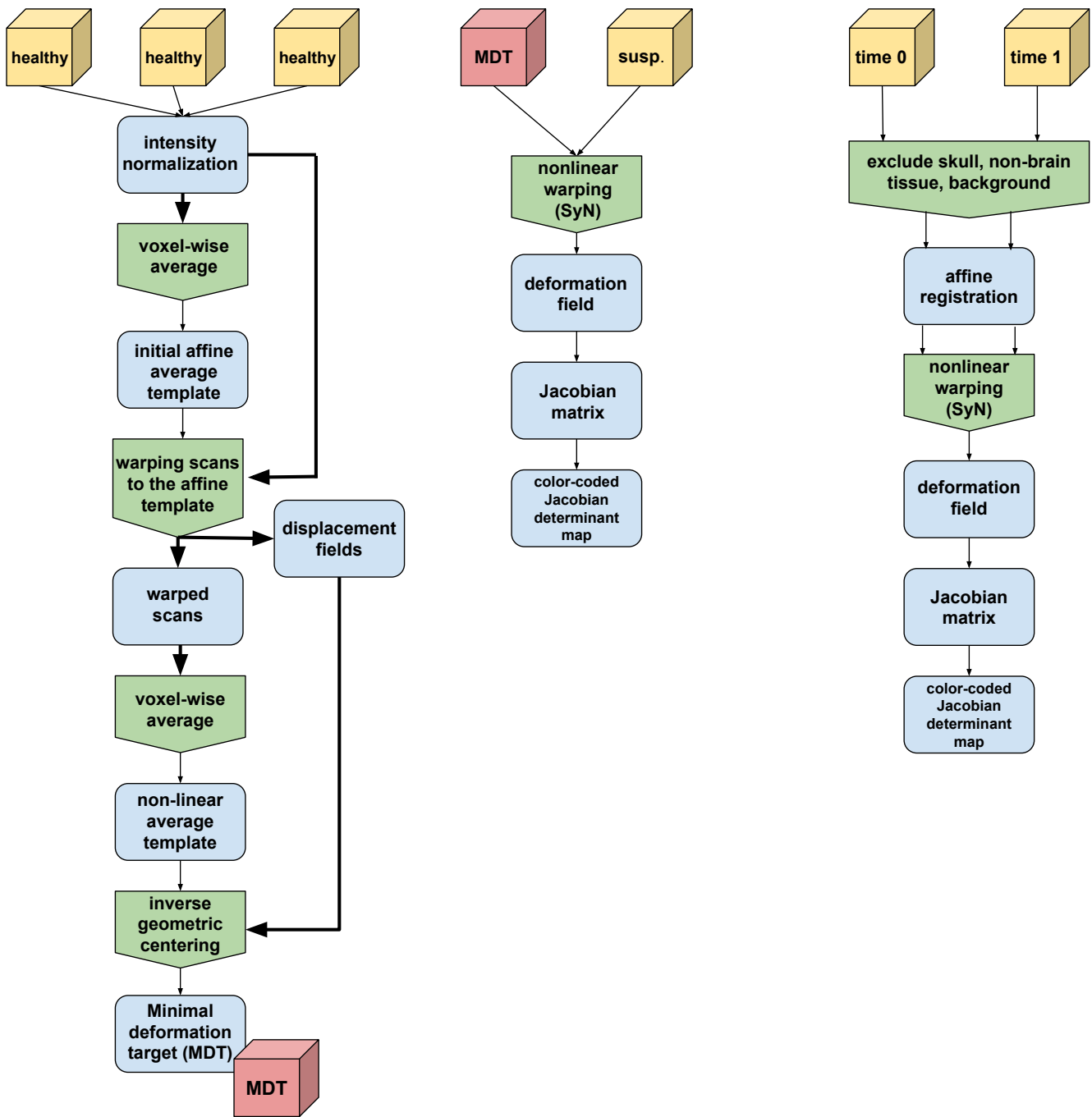
### ■ 2.4.4 Cross-Sectional TBM

After creation of the minimal deformation target (MDT), all individual screening images are aligned (warped) to it. Then, gradients of the deformation field are computed, from which Jacobian matrix is derived. Consequently, the determinant of the local Jacobian matrix is derived from the forward deformation field to characterize local volume differences. Color-coded Jacobian determinants are used to illustrate regions of volume expansion, i.e. those with  $\det \mathbf{J}(r) > 1$ , or contraction, i.e.  $\det \mathbf{J}(r) < 1$  relative to the normal group template (see Fig. 2.5b). [HX11]

### ■ 2.4.5 Longitudinal TBM

Before performing the longitudinal TBM, it is crucial to exclude skull, other non-brain tissue and background from the input images to improve the precision. [HX11]

First, the follow-up scan is linear registered to the baseline scan and then non-linearly warped to it. Afterwards, a Jacobian matrix field is derived from the gradients of the deformation field that aligns these images (see Fig. 2.5c).



(a) : average group template

(b) : cross-sectional TBM

(c) : longitudinal TBM

Figure 2.5: TBM pipeline



### ■ 2.4.6 Morphometric measures

By the end of the TBM pipeline, a numerical analysis follows. There are two types of measures:

- *Voxel-by-voxel measures* – One may use such measure when asking where are the differences between populations. Such measures produce an SPM (statistical parametric mapping) of regional differences.
- *Volume-based measures* – They answer the question whether there is a difference between populations.

A common approach is to compute the measure from the Jacobian map within a region-of-interest (ROI). There are two main definitions of ROI: temporal lobe ROI and statistically-defined ROI.

Temporal lobe ROI, including the temporal lobes of both brain hemispheres, is manually delineated on the MDT template by a trained anatomist.

Statistically-defined ROI is defined based on voxels with significant atrophic rates over time within the temporal lobes.

Statistical tests generally involve disproving a null hypothesis with a particular level of confidence. In morphometry, the null hypothesis is usually that there are no significant structural differences among a number of populations, or due to particular covariates, such as age. The objective of the tests is to demonstrate improbability of the null hypothesis. For example, if p values of less than 0.01 are deemed to be significant, then false positive results would be expected only about once out of a hundred tests.

Performing comparisons at each voxel results in many statistical tests being performed. Without any correction, the number of false-positive results would be proportional to the number of independent tests. [AF01]

The most simple method is the *Bonferroni correction* which sets the alpha value (probability of the false positive result) for the entire set of  $n$  comparisons equal to  $\alpha$  by taking the alpha value for each comparison equal to  $\alpha/n$ .

However, Bonferroni correction is very conservative, i.e., it results in very strict significance levels. In addition, it is not optimal for correlated data. More accurate way to obtain the effective number of independent statistical tests is determined using Gaussian Random Field (GRF) theory [AF01]. By using GRF theory, a correction for multiple dependent comparisons can be made to produce the appropriate rate of false-positive results.

### ■ 2.4.7 Classification

After obtaining one or more morphometric measures, it is convenient to use them for classification purposes. In the simplest case, we assume a binary classifier which discriminates between two classes – healthy individuals and AD subjects. For training such model, a set of records from both groups are needed, i.e., the training set. Each record should contain one or more

attributes, that is some morphometric measures derived by TBM, and a class assessed by a neurologist. Classifiers used in this thesis are the following:

- *Naive Bayes classifier* – Classification based on the Bayes’ theorem and assuming that the attributes are independent. That is getting posterior from prior and likelihood. See [Bis06] for more details.
- *Decision tree* – An algorithm based on a tree-like graph whose nodes contain a simple decision rule operating with a value of a single attribute. According to fulfillment of this rule, the algorithm continues in one of the two possible child subtrees. The node with no child subtrees is a *leaf* specifying the classification. Also see [Bis06].

In the training phase, such attribute is chosen so that it minimizes entropy, that is, it is the most informative attribute.

- *Support vector machine (SVM)* – SVM seeks a hyperplane which optimally splits training data in an attribute space. To describe this hyperplane only a few nearest points are needed, so called *support vectors*.

SVM allows a non-linear classification so that not linearly separable data are mapped into high-dimensional feature space – *the kernel trick* – in which they can be linearly separated. Also see [Bis06].

- *Neural network* – The neural network is a connection of several linear models, perceptrons. One perceptron can classify only linearly separable data, but according to the Universal approximation theorem a three layered neural network is capable of approximating an arbitrary function. Neural networks are usually trained through the backpropagation learning algorithm. Also see [Bis06].

A model needs to be verified after training. Such evaluation proceeds on a testing set which should be different from the training one. Sizes of these sets are generally 70 % for the training set and 30 % for the testing set.

Based on the testing data a contingency table is filled out, see Tab 2.1.

		predicted values	
		positives	negatives
actual values	positives	true positive (TP)	false negative (FN)
	negatives	false positive (FP)	true negative (TN)

**Table 2.1:** Contingency table for classifier evaluation

Then, the quality of a classifier is rated among others by these criteria:

- *Classification accuracy* tells how many samples were correctly classified.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \quad (2.2)$$

- *Sensitivity* is the probability of a positive sample being indicated as positive.

$$Se = \frac{TP}{TP + FN} \quad (2.3)$$

- *Positive predictivity* is the probability that a sample indicated as positive is truly positive.

$$PP = \frac{TP}{TP + FP} \quad (2.4)$$

These ratios are, however, only point estimates of true population proportions. To tell the degree of uncertainty a (two-sided) confidence interval should be added:

$$\left\langle \hat{p} - u_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + u_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right\rangle \quad (2.5)$$

where  $\hat{p}$  is the sample proportion (point estimate),  $n$  is the sample size,  $u_{1-\frac{\alpha}{2}}$  is the quantile of a t-distribution with  $n - 1$  degrees of freedom and  $\alpha$  is a significance level.

## 2.5 Image Registration

In several steps of the TBM pipeline it is required to align one brain image to an another in order to obtain same structures at similar positions in both images, a process usually called *image registration*. It is an iterative procedure of finding a transformation  $\mathbf{T}$ , which relates the position of features in one image or coordinate space with the position of the corresponding feature in another image or coordinate space. [HH01]

Any registration technique can be described by three components: a transformation which relates the target and source image, a similarity measure which measures the similarity between target and source image, and an optimization which determines the optimal transformation parameters as a function of the similarity measure.

Registration algorithms make use of number of different features. It can be geometrical features such as points and various structures (surface, volumes), or the algorithm can work directly on image intensity values.

### 2.5.1 Point registration

*Point registration* involves identifying corresponding points in the images to be aligned. The most common approach is to find the least square rigid-body or affine transformation that aligns the points. This transformation can subsequently be used to transform any arbitrary point from one image to the another. The Procrustes algorithm is used for determining such transformation (more details in [HH01]). It has a known solution involving computing Singular value decomposition.

This approach is very fast and robust and solves well simple practical problems.

### 2.5.2 Surface matching

*Surface matching* exploits high contrast boundaries or surfaces which can be successfully located by various segmentation algorithms. This is especially true of the skin surface.

One of the most famous algorithm is *the Head and Hat algorithm*. Two equivalent surfaces are identified in the images. The first, from higher resolution modality, is represented as a stack of disks as is referred to as head. The second surface is represented as a list of unconnected 3D points. The registration transformation is determined by iteratively transforming the (rigid) hat surface with respect to the head surface, until the closes fit of the hat on the head is found. The measure of closeness of fit used is the square of distance between a point on the hat and the nearest point on the head, in the direction of the centroid of the head. [HH01]

However, the most frequently applied algorithm for the registration of surfaces is *the Iterative Closest Point (ICP)*. The algorithm works in two stages. Firstly, the closest model point for each data point is identified. Then, a transformation  $\mathbf{T}$  is calculated aiming at the minimization of the evaluation measure  $D$  with least square error. The procedure iteratively continues until a (local) minimum is reached.

### 2.5.3 Registration with intensities

Whereas results and quality of the methods described above depends directly on the segmentation approach, this class of registrations is based on the analysis of intensities, i.e., the similarity measures are based on voxel values.

These similarity measures can either depend on the voxel intensities *directly*, e.g., SSD (Sum of Square Intensity Differences) or CC (Cross Correlation), or they can be *statistically dependent*, e.g., MI (Mutual Information):

- *Sum of Squared Intensity Differences (SSD)* works only for data of identical modalities and it is optimal only if both images differ by Gaussian noise. It is never used for intermodal and rarely for intramodal registration – application predominantly for serial MRI registration. Eq. 2.6 shows how this measure can be computed.

$$SSD = \frac{1}{N} \sum_{\mathbf{x}_A \in \Omega_{A,B}^T} |A(\mathbf{x}_A) - B^T(\mathbf{x}_A)|^2 \quad (2.6)$$

where  $\mathbf{x}_A$  is a voxel location in image  $A$ , within an overlap domain  $\Omega_{A,B}^T$  comprising  $N$  voxels.  $B^T$  denotes image  $B$  transformed into the space of image  $A$ .

- *Cross Correlation (CC)* assumes linear dependencies of the data. It applies for monomodal data such as MRI + MRI. Also see Eq. 2.7.

$$CC = \frac{\sum_{\mathbf{x}_A \in \Omega_{A,B}^T} (A(\mathbf{x}_A) - \bar{A}) \cdot (B^T(\mathbf{x}_A) - \bar{B})}{\sqrt{\sum_{\mathbf{x}_A \in \Omega_{A,B}^T} (A(\mathbf{x}_A) - \bar{A})^2 \cdot \sum_{\mathbf{x}_A \in \Omega_{A,B}^T} (B^T(\mathbf{x}_A) - \bar{B})^2}} \quad (2.7)$$

where  $\bar{A}$  and  $\bar{B}$  are mean values of the intensities within the overlapping area.

- *Mutual Information* enables registration of intermodal data. It is based on information theoretic approach which interprets the voxel intensities of the image  $A$  and image  $B$  as random numbers and describes their statistical dependency. MI describes how well one image is described by another image and can be written in the form of Eq. 2.8.

$$MI = H(A) + H(B) - H(A, B) \quad (2.8)$$

where  $H(A)$  is a marginal entropy of an image  $A$  and  $H(A, B)$  is the joint entropy of images  $A$  and  $B$  which measures the amount of information in the combined images  $A$  and  $B$ .

Generally, the images entering the registration algorithm are referred to as the reference image (or target image, fixed image) and moving image (or source image). The moving image is deformed to match the fixed image.

The registration proceeds as follows:

- Transformation of the moving image.
- Interpolation of the voxel intensities of the moving image at grid positions of the reference image (or vice versa).
- Calculation of the similarity measure.
- Abort if optimum of the similarity measure is reached.
- Adjustment of the transformation – optimization. This is a standard problem and there are standard ways to solve it (e.g. Downhill Simplex Method, Powell’s Method, Steepest Gradient Descent, the Conjugate Gradient Method, etc. [CHH14]). Fluid and elastic transformations that can be described in terms of a partial differential equation (PDE) can be obtained using existing numerical solvers.

#### ■ 2.5.4 Interpolation

Generally, transformed image points of the moving image do not have to fit into the grid of the fixed image. Therefore, it is an absolute necessity to employ some interpolation technique to obtain intensity values at the discrete coordinates in the reference space.

A choice of appropriate interpolator must be carefully considered as many registration algorithms interpolate images every iteration and thus interpolation errors can introduce modulations in the similarity measure.

Interpolation methods can be divided into two groups:

- Local interpolation methods (cell-wise). Such methods use the information stored in the neighbor nodes (vertices). In this category belong low-cost interpolators, such as *nearest neighbor* or *trilinear interpolator*. Using these methods can be very time effective, however, at the expense of quality as the result often suffers from "cusps" in the image.

- Global interpolation methods use information of all vertices or a broader neighborhood, e.g., *cubic B-Splines interpolator*. Although, the complexity of such interpolators is often much higher, the resulting images are much smoother.

### ■ 2.5.5 Transformation models

Image registration strategies can be divided into several categories according to the complexity of the transformation model used.

Rigid and affine transformations are global in nature, and thus, they cannot model local geometric differences between images. They are suitable for registration of datasets of one patient. However, this approach proves itself to be inappropriate in case of registration patient – patient, registration atlas – patient, surgery, soft tissue and moving organs.

A solution addressing this issue is a deformable registration which uses nonlinear (nonrigid) deformation of the grid. This category of registrations includes (in the order of increasing degrees of freedom): polynomial, basis functions, spline functions, elastic, fluid registration, and large deformation models (diffeomorphisms).

The registration procedure usually proceeds in two steps. Firstly, the images are roughly aligned using linear registration, i.e., centers alignment, orientation alignment, accounts for any scale factors. Afterwards, fine alignments are carried out using nonlinear registration. That means matching the internal structure by warping the data.

#### ■ Linear registration

*Rigid registration* is the simplest case allowing only translation and rotation. It can be described very compactly by 3 rotation angles and 3 translations. There are many way of parameterizing this transformation in terms of six parameters. One possible form is:

$$\mathbf{y} = \mathbf{TRx} \quad (2.9)$$

where  $\mathbf{x}$  and  $\mathbf{y}$  are image points in two different spaces in homogeneous coordinates,  $\mathbf{T}$  is a translation matrix:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

where  $t_1, t_2, t_3$  are displacements in the corresponding axes, and  $\mathbf{R}$  is a rotation matrix:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) & 0 \\ 0 & -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

where  $\phi$ ,  $\theta$  and  $\psi$  are rotation angles about  $x$ ,  $y$  and  $z$  axis, called *pitch*, *roll* and *yaw* respectively.

*Affine Registration* adds to the translation and rotation also scaling and shear. It can be described by 12 parameters in a form of matrix multiplication:

$$\mathbf{y} = \mathbf{TRSW}\mathbf{x} = \mathbf{M}\mathbf{x} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} \quad (2.12)$$

where  $\mathbf{T}$  and  $\mathbf{R}$  are translation and rotation matrices, respectively, as described above,  $\mathbf{S}$  is a matrix of scales which represents scaling along orthogonal axis, and can be represented via:

$$\mathbf{S} = \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

where  $s_1$ ,  $s_2$  and  $s_3$  are scale factors along corresponding axes, and  $\mathbf{W}$  is a shear matrix, transforming a cube into a general parallelepiped:

$$\mathbf{W} = \begin{bmatrix} 1 & w_{xy} & w_{xz} & 0 \\ w_{yx} & 1 & w_{yz} & 0 \\ w_{zx} & w_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

where  $w_{ij}$  are shear factors.

### ■ Nonlinear (deformable) registration

*Polynomial registration* is defined by a higher order polynomial. For example, the quadratic transformation is defined by second order polynomials (30 DOF):

$$\mathbf{T}(x, y, z) = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & \dots & a_{08} & a_{09} \\ a_{10} & \dots & a_{18} & a_{19} \\ a_{20} & \dots & a_{28} & a_{29} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ z^2 \\ xy \\ \dots \\ 1 \end{bmatrix} \quad (2.15)$$

However, this type of registration is quite limited since they can model only global shape changes. In addition, higher order polynomials tend to introduce artifacts such as oscillations. Therefore they are rarely used.

Instead of using a linear combination of higher order terms, one can use a linear combination of *basis functions*  $\Theta_i$  to describe the deformation field:

$$\mathbf{T}(x, y, z) = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & \dots & a_{0n} \\ a_{10} & \dots & a_{1n} \\ a_{20} & \dots & a_{2n} \\ 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Theta_0(x, y, z) \\ \dots \\ \Theta_n(x, y, z) \\ 1 \end{bmatrix} \quad (2.16)$$

A common choice is a set of (orthonormal) basis functions such as Fourier basis functions or Wavelet basis functions.

*Registration with splines* is based on assumption that a set of corresponding points or landmarks can be identified in the source and target images (control points). Probably the most used ones are *B-Splines* which have the advantage of being locally controlled functions, so that if control points change location, only local update of cost function is necessary. The deformation is smooth and continuous up to the second derivative.

*Elastic registration technique.* An elastic transformation involves computing a mapping from image  $I(\mathbf{x})$  to image  $J(\mathbf{x})$  through a deformation field  $\mathbf{u}(\mathbf{x})$ . The deformation is defined in the physical space of the image and dictates the positional difference between corresponding features in the two images. Thus, if a feature defined at  $I(\mathbf{x})$  matches a feature in  $J$  at position  $\mathbf{y}$  then the deformation field at  $\mathbf{x}$  should give  $\mathbf{u}(\mathbf{x}) = \mathbf{y} - \mathbf{x}$ . Such a deformation field may be applied to deform image  $J$  into image  $I$  by composing the mapping  $J_{deformed}(x) = J(\mathbf{x} + \mathbf{u}(\mathbf{x}))$ . In a perfect world, then  $I(\mathbf{x}) = J_{deformed}(\mathbf{x})$ , though this is rarely the case. Gradient descent optimization of an elastic mapping may be summarized (crudely) as [ATS09]:

- Compute the similarity gradient:  $\nabla E = \partial_{\mathbf{u}} \Pi(I, J(\mathbf{x} + \mathbf{u}(\mathbf{x})))$ .
- Update the deformation field:  $\mathbf{u}(\mathbf{x}) \leftarrow \mathbf{u}(\mathbf{x}) + \delta \nabla E$
- Regularize the deformation field:  $\mathbf{u}(\mathbf{x}) \leftarrow G_{\sigma} \star \mathbf{u}(\mathbf{x})$

where  $\Pi$  is the similarity,  $\delta$  is a gradient step length and  $G_{\sigma}$  is a gaussian smoother.

The idea is to model the deformation of the source image into the target image as a physical process which resembles the stretching of an elastic material such as rubber. [HH01] The principle is minimization of cost function  $C$ :

$$C(T) = -C_{external}(T) + \lambda C_{internal}(T) \quad (2.17)$$

$\lambda$  is a weighting factor,  $C_{external}$  is an external force which acts on the elastic body,  $C_{internal}$  is an internal force caused by the deformation of an elastic material, i.e., stress. As a consequence, the deformation of the elastic body stops if both forces acting on it form an equilibrium solution. The behaviour of the elastic body is described by the Navier linear elastic partial differential equation:

$$\mu \nabla^2 \mathbf{u}(x, y, z) + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}(x, y, z)) + \mathbf{f}(x, y, z) = 0 \quad (2.18)$$

where  $\mathbf{u}$  is a displacement field,  $\mu$  and  $\lambda$  are Lamé's elasticity constants,  $\mathbf{f}$  is an external force acting on the elastic body which drives the registration process. A common choice for the the external force is the gradient of a similarity measure such as a local correlation measure based on intensities, intensity differences, or intensity features such as edge and curvature. Limitations of this approach is the impossibility to model highly localized deformations since the deformation energy caused by stress increases proportionally with the strength of deformation. [HH01]



In *fluid registration* these constraints are relaxed over time, which enables the modeling of highly localized deformations including corners. The deformations of the fluid deformation are characterized by the Navier-Stokes partial differential equation:

$$\mu \nabla^2 \mathbf{v}(x, y, z) + (\lambda + \mu) \nabla(\nabla \cdot \mathbf{v}(x, y, z)) + \mathbf{f}(x, y, z) = 0 \quad (2.19)$$

similar to Eq. 2.18 except that the differentiation is carried out on the velocity field  $\mathbf{v}$  rather than on the displacement field  $\mathbf{u}$  and is solved for each time step. The relation between the velocity and displacement field is given by:

$$\mathbf{v}(x, y, z, t) = \frac{\partial \mathbf{u}(x, y, z, t)}{\partial t} + \mathbf{v}(x, y, z, t) \cdot \nabla \mathbf{u}(x, y, z, t) \quad (2.20)$$

On the one hand, the fluid registration is a powerful tool which enables to trace detailed changes. On the other hand, it is computationally expensive.

*Diffeomorphic registration* is based on large deformation diffeomorphic metric mapping (LDDMM) algorithms. A diffeomorphism is a differentiable mapping with a differentiable inverse. Modeling transformations with diffeomorphisms ensures certain desirable topological properties that cannot be guaranteed with other methods. [ATS09]

The map  $\phi$ , over time, parameterizes a family of diffeomorphisms,  $\phi(\mathbf{x}, t) : \Omega \times t \rightarrow \Omega$ , which can be generated by integrating a (potentially) time-dependent, smooth velocity field,  $\mathbf{v} : \Omega \times t \rightarrow \mathbb{R}^d$ , through the ordinary differential equation:

$$\frac{d\phi(\mathbf{x}, t)}{dt} = \mathbf{v}(\phi(\mathbf{x}, t), t), \quad \phi(\mathbf{x}, 0) = \mathbf{x} \quad (2.21)$$

where  $\mathbf{x}$  is an image coordinate (position),  $t$  is time and  $\Omega$  is an image domain. The deformation field provided by  $\phi$  is  $\mathbf{u}(\mathbf{x}) = \phi(\mathbf{x}, 1) - \mathbf{x}$ .

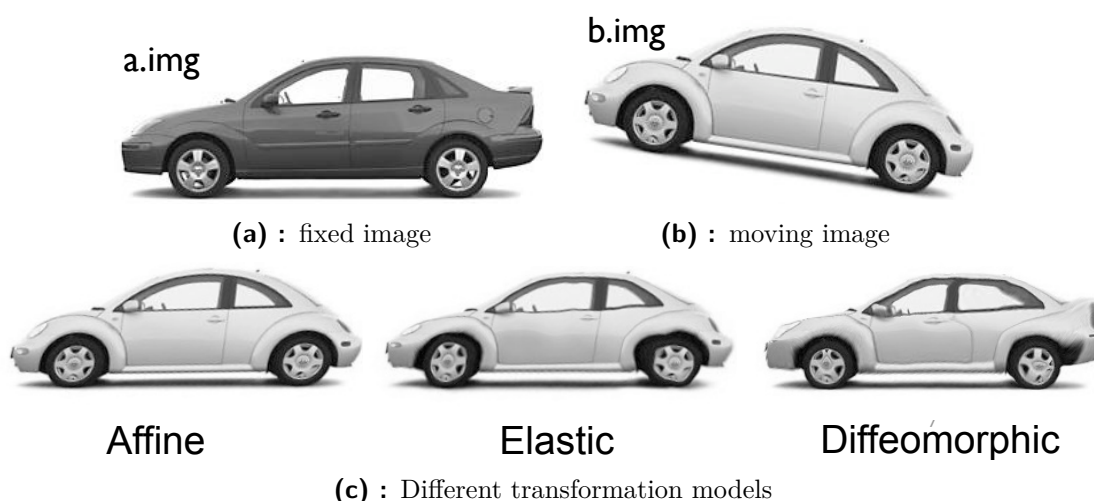
For mapping image  $I$  to  $J$ , by the diffeomorphic mapping,  $\phi$ , the following functional is minimized:

$$\mathbf{v}^* = \arg \min_{\mathbf{v}} \left\{ \int_0^1 \|L\mathbf{v}\|^2 dt + \lambda \int_{\Omega} \Pi(I, \phi(\mathbf{x}, t), J) d\Omega \right\} \quad (2.22)$$

Rough explanation: The first term  $\int_0^1 \|L\mathbf{v}\|^2 dt$  can be seen as a smoothness term, where  $L$  is a smoothing operator. In that case, a smooth velocity field is minimized. The second term  $\lambda \int_{\Omega} \Pi(I, \phi(\mathbf{x}, t), J) d\Omega$  is a data term which compares the similarity between two images.  $\lambda$  controls exactness in matching,  $\Pi$  is a similarity metric, e.g., sum of squared differences (SSD), cross-correlation (CC), or mutual information (MI).

There are number of of LDDMM formulations, e.g., SyN (geodesic SyN, greedy SyN, BSpline SyN) and DARTEL. Especially, SyN is a popular and top performing algorithm. According to [KAA<sup>+</sup>09], SyN shows out an outstanding performance in comparison to a variety of other nonlinear registration techniques.

Fig. 2.6 compares affine, elastic and diffeomorphic transformation models on an example of deforming the beetle (moving image) to the ford (fixed image) under these models. Left to right increases the degrees of freedom in the mapping and thus the registration accuracy.



**Figure 2.6:** Comparison of different transformation models [ATS09]

## 2.6 Analysis tools for MRI

This chapter concerns with tools and software bundles required for tensor-based morphometry methods (TBM) and gives a comprehensive overview of their functions and utilization. Most of them were used for an implementation and evaluation of the TBM pipeline.

### 2.6.1 MRI data formats

There are various file formats to store MR image data. DICOM and NIfTI are, however, probably the most widespread ones.

*DICOM.* Nearly all neuroimaging data are produced in Digital Imaging and Communications in Medicine (DICOM) standard which was introduced by National Electrical Manufacturers Association (NEMA) and is in detail described in NEMA standard PS3.12<sup>1</sup>. DICOM files contain not only image data but also embedded tags. On one hand, these tags describe the image data (bit depth, image size, ...), but on the other, they also carry information about patient, type of examination, etc. One DICOM file usually contains only one image, but it can also contains a whole series of images. Files in DICOM format are distinguishable by the .dcm extension.

*NIfTI* format was created as a replacement for much older Analyze format. It originated from the Data Format Working Group (DFWG) in the Neuroimaging Informatics Technology Initiative (NIfTI). One of the main advantages of this format is that it addresses problem of spatial orientation as each file contains affine transformation to specify voxel coordinates. [Gro] Images in NIfTI format are stored either in a single gzipped file (extension .nii.gz) or separately in .img or .hdr files. Vast majority of data analysis software bundles, such as SPM, FSL, AFNI, FreeSurfer or ANTs, support

<sup>1</sup><http://dicom.nema.org/medical/dicom/current/output/chtml/part12/PS3.12.html>

images only in this format. Some of the scanners support direct export to NIfTI. If it is not the case there are several tools to convert DICOM and other formats to NIfTI, e.g., *dcm2nii* tool (see below).

## ■ 2.6.2 MRICron

MRICron is a multi-platform NIfTI format image viewer.<sup>2</sup> It allows to display single slices, render volumes, show regions of interests, show color overlays and more.

Beside the viewer, this package contains another utilities: *dcm2nii* and *npm*.

### ■ *dcm2nii*

*Dcm2nii* is a tool for converting DICOM files to NIfTI. It is a command line tools as well as a GUI application.

Command line tools is pretty straightforward to use:

```
dcm2nii <options> <sourcenames>
```

There are plenty of options available. Most of them take for argument **y** (yes) or **n** (no) The most important are the following ones:

- **-a** anonymize (default **y**)
- **-d** date in filename (default **y**)
- **-e** events (series/acq) in filename (default **y**)
- **-f** source filename (default **n**)
- **-g** gzip output, filename.nii.gz (default **y**)
- **-i** ID in filename (default **n**)
- **-r** reorient image to nearest orthogonal
- **-x** reorient and crop 3D NIfTI images (default **n**)

It is often the case that each image slice is found in a separate DICOM file. The following command can be used to convert every DICOM file in a directory to a single NIfTI file:

```
dcm2nii *.dcm
```

which produces 3 files:

- 20130519\_113546t1SAGmprp2isos007a1001.nii.gz – gzipped NIfTI file with date and events in filename
- o20130519\_113546t1SAGmprp2isos007a1001.nii.gz – same as above and reoriented

---

<sup>2</sup><http://people.cas.sc.edu/rorden/mricron/index.html>

- `co20130519_113546t1SAGmprp2isos007a1001.nii.gz` – same as above and cropped

or

```
dcm2nii -a y -d n -e n -f y -g y -i n -r n -x n *.dcm
```

which produces only one file: `0_S006_I0000t1SAGmprp2iso.nii.gz` – gzipped NIFTI file with source filename and modality specifications in filename.

## ■ **npm**

NPM (Non-Parametric Mapping) is a GUI application which comes in the MRICron package. It provides a large variety of statistic tests for comparing and evaluating MR images.

## ■ **2.6.3 FSL**

FSL (FMRIB Software Library) is an extensive library of analysis tools for functional, structural and diffusion MRI data. Most of the tools can be run either from command line or as a GUI application.

For purposes of this project, the following tools will be used and analyses more in detail: BET – brain extraction, FAST – segmentation tool, FLIRT – linear registration, FNIRT – non-linear registration, FSLView – MR image viewer.

## ■ **BET**

BET is an automated method for segmenting magnetic resonance head images into brain and nonbrain. It is very robust and accurate and has been tested on thousands of data sets from a wide variety of scanners and taken with a wide variety of MRI sequences. The method, Brain Extraction Tool (BET), uses a deformable model that evolves to fit the brain's surface by the application of a set of locally adaptive model forces. The method is very fast and requires no preregistration or other pre-processing before being applied. [Smi02]

BET in command line is easy to use: `bet <input> <output> [options]`, where input and output file can be either in DICOM or NIFTI format. Although, the whole process is highly automated, the program offers few options (only selection of the most important ones):

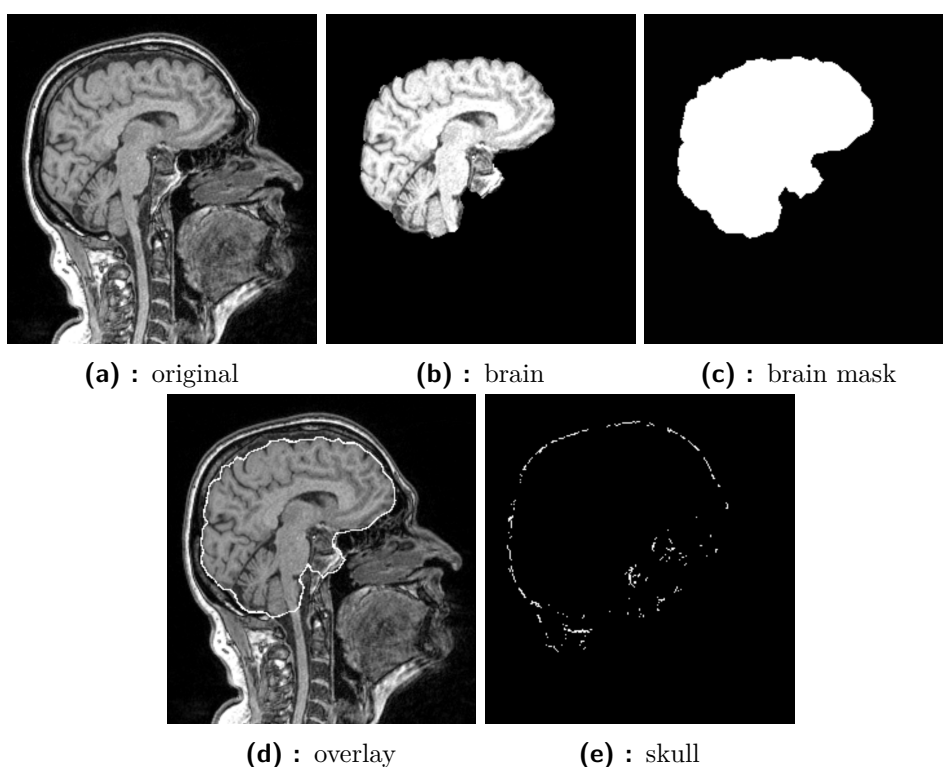
- `-o` generate brain surface outline overlaid onto original image
- `-m` generate binary brain mask
- `-s` generate approximate skull image
- `-f <f>` fractional intensity threshold (0->1); default=0.5; smaller values give larger brain outline estimates
- `-R` robust brain centre estimation (iterates BET several times)

For example, a command

```
bet b0.nii.gz brain\_ext -f 0.5 -R -o -m -s
```

outputs 4 files (also see Fig. 2.7):

- `brain_ext.nii.gz` – extracted brain
- `brain_ext_mask.nii.gz` – binary brain mask
- `brain_ext_overlay.nii.gz` – brain surface outline overlaid onto original image
- `brain_ext_skull.nii.gz` – approximate skull image



**Figure 2.7:** Example of input and outputs of BET

## ■ FAST

FAST<sup>3</sup> (FMRIB's Automated Segmentation Tool) segments a 3D image of the brain into different tissue types (Grey Matter, White Matter, CSF, etc.), whilst also correcting for spatial intensity variations (also known as bias field or RF inhomogeneities). The underlying method is based on a hidden Markov random field model and an associated Expectation-Maximization algorithm. The whole process is fully automated and can also produce a bias

<sup>3</sup><http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FAST>

field-corrected input image and a probabilistic and/or partial volume tissue segmentation. It is robust and reliable, compared to most finite mixture model-based methods, which are sensitive to noise. [ZBS01]

FAST is both a GUI application and a command-line program. We will investigate how the command-line version works. The basic structure is

```
fast [options] file(s),
```

where `file(s)` is an image to be segmented. However, it has to be a brain-only image.

Again, several options are available:

- `-t <n>` or `-type=<n>` type of image (n=1 for T1, n=2 for T2, n=3 for PD)
- `-o <base>` or `-out=<base>` basename for outputs. Output images will have filenames derived from this basename. For example, the main output, the binary segmentation: `<basename>_seg.nii.gz`.
- `-n <n>` or `-class=<n>` number of tissue-type classes. Default values is `-n 3` meaning Grey Matter, White Matter and CSF (cerebrospinal fluid).

For example, a command

```
fast -o brain b0.nii.gz
```

outputs:

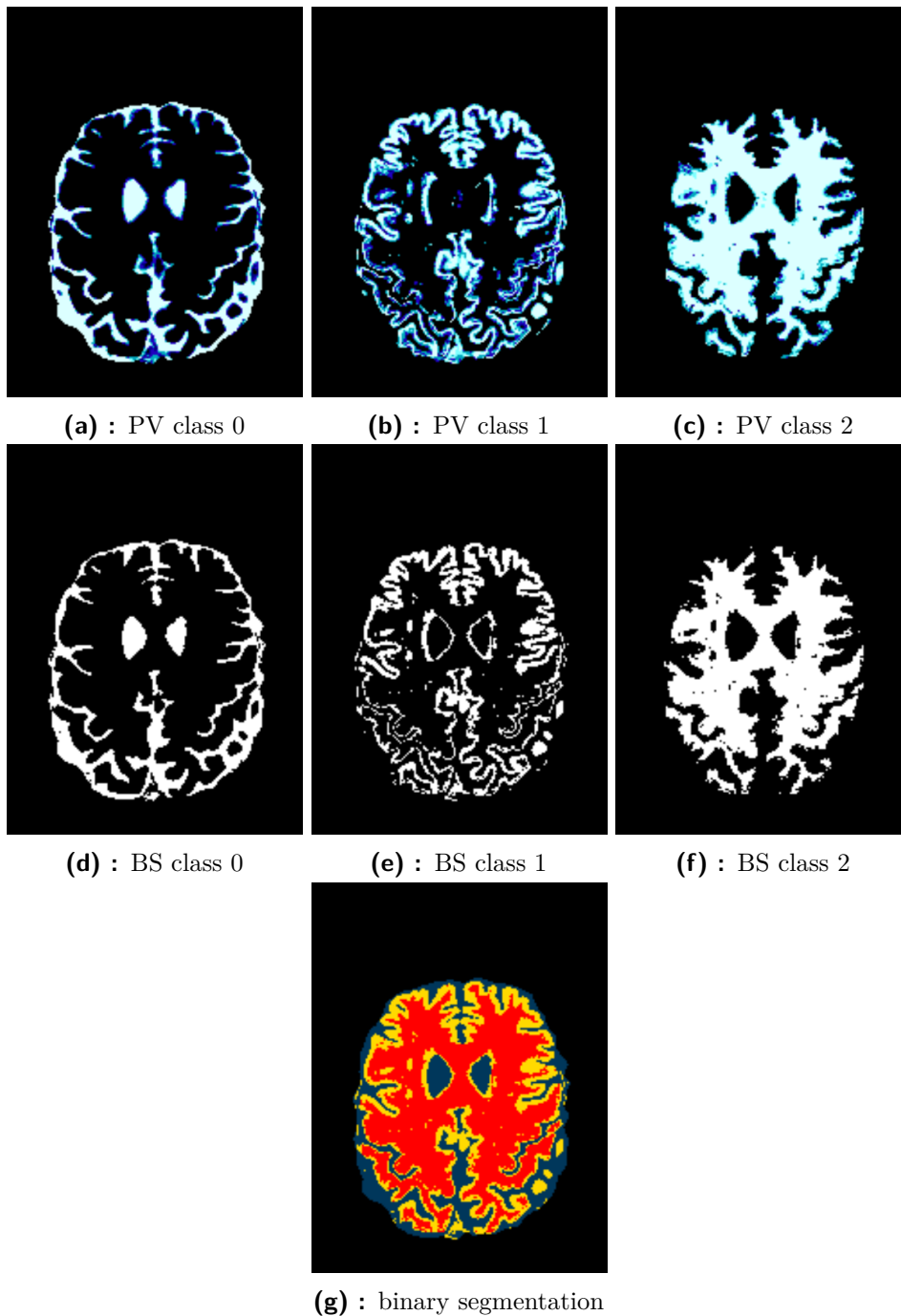
- `brain_pve_[0-2].nii.gz` – *Partial volume maps*: A (non-binary) partial volume image for each class, where each voxel contains a value in the range 0-1 that represents the proportion of that class's tissue present in that voxel. This is the default output. See Fig. 2.8a, 2.8b and 2.8c.
- `brain_seg.nii.gz` – *Binary segmentation (single image)*: This is the "hard" (binary) segmentation, where each voxel is classified into one class only. A single image contains all the necessary information, with the first class taking intensity value 1 in the image, etc. See Fig. 2.8g.
- `brain_seg_[0-2].nii.gz` – *Binary segmentation (one image per class)*: This is also a hard segmentation output; the difference is that there is one output image per class, and values are only either 0 or 1. See Fig. 2.8d, 2.8e and 2.8f.

## ■ FLIRT

FLIRT<sup>4</sup> (FMRIB's Linear Image Registration Tool) is a fully automated robust and accurate tool for linear (affine) intra- and inter-modal brain image registration.

As well as many tools from FSL package, FLIRT offers GUI application and command-line program. We will focus mainly on the later.

<sup>4</sup><http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT>



**Figure 2.8:** Example outputs of FAST (partial volumes (PV) and binary segmentation (BS))

FLIRT requires quite a large number of parameters. We will list only the most important ones:

```
flirt [options] -in <inputvol> -ref <refvol> -out <outputvol> -omat
<outputmatrix>
```

- `-in <inputvol>` input image to be registered
- `-ref <refvol>` reference image (also called target). Manual recommends to use MNI152 template images (whole head, extracted brain, brain mask, skull) located in `/usr/share/fsl/5.0/data/standard` as FLIRT is very well tested with them.
- `-out, -o <outputvol>` output volume
- `-omat <matrix-filename>` output file with transformation matrix
- `-dof <number of transform dofs>` specify number of degrees of freedom. For 3D to 3D mode (default) the DOF can be set to 12 (affine), 9 (traditional), 7 (global rescale) or 6 (rigid body). Default value is 12.
- `-searchrx <min_angle> <max_angle>`, `-searchry <min_angle> <max_angle>`, `-searchrz <min_angle> <max_angle>` set minimum and maximum search angles in x-, y- or z- axis. Default is -90 90 for each axis. If the images are not aligned but have the same orientation, it is a good practise to set -90 90 to each axis. However, if the images to be registered are possibly incorrectly oriented, it is recommended to set this option to -180 180.
- `-cost {mutualinfo,corratio,normcorr,normmi,leastsq,labeldiff,bb}` This includes the within-modality functions Least Squares and Normalised Correlation, as well as the between-modality functions Correlation Ratio (the default), Mutual Information and Normalised Mutual Information.
- `-interp {trilinear,nearestneighbour,sinc,spline}`,
- `-sincwidth <full-width in voxels>` (default is 7),
- `-sincwindow rectangular,hanning,blackman` select the interpolation method to be used in the final (reslice) transformation (it is not used for the estimation stage - trilinear interpolation is always used for the estimation of the transformation). The options for this final interpolation method are: Tri-Linear; Nearest Neighbour and Sinc. If Sinc is chosen, further window parameters (type of windowing function and window width) can also be specified.

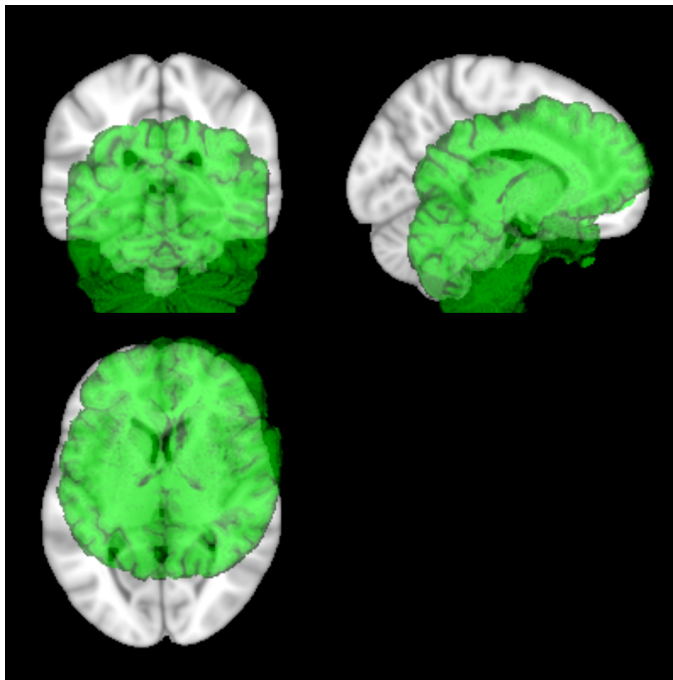
For example, a command

```
flirt -in brain.nii.gz -ref MNI152_T1_1mm_brain \
      -out brain_registered -omat brain_registered
```

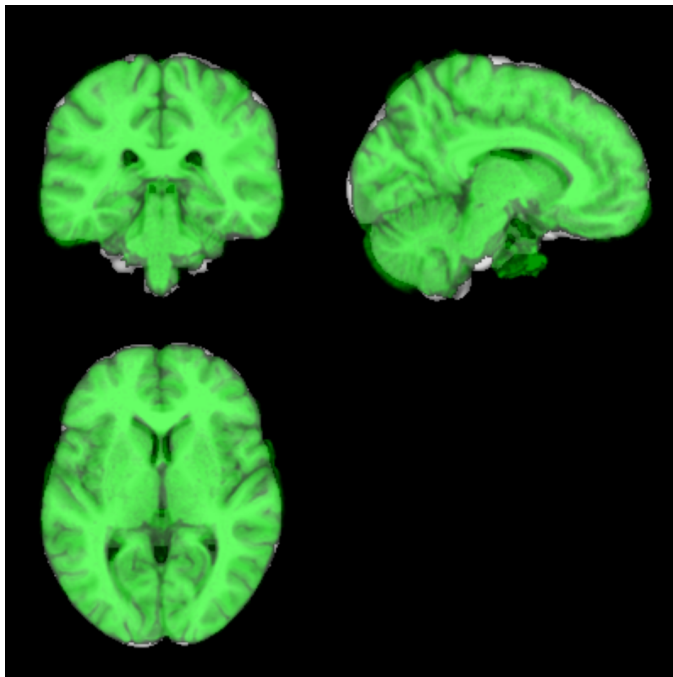
produces 2 files:

- `brain_registered.nii.gz` – a registered file. Figure 2.9a shows an original volume and reference in one picture. In Figure 2.9b, the original volume is replaced by a registered file.
- `brain_registered.mat` file contains transformation matrix.





(a) : before registration



(b) : after registration

**Figure 2.9:** Comparison of unregistered and FLIRT registered volumes (moving image – green, fixed image – grey)

## ■ SIENA

SIENA<sup>5</sup> is a package for both single-time-point (cross-sectional) and two-time-point (longitudinal) analysis of brain change, in particular, the estimation of atrophy (volumetric loss of brain tissue). SIENA has been used in many clinical studies. SIENA estimates percentage brain volume change (PBVC) between two input images, taken of the same subject, at different points in time. It calls a series of FSL programs to strip the non-brain tissue from the two images, register the two brains (under the constraint that the skulls are used to hold the scaling constant during the registration) and analyse the brain change between the two time points. It is also possible to project the voxelwise atrophy measures into standard space in a way that allows for multi-subject voxelwise statistical testing.

## ■ 2.6.4 ANTs

The ANTs framework<sup>6</sup> provides open-source functionality for deformable normalization with large deformations. Small deformation mappings and segmentation tools are also available. ANTs serves as both a base for further algorithm development and also as an application-oriented toolkit. ANTs enables diffeomorphic normalization with a variety of transformation models, optimal template construction, multiple types of diffeomorphisms, multivariate similarity metrics, diffusion tensor processing and warping, image segmentation with and without priors and measurement of cortical thickness from probabilistic segmentation. The normalization tools, alone, provide a near limitless range of functionality and allow the user to develop customized objective functions. [ATS09]

ANTs contains separate programs (e.g. `antsApplyTransform`, `antsRegistration`) as well as user-friendly wrappings of ANTs tools that enable higher-level error checking and combinations of basic ANTs functions (e.g. `antsRegistrationSyn.sh`, `antsMultivariateTemplateConstruction2.sh`).

Following sections will go deeper into usage of the most important ANTs' programs and scripts:

## ■ `antsRegistration`

This is the core program of ANTs which serves for registration of images.

The ANTs toolkit provides a hierarchy of transformations with adjustable levels of complexity, regularization, degrees of freedom and behavior as optimizers. The simplest transformation model is the translation, followed by the rigid and/or affine transform. The most complex – and most flexible – is a symmetric diffeomorphic transformation based on optimizing and integrating a time-varying velocity field (for more about diffeomorphic transformation see Sec. 2.5.5). Computation time also increases with transformation model

---

<sup>5</sup><http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/SIENA>

<sup>6</sup><http://stnava.github.io/ANTs/>

complexity. An overview of available models and similarity terms is in Tab. 2.2.

Similarity metric acronyms: CC = neighborhood cross correlation, MeanSquares = mean squared difference, MI = mutual information.

Category	Transformation	Similarity measures	Brief description
Linear	Rigid	MI, MeanSquares, GC	Rigid registration
	Similarity	MI, MeanSquares, GC	Rotation + uniform scaling
	Affine	MI, MeanSquares, GC	Affine registration
Elastic	GaussianDisplacementField	CC, MI, MeanSquares, Demons	Demons-like algorithm
	BSplineDisplacementField	CC, MI, MeanSquares, Demons	FFD variant
Diffeo.	Exponential	CC, MI, MeanSquares, Demons	$\min \mathbf{v}(\mathbf{x})$
	SyN	CC, MI, MeanSquares, Demons	locally in time $\min \mathbf{v}(\mathbf{x}, t)$
	BSplineSyN	CC, MI, MeanSquares, Demons	locally in time $\min \mathbf{v}(\mathbf{x}, t)$
	TimeVaryingVelocityField	CC, MI, MeanSquares, Demons	$\min \mathbf{v}(\mathbf{x}, t)$ over all time

**Table 2.2:** Transformations and a subset of the similarity metrics available in ANTs [ATS09]

ANTs supports both volumetric registration and point set registration. The image/point set similarity metrics in ANTs are unified in the form of a function on the images or the point sets:

`Similarity[fixedImage,movingImage,weight,samplingStrategy,parameters]`

The similarity type for the transformation is specified by `-m` option, which contains two parts: similarity type and parameters inside the brackets. The possible similarity metrics for volumetric images are:

- Cross correlation estimate: `-m CC[fixedImage,movingImage,weight,radius]`. This metric works well for intra-modality image registration. For example, `-m CC[fixed.nii,moving.nii,1,5]` specifies:
  - the fixed image: `fixed.nii`
  - the moving image: `moving.nii`
  - weight for this metric is 1 (i.e., only this metric drives the registration)
  - the region radius for computing cross correlation is 5
- Mutual information: `-m MI[fixedImage,movingImage,weight,number-of-histogram-bins,samplingStrategy,samplingPercentage]`. This metric works both well for intra-modality and inter-modality image registration. For example, the first three parameters in `-m MI[fixed.nii,moving.nii,1,32]` similar to the example above in cross correlation, except that the last parameter means that the number of bins in computing mutual information is 32.
- Global correlation: `-m GC[fixedImage,movingImage,weight,0,samplingStrategy,samplingPercentage]` This metric works both well for intra-modality and inter-modality image registration. For example, `-m GC[fixed.nii,moving.nii,1,0,Random,0.1]` uses 10% random sampling of the image to estimate the global correlation.

- Mean square difference: `-m MeanSquares[fixedImage,movingImage,weight,0]`  
This metric works for intra-modality image registration. The last parameter 0 is a padding value of no real meaning. For example, `-m MeanSquares[fixed.nii,moving.nii,1,0]`.
- Point set expectation `-m PSE` and Jensen-Tsallis BSpline `-m JTB` – similarity metrics for point sets. These will not be used in this project.

`antsRegistration` command line program takes the following options (the list is not exhaustive):

- `-d` – dimensionality (2 or 3)
- `-o` Specify the output transform prefix (output format is `.nii.gz`). Optionally, one can choose to warp the moving image to the fixed space and, if the inverse transform exists, one can also output the warped fixed image.
- `-m metric` – similarity metric. See Tab. 2.2.
- `-c MxNx0` or `-c [MxNx0,<convergenceThreshold=1e-6>,<convergenceWindowSize=10>]`  
Convergence is determined from the number of iterations per level and is determined by fitting a line to the normalized energy profile of the last N iterations (where N is specified by the window size) and determining the slope which is then compared with the convergence threshold.
- `-s MxNx0` Specify the sigma of gaussian smoothing at each level. Units are given in terms of voxels ('vox') or physical spacing ('mm'). Example usage is '4x2x1mm' and '4x2x1vox' where no units implies voxel spacing.
- `-f MxNx0` Specify the shrink factor (downsampling rate) for the virtual domain (typically the fixed image) at each level.

*Usage examples.* ANTs registration options include control of iterations (and, optionally, convergence criterion) via `-c 5000x5000x5000` which specifies that the registration uses a 3 level image pyramid with each level 5000 iterations at most. Multi-resolution options include `-s` for smoothing and `-f` for "shrink factors" i.e. downsampling rates (e.g. 8 means 1/8th resolution).

`MI[fixed,moving,1,32,Regular,0.1]` means to use Mutual Information as similarity metric with 32 bins and regularly spaced samples of 10% of the image; lower sampling rates increases speed and is useful in low-dimensional registration.

Another example of `antsRegistration` may be:

```
antsRegistration
-d 2
-r [ r64slice.nii.gz , r16slice.nii.gz ,1]
-m mattes[ r64slice.nii.gz , r16slice.nii.gz, 1 , \
          32, regular,0.1]
```

```

-t affine[ 0.1 ]
-c [500x500x50,1.e-8,20]
-s 4x2x1vox
-f 3x2x1 -l 1
-m mattes[ r64slice.nii.gz , r16slice.nii.gz, 1 , 48 ]
-t syn[ .15, 3, 0.5 ]
-c [ 50x50x50,0,5 ]
-s 1x0.5x0vox
-f 4x2x1 -l 1 -u 1 -z 1
--output [out,outWarped.nii.gz,outInverseWarped.nii.gz]

```

This registration aligns `r16slice.nii.gz` to `r64slice.nii.gz` (2D images). First, an initial translation transform using the image intensities is done. Then comes affine transformation with gradient step 0.1 using mutual information as a transformation metric (relative weight 1, 32 bins, regular sampling strategy of the 10% of the image). Algorithm is going to work on 3 levels with 500, 500 and 50 iterations max.

After the affine pre-registration, SyN registration is conducted. Standard recommended parameters are `-t syn[ .15, 3, 0.5 ]`.

## ■ **antsApplyTransforms**

Once affine matrix or non-linear displacement field are generated, they can be used to transform images. ANTs utility for this purpose is called `antsApplyTransforms`.

On this place, it should be reminded how the transform application actually works. Suppose that an affine transformation matrix has already been derived. A naive approach would be to take all image points from the moving image and transform their coordinates. This is, however, a very flawed way to do this, because, firstly, two different pixels in the original image may be mapped to the same pixel in the warped image. Secondly, some pixels in the resulting warped image may be left without any pixel value assigned.

Therefore it is preferable to employ *backward warping*. That is to get each pixel in the resulting image from its corresponding location (by the means of inverse transformation) in the first image and to employ some interpolation technique. This requires an inverse of a 4x4 affine transformation matrix which is obviously no problem. For the same reason, the exact same transformation matrix can be used for warping images from the space of the fixed image to the space of the moving image.

The final result of a deformable registration is a *deformation field*. The principle is very similar to the one above. The forward deformation field tells where to find each pixel from the resulting image in the moving image. However, these fields are hardly invertible therefore if one wants to do a backward warping, a separate backward deformation field is required. Fortunately, ANTs offers such option. As the output of a registration process, ANTs usually returns 5 files:

- `output0GenericAffine.mat` – Matlab-like file with initial affine transform matrix. In case of 3D data, this will be 4x4 matrix.
- `outputWarp.nii.gz` – displacement field of non linear transformation (mapping moving image to the fixed image). This file contains a vector with three components for each point in target space pointing to the corresponding point in the source space.
- `outputInverseWarp.nii.gz` – displacement field of non linear transformation (mapping fixed image to the moving image). This file contains a vector with with three components for rach point in source space pointing to the corresponding point in the target space.
- `outputWarped.nii.gz` – the result; moving image warped to the space of fixed image.
- `outputInverseWarped.nii.gz` – fixed image warped to the space of moving image.

## ■ ImageMath

Most basic scalar image operations – and some tensor operations – may be performed with this program. Some operations output text files, some output images and some output only to the terminal. ImageMath allows one to multiply images together (`m`), to negate images (`Neg`), to take an image to a power (`pow`), to test the invertibility of transformations (`InvId`), and to compute the gradient or laplacian of an image (`Gradient`, `Laplacian`). Many other operations are available.

## ■ N4BiasFieldCorrection

This program serves for removal of the bias field in MR images as described in Sec. 2.4.1. A state-of-the-art  $N_4$  algorithm is used.

$N_4$  is a variant of the popular  $N_3$  (nonparameteric nonuniform normalization) retrospective bias correction algorithm. Based on the assumption that the corruption of the low frequency bias field can be modeled as a convolution of the intensity histogram by a Gaussian, the basic algorithmic protocol is to iterate between deconvolving the intensity histogram by a Gaussian, remapping the intensities, and then spatially smoothing this result by a B-spline modeling of the bias field itself. The modifications from and improvements obtained over the original  $N_3$  algorithm are described in [TAC<sup>+</sup>10].

`N4BiasFieldCorrection` takes the following arguments (the list is not exhaustive):

- `-d` – image-dimensionality
- `-i` – input image filename

- `-s` – shrink factor. Running N4 on large images can be time consuming. To lessen computation time, the input image can be resampled. The shrink factor, specified as a single integer, describes this resampling. Shrink factors  $\leq 4$  are commonly used.
- `-c` [`<numberOfIterations=50x50x50x50>`,`<convergenceThreshold=0.0>`] – convergence. Convergence is determined by calculating the coefficient of variation between subsequent iterations. When this value is less than the specified threshold from the previous iteration or the maximum number of iterations is exceeded the program terminates. Multiple resolutions can be specified by using 'x' between the number of iterations at each resolution, e.g. 100x50x50.
- `-o` [`<correctedImage,<biasField>`] – output (corrected image). The output consists of the bias corrected version of the input image. Optionally, one can also output the estimated bias field.

For example, a call of this utility can look similar to this:

```
N4BiasFieldCorrection
-d 3
-i inputImage.nii.gz
-s 2
-c [100x100x100x100,0.0000000001]
-o [correctedImage.nii.gz, biasField.nii.gz]
```

### ■ `antsRegistrationSyNQuick.sh`

`antsRegistrationSyNQuick.sh` is a wrapper enabling quick evaluation of SyN algorithm. It is suitable for quick testing purposes rather than serious data analysis since this wrapper uses some default parameter values. Of course, they can be altered because this wrapper (as well as all other ANTs' wrappers) is written in bash scripting language and therefore can be changed easily.

### ■ `antsMultivariateTemplateConstruction2.sh`

A useful script for setting up optimal template construction. This utility will be very useful when creating the average group template as it can do all the work if the parameters are set correctly.

As this is the key utility in the thesis, one might look properly into the syntax and get the grasp of its functioning:

- `-d` – image dimension (2 or 3)
- `-o` – output prefix
- `-c` – parallel computation

- **-g** – gradient step size (default 0.25): smaller in magnitude results in more cautious steps
- **-i** – iteration limit (default 4)
- **-j** – number cpu cores to use locally
- **-k** – number of modalities used to construct the template (default 1)
- **-w** – modality weights used in the similarity metric (default 1)
- **-q** – Max iterations for each pairwise registration (default = 100x100x70x20): specified in the form ...xJxKxL where
  - J = max iterations at coarsest resolution (here, reduced by power of 2<sup>2</sup>)
  - K = middle resolution iterations (here, reduced by power of 2)
  - L = fine resolution iterations (here, full resolution). Finer resolutions take much more time per iteration than coarser resolutions.
- **-f** – shrink factors (default = 6x4x2x1)
- **-s** – smoothing factors (default = 3x2x1x0)
- **-n** – N4BiasFieldCorrection of moving image: 0 == off, 1 == on (default 1).
- **-r** – do rigid-body registration of inputs before creating template (default 0): Only useful when you do not have an initial template.
- **-m** – type of similarity metric used for registration (default = CC): Options are
  - CC = cross-correlation
  - MI = mutual information
  - MSQ = mean square difference
  - DEMONS = demon's metric

A similarity metric per modality can be specified. If the CC metric is chosen, one can also specify the radius in brackets, e.g., **-m CC[4]**.

- **-t** – type of transformation model used for registration (default = SyN): Options are
  - SyN = Greedy SyN
  - BSplineSyN = Greedy B-spline SyN
  - TimeVaryingVelocityField = Time-varying velocity field
  - TimeVaryingBSplineVelocityField = Time-varying B-spline velocity field



- `-z` – use this volume as the target of all inputs. When not used, the script will create an unbiased starting point by averaging all inputs. Use the full path.

A constructing of a template will be presented on the example found at <http://ntustison.github.io/TemplateBuildingExample/>. Instead of brains, male faces are used. We found this very illustrative as, first, this kind of data are inherently two-dimensional and therefore well presentable and, secondly, a reader will probably better recognise subtle changes in faces than in brain images.

The whole process of creating the template is demonstrated in Fig. 2.10. Ten male face images are entering the algorithm. In this case, the call of `antsMultivariateTemplateConstruction2.sh` is as follows:

```
antsMultivariateTemplateConstruction2.sh
-d 2
-o ${PWD}/TemplateFaces/T_
-i
-b 1
-g 0.15
-j 8
-c 2
-k 1
-w 1
-f 16x12x8x4x2x1
-s 4x4x4x2x1x0
-q 100x100x100x70x50x10
-n 0
-r 0
-m CC
-t BSplineSyN[0.1,75,0]
${PWD}/face*.nii.gz
```

That means, the function takes all files starting with `face` prefix and with `.nii.gz` extension as the input. First, an unbiased starting point is created by averaging all inputs after rigid-body registration.

Then, all images are warped to this primary template using `BSplineSyN` method and `CC` (cross-correlation) metric. Every registration proceeds in a multiresolution scheme: at first, only 1/16th resolution (with smoothing sigma of 4 voxels and a maximum of 100 iterations), then 1/12th resolution (with smoothing sigma of 4 voxels and a maximum of 100 iterations), and so on. Afterwards, all these warped images are voxel-wise averaged to create a (temporary) template.

In the next step, we find the average warp to the template and apply its inverse to the template image (i.e., the inverse geometric centering of the displacement fields). This keeps the template shape stable over multiple iterations of template building.

The algorithm continues through several iterations (in this case 4 iterations). However, the original images are no longer warped to the initial affine template but to the inverse centered average (temporary template) from the previous iteration.

The resulting image is the *average group template*, described in Chap. 2.4.2, which can be, for example, used in the cross-sectional tensor-based morphometry study as it shares common features of a population.

**Remark:** As this procedure basically wraps many iterations of registration processes, it usually takes quite a long time to build such a template. (Tens of minutes in case of 2D slices to days in case of 3D volumes!)

### ■ 2.6.5 ANTsR

After processing image data, one often needs to extract some attributes and compute statistic measures based on it. For this purpose there is *ANTsR*<sup>7</sup>, a package providing ANTs features in R (an open-source, cutting-edge statistical language and environment) as well as imaging-specific data representations, spatially regularized dimensionality reduction and segmentation tools.

There is a great advantage in this approach since both image processing and statistics evaluation can be done under one environment. *ANTsR* provides user with the *antsImage* class which contains all information about an image object (dimensions, bit depth, voxel data, etc.). Its scope is not limited to the *ANTsR* package though. Any other R package can also work with this structure and thus a large variety of statistical analysis can be performed.

Quite extensive reference manual can be found at <https://github.com/stnava/ANTsR/releases/download/latest/ANTsR-manual.pdf>. For the purpose of understanding of the code listings in the next chapter, a few key ANTsR commands is listed below:

- `antsImageRead(filename, dimension = NULL, pixeltype = "float")` – Read an image file into an object of class `antsImage`.
- `antsImageWrite(image, filename)` – Write an image object of class `antsImage` to a file.
- `abpN4(img)` – MR image bias correction based on the N4 algorithm.
- `resampleImage(image, resampleParams, useVoxels = 0, interpType = 1)` – Resample image by spacing or number of voxels with various interpolators.
  - `resampleParams` – vector of size dimension with numeric values
  - `useVoxels` – true means interpret resample params as voxel counts
  - `interpType` – one of 0 (linear), 1 (nearest neighbor), 2 (gaussian), 3 (windowed sinc), 4 (bspline)

<sup>7</sup><http://stnava.github.io/ANTsR/>

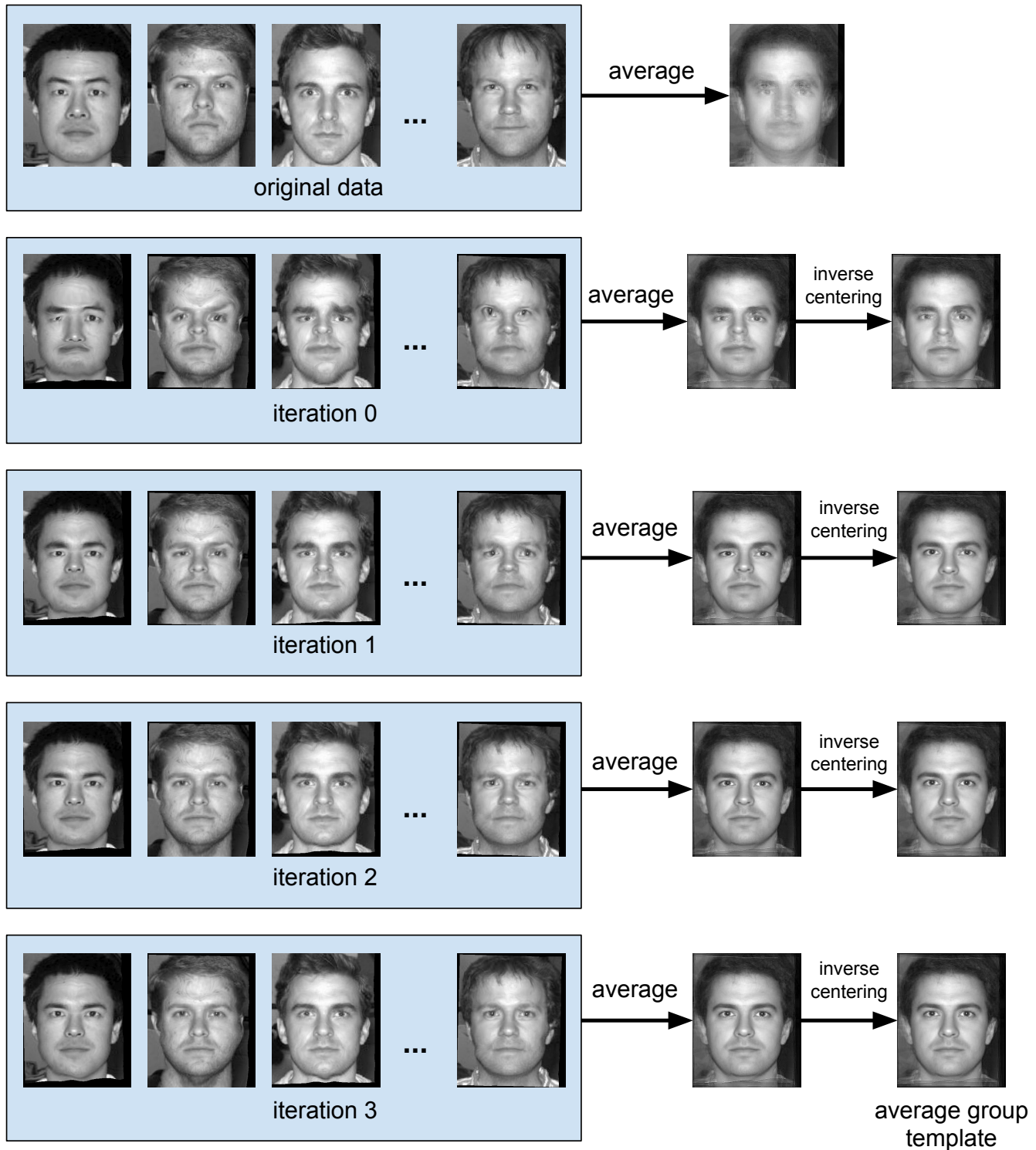


Figure 2.10: Process of creating an average group template

- `antsRegistration(fixed = NA, moving = NA, typeofTransform = "SyN", regIterations = c(40,20,0), ...)` – perform registration between two images.
  - `fixed` – fixed image to which we register the moving image
  - `moving` – moving image to be mapped to fixed space
  - `typeofTransform` – either a one stage rigid/affine mapping or a 2-stage affine+syn mapping. Mutual information metric by default. One of `Rigid`, `Affine`, `AffineFast`, `SyN` (mutual information as optimization metric), `SyNCC` (cross-correlation as optimization metric).
  - `regIteration` – vector of number of iteration at each level

This function outputs a list containing:

- `warpedmovout` – moving image warped to space of fixed image
- `warpedfixout` – fixed image warped to space of moving image
- `fwdtransforms` – transforms to move from moving to fixed image
- `invtransforms` – transforms to move from fixed to moving image
- `createJacobianDeterminantImage(domainImg, tx, doLog = 0)` – compute the jacobian determinant from a transformation file.
  - `domainImg` – image that defines transformation domain
  - `tx` – deformation transformation file name
  - `doLog` – return the log jacobian
- `getMask(img, lowThresh, highThresh)` – get a binary mask image from the given image after thresholding
- `smoothImage(img, sigma)` – smooth image
- `imageListToMatrix(imageList, mask)` – read images into rows of a matrix. Very useful for analysis of series of images.
  - `imageList` – a character vector containing a list of image files to read, in order - these are image objects, not file names
  - `mask` – an `antsImage` containing a binary mask, voxels in the mask are placed in the matrix. If not provided, estimated from first image in list.

## Chapter 3

### Pipeline and Data Analysis

In this chapter, a TBM (tensor-based morphometry) image processing pipeline is suggested and applied on real data using various tools and software bundles which were introduced in the previous chapter. From the output of this pipeline, a probabilistic map that shows which parts of the brain differ between groups (healthy and AD patients) is constructed. Furthermore, we present simple classifiers which should discriminate between brains of healthy and AD individuals.

#### 3.1 Data Overview

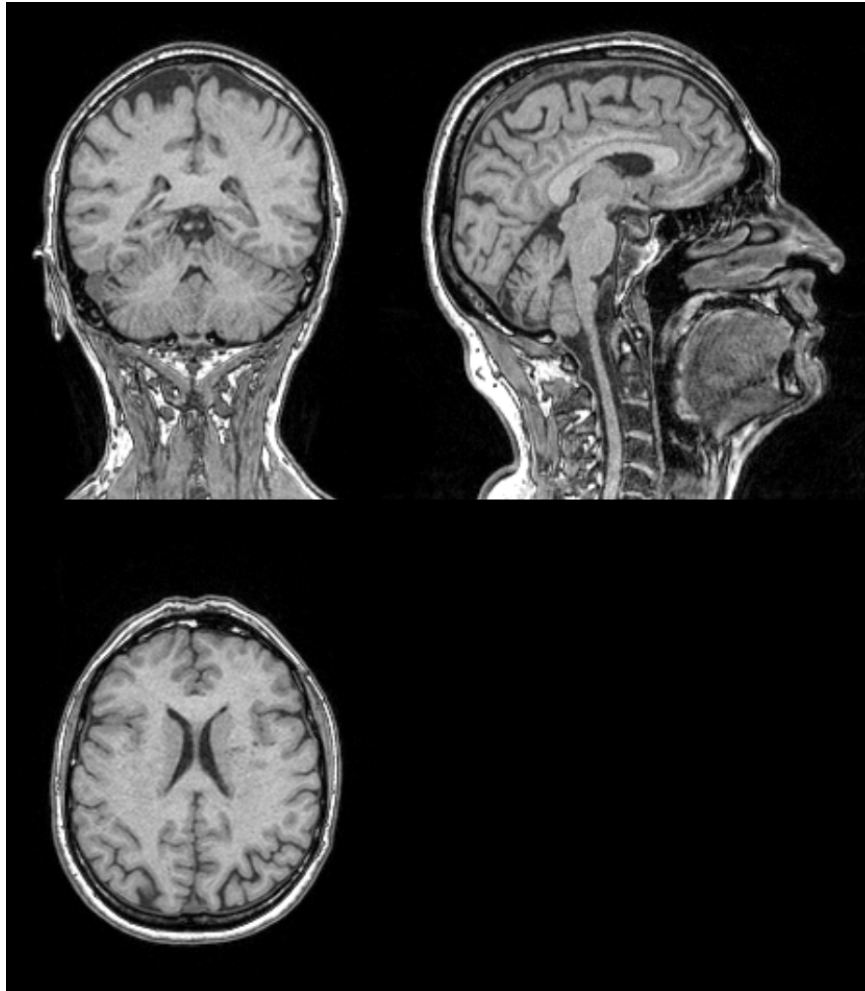
First, one needs to familiarize himself with the data. We were provided with a set of 46 head MRI scans of healthy individuals and 22 head scans of patients suffering from the Alzheimer's disease. All of them are T1-weighted and were acquired with Siemens Symphony or Avanti MRI scanner. The data were carefully chosen by an experienced neurologist. However, except for the fact that all these scans originate from elderly people no other information was delivered.

Each scan consists of 192 or 176 sagittal slices, each in a separate *dcm file* with the resolution of 512x512 pixels and 12-bit color (greyscale) depth. Physical spacing between slices is 1 mm and pixel spacing is 0.9765625 mm in both coordinates.

As the data set is quite small, one must carefully consider how to divide them between training and test set. A training set serves for training the model, whereas a test set is used for the estimate of the expected level of fit of the model. In our case, we will use 24 healthy individuals (NC, normal control) for creating an average template and the rest (22 NC and 22 AD) for the purpose of classifier. As the later set is not particularly large, it is not recommended to partition it in a conventional way (70 % training set and 30 % test set). Is it preferable to employ the *cross-validation* instead. That is to split the data into  $k$  equally sized disjoint subsets. Each subset is then used for the validation of the model trained by all left subsets. The validation results are averaged over all test subsets.

## 3.2 Image Pre-Processing Pipeline

Before entering the main TBM pipeline, the acquired data have to undergo a few pre-processing steps. Following transformations will be demonstrated on an example of a head MRI scan, see Fig. 3.1.



**Figure 3.1:** Original head MRI scan before pre-processing

- *Conversion to NIfTI format.* We use the `dcm2nii` utility from the MRICron package (see Sec. 2.6.2):

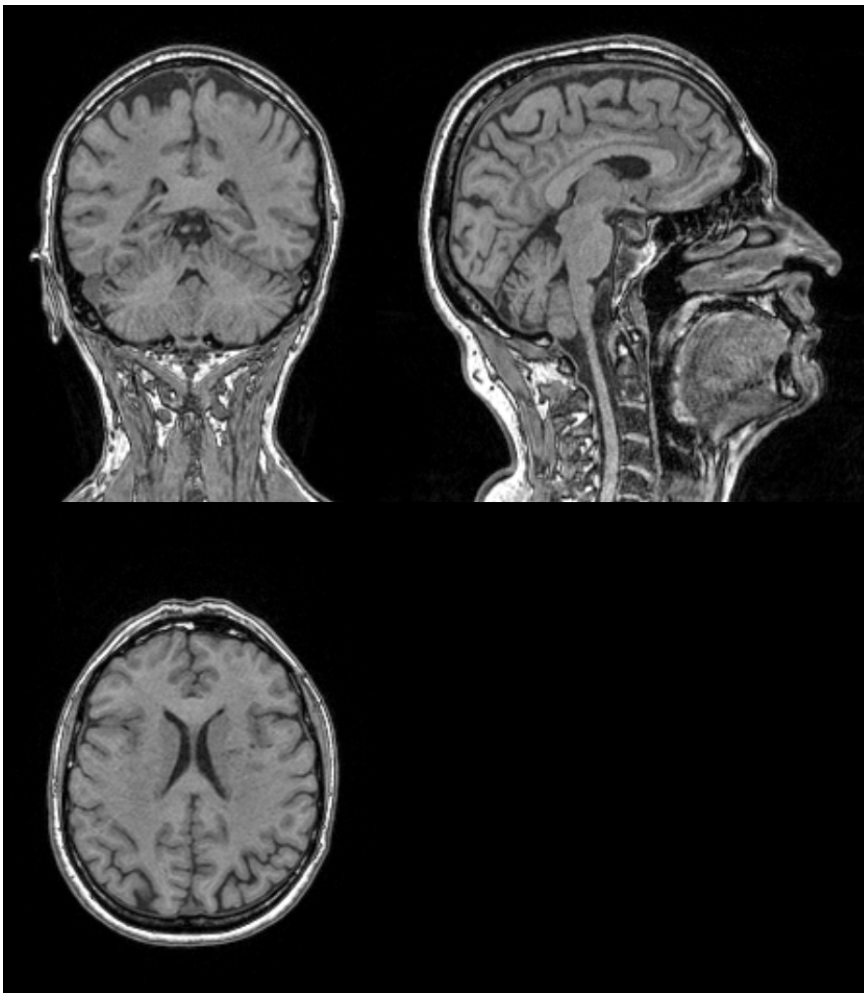
```
dcm2nii -o brain *.dcm
```

which converts all dcm files in a directory and names output file with *brain* prefix.

- *Bias field correction.* For the reasons mentioned in Sec. 2.4.1, it is necessary to remove the intensity inhomogeneity in the input MR images. For this purpose, we make use of the ANTs' `N4BiasFieldCorrection` utility with the following call (see Sec. 2.6.4):

```
N4BiasFieldCorrection
-d 3
-c [50x50x50x50,0.0000001]
-i input.nii.gz
-o repaired/repaired.nii.gz
-s 4
```

That means, the shrink factor is 4 and the number of iteration is 50, 50, 50 and 50 for each resolution respectively. These values were taken from several ANTs' examples as they prove to yield feasible results. Fig. 3.2 shows the previous scan corrected. It is worth mentioning that the color range has changed slightly.



**Figure 3.2:** Head MR image after bias correction

- *Resampling into isotropic space.* For the convenience of all following tools, it is necessary to resample the data in such way so that they

show the same physical sampling in all three dimensions. We use the *ResampleImage* program (see Sec. 2.6.4) from the ANTs bundle:

```
ResampleImage
  3
  repaired.nii.gz
  resampled.nii.gz
  1.0x1.0x1.0
  0
  4
```

That is, the 3-dimensional `repaired.nii.gz` file will be resampled with the spacing of 1 mm in all directions using a 3rd order B-Spline with an output name `resampled.nii.gz`.

Whereas the original examined head MRI scan had a resolution of 192x256x256 pixels with 1.00x0.96x0.96 mm spacing, the resampled one has a resolution of 192x250x250 pixels with 1.00x1.00x1.00 mm spacing.

- *Skull stripping.* A skull and non-brain parts of an head might negatively influence the sensitivity of the morphometric measures used. For this reason, we use the BET utility (see Sec. 2.6.3) from the FSL package to extract brain only from head MRI scans:

```
bet
  input.nii.gz
  output.nii.gz
  -f 0.5
  -R
```

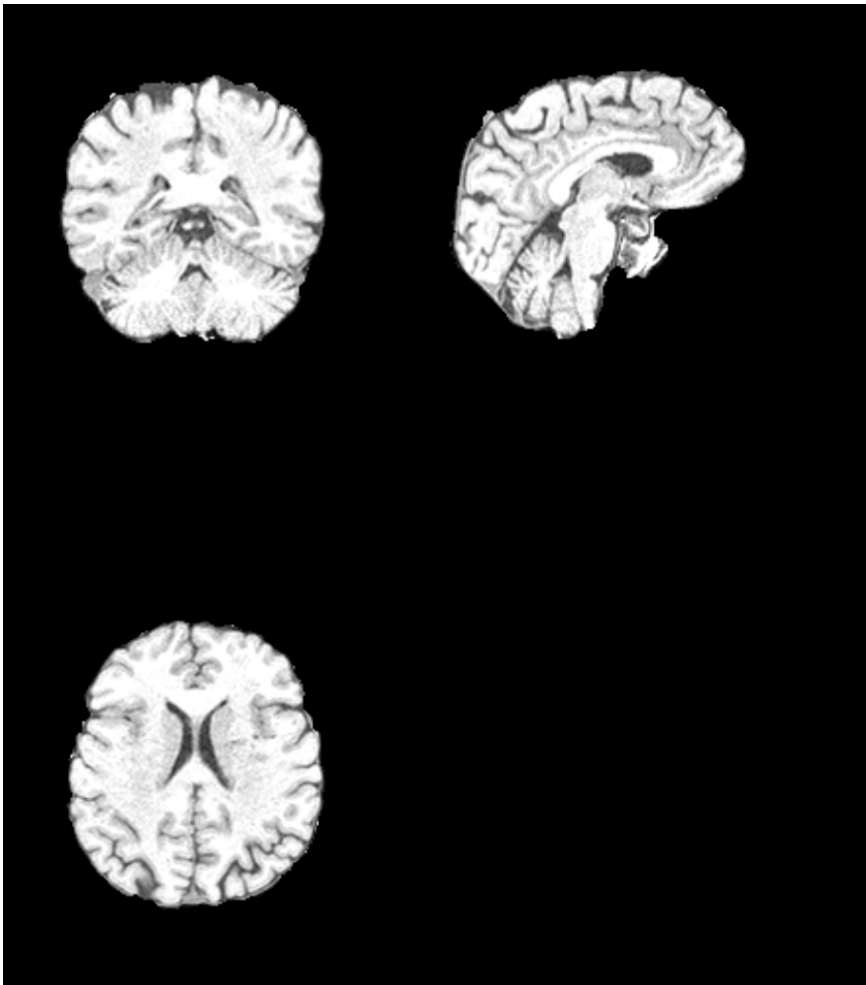
which calls the *BET* utility with a fractional intensity threshold of 0.5 and uses robust brain centre estimation (iterates BET several times). Although, this can take some time the results show much much better brain and non-brain separation than in case of a single iteration.

Fig. 3.3 shows the resulting bias corrected, isotropic resampled and skull stripped head MRI scan in three orthogonal slices. In Fig 3.4 showing a surface rendering view, a brain surface and single sulci are clearly recognizable. Therefore, we might assume that the skull stripping algorithm works well.

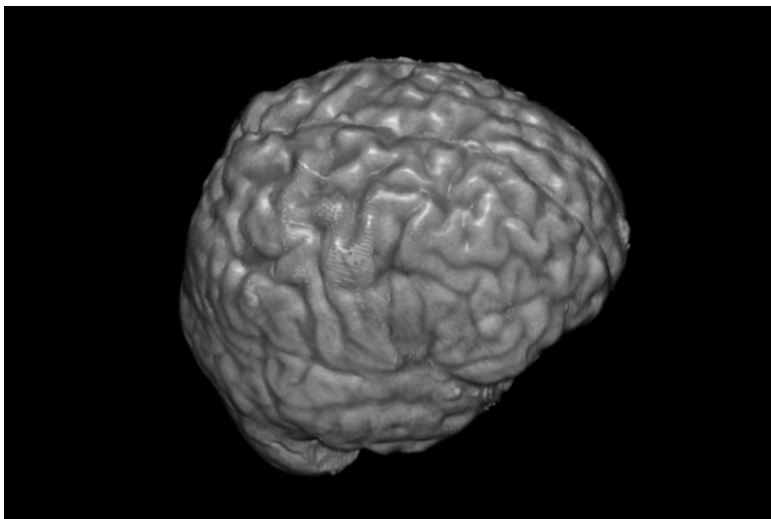
### 3.3 Average Template

Building an average template is a crucial part in the TBM pipeline as in the next steps, all examined brains are going to be compared to this template to make some inferences. From the amount of total 46 control scans (healthy scans), we reserve 24 of them for the purpose of building this template. From the detailed description in Sec. 2.4.2 and Sec. 2.6.4, it is obvious that this task





**Figure 3.3:** Skull stripped head MR image (orthogonal slices)



**Figure 3.4:** Skull stripped head MR image (surface rendering)

involves many non-linear registrations which can be very time consuming. For example, one SyN registration on an average PC (4 cores 2.20 MHz each, 4 GB RAM) takes approximately 1.5 hour. An average template from 10 brains requires about 40 SyN registrations. That means 60 hours or 2 days and 12 hours. For 24 brain volumes it will be even more. So as to shorten the execution time we, first, employed methods of parallel computing. ANTs package allows this using *PEXEC* utility. Secondly, we have run the computation on an high performing Xeon server with 32 cores (2.3GHz each), 64 threads and 128 GB RAM.

An average template is a part of various image processing pipelines therefore it is directly implemented in the ANTs package (see Sec. 2.6.4). We used the following syntax:

```
ITK_GLOBAL_DEFAULT_NUMBER_OF_THREADS=24
export ITK_GLOBAL_DEFAULT_NUMBER_OF_THREADS

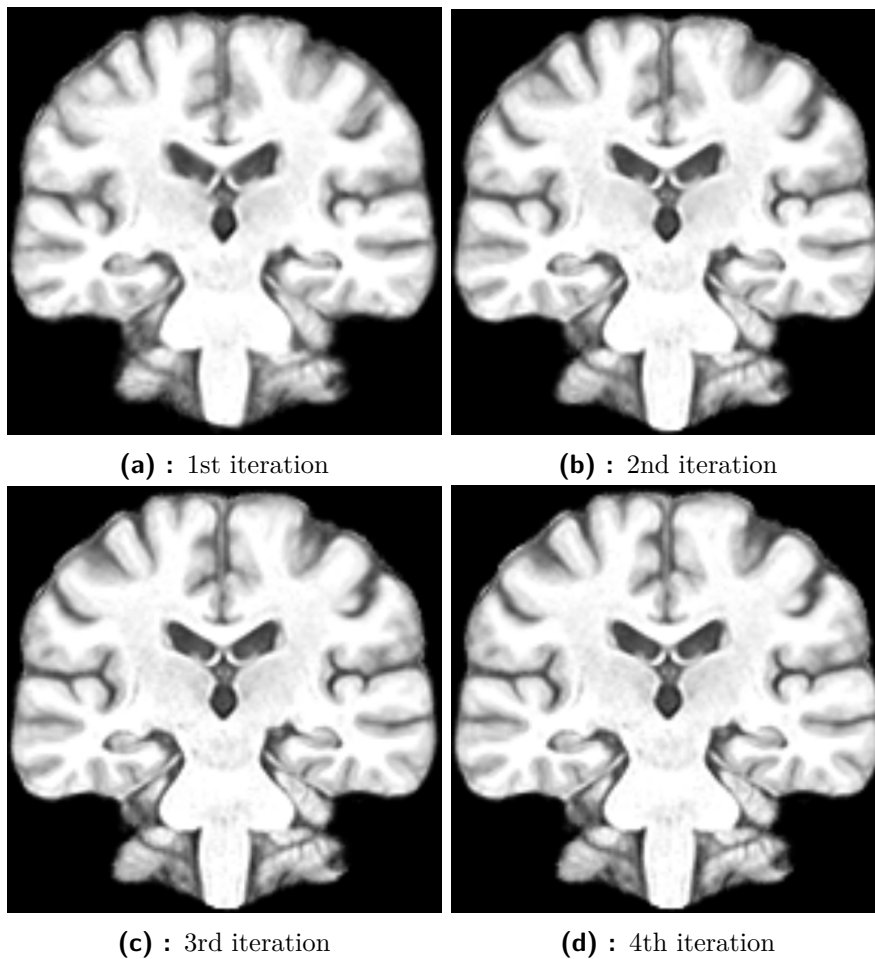
mkdir TemplateMultivariateBSplineSyN
inputPath=${PWD}/data
outputPath=${PWD}/TemplateMultivariateBSplineSyN/

${ANTSPATH}/antsMultivariateTemplateConstruction2.sh \
  -d 3 \
  -b 1 \
  -o ${outputPath}T_ \
  -i 4 \
  -g 0.25 \
  -e 0 \
  -j 6 \
  -c 2 \
  -k 1 \
  -w 1 \
  -f 8x4x2x1 \
  -s 3x2x1x0 \
  -q 100x70x50x10 \
  -n 1 \
  -r 1 \
  -l 1 \
  -m CC[2] \
  -t BSplineSyN[0.1,26,0] \
  ${inputPath}/*.nii.gz
```

Explanation: First, a global number of thread which can be used is set to 24. Then, `-c 2` enables the *PEXEC* option and `-j 6` sets the number of cores to use to 6. The computation will proceed in 4 iterations with shrink factors 8, 4, 2 and 1, smoothing sigmas 3, 2, 1 and 0 and a number of max iterations 100, 70, 50 and 10. The similarity metric is set to cross-correlation

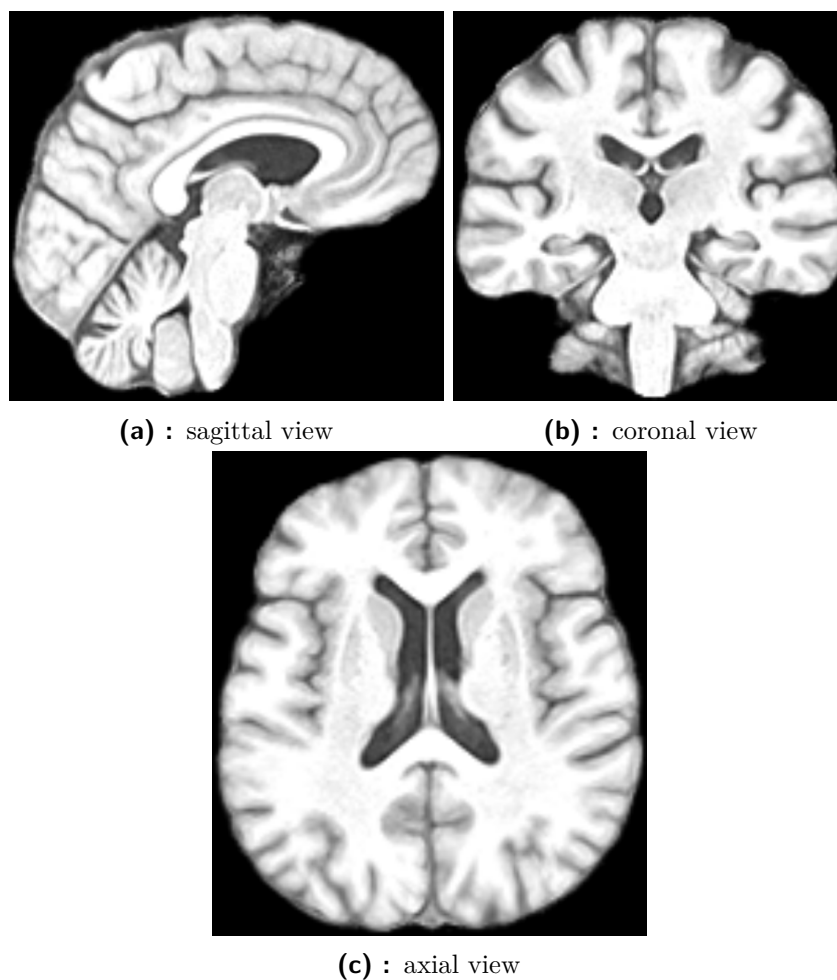
with a radius of 2 voxels and the transformation model is set to Greedy B-Spline SyN. This setting has arisen from example scripts and recommended settings. The running time on the Xeon server described above was 70 hours 24 minutes.

Fig. 3.5 shows coronal slices (number 110) of the temporary averages in each of the four iterations. Whereas the first iteration seems rather blurry, the following ones show more detail and clear separation of single lobes and sulci. The result of the last iteration is the Average Template.



**Figure 3.5:** Temporary average brains in the process of building an Average Template

One can see that the 3rd and 4th iteration do not differ much, therefore it would not probably bring any advantage to proceed with more than four iterations. For the sake of completeness, see Fig. 3.6 which shows the resulting Average Template in three orthogonal slices.



**Figure 3.6:** Average Template

## 3.4 Jacobian Determinant Image

Jacobian determinant image is an initial point for any further analysis. Therefore, we decided to derive it directly in the *R* environment (using ANTsR package). The exact procedure is described in detail in Sec. 2.3.3. That is, each individual image (after pre-processing) is non-linearly registered to the Average Template. From the resulting warp field a Jacobian matrix is derived. The following code demonstrates this procedure:

```
library(ANTsR)

#load the Average Template
t <- antsImageRead("template/T_template0.nii.gz")

#get AD data
files <- list.files(path="dataAD", pattern="*.nii.gz",
                    full.names=T, recursive=FALSE)
```

```

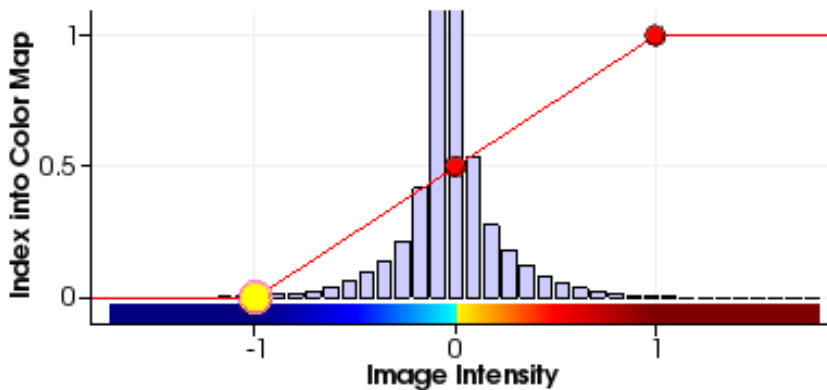
#lapply = nicer for-lopp
lapply(files, function(file) {
  #read pre-processed image
  m <- antsImageRead(file)

  #warp image (m) to the template (t)
  tx = antsRegistration( t, m, "SyNCC",
                        regIterations = c(100,50,40),
                        verbose = TRUE )

  #create log jacobian determinant map
  jac <- createJacobianDeterminantImage(t,
                                       tx$fdttransforms[1],
                                       doLog = 1)
})

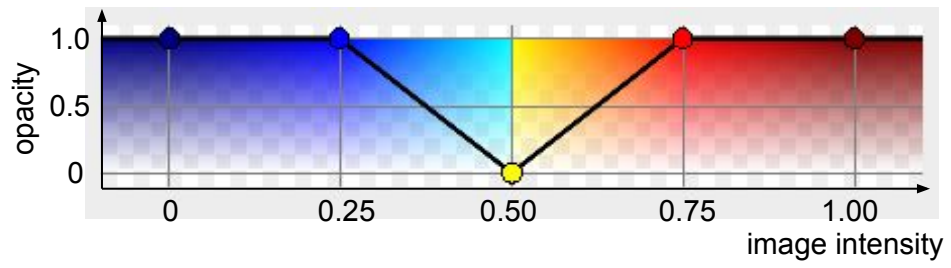
```

It is a common practice to use a logarithm on the jacobian determinant values as this makes the values symmetric about 0. There are two ways a jacobian map can be derived. Either from the forward warp field. Since such field points from every point in the template space to the individual space, the derived jacobian map also resides in the template space. Therefore this determinant jacobian map is commonly shown as an overlay over the group template. Another variant derives jacobian map from a backward mapping which is consequently shown over an individual image.

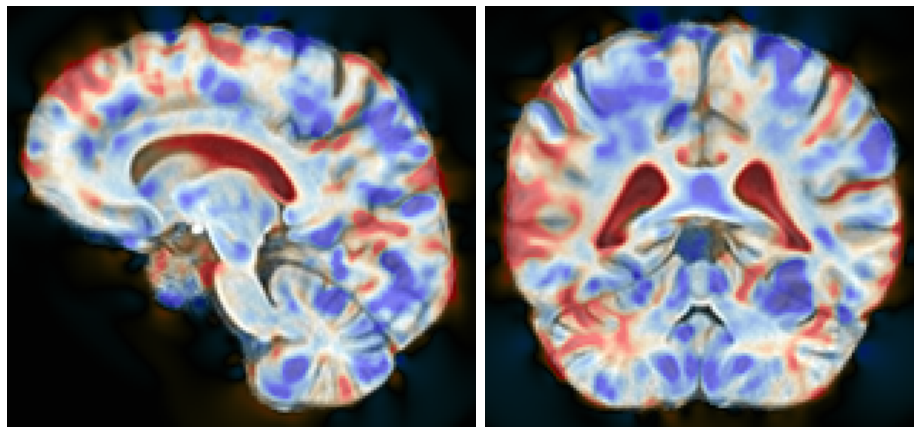


**Figure 3.7:** Histogram and clipping of a log jacobian determinant image

Fig. 3.7 shows a standard histogram of a log jacobian determinant image with an approximately symmetrical distribution about zero and steeply falling to the values of -1 and 1. For these reasons we clip the data at -1 and 1. The rest is colored according to the colormap shown in Fig 3.8. That is expanding areas (log jacobian determinant greater than zero) are set to red and shrinking areas are blue. Values close to zero are set to low opacity values so that only extremely expanding or shrinking areas are visible.

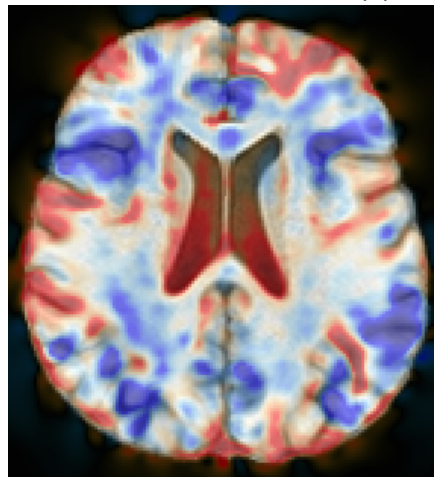


**Figure 3.8:** Suggested overlay colormap



**(a) :** sagittal view

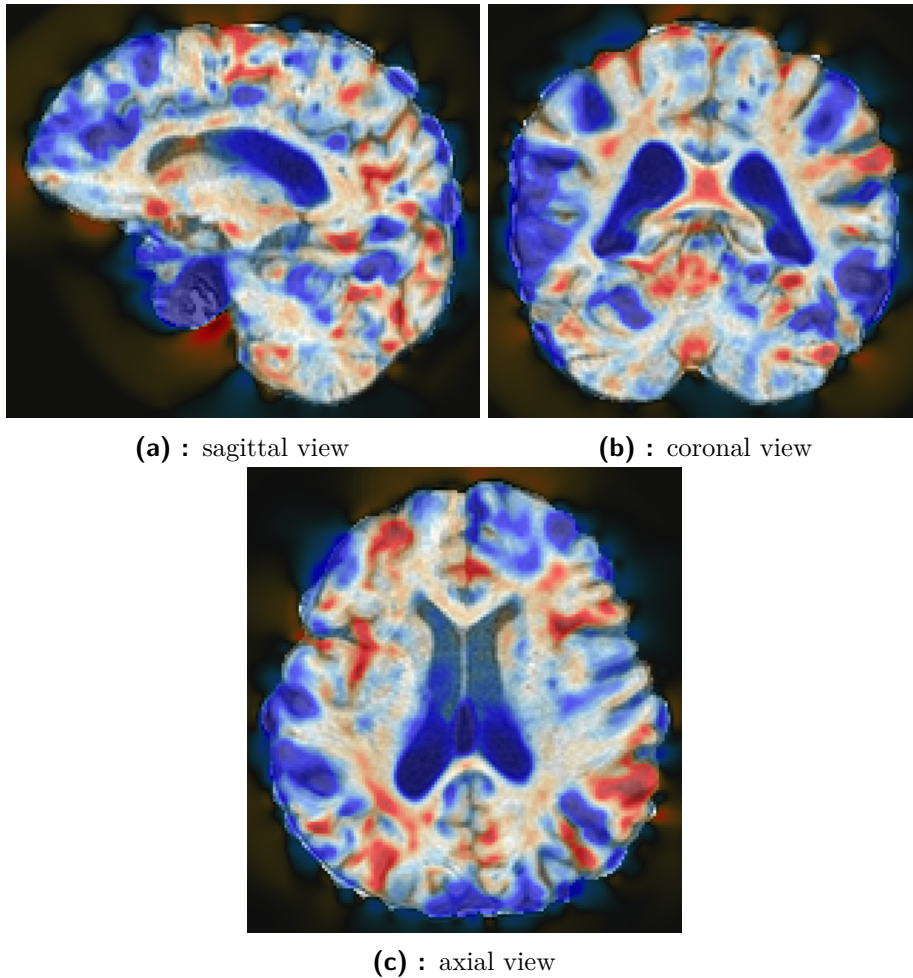
**(b) :** coronal view



**(c) :** axial view

**Figure 3.9:** Jacobian determinant map (in the template space)

For demonstration purposes, an MRI scan of an AD individual was processed by a proposed pipeline. Fig 3.9 shows log jacobian determinant map (3 orthogonal slices) overlaid on the Average Template. Red areas are bigger in the individual image compared to the template, blue ones are shrunk. Probably the most obvious feature are the lateral ventricles which are fully red meaning a particular expansion in comparison to the template. This mapping is quite useful for further statistics as it can be computed in one space only, regardless of an input image.



**Figure 3.10:** Jacobian determinant map (in the individual space)

Now, it is also possible to overlay the log jacobian map over the original image. This is useful when it is needed to see the original underlying structures. Fig. 3.10 shows the same map as the previous example, however, transformed to the individual space. It is quite interesting to notice that the coloring is roughly inverted. That is because in the previous case the original image was compared to the template, in this case, however, the template is compared to the original images. What was seen as expanding in one direction, is seen as shrinking in the opposite direction.

In principle, these maps are the main and essential output for the utilization in a medical set-up. A doctor can use them to easily locate brain locations which undergo some neurodegenerative changes and probably could make a fairly accurate diagnosis.

## 3.5 Statistical studies

However, this study can be taken further. To this point only individual brains were assessed, i.e., the question, how does one particular brain differ from the template, was answered. But we might ask whether and where do AD brains differ from healthy brain in general? What are the typical locations of atrophy in AD?

### 3.5.1 Statistical Parametric Mapping

First, we should address the question which parts of a brain are different at AD patients in comparison to normal controls. Basically, jacobian determinant maps of individuals from each population (AD and NC) are voxel-wise compared using the *t-test*. Then, a *p-value map* is overlayed over the Average Template to see where the brains of these two populations differ.

To make the method more robust, smoothing of the data before testing is required. This restricts the influence of noise and artifacts which can lead to false results.

After the *t-test*, a correction for multiple comparison should be performed as it is explained in Sec. 2.4.6. For this purpose, we use an R function `p.adjust(pvals, 'BH')` which uses Benjamini–Hochberg (BH) correction that is much less restrictive than the Bonferroni correction.

This method does not take into account the inherited spatial correlation of voxels. Bearing this in mind, we believe that this technique is satisfactory for the visualization of approximate locations of differences between groups.

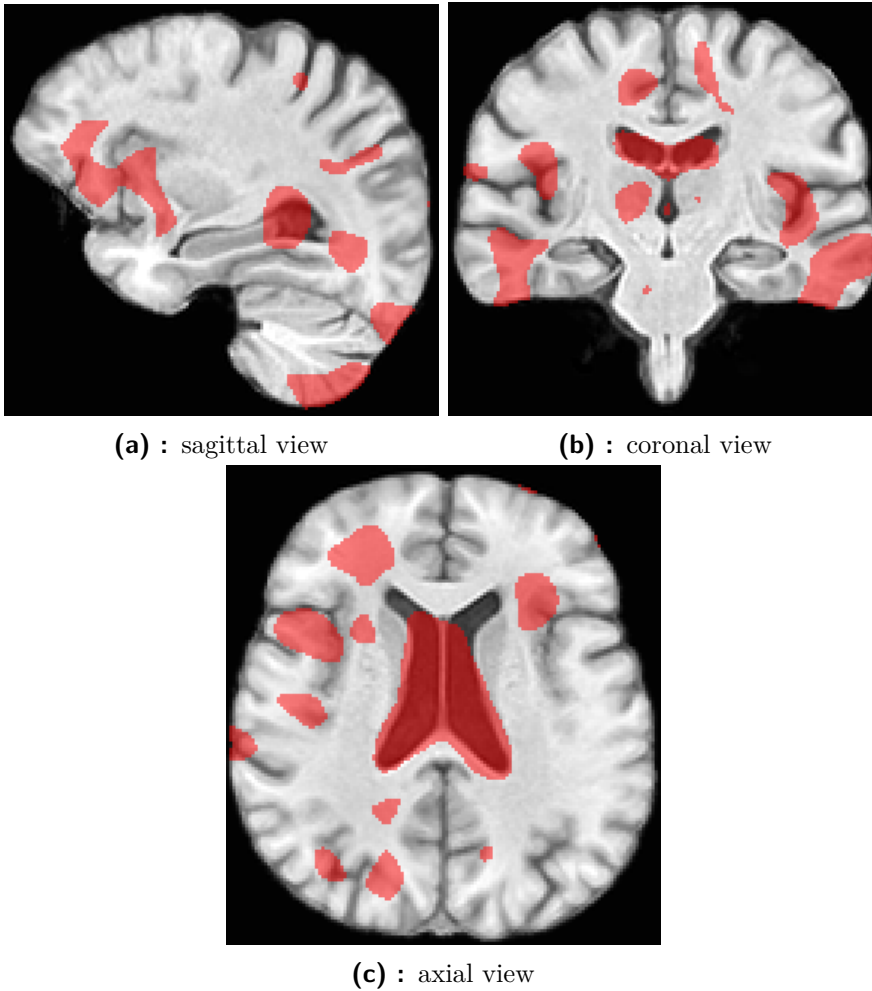
We performed this procedure with 22 NC scans (the ones which were not used for the template construction) and 22 AD scans. First, the log jacobian determinant maps were computed. Then they were Gauss-smoothed with  $\sigma = 4$  and a voxel-wise *t-test* was used. Afterwards, the resulting *p-values* (false negative rate) were adjusted with the BH correction.

Fig 3.11 shows the resulting *p-value map* on the Average Template. Red color stands for voxels with  $p < 0.05$ . Areas which are significantly different between groups are ventricles (fluid-filled spaces within the brain), temporal lobe, hippocampus, cerebellum and some parts of cortex.

### 3.5.2 Classification

Approximately knowing which parts of a brain are shrinking or expanding, a simple classifier could be built. We do not possess any additional information such as patient's gender, age, degree of AD, etc., therefore the following models might not be as sensitive as they could be.





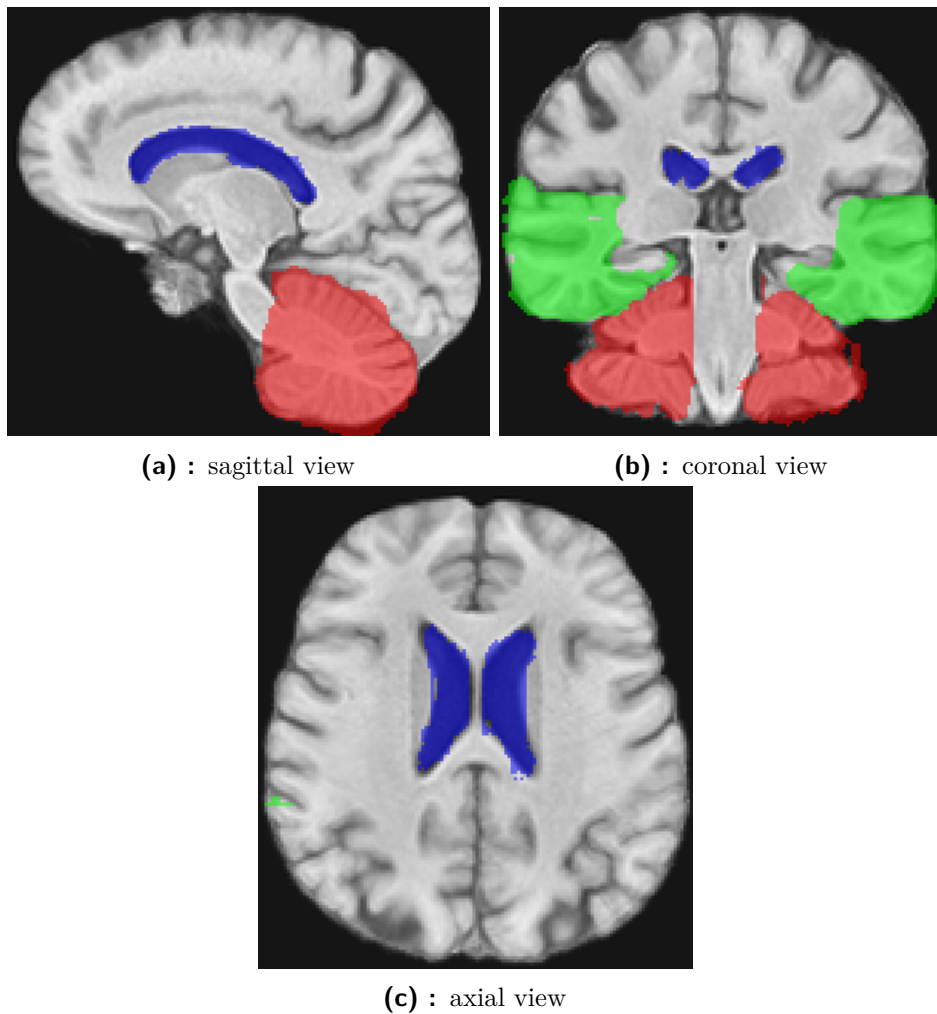
**Figure 3.11:** p-value map

Average atrophic rates within some region of interests (ROIs) are taken as attributes, i.e., a ROI is manually delineated on the Average Template and an average log jacobian determinant is computed in this area. Candidates for the ROIs were taken from the analysis above: cerebellum, temporal lobes, ventricles. Their segmentation was carried out in a software bundle *ITK-SNAP* which is a perfect tool for both viewing MR images and segmentation. The results can be seen in Fig. 3.12.

Another possible segmentation would be into white matter, grey matter and cerebrospinal fluid (CSF). To achieve this, there is an ANTsR command `kmeansSegmentation()`. The result of this segmentation is in Fig. 3.13

In the next step, an average log jacobian determinant of every ROI from every individual is computed. Results were recorded into a table, where columns are single attributes (average shrinkage rates) and each row (total 44) represents one individual, see Tab. 3.1. Last column is the known classification.

Such table could be quite useful for a neurologist to monitor patients with

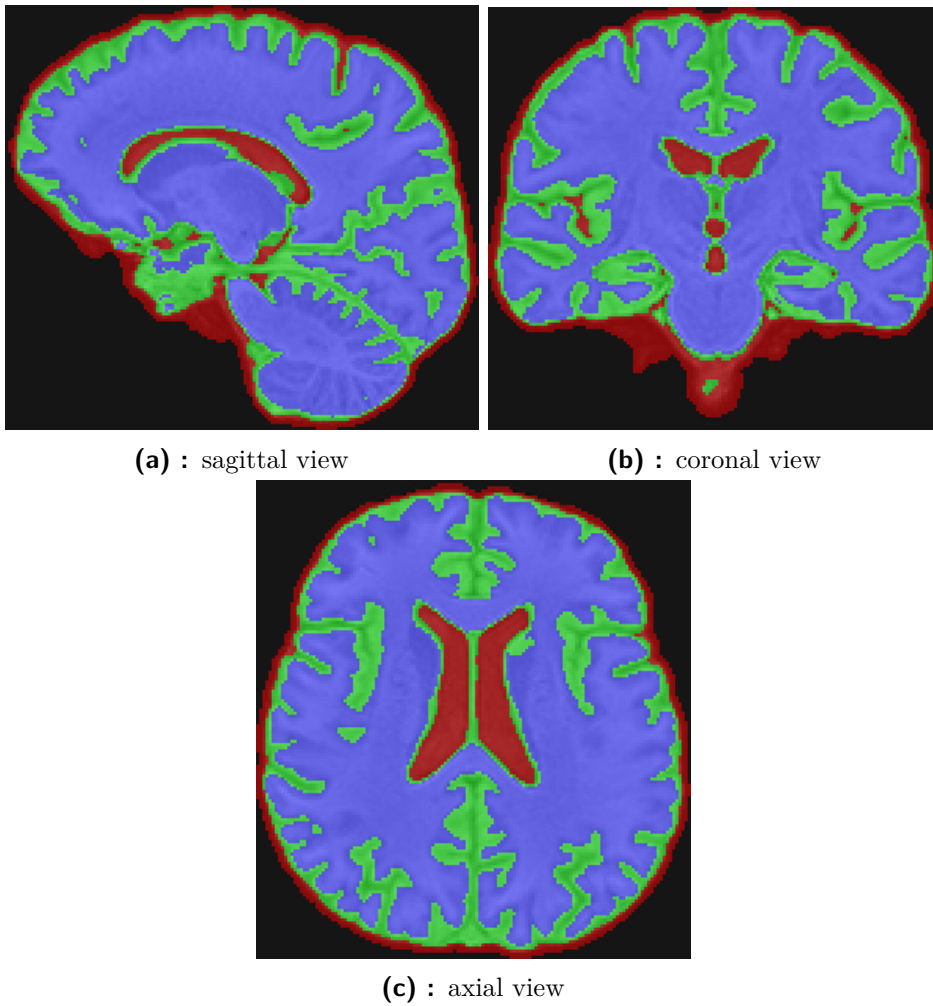


**Figure 3.12:** Manual segmentation of cerebellum (red), temporal lobes (green) and lateral ventricles (blue)

multiple scans. He could compare these atrophic summaries between single visits and tell, for example, whether the prescribed medication is taking affect or whether the atrophy worsens considerably.

However, we use this data set to train a few classifiers whose task is to tell NC and AD apart. For this purpose we use the following classifiers (also see Sec. 2.4.7):

- *Naive Bayes classifier* – Although the condition of the independence of attributes is obviously violated, it is a common practice to use Bayes classifier anyway.
- *Decision tree* – This kind of classifier is also used in order to detect which attributes contribute to the classification.
- *Support vector machine* – For the sake of simplicity a first-order polynomial kernel is used.



**Figure 3.13:** Automatic segmentation of white matter (blue), grey matter (green) and cerebrospinal fluid (red)

- *Neural network* – Number of units in the hidden layer is set to 3 to make the model more flexible but no more to avoid overfitting.

As described in Sec. 3.1, for the reason of a very limited data set, the classifiers are evaluated with a cross-validation. The data set is randomly divided into 4 subsets (4 is a divisor of a number of records in the data set) and each of them is used for testing of a model created from the remaining ones. Thus, every classifier is executed four times. Afterwards, a total classification accuracy, sensitivity and positive specificity is evaluated. Nevertheless, these ratios are only estimates of population proportions and they should be accompanied by confidence intervals. For random samples of sufficiently large size a Student t-distribution can be used. Unfortunately, this is not our case since the testing set contains only 11 samples and therefore we do not present confidence intervals because they would be very imprecise. By all means, it has to be kept in mind that these intervals are relatively wide for such small

	cerebellum	temporal lobes	lateral ventricles	CSF	grey matter	white matter	class
1	-0.0623	0.0333	0.2491	0.1893	0.0472	-0.0491	AD
2	-0.0492	0.0215	0.2412	0.1095	0.0099	-0.0513	AD
3	-0.0432	-0.0041	0.4482	0.1463	0.0088	-0.0571	AD
4-21	...						AD
20	-0.0516	0.0244	-0.1204	0.0651	0.0105	-0.0420	AD
21	-0.0050	-0.1881	-0.0422	0.0670	0.0202	-0.0588	AD
22	0.0057	-0.2064	-0.0569	0.0867	0.0371	-0.0527	AD
23	0.0318	0.0209	-0.7730	-0.0895	-0.0107	0.0065	NC
24	0.0190	0.0105	-0.7610	-0.0921	-0.0229	-0.0011	NC
25	0.0063	-0.0818	-0.0728	0.0100	-0.0394	-0.0291	NC
26-41	...						NC
42	-0.0108	-0.0021	0.6150	0.0760	-0.0314	-0.0658	NC
43	0.0519	0.0244	-0.1897	-0.0280	-0.0108	0.0054	NC
44	-0.0243	-0.0659	-0.1251	-0.0084	-0.0339	-0.0570	NC

Table 3.1: Classification attributes

number of samples.

An R code for the above described cross-validation follows:

```
#att - data frame
n_train <- dim(att)[1]
n_folds <- 4
folds_i <- sample(rep(1:n_folds, length.out = n_train))

class_err <- matrix(0, nrow = n_folds, ncol = 4)
formula <- class ~ .

for (k in 1:n_folds) {
  test_i <- which(folds_i == k)
  #train set
  train <- att[-test_i, ]
  #test set
  test <- att[test_i, ]

  #NAIVE BAYES
  model <- naiveBayes(formula, train)
  class_err[k,1] <-
    sum(predict(model,test) == test$class)/dim(test)[1]

  #DECISION TREE
  model <- rpart(formula, data = train,
                 control = rpart.control(minsplit = 5))
  class_err[k,2] <- sum(
    predict(model,newdata=att, type='class')
```

```

== att$class)/dim(att)[1]

#SUPPORT VECTOR MACHINE
model <- svm(formula, data=train, kernel='poly', cost=0.001,
             gamma = 10000, degree = 1, scale=FALSE)
class_err[k,3] <- sum(
  predict(model,newdata=att, type='class')
  == att$class)/dim(att)[1]

#NEURAL NETWORK
model <- nnet(formula, train, size=3, skip=TRUE,
             maxit=10000)
class_err[k,4] <- sum(
  predict(model,newdata=att, type='class')
  == att$class)/dim(att)[1]
}

```

The numeric outcome of this script is summarized in Tab. 3.2. The best classifiers for this kind of data are the Bayesian classifier and a neural network. They both show similar accuracy, sensitivity and positive predictivity. Around 80 % of data was correctly classified. If the image is of an AD patient, the algorithms will find it out with the probability of approx. 90 %. From all individuals classified as AD, only 79 % are indeed AD. These percentages are probably not the highest possible but we consider this result as satisfactory bearing in mind that we were given data of a mixed gender, age and AD severity.

	<b>Bayes</b>	<b>Decision tree</b>	<b>SVM</b>	<b>Neural network</b>
<b>Accuracy</b>	81.82 %	77.27 %	70.45 %	81.82 %
<b>Sensitivity</b>	86.36 %	81.82 %	77.27 %	90.91 %
<b>Positive predictivity</b>	79.17 %	75.00 %	68.00 %	76.92 %

**Table 3.2:** Evaluation of classification

Another useful piece of information would be to know which attributes contributed the most to the classification. A method to find this out is to utilize the way a decision tree is constructed. Each node in such tree is assigned an attribute with the most informative value, i.e., the one with the lowest entropy. Thus, the nodes near to the trunk of a tree are the most informative and contribute the most to the classification and the ones at the bottom of a tree or not even used are not so important.

Fig. 3.14 shows a tree diagram of the decision tree created by the script mentioned above. It is obvious that white matter and CSF attributes have the most discriminative power. Then, grey matter and temp attributes are far less discriminative and the other attributes have not even contributed to the model. That is, an average shrinkage of cerebellum seems not to be so important for the classification. On the other hand, expansion of lateral

ventricles might be important, however, it is certainly a subset of CSF which apparently carries even more information and therefore the ventricles were omitted in our model.

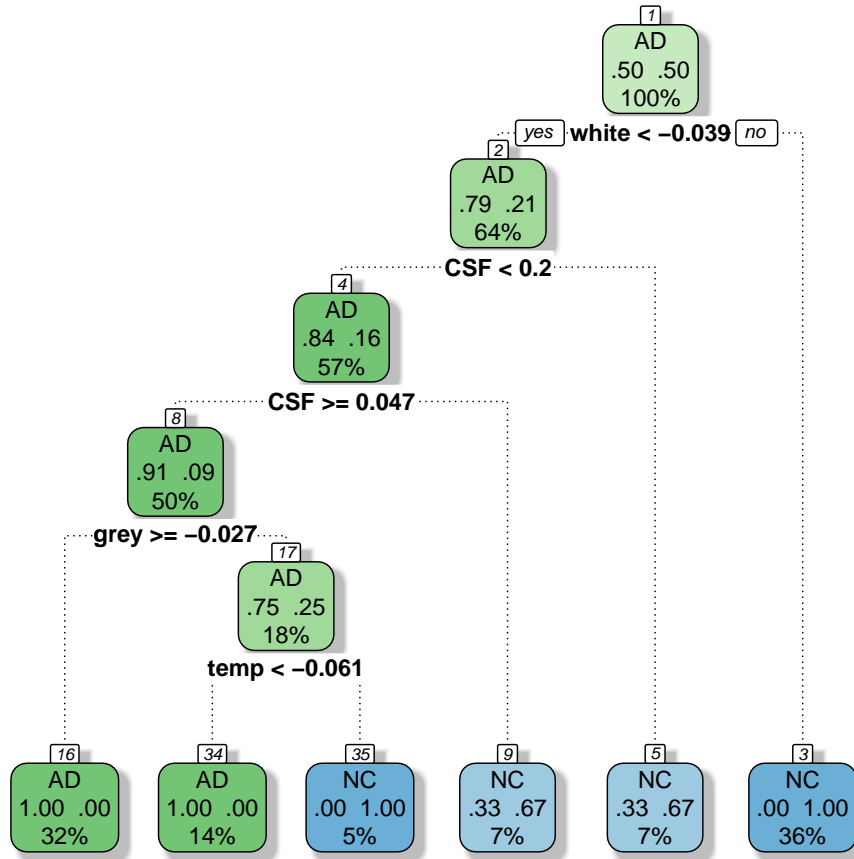



Figure 3.14: Decision tree diagram



## Chapter 4

### Conclusion

The aim of this thesis was to give an account on evaluating neurodegenerative diseases, Alzheimer's disease in particular, with the help of magnetic resonance diagnostics, to introduce basic utilities for processing of MR images and to design a simple transformation pipeline for an assessment of brain atrophy among given populations.

The first part starts with a background knowledge of magnetic resonance imaging and Alzheimer's disease. Then, an essential method for image processing – brain morphometry – is thoroughly examined. Since this technique heavily depends on image registration, the next part addresses this topic as well. The last section of the theoretical chapter presents several analysis tools for MRI with short snippets of code to show their capabilities.

The second part is concerned with the very data analysis. First, an image pre-processing pipeline is presented accompanied with various illustrations of the outcomes of single steps. Afterwards, an average template was created and used for deriving of jacobian determinant images, i.e., showing which parts of the brain have expanded or shrunk. We can claim that the graphical results fit well the general model of an atrophied brain in Alzheimer's disease (AD). We took the analysis further and statistically examined where brains of healthy individuals and AD patients differ in general. The output of this process is a probability map telling which parts of the brain are likely to be impaired in AD. Based on this, we were able to identify and delineate several brain sections, such as lateral ventricles or temporal lobes, which prove to distinguish healthy and AD brain. These findings led to creation of a few classifiers, which should be able to discriminate between these two populations. Without any prior knowledge we were able to train a classifier (3-layer neural network) with the accuracy of 80 % and sensitivity of 90 % which we find it to be a very satisfactory result.

Methods and outcomes presented in this thesis could be developed further. It would be advisable to recompute the average template with a larger input data set and, consequently, to test classifiers in order to obtain more accurate estimates of their performance.

Furthermore, we believe that the color overlay maps of shrinkage and the numerical summaries of atrophy could find their place among standard neurological diagnostic tools. Both of these have a great potential to provide

#### 4. Conclusion

---

valuable information about atrophic changes of the brain and even easily allow to track its development in time.



## Appendix A

### Bibliography

- [AF01] John Tower Ashburner and KJ Friston, *Computational neuroanatomy*, University of London England, 2001.
- [ATS09] Brian B Avants, Nick Tustison, and Gang Song, *Advanced normalization tools (ants)*, *Insight J* **2** (2009), 1–35.
- [BFBS11] Frederik Barkhof, Nick C Fox, António J Bastos-Leite, and Philip Scheltens, *Normal ageing*, *Neuroimaging in Dementia*, Springer, 2011, pp. 43–57.
- [Bis06] Christopher M. Bishop, *Pattern recognition and machine learning (information science and statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [CHH14] William R Crum, Thomas Hartkens, and DLG Hill, *Non-rigid image registration: theory and practice*, *The British Journal of Radiology* (2014).
- [Dam71] Raymond Damadian, *Tumor detection by nuclear magnetic resonance*, *Science* **171** (1971), no. 3976, 1151–1153.
- [FFF<sup>+</sup>04] Richard SJ Frackowiak, Karl J Friston, Christopher D Frith, Raymond J Dolan, and JC Mazziotta, *Human brain function*, San Diego, CA (2004).
- [FFJ<sup>+</sup>10] Giovanni B Frisoni, Nick C Fox, Clifford R Jack, Philip Scheltens, and Paul M Thompson, *The clinical use of structural mri in alzheimer disease*, *Nature Reviews Neurology* **6** (2010), no. 2, 67–77.
- [Gro] Data Format Working Group, *Nifti – neuroimaging informatics technology initiative*, <http://nifti.nimh.nih.gov/>, Accessed: 2016-04-25.
- [HH01] J.V. Hajnal and D.L.G. Hill, *Medical image registration*, Biomedical Engineering, Taylor & Francis, 2001.

- [HLP<sup>+</sup>08] Xue Hua, Alex D Leow, Neelroop Parikshak, Suh Lee, Ming-Chang Chiang, Arthur W Toga, Clifford R Jack, Michael W Weiner, Paul M Thompson, Alzheimer's Disease Neuroimaging Initiative, et al., *Tensor-based morphometry as a neuroimaging biomarker for alzheimer's disease: an mri study of 676 ad, mci, and normal subjects*, Neuroimage **43** (2008), no. 3, 458–469.
- [HX11] Thompson P. M. Hua X., *Tensor-based morphometry*, Alzheimer's Disease Neuroimaging Initiative (2011).
- [KAA<sup>+</sup>09] Arno Klein, Jesper Andersson, Babak A Ardekani, John Ashburner, Brian Avants, Ming-Chang Chiang, Gary E Christensen, D Louis Collins, James Gee, Pierre Hellier, et al., *Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration*, Neuroimage **46** (2009), no. 3, 786–802.
- [KWE75] Anil Kumar, Dieter Welti, and Richard R Ernst, *Nmr fourier zeugmatography*, Journal of Magnetic Resonance (1969) **18** (1975), no. 1, 69–83.
- [Kyb] Jan Kybic, *Zobrazovací systémy v medicíně a zpracování obrazové informace v medicíně*, Lecture Materials.
- [Lau73] Paul C Lauterbur, *Image formation by induced local interactions: examples employing nuclear magnetic resonance*.
- [LIVL14] Christian Thode Larsen, J Eugenio Iglesias, and Koen Van Leemput, *N3 bias field correction explained as a bayesian modeling method*, Bayesian and graphical Models for Biomedical Imaging, Springer, 2014, pp. 1–12.
- [MMGP07] Donald W McRobbie, Elizabeth A Moore, Martin J Graves, and Martin R Prince, *Mri from picture to proton*, Cambridge university press, 2007.
- [Rod08] Anne Brown Rodgers, *Alzheimer's disease: Unraveling the mystery*, National Institutes of Health, 2008.
- [SBB<sup>+</sup>16] P. Scheltens, K. Blennow, MMB Breteler, B. de Strooper, GB Frisoni, S. Salloway, and WM Van der Flier, *Alzheimer's disease*, LANCET **388** (2016), no. 10043, 505–517 (English).
- [Smi02] Stephen M Smith, *Fast robust automated brain extraction*, Human brain mapping **17** (2002), no. 3, 143–155.
- [Suc15] Olaf Such, *Magnetic resonance imaging (mri)*, University Lecture, 2015.
- [TAC<sup>+</sup>10] Nicholas J Tustison, Brian B Avants, Philip A Cook, Yuanjie Zheng, Alexander Egan, Paul A Yushkevich, and James C Gee, *N4itk: improved n3 bias correction*, IEEE transactions on medical imaging **29** (2010), no. 6, 1310–1320.

- [VPL07] Uro Vovk, Franjo Pernus, and Botjan Likar, *A review of methods for correction of intensity inhomogeneity in mri*, IEEE transactions on medical imaging **26** (2007), no. 3, 405–421.
- [ZBS01] Yongyue Zhang, Michael Brady, and Stephen Smith, *Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm*, Medical Imaging, IEEE Transactions on **20** (2001), no. 1, 45–57.