Improvement of MRI brain segmentation

Fully multispectral approach from the 'New Segmentation' method of Statistical Parametric Mapping

Ángel Diego Cuñado Alonso

Kongens Lyngby 2011
IMM-M.Sc.-2011-58

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-MSC: ISSN

Summary

The PET scanners show the metabolic activity of the studied biological tissues and they are very important in the clinical diagnosis of brain diseases. They generate low resolution images that can be improved with the estimated GM volume of the brain. The MRI scanners provide high resolution and can be optimized for the segmentation of anatomical structures. Therefore, the goal of this project is the improvement of a state-of-the-art automatic method that segments MRI brain volumes into GM, WM and CSF tissues.

The 'New Segmentation' method implemented in SPM8 allows multispectral input data, but it assumes non-correlated modalities. Therefore, this thesis modifies this method and its Matlab implementation in order to include correlation between modalities in the generative model, and hence use all the potential of multispectral approaches.

The modified method was compared to other uni-modal and multi-modal methods in the segmentation of two different datasets. The results showed that the multi-modal approaches were better that the uni-modal. In addition, the obtained Dice scores of the modified method were slightly higher than the ones of the original method. It was also visually checked the segmented volumes from original and modified method, and it showed that the latter is able to segment better the voxels that lie in the interface among several tissues.

Preface

This thesis was prepared at the Department of Informatics and Mathematical Modelling (IMM) of the Technical University of Denmark (DTU), as a partial fulfillment of the requirements for acquiring the M.Sc. degree in *Mathematic Modelling and Computer Science*. The project was done in close collaboration with the Neurobiology Research Unit (NRU) of Rigshospitalet in Copenhagen and the Danish Research Centre for Magnetic Resonance (DRCMR) of Hvidovre Hospital.

The work was supervised at DTU by prof. Rasmus Larsen, head of the *Image Analysis & Computer Graphics* section, and Ph.D. Koen Van Leemput, while Ph.D. Claus Svarer was the supervisor from NRU. The external collaborators at DRCMR were Ph.D. William Baare and Ph.D. Arnold Skimminge.

The time period of this thesis went from February 2011 to August 2011 with an assigned workload of 30 ECTS credits.

The focus of this work is based on medical image analysis, especially on Magnetic Resonance Image (MRI) brain segmentation.

Lyngby, 2011

Ángel Diego Cuñado Alonso

Acknowledgements

I wish to thank all the people involved in this thesis who helped me to overcome the difficulties of this challenging project. I would like to thank my supervisors prof. Rasmus Larsen and Ph.D. Koen Van Leemput for his guidance and technical feedback. A special thanks goes to my supervisor Ph.D. Claus Svarer for his personal support and for sharing with me his wide MRI expertise in numerous interesting discussions.

I would like also to thank the external collaborators Ph.D. William Baare and Ph.D. Arnold Skimminge, who introduced the acquisition setup and provided the mri dataset. I thank as well the collaboration of prof. Knut Conradsen to help me to develop a critical and rigorous approach to the problems.

And last but not least, I wish to thank the company and support of my partner, family and friends.

Mange tak,

Contents

Summary i

Preface iii

Acknowledgements v

Contents ix

Acronyms xi

1 Introduction 1
1.1 Motivation . . . 1
1.2 Dataset . . . 2
1.3 Baseline . . . 4
1.4 Project goal . . . 6
1.5 Specifications . . . 6
1.6 Thesis Outline . . . 7

2 Background 9
2.1 Brain Anatomy . . . 10
2.2 Magnetic Resonance Imaging . . . 12
2.3 Segmentation . . . 16

3 Neuroimaging Processing 23
3.1 Intensity model . . . 24
3.2 Registration . . . 30
3.3 Bias Field Correction . . . 40
3.4 Scalp-Stripping . . . 43
3.5 Smoothing . . . 46
3.6 Priors and Templates . . . 48

4 Method & Implementation 51
4.1 Objective Function . . . 52
4.2 Optimization . . . 65
4.3 Implementation . . . 76

5 Validation 87
5.1 Outputs . . . 88
5.2 Golden Standard . . . 91
5.3 Brain f4395 - Visualization . . . 92
5.4 BrainWeb phantoms - Dice Score . . . 104
5.5 CIMBI database - Age-Profile . . . 112

6 Discussion 115
6.1 Resume . . . 115
6.2 Future Work . . . 118
6.3 Conclusions . . . 119

Bibliography 128

Appendices 129

A Magnetic Resonance 131
A.1 Nuclear Magnetic Resonance . . . 131
A.2 Image Generation . . . 136

B Mathematics 137
B.1 Gaussian distribution . . . 138
B.2 2D Gaussian expression . . . 142
B.3 Cost Function of M-step . . . 145
B.4 Central and non-central moments . . . 146
B.5 Solution to a third degree equation . . . 149
B.6 Registration . . . 152

C SPM 155
C.1 Input Variables . . . 156
C.2 Original Code . . . 157
C.3 Modified Code . . . 159
C.4 Modified Code (version 2) . . . 162

D Results & Validation 167
D.1 Mixture parameters at each iteration for f4395. . . . 168
D.2 Representation of the clusters for all the tissue classes. . . . 174
D.3 Dice coefficient for BrainWeb phantoms. . . . 178
D.4 Segmentation of the BrainWeb phantoms. . . . 181
D.5 Atrophy. . . . 185

E Volumes **191**
E.1 MRI Data 192
E.2 Tissue Probability Maps for Prior Templates 194
E.3 Segmentation of volumes from subject f4395. 202

F Matlab code **211**

List of Figures **221**

List of Tables **223**

List of Algorithms **225**

Acronyms

AD Alzheimer Disease.

AIDS Acquired Immune Deficiency Syndrome.

ANN Artificial Neural Network.

BG Background.

BIC Brain Imaging Centre.

BSE Brain Surface Extractor.

BST Brain Extraction Tool.

CIMBI Center for Integrated Molecular Brain Imaging.

CNS Central Nervous System.

CSF CerebroSpinal Fluid.

CT Computed Tomography.

DCT Discrete Cosine Transform.

DFT Discrete Fourier Transform.

DRCMR Danish Research Centre for Magnetic Resonance.

DSC Dice Similarity Coefficient.

DST Discrete Sine Transform.

DTI Diffusion Tensor Imaging.

DTU Technical University of Denmark.

EM Expectation Maximization.

EMS Expectation Maximization Segmentation.

EPI Echo-Planar Imaging.

FAST FMRIB Automated Segmentation Tool.

FID free-induction decay.

FM Finite mixture.

fMRI functional Magnetic Resonance Image.

FMRIB Functional Magnetic Resonance Imaging of the Brain.

FN False Negative.

FP False Positive.

FSL FMRIB Software Library.

FWHM Full Width at Half Maximum.

GEM Generalized Expectation Maximization.

GM Grey Matter.

GMM Gaussian Mixture Models.

GNU General Public Licence.

GRE gradient echo.

HC Healthy Control.

HMM Hidden Markov Model.

HMRF Hidden Markov Random Field.

HWA Hybrid Watershed Algorithm.

i.i.d Independent and Identically Distributed.

ICBM International Consortium for Brain Mapping.

ICC IntraCraneal Cavity.

ICM Iterated Conditional Modes.

IMM Informatics and Mathematical Modelling.

LC Linear Combination.

LE Least Squares.

LM Levenberg-Marquardt.

MAP Maximum A Posteriori.

McStrip Minneapolis Consensus Strip.

MI Mutual Information.

ML Maximum Likelihood.

MNI Montreal Neurological Institute.

MoG Mixture of Gaussians.

MR Magnetic Resonance.

MRF Markov Random Field.

MRI Magnetic Resonance Image.

MS Multiple Sclerosis.

NCC Normalized Cross Correlation.

NIfTI Neuroimaging Informatics Technology Initiative.

NMI Normalized Mutual Information.

NMR Nuclear Magnetic Resonance.

NN Nearest Neighbour.

NRU Neurobiology Research Unit.

ORNLM Optimized Rician Non-Local Means.

PD Proton Density.

PET Positron Emission Tomography.

ppm parts per million.

PVE Partial Volume Effect.

r.v. Random Variable.

RF Radio Frequency.

ROI Region Of Interest.

s.t.d. Standard Deviation.

SANLM Spatial Adaptive Non-Local Means.

SBD Simulated Brain Database.

SE spin echo.

sMRI structural Magnetic Resonance Image.

SNR Signal to Noise Ratio.

SPECT Single Photon Emission Computed Tomography.

SPM Statistical Parametric Mapping.

SR saturation recovery.

SSD Sum of Squared Differences.

ST Soft Tissue.

STAPLE Simultaneous Truth and Performance Level Estimation.

SVM Support Vector Machine.

TE echo time.

TN True Negative.

TP True Positive.

TPM Tissue Probability Map.

TR time repetition.

VBM voxel-based morphometry.

VOI Volume Of Interest.

WM White Matter.

Chapter 1

Introduction

1.1 Motivation

Figure 1.1: Neurobiology Research Unit.

The Neurobiology Research Unit (NRU) of Rigshospitalet in Copenhagen (Denmark) has a particular interest in the precise segmentation of sub-cortical structures of the brain with Positron Emission Tomography (PET) scans. This kind of neuroimaging technique shows the metabolic activity of the studied biological tissues, and it is usually corrupted by artifacts that can be compensated with the anatomy of the associated structures. For this anatomy estimation is used the Magnetic Resonance (MR) images [89]. In addition, MRI scans have a higher resolution ($\sim 1mm$) over PET ($\sim 8mm$) that allows an improved Partial Volume Effect (PVE) correction [64].

The high resolution MR images are segmented into Grey Matter (GM), White Matter (WM) and CerebroSpinal Fluid (CSF) with a certain probability to generate Volume Of Interest (VOI) brain templates that are used afterwards in the reconstruction (co-registration) of PET images, as described by C. Svarer et al. [79]. In this way, it is possible to correlate the number of receptors in PET scans with the GM volume in MR images.

1.2 Dataset

The MRI dataset includes approximately 200 T_1 and T_2 weighted volumes from a 3T scanner. The original resolution for T_1 and T_2 modalities is $\sim 1mm$ and $\sim 1.1mm$, respectively. Although, they are re-sliced to have a final resolution of $\sim 1mm$ isotropic voxels. The scans are recorded at the DRCMR [53] of Hvidovre Hospital as part of the Center for Integrated Molecular Brain Imaging (CIMBI) project [58]. There are also available another 200 images with an old scanner and other images from brains with some pathologies, like Tourette syndrome, Obsessive Compulsive Disorders, Obesity, Winter Blues depression, and others; but they have been initially discarded for this project. All the scans were acquired with the Magnetom Trio scanner of Siemens [57].

Figure 1.2: Magnetom Trio scanner of Siemens.

The MR scans are made on volunteers, therefore the generated intensity volumes are not clinical data. Besides, all the images have been visually inspected by experts to ensure 'healthy' brains, which implies not lesions or neurodegenerative diseases. Hence, they are treated as Healthy Control (HC) subjects. The age span of the volunteers goes from 20 to 90 years, with scarce samples around 40 years old. For some subjects, there are several scans with some weeks or years in between; and others will be asked to repeat the scan in the following months.

The whole data set have been recorded with the same scan and acquisition protocol, which did not suffer any update or modification. The images have been co-registered in order that T_1 and T_2 scans are in the same spatial coordinate system and with the same resolution. However, the number of scans that have been re-sliced after the normalization is much smaller. None of the volumes have been hand-segmented, as it is a hard and time consuming process with high variability.

The Figure 1.3 depicts the T_1 and T_2 MRI scan of the subject `f4395`, who is a real volunteer of the CIMBI project. This kind of representation is the usual 2D way of representing 3D image data with the three orthogonal planes (coronal, sagittal and transverse).

The MR images are based on received intensity, thus the visual representation is a grey scale volume, where brighter voxels are associated with a larger intensity values. The subfigures of the top row represent the T_1 scans. In a wide sense, it can be stated that the dark voxels correspond to fluid-based tissues like GM, and bright voxels to fat-based tissues like WM. The CSF is basically water, thus it appears almost black. On the other hand, the bottom row represents the T_2 scans, and the intensity associations in this case are in general the opposite than for the previously presented T_1 scan. It can be seen in the figure that black stripes have been added to give a regular cubic shape to the 3D scans with final dimensions of $256 \times 256 \times 256$ voxels. For the case of T_2 MRI, the head is not perfectly centered, and the back part of the head appears in the left of the image for the sagittal plane, and in the top part of the image for the transverse plane. This error is due to inhomogeneities of the magnetic field that are not corrected by a shimming calibration.

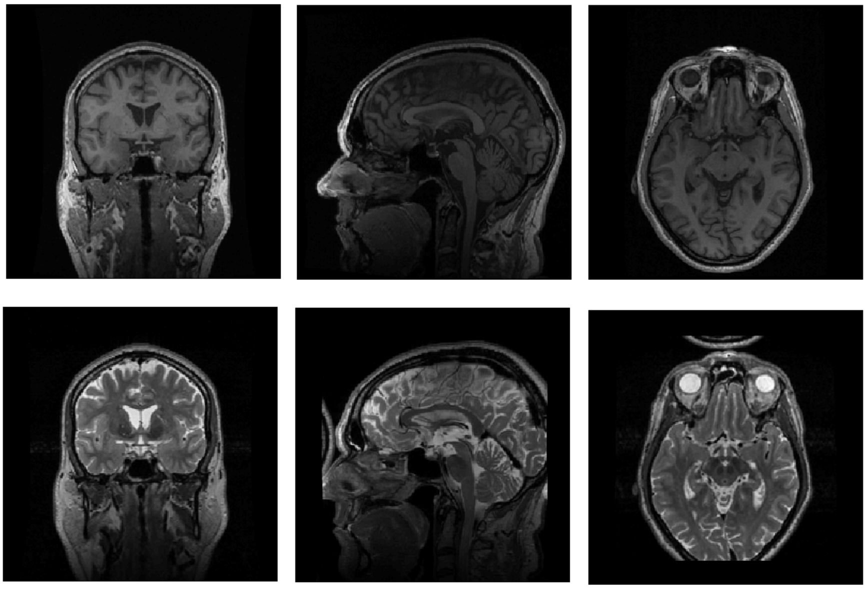


Figure 1.3: Preview of some slices of the T_1 and T_2 MRI data from subject `f4395`. Left column: Coronal plane. Middle column: Sagittal plane. Right column: Transverse plane. The first row corresponds to the T_1 scan, and the second to T_2.

1.3 Baseline

The baseline of this project corresponds to the original pipeline for the MRI brain segmentation, which is based on the 'Unified Segmentation' method developed by J. Ashburner, K. Friston et al. [26]. This method is implemented in the Matlab software of Statistical Parametric Mapping (SPM) [60]. It combines in the same generative model the classification, bias field correction and template registration. In fact, the segmentation itself is done by fitting the mixture parameters of a Mixture of Gaussians (MoG) model, where each cluster is modeled by a Gaussian. Therefore, the tissue segmentation is done by an unsupervised clustering technique.

The segmented volumes that DRCMR provides to the NRU are processed by the SPM5 software plus the voxel-based morphometry (VBM)5 toolbox. However, at the DRCMR, they are working for other projects with SPM8 plus the toolboxes VBM8 and *template'o'matic*, both from the *Structural Brain Imaging Group* at the University of Jena [52]. According to the NRU, the reason for not using the last version of the software lies on the fact that not clear improvements of the new versions have been stated that justifies the migration of all the previous segmented images into a new pipeline. However, it is now intended to do this update, thus the starting point for further improvements is SPM8.

In the original pipeline, the T_2 volumes are used to generate a binary mask of brain voxels. This mask is used in the scalp-stripping step to hide T_1 voxels that correspond to air, skin, fat, muscle, bone or meninges. After this brain tissue extraction, it is done the segmentation itself on the T_1 volumes, where a certain probability of being GM, WM, and CSF is assigned to the voxels inside of the brain to generate the Tissue Probability Map (TPM)‘s.

The Figure 1.4 depicts the main steps of the segmentation procedure as described previously. The top row corresponds to the original T_1 scan. The second row presents the brain mask extracted from T_2 data as a red overlapping layer on the T_1 original slices. For the transverse plane, it can be seen how the mask wrongly classifies part of the left eye muscle as brain tissue, namely as CSF. The bottom row corresponds to the voxels after the scalp-stripping, which are coloured according to their associated tissue class, where GM is in purple, WM in turquoise and CSF in beige.

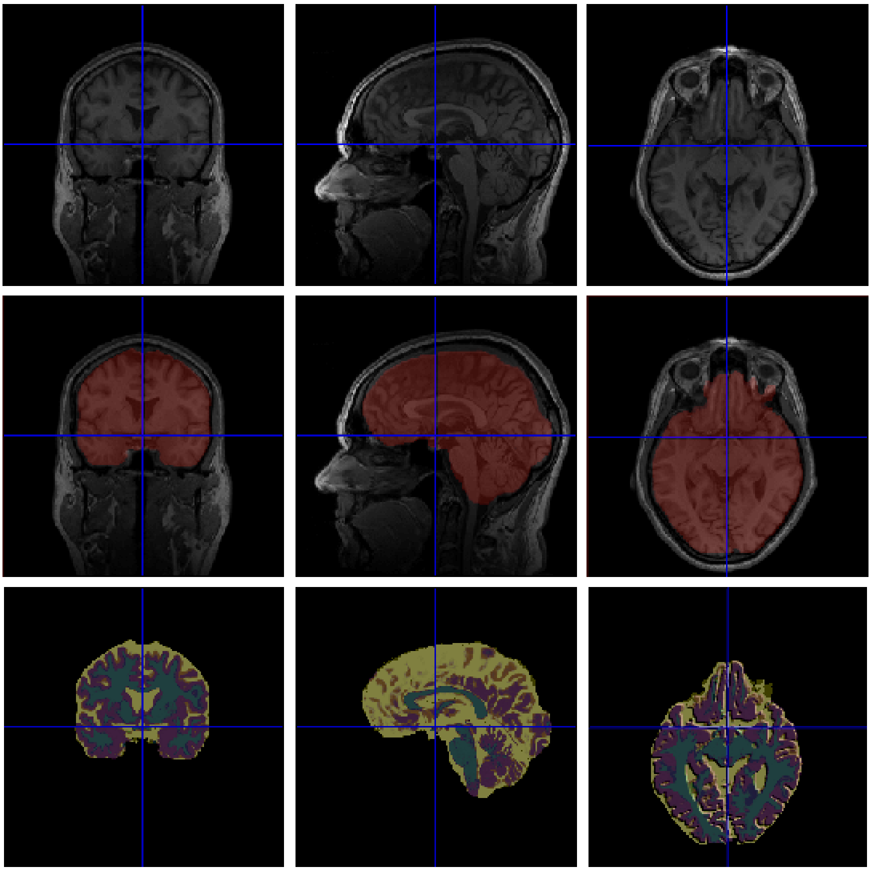


Figure 1.4: Representation of the three main steps of MRI brain segmentation done by the original pipeline, which consists on a T_2 masking and SPM5+VBM5 applied on the T_1 modality. The presented data correspond to subject `f4395`. Left column: Coronal plane. Middle column: Sagittal plane. Right column: Transverse plane. The top row corresponds to the original T_1 scan. The second row presents the brain mask extracted from T_2 data as a red overlapping layer on the T_1 original slices. The bottom row corresponds to the brain tissue generated by SPM5+VBM5, where GM is in purple, WM in turquoise and CSF in beige.

1.4 Project goal

It seems that there is still space for the improvement of the actual MRI brain segmentation pipeline. Thus, it is the scope of this project to analyze and implement a feasible enhancement of the segmentation baseline based on the available data and the start-of-the-art algorithms.

1.5 Specifications

Several meetings and discussions were needed to give shape of a specific project description. It was needed to take into account what is feasible to do in the available time according to the requirements of all involved entities. In this sense, it must be appreciated the technical advices received from DTU, NRU and DRCMR supervisors. Finally, it was agreed on several points that could be improved during this thesis:

- **Multispectral segmentation**. The available dataset includes T_1 and T_2 MRI scans. Therefore, both modalities can be combined in the segmentation process, where T_2 is not used just for masking. The tissues generally have different intensity contrast in each modality. Therefore, the use of both of them can increase the discrimination between different tissues.
- **Increase the number of tissues**. The current segmentation is based on 4 labels, i.e. GM, WM, CSF and rest. Several authors have proposed to include more tissues in order to do a more realistic and robust characterization of the head tissues.
- **Increase the number of clusters per tissue**. The original baseline characterizes each tissue with one cluster. Therefore, this number can be increased in order to fit better the intensity distribution of each class.

During the development of this thesis, it was discovered that the *Seg* toolbox ('New Segmentation') in SPM8 has already implemented these three improvements. However, the multispectral implementation of this method assumes non-correlation among modalities. Therefore, the **goal** of this project is the modification of the *Seg* toolbox in order that the method deals with correlated modalities. Therefore, the *baseline* is the *Seg* toolbox of SPM8, and the *validation* is based on the visual inspection of the generated TPM, the Dice score after the segmentation of brain phantoms and the estimation of a volume age-profile for each tissue.

1.6 Thesis Outline

The first chapter of this report corresponds to the **Introduction**, where it is presented an overview of the thesis motivation, the available dataset, the segmentation baseline and the goals.

The second chapter **Background** describes the brain anatomy, the MRI acquisition technique and different automatic segmentation methods, with special focus on SPM.

The third chapter **Neuroimaging** includes a detailed description of the image processing steps done during the segmentation, namely clustering, registration, bias field correction, brain extraction and smoothing. In addition, it is also described the applied intensity model, and the advantages of the multispectral approach and the use of prior templates.

The fourth chapter **Method & Implementation** presents the mathematical equations that have been modified to account for the correlation among modalities. Besides, it is also detailed how the modified equations are implemented in a Matlab toolbox for SPM8.

The fifth chapter **Validation** analyzes the segmentation performance of the different versions of the modified method with different dataset. Besides, the modified method is compared to the original baseline (SPM5+VBM5) and the original method (*Seg* toolbox of SPM8).

The sixth chapter **Conclusions** discusses the main features and results of the modified method, and proposes future ideas to improve the results.

Finally, the **Appendices** includes further documentation of the concepts presented in the core report.

Chapter 2

Background

The understanding of the brain is one of the last frontiers of the human knowledge. Neuroscience has been studying during decades the nervous system, and specially the brain, which controls the rest of the body and where its neurons and synapses encode the mind itself.

Science and technology have helped the neurology medicine since the first X-ray image in 1895 [35]. Since then, several advances in physics, mathematics and electronic have allowed to look from an enriched point of view the structures and operations of our own brain. All this neuroimaging development has transformed the diagnosis and treatment of neurological and neurosurgical disorders [21].

This chapter includes a brief description of the brain structures and its mains parts. Latterly, the MRI image scan technique is revised beside the acquisition protocol.

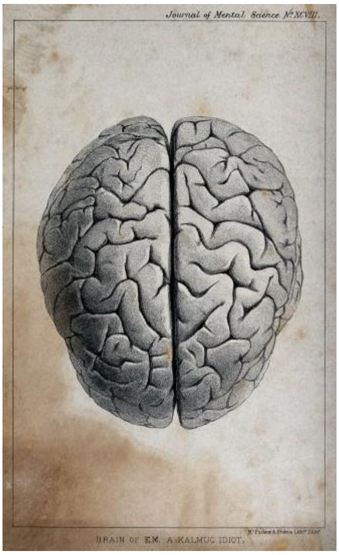

Journal of Mental Science Nº XCVIII.
BRAIN OF E.M. A KALMUC IDIOT.

Figure 2.1: *Brain of someone described as an idiot* by George Edward Shuttleworth [Wellcome Images$^{(CC)}$ [59]]

2.1 Brain Anatomy

This section describes the main features of the human brain, with special focus on the anatomy, in order to establish a connection between what is seen in the MRI scans and the real associated structure. The presented data are a compendium from [46] [63] [74].

The brain is composed by more than 100 billions of neurons and it is the centre of the Central Nervous System (CNS), where all the nervous connections merge. It is placed inside of the head and fills most of its volume, which is around 1450 cm^3 on average for human adults. Under the skin, fat, muscles and scalp, the *meninges* are the last protection of the brain. They are composed by three layers: *dura mater*, *arachnoid mater*, and *pia mater*. The brain is composed by four main structures: *cerebrum*, *diencephalon*, *brain stem* and *cerebellum*, which are depicted in the Figure 2.2 with colors red, violet, blue and green.

The **cerebrum** is the biggest part of the brain. It is approximately symmetrical, with two hemispheres, left and right. Each hemisphere can be divided into four lobes depending on which part of the scalp covers it, namely, the names are: *frontal*, *parietal*, *occipital* and *temporal*. It includes the *cerebral cortex*, *basal ganglia* and *limbic system*. In a wide sense, the cortex is considered a *cortical structure*, and the basal ganglia and limbic system are *subcortical structures*.

- The cerebral cortex is depicted in pink in the Figure 2.2, its external layer is the *neocortex*, which is composed by Grey Matter (GM) and contains most of the nerve cells. The surface is folded into *sulci* and *gyri* that give its classical wrinkled appearance, and increases its outermost surface, called *pial surface*. The formed intra-cerebral ventricles are filled by the CerebroSpinal Fluid (CSF), which is mainly water that protects the cortex. Under the neocortex but still inside of the cortex, it can be found the White Matter (WM), which connects the nerve cells of the cortex to other parts of the CNS with nerve fibers. Besides, it allows the connection between both hemispheres through the *corpus callosum*.

- The basal ganglia is a subcortical structure depicted in orange in the Figure 2.2. It comprises mainly the *striatum*, which is composed by *caudate*, *putamen* and *pallidum*.

- The limbic system is another subcortical structure presented in dark blue in the Figure 2.2. It includes the *hippocampus*, *amygdala*, and others.

The **diencephalon** (in violet) includes the *thalamus*, *hypothalamus*, *subthalamus*, and *epithalamus*. The thalamus fills around the 80% of this part and it is composed by GM. The **brain stem** (in blue) connects the brain to the spinal cord. It can be split up into three parts: *midbrain*, *pons*, and *medulla oblongata*. Its main tissue is WM. Finally, the **cerebellum** (in green) corresponds to a separate and striped strucutre at the bottom of the brain. Its inner part contains WM, and its thin cortex is composed by GM.

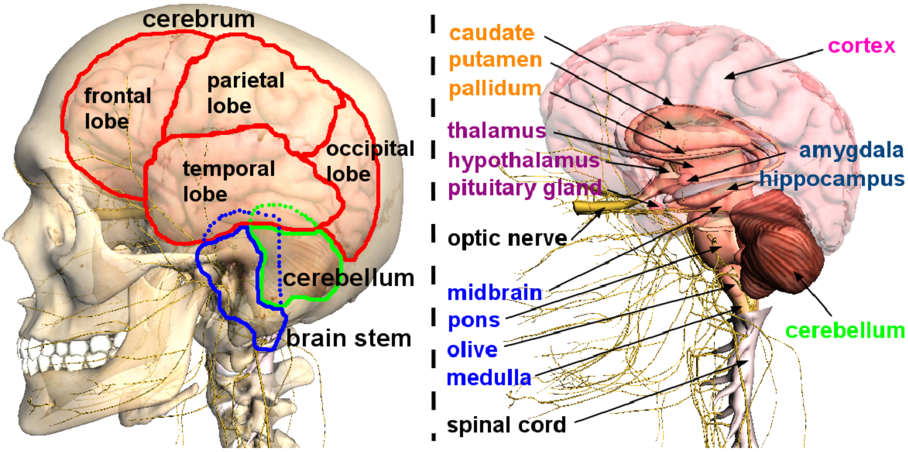


Figure 2.2: Human brain representation where the main anatomical structures are highlighted. The four main parts of the brain are presented: cerebrum (red), diencephalon (violet), brain stem (blue) and cerebellum (green). Besides, the cortical and subcortical structures of the cerebrum are also presented: cerebral cortex (pink), basal ganglia (orange) and limbic system (dark blue). [3D brain images generated by Google Body [61].]

The goal of this project is the segmentation of White Matter (WM) and Grey Matter (GM). The WM has a high content of fat, and the GM contains more water. In turn, the CerebroSpinal Fluid (CSF) is mostly composed by water. The different composition of these tissues gives a contrast in the MR scans that permits its differentiation. This phenomenon is the basement of the segmentation process, and it is presented with more details in the next section.

In this project, it is also analyzed the estimated volumes of several tissues. The IntraCraneal Cavity (ICC) corresponds to the matter inside of the scalp. Its volume is around 1700 cm^3, where the brain fills the 80%, the blood a 10% and the CSF another 10%. The brain is composed by the cerebrum in a 77%, cerebellum in a 10%, diencephalon in a 4%, and brain stem in a 9%.

2.2 Magnetic Resonance Imaging

This section explains the connection between the studied brain tissues and the intensity values acquired by the different MR imaging modalities. It is not intended that this section goes deep into the quantum phenomena and the subsequent signal processing. For a more detailed study, it can be consulted the Appendix A that is based on [9] [20] [23] [32].

There are different brain imaging modalities, like Magnetic Resonance Image (MRI), Positron Emission Tomography (PET), Diffusion Tensor Imaging (DTI), Computed Tomography (CT) and Single Photon Emission Computed Tomography (SPECT).

The MRI was mainly developed around 1980 as an application of the already studied phenomenon of Nuclear Magnetic Resonance (NMR), which leaded to several Nobel prizes. It applies static and variant magnetic fields to make resonate the molecules of the body. The effect of stopping the variant magnetic field generates signals that can be measured by a conductive field coil around the body and processed to obtain a 3D grey-scale image. The intensity, recovering time and frequency of the molecular vibrations determines the acquired intensity pattern.

This imaging technique allows to focus on the detection of different molecules by using different resonance frequencies.

- For example, if the scanner is adjusted to the Hydrogen's nuclei ${}^{1}_{1}H$, the result is a representation of the tissues depending on the level of fluid density. This effect is due to the high number of either free or bounded water molecules H_2O in the body. Although water is the most abundant, the Hydrogen's nuclei can be also found in the body as fat CH_2. This technique is known as structural Magnetic Resonance Image (sMRI) or simply MRI, because it represents the anatomy of the tissue structures.
- Likewise, the resonance frequency of the scanner can be fixed to detect the position in the brain of the Oxygen's nuclei ${}^{16}_{8}O$. The brain consumes more oxygen when is working, so several scans can detect the variation of the oxygen density along the time for each voxel, which encodes the variation of brain activity in each part of the brain. Therefore, it can be temporally correlated brain activity and location. This kind of MRI that focuses on the metabolism is known as functional Magnetic Resonance Image (fMRI).

Advantages

The MRI technique has several advantages compared to other neuroimaging techniques. For example, it is fast and it does not use ionizing radiation; therefore, it can be used several times on the patients because the absorbed radiation is minimal. Its isotropic resolution is around 1 mm^3 with 3T MRI scanners, which outperforms the 8 mm^3 of PET. It has a high versatility, because it can be used to study structural and functional brain features with different configurations. Besides, it is not affected by the hardening beam effect of CT [5] because the range of frequencies is small, and the attenuation coefficient of the tissues is almost homogeneous.

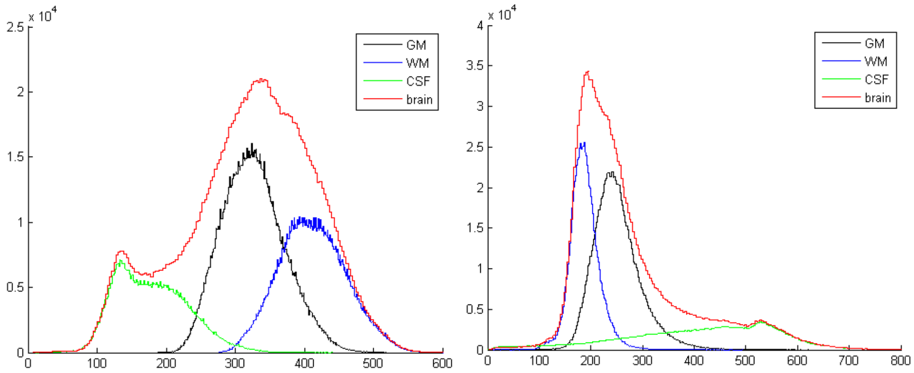
Disadvantages

On the other hand, it is an expensive and complex technique. There are many parameters that must be tuned up correctly in order to optimize the image acquisition depending on the circumstances [72]. In addition, all the metal objects of the patients should be removed before the scanning starts, which is impossible for some kind of surgical implants. Besides, this technique is only suited to analyse soft tissues because the bones have not a significant contrast in the images.

2.2.1 Relation between intensity and tissue

In the sections A.1 and A.2 of the Appendices, it is explained in details the relation between the acquired intensities by different modalities and the kind of tissue in the body. In short, it can be stated that the **$\mathbf{T_1}$ MR images have brighter voxels for WM, darker for GM, and almost black for CSF**. The T_1 images show a tumour with larger intensity value than a normal tissue. Therefore, some lesions in the WM areas can look alike GM in T_1 images due to the increase of water. Besides, the voxels with muscle tissue appear brighter than for fat. Almost the opposite intensity contrast will be expected in T_2 images. However, the exact intensity value for each tissue slightly vary depending on which part of the brain is located.

In the Figure 2.3 it is depicted the intensity histogram of the T_1 and T_2 MRI data from the subject `f4395`. The GM, WM and CSF tissues have been segmented by the original baseline. The T_1 seems to have the classes more differentiated than T_2, which facilitates the segmentation. The intensities correspond to the original volumes, which implies that the bias field correction is not taken into account.



(a) Histogram of T1 intensity.

(b) Histogram of T2 intensity.

Figure 2.3: Intensity histogram of the segmentation for the subject `f4395` using T_1 and T_2 MRI. The black line corresponds to the GM, the blue one to the WM, the green line to the CSF, and the red line to the total brain. The units of the x-axis correspond to intensity values, and the y-axis to the number of voxels for each intensity bin. All the histograms are built with 300 bins.

Here, it can be demonstrated the relation between intensities and tissues that was previously presented. The WM is less fluid-based, thus the voxels with mostly this tissue class will appear white in T_1 and dark in T_2, which corresponds to high and low intensity values, respectively. In the case of GM, it appears darker in T_1, and brighter in the T_2 images. Finally, the CSF shows a small peak but also a big lobe that almost overlap all the classes.

2.2.2 File Format

The MR images from the DRCMR are stored in a *NIfTI-1* file format created by the Neuroimaging Informatics Technology Initiative (NIfTI) [56], which is the most spread standard. It allows several coordinate systems -like Montreal Neurological Institute (MNI) space (MNI-152) or Talairach-Tournoux-, two affine coordinate definitions -orthogonal transform with 6 parameters or general affine transform with 12 parameters-, single (Nifti) or dual (Analize) file storage (.nii or .hdr/.img), affine data scaling -$truevalue = \alpha \cdot datavalue + \beta$-, several units of spatio-temporal dimensions, and others.

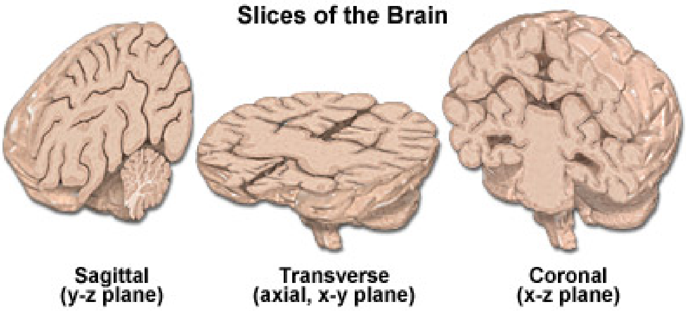


Figure 2.4: Representation of the brain slices format for the sagittal, transverse and coronal planes.

The usual presentation of MR images correspond to the three planes: coronal, sagittal and transverse; as presented in the T_1 scan of Figure 2.5. The correspondence between the presented images and the spatial planes that cut the brain is represented in the Figure 2.4.

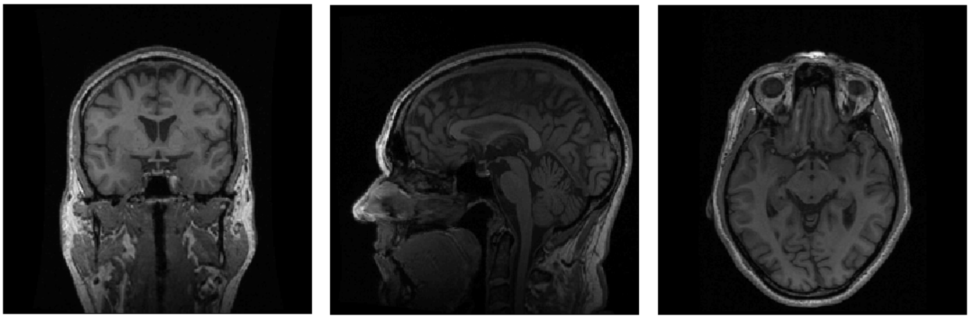


Figure 2.5: Preview of the T_1 MRI data from subject `f4395`. The presented planes correspond to coronal, sagittal and transverse.

2.3 Segmentation

The segmentation of the brain stands for its decomposition into different volumes with similar structural or functional features. In the case of structural MRI brain segmentation, the available data corresponds to a 3D map of voxels, which are the analogous of pixels in a 2D map. These voxels are grouped according to quantitative characteristics like intensity, colour or texture [72]; which implies that after the segmentation process, each voxel has an associated label explaining to which group it belongs to. The usual labels are WM, GM and CSF. The scalp, fat, skin, muscles, eyes and bones are preferable removed in a previous step or modeled with a mask during a process called *scalp stripping*, which will be briefly explained in the section 3.4.

The Figure 2.6 represents graphically the brain segmentation result on a transverse slice of MRI T_1. On the left, it appears the acquired image after pre-processing with darker colour for fluid-composed tissues (GM) and brighter colour for fat-based tissues (WM). On the right, it appears the estimated 2D segmentation represented with three colours, where each tone represents each label. In this way, red, green and blue stands for GM, WM and CSF, respectively.

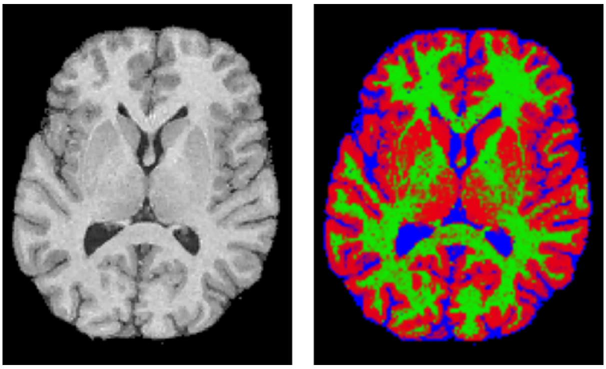


Figure 2.6: Figure with a brain segmentation of a T_1 MR image. Left: Descalped MR image. Right: Segmented image. GM in red, WM in green and CSF in blue. [Courtesy of Koen Van Leemput]

The segmentation of MR images has several and critical applications [11]. For individual subjects, it is used for quantitative image analysis, for example volume/surfaces/edges estimation or visualization of the neuroanatomy. It is also used for image guided therapy, which includes surgical and radiotherapy planning [36]. When the study is applied to groups of subjects, the purpose is usually to generate statistical atlases that encode the probability of finding each tissue at each spatial location.

The MRI segmentation can be performed by different algorithms that are based on a wide range of principles. The segmentation process can be accomplished with different levels of manual interaction. In case of a high manual interaction, the process is time consuming with high associated cost, as it is needed an import amount of time by well-trained professional to accomplish this task [72]. In addition, it introduces a high intra-subject and inter-subject variability due to the personal subjectivity [85], which reaches discrepancies higher than 20% [37]. On the other hand, highly automated methods require a deeper understanding of complex physical processes and mathematic modelling. This challenging approach tries to create a robust, objective and cost-time saving segmentation system.

2.3.1 Automatic Segmentation Methods

Image segmentation techniques have been applied to different fields apart from medical imaging, like hand-written character recognition, people/objects tracking, biometric systems, automated driving, etc. In addition, other medical imaging segmentation techniques have been largely studied for SPECT, PET, CT, X-ray. Therefore, there are lot of active research lines that are focused in the improvement of the image segmentation. Therefore, these ideas can be used as inspiration for MRI segmentation.

A preferable method must be fast, robust, and mostly automatic. Nowadays, there are several kinds of MRI segmentation techniques that apply different methods or combination of them. For example, they can be based on thresholding [78], clustering [10], watershed [71] [30], snakes [4], histogram [69], Finite mixture (FM) [45], Support Vector Machine (SVM) [27], Artificial Neural Network (ANN) [84], or Hidden Markov Model (HMM) [91].

Classical literature distinguishes between data-driven and model-driven methods, although the border is not always clear. Data-driven uses just the data to let it 'explain itself', which makes it flexible and not biased, although sensitive to the noise [34]. And model-driven methods assume that the structures of interest have a repetitive shape; and thus, a probabilistic model can be created to explain its variations. This process comprises the registration of the images into a common space, the probabilistic representation of the variations and the statistical inference. In other words, it tries to find the parameters that fit the model according the the data. The studied literature for this project focus on the last one, specially in the technique based on Gaussian Mixture Models (GMM) done by SPM. A result of the segmentation done by the original baseline (SPM5 + VBM5) is presented in the Figure 2.7 for the GM, WM and CSF tissues.

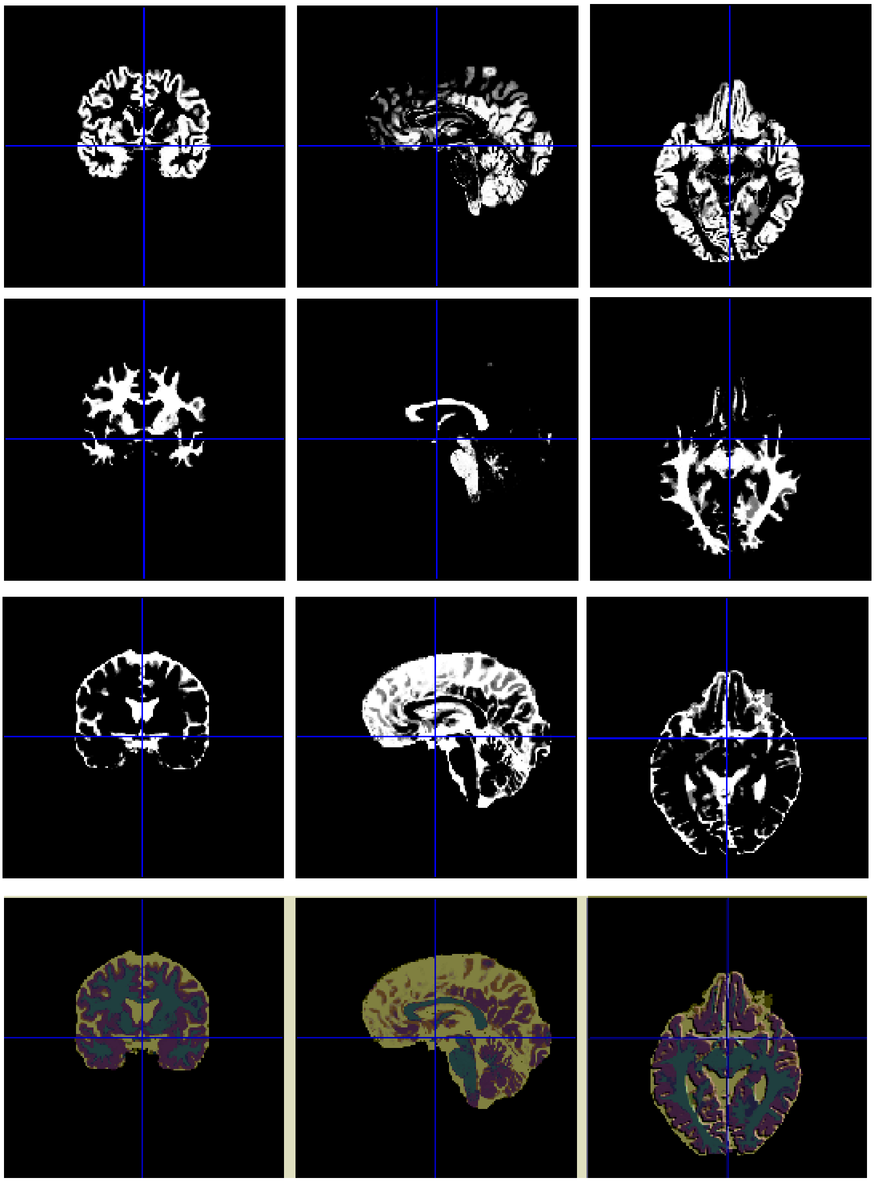


Figure 2.7: Segmentation of a T_1 MRI volumes from subject `f4395` done by the original baseline. First, second and third columns correspond to coronal, sagittal, and transverse planes, respectively. The first three rows correspond to the GM, WM and CSF tissue classes, respectively. The last line is the overlapped labels, where GM is in purple, WM in turquoise and CSF in beig.

2.3.2 Software Implementation

There are several free available tools to perform automatic MRI brain segmentation. The most popular is SPM, which is based on the 'Unified Segmentation' method. It uses a voxel-based approach with a statistical inference on the GMM. This Matlab software is developed from the theory of K. Friston and J. Ashburner from the University College of London [26] [26] [24]. As stated previously, this software/method is the baseline for this project. This implementation has several extensions, one of them is the Expectation Maximization Segmentation (EMS) created by Koen Van Leemput [82] [47] [83]. This SPM extension is a model-based automated segmentation with Markov Random Field (MRF) as regularization that uses multispectral data to improve accuracy of lesions segmentation. The VBM toolbox from the University of Jena applies a modulation to include spatial constraints in the tissue volumes. Besides, it can work without prior templates by using Maximum A Posteriori (MAP) techniques. It also includes DARTEL normalization and PVE estimation.

FMRIB Automated Segmentation Tool (FAST)/FMRIB Software Library (FSL) is a library developed by the *Analysis Group of Functional Magnetic Resonance Imaging of the Brain (FMRIB)* in the Oxford University [76] [90]. The 3D segmentation and the inhomogeneity correction is done with a method based on a Hidden Markov Random Field (HMRF) model and an Expectation Maximization (EM) algorithm. In addition, FreeSurfer is another important segmentation tool that is compatible with FSL and developed by the *Martinos Center for Biomedical Imaging* in the Massachusetts General Hospital [8] [16].

Klauschen et al. [37] compared FSL, SPM5 and FreeSurfer with the same images from the BrainWeb MRI database [55] in terms of GM and WM volumes. In general, the three methods had a deviation up to >10% from the reference values of gray and white matter. The best sensitivity corresponds to SPM. The volumetric accuracy was similar in SPM5 and FSL, but better than for FreeSurfer. The robustness against changes of image quality was also tested, and FSL showed the highest stability for white (<5%), while FreeSurfer (6.2%) scored the best for gray matter.

Although the previously mentioned software package are the most well-known, there are several more available. However, there will not be further discussion about the methods in the rest of this thesis because of two main reasons. First, because because the scope of this project and its goals are oriented on an improvement of its current baseline with SPM, and not a comparison among methods. Second, because the task of comparing methods is tough. It requires high rigour, with a validation and a dataset equally fair for all the methods, and with an implementation done with a deep knowledge and understanding of the algorithms.

2.3.3 Statistical Parametric Mapping

The 'Unified Segmentation' method of J. Ashburner and K. Friston [3] corresponds to the Matlab implementation of SPM, which is distributed under the terms of the General Public Licence (GNU). Although SPM8 includes some minor updates, these few improvements did not justify the upgrading from SPM5 in the past due to the associated migration problems, according to the NRU. Now, the NRU has the intention of updating and improving the segmentation pipeline, therefore SPM8 is the starting point of this project, which will be modified to include multispectral data and a better characterization of the tissue classes (with more tissue classes and more Gaussian clusters per tissue).

In the manual of SPM8, it can be found the sentence: *'Note that multispectral segmentation (e.g. from a registered T1 and T2 image) is not yet implemented, but is planned for a future SPM version'* [Page 44. SPM8 Manual]. However, during the the presentation of this project at the DRCMR, one participant highlighted the existence of an in-built toolbox in SPM, called 'New Segmentation', which already performs multi-spectral. The help file of this toolbox states that the general principles correspond to 'Unified Segmentation' but with some modification, as different deformation parameterization and the use of extended set of probability map and multi-channel data. It also states that the quality of the implementation has not been tested. Besides, the comments in the code say that it is assumed not correlation among modalities. Although this discover modified the initial goals, there is still room in this project for an improvement by including correlation between T_1 and T_2.

Therefore, the actual starting point is the 'New Segmentation' toolbox of SPM8. This choice has the advantage of not wasting time in a costly full implementation, and focus just on improving the state of the art algorithms. The main drawback is the lack of publications or documentation about this toolbox that explain the details about the method and its implementation, thus a reverse engineering must be done from the Matlab code. Therefore, it is needed to understand the method, modify the Matlab implementation, and tune it up for the available dataset. A little help can be the work of N. Weiskopf (Appendix A of [88]), where it is briefly explained some parts of this extension.

The discover of this Matlab extension together with some comments from the scientific community reinforces the chosen multispectral approach as an important improvement of the MRI segmentation.

2.3.3.1 Unified Segmentation

The 'Unified Segmentation' is an unsupervised parametric method for MRI segmentation that combines bias field correction, regularization and classification in the same cost function.

- **Bias field correction** of the smooth and spatially varying intensity inhomogeneities. It is based on a Discrete Cosine Transform (DCT) with low parameterization.
- **Regularization** of templates and MRI volumes. It is also based on a DCT with low parameterization.
- **Classification** of the voxel into different tissue classes. The Bayesian framework permits to include templates as a priors. These priors correspond to the Tissue Probability Atlases from the International Consortium for Brain Mapping (ICBM).

Thus, the iterative process optimizes locally each of the three group of parameters until convergence. A detailed description of these steps are included in the two following sections.

2.3.3.2 New Segmentation

The 'New Segmentation' is an extension of the 'Unified Segmentation'. It is implemented as a Matlab toolbox for SPM under the name *Seg*, and it can be found in the Batch options of SPM8. The main modifications from the original method are:

- Possibility of multi-spectral segmentation, where it is assumed non-correlation among modalities.
- Extended prior template set, with TPM for Bone and Soft Tissue (ST).
- Different initial affine registration
- Different treatment of the mixing proportions
- Different registration model and deformation parametrization

It is not the intention of the thesis to modify the bias field correction and registration. However, it is needed to understand how they work due to the high coupling with the classification step.

Chapter 3

Neuroimaging Processing

This chapter includes the processing done after the acquisition of the MR images. Although, this project focuses on the brain segmentation of SPM, there are other steps in the pipeline that should be mentioned and understood. Some of them improve slightly the result, but others are strictly needed. Each segmentation method uses different layouts, different order of the blocks or different algorithms. In the case of SPM some steps are even done iteratively [26] [83].

Each section of this chapter presents the definition of a different processing step and several possible implementations of the same are discussed. Finally, it is explained how SPM implements this step, and it is presented one example with real MRI data.

The first section of **Intensity Model** describes the MoG model and justifies several improvements from the baseline of 'Unified Segmentation', like the inclusion of more tissues and more clusters per class, or the multispectral approach with several modalities. In the section **Registration** and **Bias Field Correction**, it is explained how the templates are spatially aligned to the raw volumes and how the intensities inhomogeneities are corrected.

Finally, the last three sections include the results of the **Scalp Stripping**, the effects of the **Smoothing** and the main features of **Priors and Templates**.

3.1 Intensity model

The 'Unified segmentation' of SPM is based on a generative model of the intensity patterns from the brain MRI volumes. A Generative Model (c.f. discriminative models [7]) estimates the distribution of the posterior $P(\theta \mid \mathbf{Y}, M)$ and marginal probabilities $P(\theta \mid M)$ to compute the joint probability $P(\mathbf{Y} \mid \theta, M)$ by using the Bayes' rule [6]. This sort of modeling needs a deep understanding of the brain and neuroimaging processing. In a simple statement, it could be affirmed that *if one volume of the brain is composed by grey matter, the generative model could predict the intensity distribution of the corresponding voxel/voxels in the acquired image.* The main drawback of this modeling approach is that the assumed probability distribution of the variables could not fit the reality. Another minor disadvantage is the increase in the number of uncertainties, because higher number of model parameters (latent variables) implies a more complex model implementation and longer computation time. However, this point is outweighed by the increase in accuracy.

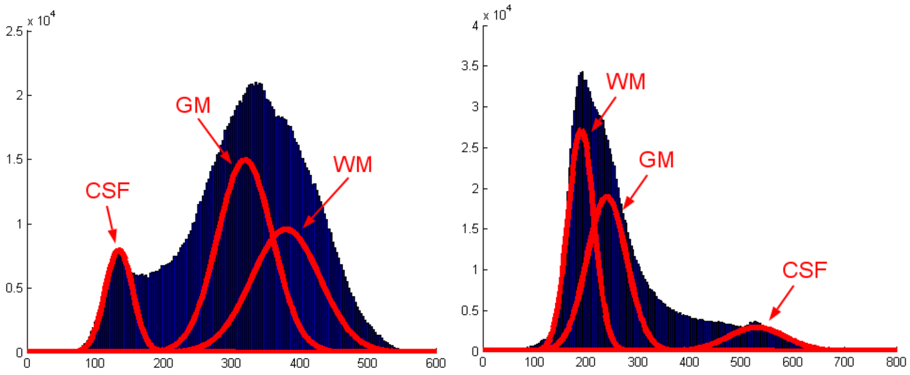
Namely, the generative model used in SPM corresponds to a Mixture of Gaussians (MoG) or Gaussian Mixture Models (GMM), where each cluster is modeled with a Normal (Gaussian) distribution. A multi-dimensional normal distribution is parameterized by the intensity mean vector μ, and the intensity covariance matrix $\mathbf{\Sigma}$. The assumption of normal distribution restricts the shape of the intensity distributions to a Gaussian bell. This restriction implies a small number of parameters, but it also means a reduction of the degrees of freedom. Therefore, there is trade-off between computation time and distribution flexibility. In the Appendix B.1, it is presented a more detailed explanation of a Gaussian distribution and its properties.

The aggregated intensity distribution is a Linear Combination (LC) of each tissue distribution. Hence, it is needed another parameter that weights each Gaussian contribution. This task corresponds to the mixing coefficient γ_k, where k stands for the number of cluster. Likewise, the scaling factor is directly proportional to the number of voxels that belong to each class. For example, a $256 \times 256 \times 256$ MR scan from the subject `f4395` is segmented by the 'Unified Segmentation' method, which associates one cluster (Gaussian) to each tissue class. The total number of voxels is $I = 16,777,216$, from where $1,246,798$ voxels are considered as brain tissue. Besides, $518,104$ voxels are classified as GM, $406,343$ voxels classified as WM, and $322,351$ voxels classified as CSF. Therefore, the corresponding mixing coefficients are $\gamma_{GM} = 0.4155$, $\gamma_{WM} = 0.3259$, and $\gamma_{CSF} = 0.2585$.

In the rest of this section, it is included an analysis of several ways to improve the intensity model of the 'Unified Segmentation' method. Hereafter, the presented histograms are obtained after applying a threshold of 0.9 to the generated TPM.

3.1.1 Several Gaussians per tissue class

In the original case, the number of clusters (Gaussians) per tissue class is one. However, this number can be larger, which implies that the aggregated distribution for each tissue is non-Gaussian, and thus it can fit better the actual intensity distribution of the MRI data. The Figure 3.1 presents the T_1 and T_2 intensity histogram of the segmentation done by the original baseline for the volumes from subject `f4395`. The overlapped red Gaussians approximate the expected intensity distribution of GM, WM and CSF. The number of non-brain voxels is large, specially for the Background (BG) class. Therefore, they are are not represented in the histograms in order to appreciate better the intensity distributions of the brain tissues. The size of each Gaussian depends on the number of voxels classified as the associated tissue class. In this case, the ratio of GM, WM and CSF voxels over the total number of brain voxels is approximately: 40%, 35%, and 25%.



(a) Histogram of T1 intensity.

(b) Histogram of T2 intensity.

Figure 3.1: Intensity histograms of the brain voxels for the subject `f4395` using T_1 and T_2 MRI. It is overlapped three red Gaussian that approximate the expected class distribution of GM, WM and CSF. The units of the y-axis correspond to the number of voxels, and the units of the x-axis are the intensity values. All the histograms are built with 300 bins of the same size. For the T_1 histogram, from the leftmost to the rightmost distribution, the tissues correspond to CSF, GM and WM. Likewise, the order is inverse for the T_2 histogram.

There is a big overlap among classes, which means that one voxel is not purely composed by one single tissue. Due to the PVE, some voxels lie in the interface between two (or more) classes. The resolution of the scanner is finite; thus, the acquired intensity at this point is a mix of the different tissues. In addition, the assumption of each tissue modeled by a Gaussian is not realistic. An increase in the number of clusters makes the distribution non-Gaussian and it can fit better the actual intensity distribution.

Hence, it seems reasonable to increase the number of clusters per tissue, although it implies a more complex mathematic implementation. Besides, it is needed an extended mixing coefficient set because several clusters would share the same TPM template.

Several authors have proposed different numbers of clusters per tissue class. For example, J. Ashburner in the comments of the 'New Segmentation' implementation proposes 2 for grey matter, 2 for white matter, 2 for CSF, 3 for bone, 4 for other soft tissues and 2 for air (background). However in the 'Unified Segmentation' method of J. Ashburner [3], it is proposed 3 for grey matter, 2 for white matter, 2 for CSF, and 5 for everything else. The lack of agreement in the numbers can be explained if it is analyzed the differences of the datasets used in each work. In other words, each MRI dataset depends on the the specific acquisition protocol, the quality of the scanner, and the scanned subject cohort. Therefore, each dataset is characterized by different parameters. Although, the suggested numbers can be used as an approximation, the only way to optimize these numbers is by empirical exploration on the available data.

3.1.2 Extended template set

More classes can be incorporated in order to model better other human tissues, like bones, muscles, scalp, air... However, a distinction must be done here among parcellation and segmentation methods. The former is intended to localize areas of the brain, e.g. thalamus, hyppocampus, cerebral cortex, amygdala, etc. On the other hand, the latter tries to detect the composition of each voxel in terms of GM, WM and CSF.

For example, T. Tasdizen [80] suggests 9 tissue classes, namely gray matter, white matter, cerebrospinal fluid, blood vessels and sinuses, eyes, bone, bone marrow, muscle, and fat tissue. However, the number of classes used is also constrained by the available segmented templates. Likewise, if there is not previously segmented images for one kind of class, there is not prior probability map that can be incorporated in the method. In this thesis, it is used the set of templates included in the *Seg* toolbox of SPM8, which corresponds to the 'New Segmentation' method. It comprises 6 tissue classes, namely: GM, WM, CSF, bone, ST and BG, which is mainly composed by hair and air. The TPM's are generated from 471 brains, with dimensions $121 \times 145 \times 121$ and $1.5mm$ isotropic voxel resolution. The main difference from the previous template set of 'Unified Segmentation' is the inclusion of tissue classes for ST and BG.

The Figure 3.2 present some slices of the bone and soft tissue in the brain. Besides, the Figure 3.3 depicts the histogram for T_1 and T_2 of the brain tissues plus an additional class that accounts for Bone and ST intensities. It can be seen the important overlap between non-brain and brain voxels in the head.

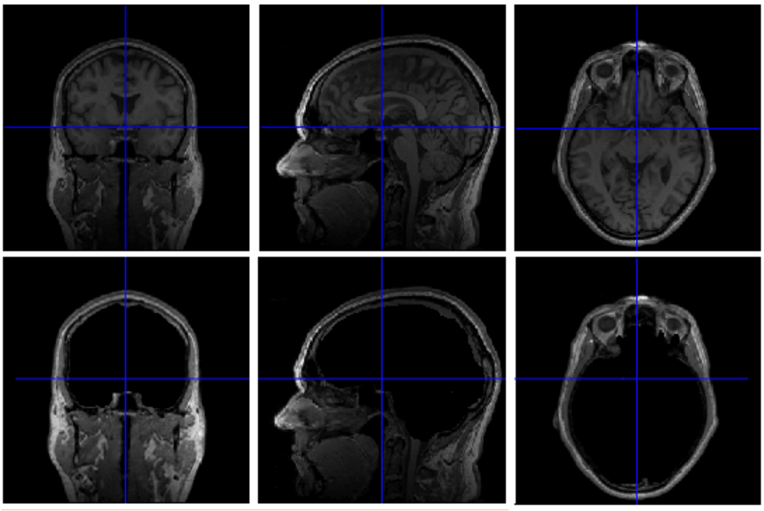
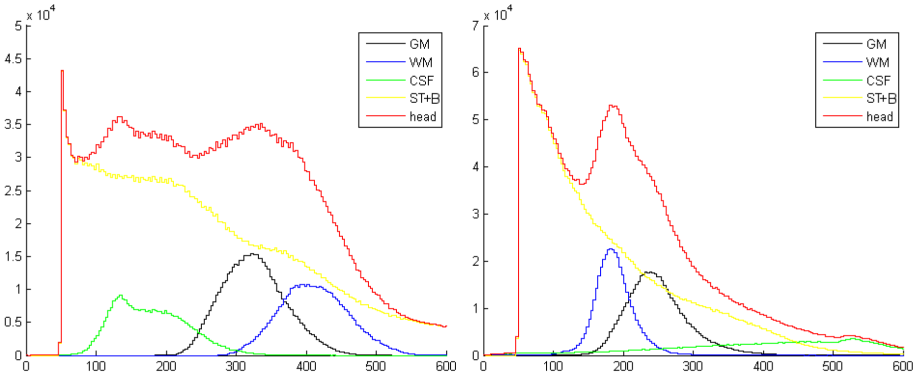


Figure 3.2: Slices of the T_1 MRI scan from the subject `f4395`. The top row contains head tissues, and the bottom row shows just Bone and ST.



(a) Histogram of T1 intensity.

(b) Histogram of T2 intensity.

Figure 3.3: Intensity histogram of the head voxels for the subject `f4395` using T_1 and T_2 MRI. The black line corresponds to the GM, the blue one to the WM, the green line to the CSF, the yellow one to the ST+Bone, and the red line to the head voxels. The units of the x-axis correspond to intensity values, and the y-axis is the number of voxels for each intensity bin. All the histograms are built with 300 bins. The voxels with intensity values lower than 50 are dismisses, as they can be considered BG.

3.1.3 Multispectral

In the previous sections, it was presented some histograms that showed the important overlap between classes. In fact, the overlap between GM and WM is higher than 10% for T_1 [22]. Therefore, a segmentation method cannot be just based on the intensity distribution from one modality. One way to solve this problem is with the use of priors that give spatial information about where is more feasible to find each tissue. Another improvement is the combination of several modalities with different intensity contrasts that increases the dimensionality of the clustering and makes more feasible the discrimination among several classes. For example, the multispectral approaches are better in the detection of the WM lesions, where the uni-modal methods missclassify them as GM. The 'New Segmentation' method already includes prior templates and a basic multispectral approach. However, the algorithm assumes non-correlation among modalities, which will be modified in this project.

Therefore, the multispectral approach stands for the use of several imaging techniques from the same anatomical structures. In order to gain something, it is needed that the tissues have different responses to the MR pulse frequencies, i.e. different intensity contrast from T_1 than for T_2. This constraint also imposes the modifications of the algorithms that are based on intensity similarities, because the intensity patterns between both modalities are different.

For example, the Figure 3.4 presents the combined 2D histogram for the GM, WM and CSF tissue. Due to the thresholding processing of the TPM's, there only a small overlap among classes. Besides, the shape of the intensity distributions on the presented histogram depends on the the applied segmentation method. In other words, if the method tries to fit the intensity distribution of each class with 5 Gaussians, it would more possible to see 5 groups per class. In this case, the histograms are obtained from the segmentation done by the original baseline, which uses T_1 for segmentation, T_2 for scalp-stripping, and one cluster per tissue.

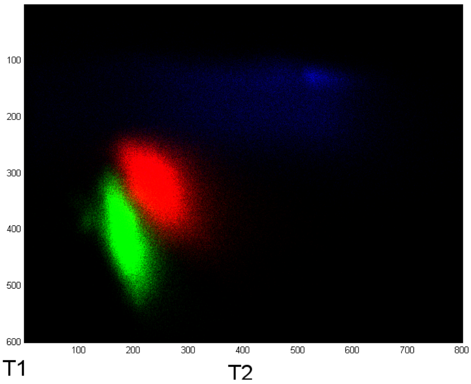


Figure 3.4: 2D intensity histogram for T_1 and T_2 MRI. The red cloud corresponds to GM, the green one to WM, and the blue one to CSF.

The Figure 3.5 depicts the joint 2D intensity histogram for T_1 and T_2 with the associated individual histograms, T_1 on the left and T_2 on the top. In the individual histograms, it is overlapped three red Gaussian that approximates the expected class distribution of GM, WM and CSF. It can be seen that the increase of dimensionality by adding T_2 allows a better separation of classes. Hence, the fully multispectral approach that is developed in this thesis seems a good improvement of MRI segmentation.

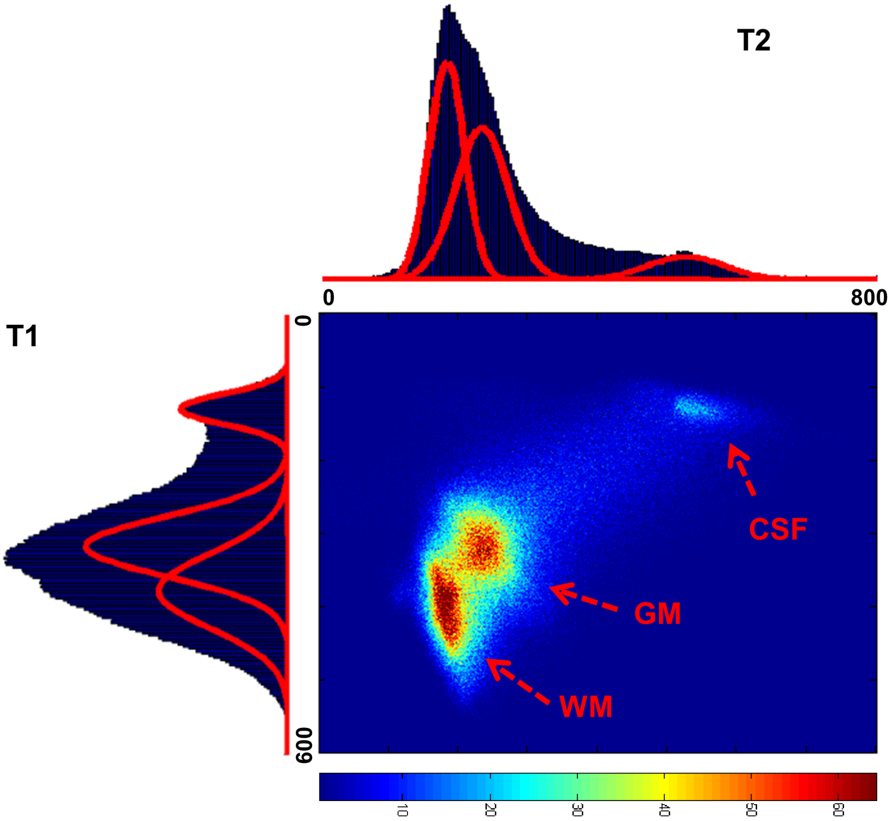


Figure 3.5: Joint 2D intensity histogram for the segmentation of the MRI scans from the subject `f4395`, which is done by the original baseline. On the edges, it is presented the associated 1D histograms of each modality, T_1 on the left and T_2 on the top. In the individual histograms, it is overlapped three red Gaussian that approximates the expected class distribution of GM, WM and CSF.

3.2 Registration

The brain volumes are represented in a 3D coordinate reference system, where each intensity value is a voxel located using three coordinates (x, y, z). In case the volumes are acquired from different scanners, patients or time epochs, the spatial correspondence of anatomical structures is partially lost. Therefore, it is needed to apply a one-by-one mapping between both coordinate spaces [23].

The term *image registration* refers to the general transformation from two different spaces. There are special cases of registration, like *co-registration* that is used for intra-subjects registrations, *re-alignment* that is used for motion correction within the same subject, and *normalization* that is used for inter-subjects registrations. The latter usually implies the registration to a standard stereotactic space, like MNI or Talairach [25].

SPM applies an affine (12 parameters) and non-linear (~ 1000 parameters) transformation. Both of them are encoded with a reduce number of parameters in order to achieve an overall good shape matching without increasing the complexity of the model. All the cortical structures are not perfectly matched due to the low number of parameters. However, it is impractical to try a perfect match between real brains, as there is no a one-to-one relationship and some structures -like sulcus and gyrus- would need to be created. Therefore, it is preferred an overall good registration, which will be followed by a smoothing step that increases the Signal to Noise Ratio (SNR).

In case of using several scans from the same patient, either uni- or multi-modal data, intra-subject registration is applied in the form of affine or rigid-body transformation. When templates are used or studies are carried out through several population groups, an inter-subject registration is used, which applies an affine transformation followed by a non-linear warping. Some authors also propose the use of just an affine transformation for inter-subject registration in order to account only the overall brain dimension differences.

After applying the transformation, the images are re-sliced in order to have intensity values associated to a spatially homogeneous cubic grid. This re-slicing implies an *interpolation* that can be either done by Nearest Neighbour (NN) (0th order), tri-linear (1st order), Lagrange polynomial (nth order), sinc or B-splines. In addition, the interpolation can be also done using windowing techniques with similar smoothing results. The interpolation method applied in SPM can be checked in the Matlab function *spm_slice_vol()*, where the default is tri-linear.

The implicit low pass filtering of the transformation, re-sampling and interpolation decrease the quality (resolution) of the volumes. Thus, the question of when and how it should be applied must be analyzed in order to avoid unnecessary data degradation. For this reason, it is usual to store the volumes in the original space with the transformation parameters in the header.

3.2.1 Affine Transformation

The Equation 3.1 presents the affine transformation $T()$ from the original volume X to the target volume Y, $X \rightarrow Y$, where A is the *transformation matrix*, and b is the intercept. The 3D volumes have dimensions $3xN$, where N is the number of variables in the volume. In the case of the MRI scans from the DRCMR, the dimensions of the volumes are $256 \times 256 \times 256$, thus $N = 17,367,040$. The intercep encodes the translation, thus the expression $(.+)$ represents the addition of b to all the N variables of dimensions $3x1$ in X, which would be equivalent to add directly $repmat(b, 1, N)$.

$$Y_{3xN} = T\left\{X_{3xN}\right\} = A_{3x3} \cdot X_{3xN}(.+)b_{3x1} \tag{3.1}$$

The affine transformation is a combination of linear transformations -namely rotation, scaling and shear- and a translation (encoded in the intercept). It is commonly applied a modification of the previous expression in order to deal with homogeneous coordinates. The transformation matrix is converted into an augmented matrix with one additional dimension. In case of volumes, the augmented transformation matrix would have dimensions $4x4$. Therefore, it is needed also to increase the dimensionality of the volumes X and Y to $4xN$.

$$\begin{bmatrix} Y \\ 1 \end{bmatrix}_{4xN} = T\left\{\begin{bmatrix} X \\ 1 \end{bmatrix}\right\}_{4xN} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}_{4x4} \cdot \begin{bmatrix} X \\ 1 \end{bmatrix}_{4xN} \tag{3.2}$$

After this conversion, the transformation matrix A can be decomposed into four individual transformation matrices, including translation, as presented in the Equation 3.3. In addition, the transformation matrix becomes orthogonal. Therefore, the inverse transformation, i.e. $Y \rightarrow X$, can be easily obtained by transposing the original transformation matrix, $A^T = A^{-1}$.

$$A_{affine} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} = A_{translation} \cdot A_{rotation} \cdot A_{scaling} \cdot A_{shear} \tag{3.3}$$

The SPM function *spm_matrix()* creates the previous transformation matrix A_{affine}. The default multiplication order of individual transformation matrices is defined as: Translation, Rotation, Scale and Shear. As SPM uses premultiplication format for the transformation matrix, the transformations will be applied in the opposite order to the original volume.

The appendix B.6 includes a short Matlab example about the formation and use of the matrix, and how affects the coordinates.

The Figure 3.6 depicts in different skews the four steps of the affine transformation: translation, rotation, scale and shear. For simplicity, the figure corresponds to a 2D space, however the same criteria will be applied for a 3D case. In the tri-dimensional case, the translation and scale would have an additional parameters in the z-axis, and the rotation and shear would have two additional parameters accounting for the dimensionality increase.

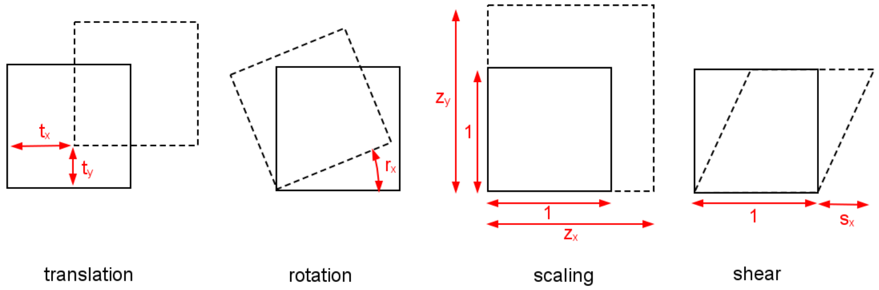


Figure 3.6: Affine 2D transformation skews. It includes translation, rotation, scale and shear of a regular square. The solid line square corresponds to the original shape with normalized dimensions (width=1, height=1). The dashed line square is the target square, i.e. original square after the individual transformation characterized by its respective parameters.

The 3D affine transformation is characterized by 12 parameters (12 degrees of freedom), grouped into 4 individual transformation matrices:

- 3 translation distances: t_x, t_y, t_z.
- 3 rotation angles (pitch, roll, yaw): u_x, u_y, u_z.
- 3 scaling (zoom) factors: z_x, z_y, z_z.
- 3 shear factors: s_x, s_y, s_z.

The specific implementation of the 4 individual transformation matrices is presented in the following equations:

$$A_{translation} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

$$A_{rotation} = A_{pitch} \cdot A_{roll} \cdot A_{yaw} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(u_x) & -sin(u_x) & 0 \\ 0 & sin(u_x) & cos(u_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \tag{3.5}$$

$$\cdot \begin{pmatrix} cos(u_y) & 0 & sin(u_y) & 0 \\ 0 & 0 & 0 & 0 \\ -sin(u_y) & 0 & cos(u_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} cos(u_z) & -sin(u_z) & 0 & 0 \\ sin(u_z) & cos(u_z) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_{scaling} = \begin{pmatrix} z_x & 0 & 0 & 0 \\ 0 & z_y & 0 & 0 \\ 0 & 0 & z_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad A_{shear} = \begin{pmatrix} 1 & s_x & s_y & 0 \\ 0 & 1 & s_z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.6}$$

Rigid Body Transformation

The rigid-body transformation is a simplification of the affine transformation where just rotation and translation are applied, thus perpendicularity and parallelism is conserved (rigid transformation). It assumes not changes in the shape of the brain, hence it can be used to register different scans from the same subject. However, not all the motion artifacts can be corrected with this step. All the non-rigid distortions are sources for errors, especially those connected with fast movements inside the scan or movements correlated with an intensity non-uniformity field.

For example, SPM applies a *motion correction* within the same subject scans for fMRI. Each scan is acquired in slightly different epochs, thus even slow voluntary or involuntary movements can shift the spatial synchronization across the images. By taking into account that the resolution of the MR images used in this project is $\sim 1mm$, phenomena like breathing or hearth pulse can move the brain a distance of the same magnitude order than the acquisition resolution. In order to correct this artifact, SPM realigns the images acquired from the same subject using a rigid body spatial transformation. This process is done individually for each modality by minimizing the Sum of Squared Differences (SSD) between images with the Gauss-Newton ascent iterative process. Once the realignment parameters have been obtained, the images are transformed, interpolated and re-sampled.

3.2.2 Non-linear

The affine transformation is not enough when it is intended to register volumes with local differences between them, which usually happens in inter-subject studies or when templates are used. In these cases, it is needed to apply a non-linear transformation (warping), like B-spline, thin plate, Fourier basis functions, and others. The SPM normalization assumes that the template is a warped version of the original image, therefore problems will arise when brains include lesions or diseases, because it is not possible to match the same structures between both brains.

The non-linear transformation in SPM is done with tri-dimensional basic functions. Namely, the DCT is preferred over the Discrete Sine Transform (DST), as it allows border deformation [25]. However, depending on the dataset, others options could be preferable. For example, a combination of DCT and DST permits free border and fix corners, or in case that more precision is required in the deformation, other high-dimensional registration methods should be used.

The Equation 3.7 presents the 3D transformation U from X to Y as three one-dimensional transforms. The term $q_{j,k}$ stands for the jth coefficient of the kth dimension ($k = 1...3$), and $d_j\left(x_i\right)$ is the jth basis function at position x_i.

$$Y = X + U = \begin{bmatrix} y_{1,i} \\ y_{2,i} \\ y_{3,i} \end{bmatrix} = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ x_{3,i} \end{bmatrix} + \begin{bmatrix} u_{1,i} = \sum_{j=1}^{J} q_{j,1} d_j\left(x_i\right) \\ u_{2,i} = \sum_{j=1}^{J} q_{j,2} d_j\left(x_i\right) \\ u_{3,i} = \sum_{j=1}^{J} q_{j,3} d_j\left(x_i\right) \end{bmatrix} \tag{3.7}$$

The basis functions of one dimension $d_j\left(x_i\right)$ for the the first M coefficients can be obtained from the Equation 3.8, which expression has been extracted from [31]. The first coefficient $d\left(i, m = 1\right)$ for all the variables is constant. The index $i = 1..I$ goes through all the variables of the volume in one dimension.

$$d\left(i, m\right) = \begin{cases} \sqrt{\frac{1}{I}} & m = 1, \quad i = 1..I \\ \sqrt{\frac{2}{I}} \cdot cos\left(\frac{\pi(2i-1)(m-1)}{2I}\right) & m = 2..N, \quad i = 1..I \end{cases} \tag{3.8}$$

In a 2D case, with images X and Y of dimensions IxJ, the shifting field $\mathbf{U}$ of same dimensions IxJ is composed by two fields that can be expressed as $\mathbf{U}_1 \approx \mathbf{D}_1\mathbf{Q}_1\mathbf{D}_2^T$ and $\mathbf{U}_2 \approx \mathbf{D}_1\mathbf{Q}_2\mathbf{D}_2^T$, where $\mathbf{D}_1$ and $\mathbf{D}_2$ have dimensions $I \times M$ and $J \times N$, respectively. Therefore, the matrix of coefficients $\mathbf{Q}$ has dimensions $M \times N$, where M is the number of coefficients and $N = I \times J$ is the total number of variables. From the Equation 3.8 is possible to build the matrix $\mathbf{D}$ with the first M coefficients of the basic function, that can be lately decomposed into the two matrices $\mathbf{D}_1$ and $\mathbf{D}_2$.

The SPM function *spm_dctmtx* generates the basic functions for DCT. For example, the Figure 3.7 presents the first basis functions generated from $spm\_dctmtx(N = 5, K = 5)$.

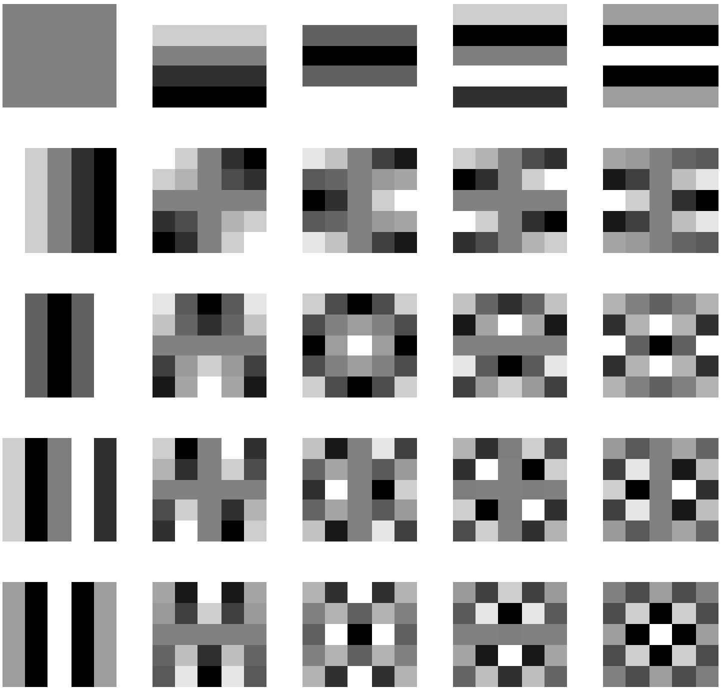


Figure 3.7: First basis functions of DCT, which correspond to the lowest frequencies. They are generated from SPM in the same way that they are used for the warping registration.

It is usually applied a two dimensional deformation field that is based on two scalar fields. One for horizontal deformations in the X-plane, and another for vertical deformations in the Y-plane. The deformation field in the X-plane shifts the voxels right (light intensity) or left (dark intensity), while in the Y-plane the voxels are shifted up (light intensity) or down (dark intensity).

3.2.3 Dissimilarity function

In case that the spatial mapping between origin and source is not given, the transformation parameters must be estimated using a similarity or dissimilarity function that gives a metric about how good/bad two images fit. These functions can be based on intensity (correlation metrics) or features (points, lines, etc). In case of using multi-modal scans, only feature-based similarity function can be used because the intensity pattern does not match among modalities. Therefore, SPM includes Mutual Information (MI), Normalized Mutual Information (NMI) and Entropy Correlation Coefficient for multi-modal studies.

The Equation 3.10 presents the MI similarity function for volumes X and Y as the *Kullback–Leibler* distance, where $H()$ is the entropy [44] [81]. The indexes i, j go through all the intensity values of each volume.

$$S_{MI}\left(X,Y\right)=\sum_{i,j}p_{XY}(i,j)log\frac{p_{XY}(i,j)}{p_X(i)p_Y(j)}=H(X)+H(Y)-H(X,Y) \quad (3.9)$$

The NMI is shown in the Equation 3.10, where it can be checked the relation between the similarity and dissimilarity function: $S_{NMI}\left(X,Y\right)=1-D_{NMI}\left(X,Y\right)$. To achieve a good registration, similarity and dissimilarity must be maximized and minimized, respectively.

$$S_{NMI}\left(X,Y\right)=\frac{S_{MI}(X,Y)}{H(X)+H(Y))}=1-\frac{H(X,Y)}{H(X)+H(Y))} \quad (3.10)$$

The term $H(X)$ stands for the entropy of X, and $H(X,Y)$ is the joint entropy of X and Y, as presented in the equation 3.11. When the registration is improved, the joint entropy is decreased.

$$\begin{aligned} H(X) &= -\sum_{i}p_X(i)log\left(p_X(i)\right) \quad (3.11)\\ H(X,Y) &= -\sum_{i,j}p_{XY}(i,j)log\left(p_{XY}(i,j)\right) \end{aligned}$$

For within modality registration, SSD or Normalized Cross Correlation (NCC) are available. The Equation 3.12 presents the latter one. The terms μ_X and μ_Y correspond to the mean intensity value in each volume.

$$S_{NCC}\left(X,Y\right)=\frac{\sum_i\left(Y-\mu_Y\right)\sum_i\left(X-\mu_X\right)}{\sqrt{\sum_i\left(Y-\mu_Y\right)^2\sum_i\left(X-\mu_X\right)^2}} \quad (3.12)$$

3.2.4 Regularization

In SPM, the parameter estimation is done iteratively together with the bias field correction in order to minimize the cost function. In addition, a regularization term is included to penalize the level of deformations according to the expected spatial variability. Therefore, the cost function to maximize has the form $(1 - S(X, Y)) + \lambda \cdot h_{reg}$, where λ stands for the elasticity constants. When its value is too big, the deformation can be underestimated; on the other side, too small values could lead to overfit the data. h_{reg} is the regularization function which is dependent on the size of the deformation. Three kinds of linear regularization are used: *membrane energy*, *bending energy* and *linear-elastic energy*.

The membrane energy for the deformation field $\mathbf{u}$ is presented in the Equation 3.13 as a sum over all the points in the three dimensions. The smaller is the deformation, the smaller is the regularization term.

$$h_{reg} = \sum_{i} \sum_{j=1}^{3} \sum_{k=1}^{3} \left(\frac{\partial u_{ij}}{\partial x_{ki}} \right) \tag{3.13}$$

The Equation 3.14 presents the bending energy of the deformation.

$$\begin{aligned} h_{reg} = & \sum_{i} \left(\left(\frac{\partial u_{1j}}{\partial x_{1i}} \right)^2 + \left(\frac{\partial u_{1j}}{\partial x_{2i}} \right)^2 + 2 \left(\frac{\partial u_{1j}}{\partial x_{1i} \partial x_{2i}} \right)^2 \right) + \\ & + \sum_{i} \left(\left(\frac{\partial u_{2j}}{\partial x_{1i}} \right)^2 + \left(\frac{\partial u_{2j}}{\partial x_{2i}} \right)^2 + 2 \left(\frac{\partial u_{2j}}{\partial x_{1i} \partial x_{2i}} \right)^2 \right) \end{aligned} \tag{3.14}$$

Finally, the Equation 3.15 presents the Linear-Elastic Energy with the two elasticity constants λ and μ.

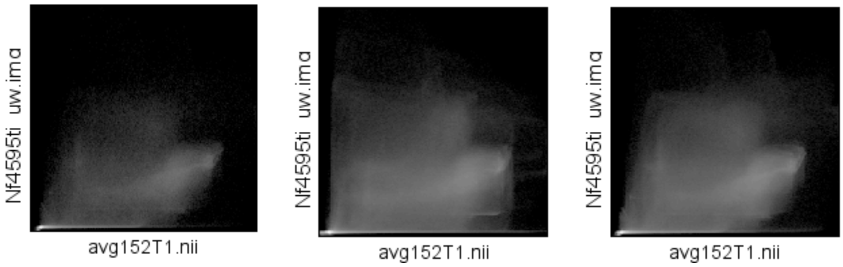
$$\sum_{j=1}^{3} \sum_{k=1}^{3} \sum_{i} \frac{\lambda}{2} \left(\frac{\partial u_{ji}}{\partial x_{ji}} \right) \left(\frac{\partial u_{ki}}{\partial x_{ki}} \right) + \frac{\mu}{4} \left(\frac{\partial u_{ji}}{\partial x_{ki}} + \frac{\partial u_{ki}}{\partial x_{ji}} \right) \tag{3.15}$$

Example of registration with SPM

Finally, an example of the brain images registration is presented here. The SPM co-registration tool is applied between T_1 of subject `f4395` (*Nf4395t1_uw.img*) and the canonical T_1 brain volumes from SPM (*avg152T1.nii*), which is averaged from 152 brains. The interpolation is tri-linear, smoothing of $7mm$, and no warping. The affine transformation from the subject `f4395` space to the canonical space corresponds to the following matrix.

$$A = \begin{pmatrix} 0.500 & 0.011 & -0.005 & -20.756 \\ -0.012 & 0.489 & 0.103 & -10.513 \\ -0.003 & -0.103 & 0.489 & 16.614 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \tag{3.16}$$

The 2D histograms of the registration can be seen in the Figure 3.8, where the middle plot represents the starting point. The left plot depicts the transformation from canonical to the subject *f4505*; and the right one the transformation from the subject *f4505* to canonical. The final histograms (right and left) have higher and less diffuse values. This increase in the concentration of points indicates that the matching of similar areas with different intensity patterns is higher, thus better registration (smaller joint probability).



(a) 2D Histogram of the transformation from canonical to the subject f4505.

(b) 2D Histogram of T1 subject f4505 and canonical in the original spaces.

(c) 2D Histogram of the transformation from the subject f4505 to canonical.

Figure 3.8: Example of volume registration in SPM with the associated 2D histograms for the original and the two transformation directions.

The Figure 3.9 depicts the original (in red square) and transformed volumes in both directions. The main change can be seen in the sagittal plane (top-right) where the canonical brain is looking down, and thus the angle of the original brain is changed through a rotation. This rotation also change the visibility of the eyes on the coronal plane. Although both volumes belong to the same modality, the chosen dissimilarity function is the NMI because the intensity patterns are slightly different between both volumes.

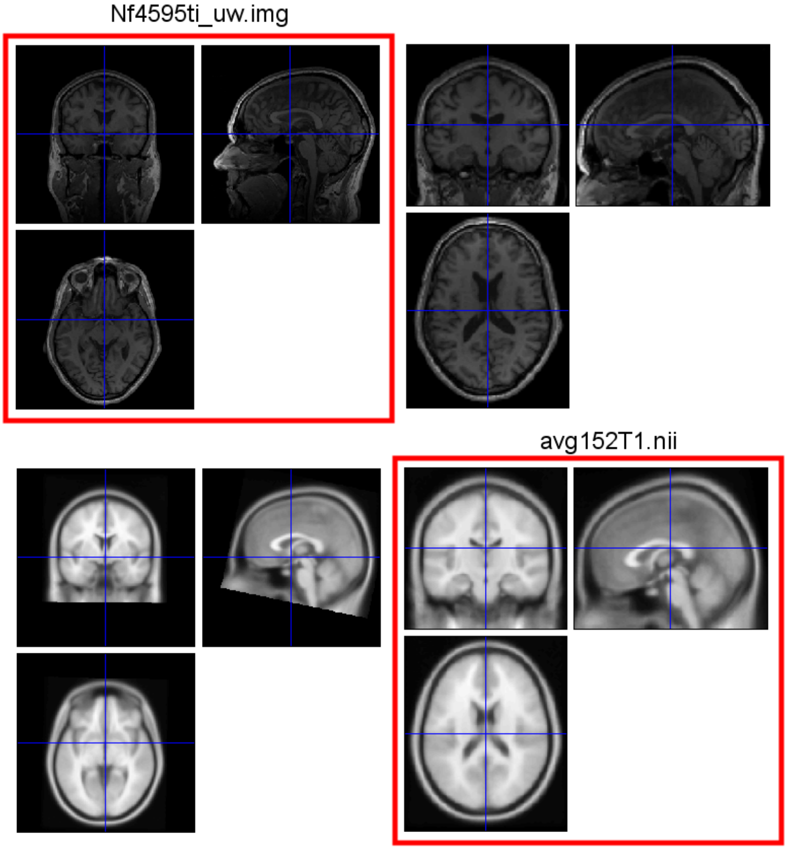


Figure 3.9: Example of volume registration in SPM. Four volumes are presented, two original (with a red square) and two transformed. Each one is represented by the three orthogonal planes: coronal, sagittal, transverse. The upper-left volume corresponds to the original T_1 volume of subject `f4395`, and the lower-right corner is the original canonical T_1 from SPM that is averaged from 152 brains. The upper-right volume is the transformed T1 of subject *f4505* taking as a reference the T_1 canonical, and the lower-left volume presents the transformation of the T_1 canonical to fit the T_1 of subject *f4505* .

3.3 Bias Field Correction

The MR images are corrupted by a smooth signal intensity variation called *intensity non-uniformity*, although some authors have used as well terms like *RF inhomogeneity* or *shading artifact*. The main source of this perturbation is the lack of homogeneity in the Radio Frequency (RF) field (non-ideal coil). Although the visual diagnosis of images is robust to certain levels of non-homogeneity (10%-30%) [75], it decreases greatly the performance of automatic segmentation methods because most of these algorithms assume intensity homogeneity within each class.

It must be distinguish two kinds of intensity inhomogeneity. One appears usually in 2D multi-slice sequences due to the rapid inter-slice intensity variation, and it can be solved by normalizing the intensities of each individual slice. The second accounts for an intensity field that smoothly varies across the volumes in 3D sequences. The latter is the motivation of this section, which is modeled by a multiplicative field that becomes an additive effect in the log-transformed domain. For a further study about sources of intensity inhomogeneity, the reader is referred to work of A. Simmons et al. [73].

The correction of this intensity inhomogeneity can involve a parametric or non-parametric representation. The former one is based on intensity distribution, like the MoG. The latter one is based on intensity histograms, where the entropy of the log-transformed intensity histogram is minimized [75]. Although, the first versions of SPM proposed a non-parametric approach [2], the most-updated released version includes a parametric model of the bias field that is integrated within the intensity generative model [3].

The bias field model created by SPM corresponds to a DCT, in the same way that the wrapping for volume registration. Likewise, the bias field has ~1000 parameters that are the coefficients of the lowest frequency basis functions. The similarity function is based on the segmentation model in order to increase the likelihood of the MoG, and the regularization can be based on either on bending energy, or basis cutoff.

As it was previously mentioned, it is assumed that the bias field p_i is multiplicative. There are different proposals to model how the interaction occurs between the noise n_i and the original intensity μ_i [3]. The Equation 3.17 presents the observed intensity for the i-voxel, y_i, where the main source of noise comes from the scanner.

$$y_i = \frac{\mu_i}{p_i} + n_i \tag{3.17}$$

In the second model, it is assumed that the noise is due to variations of tissue properties inside each voxel.

$$y_i = \frac{(\mu_i + n_i)}{p_i} \tag{3.18}$$

A combination of the two previous models is presented in Equation 3.19, where it is included noise from the scanner and from the tissue variability.

$$y_i = \frac{(\mu_i + n_i)}{p_i} + n_i' \tag{3.19}$$

The last approach log-transforms the intensities of the first model in order to use the advantages of the multiplicative field.

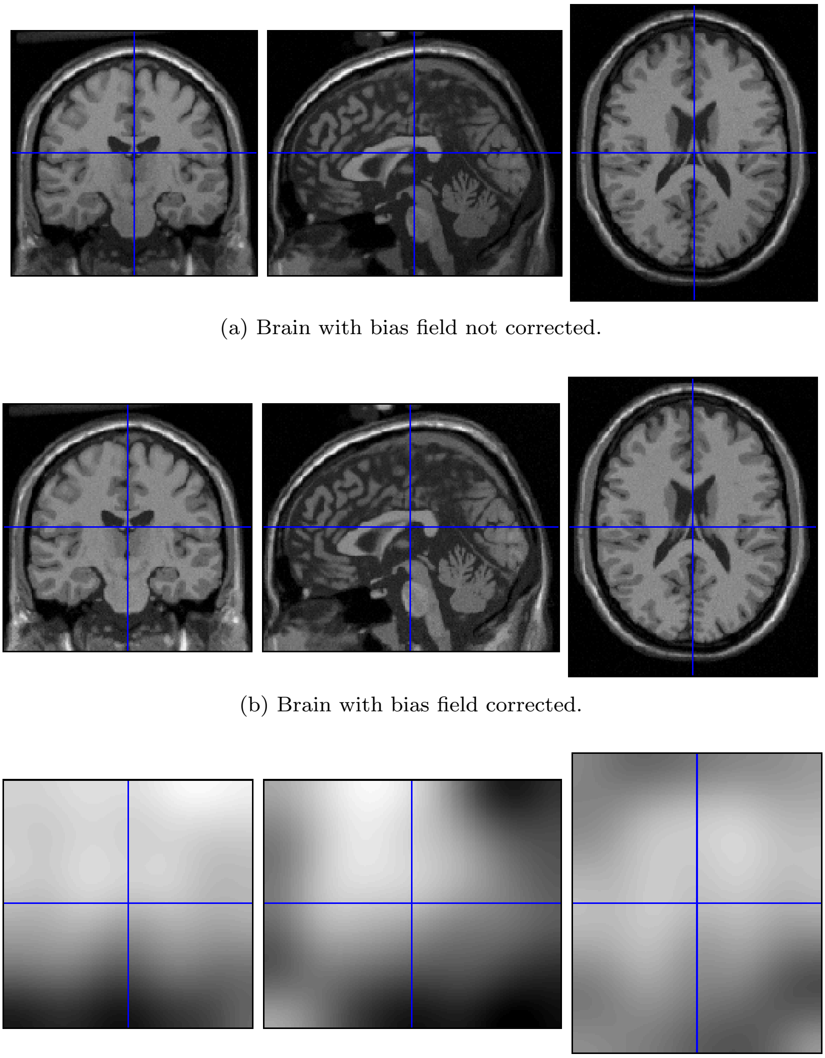
$$log\,(y_i) = log\,(\mu_i) - log\,(p_i) + n_i \quad \Rightarrow \quad y_i = \frac{\mu_i}{p_i} e^{n_i} \tag{3.20}$$

The SPM methods includes the second model, where it is assumed a good quality scanner that does not introduces strong noise.

Example of registration with SPM

For example, the Figure 3.10 depicts the process of bias field correction in SPM. The T_1 volume used in the example corresponds to a simulated scan from BrainWeb using the ICBM protocol [55]. The data has $1mm$ isotropic voxel resolution, and it has been generated with 3% of noise (relative to the brightest tissue), and 40% of intensity non-uniformity.

The two subfigures 3.10a and 3.10b, present the non-corrected and corrected brain volumes, respectively. The non-corrected volume is the one obtained in the scanner, and the corrected is the estimation of the original intensities by assuming negligible noise. Finally, the subfigure 3.10c depicts the multiplicative bias field that modulates the original intensities. The results of the corrected volumes show brighter intensity values.



(a) Brain with bias field not corrected.

(b) Brain with bias field corrected.

(c) Multiplicative bias field.

Figure 3.10: Example of bias field correction in SPM. The T_1 brain volume has $1mm$ isotropic voxel, and it has been generated with 3% of noise, and 40% of intensity non-uniformity. Each row presents a different step of the bias field correction process with the 3 orthogonal planes: coronal, sagittal, transverse.

3.4 Scalp-Stripping

This step classifies the voxels as either brain or non-brain. The result can be either a new image with just brain voxels or a binary mask, which have a value of '1' for brain voxels and '0' for the rest of tissues. In general, the brain-voxels comprises GM, WM, and CSF of the cerebral cortex and subcortical structures, including the brain stem and cerebellum, but not the cervical spinal cord. The Figure 2.2 can help to visualize these parts. The scalp, dura matter, fat, skin, muscles, eyes and bones are always classified as non-brain voxels.

For some methods, this step is mandatory and must be done before the segmentation itself. However, other methods can take benefit of a brain mask, like SPM, in order to decrease the misclassification errors of non-brain voxels. Several methods have been proposed for this processing [42], e.g. Minneapolis Consensus Strip (McStrip) [66], Hybrid Watershed Algorithm (HWA), SPM, Brain Extraction Tool (BST) and Brain Surface Extractor (BSE). In the 'New Segmentation' of SPM8, new tissue templates are included to model non-brain voxels, which helps indirectly the brain extraction. Besides, it is possible to use masks of brain voxels to reduce the computation time because less number of voxels are used, and also it avoids problems when spatial dependencies are taken into account, like in the smoothing step.

In the Figure 3.11, it is presented an example of scalp stripping done by the BST method of FSL on T_1 MRI brain images. In the middle image, the brain edge is depicted with a green line. The right-most subfigure presents the remaining brain tissue after removing the scalp. In addition, it is also presented the segmentation done by FAST.

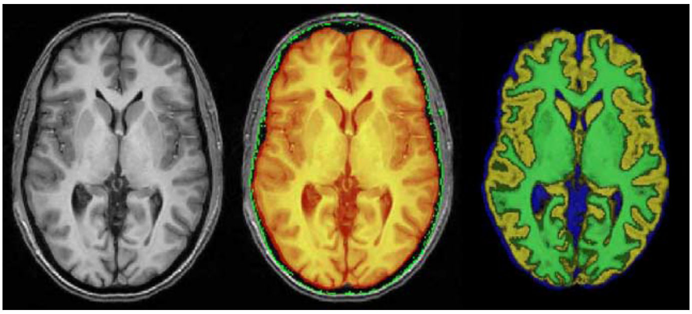


Figure 3.11: Example of scalp stripping done by the BST method of FSL on T_1 MRI brain images. In the middle image, the brain edge is depicted with a green line. The right-most subfigure presents the remaining brain tissue after removing the scalp. In addition, it is also presented the segmentation done by FAST. [Courtesy of S.M. Smith et al. [76]]

Example of scalp stripping with SPM

The Figure 3.12 shows the brain extraction result done by the original pipeline on the scan from the subject `f4395`. In this case, the T_2 MR images are used to estimate the brain mask. The coronal, sagittal, and transverse planes are presented in different states. In the first row, it is presented the row T_1 data, the second row depicts the estimated binary mask, and the bottom row presents the overlapped mask in red on the original T_1 images. In the transverse plane can be seen that the masking is not perfect and some tissue that belongs to the muscles of the right eye is included as brain tissue.

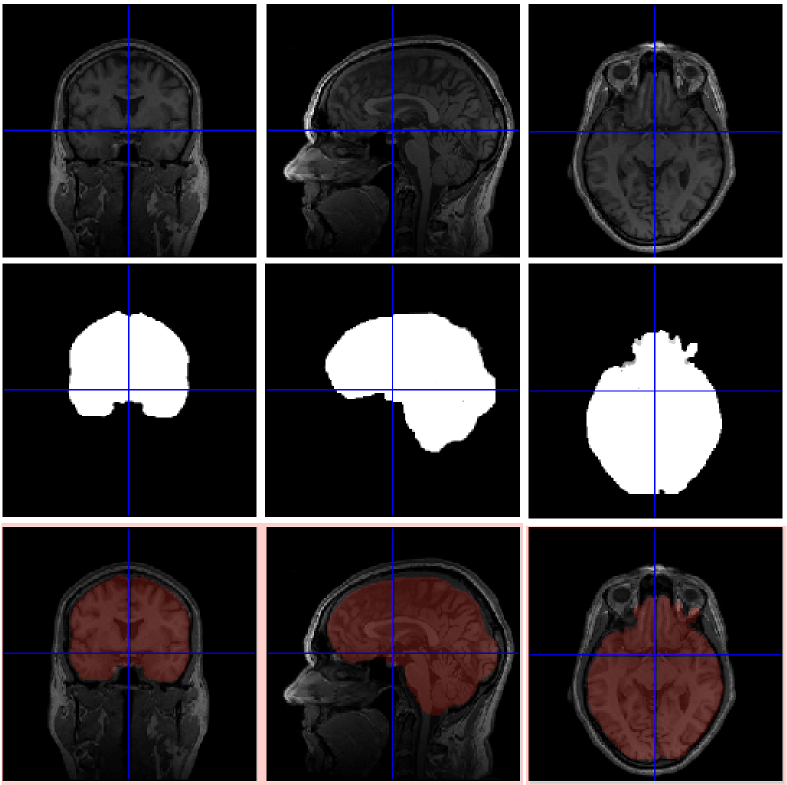


Figure 3.12: Result of the scalp stripping with the original pipeline on a T_1 MR image from subject `f4395`. First, second and third columns correspond to coronal, sagittal, and transverse planes, respectively. The top row shows the original T_1 volumes. The middle row depicts the estimated mask for each plane, where '0' is associated to non-brain (black) and '1' is associated to brain (bright). Finally, the last row presents an overlap of the mask on the raw images.

The previously presented graphical scalp stripping can be also analyzed with the associated intensity histograms. Likewise, the Figure 3.13 depicts the intensity distribution of each step. The main source of no-brain voxels is associated with the air, which appears as a big peak in the low intensity values for both modalities. In addition, there are also small lobes of no-brain that correspond to the skin, eyes, muscles, and other no-brain tissues. After the brain extraction, it is easier to recognize from the histogram the pattern of intensities associated with GM, WM, and CSF.

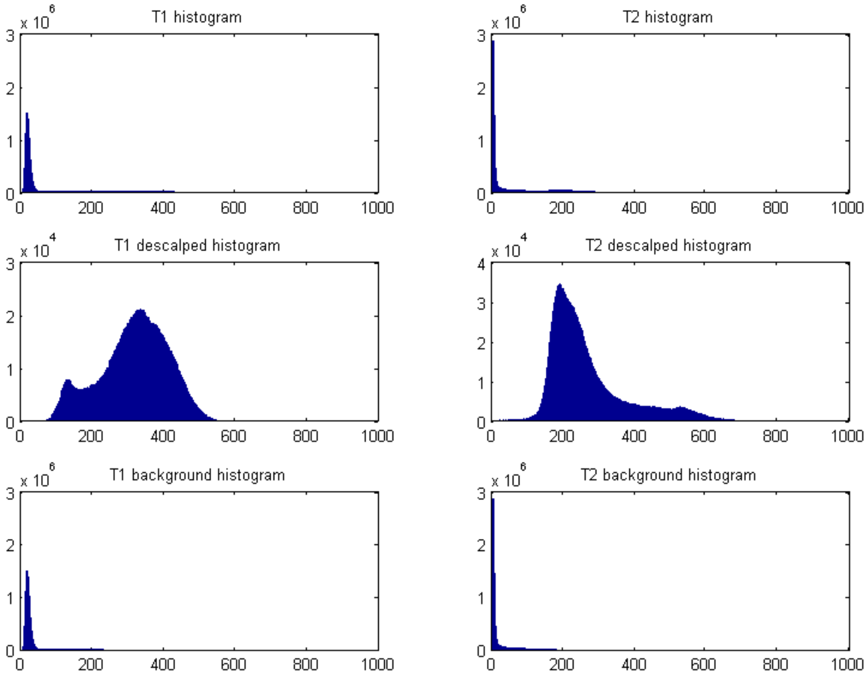


Figure 3.13: Intensity histogram of the scalp stripping done by the original baseline for T_1 and T_2 MR image from subject f4395. The first and second columns correspond to T_1 and T_2 histograms, respectively. The first row contains the original histogram of the scans. In both cases, there is a big peak in the low intensity values that corresponds to the voxels that do not belong to the human body and they appear in black. The second row contains the histogram of the scans after removing no-brain voxels. In this case, it is easier to recognize the pattern of intensities associated with GM, WM, and CSF. Finally, the last row depicts the histogram of the no-brain voxels for T_1 and T_2. The main source of them is associated with the air, although there are also small lobes that correspond to the skin, eyes, muscles, and other no-brain tissues.

3.5 Smoothing

The voxel-wise segmentation has an intrinsic spatial dependency because voxels of the same tissue class tend to be close, i.e. if one voxel is classified as one tissue, it implies that close voxels have more probabilities to belong to the same class. Therefore, spatial information must be included in the model by averaging over neighboring voxels, which blurs the intensity data in the same way that a low pass filtering.

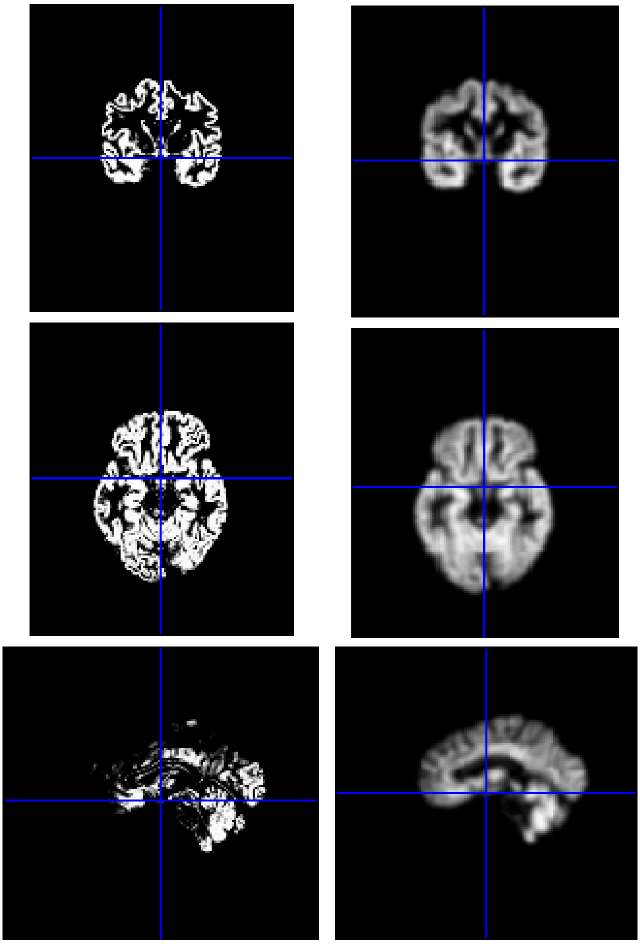
The main goal of this step is to remove isolated dissimilarities among close voxels, which increases the SNR and sensitivity. However, strong smoothness can eliminate the edge information [72]. Therefore, there is a trade off between SNR and image resolution. Besides, it provides an enhanced class overlapping that deals better with PVE, which occurs when a voxel is composed by several tissue classes. This process can be applied before or after the segmentation. In the former case, it reduces the acquisition noise or residual differences after the registration. In the latter case, it generates more uniform TPM's.

Several methods have been proposed for that purpose. One of the most used is based on MRF that ensures continuity of the tissue classes [43]. Other approaches include active contour models like snakes or a Bayesian segmentation based on local histograms [41]. For fMRI analysis, it is usually applied a *weighted average*, where each voxel represents a weighted average over its close Region Of Interest (ROI). Other neuroimgaing steps also introduces indirectly smoothness in the segmentation, like the interpolation done during the registration or the prior template matching.

In the case of SPM, the smoothing is done by the convolution of the volumes with a Gaussian kernel. The process is parameterized by the Full Width at Half Maximum (FWHM) of the Gaussian for each direction (x, y ,z). The proposed values are 6mm for single subject analyses, and 8-10mm for group analysis.

Example of smoothing with SPM

The Figure 3.14 presents the effect of smoothing. In this case the segmented GM tissue volume of the subject `f4395` obtained from the original baseline is smoothed by a Gaussian kernel of FWHM=[8mm,8mm,8mm].



(a) The GM tissue volume.

(b) The GM tissue volume after smoothing.

Figure 3.14: Example of smoothing in SPM. The segmented GM tissue volume (left column) is obtained from the original baseline. The smoothing is done with a Gaussian kernel of FWHM=[8mm,8mm,8mm], and presented in the right column. Each row presents a different plane: coronal, sagittal, transverse.

3.6 Priors and Templates

A *template* corresponds to an image/volume which encodes the average probability of finding different kinds of tissues at each spatial location. These TPM's encode an estimation of the spatial variability at each voxel. Therefore, they are used as prior classification probability in the SPM Bayesian framework. The inclusion of this prior knowledge into the method increases the robustness and accuracy of the method, although also biases the result.

In order to generate a template, several brain volumes are segmented into different tissue classes. Then, all the images are normalized in a common space, which usually corresponds to the MNI space. And finally, a volume for each tissue class is created after averaging and smoothing. The value at each voxel indicates the probability (0-1) of finding the corresponding tissue class at this position in the brain.

In addition, it is also needed an good initial estimation of the parameters, which is used as a starting point of the iterative local optimization. In the case of SPM, this point is simplified by assuming a multidimensional Normal distribution for each parameter, which can be characterized by a mean vector and a covariance matrix. The 'Unified Segmentation' does the initial affine registration maximizing the MI between the volumes and the templates, after excluding an estimation of the BG voxels. On the other hand, the 'New Segmentation' method uses the same equations than for segmentation itself except that the intensity distributions are modeled by histograms, and not by MoG.

The templates of SPM8 in 'New Segmentation' includes 6 TPM's that are used as prior templates. They have dimensions $121 \times 145 \times 121$ and $1.5mm$ of spatial resolution. They are done from 471 brains by Cynthia Jongen of the Imaging Sciences Institute at. Utrecht, NL. Each volume corresponds to a different tissues class, namely GM, WM, CSF, Bone, ST and BG. The volumes include probabilities, therefore each voxel value is in the range $[0, 1]$ and for the same voxel, the sum over the six maps is 1. Some slices for GM, WM and CSF are presented in the Figure 3.15. Besides, all the TPM templates are included in the Appendix E.2.

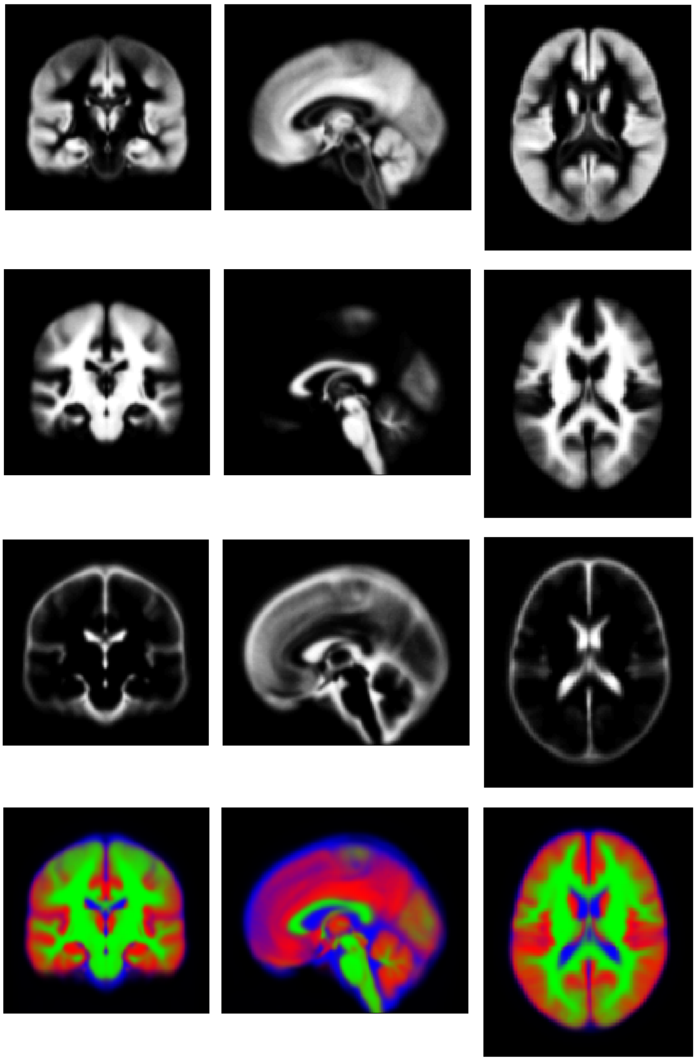


Figure 3.15: Templates for GM, WM and CSF in the 'New Segmentation' of SPM8 for the coronal, transverse and sagittal planes. The last row corresponds to a coloured overlap of the previous tissue probability maps.

Chapter 4

Method & Implementation

This chapter presents the method and mathematical concepts that are applied in the proposed MRI brain segmentation method, which is a modification of the 'New Segmentation' SPM8 toolbox.

In the **Objective Function** section, it is introduced the mathematical framework of the segmentation method. It comprises a MoG as the generative model of the intensity distribution for the tissue classes, and a Bayes inference that allows the inclusion of the prior templates into the model. The objective function is also extended to include bias field correction and registration. In addition, a regularization term is added in order to avoid unfeasible results of the inhomogeneity correction and registration.

The **Optimization** section presents the minimization of the objective function, which is done iteratively with the EM algorithm due to the high coupling of the parameters. The expressions of the mixture parameters for each iteration are calculated with the Gauss-Newton method.

The main parts of the Matlab code are included in the **Implementation** section, where several versions of the algorithm are analyzed.

4.1 Objective Function

This section includes the steps to create the mathematical expression of the objective function, which includes tissue classification, bias field correction, and registration. However, the classification is done indirectly my optimizing the mixture parameters of the model. Besides, a regularization term is added to avoid not realistic inhomogeneity correction and registration.

Due to memory restrictions, the method deals with 2D-planes and only masked voxels are analyzed, i.e. for each z-coordinate, the brain voxels of the corresponding xy-slice are mapped into memory, and then the parameters of the generative model are extracted for those voxels. In order to do it correctly, it is needed to assume independent voxels. Although this assumption is not true, it simplifies the operations and latterly in this chapter it will explained how this voxel dependency can be included in the model.

Some relevant notation is presented here and is kept through the whole chapter. The **variable *Kb*** indicates the number of tissue classes in which is aimed to segment the brain. The **variable *K*** stands for the number of clusters, which also corresponds to the number of Gaussians, as the MoG is used as clustering method. The **variable *N*** stands for the number of modalities/channels, which will determine the dimensionality of the method. The **variable *I*** stands for the number of analyzed voxels of each xy-slice. Although, the number of brain voxels inside of the mask changes for each z-coordinate, the notation I_z is not used for simplicity.

4.1.1 Input data

The acquired MRI data **Y** correspond to the intensity values of each voxel for T_1 and T_2 scans. They are represented as a bivariate sequence of I elements, where I stands for the number of voxels that are analyzed, and the 2-dimensionality comes from the chosen multispectral approach. The analytical expression of the MRI data is presented in the Equation 4.1, where the original 3D matrix structure of the brain voxels is transformed into a vector that concatenates the intensity values. Therefore, for the i-voxel, $y_{i,T1}$ represents the T_1 intensity and $y_{i,T2}$ the T_2 intensity. It is important that beforehand both modalities are correctly registered in the same space coordinates.

$$\mathbf{Y} = [\mathbf{Y}_1, \cdots \mathbf{Y}_i, \cdots \mathbf{Y}_I] = \begin{bmatrix} \mathbf{Y_{T1}} \\ \mathbf{Y_{T2}} \end{bmatrix} = \begin{bmatrix} y_{1,T1} & \cdots & y_{i,T1} & \cdots & y_{I,T1} \\ y_{1,T2} & \cdots & y_{i,T2} & \cdots & y_{I,T2} \end{bmatrix} \tag{4.1}$$

The subscripts $T1$ and $T2$ were used to describe to which modality each variable corresponds, either T_1 or T_2. It could be established a more generic way using an index $n = 1, ...N$. However, in the scope of this project is only intended to use two modalities, $\mathbf{N} = \mathbf{2}$, thus the notation is easier to understand in this format.

4.1.2 Classes

A group of voxels that follows a similar intensity and spatial distribution can be gathered into one cluster. As a tissue class tends to have similar MR intensity values and locations inside the brain, it can be assumed that one cluster can be associated to one tissue class; although this assumption will be latterly expanded. The MoG is applied for clustering, thus each cluster is modeled by a 2D-Gaussian, which is parameterized by a mean vector and a covariance matrix, as it will be presented with more details in the section 4.1.3.

Several classes

For each tissue class in which is aimed to segment the brain, it is needed a template that encodes the prior probability of each voxel belonging to the class. Although the brain segmentation is mainly focused in the detection of GM, WM and CSF voxels, the increase in the number of classes can help to avoid classifying non-brain tissues as part of the brain. The 'Unified Segmentation' method, implemented in SPM8, uses 4 tissue classes ($Kb = 4$), namely GM, WM, CSF, and non-brain. The last class is a way to include the brain extraction within the generative models in order to increase the robustness; and it is estimated as one minus the rest of classes. However, already scalp stripped brains save time because it is only needed to operate with brain voxels, $I \downarrow$. In the 'New Segmentation' method, implemented in the *Seg-toolbox* of SPM8, six different tissue templates ($Kb = 6$) are used, namely GM, WM, CSF, bone, ST and BG. The background class includes mainly air and subjects't hair. In this project, the same priors are used, therefore the number of tissue classes is fixed to six, $\mathbf{Kb} = \mathbf{6}$. Some slices of the used templates can be found in the Appendix E.2.

Several clusters per class

The first approach assumed that each cluster (Gaussian) is associated with one tissue class ($K = Kb$). However, it is a more correct approach to increase the number of clusters (Gaussians) that models one single tissue class ($K > Kb$). One main reason to follow the latter proposal is due to the problems associated with PVE. When a voxel is composed by different tissues, the intensity distribution differs from a pure tissue, thus it is needed to consider these interfaces among structures as different tissue classes. They way that the applied model deals with this problem is by assuming non-Gaussian intensity distribution, which is achieved by associating several Gaussians to one tissue class. The exact number of clusters per tissue class depends on the tissue itself. In this case, the values proposed in the 'New Segmentation' method are used. They associate 2 clusters to GM, 2 clusters with WM, 2 clusters with CSF, 3 clusters with bone, 4 clusters with ST, and 2 clusters with BG. Therefore, the total number of clusters (Gaussians) is fifteen, $\mathbf{K} = \mathbf{15}$.

4.1.3 Mixture of Gaussians

The Gaussian Mixture Models (GMM) or Mixture of Gaussians (MoG) is a generative model that characterizes the intensity distribution as a linear superposition of Gaussians, as presented in Equation 4.2, where the term K corresponds to the total number of Gaussians.

$$P\left(\mathbf{Y}\right) = \sum_{k=1}^{K} \gamma_k \cdot N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.2}$$

Therefore, the kth-cluster (Gaussian) of the MoG model is characterized by the set of parameters: $\boldsymbol{\theta}_k = \{\gamma_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. If the parameters from the K clusters are grouped into $\boldsymbol{\theta} = \{\boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, the mixing coefficient $\boldsymbol{\gamma}$ is a vector with dimensions $Kx1$, the mean vector $\boldsymbol{\mu}$ has dimensions NxK, and the covariance $\boldsymbol{\Sigma}$ has dimensions $NxNxK$, where $N = 2$ and $K = 15$. In the Matlab code, these parameters corresponds to *mg*, *mn* and *vr*, respectively.

In the previous equation, the *mixing proportion coefficient* γ_k weights the contribution of the kth-Gaussian $N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ to the linear superposition. Between the 'Unified Segmentation' and 'New Segmentation' method, there is a slightly difference treatment of this proportion factor. The former method associates one cluster (Gaussian) per tissue class, and the latter one allows the association of several clusters (Gaussian) to single one tissue class. The second approach is followed in this project, thus it must be satisfied that $\sum_{k=k_{Kb}} \gamma_k = 1$ and $0 \leq \gamma_k \leq 1$, where the term k_{Kb} stands for the k-indexes associated to the Kbth-tissue class.

The mixing coefficient, as presented in the Equation 4.2, is spatially independent. However, it has sense that the tissue mixing ratio depends on the location. Therefore, this factor will be combined with the priors to include spatial variations in the section 4.1.5.

The term $N(\mathbf{Y} \mid \mu_{\mathbf{k}}, \boldsymbol{\Sigma}_k)$ of the previous expression corresponds to the *multivariate Gaussian distribution*. The expression for an N-dimensional data $\mathbf{Y}$ is presented in the Equation 4.3, where the kth-Gaussian is parameterized by the mean vector $\mu_{\mathbf{k}}$ and the covariance matrix $\boldsymbol{\Sigma}_k$.

$$N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\left(2\pi\right)^{\frac{N}{2}} \left|\boldsymbol{\Sigma}_k\right|^{\frac{1}{2}}} exp \left\{ -\frac{1}{2} \left(\mathbf{Y} - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1} \left(\mathbf{Y} - \boldsymbol{\mu}_k\right) \right\} \tag{4.3}$$

The intensity distribution of the kth-cluster for the nth-modality is modeled as an univariate Gaussian $(\mathbf{Y_n} \mid C(\mathbf{Y}) = k) \sim N(\mu_{k,n}, \sigma^2_{k,n})$, where $C(\mathbf{Y})$ stands for the class of the variable $\mathbf{Y}$. As two modalities are used in this project, the intensity distributions of the kth-cluster for T_1 and T_2 are modeled as $(\mathbf{Y_{T1}} \mid C(\mathbf{Y}) = k) \sim N(\mu_{k,T1}, \sigma^2_{k,T1})$ and $(\mathbf{Y_{T2}} \mid C(\mathbf{Y}) = k) \sim N(\mu_{k,T2}, \sigma^2_{k,T2})$, respectively.

The combination of both modalities leads to a *bivariate normal distribution*, $N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mid_{N=2}$, which is presented in the Equation 4.4 for the kth-cluster and the ith-voxel. This expression is obtained from Equation 4.3 by constraining the number of dimensions. The process is detailed in the Appendix B.2.

$$\begin{aligned} N(\mathbf{Y} = \mathbf{y}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) &= \frac{1}{2\pi\sqrt{\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}}} \\ \cdot\, exp &\left\{ \frac{-\sigma^2_{k,T1} \cdot \sigma^2_{k,T2}}{2\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \cdot \left[\left(\frac{y_{i,T1} - \mu_{k,T1}}{\sigma_{k,T1}} \right)^2 + \left(\frac{y_{i,T2} - \mu_{k,T2}}{\sigma_{k,T2}} \right)^2 \right] \right\} \\ \cdot\, exp &\left\{ \frac{\sigma_{k,T1T2}}{\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \cdot \left[\left(y_{i,T1} - \mu_{k,T1}\right)\left(y_{i,T2} - \mu_{k,T2}\right) \right] \right\} \end{aligned} \tag{4.4}$$

The Gaussian distribution for the kth-cluster is characterized by the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$, which expressions are presented in the Equation 4.5.

$$\boldsymbol{\mu}_k = \begin{bmatrix} \mu_{k,T1} \\ \mu_{k,T2} \end{bmatrix}_{2,1} , \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \sigma_{k,T1T1} & \sigma_{k,T1T2} \\ \sigma_{k,T2T1} & \sigma_{k,T2T2} \end{bmatrix} = \begin{bmatrix} \sigma^2_{k,T1} & \sigma_{k,T1T2} \\ \sigma_{k,T1T2} & \sigma^2_{k,T2} \end{bmatrix}_{2,2} \tag{4.5}$$

The terms $\sigma_{k,nn'}$ are covariances, and $\sigma_{k,nn} = \sigma^2_{k,n}$ are variances, where $\sigma_{k,n}$ stands for the Standard Deviation (s.t.d.). As a Gaussian, the covariance matrix is symmetric $\sigma_{k,nn'} = \sigma_{k,n'n}$, their terms are real $\sigma_{k,nn'} \in \mathbb{R}$, and their variances are non-negative $\sigma^2_{k,n} \geq 0$. Therefore, the covariance matrix is positive-semidefinite, $\boldsymbol{x}'\boldsymbol{\Sigma}_k\boldsymbol{x} \geq 0,\ \forall \boldsymbol{x} \in \mathbb{R}^n$, and its determinant is non-negative, $det\left(\boldsymbol{\Sigma}_k\right) \geq 0$. If non-singularity is imposed in order to have simple inverse of the covariance matrix, the matrix is positive-definite, $\boldsymbol{x}'\boldsymbol{\Sigma}_k\boldsymbol{x} > 0,\ \forall \boldsymbol{x} \in \mathbb{R}^n$, and its determinant is always strictly positive, $det\left(\boldsymbol{\Sigma}_k\right) > 0$ [15].

The variable ρ_k is the *correlation coefficient* for each pair of modalities, as presented in Equation 4.6.

$$\rho_k = \frac{\sigma_{k,nn'}}{\sigma_{k,n} \cdot \sigma_{k,n'}} = \frac{\sigma_{k,nn'}}{\sqrt{\sigma_{k,nn} \cdot \sigma_{k,n'n'}}} \tag{4.6}$$

This parameter shows the degree of correlation between the nth and n'th modality. In case this parameter vanishes to zero, $\rho_k \to 0$, both modalities would be uncorrelated for the kth-cluster. In addition, due to the Gaussian distribution of the intensities, in case of non-correlation, the modalities would be also independent. The 'unified Segmentation' and 'New Segmentation' methods consider that the probability distribution of both modalities is uncorrelated, $\sigma_{k,T1T2} = \sigma_{k,T2T1} = 0$, which leads to $\Sigma_k = diag(\sigma^2_{k,n})$. This assumption shifts the number of independent parameters to $2 \cdot N = 4$. Although this assumption is not true, it is applied in order to simplify the expressions.

In the proposed method of this thesis, this non-correlation is not assumed. Therefore, the number of independent parameters is $N \cdot (N+3)/2 = 5$. This small number of the degrees of freedom implies fast computation, however it also implies a restriction on the shape of the intensity distribution. The total number of parameters increases quadratically with the number of modalities, although for small number of modalities ($N \downarrow$), like in this project ($N = 2$), this point is not relevant. In this case, the 5 parameters correspond to the two means $\mu_{k,T1}$ $\mu_{k,T2}$, the two variances $\sigma^2_{k,T1}$ $\sigma^2_{k,T2}$, and the covariance $\sigma_{k,T1T2}$, which in details correspond to:

- $\mu_{k,T1}$: intensity mean of the kth-cluster for the T_1 modality.
- $\mu_{k,T2}$: intensity mean of the kth-cluster class for the T_2 modality.
- $\sigma^2_{k,T1}$: intensity variance of the kth-cluster for the T_1 modality.
- $\sigma^2_{k,T1}$: intensity variance of the kth-cluster for the T_2 modality.
- $\sigma_{k,T1T2}$: intensity covariance of the kth-cluster between the T_1 and T_2.

There is also the possibility of using the correlation factor ρ_k instead of the covariance $\sigma_{k,T1T2}$, as both of them include similar information. However, this option is rejected as latterly the objective function is taken derivatives of each parameter, and the dependencies between the correlation factor and the variances would make more though the process.

4.1.4 Bayesian Inference

The Bayesian inference provides a probabilistic framework to combine the priors, the generative model M and the acquired data $\mathbf{Y}$ to estimate the model parameters $\boldsymbol{\theta}$. In addition, this statistical inference allows the probabilistic combination of different methods and data from multiple modalities, which cannot not be done with Least Squares (LE) approaches [90]. Thus, it is possible to infer the MAP at the same time from T_1 and T_2 MR images, which combination is intended in this thesis.

This method introduces prior distributions of the parameters in the model through the Bayes' theorem. The priors corresponds to a set of templates that represent how probable is to find the different tissue classes at each voxel, i.e. TPM. Therefore, there is one template for each kind of tissue class, and it must be fulfilled that the sum over all the templates for each voxel is one.

The Bayes' theorem is presented in the Equation 4.7 for the data $\mathbf{Y}$, the model M, and the model parameters $\boldsymbol{\theta}$, also called hypothesis. The expression $P(\mathbf{Y} \mid \boldsymbol{\theta}, M)$ is the *conditional probability distribution* that corresponds to the likelihood function of the GMM. The term $P(\boldsymbol{\theta} \mid \mathbf{Y}, M)$ is the *posterior distribution*, $P(\mathbf{Y} \mid M)$ is the *marginal distribution*, and $P(\boldsymbol{\theta} \mid M)$ is the *prior distribution*. This expression is derived from the factorization of the *joint probability* $P(\mathbf{Y}, \boldsymbol{\theta}, M) = P(\boldsymbol{\theta} \mid \mathbf{Y}, M) \cdot P(\mathbf{Y} \mid M) = P(\mathbf{Y} \mid \boldsymbol{\theta}, M) \cdot P(\boldsymbol{\theta} \mid M)$.

$$P(\boldsymbol{\theta} \mid \mathbf{Y}, M) = \frac{P(\mathbf{Y} \mid \boldsymbol{\theta}, M) \cdot P(\boldsymbol{\theta} \mid M)}{P(\mathbf{Y} \mid M)} \tag{4.7}$$

If the marginal probability is considered as a scaling factor, it can be stated that the *posterior* probability is proportional to the *likelihood* multiplied by the *priors*, i.e. **posterior** $\propto$ **likelihood** $\times$ **prior**. This posterior probability is the one aimed to be maximized, however the sampling of the prior distribution can be a tough task. Therefore, it is usually applied the Maximum Likelihood (ML) method that assumes directly $P(\boldsymbol{\theta} \mid \mathbf{Y}, M) \propto P(\mathbf{Y} \mid \boldsymbol{\theta}, M)$. This assumption can be based either on a flat distribution of the parameters around the peaks of the likelihood function, or neglected because of a significantly big amount of observed data set. Anyway, the ML estimation corresponds asymptotically to the Bayesian inference [6]. The *likelihood function* is connected with the conditional probability in this way: $L(\boldsymbol{\theta} \mid \mathbf{Y}, M) \propto P(\mathbf{Y} \mid \boldsymbol{\theta}, M)$. Into words, it means that the likelihood of a set of parameter $\boldsymbol{\theta}$, given the observed data $\mathbf{Y}$, is equal to the probability of the observed data $\mathbf{Y}$ given the set of parameters $\boldsymbol{\theta}$.

In conclusion, it can be stated that the ML and MAP estimation of the parameters $\boldsymbol{\theta}$ of the model M from the observed data $\mathbf{Y}$ is obtained by maximizing the conditional probability $P(\mathbf{Y} \mid \boldsymbol{\theta}, M)$. In this case, the model is a MoG, where the set of parameters correspond to $\boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$. Therefore, the likelihood function has the expression $P(\mathbf{Y} \mid \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, and is presented in the Equation 4.8.

$$\begin{aligned} P\left(\mathbf{Y} \mid \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) &= \prod_{i=1}^{I} P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \qquad (4.8) \\ &= \prod_{i=1}^{I} \left(\sum_{k=1}^{K} P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \gamma_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) \\ &= \prod_{i=1}^{I} \left(\sum_{k=1}^{K} P\left(c_i = k \mid \gamma_k\right) \cdot P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) \end{aligned}$$

The previous mathematical expression is obtained in three steps, which are described with more details here:

- First, it is assumed that the intensity distributions of the voxels are Independent and Identically Distributed (i.i.d) stochastic variables. Although this assumptions is not true, it allows to estimate the total likelihood as a simple multiplication of the individual likelihoods at each analyzed voxel $i = 1, \cdots I$. The likelihood of the ith-voxel corresponds to the expression $P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$, and stands for the probability of obtaining the intensity value $\mathbf{y}_i = \left[y_{i,T1}, y_{i,T2}\right]^T$ at the ith-voxel, given the parameters $\boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$. In order to compensate the errors from this assumption, the priors and the smoothing will incorporate in the model the spatial dependencies that are discarded here.

- In the second step, the individual likelihood is expressed as the integration of the joint probabilities at each cluster $k = 1, \cdots K$, which expression is $P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \gamma_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$. This joint probability corresponds to the probability of obtaining the intensity value $\mathbf{y}_i$ that belongs to the kth-cluster, given the parameters $\gamma_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$. As the range of clusters is discrete, the integral is converted into a sum. This step shifts the segmentation from classification to optimization of the model parameters.

- The last expression is obtained by applying the Bayes' rule on the previous joint probability for each cluster. Thus, it can be expressed as the product of the conditional probability $P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \gamma_k, \mu_k, \Sigma_k\right)$ and the prior probability $P\left(c_i = k \mid \gamma_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$. The former does not depend on the mixing coefficient, and the latter does not depend on the Gaussians parameters, therefore the previous terms can be simplified, i.e. $P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \gamma_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$, and $P\left(c_i = k \mid \gamma_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = P\left(c_i = k \mid \gamma_k\right)$.

The prior distribution $P\left(c_i = k \mid \gamma_k\right)$ is the probability of the ith-voxel of belonging to the kth-cluster just taken into account the distributions of previously acquired data. The expression will be be modified in the section 4.1.5 to include the information from the tissue templates.

The conditional distribution $P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ corresponds to the probability of obtaining an intensity value $\mathbf{y}_i = [y_{i,T1}, y_{i,T2}]^T$, given that the ith-voxel belongs to the kth-cluster and the parameters are $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. Due to the MoG model used in this method, this conditional distribution corresponds to the bivariate Gaussian, which will be modified in the Section 4.1.6 in order to include intensity inhomogeneity.

$$P(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \;=\; N(\mathbf{Y} = \mathbf{y}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\,|_{N=2} \tag{4.9}$$

4.1.5 Priors

The applied statistical framework allows the inclusion of prior information. If stationary priors are used, the prior probability for each cluster is just the mixing proportions γ_k, as presented in the Section 4.1.3.

$$P(c_i = k \mid \gamma_k) = \gamma_k \tag{4.10}$$

However, if spatial priors are introduced from TPM, the term b_{ik} is added, which stands for the probability of the ith-voxel belonging to the kth-cluster. As the probability b_{ik} depends on the location, this modification helps to include spatial dependency in the model and compensate the assumption of voxels independence.

$$P(c_i = k \mid \gamma_k) = \frac{\gamma_k \cdot b_{ik}}{\sum_{j=1}^{K} \gamma_j \cdot b_{ij}} \tag{4.11}$$

The templates used as priors are generated from previously segmented images; thus, it is needed to register the tissue templates into the same space than the MR images. The set of parameters $\boldsymbol{\alpha}$ characterizes the image registration with a non-linear warping using 3D DCT. This method is a low-dimensionality ($\sim$ 1000 parameters) approach, which implies a fast and simple processing. A further description of the method was presented in the section 3.2.

$$P(c_i = k \mid \gamma_k, \boldsymbol{\alpha}) = \frac{\gamma_k \cdot b_{ik}(\boldsymbol{\alpha})}{\sum_{j=1}^{K} \gamma_j \cdot b_{ij}(\boldsymbol{\alpha})} \tag{4.12}$$

4.1.6 Intensity Inhomogeneity

The MR scans are corrupted by a smooth intensity perturbation that modulates the intensity values, thus it can be considered as a multiplicative field. Even small levels of inhomogeneity can greatly decrease the performance of any automatic segmentation method, therefore it is needed to compensate it.

The 'Unified Segmentation' method models the bias field as a set of DCT basis functions, as explained in the Section 3.3. This kind of model has a low number of parameters and does not constraint the boundary values. Likewise, the intensity variation of the ith-voxel for the nth-modality is expressed as $\rho_{i,n}(\boldsymbol{\beta})$, which is characterized by the vector of parameters $\boldsymbol{\beta}$. The inclusion of this smooth intensity variation in the MoG model modifies the mean, variance and covariance values of the normal distribution for each cluster in the way presented in the expressions of the Equation 4.13.

$$\mu_{k,n} \rightarrow \frac{\mu_{k,n}}{\rho_{i,n}(\boldsymbol{\beta})} \quad , \quad \sigma^2_{k,n} \rightarrow \left(\frac{\sigma_{k,n}}{\rho_{i,n}(\boldsymbol{\beta})}\right)^2 \quad , \quad \sigma_{k,nn'} \rightarrow \frac{\sigma_{k,nn'}}{\rho_{i,n}(\boldsymbol{\beta}) \cdot \rho_{i,n'}(\boldsymbol{\beta})} \tag{4.13}$$

These expressions are calculated by using the general property of the covariance $Cov(a\boldsymbol{X}, b\boldsymbol{Y}) = ab \cdot Cov(\boldsymbol{X}, \boldsymbol{Y})$, where $a, b \in \mathbb{R}$ and $\boldsymbol{X}, \boldsymbol{Y}$ are Random Variable (r.v.).

The number of modalities in this project is two, T_1 and T_2, therefore the modified mean vector and covariance matrix corresponds to the expressions of the Equation 4.14 and 4.15, respectively.

$$\breve{\boldsymbol{\mu}}_{\boldsymbol{k}} = \begin{bmatrix} \breve{\mu}_{k,T1} \\ \breve{\mu}_{k,T2} \end{bmatrix} = \begin{bmatrix} \dfrac{\mu_{k,T1}}{\rho_{i,T1}(\boldsymbol{\beta})} \\ \\ \dfrac{\mu_{k,T2}}{\rho_{i,T2}(\boldsymbol{\beta})} \end{bmatrix} \tag{4.14}$$

$$\breve{\boldsymbol{\Sigma}}_{\boldsymbol{k}} = \begin{bmatrix} \breve{\sigma}^2_{k,T1} & \breve{\sigma}_{k,T1T2} \\ \breve{\sigma}_{k,T1T2} & \breve{\sigma}^2_{k,T2} \end{bmatrix} = \begin{bmatrix} \left(\dfrac{\sigma_{k,T1}}{\rho_{i,T1}(\boldsymbol{\beta})}\right)^2 & \dfrac{\sigma_{k,T1T2}}{\rho_{i,T1}(\boldsymbol{\beta}) \cdot \rho_{i,T2}(\boldsymbol{\beta})} \\ \\ \dfrac{\sigma_{k,T1T2}}{\rho_{i,T1}(\boldsymbol{\beta}) \cdot \rho_{i,T2}(\boldsymbol{\beta})} & \left(\dfrac{\sigma_{k,T2}}{\rho_{i,T2}(\boldsymbol{\beta})}\right)^2 \end{bmatrix} \tag{4.15}$$

Likewise, the normal distribution of the intensities modulated by the bias field is characterized by the new parameters $\breve{\boldsymbol{\mu}}_k$ and $\breve{\boldsymbol{\Sigma}}_k$, which transforms: $N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mid_{N=2} \rightarrow N(\mathbf{Y} \mid \breve{\boldsymbol{\mu}}_k, \breve{\boldsymbol{\Sigma}}_k) \mid_{N=2}$. From the Equation 4.9, it can be also applied that this bias field correction modifies the conditional probability in the form: $P(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \;\rightarrow\; P(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\beta})$

Finally, the Equation 4.16 presents the intensity distribution with the inclusion of the intensity inhomogeneity parameterized by the vector $\boldsymbol{\beta}$.

$$P(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\beta}) \;=\; N(\mathbf{Y} = \mathbf{y}_i \mid \breve{\boldsymbol{\mu}}_k, \breve{\boldsymbol{\Sigma}}_k) \mid_{N=2} = \tag{4.16}$$

$$= \frac{\rho_{i,T1}(\boldsymbol{\beta}) \cdot \rho_{i,T2}(\boldsymbol{\beta})}{2\pi \sqrt{\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}}} \cdot$$

$$\cdot\, exp\left\{ \frac{-\sigma^2_{k,T2}}{2\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \cdot \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)^2 \right\} \cdot$$

$$\cdot\, exp\left\{ \frac{-\sigma^2_{k,T1}}{2\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \cdot \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)^2 \right\} \cdot$$

$$\cdot\, exp\left\{ \frac{\sigma_{k,T1T2}}{\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \cdot \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right) \right\}$$

As it would be expected, the probability has higher values the closer are the intensity values to their corresponding means. The maximization of the likelihood function has singularities that makes this task not well posed. When the intensity value is close to the mean and the variance is small, the likelihood function values goes to infinity, the same would happen when the variance is too small $\sigma \to 0$. Therefore, these singularities must be detected and heuristic methods to solve them should be proposed [6].

The inclusion of registered priors and bias field correction increases the total number of parameters, i.e. $\boldsymbol{\theta} = \{\boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$. Therefore, the likelihood function has the form of the Equation 4.17, where the expressions for the prior and conditional probability have been already estimated.

$$P\left(\mathbf{Y} \mid \boldsymbol{\theta}\right) \;=\; P\left(\mathbf{Y} \mid \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) \;= \tag{4.17}$$

$$= \prod_{i=1}^{I} \left(\sum_{k=1}^{K} P\left(c_i = k \mid \gamma_k, \boldsymbol{\alpha}\right) \cdot P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\beta}\right) \right)$$

4.1.7 Regularization

The estimated bias field must model smooth variations of intensity due to RF field inhomogeneities and not fast intensity variations due different tissues, which must be modeled by the different clusters. Therefore, a regularization term $P\left(\boldsymbol{\alpha}\right)$ is added in order to penalize unfeasible values of the parameters according to prior information, which corresponds to the bending energy. Similarly, the deformations of the registration can be also penalized in case of parameters too big or small from the expected values. In both cases, the probability densities of the parameters are assumed to follow a centered Gaussian distribution, i.e. $\alpha \sim N\left(0, \mathbf{C}_{\alpha}\right)$ and $\beta \sim N\left(0, \mathbf{C}_{\beta}\right)$.

The Equation 4.18 presents the regularization terms for prior registration and bias field correction in terms of the parameters $\boldsymbol{\alpha}$ $\boldsymbol{\beta}$ and their covariance matrices $\mathbf{C}_{\alpha}$ $\mathbf{C}_{\beta}$.

$$P\left(\boldsymbol{\alpha}\right) = exp^{\left\{-\frac{1}{2}\boldsymbol{\alpha}^{T}\mathbf{C}_{\alpha}{}^{-1}\boldsymbol{\alpha}\right\}} \ , \quad P\left(\boldsymbol{\beta}\right) = exp^{\left\{-\frac{1}{2}\boldsymbol{\beta}^{T}\mathbf{C}_{\beta}{}^{-1}\boldsymbol{\beta}\right\}} \tag{4.18}$$

When the covariance is large, the parameters are expected to be large, which means more drastic deformations and less smooth bias field estimation, and vice versa. Therefore, the value of the regularization terms grows when the order of magnitude of the parameters is similar or smaller than the covariances, which means that the argument of the exponential (related to the Mahanolabis distance) gets closer to zero (centered Gaussian). The Figure 4.19 presents an example of the parameters scaled by their variances. It can be seen how big values are highly penalized.

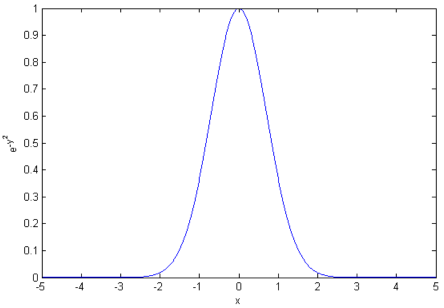


Figure 4.1: Example of regularization terms, where the parameter values are scaled by their variances. When the parameter is bigger than the variance, the regularization term vanishes fast.

Therefore, fitting the MoG model with the regularization terms implies maximizing the Equation 4.19.

$$P\left(\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{\beta} \mid \gamma, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \;=\; P\left(\mathbf{Y} \mid \gamma, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) \cdot P\left(\boldsymbol{\alpha}\right) \cdot P\left(\boldsymbol{\beta}\right) \tag{4.19}$$

4.1.8 Cost Function

The total likelihood is estimated as the product of the individual likelihoods of each voxel and the regularization terms. In order to transform the product into a sum and make easier the optimization problem, the *log*() is applied to the likelihood function to obtain the *log-likelihood function*. The *log*() is a monotone transformation, thus the value of the arguments that maximizes the likelihood are the same than for the log-likelihood. In order to avoid instability problems with small arguments of the *log*() function, it is usually added a small value to the argument. In SPM8, this small value is encoded in the variable $tiny = eps * eps$, where eps is an internal Matlab variable equal to 2^{-52} that stands for the distance from the 1.0 to the next larger bigger number in the same representation. Finally, to create the *cost function* F the sign of the log-likelihood function is changed. Therefore, the maximization of the likelihood is equivalent to the minimization of the Equation 4.20.

$$\begin{aligned} F &= -log\left(P\left(\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{\beta} \mid \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)\right) = \\ &= -log\left(P\left(\mathbf{Y} \mid \boldsymbol{\theta}\right)\right) - log\left(P\left(\boldsymbol{\alpha}\right)\right) - log\left(P\left(\boldsymbol{\beta}\right)\right) = \\ &= \varepsilon + \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{C}_\alpha{}^{-1}\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\beta}^T \mathbf{C}_\beta{}^{-1}\boldsymbol{\beta} \end{aligned} \tag{4.20}$$

From the previous expression, the *cost function* ε corresponds to the objective function F without the regularization terms, and its expression corresponds to the Equation 4.21.

$$\begin{aligned} \varepsilon &= -log\left(P\left(\mathbf{Y} \mid \boldsymbol{\theta}\right)\right) \\ &= -log\left(\prod_{i=1}^{I}\left(\sum_{k=1}^{K} P\left(c_i = k \mid \gamma_k, \boldsymbol{\alpha}\right) \cdot P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\beta}\right)\right)\right) \\ &= -\sum_{i=1}^{I}\left(log\left(\sum_{k=1}^{K} P\left(c_i = k \mid \gamma_k, \boldsymbol{\alpha}\right) \cdot P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\beta}\right)\right)\right) \end{aligned} \tag{4.21}$$

The final objective function expression is presented in the Equation 4.22 of the next page as a compendium of the previous presented steps.

If the expressions of the prior probability from Equation 4.12 and the conditional probability from Equation 4.16 are substituted in the cost function of the Equation 4.21, and included in the Equation 4.20, it is obtained the final objective function of this modified 'Unified Segmentation', which is presented in the Equation 4.22. This expression can be compared with the theoretical approach of a GMM [Equation (9.14) in C. Bishop et al. [6]]. In addition, if non-correlation among modalities is assumed, $\sigma_{k,T1T2} \rightarrow 0$, the presented expression corresponds to the one proposed by the original method [Equation (44) in J. Ashburner et al. [3]].

$$
\begin{aligned}
F &= -log\left(P\left(\mathbf{Y},\boldsymbol{\alpha},\boldsymbol{\beta} \mid \boldsymbol{\gamma},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)\right) = -log\left(P\left(\mathbf{Y} \mid \boldsymbol{\theta}\right)\right) - log\left(P\left(\boldsymbol{\alpha}\right)\right) - log\left(P\left(\boldsymbol{\beta}\right)\right) = \varepsilon + \frac{1}{2}\boldsymbol{\alpha}^T{\mathbf{C}_\alpha}^{-1}\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\beta}^T{\mathbf{C}_\beta}^{-1}\boldsymbol{\beta} = \\
&= -\sum_{i=1}^{I}\left(log\left(\sum_{k=1}^{K} P\left(c_i = k \mid \gamma_k, \boldsymbol{\alpha}\right) \cdot P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\beta}\right)\right)\right) + \frac{1}{2}\boldsymbol{\alpha}^T{\mathbf{C}_\alpha}^{-1}\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\beta}^T{\mathbf{C}_\beta}^{-1}\boldsymbol{\beta} = \\
&= -\sum_{i=1}^{I}\Bigg(log\Bigg(\sum_{k=1}^{K} \frac{\gamma_k \cdot b_{ik}(\boldsymbol{\alpha})}{\sum_{j=1}^{K}\gamma_j \cdot b_{ij}(\boldsymbol{\alpha})} \cdot \frac{\rho_{i,T1}(\boldsymbol{\beta}) \cdot \rho_{i,T2}(\boldsymbol{\beta})}{2\pi\sqrt{\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2}} \cdot \qquad (4.22) \\
&\cdot exp\left\{\frac{-\sigma_{k,T2}^2}{2\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \cdot \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)^2\right\} \cdot \\
&\cdot exp\left\{\frac{-\sigma_{k,T1}^2}{2\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \cdot \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)^2\right\} \cdot \\
&\cdot exp\left\{\frac{\sigma_{k,T1T2}}{\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \cdot \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)\right\}\Bigg)\Bigg) + \frac{1}{2}\boldsymbol{\alpha}^T{\mathbf{C}_\alpha}^{-1}\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\beta}^T{\mathbf{C}_\beta}^{-1}\boldsymbol{\beta}
\end{aligned}
$$

4.2 Optimization

Once the expression of the objective function F has been obtained, this section describes how it is minimized, which implies fitting the model. Due to the high coupling of the parameters $\boldsymbol{\theta}$, there is not a closed formulation to find the solution. The chosen approach by the 'Unified Segmentation' method is based on Iterated Conditional Modes (ICM) [3], where each parameter is locally optimized until the convergence criteria are satisfied. Each iteration comprises the individual optimization of all the parameters, where in each individual optimization one parameter value is minimized while keeping the values of the rest. Therefore, this local optimization requires a good starting point in order to avoid convergence to a local minima.

The mixture parameters -$\boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$- comprise six variables for each cluster, $\{\gamma_k,\ \mu_{k,T1},\ \mu_{k,T2},\ \sigma^2_{k,T1},\ \sigma^2_{k,T2},\ \sigma_{k,T1T2}\}$, which can be easily updated with the EM method. However, the registration $\boldsymbol{\alpha}$ and bias field correction $\boldsymbol{\beta}$ are characterized by ~ 1000, thus they are better optimized by the Levenberg-Marquardt (LM) method. The modification of the original 'Unified Segmentation' done in this project does not involve any change in how the vector parameters α and β are optimized, thus this section will focus just on the mixture parameters that are optimized by the EM scheme. Therefore, it is enough to minimize the objective function ε instead of F, because the regularization terms $P(\boldsymbol{\alpha})$ and $P(\boldsymbol{\beta})$ do not depend on the mixture parameters.

4.2.1 EM optimization

The Expectation Maximization (EM) is a well-known technique used to determine the parameters of a mixture model. The method groups the points by looking for the cluster centers and widths in the data through several iterations, where the convergence criterion is based on minimizing the likelihood function. Besides, this method generates a closed form expression of the parameters for the next iteration. Although, it guarantees the convergence to a local minimum, it is not guaranteed that this minimum is the global minimum of the log-likelihood function [6] [17] [19].

In this case, a slightly different approach is used with the application of the Generalized Expectation Maximization (GEM), where in each iteration the objective function is smaller, but not necessarily minimized [49]. The method generates an upper bound ε_{EM} of the objective function ε, where D_{KL} corresponds to the *Kullback-Leibler* distance [38].

$$\varepsilon \ \leq \ \varepsilon_{EM} \quad \Leftrightarrow \quad \varepsilon \ \leq \ \varepsilon + D_{KL} \ , \quad \forall\ D_{KL} \geq 0 \tag{4.23}$$

In the Equation 4.24, it is presented the cost function ε. It was obtained in the Equation 4.21 of the previous section, but it is repeated here for clarity.

$$\varepsilon \;=\; -\sum_{i=1}^{I} log\left(P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\theta}\right)\right) \tag{4.24}$$

The Equation 4.25 presents the Kullback-Leibler distance D_{KL}, where the term $q_{i,k}$ stands for some probability that is expected to be similar to the posterior probability $P\left(c_i = k \mid \mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\theta}\right)$. It must satisfy that $\sum_{k=1}^{K} q_{i,k} = 1$ after applying the Bayesian rule.

$$\begin{aligned} D_{KL} &= \sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(\frac{q_{i,k}}{P\left(c_i = k \mid \mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\theta}\right)}\right) \;= \\ &= \sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(q_{i,k}\right) - \sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(P\left(c_i = k \mid \mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\theta}\right)\right) \end{aligned} \tag{4.25}$$

Therefore the final upper bound of the the cost function corresponds to the expression of Equation 4.26.

$$\begin{aligned} \varepsilon_{EM} &= \varepsilon + D_{KL} \;=\; -\sum_{i=1}^{I} log\left(P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\theta}\right)\right) \;+ \\ &+ \sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(q_{i,k}\right) - \sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(P\left(c_i = k \mid \mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\theta}\right)\right) \end{aligned} \tag{4.26}$$

In order to minimize ε_{EM}, the method alternates between the E-step and the M-step for each iteration. The former minimizes ε_{EM} with respect to $q_{i,k}$, while the latter does it with respect to $\boldsymbol{\theta}$. Each step has a slightly different reformulation of the cost function, which is minimized to obtain a close equation of the variables for the next iteration. The method stops when the convergence criteria have been satisfied.

E-step

In this step, the upper bound ε_{EM} is minimized with respect to the probability $q_{i,k}$. The cost function ε does not depend on $q_{i,k}$, thus the minimization in this step only includes the Kullback-Leibler distance, $\varepsilon_{EM} = D_{KL}$. When the probability $q_{i,k}$ is equal to the posterior probability, the KL-distance is minimum. This minimum value corresponds to zero, $D_{KL} = 0$, which also implies that the upper bound of the cost function is equal to the cost function, $\varepsilon = \varepsilon_{EM}$.

$$q_{i,k} = P\left(c_i = k \mid \mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\theta}\right) \quad \Rightarrow \quad \varepsilon = \varepsilon_{EM} = D_{KL} = 0 \tag{4.27}$$

The value of the posterior probability for the (n)th-iteration, $q_{i,k}^{(n)}$, is calculated from the parameters of the $(n-1)$th-iteration, $\boldsymbol{\theta}^{(n-1)}$. The Equation 4.28 presents this expression, where the Bayesian rule has been applied.

$$\begin{aligned} q_{i,k}^{(n)} &= P\left(c_i = k \mid \mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\theta}^{(n-1)}\right) = \\ &= \frac{P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}^{(n-1)}\right)}{P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\theta}^{(n-1)}\right)} = \frac{P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}^{(n-1)}\right)}{\sum_{k=1}^{K} P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}^{(n-1)}\right)} \end{aligned} \tag{4.28}$$

The conditional probability $P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}^{(n-1)}\right)$ of the previous expression is calculated from the Equation 4.29. It combines the MoG model, the bias field correction and the priors that were explained in the previous section.

$$\begin{aligned} &P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}^{(n-1)}\right) = P\left(c_i = k \mid \boldsymbol{\theta}^{(n-1)}\right) \cdot P\left(\mathbf{Y}_i = \mathbf{y}_i \mid c_i = k, \boldsymbol{\theta}^{(n-1)}\right) = \\ &= \left(\frac{\gamma_k \cdot b_{ik}(\boldsymbol{\alpha})}{\sum_{j=1}^{K} \gamma_j \cdot b_{ij}(\boldsymbol{\alpha})}\right) \cdot \frac{\rho_{i,T1}(\boldsymbol{\beta}) \cdot \rho_{i,T2}(\boldsymbol{\beta})}{2\pi \sqrt{\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2}} \cdot \\ &\cdot exp\left\{\frac{-\sigma_{k,T2}^2}{2\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \cdot \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)^2\right\} \cdot \\ &\cdot exp\left\{\frac{-\sigma_{k,T1}^2}{2\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \cdot \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)^2\right\} \cdot \\ &\cdot exp\left\{\frac{\sigma_{k,T1T2}}{\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \cdot \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)\right\} \end{aligned} \tag{4.29}$$

M-step

In this step, the upper bound of the cost function ε_{EM} is minimized with respect to the parameters $\boldsymbol{\theta}$. Therefore, the updating equations of the parameters for the (n)th-iteration, $\boldsymbol{\theta}^{(n)}$, are estimated from the resulting cost function and the posterior probability for the (n)th-iteration, $q_{i,k}^{(n)}$, which was previously updated in the E-step.

The first term of the Kullback-Leibler distance does not depend on the parameters $\boldsymbol{\theta}$, therefore it is not included in the upper bound of the cost function in this step, as showed in the Equation 4.30. This expression is a reformulation of the Equation 4.26 where the Bayes' rule has been applied in several steps. Besides, the expression for the conditional probability $P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}\right)$ was presented in the Equation 4.29.

$$
\begin{aligned}
\varepsilon_{EM} &= -\sum_{i=1}^{I} log\left(P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\theta}\right)\right) - \sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(P\left(c_i = k \mid \mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\theta}\right)\right) \\
&= -\sum_{i=1}^{I} log\left(P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\theta}\right)\right) - \sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(\frac{P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}\right)}{P\left(\mathbf{Y}_i = \mathbf{y}_i \mid \boldsymbol{\theta}\right)}\right) \\
&= -\sum_{i=1}^{I}\sum_{k=1}^{K} q_{i,k}\; log\left(P\left(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}\right)\right) \qquad (4.30)
\end{aligned}
$$

Although the values of the mixture parameters are different for each cluster, the updating expression of each parameter is the same for all the clusters. Therefore, the function to minimize corresponds to the upper bound of the function cost for the kth-cluster, ε_{EM_k}. The complete expression is presented in the Equation B.8 of the Appendix B.3.

Therefore, the upper bound of the cost function for the kth-cluster of the M-step is minimized with respect to the parameters $\boldsymbol{\theta}$. This process imply to take derivatives of this expression with respect to each mixture parameter -$\{\gamma_k,\ \mu_{k,T1},\ \mu_{k,T2},\ \sigma^2_{k,T1},\ \sigma^2_{k,T2},\ \sigma_{k,T1T2}\}$- and forced them to be zero. This way, an updating expression for each mixture parameters is obtained.

Mixing Coefficient

The cost function of the Equation B.8 is derived with respect to γ_k and forced to be equal to zero.

$$\frac{\partial\ \varepsilon_{EM_k}}{\partial\ \gamma_k} = \frac{1}{\gamma_k}\sum_{i=1}^{I} q_{i,k} - \sum_{i=1}^{I} q_{i,k}\left(\frac{b_{ik}(\boldsymbol{\alpha})}{\sum_{j=1}^{K}\gamma_j \cdot b_{ij}(\boldsymbol{\alpha})}\right) = 0 \tag{4.31}$$

Therefore, the updating equation for the mixing coefficient corresponds to:

$${\gamma_k}^{(n)} = \frac{\sum_{i=1}^{I} {q_{i,k}}^{(n)}}{\sum_{i=1}^{I} {q_{i,k}}^{(n)}\left(\frac{b_{ik}(\boldsymbol{\alpha})}{\sum_{j=1}^{K}\gamma_j^{(n)} \cdot b_{ij}(\boldsymbol{\alpha})}\right)} \tag{4.32}$$

The Equation 4.32 can be compared with the original 'Unified Segmentation' method, which corresponds to the equation (27) of [3], and it is presented in the Equation 4.33. The original method has a slightly different expression because it was probed empirically its convergence to a smaller cost function in each iteration. Thus, the Equation 4.33 is also used in this project.

$$(\dot{\gamma}_k)^{(n)} = \frac{\sum_{i=1}^{I} {q_{i,k}}^{(n)}}{\sum_{i=1}^{I}\left(\frac{b_{ik}(\boldsymbol{\alpha})}{\sum_{j=1}^{K}\gamma_j^{(n)} \cdot b_{ij}(\boldsymbol{\alpha})}\right)} \tag{4.33}$$

Mean

The cost function of the Equation B.8 is derived with respect to $\mu_{T1,k}$ and forced to be equal to zero.

$$\begin{aligned}\frac{\partial\ \varepsilon_{EM_k}}{\partial\ \mu_{k,T1}} &= -\frac{\sigma_{k,T2}^2}{\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right) \\ &+ \frac{\sigma_{k,T1T2}}{\left(\sigma_{k,T1}^2 \cdot \sigma_{k,T2}^2 - \sigma_{k,T1T2}^2\right)} \sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right) = 0 \qquad (4.34)\end{aligned}$$

The derivative of the cost function with respect to $\mu_{T2,k}$ has similar expression, where the modality indexes are interchanged, $T1 \leftrightarrow T2$. Therefore, the updating equations for the means corresponds to:

$$\mu_{k,T1}{}^{(n)} = \left(\dot{\mu}_{k,T1}\right)^{(n)} + \left(\frac{\sigma_{k,T1T2}}{\sigma_{k,T2}^2}\right) \cdot \frac{coef_{\mu_1}}{\sum_{i=1}^{I} q_{i,k}} \qquad (4.35)$$

$$\mu_{k,T2}{}^{(n)} = \left(\dot{\mu}_{k,T2}\right)^{(n)} + \left(\frac{\sigma_{k,T1T2}}{\sigma_{k,T1}^2}\right) \cdot \frac{coef_{\mu_2}}{\sum_{i=1}^{I} q_{i,k}} \qquad (4.36)$$

These expressions can be compared with the original updating formulas. If the cross variance vanishes towards zero, $\sigma_{k,T1T2} \to 0$, the updating expressions for the mean of the modified method are equal than the original 'Unified Segmentation' ones of Equation 4.37 and 4.38.

$$\left(\dot{\mu}_{k,T1}\right)^{(n)} = \frac{\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1}\right)}{\sum_{i=1}^{I} q_{i,k}} \qquad (4.37)$$

$$\left(\dot{\mu}_{k,T2}\right)^{(n)} = \frac{\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2}\right)}{\sum_{i=1}^{I} q_{i,k}} \qquad (4.38)$$

The coefficients of the updating formulas of the mean for T_1 and T_2 are presented in the Equation 4.39 and 4.40.

$$coef_{\mu_1} = -\sum_{i=1}^{I} q_{i,k} \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right) \qquad (4.39)$$

$$coef_{\mu_2} = -\sum_{i=1}^{I} q_{i,k} \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right) \qquad (4.40)$$

Variance

The cost function of the Equation B.8 is derived with respect to $\sigma^2_{T1,k}$ and forced to be equal to zero.

$$\frac{\partial\ \varepsilon_{EM_k}}{\partial\ \sigma^2_{k,T1}} = 0 = \frac{\sigma^2_{k,T2}}{2\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)}\sum_{i=1}^{I} q_{i,k} \tag{4.41}$$

$$- \frac{\sigma^4_{k,T2}}{2\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)^2}\sum_{i=1}^{I} q_{i,k}\left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)^2$$

$$- \frac{\sigma^2_{k,T1T2}}{2\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)^2}\sum_{i=1}^{I} q_{i,k}\left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)^2$$

$$+ \frac{\sigma_{k,T1T2}\ \sigma^2_{k,T2}}{\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)^2}\sum_{i=1}^{I} q_{i,k}\left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)$$

The derivative of the cost function with respect to $\sigma^2_{T2,k}$ has similar expression than with respect to $\sigma^2_{T2,k}$, except from the interchange of modality indexes, $T1 \leftrightarrow T2$. Thus, the updating formula for the variance are presented in the Equation 4.42 and 4.43.

$${\sigma^2_{k,T1}}^{(n)} = \left(\dot{\sigma}^2_{k,T1}\right)^{(n)} + \left(\frac{\sigma_{k,T1T2}}{\sigma^2_{k,T2}}\right)\cdot\frac{coef^{(n)}_{\sigma_1}}{\sum_{i=1}^{I} q_{i,k}} \tag{4.42}$$

$${\sigma^2_{k,T2}}^{(n)} = \left(\dot{\sigma}^2_{k,T2}\right)^{(n)} + \left(\frac{\sigma_{k,T1T2}}{\sigma^2_{k,T1}}\right)\cdot\frac{coef^{(n)}_{\sigma_2}}{\sum_{i=1}^{I} q_{i,k}} \tag{4.43}$$

Therefore, the updating equation for the variance ${\sigma^2_{k,m}}^{(n)}$ corresponds to a combination of the original formula $(\dot{\sigma}^2_{k,m})^{(n)}$ plus a coefficient $coef_\sigma$ scaled by the cross variance $\sigma_{k,T1T2}$. This formulation allows to see clearly that when the cross variance is zero, $\sigma_{k,T1T2} \rightarrow 0$, the original and modified method have the same updating scheme.

The updating formulas of the original method for the T_1 and T_2 variances are presented in the Equation 4.44 and 4.44, respectively. The equations are presented as dependent on the central moments and as dependent of the non-central moments.

$$\left(\dot{\sigma}^2_{k,T1}\right)^{(n)} = \frac{\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)^2}{\sum_{i=1}^{I} q_{i,k}} = \tag{4.44}$$

$$\frac{\sum_{i=1}^{I} q_{i,k} \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1}\right)^2 - 2\mu_{k,T1} \sum_{i=1}^{I} q_{i,k} \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1}\right) + \mu^2_{k,T1} \sum_{i=1}^{I} q_{i,k}}{\sum_{i=1}^{I} q_{i,k}}$$

$$\left(\dot{\sigma}^2_{k,T2}\right)^{(n)} = \frac{\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)^2}{\sum_{i=1}^{I} q_{i,k}} = \tag{4.45}$$

$$\frac{\sum_{i=1}^{I} q_{i,k} \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2}\right)^2 - 2\mu_{k,T2} \sum_{i=1}^{I} q_{i,k} \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2}\right) + \mu^2_{k,T2} \sum_{i=1}^{I} q_{i,k}}{\sum_{i=1}^{I} q_{i,k}}$$

Finally, the coefficients that modify the original updating formulas of the variance in order to include correlation between modalities T_1 and T_2 are presented in the Equation 4.46 and 4.47.

$$\begin{aligned} coef_{\sigma_1} &= \sigma_{k,T1T2} \sum_{i=1}^{I} q_{i,k} + \left(\frac{\sigma_{k,T1T2}}{\sigma^2_{k,T2}}\right) \sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)^2 \\ &\quad - 2 \sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right) \end{aligned} \tag{4.46}$$

$$\begin{aligned} coef_{\sigma_2} &= \sigma_{k,T1T2} \sum_{i=1}^{I} q_{i,k} + \left(\frac{\sigma_{k,T1T2}}{\sigma^2_{k,T1}}\right) \sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)^2 \\ &\quad - 2 \sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right) \end{aligned} \tag{4.47}$$

Cross Variance

The cost function of the Equation B.8 is derived with respect to $\sigma_{T1T2,k}$ and forced to be equal to zero.

$$\frac{\partial\ \varepsilon_{EM_k}}{\partial\ \sigma_{k,T1T2}} = 0 = -\frac{\sigma_{k,T1T2}}{\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)}\sum_{i=1}^{I} q_{i,k} \tag{4.48}$$

$$+\frac{\sigma^2_{k,T2}\cdot\sigma_{k,T1T2}}{\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)^2}\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)^2$$

$$+\frac{\sigma^2_{k,T1}\cdot\sigma_{k,T1T2}}{\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)^2}\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)^2$$

$$-\frac{\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}}{\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)^2}\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)$$

$$-\frac{\sigma^2_{k,T1T2}}{\left(\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)^2}\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)$$

The solution of this expression for the unknown factor $x = \sigma_{k,T1T2}$ is a third degree equation in the form:

$$ax^3 + bx^2 + cx^1 + dx^0 = 0 \tag{4.49}$$

where the coefficients correspond to:

$$\begin{aligned}
a &= \sum_{i=1}^{I} q_{i,k}\\
b &= -\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)\\
c &= -\ \sigma^2_{k,T1}\ \sigma^2_{k,T2}\sum_{i=1}^{I} q_{i,k}\ +\ \sigma^2_{k,T2}\sum_{i=1}^{I} q_{i,k}\left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)^2\\
&\quad +\sigma^2_{k,T1}\sum_{i=1}^{I} q_{i,k}\left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)^2\\
d &= -\ \sigma^2_{k,T1}\ \sigma^2_{k,T2}\sum_{i=1}^{I} q_{i,k}\left(\rho_{i,T1}(\boldsymbol{\beta})\cdot y_{i,T1}-\mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta})\cdot y_{i,T2}-\mu_{k,T2}\right)
\end{aligned}$$

4.2.2 Central moments

The previous expressions were defined in terms of the central moments for a 2-dimensional variable; therefore, they can be reformulated in an easier way by introducing specific variables for these expressions.

The moment of zero order is:

$$mom0 \; = \sum_{i=1}^{I} q_{i,k}$$

The elements of the central moment of first order are:

$$\begin{aligned} mom1c_{T1} &= \sum_{i=1}^{I} q_{i,k} \;\; \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right) \\ mom1c_{T2} &= \sum_{i=1}^{I} q_{i,k} \;\; \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right) \end{aligned}$$

The elements of the central moment of second order are:

$$\begin{aligned} mom2c_{T1} &= \sum_{i=1}^{I} q_{i,k} \;\; \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)^2 \\ mom2c_{T2} &= \sum_{i=1}^{I} q_{i,k} \;\; \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)^2 \\ mom2c_{T1T2} &= \sum_{i=1}^{I} q_{i,k} \;\; \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right) \end{aligned}$$

In the Appendix B.4, it is included a deep explanation about the central and non-central moments for this case.

Therefore, the coefficients of the updating formulas where the moments are substituted by the previous variables are:

For the mean coefficients:

$$coef_{\mu_1} \;=\; -\; mom1c_{T2} \tag{4.50}$$

$$coef_{\mu_2} \;=\; -\; mom1c_{T1} \tag{4.51}$$

For the variance coefficients:

$$coef_{\sigma_1} \;=\; \sigma_{k,T1T2}\; mom0 \;+\; \left(\frac{\sigma_{k,T1T2}}{\sigma^2_{k,T2}}\right) mom2c_{T2} \;-\; 2 \cdot mom2c_{T1T2} \tag{4.52}$$

$$coef_{\sigma_2} \;=\; \sigma_{k,T1T2}\; mom0 \;+\; \left(\frac{\sigma_{k,T1T2}}{\sigma^2_{k,T1}}\right) mom2c_{T1} \;-\; 2 \cdot mom2c_{T1T2} \tag{4.53}$$

For the cross variance coefficients of the third degree equation:

$$a \;=\; mom0$$

$$b \;=\; -\; mom2c_{T1T2}$$

$$c \;=\; -\; \sigma^2_{k,T1}\; \sigma^2_{k,T2}\; mom0 \;+\; \sigma^2_{k,T2}\; mom2c_{T1} \;+\; \sigma^2_{k,T1}\; mom2c_{T2}$$

$$d \;=\; -\; \sigma^2_{k,T1}\; \sigma^2_{k,T2}\; mom2c_{T1T2}$$

The mixing coefficient is not presented here because the equation of the original and modified approach are the same.

4.3 Implementation

This section introduces the Matlab code of the updating expressions for the E-step ($q_{i,k}^{(n)}$) and M-step (${\gamma_k}^{(n)}$, ${\mu_{k,T1}}^{(n)}$, ${\mu_{k,T2}}^{(n)}$, ${\sigma_{k,T1}^2}^{(n)}$, ${\sigma_{k,T2}^2}^{(n)}$, ${\sigma_{k,T1T2}}^{(n)}$).

First, it is presented the Matlab framework of the implementation, which corresponds to a toolbox in SPM8. The main flow of the program and the variable structures are also explained in order to justify how the updating formulas of this project are included.

4.3.1 SegT1T2 toolbox

The implementations starts with the creation of a toolbox with the name *SegT1T2*, which is a modification of the *Seg* toolbox of 'New Segmentation'. It allows the inclusion of only two input channels for the segmentation, which must correspond to T_1 and T_2 MRI modalities. The extension of the filenames for the program files and volume results of this toolbox is 'seg8T1T2'.

Several parts of the code from different files of the toolbox have been modified in order that the calls among functions works well with the new variables, paths and filenames. However, the most important modifications from the original toolbox can be found in the following files:

- *tbx_cfg_preproc8T1T2.m*: Configuration file that is modified conveniently to use the corresponding paths, the new help/comments hints, and the modified filename extension for the results. It also launches the function *spm_preproc_runT1T2()* with the corresponding parameters.

- *spm_preproc_runT1T2.m*: Function that loads the priors, creates the initial affine registration between input volumes and templates, launches the function *spm_preproc8T1T2()*, and eventually saves the results.

- *spm_preproc8T1T2.m*: Function that does the segmentation itself (fitting the model), where the modified expressions for the optimization of the mixture parameters are included. The input and output variables of this function are deeply explained in the Appendix C.

Therefore, the rest of this section about Matlab implementation will focus on the file *spm_preproc8T1T2.m*.

First, the function *spm_preproc8T1T2()* creates an xyz grid in order to index the voxels of the volumes, c.f *meshgrid()* and *ndgrid()* Matlab functions. For efficiency reasons, not all the voxels are used for segmentation; by default, only one over three voxels is analyzed, *obj.samp=3*. Therefore, the 3D spatial grid *[x0, y0, z0]* and the volume dimensions *d0=size([x0, y0, z0])* are reduced by this factor from the original values. This spatial down-sampling also modifies the transformation matrices.

As it was mentioned at the beginning of this chapter, due to memory restrictions, only one xy-slice is analyzed at each time. Therefore, for each z-coordinate, one 2D-slice is loaded and partial statistics are estimated. The assumption of independence among voxels allows to work in this way, and aggregate the partial results of all the slices at the end. The data for each slice is stored in an internal variable *buf*. This variable corresponds to an array of structs with *z=d0(3)* elements, which number stands for the number of total xy-slices to analyze.

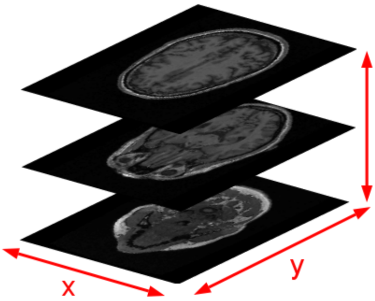


Figure 4.2: Transverse slices that correspond to xy-slices indexed by the z-coordinate.

The variable *buf* has the following fields for each value of z:

- buf(z).msk <d0(3)xd0(3)> Logical 2D-mask with a value of '1' for voxels of this slice to analyze, and zero for the rest. This mask is a combination of the input mask (optional) and an additional mask where zero, infinite and NaN values are also discarded.

- buf(z).nm <1x1 double> Number of voxels inside of the mask for this slice, i.e. I_z. If this number is zero, it is not needed to analize this slice.

- buf(z).f <1x2 cell> Masked input MRI data in the form presented in the Equation 4.1 for $\mathbf{Y}$. Each one of the two elements of the cell is an array of intensity values <nmx1 single>. Therefore, the first and second elements of the cell are $\mathbf{Y_{T1}}$ and $\mathbf{Y_{T2}}$. They are mapped into memory with the function *spm_sample_vol()*.

- buf(z).dat <nmxKb single> Tissue Probability Maps that are sampled for the xy-slice of the z-coordinate. This variable stands for the term $b_{i,k}$ in the Equation 4.12. There are Kb different tissue classes; thus, there also Kb different prior templates.

- buf(z).bf <1x2 cell> Masked bias field for each modality, T_{T1} and T_{T2}, where each channel is an array <nmx1 single>. Therefore, the first element of the cell is $\rho_{i,T1}(\boldsymbol{\beta})$, and the second is $\rho_{i,T2}(\boldsymbol{\beta})$.

Afterwards, the starting estimates of the parameters for the mixture model -$\boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$-, the prior registration -$\boldsymbol{\alpha}$- and the bias field correction -$\boldsymbol{\beta}$- are calculated with the original method. The new updating expressions for the mixture model parameters have not been included to calculate the initial values in order to ensure stability in the first iteration.

The actual estimation of the parameters starts from the line 380 (aprox.) of the file *spm_prepoc8T1T2.m*. It comprises a maximum of 12 iterations, *iter1=1:12*, and each iteration has three blocks: *estimation of cluster parameters*, *estimation of bias field parameters*, and *estimation of deformation parameters*. For each *iter1* iteration, the log-likelihood value is eventually calculated in order to check the convergence. For this thesis, only the first block (*estimation of cluster parameters*) is relevant because the rest do not suffer any modification from the original method. This block runs iteratively 20 times, *subit=1:20*, which means a maximum of 240 times in total. Each iteration comprises the evaluations of the updating equations for the E-step and M-step with the newly calculated values. The evaluation of these expressions is done for each xy-slice individually, *z=1:length(z0)*. To gather all the previously mentioned steps in a clear form, the Algorithm 1 presents the main control flow of this program.

Algorithm 1 Control flow of the function spm_prepoc8T1T2.m

```
$create\ struct\ buz$
$estimate\ starting\ value\ of\ parameters$
for $iter1 = 1 \to 12$ do
    $estimate\ cluster\ parameters$
    for $subit = 1 \to 20$ do
        for $z = 1 \to length(z0)$ do
            $E - step$
            $M - step$
        end for
    end for
    $estimate\ bias\ field\ parameters$
    $estimate\ deformation\ parameters$
    $update\ loglikelihood$
end for
```

The numbers of the lines for the presented Maltab code in the rest of this section are approximately the real numbering of the files. However, small differences can arise due to the inclusion/deletion of comments or the elimination of test code in the release version of the code that is used just for debugging. In addition, some parts of the code are re-arrange from the original method in order to have a more clear structure, although the logical flow remains the same.

4.3.1.1 E-step in Matlab

This step updates the value of the probability $q_{i,k}$. The updating expressions are the same than in the original method, but they are repeated here to justify that they are also useful for the modified multispectral approach. The reason to re-use the code is that the assumption of non-correlation among modalities is not applied here, although it was assumed in the original article [3].

First, it is presented the internal function *likelihoods()*, which estimates the value of an N-dimensional Gaussian function of parameters $\boldsymbol{\mu}$ (mn) and $\boldsymbol{\Sigma}$ (vr) with respect to the value $\rho_{i,n}(\boldsymbol{\beta}) \cdot y_{i,n}$ ($bf\{n\}$ $*$ $f\{n\}$). Besides, the result is multiplied by the mixing coefficient $\boldsymbol{\gamma}$ (mg).

```
function p = likelihoods(f,bf,mg,mn,vr)
K  = numel(mg);
N  = numel(f);
M  = numel(f{1});
cr = zeros(M,N);
for n=1:N,
    cr(:,n) = double(f{n}(:)).*double(bf{n}(:));
end
p  = zeros(numel(f{1}),K);
for k=1:K,
    amp    = mg(k)/sqrt((2*pi)^N * det(vr(:,:,k)));
    d      = cr - repmat(mn(:,k)',M,1);
    p(:,k) = amp * exp(-0.5* sum(d.*(d/vr(:,:,k)),2));
end
```

After calling the function *likelihoods()* with appropriate parameters, the result is multiplied by the registered priors $b_{ik}(\boldsymbol{\alpha})$, which are stored in the temporal variable b. Therefore, the variable q contains the conditional probability $P\big(\mathbf{Y}_i = \mathbf{y}_i, c_i = k \mid \boldsymbol{\theta}\big)$ from the Equation 4.29 without the factor $\sum_{j=1}^{K} \gamma_j \cdot b_{ij}(\boldsymbol{\alpha})$. As this term appears in the numerator and denominator of the expression to estiamte $q_{i,k}$, it is not needed to be calculated because it will vanish anyway. The final $q_{i,k}$ value is obtained from the Equation 4.28, where a small value *tiny* has been added to the denominator in order to ensure stability.

```
q = likelihoods(buf(z).f,buf(z).bf,mg,mn,vr);
for k1=1:Kb,
    b = double(buf(z).dat(:,k1));
    for k=find(lkp==k1),
        q(:,k) = q(:,k).*b;
    end
    clear b
end
sq = sum(q,2);
for k=1:K,
        q(:,k) = q(:,k)./(sq+tiny);
end
```

4.3.1.2 M-step in Matlab

In this step, the central and non-central moments are calculated, then the values of the original mixture parameters are estimated, and finally the modified updating formulas of the mixture parameters are evaluated using the moments and original mixture paramters previously estimated.

Central and non-central Moments in Matlab

The estimation of the moments starts with the calculation of the variable cr, which is the equivalent of the intensity modulated by the bias field $(P(\beta) \cdot Y)$, i.e. $cr(i,n) = \rho_{i,n}(\boldsymbol{\beta}) \cdot y_{i,n}$. In the line 407 of the following code, the variable $buf(z).f\{n\}$ is the masked intensity value for the z-slice in the nth-channel, and the variable $buf(z).bf\{n\}$ is the exponential of the masked bias field for the z-slice in the nth-channel. With this variables is finally obtained the non-central moments of zero, first and second order.

Afterwards, the mean value is removed from the intensity values of the variable cr and stored in the variable crc, i.e. $crc(i,n) = \rho_{i,n}(\boldsymbol{\beta}) \cdot y_{i,n} - \mu_{k,n}$, in order to obtain the central moments of first and second order.

```
405 cr = zeros(size(q,1),N);
406 for n=1:N,
407    cr(:,n)   = double(buf(z).f{n}.*buf(z).bf{n});
408 end
409 for k=1:K,
410    % Non-central moments
411    mom0(k)      = mom0(k)      + sum(q(:,k));
412    mom1(:,k)    = mom1(:,k)    + (q(:,k)'*cr)';
413    mom2(:,:,k) = mom2(:,:,k) + (repmat(q(:,k),1,N).*cr)'*cr;
414    % Central moments
415    crc = cr - repmat(mn(:,k)',size(q,1),1);
416    mom1c(:,k)    = mom1c(:,k)    + (q(:,k)'*crc)';
417    mom2c(:,:,k) = mom2c(:,:,k) + (repmat(q(:,k),1,N).*crc)'*crc;
418 end
```

It must be highlighted that the computation and the variables cr, $mom0(k)$, $mom1(:,k)$, and $mom2(:,:,k)$ was already implemented in the original method, thus this part of the code is not genuine. They are reproduced here for a clear visualization of the environment needed to calculated the central moments, which implementation is genuine. In order to check that the equations of the central moments and their Matlab implementation is correct, they are estimated in other ways to check their validity. The other different approaches and the results are presented in the Appendix B.4.

Original mixture parameters in Matlab

Once the moments are estimated, the updating formulas of the original method are evaluated. The equations are implemented in a matrix form and stored in the variables *mgX*, *mnX* and *vrX*, which corresponds to the mixing coefficient, mean vector and covariance matrix, respectively.

```
%%%%%%%%%%%%%%%%%%%  Original Equations  %%%%%%%%%%%%%%%%%%%%%%

% ------------------  Mixing coefficient  ----------------------
tmp = mom0(lkp==lkp(k));
mgX(k) = (mom0(k)+tiny)/sum(tmp+tiny);

% ------------------------  Mean  ------------------------------
mnX(:,k) = mom1(:,k)/(mom0(k)+tiny);

% ----------------------  Variance  ----------------------------
vrX(:,:,k) = (mom2(:,:,k) - ...
             mom1(:,k)*mom1(:,k)'/mom0(k))/(mom0(k)+tiny) + vr0;
```

In the previous code, the term *tpm* stands for $\left(\frac{b_{ik}(\boldsymbol{\alpha})}{\sum_{j=1}^{K}\gamma_j \cdot b_{ij}(\boldsymbol{\alpha})}\right)$.

The Equation 4.33 gives value for the mixture coefficient, while the Equations 4.37 and 4.38 give value for the two elements of the mean vector. However, the variance is calculated with the Equation 4.54, in contrast to the previously presented Equations 4.44 and 4.45. The main difference among them is that the equation presented here is defined in terms of the non-central moments, and the others are presented in terms of the central moments.

$$\left(\dot{\sigma}_{k,m}^2\right)^{(n)} = \frac{\sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,m}(\boldsymbol{\beta}) \cdot y_{i,m}\right)^2 \ -\ \mu_{k,m} \sum_{i=1}^{I} q_{i,k}\ \left(\rho_{i,m}(\boldsymbol{\beta}) \cdot y_{i,m}\right)}{\sum_{i=1}^{I} q_{i,k}} \tag{4.54}$$

Besides, in the original implementation of the variance, it is used the value of the mean from the current iteration, $(\mu_k)^{(n)}$. Although, this implementation seems to work for the original method, some instability problems arose during the implementation of the modified method that were solved by adding afterwards two additional lines. These two lines over-write the values of the variances for an expression in terms of the central moments, where the mean corresponds to the previous iteration, $(\mu_k)^{(n-1)}$.

```
% For estability
vrX(1,1,k) = mom2c(1,1,k)/(mom0(k)+tiny);
vrX(2,2,k) = mom2c(2,2,k)/(mom0(k)+tiny);
```

Modified mixture parameters in Matlab

The next step is to update the previous values of the mixture parameters with the formulas of the modified method. For each parameter, it is estimated its coefficient and then the final value is calculated as a combination of the original value and this coefficient. As a reminder, the original values were calculated with the the original method, except for the case of the variances.

```matlab
%%%%%%%%%%%%%%%%%%%  Modified Equations  %%%%%%%%%%%%%%%%%%%%%%
% ------------------  Mixing coefficient  ---------------------
mg(k)  = mgX(k);

% -------------------------  Mean  ----------------------------
coefm1 = -mom1c(2,k);
coefm2 = -mom1c(1,k);
mn(1,k)  = mnX(1,k)  ...
       + (ovr(1,2,k)*coefm1)/(ovr(2,2,k)*mom0(k)+tiny);
mn(2,k)  = mnX(2,k)  ...
       + (ovr(1,2,k)*coefm2)/(ovr(1,1,k)*mom0(k)+tiny);

% ----------------------  Covariance  -------------------------
% >> Variance
coefs1 = ovr(1,2,k)*mom0(k)  ...
         + (ovr(1,2,k)/(ovr(2,2,k)+tiny))*mom2c(2,2,k)  ...
         - 2*mom2c(1,2,k);
coefs2 = ovr(1,2,k)*mom0(k)  ...
         + (ovr(1,2,k)/(ovr(1,1,k)+tiny))*mom2c(1,1,k)  ...
         - 2*mom2c(1,2,k);
vr(1,1,k)  = vrX(1,1,k)  ...
       + (ovr(1,2,k)*coefs1)/(ovr(2,2,k)*mom0(k)+tiny);
vr(2,2,k)  = vrX(2,2,k)  ...
       + (ovr(1,2,k)*coefs2)/(ovr(1,1,k)*mom0(k)+tiny);
```

The expression of the mixture value is the same than in the original method. The values of the coefficients are estimated with the Equations 4.50 and 4.51 for the mean, while the Equations 4.52 and 4.52 are used for the variances. Finally, the values $\mu_{k,T1}$, $\mu_{k,T2}$, $\sigma^2_{k,T1}$ and $\sigma^2_{k,T2}$ are updated with the Equations 4.35, 4.36, 4.42 and 4.43, respectively.

In this implementation, it is used the term *ovr*, which stands for the previous value of the covariance matrix, i.e. $ovr^{(n)} = vr^{(n-1)}$, in order to ensure stability.

Cross Variance in Matlab

The estimation of the cross-variance $\sigma_{k,T1T2}$ is quite different respect to the previous parameters, because there is not an unique closed-form expression. Its value is obtained from solving a 3th degree equation with real coefficients, which has at least one real solution, $x \in \mathbb{R}$ [50]. In this case, the solution can be positive or negative, as the cross-variance can have both signs. An example of a cubic function in presented in the Figure 4.3, where it can be seen the three zero-crossings that corresponds to each one of the three solutions.

To solve this cubic function, it can be used a closed-form approach with one expression for each solution, or it can be solved by looking for the roots of the equation. Both methods are presented in the Appendix B.5 with a test code to compare them. However, the latter needs a starting point that must be chosen carefully in order to be able to find the three solutions in a significant short time. In addition, the former approach is more precise and 40 times faster. Therefore, it is chosen to look for the solutions of the cubic equation with the closed-form equations.

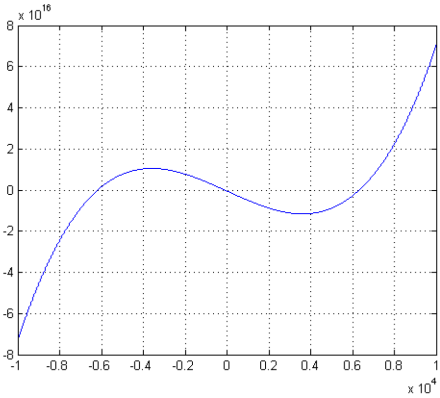


Figure 4.3: Plot of a 3th degree equation with three zero-crossings, which implies three solutions to the equation.

As it was stated in the Section 4.1.3, the covariance matrix is positive-definite, $\boldsymbol{x}'\boldsymbol{\Sigma}_k\boldsymbol{x} > 0,\ \forall\boldsymbol{x} \in \mathbb{R}^n$. Therefore, for the 2-dimensional case, it must satisfy $\sigma_{T1}^2 \cdot \sigma_{T2}^2 > \sigma_{T1T2}^2$. In addition, the cross-variance must be real valued, i.e. $\sigma_{T1T2} \in \mathbb{R}$. Hence, the criterion to select which one of the three solutions is valid will be based on the previous two restrictions. In addition, due to the finite numerical precision of Matlab, it is allowed a small margin of error, which is namely $tiny = 4.9304 \cdot 10^{-032}$ for the first restriction, and 10^{-4} for the second.

Finally, in case that none of the solutions satisfy the criteria, the original cross-variance value is chosen in order to ensure the stability of the method. However, it has been empirically probed with the available dataset that this point is never reached.

The Algorithm 2 presents the logic flow of this part.

Algorithm 2 Algorithm to estimate the adequate cross-variance value

```
$sol \leftarrow get(sol1)$
**if** $\left(\sigma_{T1}^2 \cdot \sigma_{T2}^2 - sol^2 < tiny\right)$ $OR$ $\left(abs(imag(sol) > 10^{-4}\right)$ **then**
  $sol \leftarrow get(sol2)$
  **if** $\left(\sigma_{T1}^2 \cdot \sigma_{T2}^2 - sol^2 < tiny\right)$ $OR$ $\left(abs(imag(sol) > 10^{-4}\right)$ **then**
    $sol \leftarrow get(sol3)$
    **if** $\left(\sigma_{T1}^2 \cdot \sigma_{T2}^2 - sol^2 < tiny\right)$ $OR$ $\left(abs(imag(sol) > 10^{-4}\right)$ **then**
      $sol \leftarrow get(solOriginal)$
    **end if**
  **end if**
**end if**
$sol \leftarrow real(sol)$
```

The solution of the equation $y = coef3\dot{x}^3 + coef2\dot{x}^2 + coef1\dot{x} + coef0$ is implemented in the following code, and corresponds to the previous algorithm.

```matlab
% >> Cross-variance

% Coefficients
coef3 = mom0(k);
coef2 = -mom2c(1,2,k);
coef1 = -ovr(1,1,k)*ovr(2,2,k)*mom0(k) ...
        +ovr(2,2,k)*mom2c(1,1,k) + ovr(1,1,k)*mom2c(2,2,k);
coef0 = -ovr(1,1,k)*ovr(2,2,k)*mom2c(1,2,k);

% Look for the correct solution
x = solution3th(coef3,coef2,coef1,coef0,1);
if ((vr(1,1,k)*vr(2,2,k)-x^2)<tiny) || (abs(imag(x))>1e-4)
   x = solution3th(coef3,coef2,coef1,coef0,2);
   if (vr(1,1,k)*vr(2,2,k)-x^2)<tiny || (abs(imag(x))>1e-4)
      x = solution3th(coef3,coef2,coef1,coef0,3);
      if (vr(1,1,k)*vr(2,2,k)-x^2)<tiny || (abs(imag(x))>1e-4)
         x = vrX(1,2,k);
      end
   end
end

% Give values
vr(1,2,k) = real(x);
vr(2,1,k) = vr(1,2,k);

% Ensure estability
vr(:,:,k) = vr(:,:,k) + vr0;
```

In the previous code, it is used a Matlab function that returns one of the three possible solutions of the cubic equation with coefficients *coef3*, *coef2*, *coef1* and *coef0*. It corresponds to *solution3th(coef3,coef2,coef1,coef0,opt)*, where *opt* is an index to select one of the three possible solutions, $opt \in \{1, 2, 3\}$.

In the last line of code, a term *vr0* is added to the covariance matrix in order to ensure stability. By default, the interpolation method is NN, which implies that the term is estimated as in the presented code, where *pinfo(1,1)* satisfies that $intensity = voxelvalue \cdot pinfo(1) + pinfo(2)$.

```
        vr0(n,n) = 0.083*V(n).pinfo(1,1);
```

Stopping criterion

In each iteration, the log-likelihood value is estimated in the variable *ll* in order to check how well fitted is the model with the current value of the parameters. It is calculated as a combination of the log-likelihoods from the mixture parameters (*llm*), the registration parameters (*llr*), and the bias field parameters (*llrb*). The approximate equations to obtain these values are presented in the following code. In the case of the mixture log-likelihood, the variable *sq* was previously calculated in the E-step, and it is also added a small vale *tiny* in order to avoid instability when the argument of the logarithm is small.

```
    llm = sum(log(sq + tiny));
    llr=-.5*sum(sum(sum(Twarp1.*optimNn('vel2mom',Twarp1,prm,scal)))));
    llrb = chan(n1).ll;
    ll = llm + llr + llrb;
```

The method establishes a stopping criterion to stop the simulation or to change the step size of the LM optimization. Therefore, the likelihood difference between two iterations must be bigger than $tol1 = 1e{-}4$. The Figure 4.4 presents the correspondence between the likelihood and the log-likelihood.

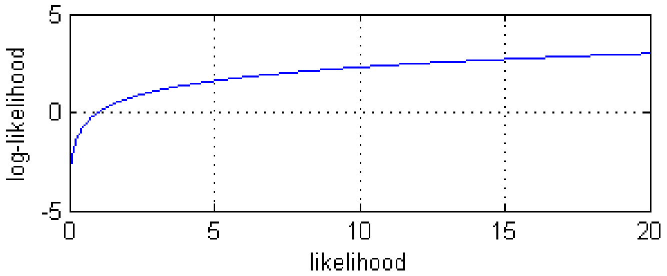


Figure 4.4: Log-likelihood function.

4.3.2 Modifications

The previously presented method and implementation has, at least, two points that can be modified, namely:

- **Starting values**: The optimization is done locally, it is hence important to start with good values in order to avoid the convergence to a local minimum. The initialization of the moments and mixture parameters happens before the main loop, as it was presented in the Algorithm 1. In the original method, the updating equations are slightly changed in order to use them in the initialization step. In this modified method, it is used the same initialization equations than in the original method; however, another set of equations can be derived from the updating equations of the modified method. Therefore, one option is to use the initialization equations from the original method, and the other option is to create other initialization equations based on the modified method. In case of the second option and without further knowledge, the same updating equations of the modified method can be used for the initialization process.
- **Updating values**: The presented equations use the results from the previous iteration to update the parameter values. However, in the original method, the most-updated values of the current iteration are used to update the rest of parameters. An alternative implementation of the modified method can try to speed up the value propagation without incurring in instability problems.

Therefore, the combination of the two points generates four different versions of the modified algorithm. It must be highlighted that all these versions use the modified updating equations that include correlation among modalities.

- The **version 1** updates the parameters with the results of the previous iteration and the initialization equations correspond to the original method.
- The **version 2** also uses just the values from the previous iteration, but in the initialization step, it is used the modified updating equations.
- The **version 3** updates the parameters with the most updated values of the current iteration and uses the initialization of the original method.
- Finally, the **version 4** updates the values with the most recently updated parameters, but uses the updating equations of the modified method for the initialization step.

All these combination will be analyzed in the next section in order to check their performance. In the Appendices, it is included the relevant parts of code for the original method (Appendix C.2), the modified method (Appendix C.3), and the modified method with faster value propagation (Appendix C.4).

Chapter 5

Validation

This chapter presents the segmentation of several brain volumes done by the modified method in comparison with the original method ('New Segmentation') and the original baseline (SPM5+VBM5). The analyzed results comprise the log-likelihood values, the mixture parameters and the generated probability maps for each tissue class.

First, the **Outputs** section presents the variables and volumes that are generated during the segmentation. Besides, it is included a description of several ways to interpret the generated tissue probability maps after the segmentation. A brief discussion about the different approaches to address the performance of the MRI brain segmentation methods is included in the section **Golden Standard**, where a special emphasis is placed in the choice of the ground truth.

The segmentation has been done with default parameters by the original (*Seg*), and the four versions of the modified method (*SegT1T2*). In the section **Brain f4395**, the dataset comprises the brain volumes of the subject `f4395`, while in the section **BrainWeb phantoms**, it has been used a set of brain phantoms with different levels of noise. For the last dataset, the methods are compared in terms of the Dice score.

Finally, the section **CIMBI dataset** presents an analysis of the tissue brain volumes acquired at the DRCMR, where the volume age-profile is estimated.

5.1 Outputs

The first section of this chapter presents a short description about the outputs of the segmentation method, and how the generated probability maps for each tissue are analyzed. The main output variables are the mixture parameters and the final likelihood value. The former ones determine the shape of the clusters, and the latter indicates how well these parameters fit the MRI dataset according to the generative model. The variables are:

- Mixing coefficient (mg): <Kx1 double> array with the final γ values.
- Mean (mn): <2xK double> 2D matrix with the final μ values.
- Covariance (vr): <2x2xK double> 3D matrix with the final Σ values.
- Log Likelihood (ll): <1x1 double> final log-likelihood value.

For example, in the Table 5.1, it is presented the values of the mixing coefficient for the subject `f4395`, where the version 1 of the modified method has been applied with default parameters. As it was stated, the 'New Segmentation' deals in a different way with this factor, and they represent the contribution/proportion of each cluster to the corresponding tissue class. In this case, the number of clusters is $K = 15$ and the number of tissue classes is $Kb = 6$, where the specific associations among them is done with the variable lkp.

Table 5.1: Results of the brain tissue segmentation for the brain scans from the subject f4395 with T_1 and T_2 MR modalities. The applied algorithm is the version 1 of the modified method with default parameters.

Class name	GM		WM		CSF	
Class number (lkp)	1	1	2	2	3	3
Mixing factor (mg)	0.700	0.300	0.718	0.282	0.506	0.494

Class name	Bone			ST			
Class number (lkp)	4	4	4	5	5	5	5
Mixing factor (mg)	0.370	0.437	0.194	0.180	0.271	0.354	0.195

Class name	BG	
Class number (lkp)	6	6
Mixing factor (mg)	0.891	0.109

Therefore, the clusters (Gaussians) are grouped together into Kb tissue classes with the previous weighting values. The order of the clusters is random inside each tissue class, e.g. the GM intensities modeled by the first cluster can correspond to the second cluster for other version of the method or to no-one, which can make difficult and worthless to compare clusters one by one.

An automatic segmentation method ideally associates each voxel to one, and only one, of the classes. Therefore, it can be created a random variable $\mathbf{C}$ that has values in the range of k, where $k \in [1, Kb]$, and Kb stands for the number of different tissue classes. Hence, for the i-voxel, the value of $\mathbf{C}_i$ indicates to which class the voxel has been assigned. This idea is mathematically presented in the Equation 5.1, where I stands for the total number of voxels. The array expression corresponds to a 3D volume where all the voxels have been placed in order along one dimension.

$$\mathbf{C} = [\mathbf{C}_1, \cdots \mathbf{C}_i, \cdots \mathbf{C}_I]^T_{1xI} \;\;,\;\;\; \mathbf{C}_i \in [1, Kb] \tag{5.1}$$

Due to the applied Bayesian framework, the result is not a direct association between voxels and tissues, as in the previous case. The method generates one probability map for each tissue class. Therefore, the previous expression is expanded by Kb rows, as presented in the Equation 5.2, where each row corresponds to a different tissue class and the term $C_{i,k}$ stands for the probability of the ith-voxel belonging to the kth-class.

$$\begin{bmatrix} C_{1,1} & \cdots & C_{i,1} & \cdots & C_{I,1} \\ \vdots & & \vdots & & \vdots \\ C_{1,k} & \cdots & C_{i,k} & \cdots & C_{I,k} \\ \vdots & & \vdots & & \vdots \\ C_{1,Kb} & \cdots & C_{i,Kb} & \cdots & C_{I,Kb} \end{bmatrix}^T_{KbxI} \tag{5.2}$$

The previous expression is a *stochastic matrix*, which must satisfy that the sum over all the classes must be one for each voxel. The matrix elements correspond to probability values in the range $[0, 1]$, where high probabilities are associated to high intensities in the images (white), and vice versa.

$$0 \leq C_{i,k} \leq 1 \;,\; \forall i, k \qquad \textit{and} \qquad \sum_{k=1}^{Kb} C_{i,k} = 1 \tag{5.3}$$

Each row of the Equation 5.2 is stored in a different file, where is resized into a 3D matrix that must satisfy $I = height \times width \times depth$. For example, the generated tissue probability maps for the subject `f4395` are stored in the files: '*c1gf4395_mpr.nii*', '*c2gf4395_mpr.nii*', '*c3gf4395_mpr.nii*', '*c4gf4395_mpr.nii*', and '*c5gf4395_mpr.nii*'. These files correspond to the GM, WM, CSF, bone and ST. The BG map is not directly stored, but it can be generated as one minus the rest of volumes. In the Appendix E.3, some slices of these generated probability maps are presented.

For volume studies, where it is not needed to specify an unique class for each voxel, it is assumed that each voxel is composed by several tissues. Therefore, the total voxel volume is split up into different classes. The volume ratio of each tissue corresponds to the associated value of the probability map, and the total volume of each class is obtained by simple integration over each TPM. This formulation deals better with the PVE that happens when a voxel is composed by several tissues, thus the acquired intensity value is a combination of different intensity patterns. However, in case it is needed to associate each voxel to one, and only one, tissue class in order to apply validation tests, there are three ways to generate a result like the Equation 5.1 from the TPM of the Equation 5.2.

- *Thresholding*: A threshold value is established, and the voxels with higher probability than this value for one class are assigned to this tissue. For example, in the 'New Segmentation' method, it is suggested a threshold of 0.5 [3]. In case of setting a threshold smaller than 0.5, there can be situations where two tissues have higher probabilities than the threshold for one voxel, which generates an ambiguity problem. In addition, with a threshold of 0.5, it is considered with the same weight a voxel with probability 0.95 than another with probability 0.55.
- *Majority Voting*: An additional criterion to solve the previous ambiguity consists on associating to each voxel the tissue label of the TPM with higher probability. It implies a brute force search through all the voxels and maps. This approach solves some ambiguity problems, and it is considered the common way to deal with TPM. However, there are situation where this method is not optimal. If the voxel lies in the interface between two tissues with probabilities for GM, WM and CSF of $\{0.50, 0.45, 0.05\}$; the voxel is considered as GM, although a proportional classification of GM and WM would be more fair. In another situation, with for example the following probabilities $\{0.35, 0.25, 0.40\}$, the voxel is classified as CSF, even though it is more likely to be a brain voxel.
- *Majority Voting + neighbourhood information*: An improvement of the previous method includes information from neighboring voxels in order to assign the class membership. For example, the segmentation library of FSL [76] [90] includes an MRF model that shifts the probabilities to either 0 or 1 depending on the class of the closest voxels. Besides, the VBM8 toolbox includes a denoising filter based on Spatial Adaptive Non-Local Means (SANLM).

The second option is the one applied in this thesis, and its effect on the probability maps will be also analyzed in the following section.

5.2 Golden Standard

In the MRI brain segmentation field, there is not voxel-wise *golden standard* (ground truth). This lack of a reliable correspondence between tissue classes and the acquired intensity value of each voxel makes difficult to compare the accuracy and reproducibility of the algorithms [93]. There is an extent literature about several validation methods that can be grouped into three main ways depending on which reference is used [70].

- **Ex-vivo manual segmentation**: The histological biopsy of the brain could be the ground truth, as it directly addresses the kind of tissue. However, it is needed to associate the part of the body under the microscope to the group of acquired voxels before or after the dead, which is a laborious and hard task. Besides, it is obvious that the study can only be done on dead people. One example of this kind of database corresponds to the *Visible Human Project* [1].

- **Image manual segmentation**: Traditionally, the expert manual segmentation through visual analysis of the acquired images has been considered the reference standard. However, the process is time consuming and costly, as it is needed an import amount of time by well-trained professionals to accomplish this task [72]. In addition, it introduces a high intra-subject and inter-subject variability due to the personal subjectivity [85], which can reach discrepancy rates higher than 20% for the simulated data and higher than 24% for the real data sets [37]. Some databases with manual segmentations are available at the *Internet Brain Segmentation Repository* of Massachusetts General Hospital. The data include 20 Normal Subjects scanned with T1-weighted MRI and expert segmentation with three tissue labels, namely GM, WM and no-brain [54].

- **Phantoms**: The most used validation technique consists on simulating MR images by an artificial physical or digital generative model. These data can be used to evaluate the performance of neuroimaging methods with a common and realistic known truth [48]. For example, BrainWeb from MNI [55] provides a Simulated Brain Database (SBD). It consists on simulated MRI data volumes for T_1, T_2 and Proton Density (PD) modalities. Two anatomical models -normal and Multiple Sclerosis (MS)- are available, as well as different slice thicknesses, noise levels, and bias field [14] [40]. Although, these phantoms can provide an accurate reference standard, they do not reproduce in a realistic way all the range of different scenarios in the clinical data [86].

The last option is the one selected for this thesis because it provides a reliable ground truth. However, it is not possible to make a generalization of the obtained results as the brain and scanning variability is not totally included.

5.3 Brain f4395 - Visualization

Once the method is implemented, it is needed to validate its performance. Therefore, it is checked that the result is what is expected to be. In this case, the T_1 and T_2 MRI brain volumes of the subject `f4395` are segmented by the original and the four versions of the modified method.

The *Seg* and *SegT1T2* toolboxes can be tuned up with a set of parameters, which are equal for both of them. The selection of the exact value for each parameter depends on the dataset, and it is a laborious task that is usually done through an empirical exploration. Therefore, the default values are used in this first approach as they seem reasonable, although they are not optimal. A brief description of these parameters and the default values are presented here:

- *Number of Gaussians*: Number of clusters (Gaussians) that are associated to each tissue class. Due to the used template dataset, the number of tissue classes is 6. Default value: [1,1,2,2,3,3,4,4,4,4,5,5,5,5,6,6], which means 15 Gaussians where 2 clusters are for GM, 2 clusters for WM, 2 clusters for CSF, 3 clusters for Bone, 4 clusters for ST, and 2 clusters for BG.

- *Sampling distance*: Distance between voxels for the volume spatial downsampling step. It introduces a trade-off between segmentation accuracy and computation speed. Default value: 3.

- *Bias regularization*: constant that weights the regularization term associated to the bias field correction in the cost function. In case of low intensity non-uniformity, it should be also small. Default value: 0.0001 (very light regularization).

- *Bias FWHM*: Bias smoothness in terms of the FWHM value of the Gaussian filter. It encodes the limit between intensity variations due to the bias field (low frequency) and due to the different tissues (high frequency). Default value: 60 mm.

- *Warping regularization*: constant that weights the warping regularization term in the cost function. The larger the value, the higher the penalization to large warping parameters. Default value: 4.

- *Affine regularization*: Initial registration between volumes and templates. Default value: 'ICBM space template - European brains'.

Although, the image registration and bias field correction have not been directly changed from the original method, these steps are also analyzed in order to check that the modification of the mixture parameters equations does not decrease their performance.

5.3.1 Original data

The T_1 and T_2 MR images are scanned by a 3T scan with a final resolution of $\sim 1mm$ isotropic voxels. They are recorded at the DRCMR [53] of Hvidovre Hospital. The Figure 5.1 presents some slices of the original data from the subject `f4395`, which is the one used in this section. More slices are presented in the Appendix E.1.

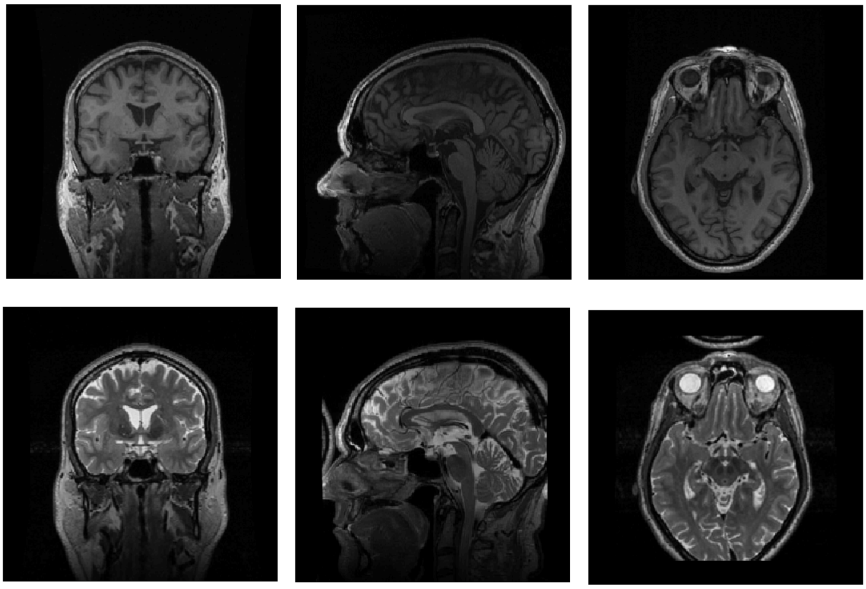


Figure 5.1: Original MRI brain volumes of the subject `f4395`. The top row contains the T_1 modality, and the bottom row has the T_2 modality. The planes of each column correspond to coronal, sagittal and transverse.

In the T_2 images, it can be seen that the back part of the head is placed in front of the nose. This effect is due to the *field shim*, which is a magnetic field inhomogeneity generated by the ferromagnetic coil. This perturbation introduces errors in the circular k-space needed to reconstruct the image from the Discrete Fourier Transform (DFT). In order to avoid this effect, the scanner must be correctly calibrate before the scan in a process called *shim correction*, which can be active or passive. This error reduces the performance of the prior template regularization, as the structures differ.

In the previous chapter, it was stated that both modalities must be aligned, which implies that it is needed a previous step to register T_1 and T_2 in the same space. This pre-processing step is done at the DRCMR, and the result is presented in the Figure 5.2. In addition, the brain volumes are normalized into the ICBM/MNI space. They can be compared with the phantoms of the Figure 5.9, which are originally created in this space. The position, shape and intensities look alike; therefore, the brain volumes seem correctly aligned and normalized. This previous normalization of the images implies a soft wrapping of the prior templates. Besides, it can be seen in this figure the effect of the shim artefact after the registration.

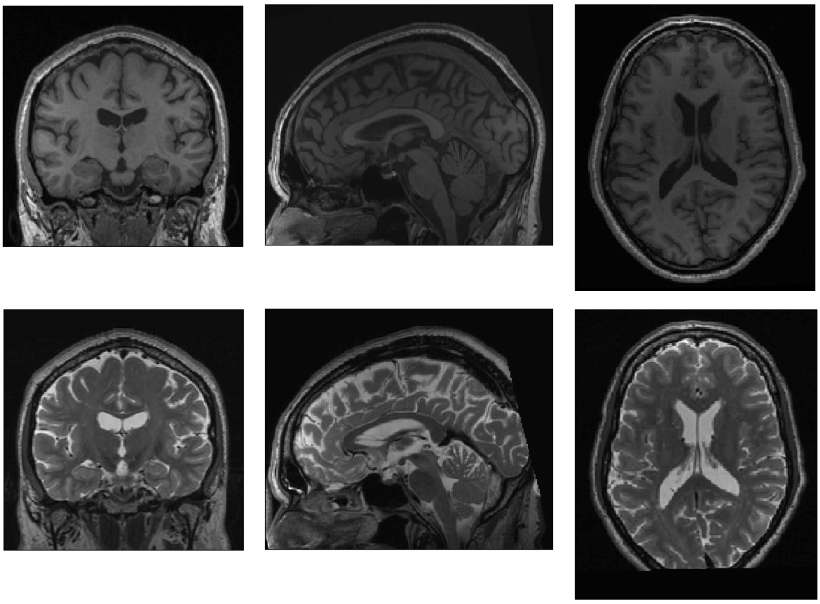


Figure 5.2: Registered brain volumes of the subject `f4395` in the ICBM/MNI space. The top row contains the T_1 modality, and the bottom row has the T_2 modality. The planes correspond to coronal, sagittal and transverse.

These volumes are segmented with the original (*Seg*) and the four versions of the modified method (*SegT1T2*) with default parameters. The main results are presented in the following points.

5.3.2 Convergence

First, it is analyzed the convergence of the methods in terms of the needed number of iterations until the stopping criterion is satisfied. Besides, the log-likelihood value is also analyzed, which measures how well the parameters fit the generative model. The parameter set includes mixture, registration and bias-field. For the different methods, the Table 5.2 collects the final log-likelihood value and the number of iterations until convergence.

Table 5.2: Performance of the original (*Seg* toolbox) and the modified methods (*SegT1T2* toolbox) in the segmentation of the brain from the subject `f4395` with default parameters. The results are expressed in terms of the final log-likelihood value and the number of iterations until convergence.

	updating	initialization	log-likelihood	#iterations
Original	-	-	$-1.8405 \cdot 10^6$	83
Modified (v.1)	slow	original	$-1.8446 \cdot 10^6$	130
Modified (v.2)	slow	modified	$-1.8426 \cdot 10^6$	123
Modified (v.3)	fast	original	$-1.8405 \cdot 10^6$	175
Modified (v.4)	fast	modified	$-1.8419 \cdot 10^6$	175

The original method fits slightly better the model parameters than most of the modified versions. However, the differences are really small, and this value does not show directly the quality of the real segmentation but gives an idea about how well the method converges. On the other hand, one result that is more evident is the number of iterations. The original method needed much less number of iterations, which in practice implies a smaller computation time.

The fact that the modified methods need more iterations to converge implies that the improvement in each iteration is smaller. As the stopping criterion is based on the log-likelihood difference between two consecutive iterations, there are more chances to stop the optimization before it reaches the optimal point because the fitting process is too slow. Besides, it seems contradictory that the modified versions with 'fast' propagation (v.3 & v.4) needed more iterations than the versions with 'slow' propagation (v.1 & v.2). However, they finally achieved a good fitting with even better log-likelihood values, which could imply a bad initialization (far from the local minimum) but a good updating (proper cost-function minimization).

The Figure 5.3 presents the evolution of the log-likelihood values at each iteration for the original (red line) and the modified methods (blue or black lines). The blue color corresponds to the modified versions with 'slow' value propagation (v.1 & v.2), while the 'fast' propagation versions are in black (v.3 & v.4). The solid lines represent the versions with original starting equations (v.1 & v.3), while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization (v.2 & v.4).

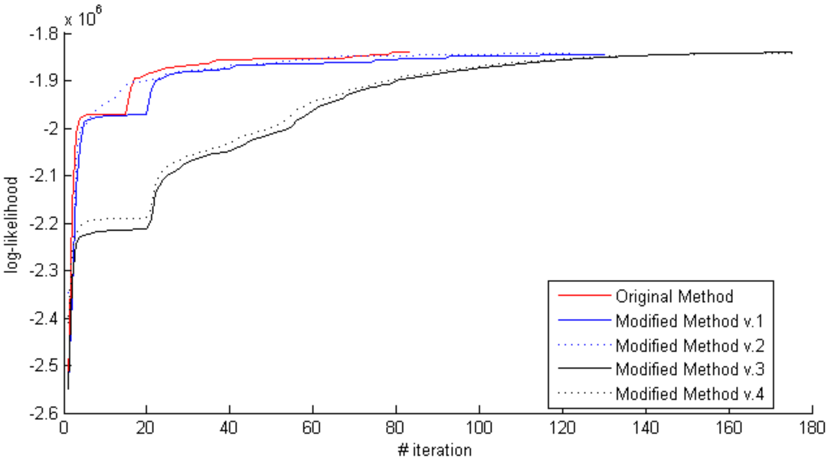


Figure 5.3: Log-likelihood value at each iteration for the original method (Seg toolbox), and the four versions of the modified method (SegT1T2 toolbox). The red line corresponds to the original method. The blue color corresponds to the modified versions with 'slow' value propagation, while the 'fast' propagation versions are in black. The solid lines represent the versions with original starting equations, while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization.

The first 15/20 iterations correspond to the initialization step, which implies much worse log-likelihood values. If the initialization equations cannot improve further the log-likelihood value, the method starts using the updating equations. This criterion makes that each method finishes the initialization at different epoch, which can be seen in the plot as a jump. The version 2 (dotted blue line) is the fastest in the beginning, but then it has problems in the last iterations to minimize the cost function. The initialization step (iterations 1-20) for versions 3 and 4 places them in a worse starting situation, although they have a stable evolution that reaches a good fitting in the end. The versions with modified initialization equations (dotted lines) are slightly faster than with the original initialization equations (solid lines) in all the cases. Although the version 3 of the modified method is slow, the final fitting is very good; therefore, the rest of this section will show the results for this method.

5.3.3 Mixture Parameters

The segmentation algorithm is based on the GMM, where each cluster (Gaussian) is parameterized by the mixture parameters. In the Table 5.3, it is gathered the final estimated values for the original and the version 3 of the modified method, which is presented with *(*)*. In addition, in the Appendix D.1, it is presented the evolution of the mixture parameters at each iteration for the original and all the modified methods.

Table 5.3: Values of the mixture parameters after the segmentation of the MRI brain scans from the subject `f4395`. The applied segmentation algorithms include the original and the version 3 of the modified method *(*)*.

Class name	GM		WM		CSF	
Class number (lkp)	1	1	2	2	3	3
Mixing factor (mg)	0.7041	0.2959	0.5147	0.4853	0.6665	0.3335
Mixing factor (mg)*	0.6891	0.3109	0.6190	0.3810	0.6699	0.3301
Mean T_1 (μ_{T1})	328.27	276.58	393.75	417.09	215.78	141.09
Mean T_1 (μ_{T1})*	402.24	347.19	485.31	509.57	264.54	173.48
Mean T_2 (μ_{T2})	180.65	248.59	148.87	137.55	253.76	411.49
Mean T_2 (μ_{T2})*	310.88	415.01	252.79	232.38	460.47	715.15
Variance T_1 (σ_{T1})	662	3226	518	154	2848	400
Variance T_1 (σ_{T1})*	976	4739	774	192	4280	604
Variance T_2 (σ_{T2})	399	2733	392	117	10725	1109
Variance T_2 (σ_{T2})*	1171	7247	1011	261	29565	3538
Covariance (σ_{T1T2})	-217	-1630	-116	-57	-1394	-159
Covariance (σ_{T1T2})*	-393	-3130	-238	-90	-2379	-319

As it was previously stated, the order of the clusters inside each tissue class is random. In addition, one cluster for one method can model the intensities associated to several clusters by other method. Therefore, it cannot be directly done a comparison one by one. The level of intensity inhomogeneity correction also affects these cluster parameters, and this discrepancy of values between original and modified method is due to this effect.

It is seen that all the presented tissues have a negative covariance, which implies that larger T_1 intensity values are associated with smaller T_2 intensity values, as it is explained with more details in the Appendix B.1. This result is something expected, and it was previously commented in the Section A.2, where the properties of the MRI scan were presented.

In the Figure 5.4, it is presented a zoomed representation of the clusters done by the original method and the version 3 of the modified method. The Gaussians are parameterized by the values of the Table 5.3. Besides, in the Appendix D.2, it can be found the zoomed out representation with all the tissue classes done by the four modified methods. In the presented figure, it can be seen that the patterns for GM and WM are similar, although the position and variances are bigger for the modified method. As it was previously explained, the main reason to this effect is the different bias field correction, which is stronger for the modified method. In both methods, the clusters of GM and WM are not greatly overlapped. However, the ST class overlaps them, which can only be corrected by the spatial information from the prior templates. Finally, the CSF has a big overlap over GM, as it was expected due to their similar intensity patterns. The large clusters (high variance) indicates that the method does not gain any information from them, thus they could be removed.

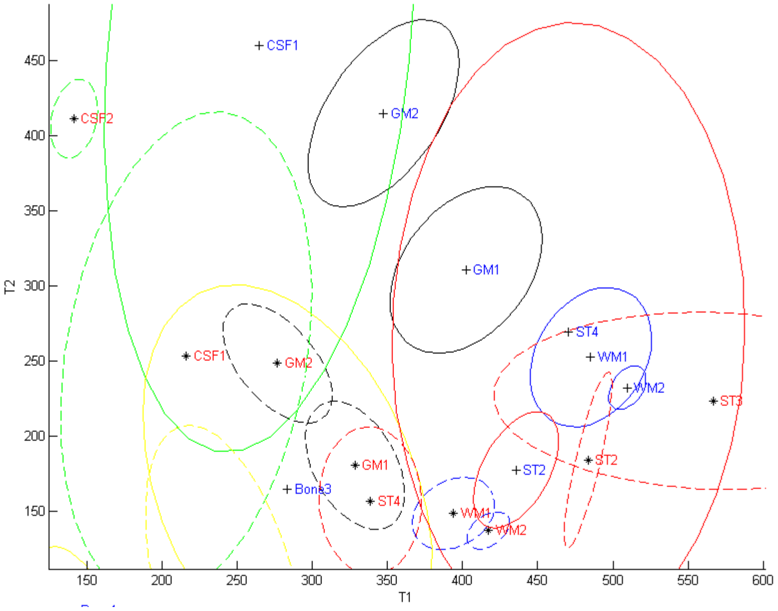


Figure 5.4: Zoom of the cluster representation done by the original and the version 3 of the modified method. The lines correspond to the contour of the Gaussians cut at FWHM, and weighted by the mixing coefficient. The contours of the clusters done by the original method are presented with dotted lines, the centers with the symbol *, and the text labels in red. The version 3 of the modified method presents the contours with solid lines, the center with the symbol +, and the text labels in blue. The depicted tissues comprise GM (black), WM (blue), CSF (green), ST (red), bone (yellow), and BG (magenta).

5.3.4 Bias Field Correction overview

In this point, it is analyzed the correction of the slow intensity inhomogeneities. Each modality is affected by a different perturbation pattern, therefore the bias field is modeled independently for T_1 and T_2.

The Figure 5.5 presents the results for the version 3 of the modified method with default parameters on the brain `f4395`. The six columns correspond to the three planes (coronal, sagittal and transverse) for T_1 and T_2 modalities. The top row contains the original volumes, and the middle row presents the volumes after the intensity inhomogeneity correction. Finally, the bottom row represents the estimated bias field. In this figure, the images have been intensity scaled and re-sized for visualization reasons.

In the figure, it can be seen two examples of how the bias field modulates the intensity. The left-most circle presents an area of the brain than it was acquired darker than it should be, while the right-most circle presents the opposite. In addition, it can be seen how the central WM of the T_2 volumes is corrected to be more white. This increase in the brightness of the volumes could be the reason why the mean and variances are much higher than in the modified method.

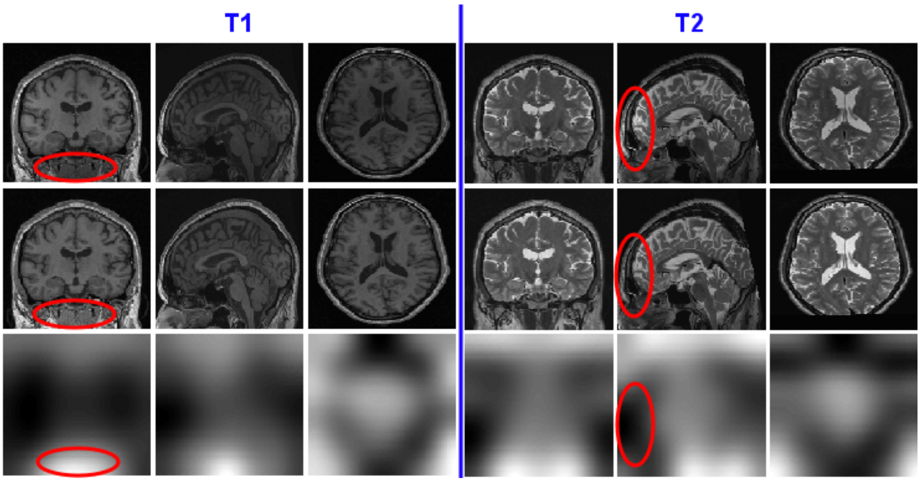
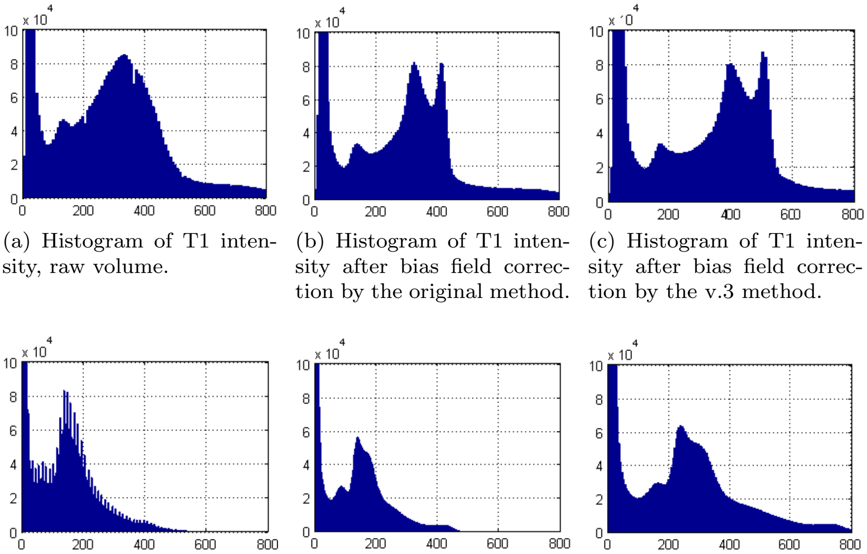


Figure 5.5: Bias Field correction for both T_1 and T_2 modalities. The top row contains the original volumes for T_1 and T_2 in the three planes. The middle row presents the volumes after the intensity inhomogeneity correction, and the bottom row represents the estimated bias field.

The Figure 5.6 presents the intensity histogram of the volumes before and after the bias field correction for both modalities and methods. The presentation is zoomed into the brain voxel intensities, and the big peak of dark intensities, which correspond to BG voxels, is cropped. After the correction, the expected Gaussian for each tissue appear more narrow and with less overlap among them. Thus, it is easier to split them up. In addition, the intensity distributions are more smooth and spread afterwards.

It can be seen that the modified method applies a stronger non-homogeneity correction that shifts all the intensities to higher values (more bright). This effect explains why the obtained cluster parameters have much higher mean, variance and covariance values for the original method. Besides, there are more voxels that are shifted from intensities of BG to intensities of brain tissues, which makes that the number of brain voxels increases.



(a) Histogram of T1 intensity, raw volume.

(b) Histogram of T1 intensity after bias field correction by the original method.

(c) Histogram of T1 intensity after bias field correction by the v.3 method.

(d) Histogram of T2 intensity, raw volume.

(e) Histogram of T2 intensity after bias field correction by the original method.

(f) Histogram of T2 intensity after bias field correction by the v.3 method.

Figure 5.6: Effect of the bias field correction in terms of the intensity histogram. The results corresponds to the segmentation of the T_1 and T_2 MRI scans from the subject `f4395` done by the original and the version 3 of the modified method. The units of the y-axis correspond to the number of voxels, and the units of the x-axis are the intensity values. All the histograms are built with 300 bins of the same size.

5.3.5 Segmentation overview

Brain extraction: The 'New Segmentation' method includes an extended set of templates with additional no-brain tissues in order to characterize better Bone and ST voxels. The proposed modification takes advantage of it, and also includes these extra tissue classes that helps in the scalp stripping process. The results of the Figure 5.7 shows a good performance, although some isolated voxels that are classified as brain are out of the cranium volume.

Tissue segmentation: In the Figure 5.7, it is presented the central slices of the obtained probability maps for GM, WM and CSF. The segmentation is done by the version 3 of the modified method with defaults parameters. From the presented figures, it can be seen that the result of the process seems right, although it is not yet possible to determine the quality. In the Appendix E.3, it can be found more slices of these generated volumes. The bottom row of this figure depicts an overlapped representation of the generated TPM. It is hardly see any important overlap among tissues, which implies that the probability values are close to one or zero.

Comparison with baseline: The segmentation result for the modified v.3 method can be compared with the original segmentation baseline, i.e. SPM5 + VBM5. The segmentation for the subject `f4395` done by the original baseline was presented in the Figure 1.4 and 2.7, and the segmentation for the v.3 method is presented in the Figure 5.7. For a clear visualization of the comparison, the Figure 5.8 presents the overlap of tissue volumes between the original baseline (red) and the modified v.3 method (green). The former detects more GM voxels in the occipital lobe, in the lower part of the brain stem, and in the utter part of the parietal lobe. However, the latter method finds in general more GM voxels in the brain, which are classified as CSF by the original baseline. The version 3 of the modified method finds more WM voxels in the brain stem, cerebellum and inner part of the occipital lobe. Although it cannot be seen in this plot, the v.3 method classifies correctly the right eye muscle as ST, instead of as CSF like the original baseline does. The spinal cord is more expected to be WM, as a continuation of the brain stem; however, the original baseline classifies it as GM. The segmentation of the cerebellum is hard, because the cortex has less than $1mm$ of width, smaller than the voxel resolution; therefore, PVE problems arise. However, without a ground truth segmentation is not possible to estimate objectively the quality differences between both methods, although some segmentation errors were solved by the modified method.

Apart from the visual inspection of volumes and parameters done in this section, the next section addresses the quality of the modified method with a more objective approach.

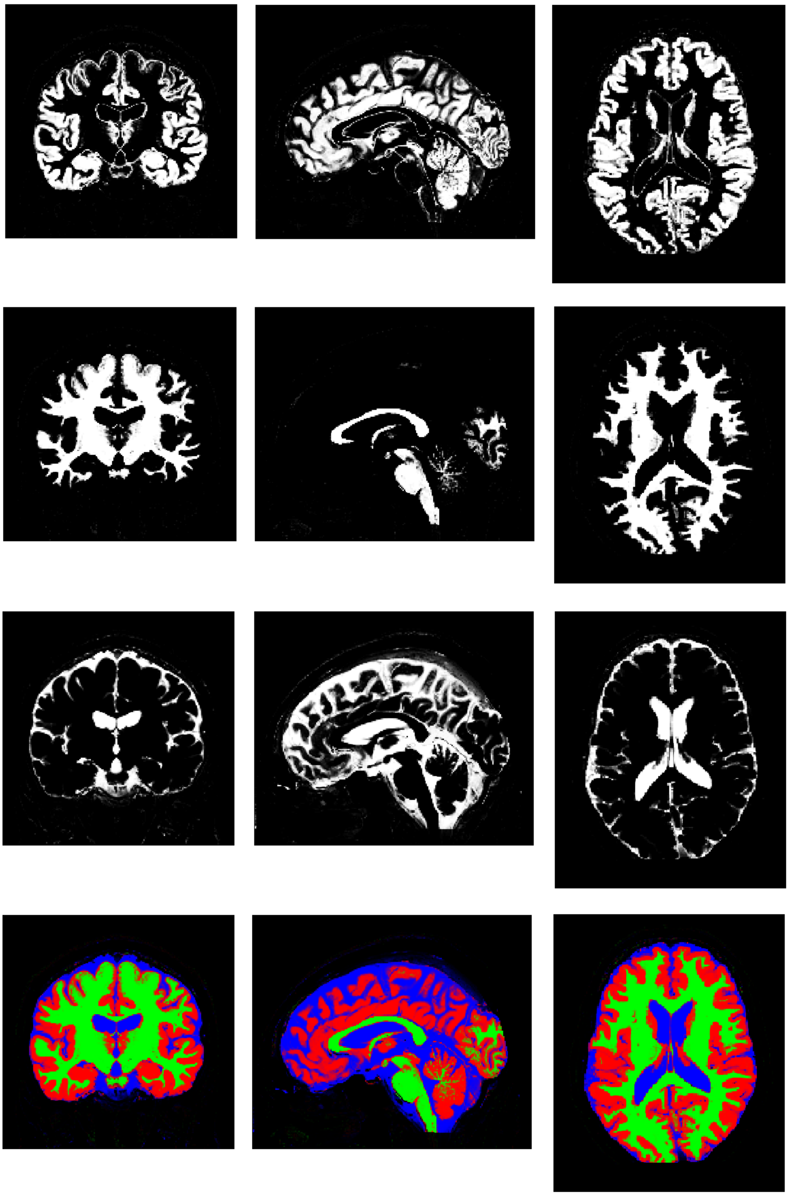


Figure 5.7: Probability and overlapped segmented tissues of MRI data from subject `f4395`. The first three rows correspond to GM, WM and CSF, respectively. The bottom row is a representation of the overlapped tissues for GM (red), WM (green) and CSF (blue). The color of each pixel is an RGB combination weighted by the associated tissue probability. The images correspond to the coronal, sagittal and transverse planes of the central brain slices.

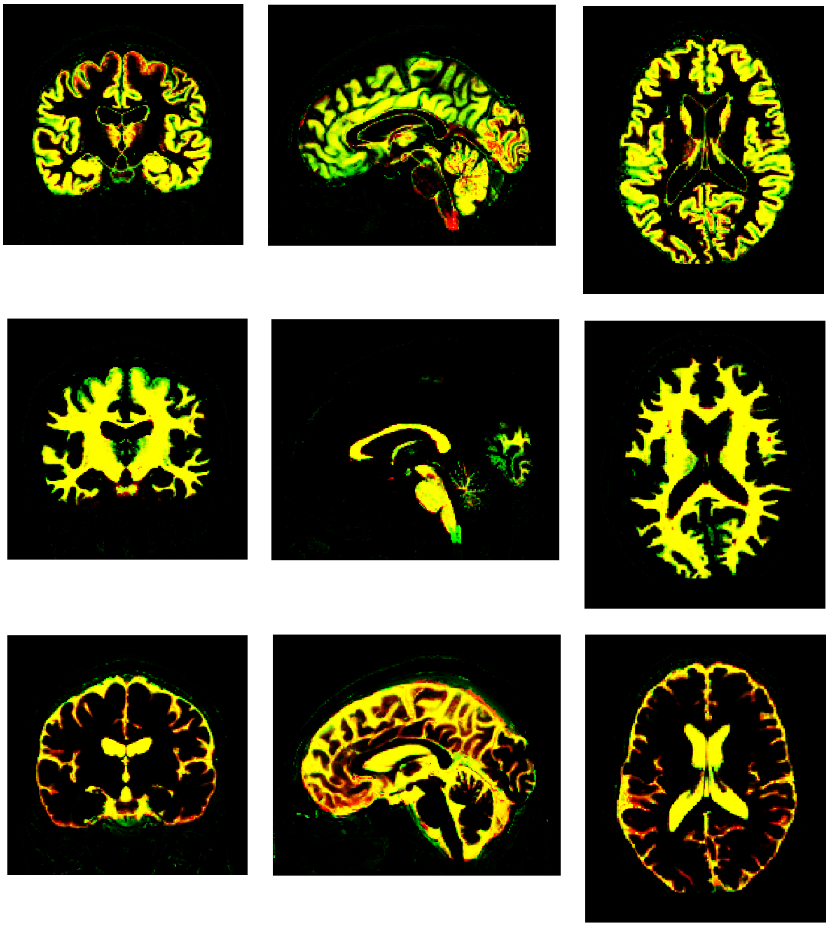


Figure 5.8: Overlapped probability maps generated by the original baseline (SPM5 + VBM5) and the version 3 of the modified method for the MRI scans from subject `f4395`. The rows correspond to GM, WM and CSF, respectively. The color of each pixel is an RGB combination between the probabilities from the original baseline (red) and the modified v.3 method (green). The images correspond to the coronal, sagittal and transverse planes of the central brain slices.

5.4 BrainWeb phantoms - Dice Score

In the previous section, it was analyzed the results of the segmentation in a general way, i.e. visualizing the log-likelihood value, the mixture parameters and the generated probability maps. In this section, it is studied the segmentation quality of the modified methods with respect to the original method. In order to compare automatic segmentation methods, it is used a set of MRI brain volumes where the ground truth is previously known. These data correspond to the *BrainWeb* phantoms [55], which belongs to the Simulated Brain Database (SBD) done by the McConnell Brain Imaging Centre (BIC) of the Montreal Neurological Institute (MNI) [12] [13] [39] [40].

In this case, it is used volumes from the Normal anatomical model with dimensions 181 × 217 × 181 and $1mm$ isotropic resolution. The T_1 and T_2 modalities are used with different noise levels: 3% and 9%, which are calculated from the brightest tissue. The level of intensity non-uniformity is fixed to 20%. There is the possibility to set this paramter to 0% or 40%. However, it is not the goal of this thesis to analyze in details the bias field correction; thus, a single middle value of perturbation level is chosen. In addition, if this value is set to zero, the default parameters of the toolbox must be changed because the method expects some level of bias field distortion, i.e. if there is not smooth intensity non-homengeity in the scans, the method will try to fit it anyway (unless other parameters are used), which will generate errors.

The Figure 5.9 and 5.10 depicts some slices of the BrainWeb phantoms for the two levels of noise. It can be seen that there are some artifacts above the head that will challenge the segmentation. These slices can be compared with the brain volumes from the subject `f4395`, and it can be seen that the phantoms are quite realistic brain MR images, although it is hard to say how well they describe the reality.

In this section, the BrainWeb phantoms are segmented by six methods in order to study their different performance, and all of them are executed with default parameters. The analyzed methods comprise:

- Original method (*Seg*) with just T_1 modality.
- Original method (*Seg*) with T_1 and T_2 modalities.
- Modified method (*SegT1T2*), version1, with T_1 and T_2 modalities.
- Modified method (*SegT1T2*), version2, with T_1 and T_2 modalities.
- Modified method (*SegT1T2*), version3, with T_1 and T_2 modalities.
- Modified method (*SegT1T2*), version4, with T_1 and T_2 modalities.

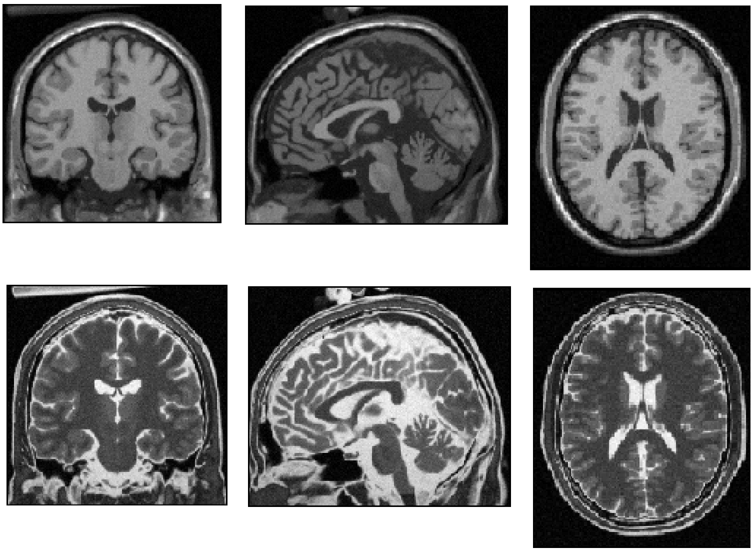


Figure 5.9: BrainWeb phantoms for T_1 and T_2 modalities with 3% level of noise and 20% of intensity non-uniformity level.

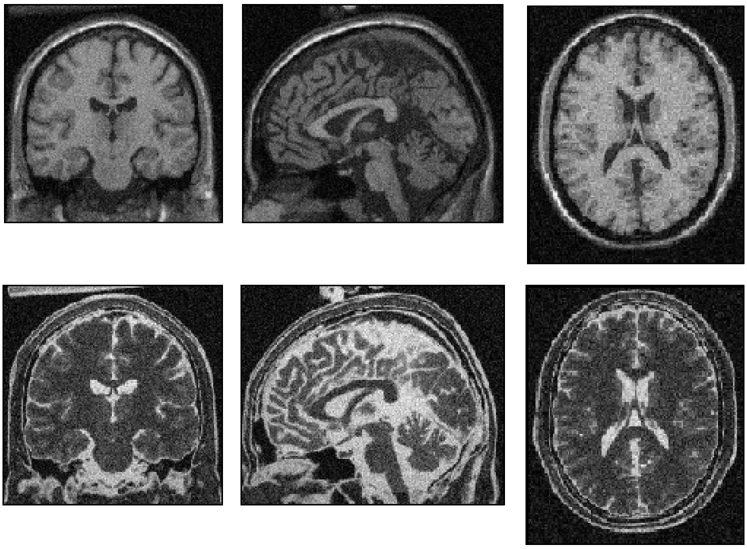


Figure 5.10: BrainWeb phantoms for T_1 and T_2 modalities with 9% level of noise and 20% of intensity non-uniformity level.

The original tissue classes for each voxel of the simulated MRI data are available in a set of probaility maps called *atlas*. Hence, it is possible to compare the results of the different segmentation methods with respect to a ground truth. The chosen metric to evaluate the performance corresponds to the Dice Similarity Coefficient (DSC) [18] [92]. This coefficient is widely used in the quality evaluation of segmentation algorithms [67] [68] [82]. The generated result is a number between 0 and 1 for each label (tissue class), where the perfect segmentation gives a value of 1, and the random classification gives values around 0.5. This coefficient is related to the *Jaccard index* with the expression: $J = D/(2 - D)$. The DSC is a measurement of the similarity (overlapping) between two groups (volumes), and it does not give further information about which is the source of errors in the segmentation, either type I or II.

In the Equation 5.4, it is presented the expression to obtain the Dice Coefficient for the tissue class k. The variables A_k and B_k represent two groups of voxels. The former includes the voxels that are classified as the kth-class by the method A, and the latter includes voxels classified as the kth-class by the method B. The symbol $\cap$ stands for the intersection of two groups and $|.|$ for cardinality, Therefore, the expression $|A_k \cap B_k|$ refers to the numbers of voxels classified as the kth-class by both methods at the same voxels. In the second term of this equation, the Dice Coefficient is presented in terms of the confusion matrix elements, i.e. True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). These values are obtained after applying majority voting (or thresholding) to the TPM's generated by each method.

$$Dice_k = 2\,\frac{|A_k \cap B_k|}{|A_k| + |B_k|} = 2\,\frac{TP_k}{(TP_k + FP_k) + (TP_k + FN_k)} \tag{5.4}$$

In this case, the group A is obtained from the segmentation done by any of the automatic algorithms, and the group B includes the true labels for each voxel (atlas). Therefore, for each class and each dataset, six DSC values are obtained, which correspond to the 2 original methods and the 4 modified methods. As it was previously stated, two datasets are used with two different noise levels. Finally, the tissues are grouped into 6 classes, which are GM, WM, CSF, Bone-ST, BG and total.

Once the TPM's are obtained with the segmentation methods, they are converted into binary maps, i.e. $[0,1] \rightarrow \{0,1\}$. These maps have either one or zero values, which represents tissue or not at each voxel. In this case, the Majority Voting technique is applied. The process can be seen in the Figure 5.11 for the sagittal plane of the GM tissue maps. The slices of the middle column have a noise reduction with respect to the first column. Besides, the last column presents the probability change at each voxel with the overlapping of the two previous columns, where yellow voxels indicate almost not change of probability, red voxels show a decrease, and green voxels an increase. In the case of the atlas, there are more probability changes (red/green voxels) after Majority Voting than for the segmentation methods. This fact implies that the generated TPM's have values closer to either 1 or 0, which could imply problems to deal with PVE. In the Appendix D.4, all the results are presented.

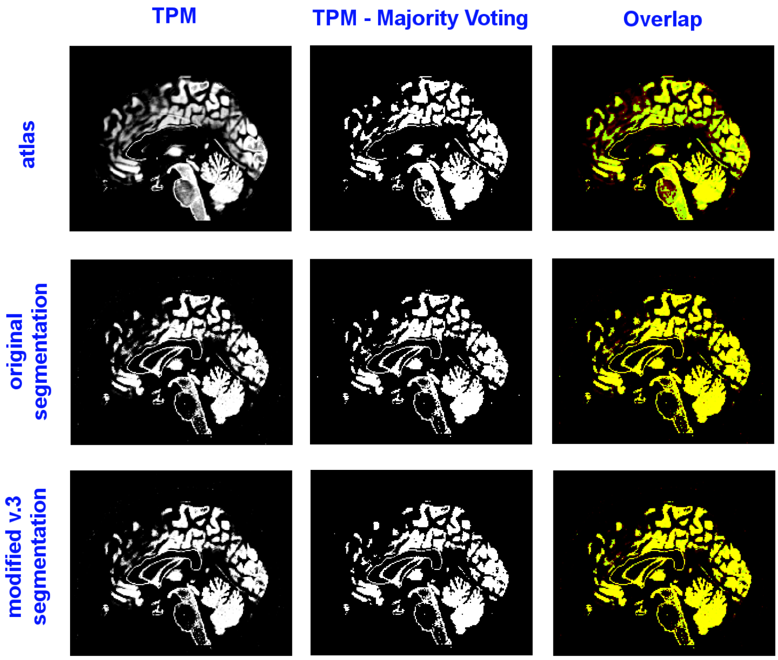


Figure 5.11: Majority Voting process for the original and modified v.3 segmentation method. It is depicted the sagittal plane of the GM tissue maps. The dataset correspond to the BrainWeb phantoms for T_1 and T_2 modalities with 3% level of noise and 20% of intensity non-uniformity level. The overlapped representation in the last column depicts in yellow the voxels that barely changed their values, in red the voxels that are not finally classified as tissue although thay had a high probability value, and in green the voxels that are eventually labelled as the tissue class without a high probability value.

After the segmentation and the majority voting process, the binary tissue maps are voxel-wise compared to the atlases with the true tissue labels. This comparison produces the confusion matrix values (TP, TN, FP and FN) that are combined to generate the final DSC for each class and method.

The Figure 5.12 shows the performance of the segmentation methods in terms of the Dice score for the BrainWeb phantoms with noise level of 3% and 9%. Each colour corresponds to a different tissue class, namely GM, WM, CSF, Bone-ST, BG and total. And for each class, the results of the six methods are presented in the same order. The complete data of the results is included in the Appendix D.3, where three tables are presented with the results for 0%, 3% and 9% of noise level when applying majority voting and thresholding with values 0.7 and 0.9.

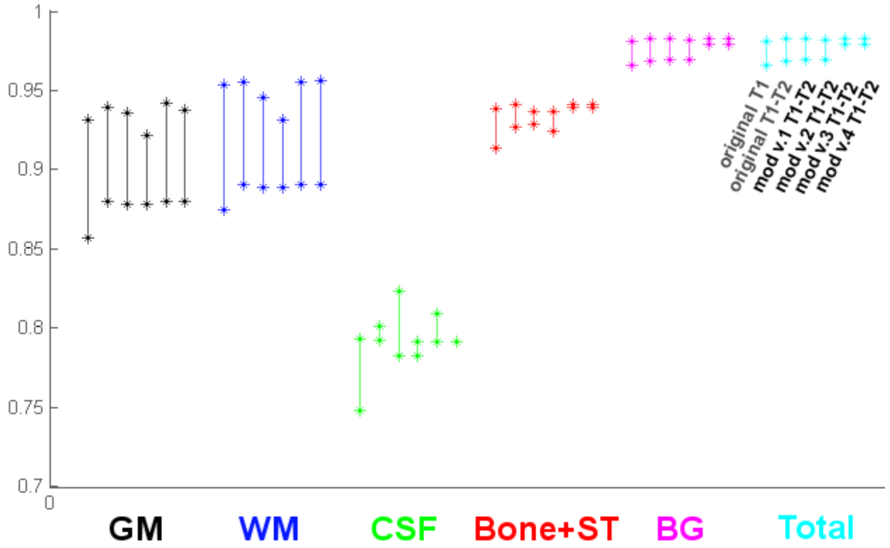


Figure 5.12: Comparison of six segmentation methods in terms of the Dice score after applying majority voting on the probability maps. The methods includes: original method with T1, original method with T1 and T2, modified method (ver.1) with T1 and T2, modified method (ver.2) with T1 and T2, modified method (ver.3) with T1 and T2, and modified method (ver.4) with T1 and T2. The segmentation is analyzed for the labels: GM, WM, CSF, Bone-ST, BG and total. The two dots for each type are associated with the BrainWeb phantoms with 3% and 9% of noise level.

Noise analysis: In all the cases, the results with 9% of noise give worse performance (smaller Dice coefficient value) than for a level of 3%. This result is expected becasue the visual inspection of the Figures 5.9 and 5.10 already showed the strong quality degradation in the volumes when the noise level is high. The uni-modal original method is less robust and scores much worse with important levels of noise. Besides, the versions 3 and 4 are very stable in the detection of ST, Bone and BG, where they show good results even on noisy scans. The results for the last class is not too much representative because it is biased by the results of the BG, which is the class with the biggest number of voxels.

Multispectral analysis: The comparison between the original method with one (T_1) or two modalities (T_1-T_2) shows that the multispectral approach achieves better results for the segmentation of all the classes, specially in the classification of CSF voxels.

Modified methods analysis: The version 1 of the modified method scores just better than the rest in the detection of CSF with low level of noise, while the version 2 scores worse than all the multispectral methods for all the cases. The version 3 and 4 of the modified method have similar results than the multispectral original method, except in the case of CSF where the version 3 scores much better. As it was seen in the cluster representation of the Figure 5.4, there is an important overlap between GM and CSF intensities; thus, a better classification of the latter will improve the classification of the former. However, in practice the modified v.3 scores just slightly better than the original in the detection of GM.

As a compendium of the previous results, it can be stated that the modified v.3 presents a Dice score slightly higher than the original method for all the classes. Besides, in the case of CSF, the performance is much better; and in the case of non-brain tissues, the method is very stable. Therefore, it can be stated that **the version 3 is the final version of the modified method**, which is slightly more accurate and robust than the original multispectral method.

The final part of this section compares visually the version 3 of the modified method with the original method in order to analyze from where the improvements come. Namely, it is presented one example for GM and other for CSF. However, the results are very similar between both methods. It is not presented and example for WM because the differences between both method in the segmentation of this tissue are not significant.

The Figure 5.13 shows the segmentation results of the T_1 and T_2 MRI brain volumes of the BrainWeb phantoms with 3% level of noise. The colours are associated with the confusion matrix elements: yellow (TP), black (TN), green (FP), and red (FN). It can be seen that both methods fail in the detection of the corpus callosum border (green) and the main part of the brain stem (red). The problem with the corpus callosum is that it is composed by pure WM tissue, and it appears very brigth in the T_1 scans. Therefore, the border voxels between the corpus callosum (bright) and the water (dark) combines both intensity contrasts, which implies a PVE. The brain stem is composed by WM, although several detection methods missclasify it as GM. In the figure, it is zoomed an area where the modified v.3 has less number of FP (green) and FN (red) in an interface between GM and CSF.

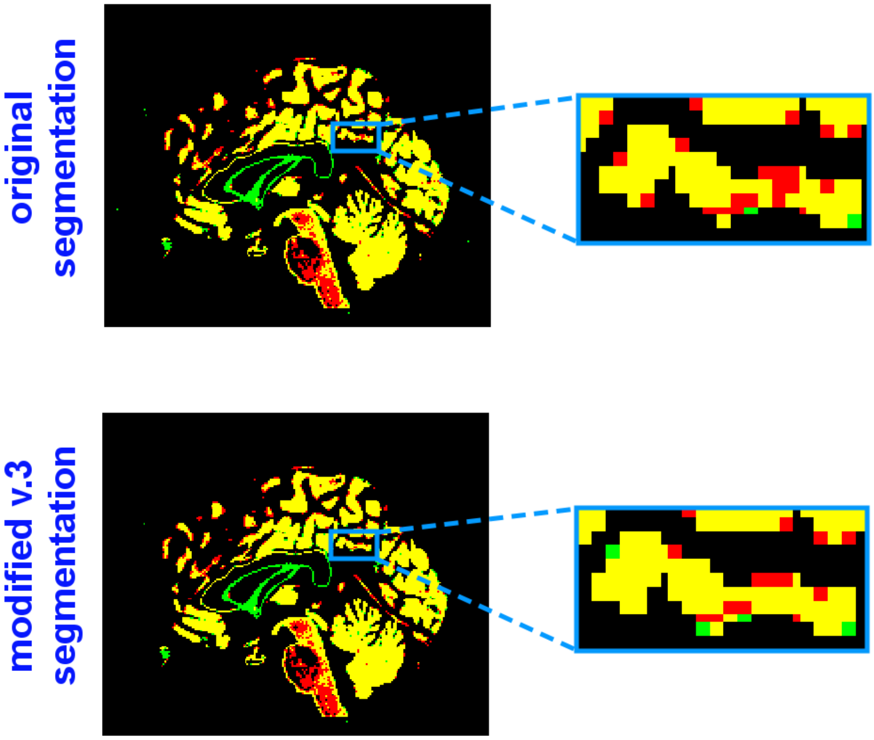


Figure 5.13: Example of the improvement of the version 3 of the modified method with respect to the original one in the segmentation of GM. The voxels present the segmentation results of the T_1 and T_2 MRI brain volumes of the BrainWeb phantoms with 3% level of noise. The colour is associated with the confusion matrix elements: yellow (TP), black (TN), green (FP), and red (FN).

In the Figure 5.14, it is presented a similar representation for the detected CSF voxels. Although, the Dice score is better for the modified v.3 method than for the original method, the visual differences are small. The main misclassification done by both methods falls outside of the brain. There are three big green areas on the top of the image that shows that these voxels are wrongly detected as CSF. Besides, the bottom of the slices shows more red points, which implies that the methods were not able to detect CSF. The explanation for both errors could come from a strong bias field correction. It seems that the bias field correction tries to compensate the artefact above the head that was presented in the Figures 5.9 and 5.10. In addition, it is presented another example where the modified v.3 classified better the CSF tissue between both hemispheres.

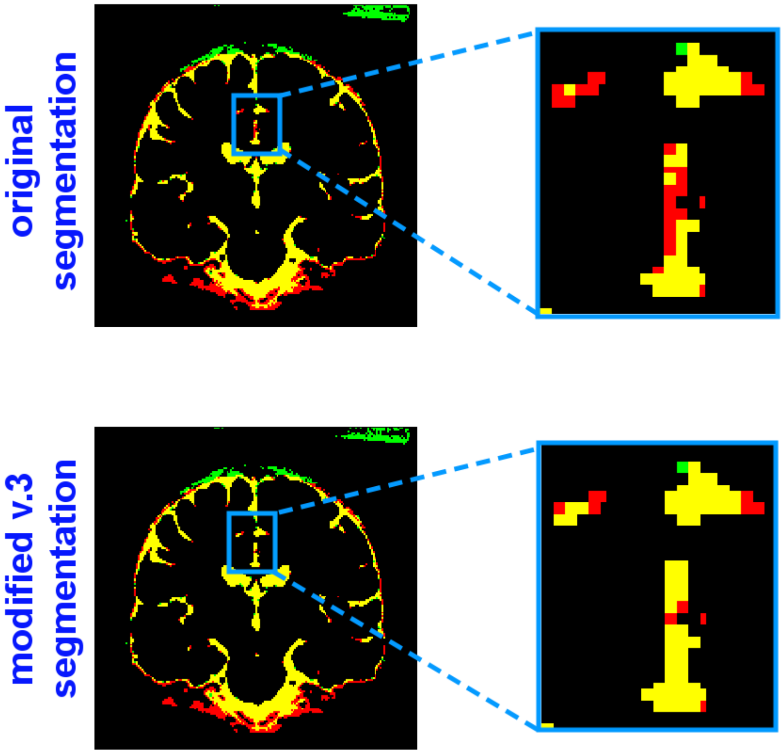


Figure 5.14: Example of the improvement of the version 3 of the modified method with respect to the original one in the segmentation of CSF. The voxels present the segmentation results of the T_1 and T_2 MRI brain volumes of the BrainWeb phantoms with 3% level of noise. The colour is associated with the confusion matrix elements: yellow (TP), black (TN), green (FP), and red (FN).

5.5 CIMBI database - Age-Profile

The MR images can greatly vary with different subjects and scanning protocols. Therefore, the performance results obtained with the BrainWeb phantoms cannot be directly extrapolated to other datasets, i.e. the evaluation of a method is specific to the used dataset. Another proposed validation approach is based on the expected brain atrophy through the years [76]. There are research lines that have studied the decrease of GM and WM in the human ageing from post mortem [62] and in vivo brains [33] [29]. Therefore, a segmentation algorithm applied to the scans of patients of different ages should show the same characteristic age-profile.

In the Appendix D.5, it is presented a discussion about the effects of the ageing in the brain, i.e. brain atrophy. As a conclusion, it is stated that **the brain ageing can be characterized by a subtle decrease of GM and WM volume, and a significat increase of CSF**. In case that the segmentation shows a severe decrease in the neocortical thickness (GM volume), it could be assumed that the brain is affected by a disease that induces brain changes not connected to the aging itself. As the dataset of this thesis only includes healthy brains, it is not expected big GM variations. In addition, brain lesions in the WM tissues have more water; thus, some automatic segmentation methods missclassify these areas as GM due to the PVE.

The Figure 5.15 presents the volume estimation for 6 subjects of the CIMBI project. The cohort includes 4 females (26, 30, 54 and 64 years old) and 2 males (53 and 56 years old). These brain have been segmented with default parameters by the original and the version 3 of the modified method into GM, WM and CSF. The volumes of each class are normalized by the ICC volume of each subject. Besides, at the end of the Appendix D.5, it is presented a simple linear regression analysis of the volume age profile of these subjects.

The results show a small decrease of GM, an increase of CSF, and constant value of WM. In addition, there is not significant differences between both methods. However, due to the small used dataset, it is not possible to infer any further conclusions.

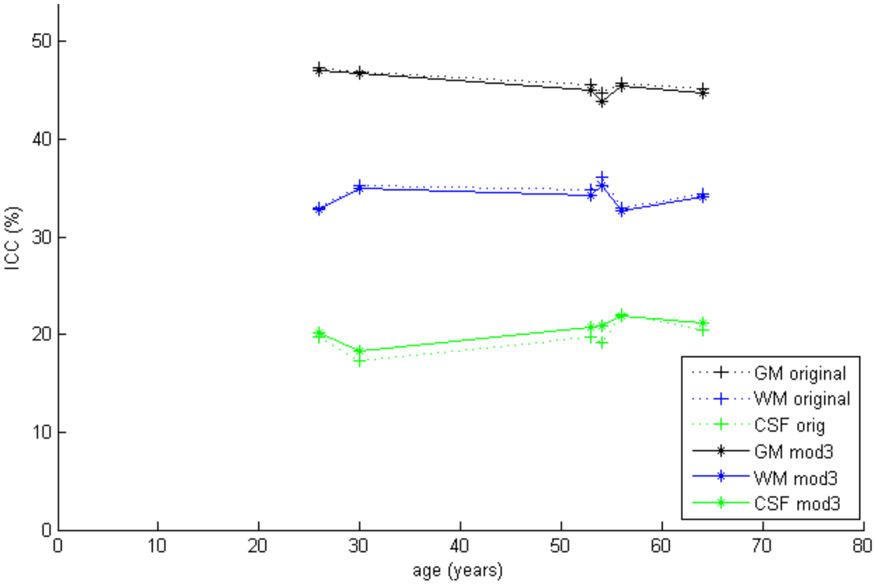


Figure 5.15: Volume age profile for GM, WM and CSF generated from the the segmentation of six volumes. The methods applied correspond to the original and version 3 of the modified method with default parameters. The tissue classes corresponds to GM, WM and CSF, which are depicted in black, blue and green, respectively. The volumes of each class are normalized by the ICC volume of each subject.

Chapter 6

Discussion

This final chapter presents a resume of the the main conclusions gathered from the literature study and the obtained segmentation results. Besides, several ways are proposed to improve the performance of the modified segmentation method developed in this thesis.

6.1 Resume

In the beginning of this report, it was analyzed the SPM implementation for MRI brain segmentation, which is based on a MoG model. In this implementation, the mixture parameters that characterize each cluster are iteratively optimized in order to minimize the cost function. This cost function also includes the template registration and the bias field correction. In a wide sense, it could be said that the MoG introduces intensity information and the prior templates include spatial information in the model, while the bias field correction reduces the intensity inhomogeneity perturbations. The result of the segmentation is a TPM that encodes the probability of belonging to a specific tissue class at each voxel.

This method relies on several assumptions:

- The intensity probability distribution of each tissue is modeled by one Gaussian, which is efficient because of the low number of needed parameters, but not realistic.
- The voxels are assumed to be spatially independent, thus the intensity values can be considered as i.i.d variables. This restriction is partially compensated by the inclusion of prior templates and smoothing.
- The intensity is assumed to be homogeneous within each tissue class, which is not totally true as the intensity value for each tissue varies slightly depending on its location in the brain.

It was analyzed the multispectral approach with 1D and 2D intensity histograms for several tissue classes, and it was seen that the increase of dimensionality by adding the T_2 modality allows a better separation of classes, which are less overlapped. The only drawbacks were the need of co-registering T_1 and T_2 volumes in the same space, and the increase of model complexity. Besides, the T_2 scans are usually more noisy due to moving artifacts, because the patients move inside of the scanner, which entails an increase of GM estimation. Although the contribution of bias field correction, prior templates and registration was also studied, the main focus was placed in the intensity model of SPM.

The 'New Segmentation' method of SPM8 was the starting point of this thesis, which already includes an extended set of tissue templates, the possibility of using several clusters per tissue class, and a multispectral approach. However, the method assumes non-correlation among modalities. This point was modified in this project and its development and implementation comprised the main workload of this thesis. Four different versions of the modified algorithm were created, and all of them include correlation among modalities. The differences among them are based on the kind of initialization and the type of value propagation scheme that is applied.

During the validation process, and without further knowledge about which parameter values are better for each dataset, all the methods were executed with default parameters. Two datasets were used for the validation, the first one included the scans from the DRCMR, and the second one comprised the BrainWeb phantoms for several levels of noise. Both datasets included T_1 and T_2 modalities, although only the last one had maps with the true tissue labels for each voxel. The generated TPM's by each method were converted into binary maps after applying majority voting to each voxel. The results of this process showed that the original and modified methods have almost not overlap among tissues and the probabilities at each voxel are close to either 0 or 1.

The results obtained from the segmentation of the first dataset showed that the modified methods need longer computation time than the original method; although their final log-likelihood values were similar. Besides, it was visually analyzed the generated tissue maps and the volume age-profile. The results were correct for the original and the modified methods, but without a reference it was not possible to address quality differences. It was surprising that the value of the mixture parameters obtained by the modified methods were much higher than the ones obtained by the original one. This effect is due to the strong estimated bias field correction that spreads the intensities and shifts them towards larger values, i.e. the image is brighter. It must be reminded that the bias field correction implementation is the same in the original and modified method. Hence, the interaction between original bias field estimation and modified updating equations of the mixture parameters generates this effect. Although the estimated bias field is stronger, the segmentation results are correct, which means that the intensity inhomogeneity correction behaves more like a kernel that transforms the intensities into another space where the segmentation is easier.

The second dataset includes the atlas with the true labels for each voxel, therefore it was possible to compare objectively the results. In this case, six methods were compared, namely the original method with T_1, the original method with T_1 and T_2, and the four versions of the modified method with T_1 and T_2 modalities. The Dice score showed that the multispectral approaches were better that the uni-modal one for all the situations. In addition, the best method was the version 3 of the modified method, which Dice scores were slightly higher than the original method for all the classes. In addition, the performance of v.3 method was much better for the CSF detection and much more stable for non-brain tissues than the original method. Finally, it was visually inspected the segmented volumes from original and v.3 method, which showed that the latter method is able to segment better the voxels that lie in the interface between several tissues. Therefore, the version 3 of the modified method deals better with PVE. However, both of them had problems in the detection of the brain stem and the water around the corpus callosum.

6.2 Future Work

The presented results showed that the version 3 of the modified method performs slightly better than the original method, which assumes non-correlated modalities. However, several problems are presented that give room for further improvements of the method.

The modified methods had a slower optimization rate with more needed iterations until **convergence**. One solution to this problem is the development of another set of initialization equations that give better starting value to the parameters. It was seen that versions 2 and 4, which had modified initialization equations, were much faster in the beginning than versions 1 and 3. In addition, after the initialization, the version 2 had a great convergence, although the optimization in the final iterations was poor. Therefore, it can be implemented a method that uses a new set of initialization equations during the initialization, and afterwards the updating equations of the v.2 method are used for the first iterations, and the updating equations of the v.3 method are used for the last iterations.

Another problem is the **bias field correction**, which estimation was very strong. Although the segmentation results were right, its behaviour was surprising because it spread widely the intensity distributions when the modified updating equations were used. In addition, for the case of the BrainWeb phantoms, it was seen that original and modified methods classified wrongly some voxels out of the head. Although a better masking could avoid these problems, the reason of classifying voxels out of the head as brain tissue is due to a strong bias field estimation. Therefore, it would be needed to review the implementation of the bias field correction.

The subject cohort of the DRCMR dataset has a wide range of ages. Due to the atrophy, the brain structures change through the time. In addition, the **brain variability**, specially of the cerebral cortex, is very high among individuals. Therefore, the idea of using just one averaged template like in SPM [28] seems not enough. For example, Free Surfer includes 10 different templates sets for kid, young and old brains [51]. Besides several authors have proposed different methods to account for this brain variability, e.g. dictionaries, hierarquical modeling, label fusion [67] or Simultaneous Truth and Performance Level Estimation (STAPLE) [87].

An important source of errors is the **alignment between $\mathbf{T_1}$ and $\mathbf{T_2}$**, which in the current pipeline is done as a pre-processing step. Therefore, it is proposed to include the co-registration of both modalities in the cost function. Likewise, each iteration would update also the parameters that model the registration between both modalities.

The SPM method assumes **independent voxels**; however, neigbouring voxels are expected to belong to the same tissue class. Likewise, VBM applies a HMRF to include spatial information in the model [91]. Besides, it is also possible to apply smoothing techniques like a de-noising filter. For example, the Optimized Rician Non-Local Means (ORNLM) filter is used in VBM5 and Spatial Adaptive Non-Local Means (SANLM) filter is used in VBM8.

The 'New Segmentation' method is very sensitive to the **value parameters**. Therefore, it is needed to estimate which values are better for the toolbox parameters in order that the segmentation of the DRCMR dataset is optimal. However, the parameters that are optimal for one dataset could not be optimal for another dataset. Therefore, the only way to set up the parameters without a reliable ground truth is through empirical exploration.

6.3 Conclusions

Graphically, it was seen that the multispectral approach with T_1 and T_2 MRI modalities could improve the brain tissue segmentation. The results of several multispectral methods on two different datasets showed that this hypothesis came to be true. In fact, the fully multispectral implementation proposed in this thesis, where the modalities are not supposed to be non-correlated, is slightly more accurate and robust than multispectral methods were this assumption is kept. Therefore, future improvements in this study line seem to be promising.

Bibliography

[1] M.J. Ackerman. The visible human project. *Proceedings of the IEEE*, 86(3):504–511, 2002.

[2] J. Ashburner and K.J. Friston. Voxel-based morphometry–the methods. *Neuroimage*, 11(6):805–821, 2000.

[3] J. Ashburner and K.J. Friston. Unified segmentation. *Neuroimage*, 26(3):839–851, 2005.

[4] M.S. Atkins and B.T. Mackiewich. Fully automatic segmentation of the brain in MRI. *Medical Imaging, IEEE Transactions on*, 17(1):98–107, 1998.

[5] J.F. Barrett and N. Keat. Artifacts in CT: Recognition and Avoidance1. *Radiographics*, 24(6):1679, 2004.

[6] C.M. Bishop and SpringerLink (Online service). *Pattern recognition and machine learning*, volume 4. Springer New York:, 2006.

[7] G. Bouchard and B. Triggs. The tradeoff between generative and discriminative classifiers. In *IASC International Symposium on Computational Statistics (COMPSTAT)*, pages 721–728. Citeseer, 2004.

[8] E. Busa. *FreeSurfer Manual.* Massachusetts General Hospital, Boston, Massachusetts., 2002.

[9] P.T. Callaghan. *Principles of nuclear magnetic resonance microscopy.* Oxford University Press, USA, 1993.

[10] M.C. Clark, L.O. Hall, D.B. Goldgof, L.P. Clarke, R.P. Velthuizen, and M.S. Silbiger. MRI segmentation using fuzzy clustering techniques. *Engineering in Medicine and Biology Magazine, IEEE*, 13(5):730–742, 2002.

[11] L.P. Clarke, R.P. Velthuizen, M.A. Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, and M.L. Silbiger. MRI segmentation: methods and applications. *Magnetic resonance imaging*, 13(3):343–368, 1995.

[12] C.A. Cocosco, V. Kollokian, K.S.K. Remi, G.B. Pike, and A.C. Evans. Brainweb: Online interface to a 3d mri simulated brain database. In *NeuroImage*. Citeseer, 1997.

[13] D.L. Collins, A.P. Zijdenbos, V. Kollokian, J.G. Sled, NJ Kabani, C.J. Holmes, and A.C. Evans. Design and construction of a realistic digital brain phantom. *Medical Imaging, IEEE Transactions on*, 17(3):463–468, 1998.

[14] D.L. Collins, A.P. Zijdenbos, V. Kollokian, J.G. Sled, N.J. Kabani, C.J. Holmes, and A.C. Evans. Design and construction of a realistic digital brain phantom. *Medical Imaging, IEEE Transactions on*, 17(3):463–468, 2002.

[15] K. Conradsen. An Introduction to Statistics. *IMSOR, DTU*, 1976.

[16] A.M. Dale, B. Fischl, and M.I. Sereno. Cortical Surface-Based Analysis: Segmentation and Surface Reconstruction. *Neuroimage*, 9(2):179–194, 1999.

[17] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[18] L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[19] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern classification john wiley & sons. *Inc., New York*, 2000.

[20] R.R. Ernst, G. Bodenhausen, and A. Wokaun. *Principles of nuclear magnetic resonance in one and two dimensions*. Oxford University Press, USA, 1990.

[21] A.G. Filler. The history, development and impact of computed imaging in neurological diagnosis and neurosurgery: CT, MRI, and DTI. 2009.

[22] B. Fischl, D.H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, et al. Whole Brain Segmentation:: Automated Labeling of Neuroanatomical Structures in the Human Brain. *Neuron*, 33(3):341–355, 2002.

[23] J.M. Fitzpatrick, D.L.G. Hill, and C.R. Maurer Jr. Image registration. *Handbook of Medical Imaging: Medical Image Processing and Analysis*, 2:447–514.

[24] J. Kiebel J. Nichols T. Friston, K. Ashburner and W. Penny. *Statistical parametric mapping : the analysis of funtional brain images.* Elsevier Ltd, 2007.

[25] K.J. Friston, J. Ashburner, C.D. Frith, J.B. Poline, J.D. Heather, and R.S.J. Frackowiak. Spatial registration and normalization of images. *Human brain mapping*, 3:165–189, 1995.

[26] K.J. Friston, A.P. Holmes, K.J. Worsley, J.P. Poline, C.D. Frith, and R.S.J. Frackowiak. Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4):189–210, 1994.

[27] T.V. Gestel, JAK Suykens, G. Lanckriet, A. Lambrechts, B.D. Moor, and J. Vandewalle. Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel fisher discriminant analysis. *Neural Computation*, 14(5):1115–1147, 2002.

[28] C.D. Good, I.S. Johnsrude, J. Ashburner, R.N.A. Henson, K.J. Friston, and R.S.J. Frackowiak. A voxel-based morphometric study of ageing in 465 normal adult human brains. *Neuroimage*, 14(1):21–36, 2001.

[29] C.R.G. Guttmann, F.A. Jolesz, R. Kikinis, R.J. Killiany, M.B. Moss, T. Sandor, and M.S. Albert. White matter changes with normal aging. *Neurology*, 50(4):972, 1998.

[30] G. Hamarneh and X. Li. Watershed segmentation using prior shape and appearance knowledge. *Image and Vision Computing*, 27(1-2):59–68, 2009.

[31] A.K. Jain. *Fundamentals of digital image processing.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1989.

[32] J. Jan. *Medical Image Processing, Reconstruction and Restoration: Concepts and Methods.* Taylor and Francis, 2005.

[33] T.L. Jernigan, S.L. Archibald, C. Fennema-Notestine, A.C. Gamst, J.C. Stout, J. Bonner, and J.R. Hesselink. Effects of age on tissues and regions of the cerebrum and cerebellum. *Neurobiology of Aging*, 22(4):581–594, 2001.

[34] J.R. Jimenez-Alaniz, V. Medina-Banuelos, and O. Yáñez-Suárez. Data-driven brain MRI segmentation supported on edge confidence and a priori tissue information. *Medical Imaging, IEEE Transactions on*, 25(1):74–83, 2005.

[35] B. Kevles. *Naked to the bone: Medical imaging in the twentieth century.* Rutgers Univ Pr, 1997.

[36] V.S. Khoo, D.P. Dearnaley, D.J. Finnigan, A. Padhani, S.F. Tanner, and M.O. Leach. Magnetic resonance imaging (MRI): considerations and applications in radiotherapy treatment planning. *Radiotherapy and Oncology*, 42(1):1–15, 1997.

[37] F. Klauschen, A. Goldman, V. Barra, A. Meyer-Lindenberg, and A. Lundervold. Evaluation of automated brain MR image segmentation and volumetry methods. *Human brain mapping*, 30(4):1310–1327, 2009.

[38] S. Kullback. The kullback-leibler distance. *The American Statistician*, 41(4):340–341, 1987.

[39] R.K.S. Kwan, A. Evans, and G. Pike. An extensible mri simulator for post-processing evaluation. In *Visualization in Biomedical Computing*, pages 135–140. Springer, 1996.

[40] R.K.S. Kwan, A.C. Evans, and G.B. Pike. Mri simulation-based evaluation of image-processing and classification methods. *Medical Imaging, IEEE Transactions on*, 18(11):1085–1097, 1999.

[41] K.W. Laidlaw, D.H. Fleischer and A.H. Barr. Partial-volume bayesian classification of material mixtures in mr volume data using voxel histograms. *Medical Imaging, IEEE Transactions on*, 17(1):74–86, 1998.

[42] K.K. Leung, J. Barnes, M. Modat, G.R. Ridgway, J.W. Bartlett, N.C. Fox, and S. Ourselin. Brain MAPS: An automated, accurate and robust brain extraction technique using a template library. *NeuroImage*, 2010.

[43] S.Z. Li. *Markov random field modeling in image analysis.* Springer-Verlag New York Inc, 2009.

[44] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *Medical Imaging, IEEE Transactions on*, 16(2):187–198, 1997.

[45] G.J. McLachlan and D. Peel. *Finite mixture models*, volume 299. Wiley-Interscience, 2000.

[46] A.K.H. Miller, R.L. Alston, and J. Corsellis. Variation with age in the volumes of grey and white matter in the cerebral hemispheres of man: measurements with an image analyser. *Neuropathology and applied neurobiology*, 6(2):119–132, 1980.

[47] N. Moon, E. Bullitt, K. Van Leemput, and G. Gerig. Automatic brain and tumor segmentation. *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2002*, pages 372–379, 2002.

[48] B. Moretti, J.M. Fadili, S. Ruan, D. Bloyet, and B. Mazoyer. Phantom-based performance evaluation: Application to brain segmentation from magnetic resonance images. *Medical image analysis*, 4(4):303–316, 2000.

[49] R.M. Neal and G.E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 89:355–368, 1998.

[50] R.W.D. Nickalls. A new approach to solving the cubic: Cardan's solution revealed. *The Mathematical Gazette*, 77(480):354–359, 1993.

[51] Online: Athinoula A. Martinos Center for Biomedical Imaging. Freesurfer. http://surfer.nmr.mgh.harvard.edu/, February 2011.

[52] Online: Department of Psychiatry, University of Jena. Structural brain mapping group. http://dbm.neuro.uni-jena.de/, February 2011.

[53] Online: Hvidovre Hospital. Danish research centre for magnetic resonance. http://www.drcmr.dk/, February 2011.

[54] Online: Massachusetts General Hospital, Center for Morphometric Analysis. Internet brain segmentation repository. http://www.cma.mgh.harvard.edu/ibsr/data.html, February 2011.

[55] Online: McConnell Brain Imaging Centre (BIC), The Lundbeck Foundation. Brainweb: Simulated brain database. http://www.bic.mni.mcgill.ca/brainweb/, May 2011.

[56] Online: NIfTI. Neuroimaging informatics technology initiative. http://nifti.nimh.nih.gov/, February 2011.

[57] Online: Siemens AG. Magnetom trio. http://www.medical.siemens.com/siemens/en_US/gg_mr_FBAs/files/brochures/Trio_Brochure.pdf, February 2011.

[58] Online: The Lundbeck Foundation. Center for integrated molecular brain imaging (cimbi). http://http://www.cimbi.dk/, February 2011.

[59] Online: Wellcome Images. Terms of use. http://images.wellcome.ac.uk/indexplus/page/Terms+of+Use.html?, February 2011.

[60] Online: Wellcome Trust Centre for Neuroimaging. Statistical parametric mapping. http://www.fil.ion.ucl.ac.uk/spm/, February 2011.

[61] Online: Zygote Media Group Inc. & Google. Google body. http://bodybrowser.googlelabs.com, 2011.

[62] B. Pakkenberg and H.J.G. Gundersen. Neocortical neuron number in humans: effect of sex and age. *The Journal of Comparative Neurology*, 384(2):312–320, 1997.

[63] D. Purves, G.J. Augustine, D. Fitzpatrick, L.C. Katz, A.S. LaMantia, J.O. McNamara, and S.M. Williams. Neuroscience. Sunderland, MA. *Sinauer Associates. Retrieved June*, 7:2008, 2001.

[64] M. Quarantelli, K. Berkouk, A. Prinster, B. Landeau, C. Svarer, L. Balkay, B. Alfano, A. Brunetti, J.C. Baron, and M. Salvatore. Integrated software for the analysis of brain PET/SPECT studies with partial-volume-effect correction. *Journal of Nuclear Medicine*, 45(2):192, 2004.

[65] L. Regeur, G. Badsberg Jensen, H. Pakkenberg, SM Evans, and B. Pakkenberg. No global neocortical nerve cell loss in brains from patients with senile dementia of Alzheimer's type. *Neurobiology of aging*, 15(3):347–352, 1994.

[66] K. Rehm, K. Schaper, J. Anderson, R. Woods, S. Stoltzner, and D. Rottenberg. Putting our heads together: a consensus approach to brain/non-brain segmentation in T1-weighted MR volumes. *Neuroimage*, 22:1262–1270, Jul 2004.

[67] M. Sabuncu, B. Yeo, K. Van Leemput, B. Fischl, and P. Golland. A Generative Model for Image Segmentation Based on Label Fusion. *Medical Imaging, IEEE Transactions on*, (99):1.

[68] M. Sabuncu, B. Yeo, K. Van Leemput, T. Vercauteren, and P. Golland. Asymmetric image-template registration. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 565–573, 2009.

[69] Z.Y. Shan, G.H. Yue, and J.Z. Liu. Automated histogram-based brain segmentation in T1-weighted three-dimensional magnetic resonance head images. *NeuroImage*, 17(3):1587–1598, 2002.

[70] D.W. Shattuck, G. Prasad, M. Mirza, K.L. Narr, and A.W. Toga. Online resource for validation of brain segmentation methods. *NeuroImage*, 45(2):431–439, 2009.

[71] J. Sijbers, P. Scheunders, M. Verhoye, A. Van der Linden, D. Van Dyck, and E. Raman. Watershed-based segmentation of 3D MR data for volume quantization. *Magnetic Resonance Imaging*, 15(6):679–688, 1997.

[72] A. Simmons, S.R. Arridge, GJ Barker, and P.S. Tofts. Segmentation of neuroanatomy in magnetic resonance images. In *Proceedings of SPIE*, volume 1652, page 2, 1992.

[73] A. Simmons, P.S. Tofts, G.J. Barker, and S.R. Arridge. Sources of intensity nonuniformity in spin echo images at 1.5 t. *Magnetic Resonance in Medicine*, 32(1):121–128, 1994.

[74] S. Simon. *The brain: our nervous system.* William Morrow & Co, 1997.

[75] J.G. Sled, A.P. Zijdenbos, and A.C. Evans. A nonparametric method for automatic correction of intensity nonuniformity in mri data. *Medical Imaging, IEEE Transactions on*, 17(1):87–97, 1998.

[76] S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, M. De Luca, I. Drobnjak, D.E. Flitney, et al. Advances in functional and structural MR image analysis and implementation as FSL. *Neuroimage*, 23:S208–S219, 2004.

[77] P. Subbiah, P. Mouton, H. Fedor, J.C. McArthur, and J.D. Glass. Stereological analysis of cerebral atrophy in human immunodeficiency virus-associated dementia. *Journal of Neuropathology & Experimental Neurology*, 55(10):1032, 1996.

[78] H. Suzuki and J. Toriwaki. Automatic segmentation of head MRI images by knowledge guided thresholding. *Computerized medical imaging and graphics*, 15(4):233–240, 1991.

[79] C. Svarer, K. Madsen, S.G. Hasselbalch, L.H. Pinborg, S. Haugbøl, V.G. Frøkjær, S. Holm, O.B. Paulson, and G.M. Knudsen. MR-based automatic delineation of volumes of interest in human brain PET images using probability maps. *Neuroimage*, 24(4):969–979, 2005.

[80] T. Tasdizen, D. Weinstein, and J.N. Lee. Automatic tissue classification for the human head from multispectral mri. 2004.

[81] P. Thévenaz and M. Unser. Optimization of mutual information for multiresolution image registration. *Image Processing, IEEE Transactions on*, 9(12):2083–2099, 2000.

[82] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens. Automated model-based tissue classification of MR images of the brain. *Medical Imaging, IEEE Transactions on*, 18(10):897–908, 2002.

[83] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens. A unifying framework for partial volume segmentation of brain MR images. *Medical Imaging, IEEE Transactions on*, 22(1):105–119, 2003.

[84] Y. Wang, T. Adali, S.Y. Kung, and Z. Szabo. Quantification and segmentation of brain tissues from MR images: A probabilistic neural network approach. *Image Processing, IEEE Transactions on*, 7(8):1165–1181, 2002.

[85] S. Warfield, J. Dengler, J. Zaers, C.R.G. Guttmann, W.M. Wells, G.J. Ettinger, J. Hiller, and R. Kikinis. Automatic identification of gray matter structures from MRI to improve the segmentation of white matter lesions. *Journal of Image Guided Surgery*, 1(6):326–338, 1995.

[86] S. Warfield, K. Zou, and W. Wells. Validation of image segmentation and expert quality with an expectation-maximization algorithm. *Medical Image Computing and Computer-Assisted Interventionï¿½MICCAI 2002*, pages 298–306, 2002.

[87] S.K. Warfield, K.H. Zou, and W.M. Wells. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *Medical Imaging, IEEE Transactions on*, 23(7):903–921, 2004.

[88] N. Weiskopf, A. Lutti, G. Helms, M. Novak, J. Ashburner, and C. Hutton. Unified segmentation based correction of r1 brain maps for rf transmit field inhomogeneities (unicort). *NeuroImage*, 54(3):2116 – 2124, 2011.

[89] R.P. Woods, J.C. Mazziotta, et al. MRI-PET registration with automated algorithm. *Journal of computer assisted tomography*, 17(4):536, 1993.

[90] M.W. Woolrich, S. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, and S.M. Smith. Bayesian analysis of neuroimaging data in FSL. *Neuroimage*, 45(1):S173–S186, 2009.

[91] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *Medical Imaging, IEEE Transactions on*, 20(1):45–57, 2001.

[92] A.P. Zijdenbos, B.M. Dawant, R.A. Margolin, and A.C. Palmer. Morphometric analysis of white matter lesions in mr images: method and validation. *Medical Imaging, IEEE Transactions on*, 13(4):716–724, 1994.

[93] K.H. Zou, S.K. Warfield, A. Bharatha, C.M.C. Tempany, M.R. Kaus, S.J. Haker, W.M. Wells III, F.A. Jolesz, and R. Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index: scientific reports. *Academic radiology*, 11(2):178, 2004.

Appendices

Appendix A

Magnetic Resonance

This section describes the MRI physics, specially the NMR effect. The information is based on [9] [20] [23] [32].

A.1 Nuclear Magnetic Resonance

Nucleus with one single proton, like ^1H, have positive charge. Under an external magnetic field B_0, the proton is polarized and they spin at a fixed frequency w_L, called *Larmor frequency*. They have a rotation movement around its own axis called *spin*. The spin has two quantum states $\pm 1/2$ depending on the direction. They are namely parallel and anti-parallel, and state for a low and high energy state, respectively. The energy level difference between both states is proportional to the applied field. This rotating magneto induces a magnetic moment m pointing in the direction of the rotation axis. The proton has no mass, therefore instead of spinning, it suffers a precession movement around the axis. This movement is characterized by the magnetic moment m describing a circumference, which a certain tilt angle α, as presented in the Figure A.1.

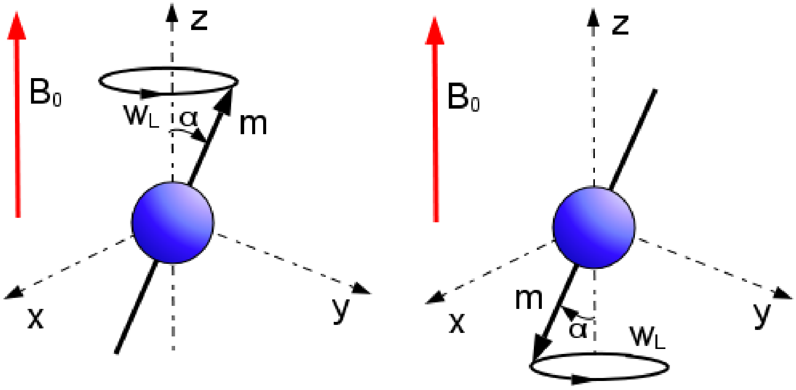


Figure A.1: Representation of the precession movement of one proton in the coordinate axes. The vector m corresponds to the magnetic moment spinning at a frequency w_L under the external magnetic field B_0. Left: parallel direction corresponding to a low energy state. Right: anti-parallel direction corresponding to a high energy state.

The Equation A.1 shows the dependency of the spin frequency w_L respect to the *gyromagnetic radio* γ and the intensity of the applied magnetic field B_0. The magnetic field unit is Tesla T, which is equivalent to $1e^4$ gauss.

$$w_L = -\gamma \cdot \mathbf{B}_0 \tag{A.1}$$

The gyromagnetic radio for the Hydrogen ^1H is approximately 42.57 MHz/T, thus in a 3T scanner the associated spin frequency is around 127Hz. The higher is the applied static field, the better the image quality. The data for this project is acquired with a 3 Tesla scanner. The static field of MRI can be compared with other common magnetic fields, like the human brain with less than $1e^{-8}$ gauss, the Earth with around 0.5 gauss in average or a typical refrigerator magneto of 50 gauss. The choice of the Hydrogen is due to its high magnetic response, and high abundance in biological tissues, specially as water. In this point, it appears one of the advantages of MRI, its working frequency is much lower than the one used for CT and X-rays, which can be higher than 30PHz. Radiation with frequencies higher than 1PHz are considered as ionizing and they break the molecular bonds; in opposite to the non-ionizing ones that just heat. In human tissues, the nuclei ^1H appears combined in water H_2O and fat CH_2 with slightly different resonant frequencies which creates the contrast in the MR images.

A.1.1 Perturbation

In the nature, the magnetic moments m of each nuclei are usually pointing in random directions and with different phases. As explained previously, if an external magnetic field B_0 is applied, the nucleus will torque its precession axis following the magnetic field. It is considered that this applied field is static and space homogeneous. However, the lack of phase coherence makes that the overall macroscopic magnetic field net M can be almost neglected. For example, the Equation A.2 represents the ratio of nucleus with parallel and anti-parallel spin according from the Boltzmann's equation, where h stands for the Planck's constant and k for the Boltzmann's constant. At an ambient temperature of $T = 27° + 273.15°$, the excess of spin-up particles accounts for just 10 parts per million (ppm).

$$\frac{spin\_up}{spin\_down} = exp\left(\frac{\gamma \cdot h \cdot B_0}{k \cdot T}\right) \tag{A.2}$$

In order to increase the coherence among particles, a non-static magnetic field perturbation B_{RF} is applied in the form of RF pulse at the corresponding Larmour frequency. In order to increase the effect, the applied pulse is usually normal to the original static field B_0, and in a narrow frequency band around w_L. This perturbation induces another precession movement with spinning frequency w_1 in the B_{RF} direction meanwhile the field is applied. The combination of both precession movements created by B_0 and B_{RF} determines the movement of the magnetic moment of the particle. As the static field is stronger than the non-static, the precession frequency w_L will be bigger than w_1, which generates a spiral trajectory of the macroscopic field M, and thus an increase in the transverse magnetization.

From a quantum point of view, this perturbation will excite the nucleus and flip the spin of some particles. If they were in a lower energy state (spin-up or parallel direction), they will shift to a higher energy one (spin-down or anti-parallel direction). On the other hand, if they were already in a high energy state, they will release energy as a photon at the resonance frequency and they will shift to a low energy state. This growing number of anti-parallel particles increases the *flip angle*, which could reaches values close to 180°. Although, a single particle can only have two precession angles, i.e. parallel 0° and anti-parallel 180°, a volume of several particles can have values between 0° and 180° as an average over all the particles included in the volume.

During this phase, the number of particles of both spins are equal. The macroscopic magnetization M precesses at the Larmour frequency in the transverse plane, i.e. xy-plane. After this 90° *flip*, the longitudinal component in the z-direction of the magnetic field vanishes.

The stimulated NMR response on the ^{1}H nucleus is presented in Figure A.2. On the left, it is presented the random spin moment alignment of the nucleus when no external field is applied. On the middle, it is depicted the nuclei polarization trying to follow the applied magnetic field B_0 in the parallel direction. On the right, it is presented the precession movement change under the perturbation. Usually, the angle of attack is close 90° to increase the observed energy. The final flip angle is an average over all the particles in the volume.

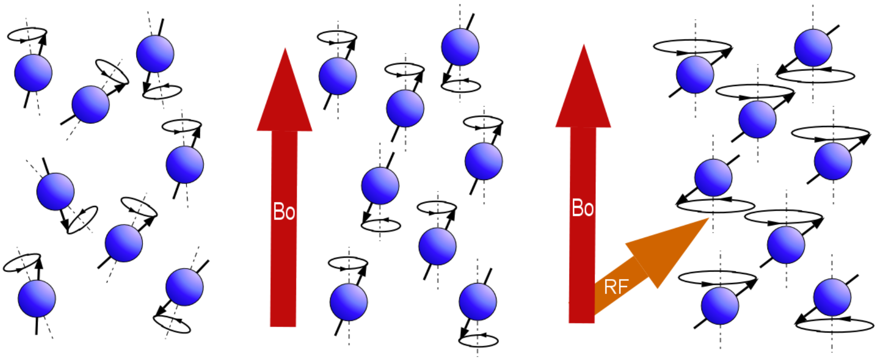


Figure A.2: Nuclear Magnetic Resonance on the ^{1}H nucleus, composed by one proton. On the left, it is presented the random spin moment alignment of the nucleus when no external field is applied. On the middle, it is depicted the nuclei polarization trying to follow the applied magnetic field B_0 in the parallel direction. The alignment is not perfect due to the plasticity of the materials; however, for simplicity, the ideal case is presented. On the right, it is presented the precession movement change under the the RF perturbation of the B_{RF} field. The combination of both magnetic fields creates a spiral trajectory of the magnetic moment that is not presented, again for simplicity. Usually, the angle of attack is close 90° to increase the observed energy. The final flip angle is an average over all the particles in the volume.

A.1.2 Relaxation

Once the perturbation is over, the *relaxation* period starts and the thermal equilibrium is recovered. The signal received in the coils during this phase is called free-induction decay (FID) signal. During this process, the total magnetic moment M gradually goes back to the original z-direction and the phase coherence is lost. This decay can be characterized by two relaxation times, T_1 and T_2, related to the exponential nature of this process that follows the Bloch Equations. T_1 lasts longer than T_2, $T_1 > T_2$. As presented in the Figure A.3, after the 90° flip, the transverse field M_{xy} is maximum and the longitudinal M_z is minimum. The decay of both fields to the the 63% of its respective maximum value determines the corresponding values. In addition, other parameters can also been acquired from the FID with NMR imaging, like PD and T_2^*, which stand for the proton density and envelope of the T_2 decay, respectively. However they are not included in the dataset of this project, so no further comment will be done.

The two relaxation times can be characterized as follows:

- **$\mathbf{T_1}$ (spin-lattice relaxation)**. It measures the recovery of magnetic field in the direction of the static magnetic field B_0 (longitudinal). It is due to the particle moving back to the low energy state after emitting a photon at the Lambour frequency. The usual values are about 240 to 810 msec, and they comprise the time to recover the 63% of the original magnetization in the longitudinal direction.
- **$\mathbf{T_2}$ (spin-spin relaxation)**. It represents the decay of the magnetic field orthogonal to B_0 (transverse) due to the dephasing of spins caused by proton interaction with its environment. They are usually in the range of 40-100 msec, which is the time of decay for the 63% of the transverse magnetization.

The solids, like the scalp, have no signal in MRI due to the short relaxation time. The gases and the free pure water have both equal T_1 and T_2, which for the water can last even for some seconds because of the great absorption of the RF signal that keeps them in phase. In liquids, the T_1 is bigger than T_2. Therefore, the GM has bigger value for T_1 than for T_2, while the opposite occurs with the WM.

The shorter is the relaxation time, the brighter is the acquired MR image. Thus, a T_1 **image will have brighter voxels for WM, darker for GM, and almost black for CSF**. In addition, for T_1, a tumour has bigger acquired intensity value than a normal tissue, and muscle tissue bigger than fat. Therefore, some lesions can resemble GM in T_1 images. Almost the opposite contrast will be expected in T_2 images.

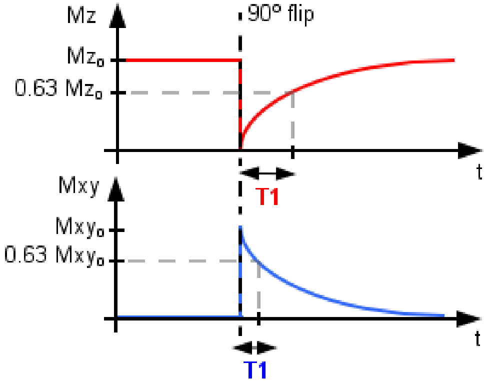


Figure A.3: Relaxation time T_1 (red) and T_2 (ble) during the perturbance and relaxation processes. The epoch of the 90° flip is marked with a vertical dashed line. The top plot presents the time-evolution of the longitudinal component of the magnetic field M_z (in red), from where it is extracted the T_1 value. The bottom plot includes the transverse field M_{xy} and the T_2 time.

A.2 Image Generation

The images that are arranged to prevail either T_1 or T_2 are called T_1-weighted and T_2-weighted, respectively. Sophisticated RF pulse sequences are carried out in order to determine correctly both relaxation time values from the FID. Scanning for a long time creates better quality images, but introduces other artifacts due to the patient motion, and also it is a more expensive procedure. Two important parameters are used to increase the contrast of the images: echo time (TE) and time repetition (TR). TE stands for the time at which the signal is measured after the RF emission; and TR is the time between two consecutive RF emissions. If both are small compared to the relaxation time of T_1, the T_1 contrast is enhanced and generates the T_1-weighted images. The gradient echo (GRE) and saturation recovery (SR) techniques are used for this purpose. On the other hand, a T_2-weighted image is acquired with long time constants, like spin echo (SE) technique does. In order to spatially associate the intensity value of each voxel with its 3D position, a gradient of the magnetic field B_0 is applied to generate planes with different Larmour frequency. Therefore, at each plane the spins will have different frequency, i.e. gradients transform spatial dimensions into frequency dimensions. Afterwards, with complex signal processing based on DFT the intensity images are reconstructed. The direction of the field can be changed as a roll to create a k-space that can be decoded in the Fourier domain. Sometimes, the periodicity of the transformation creates wraparound effects in the reconstructed images. Besides, the acquisition times of this procedure can be shorten by the Echo-Planar Imaging (EPI) technique.

Appendix B

Mathematics

This chapter includes mathematic equations of some parts of the implementation that are included here with more details. In order to ensure a clear understanding, some pages are horizontally formatted, i.e. landscape. The included concepts are:

- Theory of the Gaussian distribution in one and two dimensions.
- Mathematical process to obtain the 2D Gaussian distribution expression from the multidimensional normal equation.
- Full expression of the upper bound of the cost function in the M-step.
- Theory and Matlab implementation of the central and non-central moments.
- Solution of a third degree equation in Matlab.
- 2D example of registration in Matlab.

B.1 Gaussian distribution

In the Mixture of Gaussians (MoG) model used in SPM, each cluster is modeled with a Normal (Gaussian) distribution, which can be parameterized by the intensity mean vector μ, and intensity covariance matrix $\mathbf{\Sigma}$.

The Figure B.1 depicts the uni-dimensional Gaussian distribution bell, which is characterized by the parameters $\mu = 2$, and $\sigma = 0.5$ ($\sigma^2 = 0.25$). Most of the probability is around the mean value and it decreases fast when the variance is small, i.e. the bell is more narrow. The integral of the probability density returns probability. Therefore, the area under the curve for $x = \pm\sigma$ accounts for the 68% of the total probability, for $x = \pm 2\sigma$ accounts for the 95%, and for $x = \pm 3\sigma$ accounts for the 99.7%.

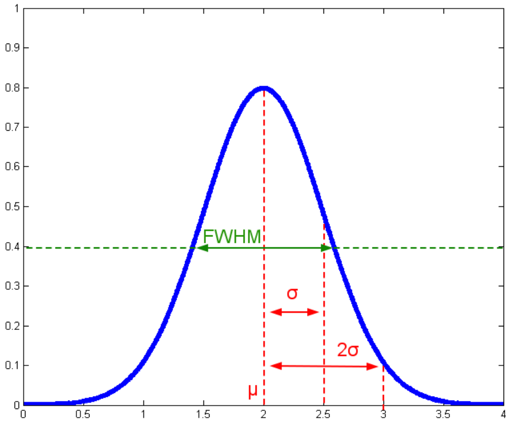


Figure B.1: Normal (Gaussian) Distribution with parameters $\mu = 2$, and $\sigma = 0.5$.

The general expression of uni-dimensional Normal distribution is presented in the Equation B.1. The maximum amplitude is at the center of the bell, $x = \mu$, which corresponds to a value of $f(x)_{max} = 1/\sqrt{2\pi}\sigma$, while the FWHM is $FWHM = 2\sqrt{2Ln(2)}\sigma$.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x-\mu}{2\sigma^2}} \tag{B.1}$$

The Equation B.2 presents the bi-dimensional Normal distribution for the variables x_1 and x_2. It is characterized by the mean (μ_1,μ_2) and variance (σ_1,σ_2) for each variable. In addition, it counts with an additional parameter that measures the correlation between them, the cross covariance σ_{12}.

$$f(x_1, x_2) = \frac{1}{2\pi\,\sqrt{\sigma_1^2 \cdot \sigma_2^2 - \sigma_{12}^2}}\, e^{\left\{\frac{-\sigma_1^2\cdot\sigma_2^2}{2(\sigma_1^2\cdot\sigma_2^2-\sigma_{12}^2)}\cdot\left[\frac{(x_1-\mu_1)^2}{\sigma_1^2}+\frac{(x_2-\mu_2)^2}{\sigma_2^2}-2\sigma_{12}\cdot\frac{(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1^2\cdot\sigma_2^2}\right]\right\}} \tag{B.2}$$

In case both variables are non-correlated, $\sigma_{12} = 0$, the joint probability is just the product of both distributions, and the distribution variances lies on the reference axes. However, if the variables are dependent, the 2D Gaussian bell appears rotated. The Equation B.3 presents the expression to calculate the rotation angle of the Gaussian respect to the reference system. This angle corresponds to the rotation of the eigenvector axes from the original distribution to one with $\sigma 12 = 0$. In addition, this characteristic allows to express a distribution with no diagonal covariance matrix into one distribution with diagonal covariance, just by rotating the reference axes.

$$\varphi = \frac{1}{2}\, atan \left(\frac{2\ \sigma_{12}}{\sigma_2 - \sigma_2} \right) \tag{B.3}$$

The *tan()* function is π-periodic, thus it gives two valid solutions in the interval $[0, 2\pi)$, which correspond to φ and $\varphi + \pi$. In addition, in case the axis of the Gaussian are equal, $\sigma_1 = \sigma_2$, the angle has four solution in $[0, 2\pi)$, namely $\left\{\varphi,\ \varphi + \frac{\pi}{2},\ \varphi + \pi,\ \varphi + 3\frac{\pi}{2}\right\}$. The obtained angle has the same sign than the cross covariance and the correlation. Therefore, for a positive cross covariance higher values of x_1 implies higher values of x_1, and the apposite.

$$\sigma_{12} > 0 \ \Rightarrow \ \begin{cases} x_1 \uparrow \Rightarrow \ x_2 \uparrow \\ x_1 \downarrow \Rightarrow \ x_2 \downarrow \end{cases} \qquad \sigma_{12} < 0 \ \Rightarrow \ \begin{cases} x_1 \uparrow \Rightarrow \ x_2 \downarrow \\ x_1 \downarrow \Rightarrow \ x_2 \uparrow \end{cases}$$

The values of the covariance depends on the scale and units of the dataset. Therefore, in order to compare the dispersion of values between two variables, it is usually applied the *correlation coefficient*, which normalized each centered variable by its covariance.

$$x \ \rightarrow \ \frac{x - \mu}{\sigma}$$

The following code collects these ideas in order to estimate from the 2D covariance matrix the rotation angle.

```
% Extract eigenvectors and eigenvalues from ovariance matrix.
[EigenVectors,EigenValues] = eig(CovarianceMatrix);
% Angle of rotation and angles of the axis
AngleRotation = 0.5*atan(2*CovarianceMatrix(1,2)/...
            (CovarianceMatrix(1,1)-CovarianceMatrix(2,2)))*(180/pi);
AnglesAxis = atan2(EigenVectors(2,:),EigenVectors(1,:))*(180/pi);
```

The Figure B.2 presents a 2G Gaussian distribution with the parameters $\boldsymbol{\mu} = [2, 3]$, $\boldsymbol{\sigma} = [1, 4]$, and $\sigma_{12} = 0$. The bigger is the variance for each variable, the wider is the distribution along its axis, as it can be seen. In this case, both variables are independent, thus the maximum dispersion of values occurs is the reference axes.

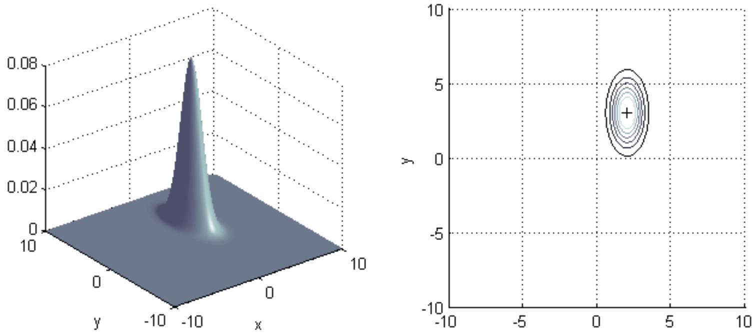


Figure B.2: 2D Gaussian with 0 degrees rotation. The parameters are $\boldsymbol{\mu} = [2, 3]$ and $\boldsymbol{\sigma} = [1, 4]$, and $\sigma_{12} = 0$. The symbol '+' is to the center of the bell.

In the Figure B.3, it is depicted a 2G Gaussian distribution with different parameters. In this case, the variance values are larger, thus the points are more spread and the distribution maximum is smaller. Besides, the negative cross-covariance rotates the maximum dispersion axes with a negative angle.

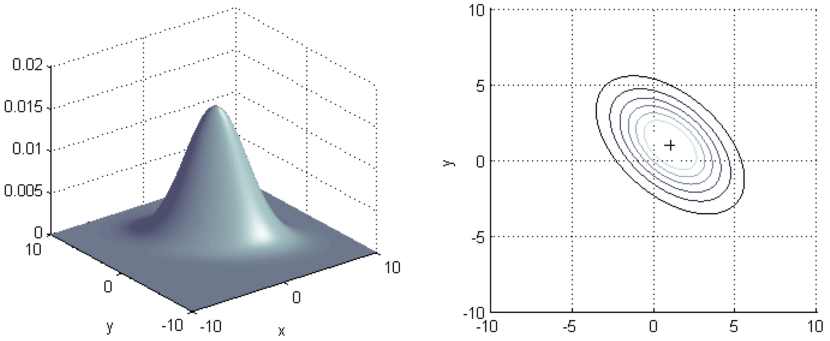
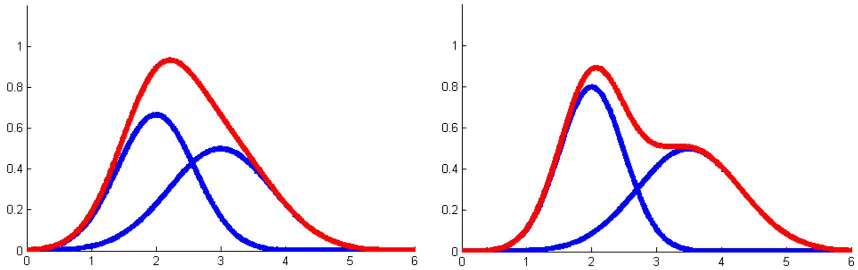


Figure B.3: 2D Gaussian with -45 degrees clockwise rotation. The parameters are $\boldsymbol{\mu} = [1, 1]$ and $\boldsymbol{\sigma} = [10, 10]$, and $\sigma_{12} = -5$. The symbol '+' corresponds to the center of the bell.

The segmentation method of SPM tries to estimate the Gaussians parameters that characterize each cluster, while several clusters are associated to one tissue class. In the MoG, there are several problems in the detection of Gaussians, specially when they too much close among them. When the distance between them is too small and their variances are big, the lobes are overlapped and the mean of the Gaussian is shifter. If the distance is even smaller, it could be impossible to estimate how many Gaussians are, ans maybe is just detected o big distribution that aggregates all of them. In the Figure B.4, it is presented, wlog, the previous problems for the case of two Gaussians.



(a) Only one Gaussian detected.

(b) Gaussian means are shifted.

Figure B.4: Example of two problems in the MOG model. In blue is drawn the original two distributions, and in red is presented the aggregation of them.

B.2 2D Gaussian expression

The k-cluster of the MoG is modeled by a 2-dimensional Gaussian due to the inclusion of two modalities/channels. It is characterized by its mean and variance. In this section, it is presented the steps to generate the expression of a Gaussian distribution in two dimensions from the general multidimensional expression, i.e. $N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mid_{N=2}$.

The Equation B.4 corresponds to the multivariate Gaussian distribution for an N-dimensional variable $\mathbf{Y}$, which is parameterized by the mean vector $\boldsymbol{\mu}_k$, and the covariance matrix $\boldsymbol{\Sigma}_k$, which rank corresponds to the dimensionality N. In the expression, $|\cdot|$ stands for the determinant, and $()^{-1}$ is the inverse matrix.

$$N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{N}{2}} \left|\boldsymbol{\Sigma}_k\right|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}\left(\mathbf{Y}-\boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\mathbf{Y}-\boldsymbol{\mu}_k\right)\right\} \tag{B.4}$$

The dimension N stands for the number of modalities, in this case is fixed to $N = 2$, i.e. T_1 and T_2. Thus, the variable $\mathbf{Y}$ corresponds to $\mathbf{Y} = \begin{bmatrix}\mathbf{Y}_{T1} & \mathbf{Y}_{T2}\end{bmatrix}^T$, $\boldsymbol{\mu}_k$ is the mean vector with dimensions $2x1$, and $\boldsymbol{\Sigma}_k$ is the covariance matrix with dimensions $2x2$. The expressions presented here correspond to the distribution of the kth-cluster, where $C\left(\mathbf{Y}\right)$ stands for the class of the variable $\mathbf{Y}$.

$$\boldsymbol{\mu}_k = E(\mathbf{Y} \mid C\left(\mathbf{Y}\right) = k) = \begin{bmatrix} E\{\mathbf{Y_{T1}} \mid C\left(\mathbf{Y}\right) = k\} \\ E\{\mathbf{Y_{T2}} \mid C\left(\mathbf{Y}\right) = k\} \end{bmatrix}_{2,1} = \begin{bmatrix} \mu_{k,T1} \\ \mu_{k,T2} \end{bmatrix}_{2,1}$$

$$\boldsymbol{\Sigma}_k = Cov(\mathbf{Y} \mid C\left(\mathbf{Y}\right) = k) = \begin{bmatrix} E\{\mathbf{Y_{T1}} - \mu_{T1} \cdot \mathbf{Y_{T1}} - \mu_{T1} \mid C\left(\mathbf{Y}\right) = k\} & E\{\mathbf{Y_{T1}} - \mu_{T1} \cdot \mathbf{Y_{T2}} - \mu_{T2} \mid C\left(\mathbf{Y}\right) = k\} \\ E\{\mathbf{Y_{T2}} - \mu_{T2} \cdot \mathbf{Y_{T1}} - \mu_{T1} \mid C\left(\mathbf{Y}\right) = k\} & E\{\mathbf{Y_{T2}} - \mu_{T2} \cdot \mathbf{Y_{T2}} - \mu_{T2} \mid C\left(\mathbf{Y}\right) = k\} \end{bmatrix}_{2,2}$$

$$= \begin{bmatrix} \sigma_{k,T1T1} & \sigma_{k,T1T2} \\ \sigma_{k,T2T1} & \sigma_{k,T2T2} \end{bmatrix}_{2,2} = \begin{bmatrix} \sigma^2_{k,T1} & \sigma_{k,T1T2} \\ \sigma_{k,T1T2} & \sigma^2_{k,T2} \end{bmatrix}_{2,2} = \begin{bmatrix} \sigma^2_{k,T1} & \rho_k\ \sigma_{k,T1}\sigma_{k,T2} \\ \rho_k\ \sigma_{k,T1}\sigma_{k,T2} & \sigma^2_{k,T2} \end{bmatrix}_{2,2}$$

The distribution is normal, thus the covariance matrix is symmetric, which implies that the cross-correlation terms are equal, i.e. $\sigma_{k,T1T2} = \sigma_{k,T2T1}$. In case that the Gaussians are uncorrelated, i.e. $\rho_k = 0$, the cross-correlation terms are also null and the covariance matrix is diagonal.

The determinant of the covariance matrix is:

$$|\mathbf{\Sigma}_k| = det\left(\mathbf{\Sigma}_k\right) = det\left(\begin{bmatrix} \sigma^2_{k,T1} & \sigma^2_{k,T1T2} \\ \sigma_{k,T1T2} & \sigma^2_{k,T2} \end{bmatrix}\right) = \sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}$$

The inverse of the covariance matrix is:

$$\mathbf{\Sigma}_k^{-1} = inv\left(\mathbf{\Sigma}_k\right) = \frac{1}{det\left(\mathbf{\Sigma}_k\right)} \cdot \begin{bmatrix} (-1)^{1+1}\left|\sigma^2_{k,T2}\right| & (-1)^{1+2}\left|\sigma_{k,T1T2}\right| \\ (-1)^{2+1}\left|\sigma_{k,T1T2}\right| & (-1)^{2+2}\left|\sigma^2_{k,T1}\right| \end{bmatrix} = \frac{1}{\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}} \cdot \begin{bmatrix} \sigma^2_{k,T2} & -\sigma_{k,T1T2} \\ -\sigma_{k,T1T2} & \sigma^2_{k,T1} \end{bmatrix}$$

The square of the Mahanolabis distance corresponds to:

$$\left(\mathbf{Y} - \boldsymbol{\mu}_k\right)^T \mathbf{\Sigma}_k^{-1}\left(\mathbf{Y} - \boldsymbol{\mu}_k\right) = \begin{bmatrix} \mathbf{Y}_{T1} - \mu_{k,T1} & \mathbf{Y}_{T2} - \mu_{k,T2} \end{bmatrix} \cdot \frac{1}{\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}} \cdot \begin{bmatrix} \sigma^2_{k,T2} & -\sigma_{k,T1T2} \\ -\sigma_{k,T1T2} & \sigma^2_{k,T1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Y}_{T1} - \mu_{k,T1} \\ \mathbf{Y}_{T2} - \mu_{k,T2} \end{bmatrix} =$$

$$= \frac{\sigma_{k,T1} \cdot \sigma_{k,T2}}{\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}} \left[\left(\frac{\mathbf{Y}_{T1} - \mu_{k,T1}}{\sigma_{k,T1}}\right)^2 + \left(\frac{\mathbf{Y}_{T2} - \mu_{k,T2}}{\sigma_{k,T2}}\right)^2\right] - \frac{2\ \sigma_{k,T1T2}}{\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}} \left[\left(\mathbf{Y}_{T1} - \mu_{k,T1}\right)\left(\mathbf{Y}_{T2} - \mu_{k,T2}\right)\right]$$

The determinant, inverse and Mahanolabis distance have been previously calculated, and now are included in the general Equation B.4 to obtain the *bivariate normal distribution*, as defined in the Equation B.5.

$$N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{exp\left\{-\frac{\sigma^2_{k,T1}\sigma^2_{k,T2}}{2\left(\sigma^2_{k,T1}\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)}\cdot\left[\left(\frac{\mathbf{Y}_{T1}-\mu_{k,T1}}{\sigma_{k,T1}}\right)^2+\left(\frac{\mathbf{Y}_{T2}-\mu_{k,T2}}{\sigma_{k,T2}}\right)^2\right]+\frac{\sigma_{k,T1T2}}{\left(\sigma^2_{k,T1}\sigma^2_{k,T2}-\sigma^2_{k,T1T2}\right)}\cdot[(\mathbf{Y}_{T1}-\mu_{k,T1})(\mathbf{Y}_{T2}-\mu_{k,T2})]\right\}}{2\pi\ \sqrt{\sigma^2_{k,T1}\cdot\sigma^2_{k,T2}-\sigma^2_{k,T1T2}}} \tag{B.5}$$

If the correlation factor ρ_k is introduced in the previous equation, the Equation B.7 is generated.

$$\rho_k = \frac{\sigma_{k,T1T2}}{\sigma_{k,T1}\cdot\sigma_{k,T2}} = \frac{\sigma_{k,T1T2}}{\sqrt{\sigma_{k,T1T1}\cdot\sigma_{k,T2T2}}} \tag{B.6}$$

The following equation can be compared with the expression in K. Conradsen et al. [Page 78, [15]]. Both equations are equal, thus the 2-dimensional Gaussian expression of Equation B.5 is correct.

$$N(\mathbf{Y} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{exp\left\{-\frac{1}{2\left(1-\rho^2_k\right)}\left[\left(\frac{\mathbf{Y}_{T1}-\mu_{k,T1}}{\sigma_{k,T1}}\right)^2+\left(\frac{\mathbf{Y}_{T2}-\mu_{k,T2}}{\sigma_{k,T2}}\right)^2\right]+\frac{\rho_k}{1-\rho^2_k}\left[\frac{\left(\mathbf{Y}_{T1}-\mu_{k,T1}\right)\left(\mathbf{Y}_{T2}-\mu_{k,T2}\right)}{\sigma_{k,T1}\cdot\sigma_{k,T2}}\right]\right\}}{2\pi\ \sigma_{k,T1}\ \sigma_{k,T2}\ \sqrt{1-\rho^2_k}} \tag{B.7}$$

B.3 Cost Function of M-step

The Equation B.8 presents the complete upper bound of the function cost for the kth-cluster, ε_{EM_k}. It is generated from the Equation 4.30.

$$
\begin{aligned}
\varepsilon_{EM_k} = \quad & - log\left(\gamma_k\right) \cdot \sum_{i=1}^{I} q_{i,k} + \sum_{i=1}^{I} q_{i,k}\, log\left(\sum_{j=1}^{K} \gamma_j \cdot b_{ij}(\boldsymbol{\alpha})\right) \\
& - \sum_{i=1}^{I} q_{i,k}\, log\left(\frac{\rho_{i,T1}(\boldsymbol{\beta}) \cdot \rho_{i,T2}(\boldsymbol{\beta}) \cdot b_{ik}(\boldsymbol{\alpha})}{2\pi}\right) - \frac{1}{2}\, log\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right) \cdot \sum_{i=1}^{I} q_{i,k} \\
& + \frac{\sigma^2_{k,T2}}{2\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \sum_{i=1}^{I} q_{i,k}\, \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)^2 \\
& + \frac{\sigma^2_{k,T1}}{2\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \sum_{i=1}^{I} q_{i,k}\, \left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)^2 \\
& - \frac{\sigma_{k,T1T2}}{\left(\sigma^2_{k,T1} \cdot \sigma^2_{k,T2} - \sigma^2_{k,T1T2}\right)} \sum_{i=1}^{I} q_{i,k}\, \left(\rho_{i,T1}(\boldsymbol{\beta}) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(\rho_{i,T2}(\boldsymbol{\beta}) \cdot y_{i,T2} - \mu_{k,T2}\right)
\end{aligned}
\tag{B.8}
$$

B.4 Central and non-central moments

In the implementation of the segmentation method, the moments of a 2-D discrete variable are used. Namely, the original 'New segmentation' method just uses non-central moments, while the modified method uses the central moments as well. The two dimensions are because the inclusion of the two modalities: T_1 and T_2.

Non-central moments

The *non-central moment* of nth-order for the discrete variable Y belonging to the kth-cluster corresponds to the Equation B.9, where E stands for the expectation operator, q_{ik} is the probability of the intensity value y_i included in the kth cluster, and I is the number of elements (voxels) that are analyzed.

$$E\left\{Y^n\right\}_k = \sum_{i=1}^{I} q_{ik} \cdot y_i^n \tag{B.9}$$

In this case, the random variable is Y is modulated in amplitude by the bias field, thus the discrete variable correspond to $P\left(\beta\right) \cdot Y$, where $P\left(\beta\right)$ stands for the bias field parameterized by β. The zero, first (*mean*), and second moments for the kth-cluster are presented here:

$$mom0(k) = \mathbf{\Sigma_k^0} = \sum_{i=1}^{I} q_{ik} \quad , \quad mom1(:,k) = \mathbf{\Sigma_k^1} = \begin{bmatrix} \sum_{i=1}^{I} q_{ik}\left(p_{i,T1}(\beta) \cdot y_{i,T1}\right) \\ \sum_{i=1}^{I} q_{ik}\left(p_{i,T2}(\beta) \cdot y_{i,T2}\right) \end{bmatrix}_{2x1}$$

$$mom2(:,:,k) = \mathbf{\Sigma_k^2} = \begin{bmatrix} \sum_{i=1}^{I} q_{ik}\left(p_{i,T1}(\beta) \cdot y_{i,T1}\right)^2 & \sum_{i=1}^{I} q_{ik}\left(p_{i,T1}(\beta) \cdot y_{i,T1}\right)\left(p_{i,T2}(\beta) \cdot y_{i,T2}\right) \\ \sum_{i=1}^{I} q_{ik}\left(p_{i,T2}(\beta) \cdot y_{i,T2}\right)\left(p_{i,T1}(\beta) \cdot y_{i,T1}\right) & \sum_{i=1}^{I} q_{ik}\left(p_{i,T2}(\beta) \cdot y_{i,T2}\right)^2 \end{bmatrix}_{2x2}$$

Central moments

If the moments are considered once the mean μ has been subtracted from the variable, i.e. $Y-\mu$, then they correspond to the *central moments*, which are presented in the Equation B.10 for nth-order of the discrete variable Y belonging to the kth-cluster. E corresponds to the expectation operator, q_{ik} is the probability of the intensity value y_i included in the kth cluster, and I is the number of elements (voxels) that are analyzed.

$$E\left\{(Y-\mu)^n\right\}_k = \sum_{i=1}^{I} q_{ik} \cdot (y_i - \mu)^n \tag{B.10}$$

Likewise, the random variable Y is modulated in amplitude by the bias field $P(\beta)$, thus the discrete variable correspond to $P(\beta) \cdot Y$. The zero central moment is the same than the zero non-central moment, thus it is not needed to repeat the expression. The first and second (*variance*) moments for the kth-cluster are presented here:

$$mom1c(:,k) = \hat{\mathbf{\Sigma}}_{\mathbf{k}}^{\mathbf{1}} = \mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{1}} - \mu_{\mathbf{k}} \cdot \mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{0}} = \begin{bmatrix} \sum_{i=1}^{I} q_{ik}\left(p_{i,T1}(\beta) \cdot y_{i,T1} - \mu_{k,T1}\right) \\ \sum_{i=1}^{I} q_{ik}\left(p_{i,T1}(\beta) \cdot y_{i,T2} - \mu_{k,T2}\right) \end{bmatrix}_{2x1}$$

$$mom2c(:,:,k) = \hat{\mathbf{\Sigma}}_{\mathbf{k}}^{\mathbf{2}} = \mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{2}} + \mu_{\mathbf{k}} \cdot \mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{0}} \cdot {\mu_{\mathbf{k}}}^T - \mu_{\mathbf{k}} \cdot {\mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{1}}}^T - \mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{1}} \cdot {\mu_{\mathbf{k}}}^T =$$

$$\begin{bmatrix} \sum_{i=1}^{I} q_{ik}\left(p_{i,T1}(\beta) \cdot y_{i,T1} - \mu_{k,T1}\right)^2 & \sum_{i=1}^{I} q_{ik}\left(p_{i,T1}(\beta) \cdot y_{i,T1} - \mu_{k,T1}\right)\left(p_{i,T2}(\beta) \cdot y_{i,T2} - \mu_{k,T2}\right) \\ \sum_{i=1}^{I} q_{ik}\left(p_{i,T2}(\beta) \cdot y_{i,T2} - \mu_{k,T2}\right)\left(p_{i,T1}(\beta) \cdot y_{i,T1} - \mu_{k,T1}\right) & \sum_{i=1}^{I} q_{ik}\left(p_{i,T2}(\beta) \cdot y_{i,T2} - \mu_{k,T2}\right)^2 \end{bmatrix}_{2x2}$$

The variables $\mu_{\mathbf{k}}$ and $\mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{1}}$ corresponds to the first non-central moment. However, they do not express the same. The former includes all the values of the variable Y, and the latter just $i = 1..I$ values from variable Y. Therefore, when I includes all the values, both are equal and $\hat{\mathbf{\Sigma}}_{\mathbf{k}}^{\mathbf{2}} = \mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{2}} - \mu_{\mathbf{k}} \cdot {\mu_{\mathbf{k}}}^T$. The method analyzes one slice at each time, thus I stands for the number of voxels of each slice, while $\mu_{\mathbf{k}}$ is the mean intensity value of the kth-cluster for the whole brain.

Test of the implementation of the central moments in Matlab

In order to check that the equations and the Matlab code of the central moments are correct, they are estimated in other ways (different that the proposed in the Matlab implementation of Section 4.3) to check their validity.

The first option corresponds to:

```
% Similar expressions for the central moments
mom1c(:,k)   = mom1(:,k)   - mn(:,k)*mom0(k);
mom2c(:,:,k) = mom2(:,:,k) + mn(:,k)*mom0(k)*mn(:,k)' - mn(:,k)*mom1(:,k)' - mom1(:,k)*mn(:,k)';
```

The second option includes the decomposition of each element of the first and second central moments. As the dimensionality is low, the extra effort compensates in order to ensure correct values.

```
% Similar expressions for the central moments
mom1c(:,k)  = mom1(1,k) - mn(1,k)*mom0(k);
mom1c(:,k)  = mom1(1,k) - mn(1,k)*mom0(k);
mom2c(1,1,k) = mom2(1,1,k) + mn(1,k)^2*mom0(k) - 2*mn(1,k)*mom1(1,k);
mom2c(2,2,k) = mom2(2,2,k) + mn(2,k)^2*mom0(k) - 2*mn(2,k)*mom1(2,k);
mom2c(1,2,k) = mom2(1,2,k) + mn(1,k)*mn(2,k)*mom0(k) - mn(1,k)*mom1(2,k) - mn(2,k)*mom1(1,k);
mom2c(2,1,k) = aux_2(1,2,k);
```

Finally, the central moments are calculated following the original expression and also following these two additional forms. The three results give the same values, thus it has been numerically checked that the equations and the Matlab implementation matches.

B.5 Solution to a third degree equation

This point describes two possible approaches to give value for the cross-variance in the M-step of the EM‘optimization, which corresponds to a third degree equation. This kind of functions can be solved with closed-form expressions or looking for the zero-crossings. Here, it is presented a small benchmark to compare both approaches. In both cases, the final solution is chosen according to the criteria discussed in the Section 4.3.

The first method uses the former approach with closed-form expressions.

```
% Closed-form equations
x = solution3th(coef3,coef2,coef1,coef0,1);
if ((vr(1,1,k)*vr(2,2,k)-x^2)<tiny)||(abs(imag(x))>1e-4)
   x = solution3th(coef3,coef2,coef1,coef0,2);
   if (vr(1,1,k)*vr(2,2,k)-x^2)<tiny)||(abs(imag(x))>1e-4)
      x = solution3th(coef3,coef2,coef1,coef0,3);
      if (vr(1,1,k)*vr(2,2,k)-x^2)<tiny||(abs(imag(x))>1e-4)
         x = vrX(1,2,k);
     end
   end
end
solution = real(solution);
```

This function returns the solution of a 3rd degree equation with coefficients *coef3*, *coef2*, *coef1* and *coef0*.

```
function solution = solution3th(coef3,coef2,coef1,coef0,opt)
 % Function that returns the solution of a 3rd degree equation:
 % 'y = coef3*x^3 + coef2*x^2 + coef1*x + coef0'

 switch opt
 case 1
   solution = (((coef0/(2*coef3) + coef2^3/(27*coef3^3) ...
     - (coef1*coef2)/(6*coef3^2))^2 + (coef1/(3*coef3) ...
     - coef2^2/(9*coef3^2))^3)^(1/2) - coef2^3/(27*coef3^3) ...
     - coef0/(2*coef3) + (coef1*coef2)/(6*coef3^2))^(1/3) ...
     - (coef1/(3*coef3) - coef2^2/(9*coef3^2))/(((coef0/(2*coef3) ...
     + coef2^3/(27*coef3^3) - (coef1*coef2)/(6*coef3^2))^2 ...
     + (coef1/(3*coef3) - coef2^2/(9*coef3^2))^3)^(1/2) ...
     - coef2^3/(27*coef3^3) - coef0/(2*coef3) ...
     + (coef1*coef2)/(6*coef3^2))^(1/3) - coef2/(3*coef3);
 case 2
   solution = (coef1/(3*coef3) ...
     - coef2^2/(9*coef3^2))/(2*(((coef0/(2*coef3) ...
     + coef2^3/(27*coef3^3) - (coef1*coef2)/(6*coef3^2))^2 ...
     + (coef1/(3*coef3) - coef2^2/(9*coef3^2))^3)^(1/2) ...
     - coef2^3/(27*coef3^3) - coef0/(2*coef3) ...
     + (coef1*coef2)/(6*coef3^2))^(1/3)) ...
     - (((coef0/(2*coef3) + coef2^3/(27*coef3^3) ...
     - (coef1*coef2)/(6*coef3^2))^2 + (coef1/(3*coef3) ...
     - coef2^2/(9*coef3^2))^3)^(1/2) - coef2^3/(27*coef3^3) ...
     − coef0/(2∗coef3) + (coef1∗coef2)/(6∗coef3^2))^(1/3)/2 ...
     − coef2/(3∗coef3) − (3^(1/2)∗((((coef0/(2∗coef3) ...
     + coef2^3/(27∗coef3^3) − (coef1∗coef2)/(6∗coef3^2))^2 ...
     + (coef1/(3∗coef3) − coef2^2/(9∗coef3^2))^3)^(1/2) ...
     − coef2^3/(27∗coef3^3) − coef0/(2∗coef3) ...
     +(coef1∗coef2)/(6∗coef3^2))^(1/3) + (coef1/(3∗coef3) ...
     − coef2^2/(9∗coef3^2))/(((coef0/(2∗coef3) ...
     + coef2^3/(27∗coef3^3) − (coef1∗coef2)/(6∗coef3^2))^2 ...
     + (coef1/(3∗coef3) − coef2^2/(9∗coef3^2))^3)^(1/2) ...
     − coef2^3/(27∗coef3^3) − coef0/(2∗coef3) ...
     + (coef1∗coef2)/(6∗coef3^2))^(1/3))∗1i)/2;
 case 3
   solution = (coef1/(3∗coef3) ...
     − coef2^2/(9∗coef3^2))/(2∗(((coef0/(2∗coef3) ...
     + coef2^3/(27∗coef3^3) − (coef1∗coef2)/(6∗coef3^2))^2 ...
     + (coef1/(3∗coef3) − coef2^2/(9∗coef3^2))^3)^(1/2) ...
     − coef2^3/(27∗coef3^3) − coef0/(2∗coef3) ...
     + (coef1∗coef2)/(6∗coef3^2))^(1/3)) ...
     − (((coef0/(2∗coef3) + coef2^3/(27∗coef3^3) ...
     − (coef1∗coef2)/(6∗coef3^2))^2 + (coef1/(3∗coef3) ...
     − coef2^2/(9∗coef3^2))^3)^(1/2) − coef2^3/(27∗coef3^3) ...
     − coef0/(2∗coef3) + (coef1∗coef2)/(6∗coef3^2))^(1/3)/2 ...
     − coef2/(3∗coef3) + (3^(1/2)∗((((coef0/(2∗coef3) ...
     + coef2^3/(27∗coef3^3) − (coef1∗coef2)/(6∗coef3^2))^2 ...
     + (coef1/(3∗coef3) − coef2^2/(9∗coef3^2))^3)^(1/2) ...
     − coef2^3/(27∗coef3^3) − coef0/(2∗coef3) ...
     + (coef1∗coef2)/(6∗coef3^2))^(1/3) + (coef1/(3∗coef3) ...
     − coef2^2/(9∗coef3^2))/(((coef0/(2∗coef3) ...
     + coef2^3/(27∗coef3^3) − (coef1∗coef2)/(6∗coef3^2))^2 ...
     + (coef1/(3∗coef3) − coef2^2/(9∗coef3^2))^3)^(1/2) ...
     − coef2^3/(27∗coef3^3) − coef0/(2∗coef3) ...
     + (coef1∗coef2)/(6∗coef3^2))^(1/3))∗1i)/2;
end
```

The second method consist on looking for the roots of the equations, i.e. find the zero-crossing points where the sign of the function changes. In this case, it is used the Matlab function *fzeros()* that applies a bisection method. As a starting point, the chosen values are $\{varOriginal, -varOriginal, +varOriginal\}$.

```
% Bilinear interpolation
[x_value,fval,exitflag] = ...
 fzero(@(x) coef3∗x^3+coef2∗x^2+coef1∗x+coef0,vrX(1,2,k));
if ((vr(1,1,k)∗vr(2,2,k)−x_value^2)<tiny) || ¬exitflag
 [x_value,fval,exitflag] = ...
  fzero(@(x) coef3∗x^3+coef2∗x^2+coef1∗x+coef0,10∗vrX(1,2,k));
  if (vr(1,1,k)∗vr(2,2,k)−x_value^2)<tiny || ¬exitflag
   [x_value,fval,exitflag] = ...
    fzero(@(x) coef3∗x^3+coef2∗x^2+coef1∗x+coef0,−10∗vrX(1,2,k));
   if (vr(1,1,k)∗vr(2,2,k)−x_value^2)<tiny || ¬exitflag
    x_value = vrX(1,2,k);
  end
 end
end
```

The following code presents the test of the two methods:

```
% ----------------  Test of both methods  ------------------

% Parameters
N = 1000;
time1 = 0;  time2 = 0;

% N iterations
for i=1:N

    % Coefficints
    coef3 = 100*randn(1);
    coef2 = 100*randn(1);
    coef1 = 100*randn(1);
    coef0 = 100*randn(1);

    % Exact solution
    tic
    solution1 = solution3th(coef3,coef2,coef1,coef0,1);
    solution2 = solution3th(coef3,coef2,coef1,coef0,2);
    solution3 = solution3th(coef3,coef2,coef1,coef0,3);
    time1 = time1 + toc;

    % Bilinear interpolation
    tic
    x_value1 = fzero(@(x) coef3*x^3+coef2*x^2+coef1*x+coef0, 0);
    x_value2 = fzero(@(x) coef3*x^3+coef2*x^2+coef1*x+coef0, -1000);
    x_value3 = fzero(@(x) coef3*x^3+coef2*x^2+coef1*x+coef0, 1000);
    time2 = time2 + toc;

end

% Display
disp(['Averaged time by exact method: ',num2str(time1/N),'secs']);
disp(['Averaged time by bilinear interp: ',num2str(time2/N),'secs']);
```

The final time per iteration was $time1 = 0.0046$ and $time2 = 0.00010836$, both of them in seconds. It means that the exact method is 40 times faster. In addition, the bilinear interpolation method was not always able to find the three solution; although when it did it, the magnitude difference between both methods was neglectable.

B.6 Registration

Example in Matlab, where a 1-unit square is transformed according to the four individual affine transformations. The code uses the pre-multiplication of the 2D affine transformation matrix.

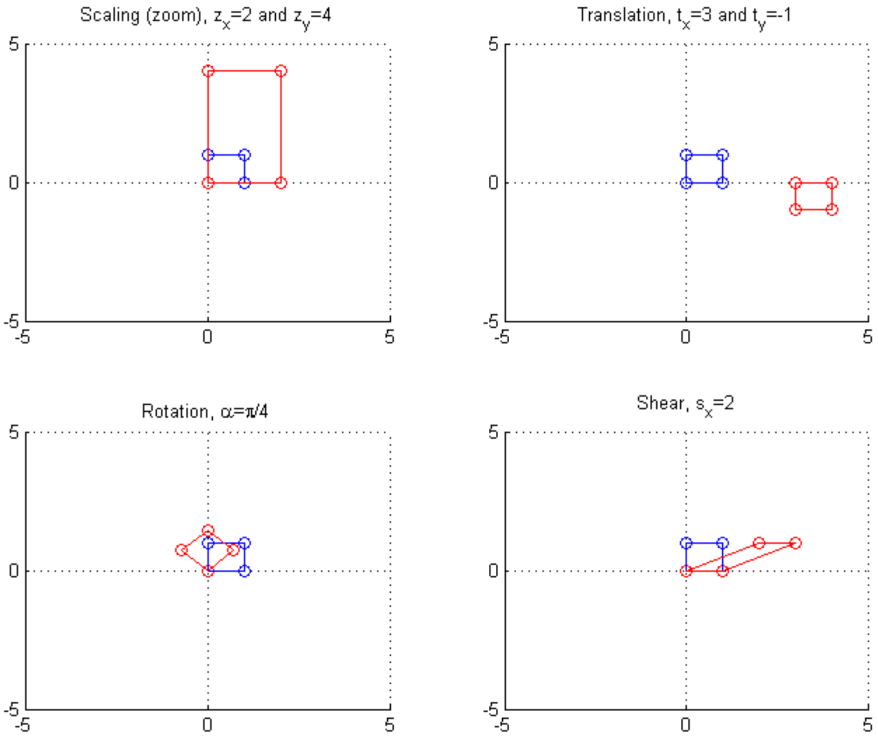


Figure B.5: Affine Transformation example in 2D. The original shape is a blue square with vertices [0,0], [0,1], [1,0] and [1,1]. In the top-left figure, it is applied a scaling of {2,4}. In the top-righ figure, it is applied a translation of {3,-1}. In the top-left figure, it is applied a rotation of $\pi/4$. In the bottom-right figure, it is applied a shear of 2.

```
% Original square
X=[0 0 1 1;0 1 0 1];
X_ext=[X; 1 1 1 1];

% Scaling
scaling=[2 4];

A=[scaling(1) 0; 0 scaling(2)];
A_ext=[A(1,:) 0; A(2,:) 0; 0 0 1];

Y_ext=A_ext*X_ext;
Y=Y_ext(1:2,:);

figure,
subplot(2,2,1)
title([’Scaling (zoom), z_{x}=’,num2str(scaling(1)),...
’ and z_{y}=’,num2str(scaling(2))])
hold on
scatter(X(1,:),X(2,:),’b’);
scatter(Y(1,:),Y(2,:),’r’);
line([X(1,1) X(1,2) X(1,1) X(1,3) X(1,4) X(1,3) X(1,4) X(1,2)],...
[X(2,1) X(2,2) X(2,1) X(2,3) X(2,4) X(2,3) X(2,4) X(2,2)],...
’Color’,’b’)
line([Y(1,1) Y(1,2) Y(1,1) Y(1,3) Y(1,4) Y(1,3) Y(1,4) Y(1,2)],...
[Y(2,1) Y(2,2) Y(2,1) Y(2,3) Y(2,4) Y(2,3) Y(2,4) Y(2,2)],...
’Color’,’r’)
hold off
axis([-5 5 -5 5])
grid on


% Translation
trans=[3 -1];

A=[1 0; 0 1];
A_ext=[A(1,:) trans(1); A(2,:) trans(2); 0 0 1];

Y_ext=A_ext*X_ext;
Y=Y_ext(1:2,:);

subplot(2,2,2)
title([’Translation, t_{x}=’,num2str(trans(1)),...
’ and t_{y}=’,num2str(trans(2))])
hold on
scatter(X(1,:),X(2,:),’b’);
scatter(Y(1,:),Y(2,:),’r’);
line([X(1,1) X(1,2) X(1,1) X(1,3) X(1,4) X(1,3) X(1,4) X(1,2)],...
[X(2,1) X(2,2) X(2,1) X(2,3) X(2,4) X(2,3) X(2,4) X(2,2)],...
’Color’,’b’)
line([Y(1,1) Y(1,2) Y(1,1) Y(1,3) Y(1,4) Y(1,3) Y(1,4) Y(1,2)],...
[Y(2,1) Y(2,2) Y(2,1) Y(2,3) Y(2,4) Y(2,3) Y(2,4) Y(2,2)],...
’Color’,’r’)
hold off
axis([-5 5 -5 5])
```

```
grid on


% Rotation (yaw)
angle=pi/4;

A=[cos(angle) -sin(angle); sin(angle) cos(angle)];
A_ext=[A(1,:) 0; A(2,:) 0; 0 0 1];

Y_ext=A_ext*X_ext;
Y=Y_ext(1:2,:);

subplot(2,2,3)
title(['Rotation, \alpha=\pi/',num2str(pi/angle(1))])
hold on
scatter(X(1,:),X(2,:),'b');
scatter(Y(1,:),Y(2,:),'r');
line([X(1,1) X(1,2) X(1,1) X(1,3) X(1,4) X(1,3) X(1,4) X(1,2)],...
[X(2,1) X(2,2) X(2,1) X(2,3) X(2,4) X(2,3) X(2,4) X(2,2)],...
'Color','b')
line([Y(1,1) Y(1,2) Y(1,1) Y(1,3) Y(1,4) Y(1,3) Y(1,4) Y(1,2)],...
[Y(2,1) Y(2,2) Y(2,1) Y(2,3) Y(2,4) Y(2,3) Y(2,4) Y(2,2)],...
'Color','r')
hold off
axis([-5 5 -5 5])
grid on


% Shear
shear=[2];

A=[1 shear(1); 0 1];
A_ext=[A(1,:) 0; A(2,:) 0; 0 0 1];

Y_ext=A_ext*X_ext;
Y=Y_ext(1:2,:);

subplot(2,2,4)
title(['Shear, s_{x}=',num2str(shear(1))])
hold on
scatter(X(1,:),X(2,:),'b');
scatter(Y(1,:),Y(2,:),'r');
line([X(1,1) X(1,2) X(1,1) X(1,3) X(1,4) X(1,3) X(1,4) X(1,2)],...
[X(2,1) X(2,2) X(2,1) X(2,3) X(2,4) X(2,3) X(2,4) X(2,2)],...
'Color','b')
line([Y(1,1) Y(1,2) Y(1,1) Y(1,3) Y(1,4) Y(1,3) Y(1,4) Y(1,2)],...
[Y(2,1) Y(2,2) Y(2,1) Y(2,3) Y(2,4) Y(2,3) Y(2,4) Y(2,2)],...
'Color','r')
hold off
axis([-5 5 -5 5])
grid on
```

APPENDIX C

SPM

This chapter includes a deeper explanation of the variables and the Matlab code of the file *spm_preproc8T1T2.m* of the 'SegT1T2' toolbox for SPM8.

A common variable struct used for both MRI data and templates corresponds to the output of the function *spm_vol()*. From the complete filename of the volumes, this function creates a struct with the following fields:

- V.fname <string> path, name and extension of the file with the volumes, which can be stored in either format *.nii* or *.hdr/.img*.
- V.mat <4x4 double> pre-multiplication affine transformation matrix.
- V.dim <1x3 double> dimensions of the original volume for the x, y and z coordinates.
- V.dt <1x2 double> format of the NIFTI-1 files according to *spm_type()*.
- V.pinfo <3x1 double> scaling factor of each plane.

The following two sections present the main input and output variables with information about its dimensions and a short description. It can be assumed that: $N = 2$, $Kb = 6$, and $K = 15$.

C.1 Input Variables

The input variable is *obj*, which main fields are:

Volumes: MRI individual brain volumes to segment. There are two modalities, T_1 and T_2, which are stored in *obj.image(1)* and *obj.image(2)*. Afterwards, the volumes are stored in V=*obj.image*.

- obj.image <2x1 struct> individual volumes in the format of *spm_vol()*.

Tissue Probability Maps: The struct is obtained from the function *spm_load_priors8()* that loads the volumes into memory with *spm_slice_vol()*.

- obj.tpm.V <Kbx1 struct> templates in the format of *spm_vol()*.
- obj.tpm.M <4x4 double> pre-multiplication affine transformation matrix.
- obj.tpm.dat <Kbx1 cell> mapped volumes in a 3D matrix.
- obj.tpm.bg1 <Kbx1 double> background value for each template.
- obj.tpm.bg2 <Kbx1 double> background value for each template.
- obj.tpm.tiny <1x1 double> tiny value $\sim 10^{-3}$.
- obj.tpm.deg <1x1 double> B-spline degree.

Affine Transformation: Transformation between the voxels of the templates and the voxels of the individual volumes to segment.

- obj.Affine <4x4 double> pre-multiplication affine transformation matrix.

Gaussians per tissue: Look-up table with the number of clusters associated to each tissue class.

- obj.lkp <Kx1 double> *lkp=[1,1,2,2,3,3,4,4,4,5,5,5,5,6,6];*

The transformation matrices of the different variables must accomplish that: *obj.tpm.M* $* X = Affine * V(1).mat * Y$, where X corresponds to a volume in the voxel-template coordinates and Y is a volume in the voxel-individual coordinates. As, both channels are previously registered in the same space, the affine transformation for both modalities is the same, $V(1).mat = V(2).mat$.

C.2 Original Code

Part of the code from the file *spm_preproc8.m* that corresponds to the *Seg* toolbox. This extract shows how the values for the mixture parameters are estimated.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Estimate cluster parameters
%------------------------------------------------------------
for subit=1:20,
    oll  = ll;
    mom0 = zeros(K,1)+tiny;
    mom1 = zeros(N,K);
    mom2 = zeros(N,N,K);
    ll   = llr+llrb;
    for z=1:length(z0),
        if ¬buf(z).nm, continue; end
        q = likelihoods(buf(z).f,buf(z).bf,mg,mn,vr);
        for k1=1:Kb,
            b = double(buf(z).dat(:,k1));
            for k=find(lkp==k1),
                q(:,k) = q(:,k).*b;
            end
            clear b
        end
        sq = sum(q,2)+tiny;
        ll = ll + sum(log(sq + tiny));
        cr = zeros(size(q,1),N);
        for n=1:N,
            cr(:,n)  = double(buf(z).f{n}.*buf(z).bf{n});
        end
        for k=1:K, % Moments
            q(:,k)      = q(:,k)./sq;
            mom0(k)     = mom0(k)     + sum(q(:,k));
            mom1(:,k)   = mom1(:,k)   + (q(:,k)'*cr)';
            mom2(:,:,k) = mom2(:,:,k) + (repmat(q(:,k),1,N).*cr)'*cr;
        end
        clear cr
    end

   %fprintf('MOG:\t%g\t%g\t%g\n', ll,llr,llrb);

    % Priors
   %nmom = struct('mom0',mom0,'mom1',mom1,'mom2',mom2);
    if exist('omom','var') && isfield(omom,'mom0') && ...
        numel(omom.mom0) == numel(mom0),
        mom0 = mom0 + omom.mom0;
        mom1 = mom1 + omom.mom1;
        mom2 = mom2 + omom.mom2;
    end

    % Mixing proportions, Means and Variances
```

```
        for k=1:K,
            tmp       = mom0(lkp==lkp(k));
            mg(k)     = (mom0(k)+tiny)/sum(tmp+tiny);
            mn(:,k)   = mom1(:,k)/(mom0(k)+tiny);
            vr(:,:,k) = (mom2(:,:,k) - mom1(:,k)*mom1(:,k)'/mom0(k))/ ...
                              (mom0(k)+tiny) + vr0;
        end

        if subit>1 || iter>1,
            spm_chi2_plot('Set',ll);
        end
        if ll-oll<tol1*nm,
            % Improvement is small, so go to next step
            break;
        end
    end
```

C.3 Modified Code

Part of the code from the file *spm_preproc8T1T2.m* that corresponds to the *SegT1T2* toolbox. This extract shows how the values for the mixture parameters are estimated.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Estimate cluster parameters
%-------------------------------------------------------------
for subit=1:20,
    oll  = ll;
    mom0 = zeros(K,1)+tiny;
    mom1 = zeros(N,K);
    mom2 = zeros(N,N,K);
    mom1c = zeros(N,K);
    mom2c = zeros(N,N,K);
    ll   = llr+llrb;
    for z=1:length(z0),
        if ¬buf(z).nm, continue; end
        q = likelihoods(buf(z).f,buf(z).bf,mg,mn,vr);
        for k1=1:Kb,
            b = double(buf(z).dat(:,k1));
            for k=find(lkp==k1),
                q(:,k) = q(:,k).*b;
            end
            clear b
        end
        sq = sum(q,2)+tiny;
                for k=1:K,
            q(:,k) = q(:,k)./sq;
                end
        ll = ll + sum(log(sq + tiny));
        cr = zeros(size(q,1),N);
        for n=1:N,
            cr(:,n)  = double(buf(z).f{n}.*buf(z).bf{n});
        end
        for k=1:K, % Moments
            % Non-centered moments
            mom0(k)     = mom0(k)     + sum(q(:,k));
            mom1(:,k)   = mom1(:,k)   + (q(:,k)'*cr)';
            mom2(:,:,k) = mom2(:,:,k) + (repmat(q(:,k),1,N).*cr)'*cr;
            % Central moments
            crc = cr - repmat(mn(:,k)',size(q,1),1);
            mom1c(:,k)   = mom1c(:,k)+(q(:,k)'*crc)';
            mom2c(:,:,k) = mom2c(:,:,k)+(repmat(q(:,k),1,N).*crc)'*crc;
        end
        clear cr crc
    end

   %fprintf('MOG:\t%g\t%g\t%g\n', ll,llr,llrb);

   % Priors
```

```
    %nmom = struct('mom0',mom0,'mom1',mom1,'mom2',mom2);
    if exist('omom','var') && isfield(omom,'mom0') && ...
        numel(omom.mom0) == numel(mom0),
        mom0 = mom0 + omom.mom0;
        mom1 = mom1 + omom.mom1;
        mom2 = mom2 + omom.mom2;
    end

    % Mixing proportions, Means and Variances
    mgX = zeros(size(mg));
    mnX = zeros(size(mn));
    vrX = zeros(size(vr));
    ovr = vr;
    for k=1:K,
        %%%%%%%%%%%%%  Original Equations  %%%%%%%%%%%%%%%%
        % ------------  Mixing coefficient  ---------------
        tmp = mom0(lkp==lkp(k));
        mgX(k) = (mom0(k)+tiny)/sum(tmp+tiny);
        % ------------------  Mean  -----------------------
        mnX(:,k) = mom1(:,k)/(mom0(k)+tiny);
        % -----------------  Variance  -------------------
        vrX(:,:,k) = (mom2(:,:,k) - ...
            mom1(:,k)*mom1(:,k)'/mom0(k))/(mom0(k)+tiny) + vr0;
        % For estability
        vrX(1,1,k) = mom2c(1,1,k)/(mom0(k)+tiny);
        vrX(2,2,k) = mom2c(2,2,k)/(mom0(k)+tiny);
        %%%%%%%%%%%%%  Modified Equations  %%%%%%%%%%%%%%%%
        % ------------  Mixing coefficient  ---------------
        mg(k) = mgX(k);
        % ------------------  Mean  -----------------------
        coefm1 = -mom1c(2,k);
        coefm2 = -mom1c(1,k);
        mn(1,k) = mnX(1,k) ...
               + (ovr(1,2,k)*coefm1)/(ovr(2,2,k)*mom0(k)+tiny);
        mn(2,k) = mnX(2,k) ...
               + (ovr(1,2,k)*coefm2)/(ovr(1,1,k)*mom0(k)+tiny);
        % -----------------  Variance  -------------------
        % >> Variance
        coefs1 = ovr(1,2,k)*mom0(k) ...
                 + (ovr(1,2,k)/(ovr(2,2,k)+tiny))*mom2c(2,2,k) ...
                 - 2*mom2c(1,2,k);
        coefs2 = ovr(1,2,k)*mom0(k) ...
                 + (ovr(1,2,k)/(ovr(1,1,k)+tiny))*mom2c(1,1,k) ...
                 - 2*mom2c(1,2,k);

        vr(1,1,k) = vrX(1,1,k) ...
               + (ovr(1,2,k)*coefs1)/(ovr(2,2,k)*mom0(k)+tiny);
        vr(2,2,k) = vrX(2,2,k) ...
               + (ovr(1,2,k)*coefs2)/(ovr(1,1,k)*mom0(k)+tiny);
        % >> Cross-variance
        % Coefficients
        coef3 = mom0(k);
        coef2 = -mom2c(1,2,k);
        coef1 = -ovr(1,1,k)*ovr(2,2,k)*mom0(k) ...
           + ovr(2,2,k)*mom2c(1,1,k) + ovr(1,1,k)*mom2c(2,2,k);
        coef0 = -ovr(1,1,k)*ovr(2,2,k)*mom2c(1,2,k);
        % Look for the correct solution
        solution = solution3th(coef3,coef2,coef1,coef0,1);
        if ((vr(1,1,k)*vr(2,2,k)-solution^2)<tiny) || ...
                             abs(imag(solution))>1e-4
            solution = solution3th(coef3,coef2,coef1,coef0,2);
            if (vr(1,1,k)*vr(2,2,k)-solution^2)<tiny || ...
                                abs(imag(solution))>1e-4
                solution = solution3th(coef3,coef2,coef1,coef0,3);
                if (vr(1,1,k)*vr(2,2,k)-solution^2)<tiny || ...
                                    abs(imag(solution))>1e-4
                    solution = vrX(1,2,k);
                end
            end
        end
        % Give values
        vr(1,2,k) = real(solution);
        vr(2,1,k) = vr(1,2,k);
        % Ensure estability
        vr(:,:,k) = vr(:,:,k) + vr0;
    end

    if subit>1 || iter>1,
        spm_chi2_plot('Set',ll);
    end
    if ll-oll<tol1*nm,
        % Improvement is small, so go to next step
        break;
    end
end
```

C.4 Modified Code (version 2)

Part of the code from the file *spm_preproc8T1T2.m* that corresponds to the *SegT1T2* toolbox. This extract shows how the values for the mixture parameters are estimated. This second version corresponds to an implementation where the updating equations use the most-updated values of the current iteration, and not the values from the previous iteration.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Estimate cluster parameters
%-------------------------------------------------------------
for subit=1:20,
    oll  = ll;
    mom0 = zeros(K,1)+tiny;
    mom1 = zeros(N,K);
    mom2 = zeros(N,N,K);
    ll   = llr+llrb;
    for z=1:length(z0),
        if ¬buf(z).nm, continue; end
        q = likelihoods(buf(z).f,buf(z).bf,mg,mn,vr);
        for k1=1:Kb,
            b = double(buf(z).dat(:,k1));
            for k=find(lkp==k1),
                q(:,k) = q(:,k).*b;
            end
            clear b
        end
        sq = sum(q,2)+tiny;
for k=1:K,
            q(:,k) = q(:,k)./sq;
end
        ll = ll + sum(log(sq + tiny));
        cr = zeros(size(q,1),N);
        for n=1:N,
            cr(:,n)  = double(buf(z).f{n}.*buf(z).bf{n});
        end
        for k=1:K, % Moments
            % Non-centered moments
            mom0(k)     = mom0(k)    + sum(q(:,k));
            mom1(:,k)   = mom1(:,k)  + (q(:,k)'*cr)';
            mom2(:,:,k) = mom2(:,:,k)+(repmat(q(:,k),1,N).*cr)'*cr;
        end

        clear cr
    end

    % Central moments
    mom1c = zeros(N,K);
    mom2c = zeros(N,N,K);
    for z=1:length(z0),
        if ¬buf(z).nm, continue; end
        q = likelihoods(buf(z).f,buf(z).bf,mg,mn,vr);
        for k1=1:Kb,
            b = double(buf(z).dat(:,k1));
            for k=find(lkp==k1),
                q(:,k) = q(:,k).*b;
            end
            clear b
        end
        sq = sum(q,2)+tiny;
for k=1:K,
            q(:,k) = q(:,k)./sq;
end
        cr = zeros(size(q,1),N);
        for n=1:N,
            cr(:,n)  = double(buf(z).f{n}.*buf(z).bf{n});
        end
        for k=1:K, % Moments
            % Central moments
            aux_mn = mom1(:,k)/(mom0(k)+tiny);
            crc = cr - repmat(aux_mn',size(q,1),1);
            mom1c(:,k)   = mom1c(:,k)   + (q(:,k)'*crc)';
            mom2c(:,:,k) = mom2c(:,:,k)+(repmat(q(:,k),1,N).*crc)'*crc;
        end

        clear cr crc aux_mn
    end


   %fprintf('MOG:\t%g\t%g\t%g\n', ll,llr,llrb);

   % Priors
   %nmom = struct('mom0',mom0,'mom1',mom1,'mom2',mom2);
    if exist('omom','var') && isfield(omom,'mom0') ...
        && numel(omom.mom0) == numel(mom0),
        mom0 = mom0 + omom.mom0;
        mom1 = mom1 + omom.mom1;
        mom2 = mom2 + omom.mom2;
    end

%                 load('execution2.mat')

    % Mixing proportions, Means and Variances
    mgX = zeros(size(mg));
    mnX = zeros(size(mn));
    vrX = zeros(size(vr));
    %ovr = vr;
    for k=1:K,
        %%%%%%%%%%%%%  Original Equations  %%%%%%%%%%%%%%%%
        % ------------ Mixing coefficient ---------------
        tmp = mom0(lkp==lkp(k));
        mgX(k) = (mom0(k)+tiny)/sum(tmp+tiny);
        % ------------------- Mean -----------------------
        mnX(:,k) = mom1(:,k)/(mom0(k)+tiny);
        % ------------------ Variance -------------------
        vrX(:,:,k) = (mom2(:,:,k) - ...
            mom1(:,k)*mom1(:,k)'/mom0(k))/(mom0(k)+tiny) + vr0;
        % For estability
        %vrX(1,1,k) = mom2c(1,1,k)/(mom0(k)+tiny);
        %vrX(2,2,k) = mom2c(2,2,k)/(mom0(k)+tiny);
        %%%%%%%%%%%%% Modified Equations %%%%%%%%%%%%%%%%
        % ------------ Mixing coefficient ---------------
        mg(k) = mgX(k);
        % ------------------ Mean -----------------------
        coefm1 = -mom1c(2,k);
        coefm2 = -mom1c(1,k);
        mn(1,k) = mnX(1,k) ...
               + (vr(1,2,k)*coefm1)/(vr(2,2,k)*mom0(k)+tiny);
        mn(2,k) = mnX(2,k) ...
               + (vr(1,2,k)*coefm2)/(vr(1,1,k)*mom0(k)+tiny);
        % ----------------- Variance -------------------
        % >> Variance
        coefs1 = vr(1,2,k)*mom0(k) ...
                 + (vr(1,2,k)/(vr(2,2,k)+tiny))*mom2c(2,2,k) ...
                 - 2*mom2c(1,2,k);
        coefs2 = vr(1,2,k)*mom0(k) ...
                 + (vr(1,2,k)/(vr(1,1,k)+tiny))*mom2c(1,1,k) ...
                 - 2*mom2c(1,2,k);

        vr(1,1,k) = vrX(1,1,k) ...
               + (vr(1,2,k)*coefs1)/(vr(2,2,k)*mom0(k)+tiny);
        vr(2,2,k) = vrX(2,2,k) ...
               + (vr(1,2,k)*coefs2)/(vr(1,1,k)*mom0(k)+tiny);
        % >> Cross-variance
        % Coefficients
        coef3 = mom0(k);
        coef2 = -mom2c(1,2,k);
        coef1 = -vr(1,1,k)*vr(2,2,k)*mom0(k) ...
           + vr(2,2,k)*mom2c(1,1,k) + vr(1,1,k)*mom2c(2,2,k);
        coef0 = -vr(1,1,k)*vr(2,2,k)*mom2c(1,2,k);
        % Look for the correct solution
        solution = solution3th(coef3,coef2,coef1,coef0,1);
        if ((vr(1,1,k)*vr(2,2,k)-solution^2)<tiny) || ...
            abs(imag(solution))>1e-4
            solution = solution3th(coef3,coef2,coef1,coef0,2);
            if (vr(1,1,k)*vr(2,2,k)-solution^2)<tiny || ...
                abs(imag(solution))>1e-4
                solution = solution3th(coef3,coef2,coef1,coef0,3);
                if (vr(1,1,k)*vr(2,2,k)-solution^2)<tiny || ...
                    abs(imag(solution))>1e-4
                    solution = vrX(1,2,k);
                end
            end
        end
        % Give values
        vr(1,2,k) = real(solution);
        vr(2,1,k) = vr(1,2,k);
        % Ensure estability
        vr(:,:,k) = vr(:,:,k) + vr0;

    end

```

```

     if subit>1 || iter>1,
        spm_chi2_plot('Set',ll);
    end
    if ll-oll<tol1*nm,
        % Improvement is small, so go to next step
        break;
    end
end
```

Appendix D

Results & Validation

This chapter includes the results of the segmentation for the original method and modified versions with more details.

First, it is included the evolution of the mixture parameter values at each iteration in the segmentation of the T_1 and T_2 MRI scan from the subject `f4395`. The used methods comprise the original method and the four versions of the modified method.

Secondly, it is presented a representation of the clusters for the original and the four versions of the modified method in the segmentation of the MR scan from the subject `f4395`. It must be highlighted that due to the strong bias field correction, the intensity values are much different.

The third section presents 3 tables with the Dice scores and likelihood values of the original and modified methods in the segmentation of BrainWeb phantoms with several noise levels. In addition, it is included the results for several probability processing methods, i.e. majority voting and threshold, where the extracranial class is a compendium of ST, bone and BG.

The fourth section presents and overlapped representation of the confusion matrix elements for each voxel in the segmentation of the BrainWeb phantoms by original and v.3 modified method.

Finally, the fourth section presents the effect of the atrophy in the volumes of several brain tissues.

D.1 Mixture parameters at each iteration for f4395.

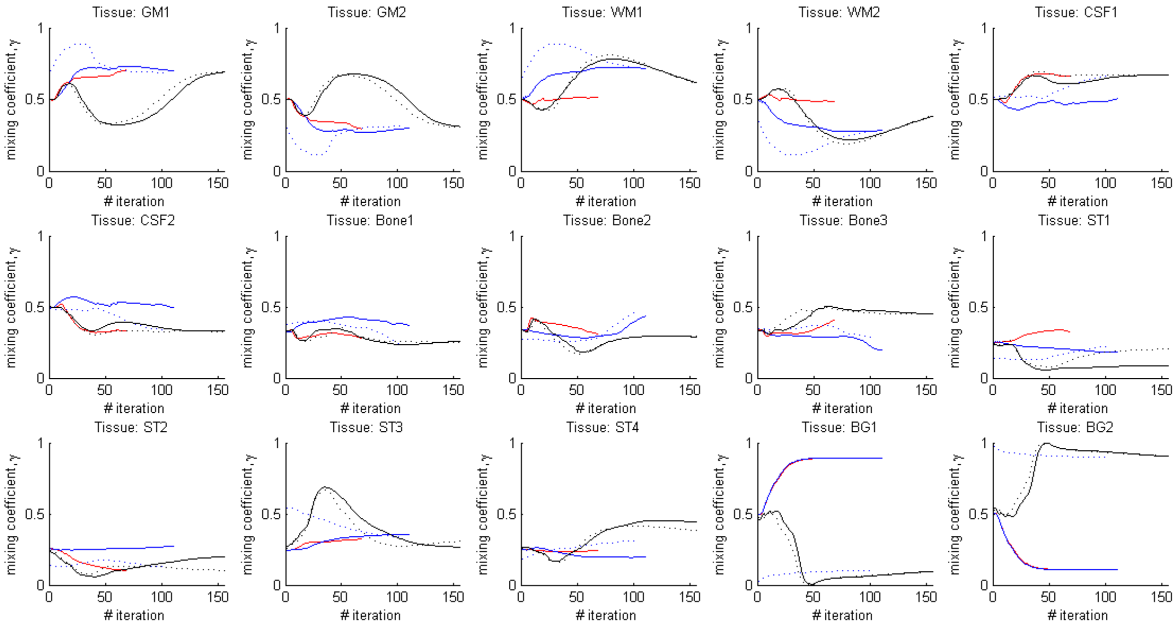


Figure D.1: Mixture Coefficient through iterations. Results for the segmentation of the T_1 and T_2 MRI scan from the subject f4395. The red line corresponds to the original method. The blue color correspond to the modified versions with slow value propagation, while the 'fast' propagation versions are in black. The solid lines correspond to the versions with original starting equations, while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization. The first 15-20 first iterations are removed as they correspond to the initialization.

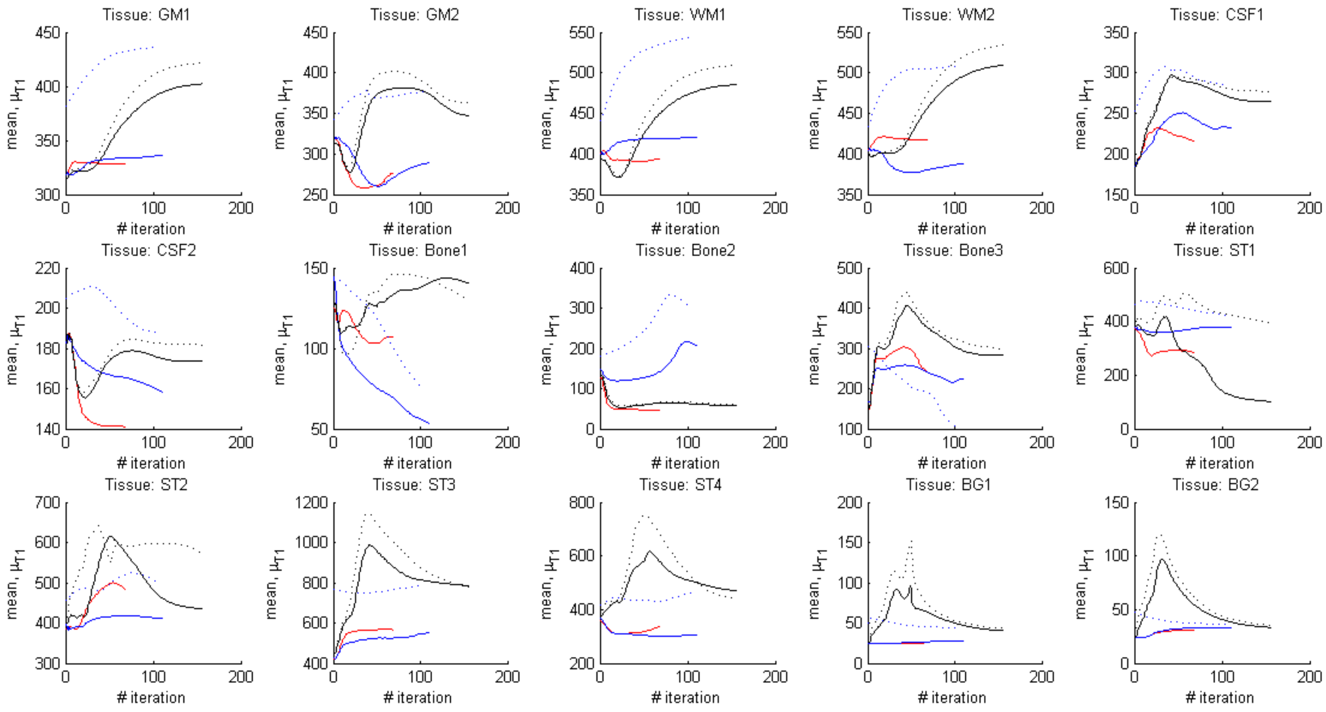


Figure D.2: T1 Mean Value through iterations. Results for the segmentation of the T_1 and T_2 MRI scan from the subject f4395. The red line corresponds to the original method. The blue color correspond to the modified versions with slow value propagation, while the 'fast' propagation versions are in black. The solid lines correspond to the versions with original starting equations, while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization. The first 15-20 first iterations are removed as they correspond to the initialization.

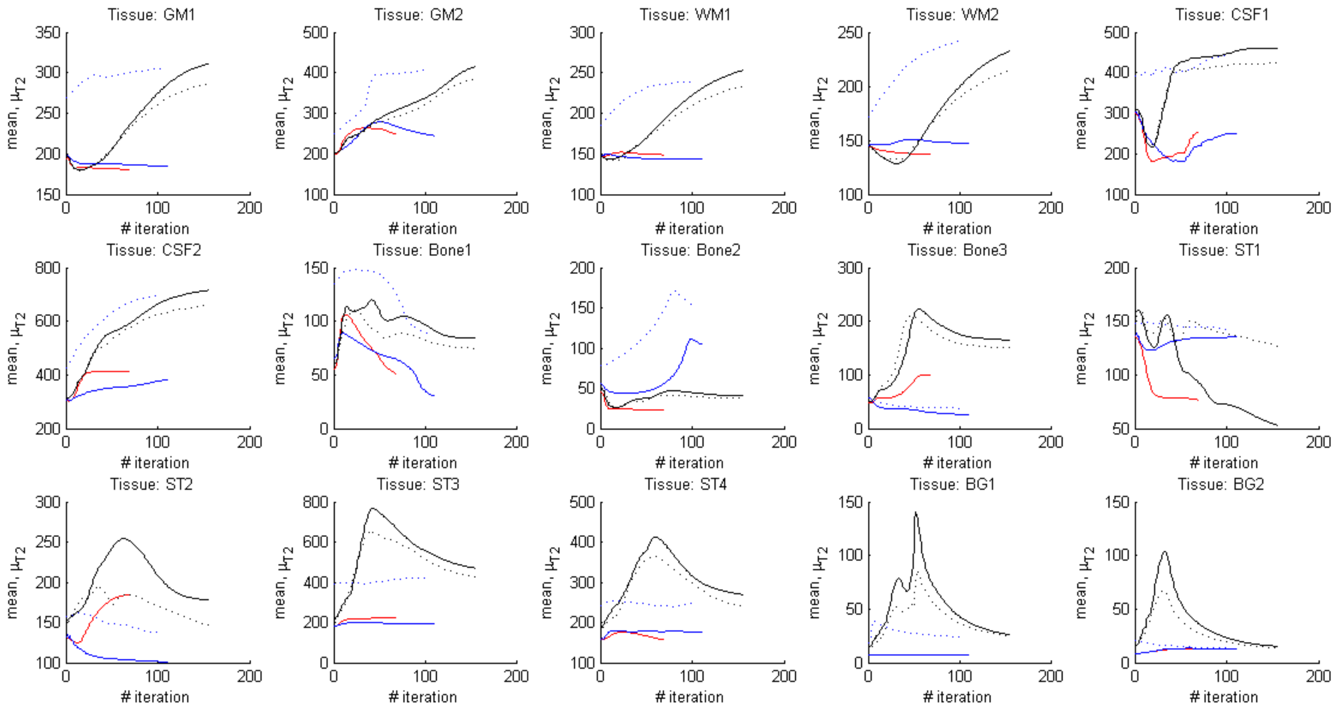


Figure D.3: T2 Mean Value through iterations. Results for the segmentation of the T_1 and T_2 MRI scan from the subject f4395. The red line corresponds to the original method. The blue color correspond to the modified versions with slow value propagation, while the 'fast' propagation versions are in black. The solid lines correspond to the versions with original starting equations, while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization. The first 15-20 first iterations are removed as they correspond to the initialization.

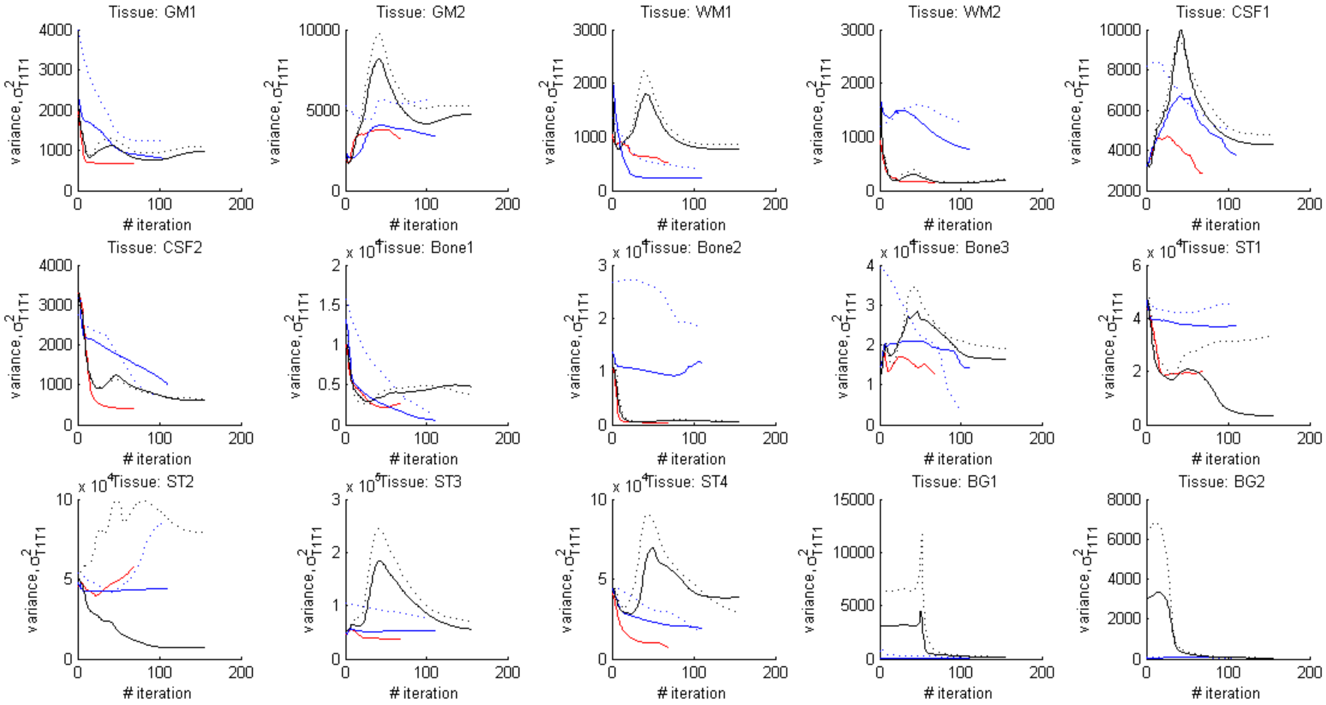


Figure D.4: T1 Variance Value through iterations. Results for the segmentation of the T_1 and T_2 MRI scan from the subject f4395. The red line corresponds to the original method. The blue color correspond to the modified versions with slow value propagation, while the 'fast' propagation versions are in black. The solid lines correspond to the versions with original starting equations, while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization. The first 15-20 first iterations are removed as they correspond to the initialization.

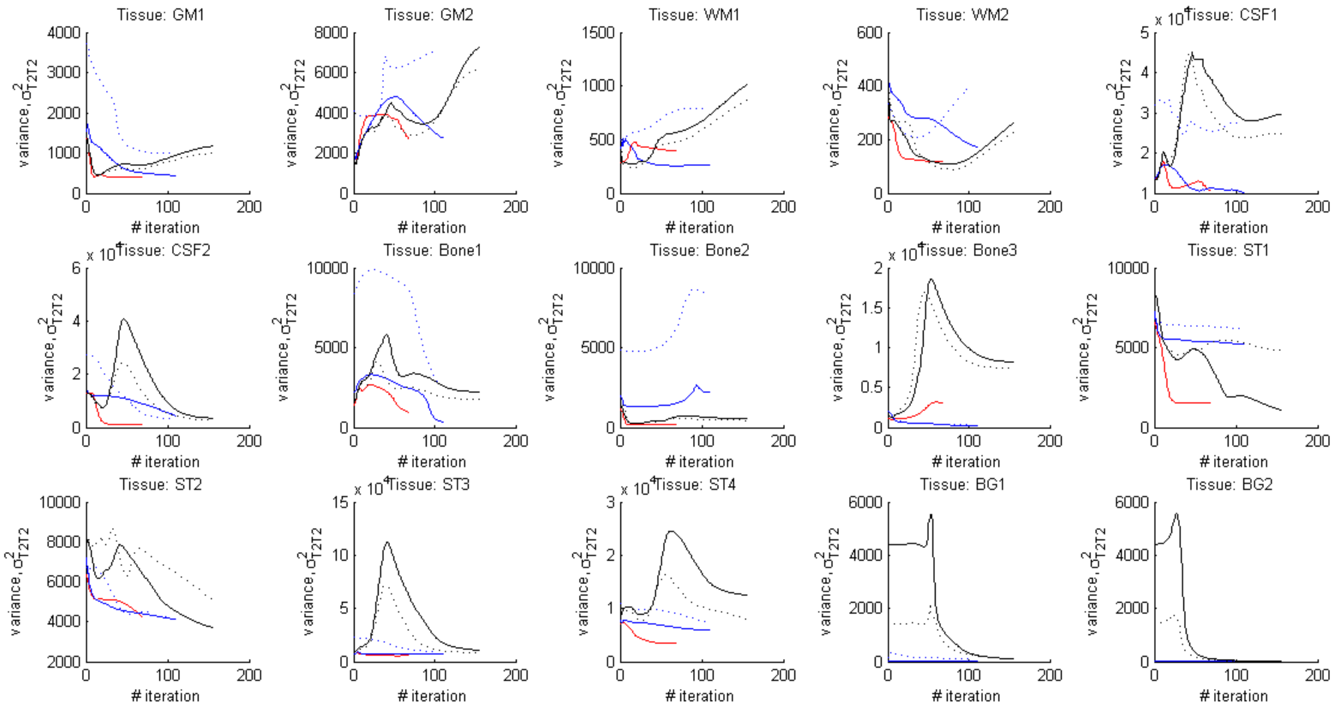


Figure D.5: T2 Variance Value through iterations. Results for the segmentation of the T_1 and T_2 MRI scan from the subject f4395. The red line corresponds to the original method. The blue color correspond to the modified versions with slow value propagation, while the 'fast' propagation versions are in black. The solid lines correspond to the versions with original starting equations, while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization. The first 15-20 first iterations are removed as they correspond to the initialization.

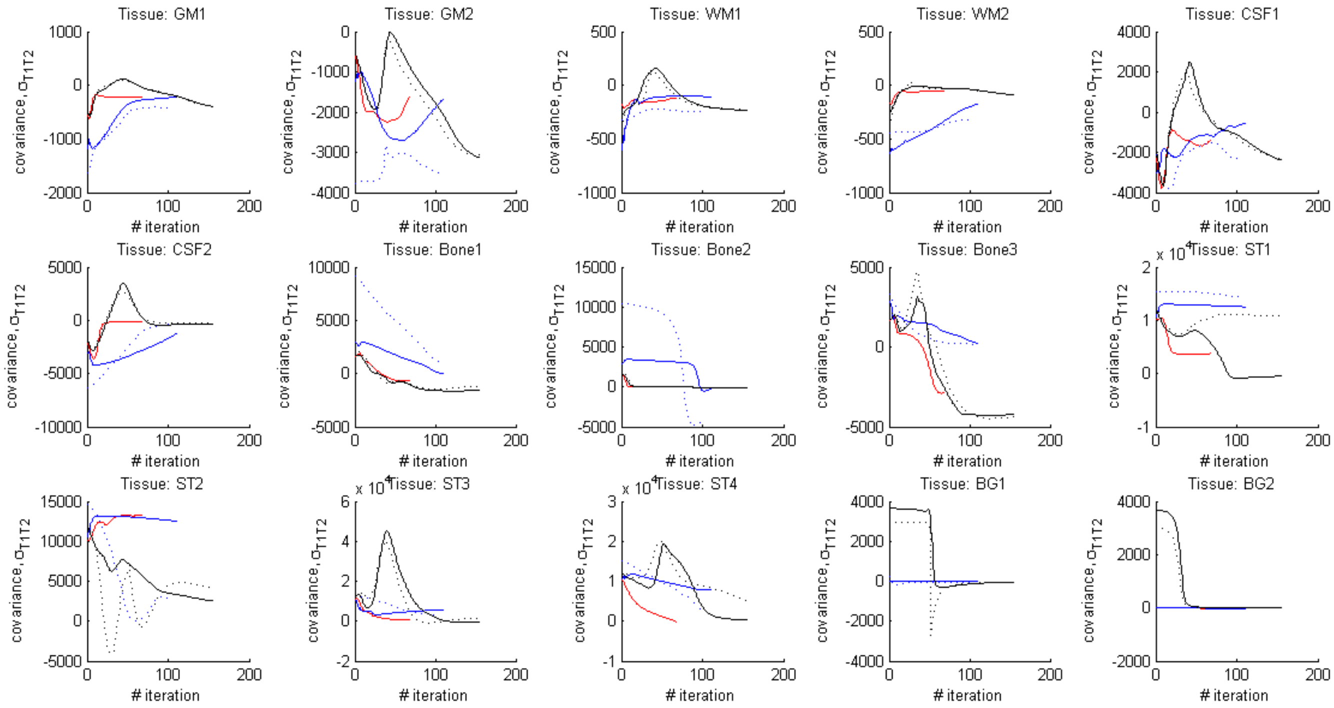


Figure D.6: T1-T2 Covariance Value through iterations. Results for the segmentation of the T_1 and T_2 MRI scan from the subject f4395. The red line corresponds to the original method. The blue color correspond to the modified versions with slow value propagation, while the 'fast' propagation versions are in black. The solid lines correspond to the versions with original starting equations, while the dotted ones are associated to the versions that apply modified equations also for the parameter initialization. The first 15-20 first iterations are removed as they correspond to the initialization.

D.2 Representation of the clusters for all the tissue classes.

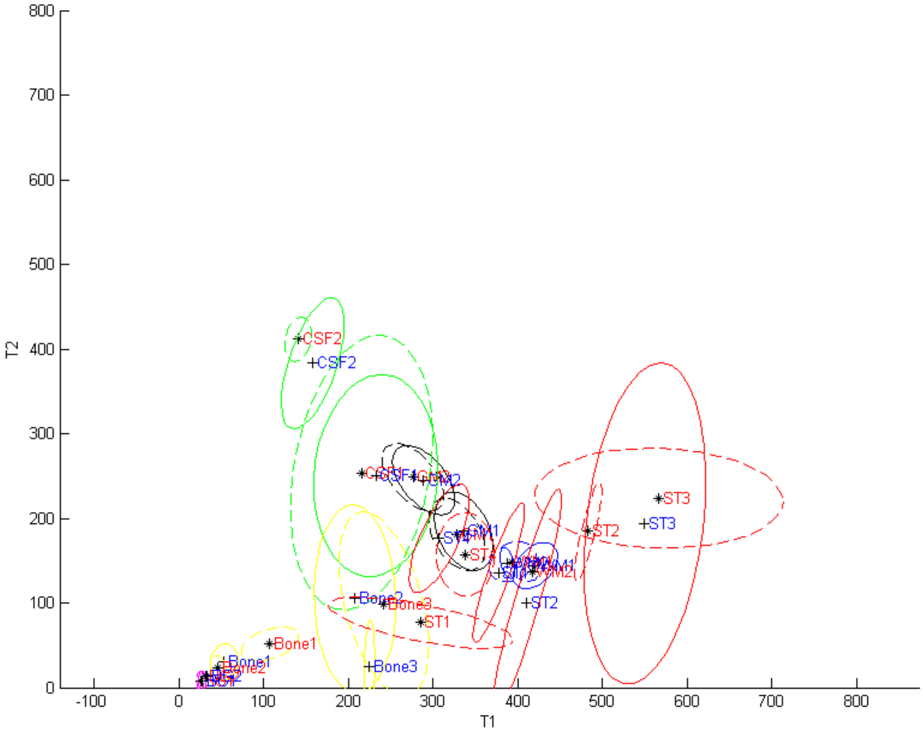


Figure D.7: Representation of the clusters for all the tissue classes done bu the version 1 of the modified method. The lines correspond to the contour of the Gaussian cut at FWHM, and weighted by the mixing coefficient. The contours of the clusters done by the original method are presented with dotted lines, the centers with the symbol *, and the text labels in red. The version 1 of the modified method presents the contours with solid lines, the center with the symbol +, and the text labels in blue. The depicted tissues comprise GM (black), WM (blue), CSF (green), ST (red), bone (yellow), and BG (magenta).

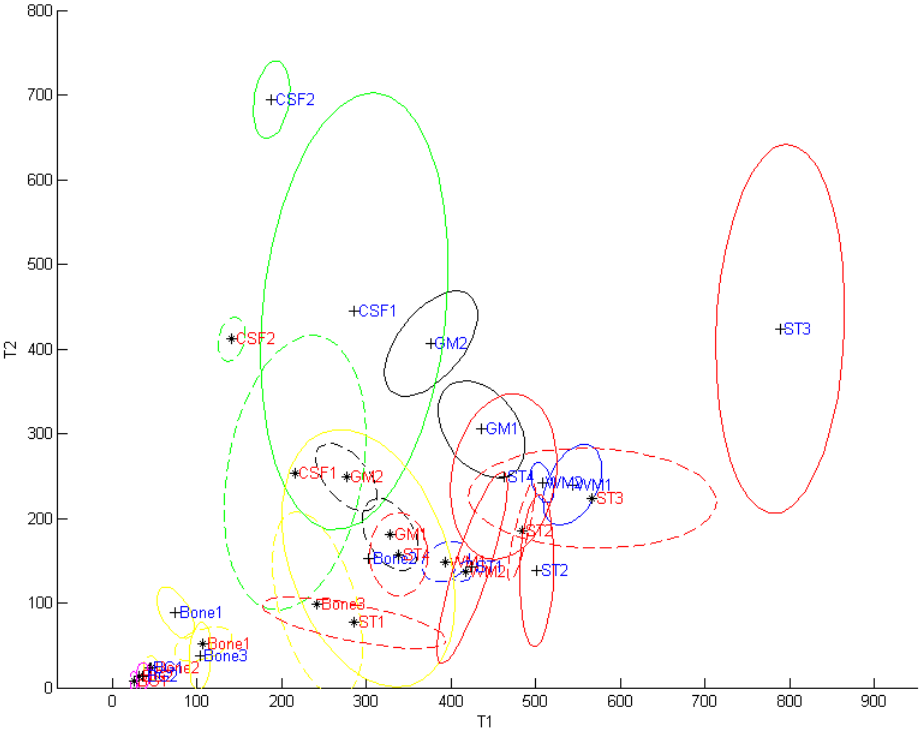


Figure D.8: Representation of the clusters for all the tissue classes done bu the version 2 of the modified method. The lines correspond to the contour of the Gaussian cut at FWHM, and weighted by the mixing coefficient. The contours of the clusters done by the original method are presented with dotted lines, the centers with the symbol *, and the text labels in red. The version 2 of the modified method presents the contours with solid lines, the center with the symbol +, and the text labels in blue. The depicted tissues comprise GM (black), WM (blue), CSF (green), ST (red), bone (yellow), and BG (magenta).

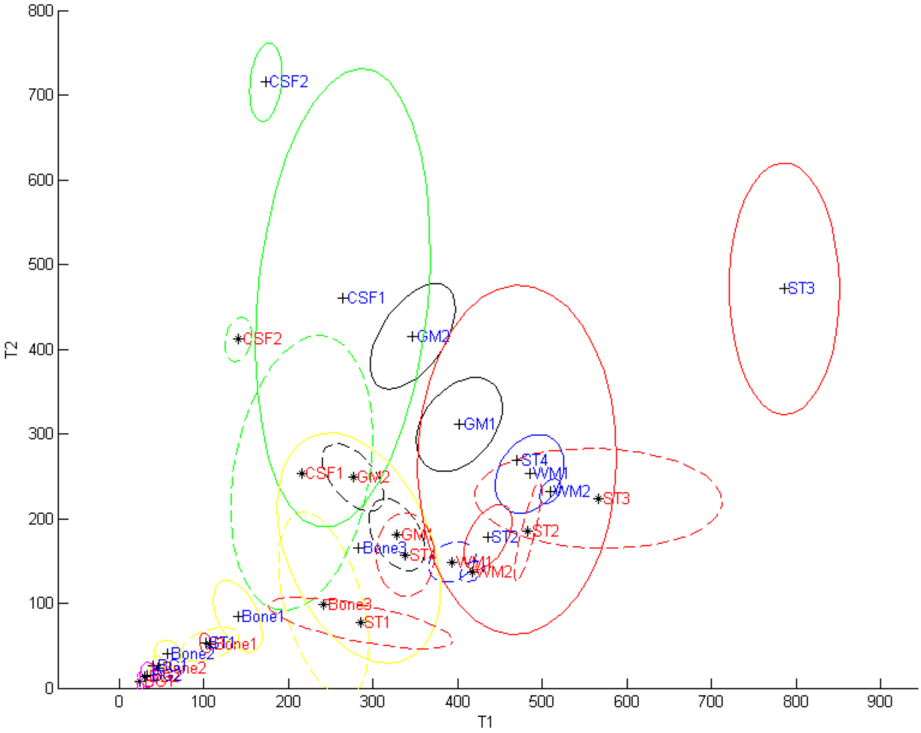


Figure D.9: Representation of the clusters for all the tissue classes done bu the version 3 of the modified method. The lines correspond to the contour of the Gaussian cut at FWHM, and weighted by the mixing coefficient. The contours of the clusters done by the original method are presented with dotted lines, the centers with the symbol *, and the text labels in red. The version 3 of the modified method presents the contours with solid lines, the center with the symbol +, and the text labels in blue. The depicted tissues comprise GM (black), WM (blue), CSF (green), ST (red), bone (yellow), and BG (magenta).

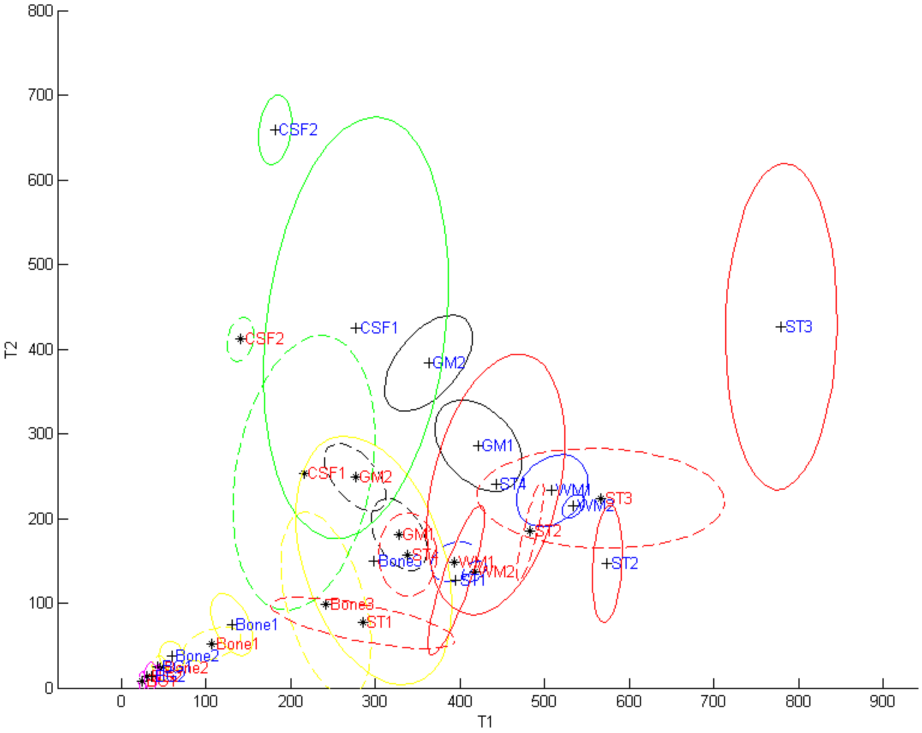


Figure D.10: Representation of the clusters for all the tissue classes done bu the version 4 of the modified method. The lines correspond to the contour of the Gaussian cut at FWHM, and weighted by the mixing coefficient. The contours of the clusters done by the original method are presented with dotted lines, the centers with the symbol *, and the text labels in red. The version 4 of the modified method presents the contours with solid lines, the center with the symbol +, and the text labels in blue. The depicted tissues comprise GM (black), WM (blue), CSF (green), ST (red), bone (yellow), and BG (magenta).

D.3 Dice coefficient for BrainWeb phantoms.

Table D.1: Result of the validation in terms of the Dice coefficient and the log-likelihood value. Six different methods are used for the segmentation of a BrainWeb phantom with **noise=0%**, RF=20% and 1mm isotropic resolution. The processing of the tissue probability maps have been done with Majority Voting and with several thresholds.

		Dice_{GM}	Dice_{WM}	Dice_{CSF}	$\text{Dice}_{nobrain}$	Dice_{T}	*loglikelihood*
Original method (T1)	Majority Voting	0.9368	0.9553	0.8019	0.9862	0.9673	$-8.6349\ 10^6$
	Threshold (>0.7)	0.9128	0.9675	0.7423	0.9862	0.9646	$-8.6349\ 10^6$
	Threshold (>0.9)	0.8080	0.9246	0.5939	0.9792	0.9426	$-8.6349\ 10^6$
Original method (T1+T2)	Majority Voting	0.8057	0.8853	0.5084	0.9844	0.9293	$-2.0871\ 10^6$
	Threshold (>0.7)	0.7895	0.8381	0.5075	0.9841	0.9270	$-2.0871\ 10^6$
	Threshold (>0.9)	0.7612	0.7673	0.4991	0.9826	0.9230	$-2.0871\ 10^6$
Modified method (ver1)	Majority Voting	–	–	–	–	–	–
	Threshold (>0.7)	–	–	–	–	–	–
	Threshold (>0.9)	–	–	–	–	–	–
Modified method (ver2)	Majority Voting	–	–	–	–	–	–
	Threshold (>0.7)	–	–	–	–	–	–
	Threshold (>0.9)	–	–	–	–	–	–
Modified method (ver3)	Majority Voting	0.8153	0.8519	0.7392	0.9871	0.9400	$-2.5799\ 10^6$
	Threshold (>0.7)	0.7305	0.8292	0.7909	0.9822	0.9313	$-2.5799\ 10^6$
	Threshold (>0.9)	0.5633	0.7786	0.7941	0.9720	0.9117	$-2.5799\ 10^6$
Modified method (ver4)	Majority Voting	0.7270	0.7534	0.6057	0.9805	0.9059	$-2.7565\ 10^6$
	Threshold (>0.7)	0.5900	0.7519	0.6549	0.9762	0.8969	$-2.7565\ 10^6$
	Threshold (>0.9)	0.3306	0.7141	0.6423	0.9603	0.8724	$-2.7565\ 10^6$

Table D.2: Result of the validation in terms of the Dice coefficient and the log-likelihood value. Six different methods are used for the segmentation of a BrainWeb phantom with **noise=3%**, RF=20% and 1mm isotropic resolution. The processing of the tissue probability maps have been done with Majority Voting and with several thresholds.

		Dice$_{GM}$	Dice$_{WM}$	Dice$_{CSF}$	Dice$_{nobrain}$	Dice$_{T}$	$loglikelihood$
Original method (T1)	Majority Voting	0.9292	0.9539	0.7930	0.9846	0.9646	$-1.1255\ 10^6$
	Threshold (>0.7)	0.9132	0.9273	0.7717	0.9837	0.9609	$-1.1255\ 10^6$
	Threshold (>0.9)	0.8384	0.8736	0.6575	0.9775	0.9439	$-1.1255\ 10^6$
Original method (T1+T2)	Majority Voting	0.9386	0.9555	0.8013	0.9868	0.9684	$-2.9534\ 10^6$
	Threshold (>0.7)	0.9178	0.9224	0.7937	0.9851	0.9628	$-2.9534\ 10^6$
	Threshold (>0.9)	0.8337	0.8578	0.7486	0.9742	0.9412	$-2.9534\ 10^6$
Modified method (ver1)	Majority Voting	0.9328	0.9452	0.8234	0.9869	0.9675	$-2.9795\ 10^6$
	Threshold (>0.7)	0.9116	0.9315	0.7962	0.9856	0.9630	$-2.9795\ 10^6$
	Threshold (>0.9)	0.8208	0.8810	0.7011	0.9730	0.9390	$-2.9795\ 10^6$
Modified method (ver2)	Majority Voting	0.9186	0.9312	0.7916	0.9849	0.9608	$-3.0085\ 10^6$
	Threshold (>0.7)	0.9106	0.9085	0.7612	0.9823	0.9566	$-3.0085\ 10^6$
	Threshold (>0.9)	0.8306	0.8520	0.6659	0.9745	0.9381	$-3.0085\ 10^6$
Modified method (ver3)	Majority Voting	0.9408	0.9557	0.8106	0.9872	0.9693	$-2.9555\ 10^6$
	Threshold (>0.7)	0.9217	0.9224	0.8009	0.9860	0.9642	$-2.9555\ 10^6$
	Threshold (>0.9)	0.8378	0.8576	0.7460	0.9751	0.9421	$-2.9555\ 10^6$
Modified method (ver4)	Majority Voting	0.9368	0.9560	0.7930	0.9872	0.9681	$-2.9521\ 10^6$
	Threshold (>0.7)	0.9157	0.9285	0.7916	0.9863	0.9641	$-2.9521\ 10^6$
	Threshold (>0.9)	0.8342	0.8693	0.7666	0.9774	0.9458	$-2.9521\ 10^6$

Table D.3: Result of the validation in terms of the Dice coefficient and the log-likelihood value. Six different methods are used for the segmentation of a BrainWeb phantom with **noise=9%**, RF=20% and 1mm isotropic resolution. The processing of the tissue probability maps have been done with Majority Voting and with several thresholds.

		Dice$_{GM}$	Dice$_{WM}$	Dice$_{CSF}$	Dice$_{nobrain}$	Dice$_{T}$	*loglikelihood*
Original method (T1)	Majority Voting	0.8547	0.8744	0.7489	0.9851	0.9459	$-1.2722\ 10^6$
	Threshold (>0.7)	0.8172	0.8537	0.7054	0.9847	0.9418	$-1.2722\ 10^6$
	Threshold (>0.9)	0.7010	0.7952	0.5801	0.9776	0.9246	$-1.2722\ 10^6$
Original method (T1+T2)	Majority Voting	0.8764	0.8906	0.7925	0.9848	0.9517	$-3.3943\ 10^6$
	Threshold (>0.7)	0.8448	0.8706	0.7926	0.9843	0.9488	$-3.3943\ 10^6$
	Threshold (>0.9)	0.7447	0.8197	0.7350	0.9756	0.9317	$-3.3943\ 10^6$
Modified method (ver1)	Majority Voting	0.8740	0.8888	0.7830	0.9834	0.9493	$-3.4010\ 10^6$
	Threshold (>0.7)	0.8386	0.8690	0.7896	0.9824	0.9460	$-3.4010\ 10^6$
	Threshold (>0.9)	0.7322	0.8174	0.7433	0.9711	0.9265	$-3.4010\ 10^6$
Modified method (ver2)	Majority Voting	0.8740	0.8891	0.7831	0.9836	0.9496	$-3.4009\ 10^6$
	Threshold (>0.7)	0.8388	0.8692	0.7901	0.9827	0.9463	$-3.4009\ 10^6$
	Threshold (>0.9)	0.7327	0.8175	0.7472	0.9715	0.9271	$-3.4009\ 10^6$
Modified method (ver3)	Majority Voting	0.8763	0.8905	0.7921	0.9848	0.9516	$-3.3943\ 10^6$
	Threshold (>0.7)	0.8446	0.8706	0.7923	0.9843	0.9488	$-3.3943\ 10^6$
	Threshold (>0.9)	0.7444	0.8196	0.7348	0.9756	0.9318	$-3.3943\ 10^6$
Modified method (ver4)	Majority Voting	0.8763	0.8906	0.7914	0.9847	0.9515	$-3.3944\ 10^6$
	Threshold (>0.7)	0.8448	0.8706	0.7916	0.9842	0.9487	$-3.3944\ 10^6$
	Threshold (>0.9)	0.7448	0.8197	0.7348	0.9755	0.9317	$-3.3944\ 10^6$

D.4 Segmentation of the BrainWeb phantoms.

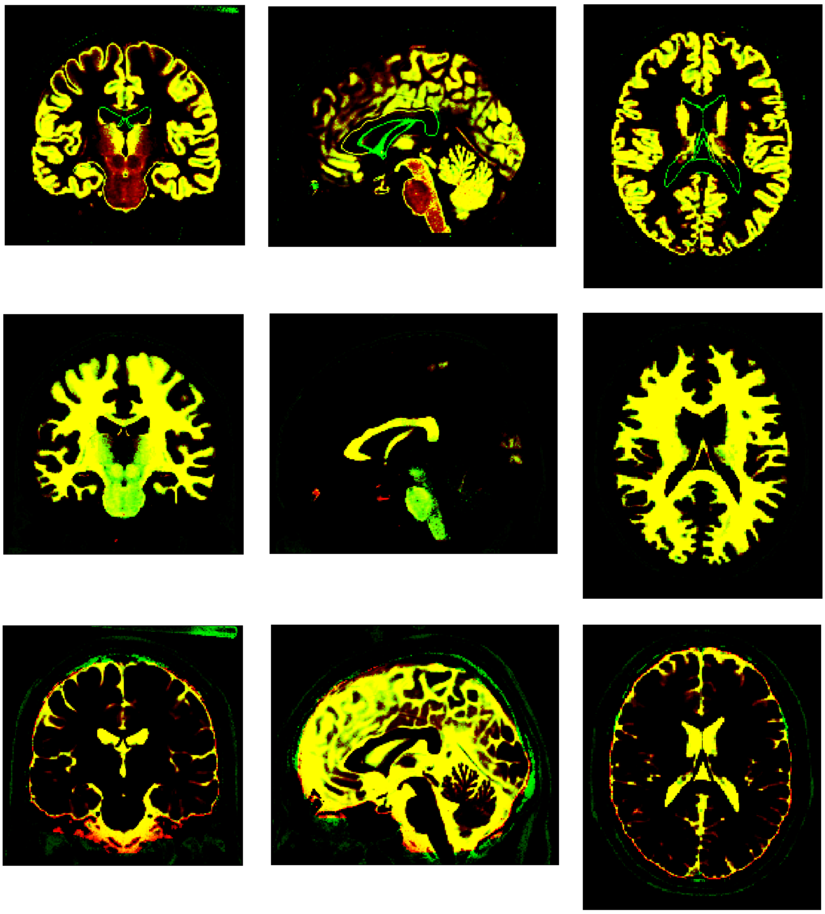


Figure D.11: Representation of the quality of the **original** segmentation method for GM, WM and CSF, **before Majority Voting**. The voxels are presented in yellow (TP), black (TN), green (FP), and red (FN) depending if their classification result in comparison with the ground truth. The segmented MRI brain volumes correspond to the BrainWeb phantoms of T_1 and T_2 modalities with a 1mm resolution, 3% level of noise and 20% of intensity non-uniformity level.

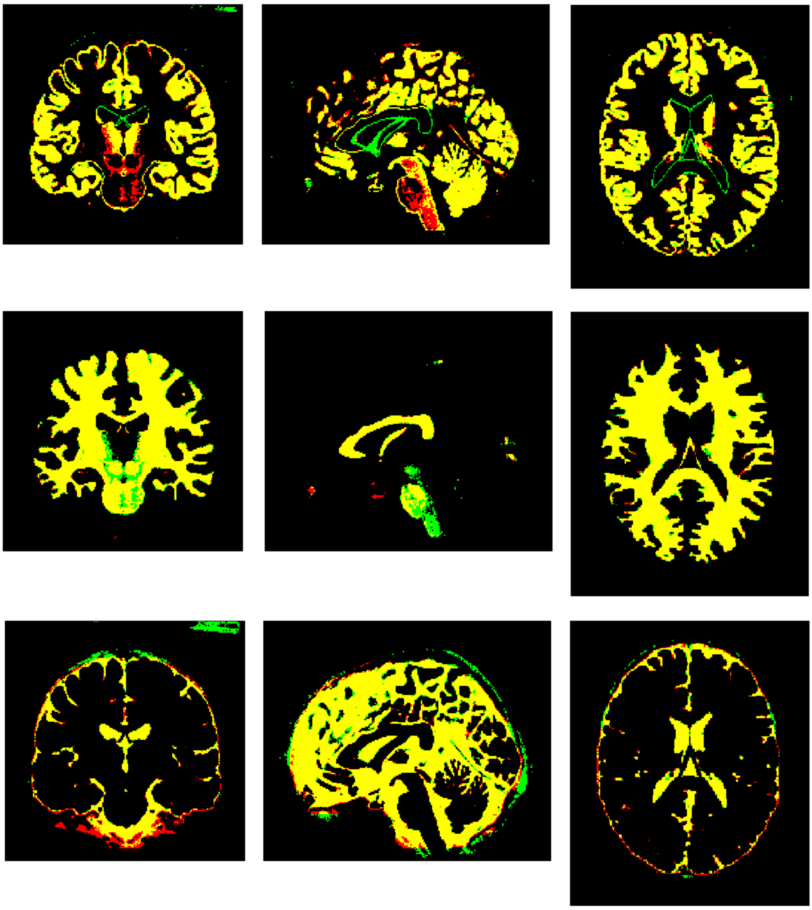


Figure D.12: Representation of the quality of the **original** segmentation method for GM, WM and CSF, **after Majority Voting**. The voxels are presented in yellow (TP), black (TN), green (FP), and red (FN) depending if their classification result in comparison with the ground truth. The segmented MRI brain volumes correspond to the BrainWeb phantoms of T_1 and T_2 modalities with a 1mm resolution, 3% level of noise and 20% of intensity non-uniformity level.

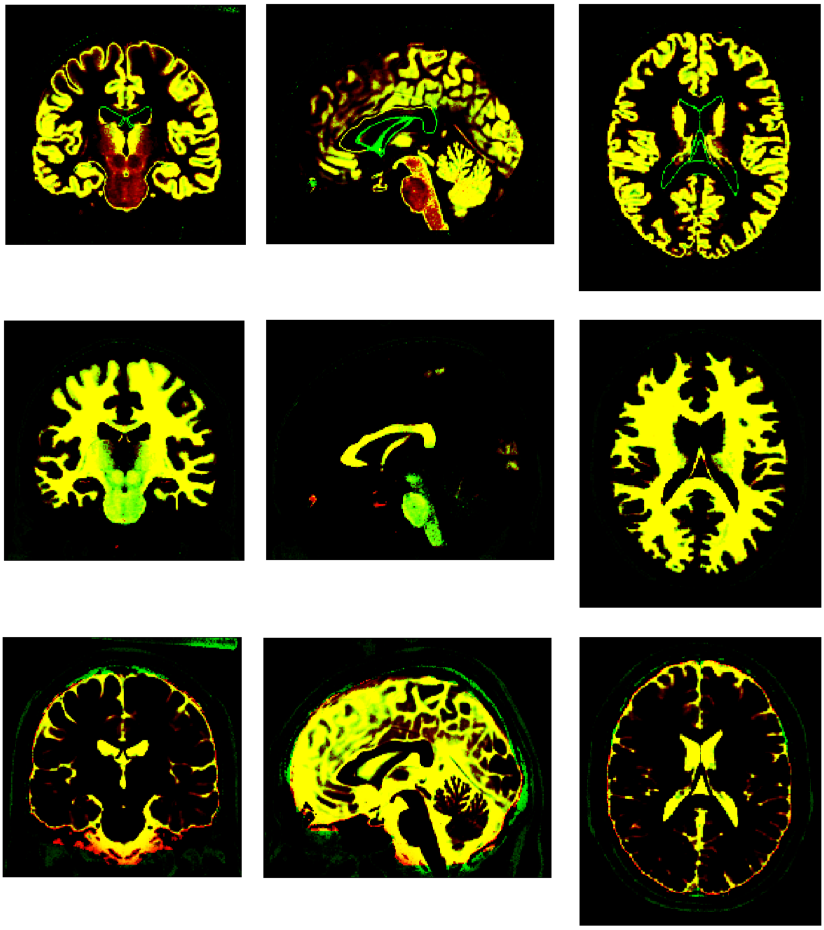


Figure D.13: Representation of the quality of the **modified version 3** segmentation method for GM, WM and CSF, **before Majority Voting**. The voxels are presented in yellow (TP), black (TN), green (FP), and red (FN) depending if their classification result in comparison with the ground truth. The segmented MRI brain volumes correspond to the BrainWeb phantoms of T_1 and T_2 modalities with a 1mm resolution, 3% level of noise and 20% of intensity non-uniformity level.

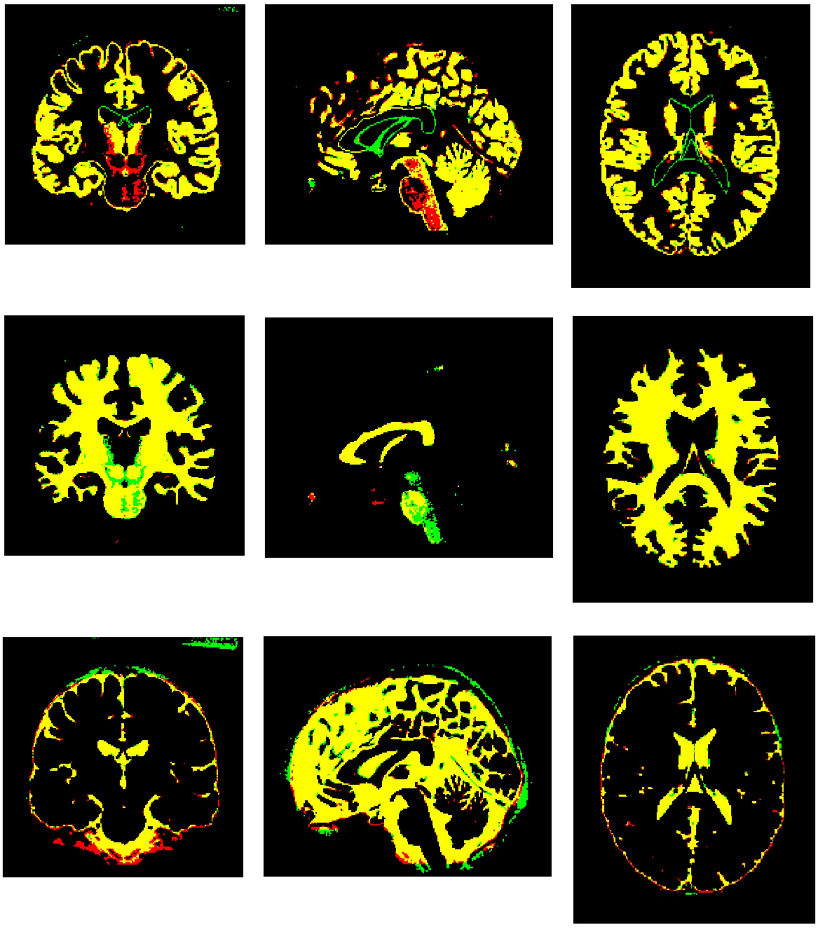


Figure D.14: Representation of the quality of the **modified version 3** segmentation method for GM, WM and CSF, **after Majority Voting**. The voxels are presented in yellow (TP), black (TN), green (FP), and red (FN) depending if their classification result in comparison with the ground truth. The segmented MRI brain volumes correspond to the BrainWeb phantoms of T_1 and T_2 modalities with a 1mm resolution, 3% level of noise and 20% of intensity non-uniformity level.

D.5 Atrophy.

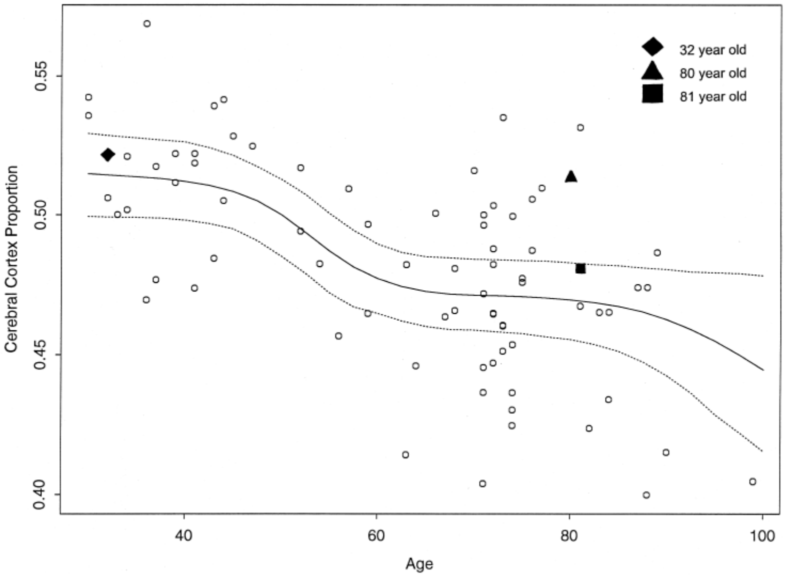
The brain functions decline after certain age in different manners, as the decrease of short-term memory, verbal ability, and intellectual performance, or the increase of the reaction time. This lost of capacity is due to the brain *atrophy*, i.e. loss of brain parenchyma and changes in the associated anatomical structures (e.g. decreased efficiency of neurotrasmitters in survival neurons). Taking into account that the variations of human neocortex volume are equally distributed among different people, it is possible to study the pattern of changes on average.

In the ageing process, it is detected a large decrease of cortex volume that goes together with a large decrease of the pial surface (external boundary of the cortex) and a small decrease of the neocortical thickness. This process contrasts with the known effects in the brain of some common diseases, like Acquired Immune Deficiency Syndrome (AIDS) or Alzheimer Disease (AD), where the neocortical thickness is the most affected [65] [77]. Thus, it can be isolated the age-related changes in the brain from diseases that affect the brain.

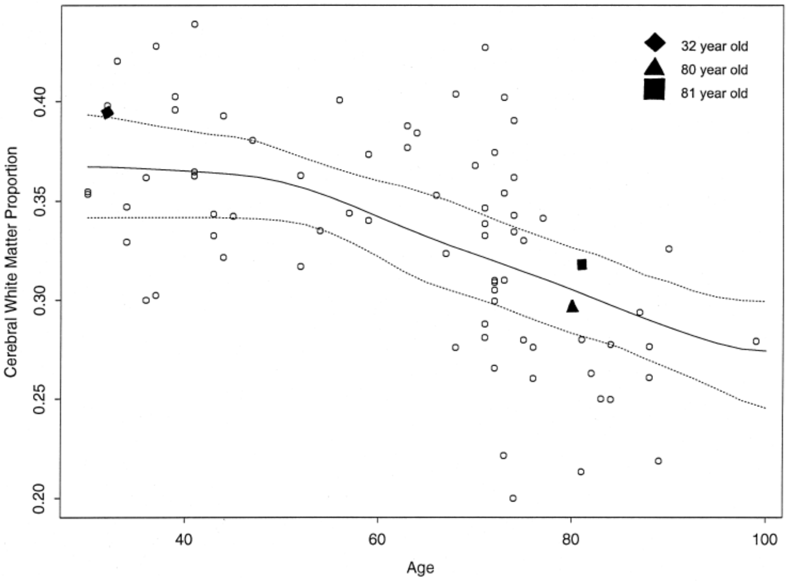
B. Pakkenberg et al. [62] did a study with 94 human brains from Danish dead people between 20 years and 90 years old. The study showed that the brain variation was mostly determined by the gender and age. Specifically for the age, taking into account the overall life span, it was observed a 12.3% decrease of cortex volume, 28% of white matter, but not significant variations were found in neocortical thickness or gray matter volume. In addition, the volume losses appeared together with an increase of CSF.

Another study of the brain atrophy was done by T. L. Jernigan et al. [33] with MRI from healthy volunteers aged from 30 to 99 years. The results showed that the hippocampus losses were significant, and the frontal lobes were affected by a decrease of cortical volume and an increase of white matter abnormalities. In addition, the decrease of white matter over the life range in the cerebral and cerebellar structures was bigger that the gray matter, with 14% in the cerebral cortex, 35% in the hippocampus, and 26% in the cerebral white matter.

The Figure D.15 depicts the results of this last study as the correlation between volume and age for different tissues. Each subfigure presents a different tissue: cerebral cortex, and cerebral white matter. It can be seen a big brain loss around the 55 years old. The graph shows a scarce number of samples around the mentioned age because it is usually hard to include an important number of volunteers in this range of years. This problem is also concerning the CIMBI projects, where the volunteers in range around 40 years old is limited.



(a) Estimated volume of cerebral cortex by age.



(b) Estimated volume of cerebral WM by age.

Figure D.15: Estimated volumes related to the age. The filled line corresponds to the smooth trend and the dashed lines to the variability. Three subjects of 32, 80 and 81 years are highlighted. [Courtesy of B. Pakkenberg et al. [62]]

Finally, C. R. G. Guttmann [29] evaluated brain tissue of 72 healthy volunteers with ages from 18 to 81 years using MRI. The proportion of white matter of brain older than 59 years was much lower, while the CSF fraction was bigger, and the gray matter did not suffer a significant variation. These results are similar to the ones presented in the previous studies and they are also consistent with neuropathologic reports in human beings.

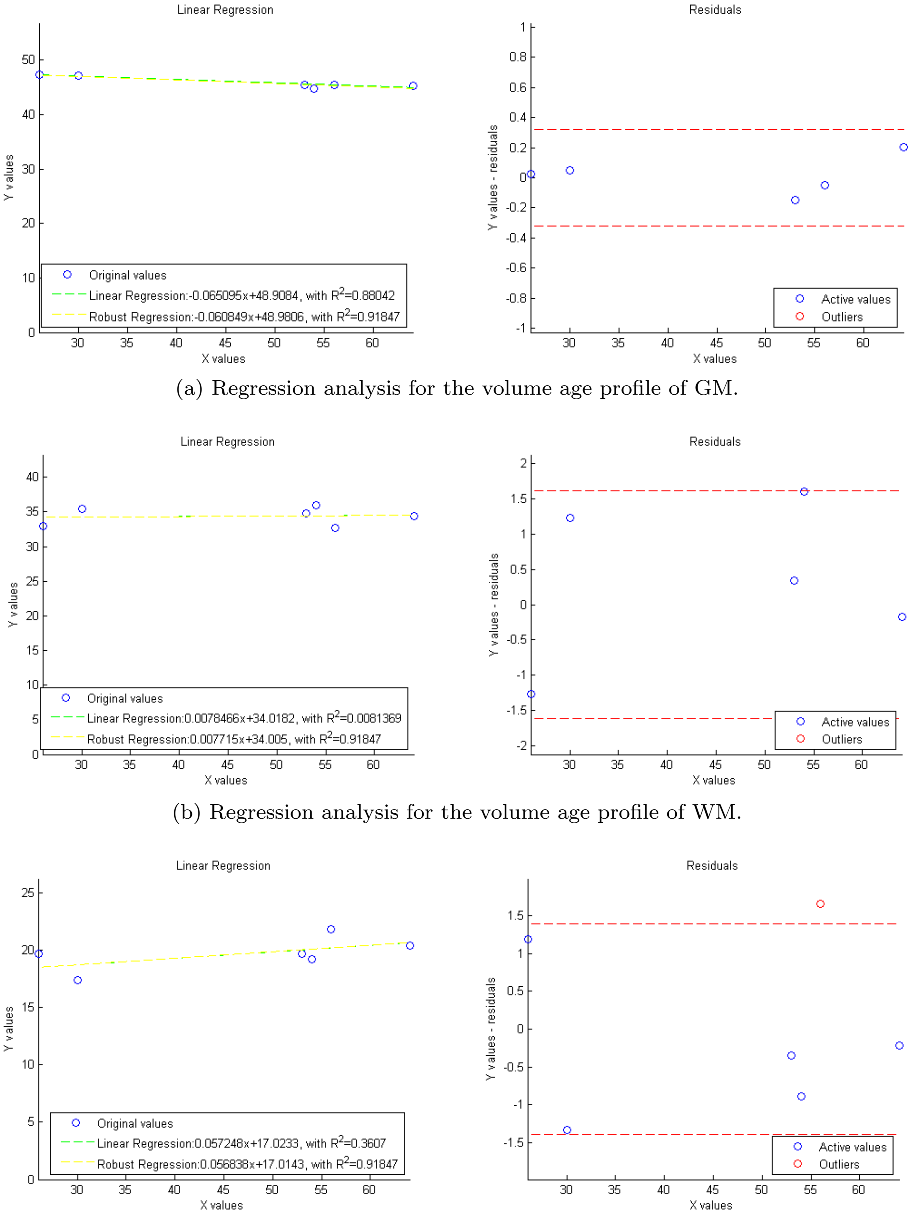
Table D.4: Atrophy of the brain related to the ageing, as a measure of the ICC volume variation of different tissues. Mean value and deviation are presented. [Data from C. R. G. Guttmann [29]]

Age(%)	**GM(%)**	**WM(%)**	**CSF(%)**	**Lesion(%)**
23.3 ± 7.6	48.7 ± 1.8	38.9 ± 2.8	7.1 ± 3.1	0.25 ± 0.13
45.6 ± 2.7	46.0 ± 1.4	37.5 ± 1.8	9.9 ± 2.0	0.40 ± 0.17
55.0 ± 2.3	44.6 ± 1.9	38.5 ± 2.2	10.6 ± 2.0	0.30 ± 0.13
66.0 ± 2.9	46.6 ± 2.3	35.0 ± 2.7	12.3 ± 2.9	0.30 ± 0.10
73.5 ± 3.0	47.2 ± 3.3	33.0 ± 4.1	13.4 ± 2.1	0.29 ± 0.11

The results of the study are presented in the Table D.4. It can be seen how GM volume slightly decreases and the WM suffers an important decrease. However, there is a in increase of GM after the 50-years that is not expected. Although, it is not explained in the article, sometimes the increase of GM in old patients is due to the missclassification of lesions as GM.

As a conclusion, it can be stated that **the brain ageing can be characterized by a subtle decrease of GM and WM volume, and a big increase of CSF**. In case that the segmentation shows a severe decrease in the neocortical thickness or gray matter volume, it could be assumed that the brain is affected by a disease that induces brain changes not connected to the aging itself. As the dataset of this thesis only includes healthy brains, it is not expected GM variations.

In the Figures D.16 and D.17, it is presented a simple linear regression analysis of the volume age profile of six subjects. Six brains from the CIMBI database have been segmented by the original and modified v.3 method into GM, WM and CSF. The volumes of each class are normalized by the ICC of each subject. The results show a small decrease of GM, an increase of CSF, and a constant value of WM. However, due to the small used dataset, it is not possible to infer any further conclusion.

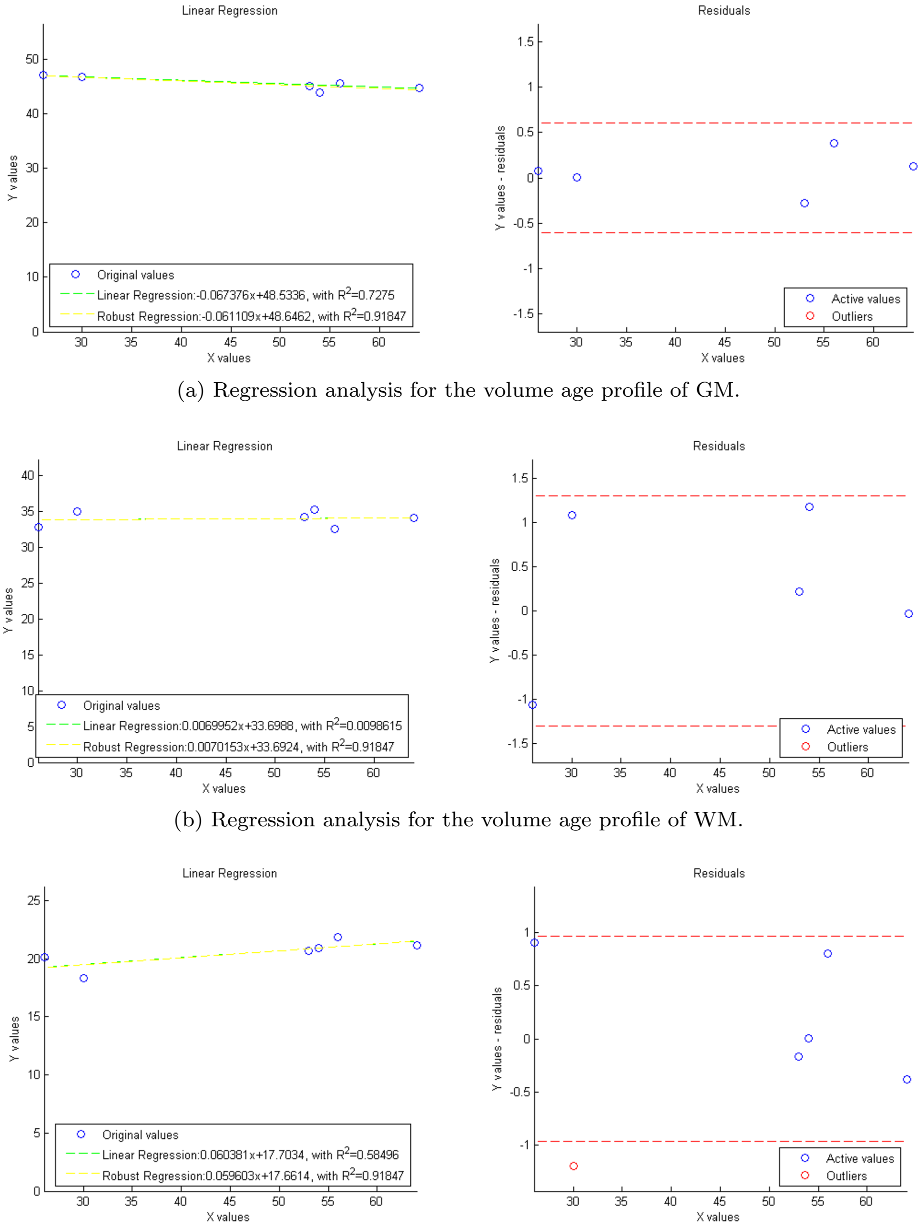


(a) Regression analysis for the volume age profile of GM.

(b) Regression analysis for the volume age profile of WM.

(c) Regression analysis for the volume age profile of CSF.

Figure D.16: Linear regression analysis of the volume age profile of six subjects. Six brains from the CIMBI database have been segmented by the original method into GM, WM and CSF.



(a) Regression analysis for the volume age profile of GM.

(b) Regression analysis for the volume age profile of WM.

(c) Regression analysis for the volume age profile of CSF.

Figure D.17: Linear regression analysis of the volume age profile of six subjects. Six brains from the CIMBI database have been segmented by the modified v.3 method into GM, WM and CSF.

Appendix E

Volumes

This chapter presents different brain volumes for several slices in the three planes (coronal, sagittal, transversal).

- *MRI Data.* Co-registered MR volumes for T_1 and T_2 modality from the scans of the subject `f4395`.
- *Tissue Probability Maps as Templates.* In includes 6 TPM's that are used as prior templates. They have dimensions $121 \times 145 \times 121$ and $1.5mm$ of spatial resolution. They are done from 471 brains by Cynthia Jongen of the Imaging Sciences Institute at. Utrecht, NL. Each volume corresponds to a different tissues class, namely GM, WM, CSF, bone, ST and BG. In addition, two additional volumes are presented, which correspond to the overlapping of the different tissues.
- *Segmentation of volumes from subject f4395.* In includes 6 probability maps generated from the version 3 of the modified method. The segmentation has been done with default parameters for the data of the subject `f4395`. In addition, two additional volumes are presented, which correspond to the overlapping of the different tissues.
- *Segmentation of BrainWeb phantoms.* Performance results of the original method and version 3 of the modified method. The slices are presented before and after Majority Voting.

E.1 MRI Data

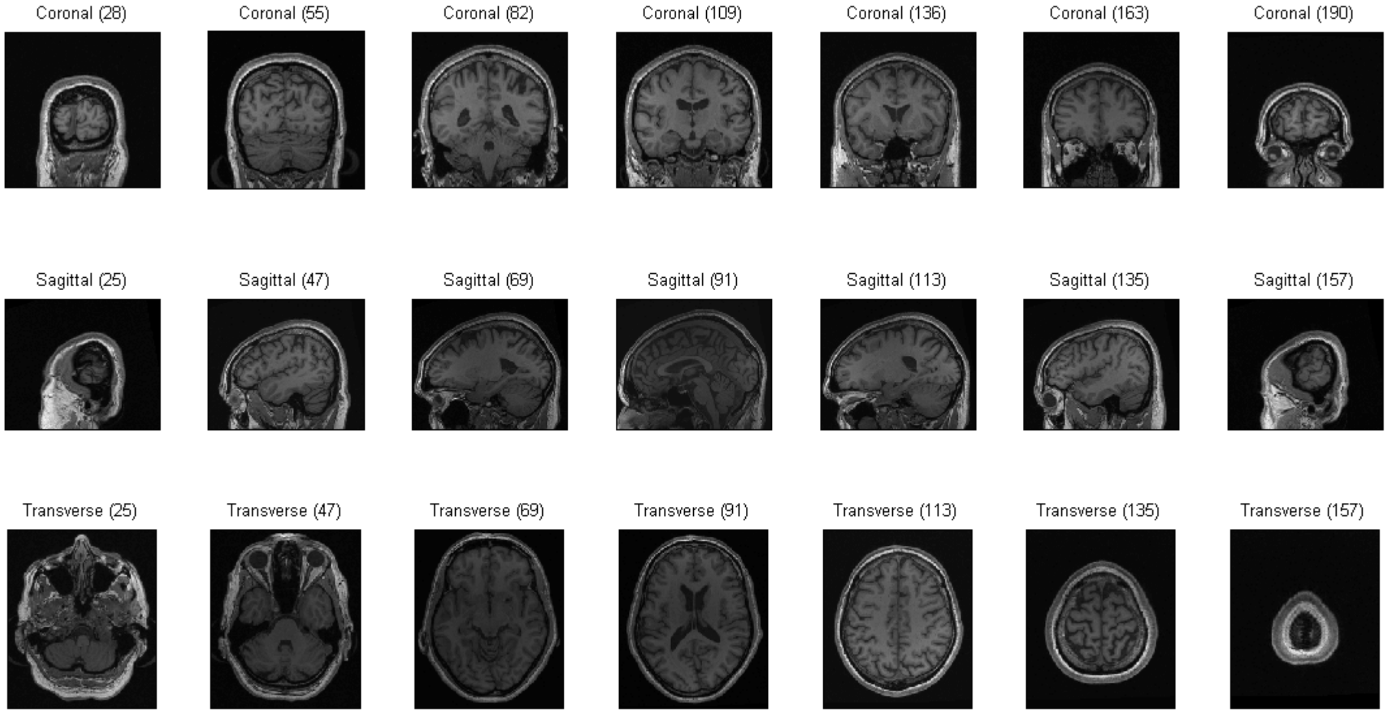


Figure E.1: T1 MRI volume of subject f4395.

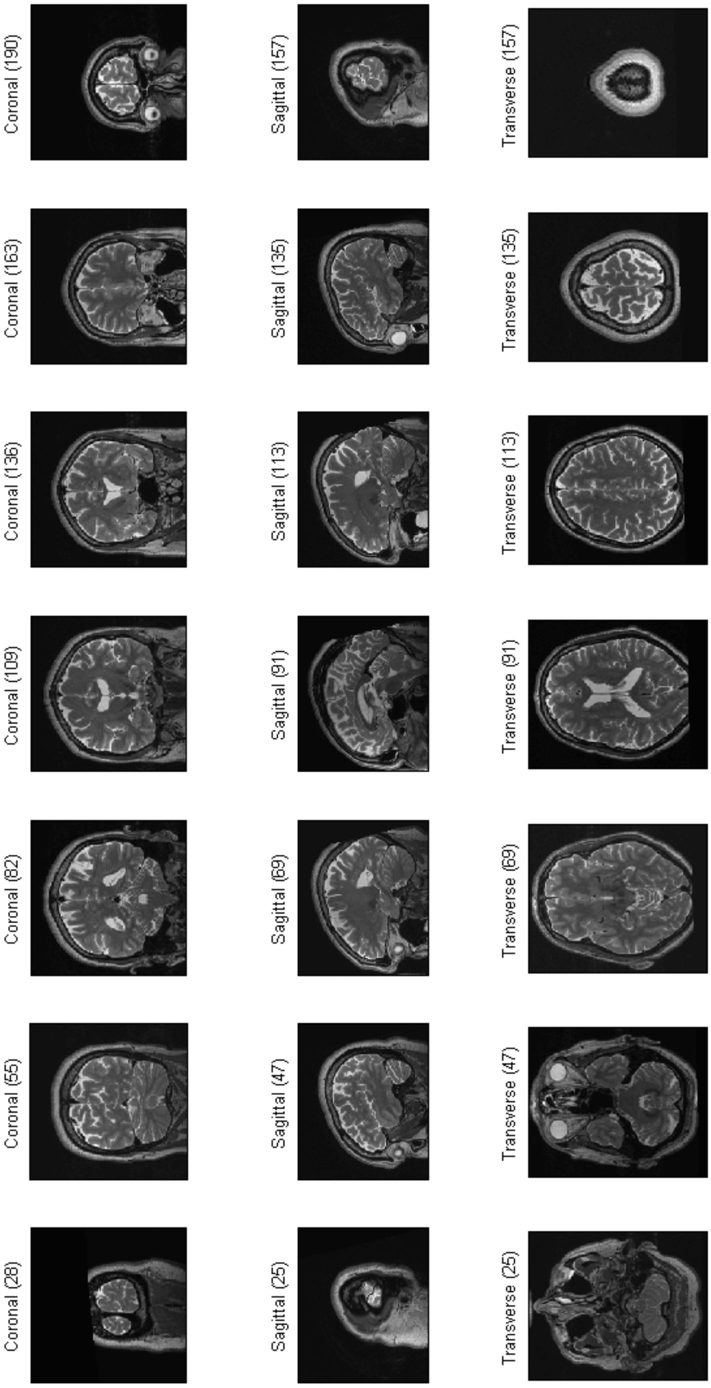


Figure E.2: T2 MRI volume of subject f4395.

E.2 Tissue Probability Maps for Prior Templates

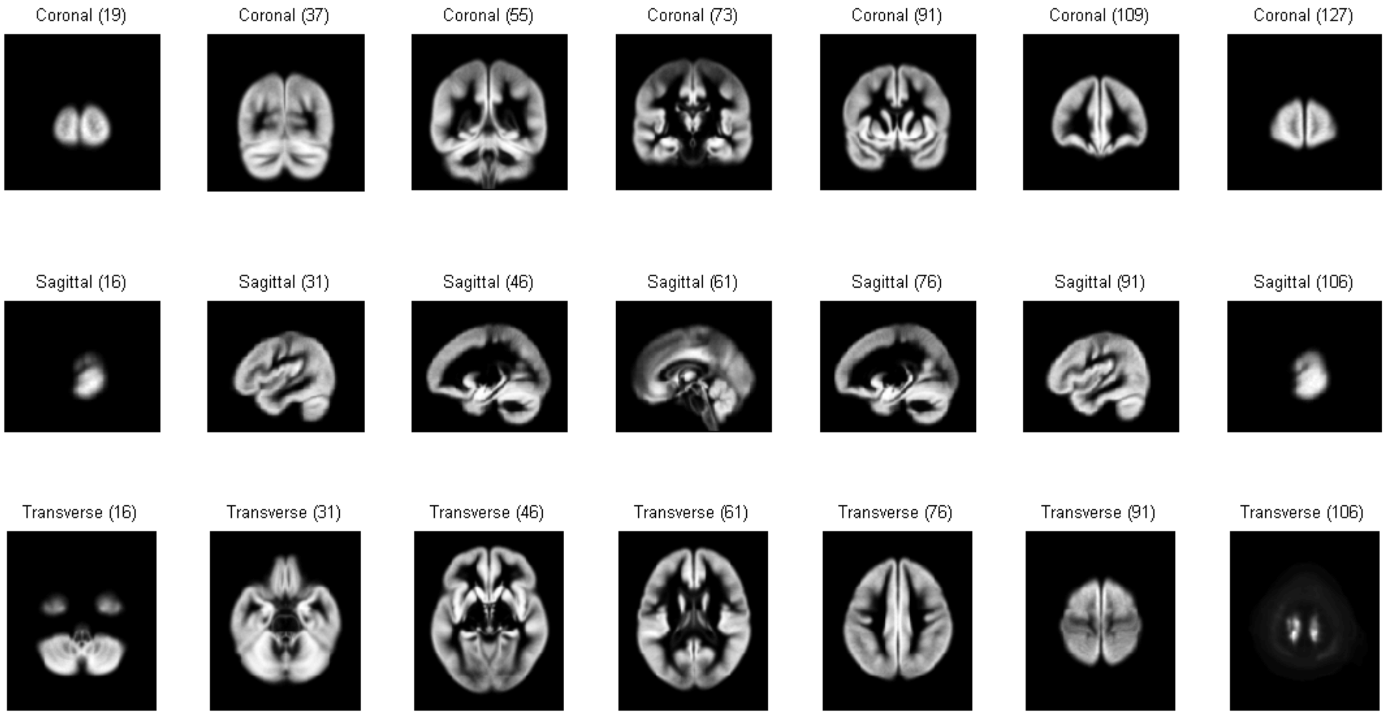


Figure E.3: Tissue Probability Map for Prior Templates - Grey Matter.

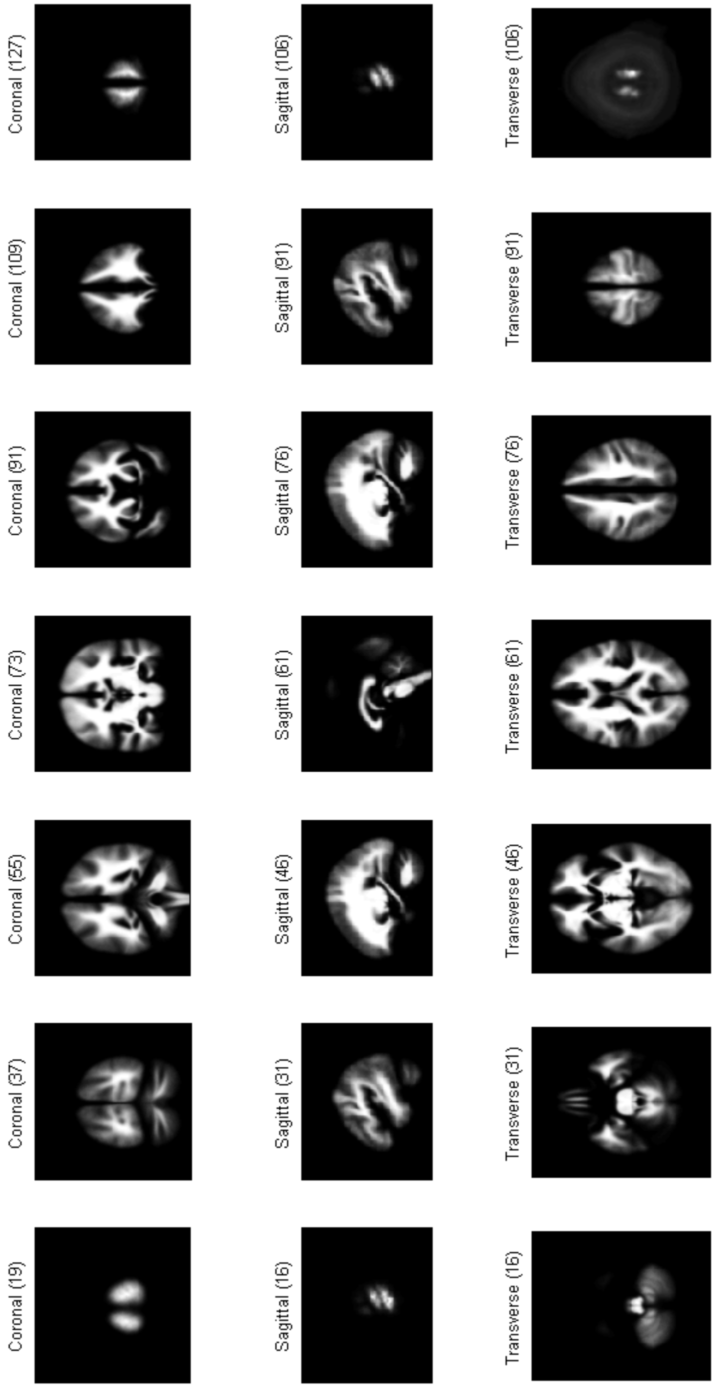


Figure E.4: Tissue Probability Map for Prior Templates - White Matter.

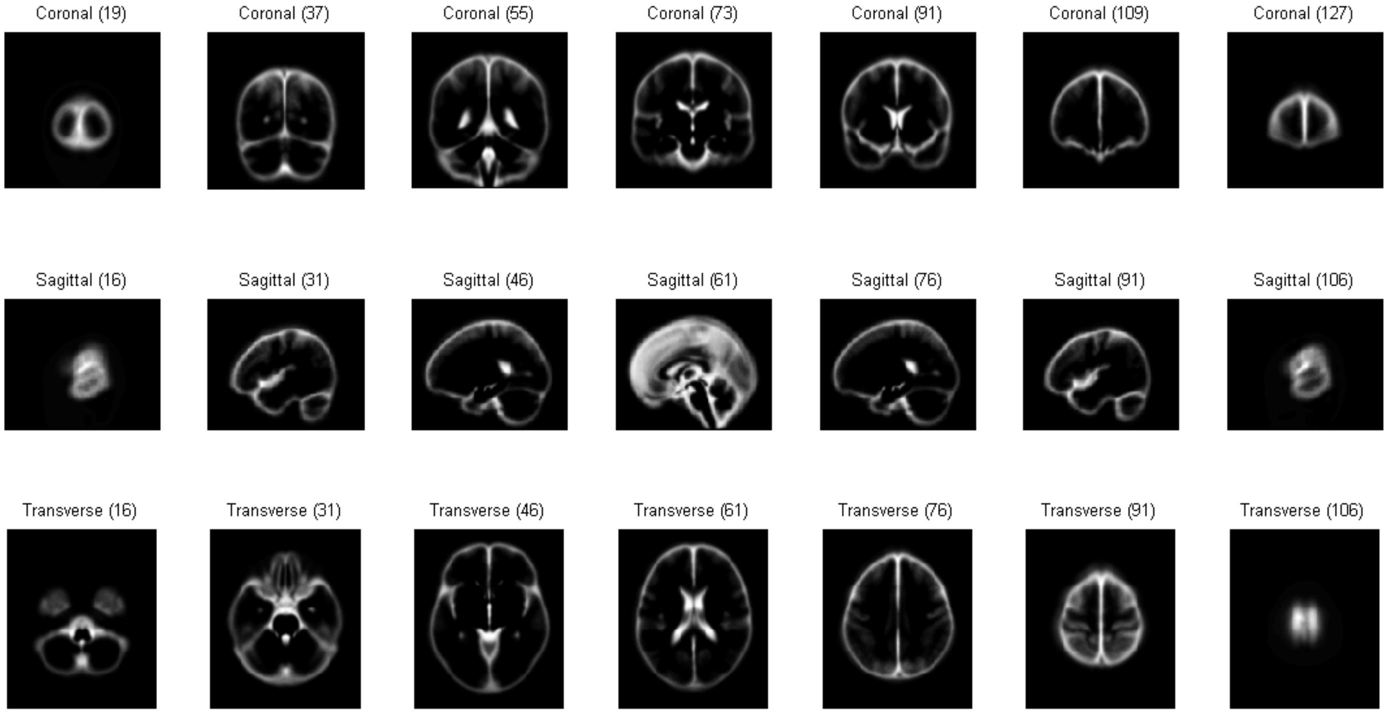


Figure E.5: Tissue Probability Map for Prior Templates - CerebroSpinal Fluid.

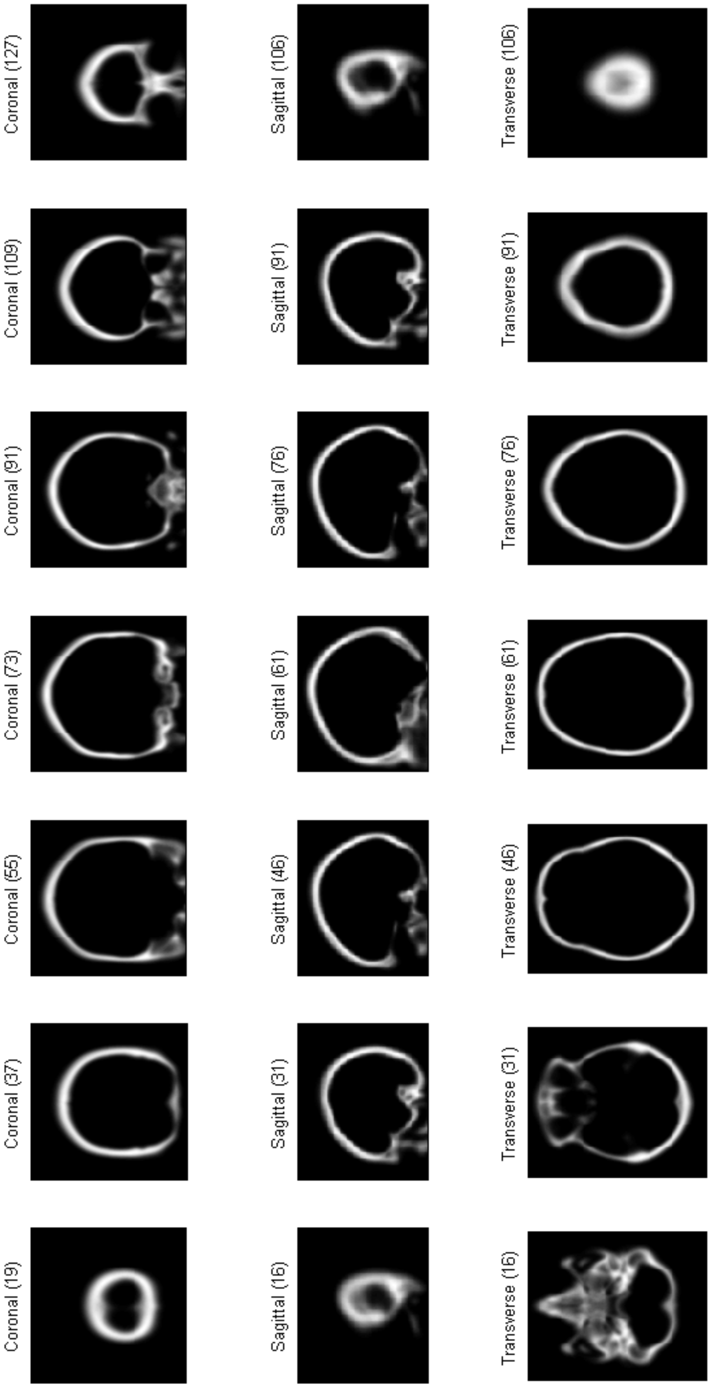


Figure E.6: Tissue Probability Map for Prior Templates - Bone.

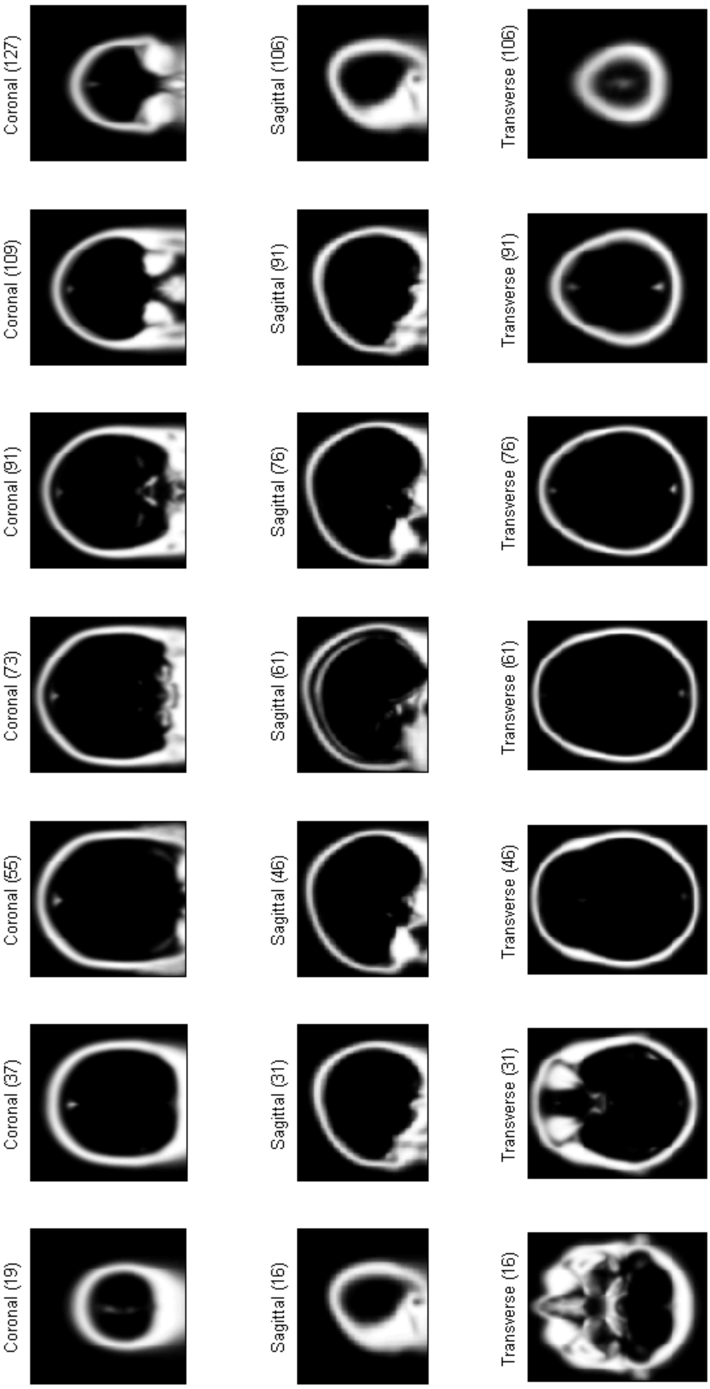


Figure E.7: Tissue Probability Map for Prior Templates - Soft Tissue.

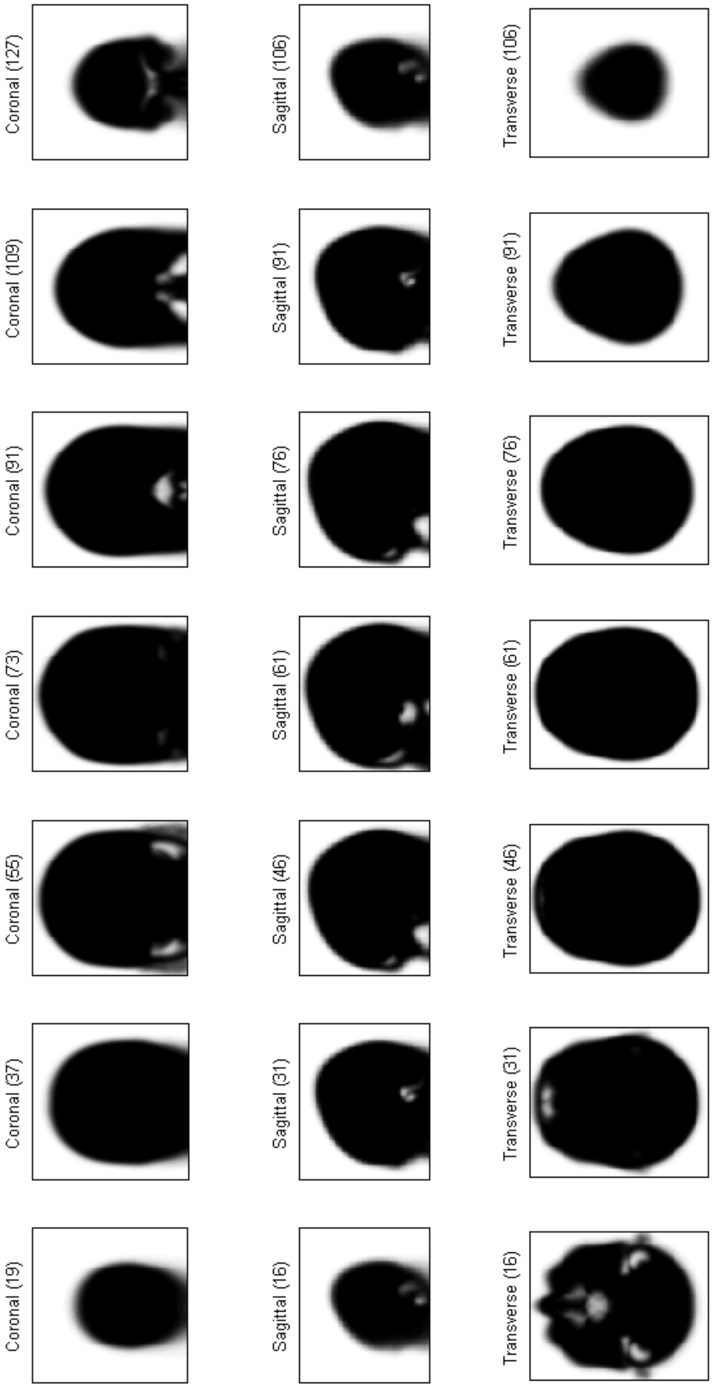


Figure E.8: Tissue Probability Map for Prior Templates - Background.

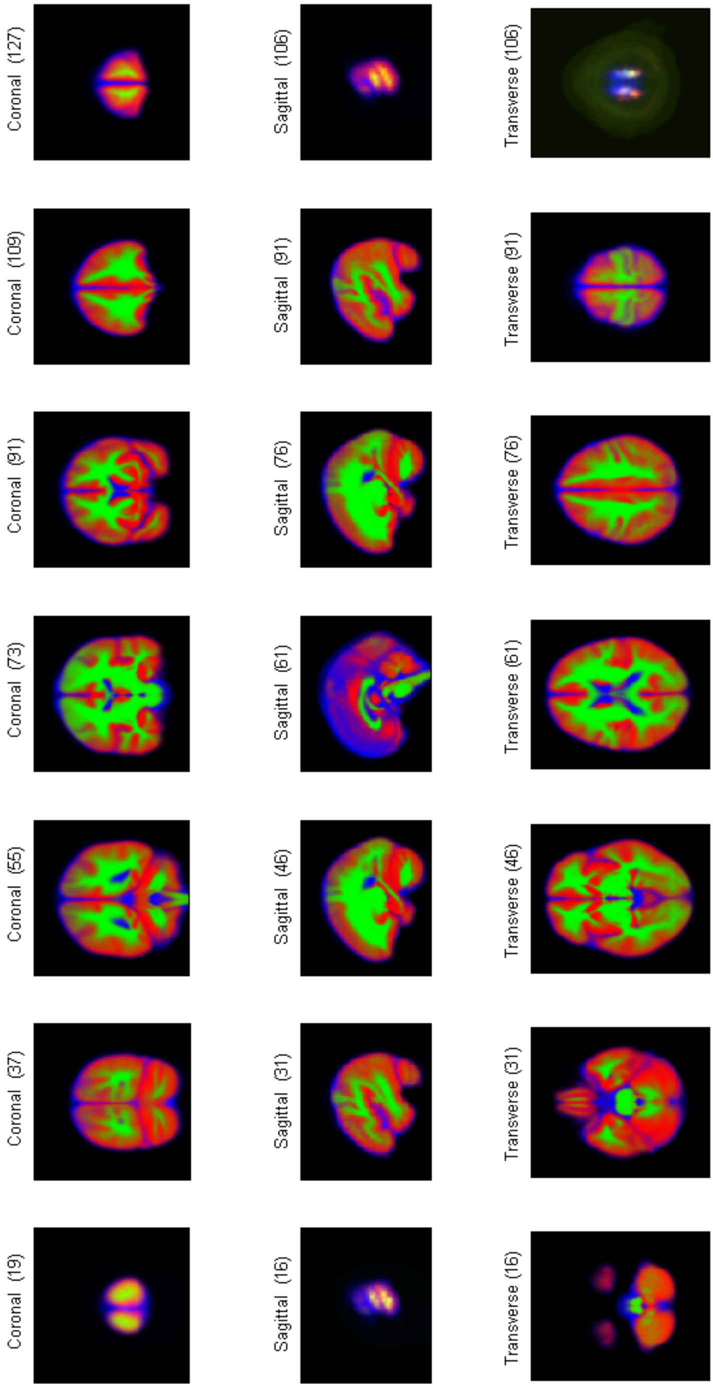


Figure E.9: Template Tissue Probability Maps - GM/WM/CSF overlap. GM, WM and CSF are presented in red, green and blue, respectively.

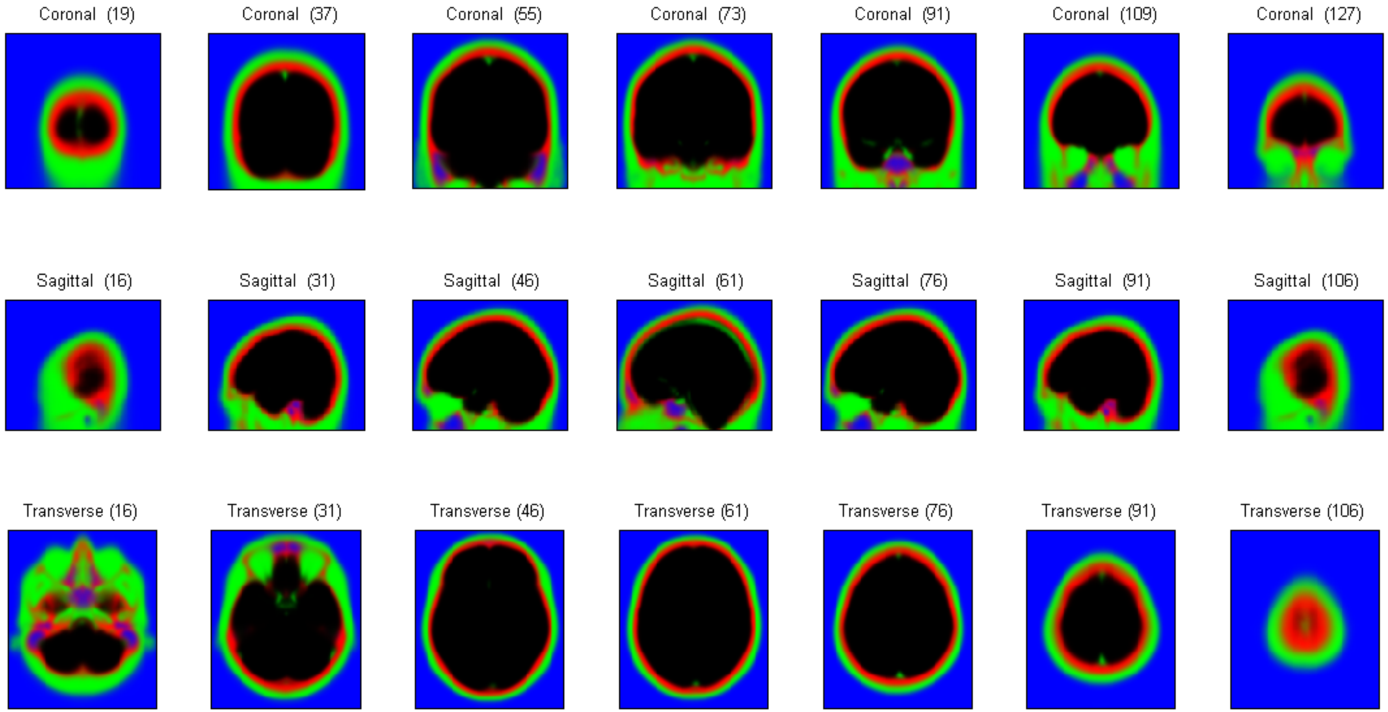


Figure E.10: Template Tissue Probability Maps - Bone/ST/BG overlap. ST, Bone and BG are presented in red, green and blue, respectively.

E.3 Segmentation of volumes from subject f4395.

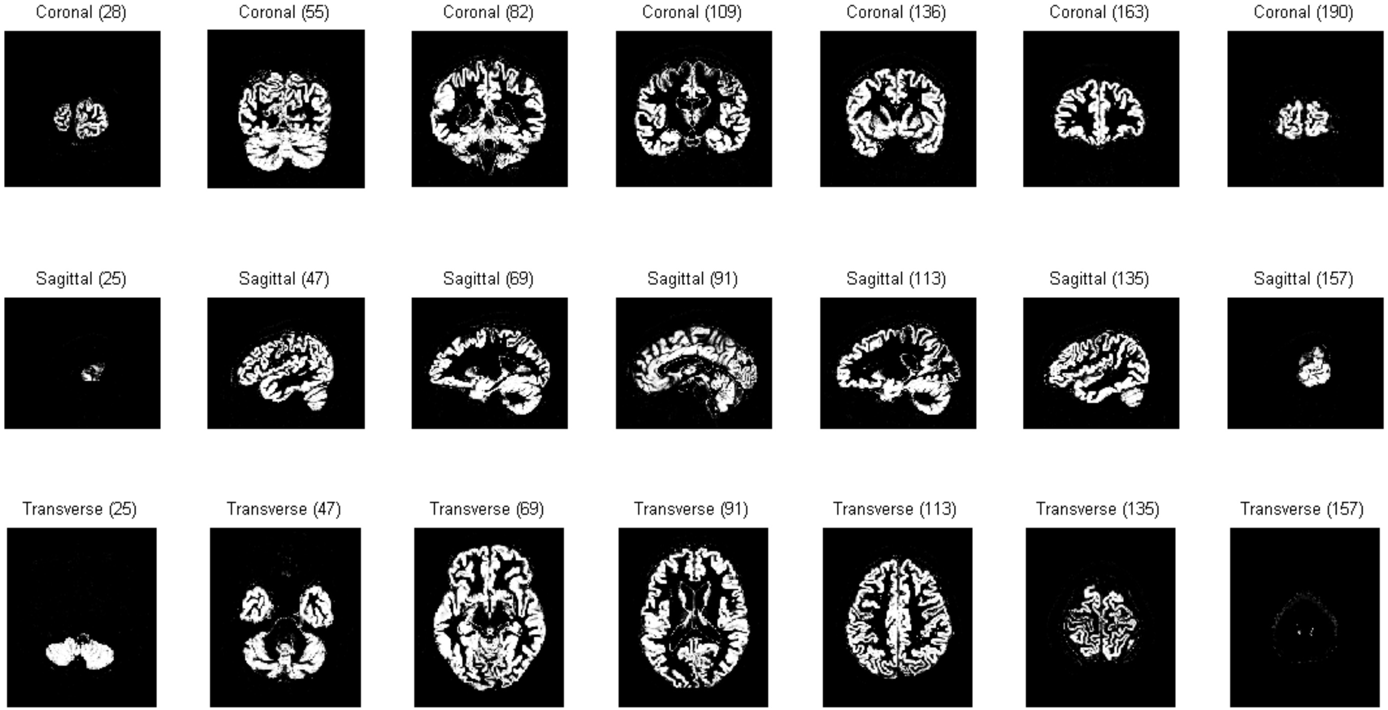


Figure E.11: Segmentation of volumes from subject f4395 - Grey Matter.

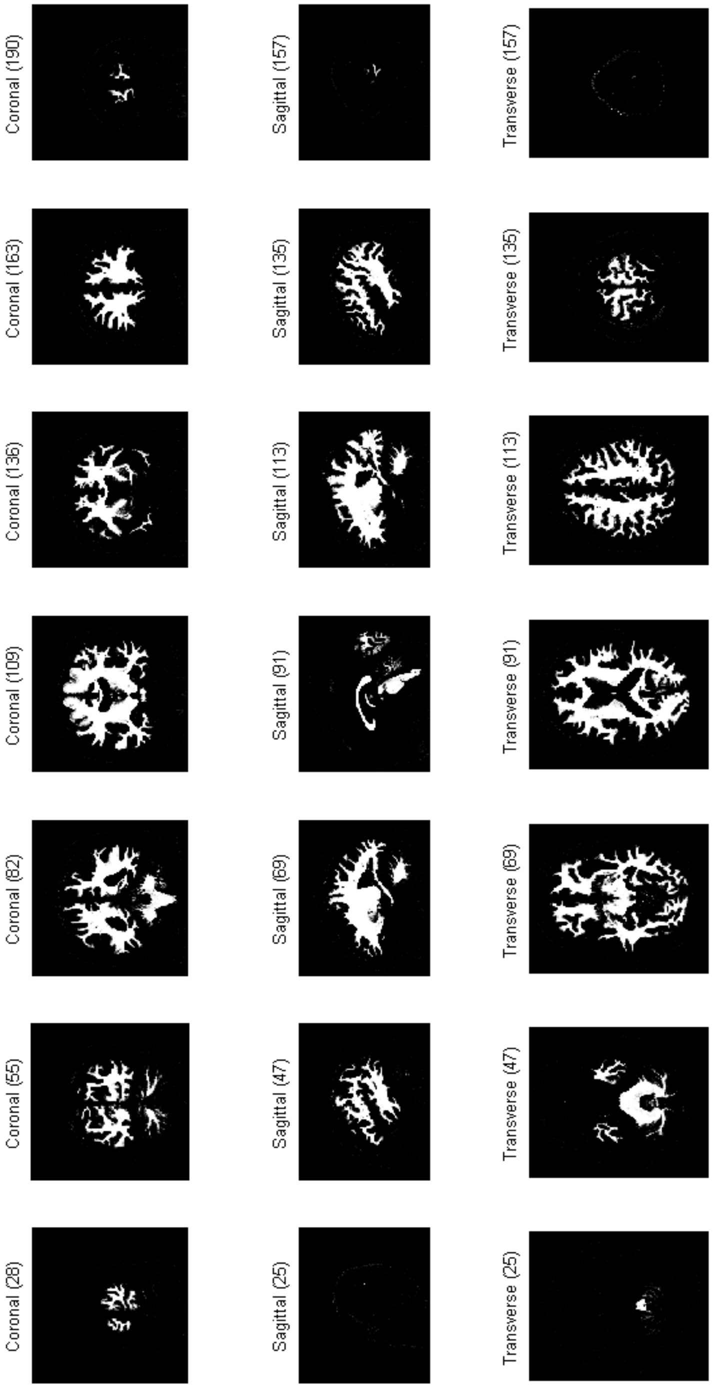


Figure E.12: Segmentation of volumes from subject f4395 - White Matter.

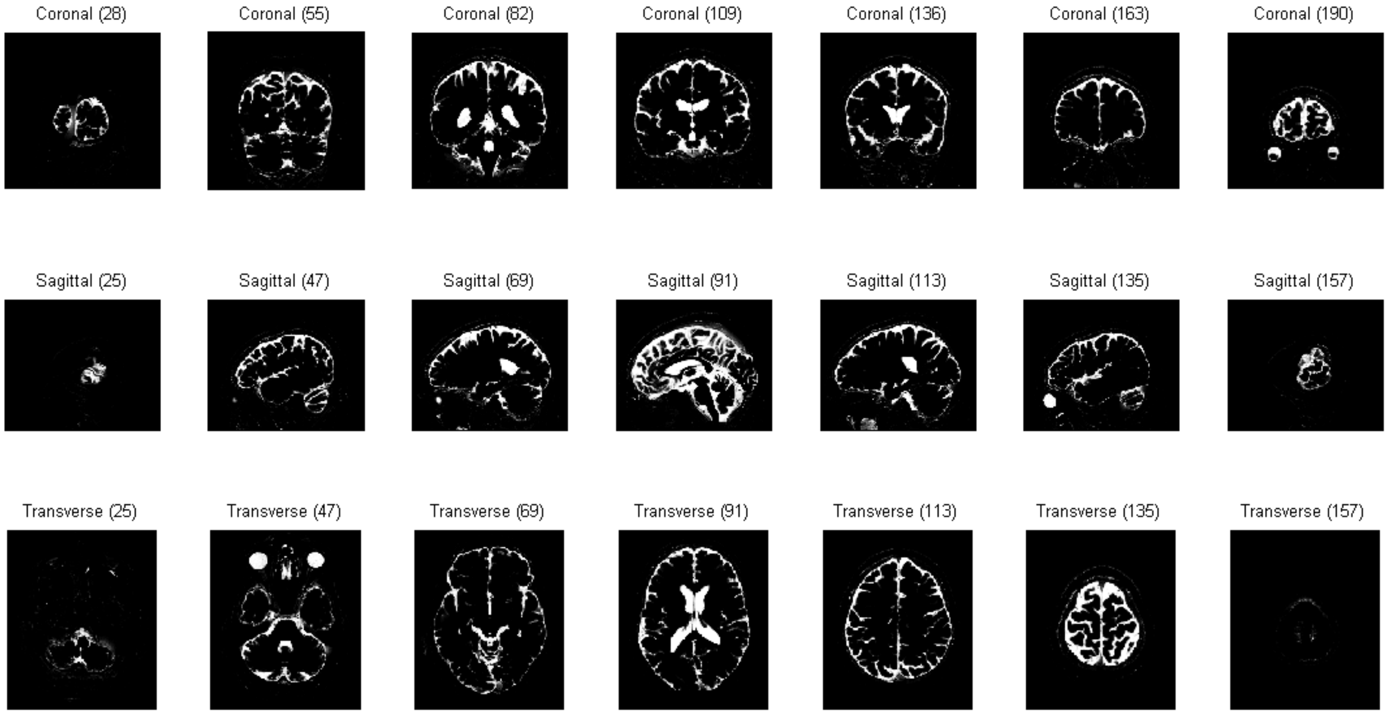


Figure E.13: Segmentation of volumes from subject f4395 - CerebroSpinal Fluid.

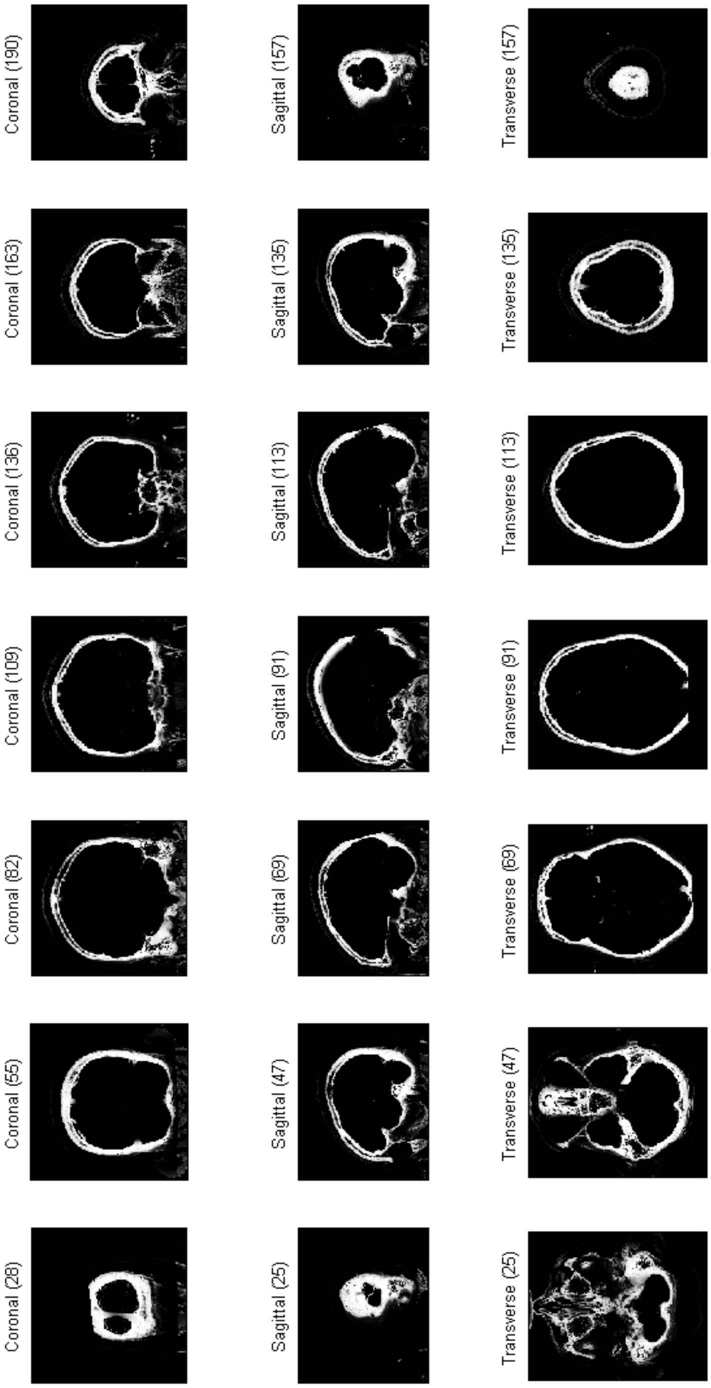


Figure E.14: Segmentation of volumes from subject f4395 - Bone.

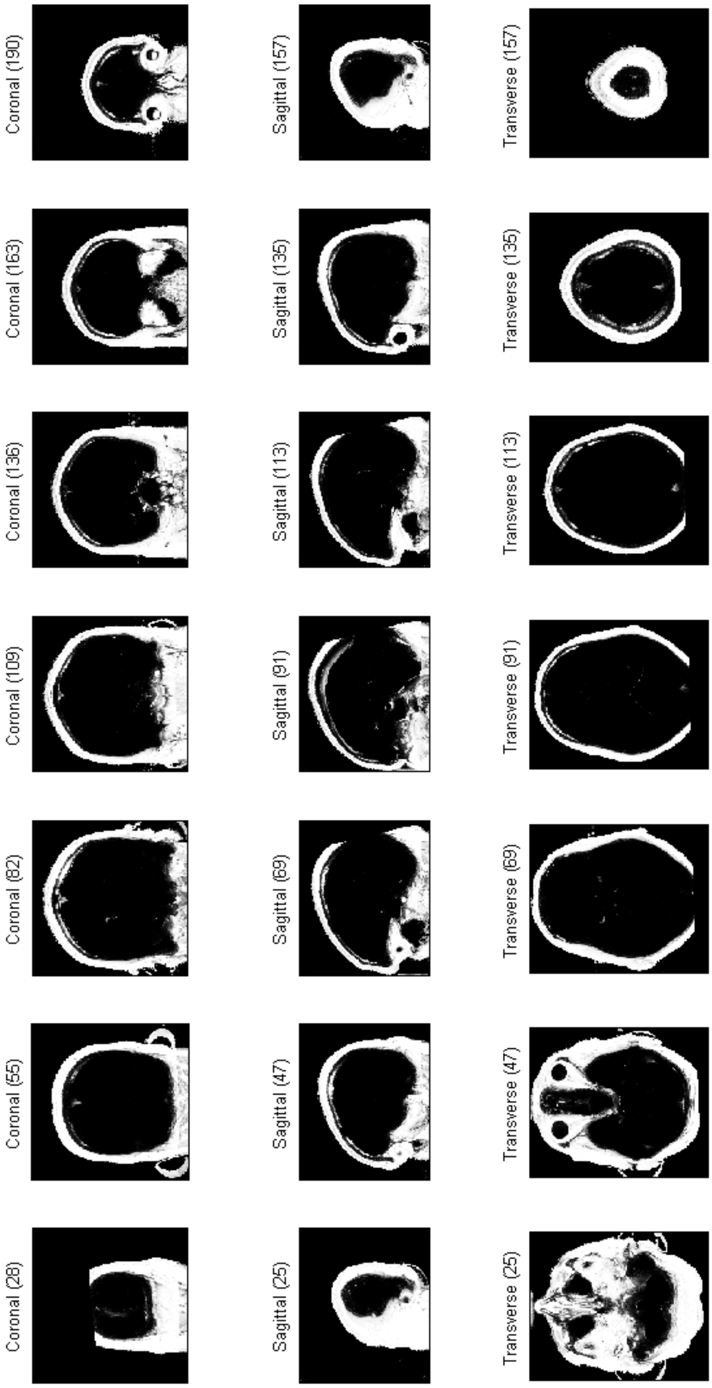


Figure E.15: Segmentation of volumes from subject f4395 - Soft Tissue.

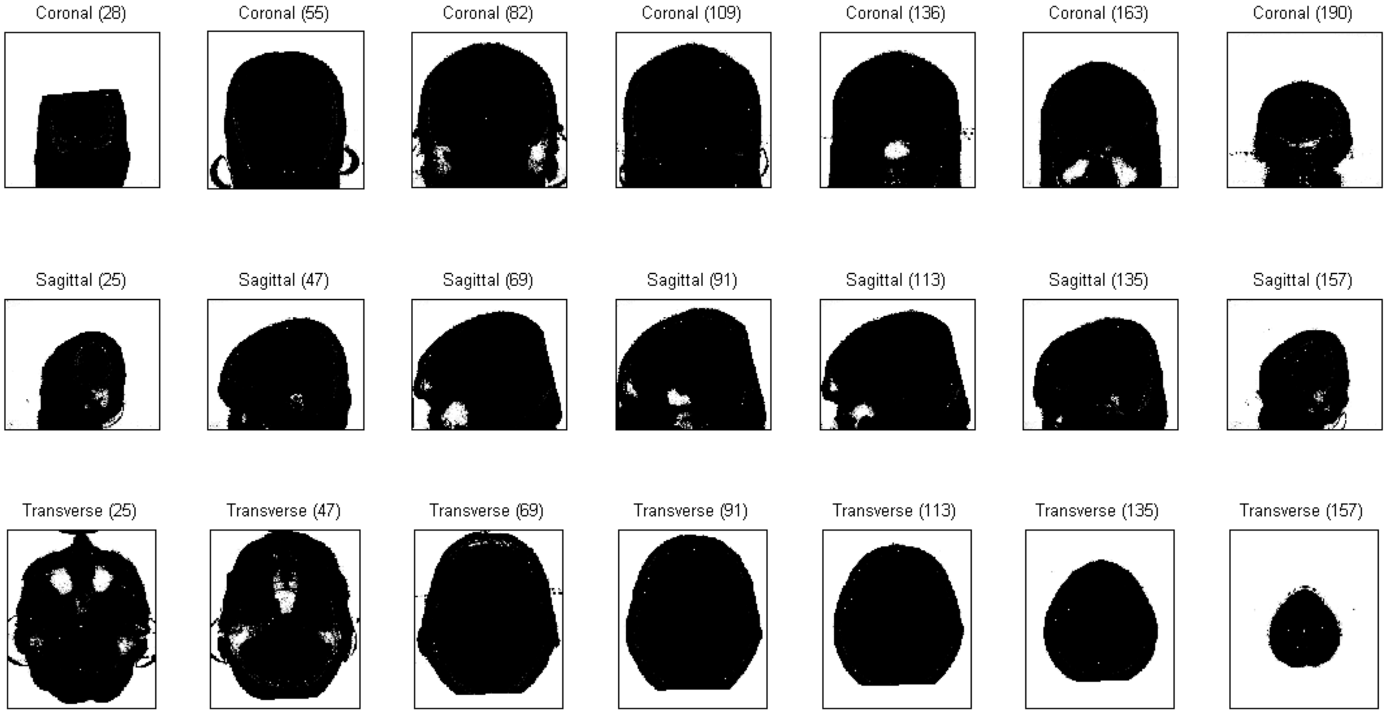


Figure E.16: Segmentation of volumes from subject f4395 - Background.

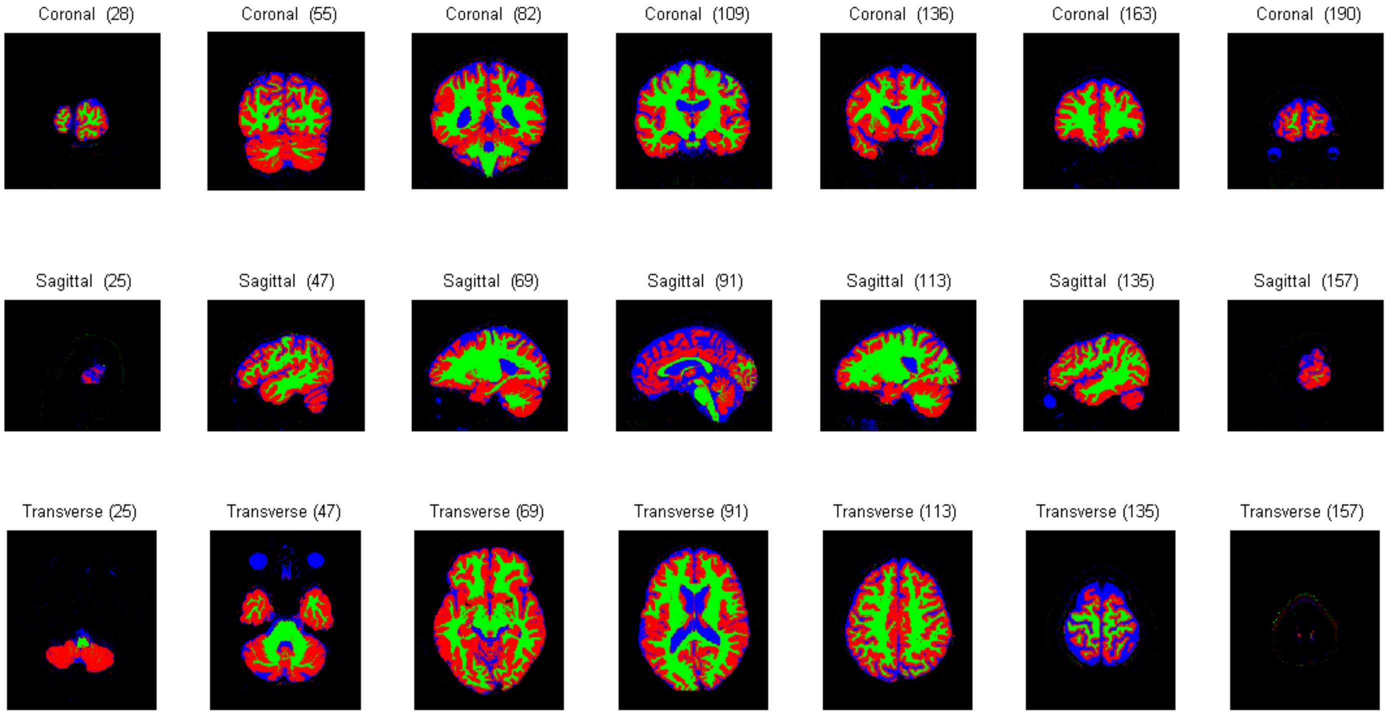


Figure E.17: Segmentation of volumes from subject f4395 - GM/WM/CSF overlap. GM, WM and CSF are presented in red, green and blue respectively.

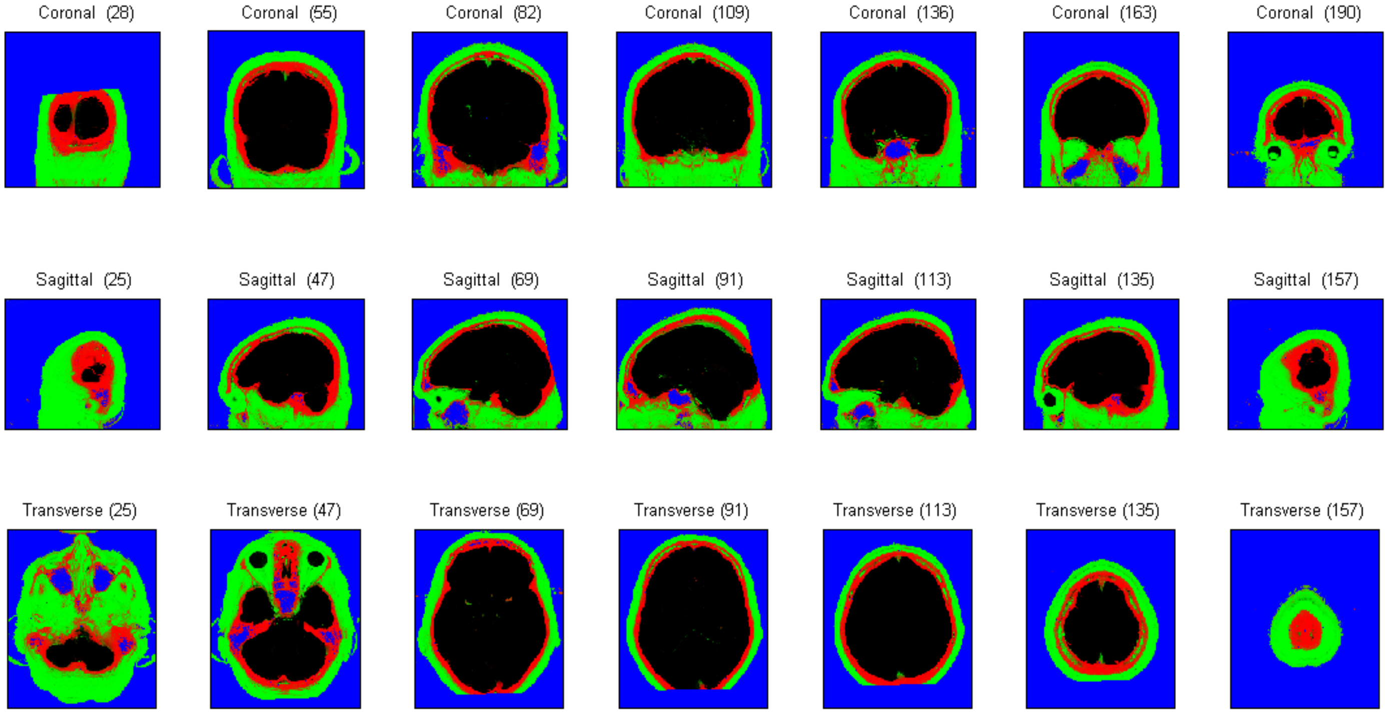


Figure E.18: Segmentation of volumes from subject f4395 - ST/Bone/BG overlap. ST, Bone and BG are presented in red, green and blue, respectively.

Appendix F

Matlab code

This chapter includes the Matlab code of several functions that have been used to depicts volumes generated by SPM segmentation.

- classify_voxels(): Function that coverts a probability map into a binary map. It applies majority voting or thresholding on the tissue volumes.
- plot_volume(): Scrip that plots several slices of one volume in the three planes (coronal, sagittal and transverse).
- plot_volume_overlap(): Script that presents the overlap of 2 or three volumes for several planes.
- GenerateRGB(): Function that combines three images into one RGB image with different levels of scaling.

classify_voxels()

```
function [atlasc,volc,results] = classify_voxels(atlas,vol,th,mask)
%   Function that asigns the tissue class according to the
%  generated TPM after the segmentation, thus converts
%  probability maps into binary maps
%   atlas: true labels in a cell with four 3D matrices.
%           GM, WM, CSF, no—brain and BG
%   vol: segmented volumes in a cell with four 3D matrices.
%           GM, WM, CSF, no—brain and BG
%   th: labeling threshold; if th=0, mojority voting
%   mask: it can be a mask of an intensity map
%
% example: atlas={GMatlas,WMatlas,CSFatlas,nobrainatlas,BGatlas};
%    vol  ={GMvol,WMvol,CSFvol,nobrainvol,BGvol};
%   [atlasc,volc,results] = classify_voxels(atlas,vol,0);
%
%   Author: Angel Diego Cuñado Alonso (diegoalonso@ieee.org)
%   Technical University of Denmark, DTU (2011)
%


% Mask
if nargin < 4,
    mask = single(ones(size(vol{1})));
end


%%%%%%%%%%%%%   Assign tissue labels (atlas)   %%%%%%%%%%%%%%%%%
% GM, WM, CSF and no—brain voxels
atlasc{1} = single(zeros(size(atlas{1})));
atlasc{2} = single(zeros(size(atlas{2})));
atlasc{3} = single(zeros(size(atlas{3})));
atlasc{4} = single(zeros(size(atlas{4})));
atlasc{5} = single(zeros(size(atlas{5})));
if th==0
    % Select class according to maximum probability
    [x,idx] = max([atlas{1}(:), atlas{2}(:), atlas{3}(:), ...
                            atlas{4}(:), atlas{5}(:)],[],2);
    clear x
    atlasc{1}(idx==1) = mask(idx==1);
    atlasc{2}(idx==2) = mask(idx==2);
    atlasc{3}(idx==3) = mask(idx==3);
    atlasc{4}(idx==4) = mask(idx==4);
    atlasc{5}(idx==5) = mask(idx==5);
else
    % Select class according to threshold
    atlasc{1}(atlas{1}(:)>th) = mask(atlas{1}(:)>th);
    atlasc{2}(atlas{2}(:)>th) = mask(atlas{2}(:)>th);
    atlasc{3}(atlas{3}(:)>th) = mask(atlas{3}(:)>th);
    atlasc{4}(atlas{4}(:)>th) = mask(atlas{4}(:)>th);
    atlasc{5}(atlas{5}(:)>th) = mask(atlas{5}(:)>th);
end

```

```
%%%%%%%%%%%%   Assign tissue labels (volumen)   %%%%%%%%%%%%%%%%%
% GM, WM, CSF and no—brain voxels
volc{1} = single(zeros(size(vol{1})));
volc{2} = single(zeros(size(vol{2})));
volc{3} = single(zeros(size(vol{3})));
volc{4} = single(zeros(size(vol{4})));
volc{5} = single(zeros(size(vol{5})));
if th==0
    % Select class according to maximum probability
    [x,idx] = max([vol{1}(:), vol{2}(:), vol{3}(:), ...
                            vol{4}(:), vol{5}(:)],[],2);
    clear x
    volc{1}(idx==1) = mask(idx==1);
    volc{2}(idx==2) = mask(idx==2);
    volc{3}(idx==3) = mask(idx==3);
    volc{4}(idx==4) = mask(idx==4);
    volc{5}(idx==5) = mask(idx==5);
else
    % Select class according to threshold
    volc{1}(vol{1}(:)>th) = mask(vol{1}(:)>th);
    volc{2}(vol{2}(:)>th) = mask(vol{2}(:)>th);
    volc{3}(vol{3}(:)>th) = mask(vol{3}(:)>th);
    volc{4}(vol{4}(:)>th) = mask(vol{4}(:)>th);
    volc{5}(vol{5}(:)>th) = mask(vol{5}(:)>th);
end


%%%%%%%%%%%%%%%%%%%%%%%%   Classify  %%%%%%%%%%%%%%%%%%%%%%
% Dice=(2TP)/(2TP+FP+FN)
for i=1:5
    results{i}.TP = nnz((volc{i}==atlasc{i}).*(volc{i}==1));
    results{i}.TN = nnz((volc{i}==atlasc{i}).*(volc{i}==0));
    results{i}.FP = nnz((volc{i}≠atlasc{i}).*(volc{i}==1));
    results{i}.FN = nnz((volc{i}≠atlasc{i}).*(volc{i}==0));
    results{i}.Dice = (2*results{i}.TP)/...
                (2*results{i}.TP+results{i}.FP+results{i}.FN);
end


return
```

plot_volume()

```
function plot_volume(vol,num_slices,opt)
% Script that plots slices of the three planes of the brain volume
%   vol: 3D matrix of the brain intensity values of each voxel
%   num_slices: total (odd) number of plotted slices per plane
%   opt: plot option, opt=0 uses imagesc(), opt=1 uses image()
%
%   Author: Angel Diego Cuñado Alonso (diegoalonso@ieee.org)
%   Technical University of Denmark, DTU (2011)
%

    % Default options
    if nargin < 2,  num_slices = 5; end
    if nargin < 3,  opt = 0;        end

    figure
    colormap(gray)

    % Estimate slices to represent (odd number)
    if mod(num_slices+1,2), num_slices = num_slices+1; end
    half_slice = round(size(vol)/2)';
    step_slice = floor(half_slice/(((num_slices-1)/2)+1));
    slice = zeros(3,num_slices);
    slice(:,(num_slices+1)/2) = half_slice;
    for i=1:(num_slices-1)/2
        slice(:,((num_slices+1)/2)-i) = half_slice-i*step_slice;
        slice(:,((num_slices+1)/2)+i) = half_slice+i*step_slice;
    end

    % Coronal
    for i=1:num_slices
      subplot(3,num_slices,i)
      plane=fliplr(rot90(reshape(vol(:,slice(2,i),:),size(vol,1),
          size(vol,3))));
      if opt, image(plane); else  imagesc(plane); end
      title(['Coronal (',num2str(slice(2,i)),')']);
      axis image; set(gca,'XTick',[],'YTick',[]);
    end

    % Sagittal
    for i=1:num_slices
      subplot(3,num_slices,i+num_slices)
      plane = fliplr(rot90(reshape(vol(slice(1,i),:,:),size(vol,2),
          size(vol,3))));
      if opt, image(plane); else  imagesc(plane); end
      title(['Sagittal (',num2str(slice(1,i)),')']);
      axis image; set(gca,'XTick',[],'YTick',[]);
    end

    % Transverse
    for i=1:num_slices
      subplot(3,num_slices,i+2*num_slices)
      plane = rot90(reshape(vol(:,:,slice(3,i)),size(vol,1),size(vol,2)));
      if opt, image(plane); else  imagesc(plane); end
      title(['Transverse (',num2str(slice(3,i)),')']);
      axis image; set(gca,'XTick',[],'YTick',[]);
    end

    % Maximize figure window
    set(gcf, 'Position', get(0,'Screensize'));

end
```

plot_volume_overlap()

```
function plot_volume_overlap(vols,num_slices,opt)
% Script that plots several slices of the three planes of the
%  overlaped brain volume
%   vols: cell of 3D matrices of brain intensity values, with
%        maximum 3 co−registered volumes with the same dimensions
%   num_slices: total (odd) number of plotted slices per plane
%   opt: plot option, opt=0 uses imagesc(), opt=1 uses image()
%
%   example: plot_volume_overlap({vol1, vol2, vol3},[100,90,100])
%
%   Author: Angel Diego Cuñado Alonso (diegoalonso@ieee.org)
%   Technical University of Denmark, DTU (2011)
%

    % Different number of input volumes
    switch numel(vols),
      case 1,
        Nvol = 1;
        dim = size(vols{1});
        vols = {vols{1},zeros(dim,'uint16'),zeros(dim,'uint16')};
      case 2,
        Nvol = 2;
        dim = size(vols{1});
        vols = {vols{1},vols{2},zeros(dim,'uint16')};
      case 3,
        Nvol = 3;
        dim = size(vols{1});
      otherwise,
        Nvol = 1;
        dim = size(vols);
        vols = {vols,zeros(dim,'uint16'),zeros(dim,'uint16')};
    end

    % Default options
    if nargin < 2,  num_slices = 5; end
    if nargin < 3,  opt = 0;        end

    % Estimate slices to represent (odd number)
    if mod(num_slices+1,2), num_slices = num_slices+1; end
    half_slice = round(dim/2)';
    step_slice = floor(half_slice/(((num_slices-1)/2)+1));
    slice = zeros(3,num_slices);
    slice(:,(num_slices+1)/2) = half_slice;
    for i=1:(num_slices-1)/2
        slice(:,((num_slices+1)/2)-i) = half_slice-i*step_slice;
        slice(:,((num_slices+1)/2)+i) = half_slice+i*step_slice;
    end

    % Extract the images and overlap
    plane = cell(3,3);
    planes = cell(3,num_slices);
    for i=1:num_slices
      for j=1:3
        plane{j,1}=fliplr(rot90(reshape(vols{j}(:,slice(2,i),:),dim(1),dim(3))));
        plane{j,2}=fliplr(rot90(reshape(vols{j}(slice(1,i),:,:),dim(2),dim(3))));
        plane{j,3}=rot90(reshape(vols{j}(:,:,slice(3,i)),dim(1),dim(2)));
      end
      for k=1:3
          planes{k,i} = GenerateRGB(plane{1,k},plane{2,k},plane{3,k},2);
      end
    end

    % Plot
    figure
    colormap(gray)
    planes_str = ['  Coronal ';' Sagittal ';'Transverse'];
    slice_order = [2,1,3];
    for j=1:3
      for i=1:num_slices
        subplot(3,num_slices,i+(j-1)*num_slices)
        if opt, image(planes{j,i}); else  imagesc(planes{j,i}); end
        title([planes_str(j,:),' (',num2str(slice(slice_order(j),i)),')']);
        axis image; set(gca,'XTick',[],'YTick',[]);
      end
    end
    set(gcf, 'Position', get(0,'Screensize'));  % Maximize figure window

end
```

GenerateRGB()

```
function rgb_out = GenerateRGB(imageRed, imageGreen, imageBlue, opt)
% Function that overlaps three images in different RGB channel
% imageRed, imageGreen, imageBlue: 2D matrices of each image
% opt=0: original values;    opt=1: 255;    opt=2: scaled values
% Ex. figure; imagesc(GenerateRGB([0 0;0 0],[0,0;0 0],[1 2;3 4],1));
%
%   Author: Angel Diego Cuñado Alonso (diegoalonso@ieee.org)
%   Technical University of Denmark, DTU (2011)
%


    % Format images
    imageRed   = double(imageRed);
    imageGreen = double(imageGreen);
    imageBlue  = double(imageBlue);

    % Scaling factor
    if nargin < 4,  opt=0;    end
    switch opt,
      case 0,
        scaling = 1;
      case 1,
        scaling = 255;
      case 2,
        scaling = (2^8)./[max(imageRed(:)),max(imageGreen(:)),max(imageBlue(:))];
    end
    scaling = cast(scaling,’double’);

    % Initialize red, green, and blue matrices
    blue = zeros(max([size(imageRed);size(imageGreen);size(imageBlue)]),’double’);
    green = blue;
    red = blue;

    % Scale images
    red(1:size(imageRed,1), 1:size(imageRed,2)) = scaling(1) * imageRed;
    green(1:size(imageGreen,1), 1:size(imageGreen,2)) = scaling(2) * imageGreen;
    blue(1:size(imageBlue,1), 1:size(imageBlue,2)) = scaling(3) * imageBlue;

    % Combine the red, green, and blue components into an RGB image.
    rgb_out = cast(cat(3, red, green, blue),’uint8’);

return
```

List of Figures

1.1 Neurobiology Research Unit 1
1.2 Magnetom Trio scanner of Siemens 2
1.3 T1 and T2 MRI data preview of subject f4395. 3
1.4 MRI segmentation steps of the original pipeline. 5

2.1 Shuttleworth brain drawing 9
2.2 Human brain representation 11
2.3 Intensity histogram of the segmentation 14
2.4 Representation of brain slices. 15
2.5 T1 MRI data preview with SPM5 15
2.6 Segmentation of a T1 MR image 16
2.7 Segmentation of a T1 MR image with original baseline. 18

3.1 Intensity histograms of the MOG. 25
3.2 Human bone and soft tissue in the head 27
3.3 Intensity histogram of the head voxels 27
3.4 Overlapped 2D intensity histogram of T1 and T2. 28
3.5 Joint 2D intensity histogram of T1 and T2. 29
3.6 Affine transformation 32
3.7 First basis functions of DCT 35
3.8 Example of volume registration in SPM - histograms 38
3.9 Example of volume registration in SPM 39
3.10 Example of bias field correction. 42
3.11 Example of scalp stripping doen by BET 43
3.12 Result of the scalp stripping with the original pipeline. 44
3.13 Intensity histogram of the scalp stripping with SPM. 45
3.14 Example of smoothing. 47

3.15 Templates for GM, WM and CSG in 'New Segmentation'. 49

4.1 Example of regularization 62
4.2 Transverse slices 77
4.3 Plot of a 3th degree equation. 83
4.4 Log-likelihood function. 85

5.1 Original volumes of f4395. 93
5.2 Registered volumes of f4395. 94
5.3 Log-likelihood value at each iteration. 96
5.4 Zoom of the cluster representation. 98
5.5 Bias Field correction for T1 and T2 modalities. 99
5.6 Effect of the bias field correction in the intensity histogram. . . . 100
5.7 Probability and overlapped tissues of f4395. 102
5.8 Overlapped tissues of f4395 with original and new baseline. . . . 103
5.9 BrainWeb phantoms with 3% noise. 105
5.10 BrainWeb phantoms with 9% noise. 105
5.11 Majority Voting process for original and modified v.3 method. . . 107
5.12 Comparison segmentation methods in terms of the Dice score. . . 108
5.13 Example of the improvement of v.3 in the segmentation of GM. . 110
5.14 Example of the improvement of v.3 in the segmentation of CSF. 111
5.15 Volume age profile for six volumes. 113

A.1 Precession movement of NRU phenomenon 132
A.2 Stimulation of the Nuclear Magnetic Resonance 134
A.3 Relaxation time T1 T2 136

B.1 Gaussian Distribution 138
B.2 2D Gaussian with 0 deg. rotation. 140
B.3 2D Gaussian with 45 deg. rotation. 140
B.4 Examples of problems in the MOG model. 141
B.5 Affine Transformation example in 2D. 152

D.1 Mixture Coefficient through iterations. 168
D.2 T1 Mean Value through iterations. 169
D.3 T2 Mean Value through iterations. 170
D.4 T1 Variance Value through iterations. 171
D.5 T2 Variance Value through iterations. 172
D.6 T1-T2 Covariance Value through iterations. 173
D.7 Clusters Representation for the modified version 1. 174
D.8 Clusters Representation for the modified version 2. 175
D.9 Clusters Representation for the modified version 3. 176
D.10 Clusters Representation for the modified version 4. 177
D.11 Performance of the original method. 181

D.12 Performance of the original method. 182
D.13 Performance of the modified v.3 method. 183
D.14 Performance of the modified v.3 method after Majority Voting. . 184
D.15 Estimated volumes related to the age 186
D.16 Regression of volume age profile - Original Method. 188
D.17 Regression of volume age profile - Original Method. 189

E.1 T1 MRI volume of subject f4395. 192
E.2 T2 MRI volume of subject f4395. 193
E.3 Tissue Probability Map for Prior Templates - Grey Matter. . . . 194
E.4 Tissue Probability Map for Prior Templates - White Matter. . . 195
E.5 Tissue Probability Map for Prior Templates - CerebroSpinal Fluid.196
E.6 Tissue Probability Map for Prior Templates - Bone. 197
E.7 Tissue Probability Map for Prior Templates - Soft Tissue. 198
E.8 Tissue Probability Map for Prior Templates - Background. 199
E.9 Template Tissue Probability Maps - GM/WM/CSF. 200
E.10 Template Tissue Probability Maps - Bone/ST/BG. 201
E.11 Segmentation of volumes from subject f4395 - Grey Matter. . . . 202
E.12 Segmentation of volumes from subject f4395 - White Matter. . . 203
E.13 Segmentation of volumes from subject f4395 - CerebroSpinal Fluid.204
E.14 Segmentation of volumes from subject f4395 - Bone. 205
E.15 Segmentation of volumes from subject f4395 - Soft Tissue. 206
E.16 Segmentation of volumes from subject f4395 - Background. . . . 207
E.17 Segmentation of volumes from subject f4395 - GM/WM/CSF. . . 208
E.18 Segmentation of volumes from subject f4395 - ST/Bone/BG. . . 209

List of Tables

5.1 Results of the segmentation for the subject f4395. 88
5.2 Performance of the original and the modified methods. 95
5.3 Values of the mixture parameters. 97

D.1 Result of the segmentation of BrainWeb phantoms with noise=0%.178
D.2 Result of the segmentation of BrainWeb phantoms with noise=3%.179
D.3 Result of the segmentation of BrainWeb phantoms with noise=9%.180
D.4 Atrophy of the brain related to the ageing 187

List of Algorithms

1 Control flow of the function spm_prepoc8T1T2.m 78
2 Algorithm to estimate the adequate cross-variance value 84