

Mariana Bustamante

Detection and Quantification of Small Changes in MRI Volumes

Centre for Image Analysis



UPPSALA
UNIVERSITET

Abstract

Please use no more than 300 words and avoid mathematics or complex script.

*Dedicated to all hard-working
doctoral students at Uppsala University*

Contents

| | | |
|-------|--------------------------------------|----|
| 1 | Introduction | 7 |
| 1.1 | Description of the problem | 7 |
| 1.2 | Importance | 7 |
| 1.3 | Related Works | 8 |
| 2 | Methods | 10 |
| 2.1 | Registration | 10 |
| 2.1.1 | Registration Methods used | 11 |
| | Affine Registration | 11 |
| | B-Spline Registration | 12 |
| | BRAINS Demon Warp Registration | 12 |
| 2.2 | Morphometry | 15 |
| 2.2.1 | Morphometry methods used | 15 |
| | Voxel-based Morphometry | 15 |
| | Tensor-based Morphometry | 16 |
| 3 | Proposed Solutions | 18 |
| 3.1 | Tools used | 18 |
| 3.1.1 | 3D Slicer | 18 |
| 3.1.2 | ITK | 19 |
| 3.1.3 | Other Tools | 19 |
| | Programming Languages | 19 |
| | MATLAB | 19 |
| | ParaView | 20 |
| 3.2 | Implemented Methods | 20 |
| 3.2.1 | Voxel-based method | 20 |
| | Example | 21 |
| | Technical Details | 23 |
| 3.2.2 | Tensor-based method | 24 |
| | Example | 25 |
| | Technical Details | 28 |
| 4 | Experiments | 29 |
| 4.1 | Experiments Conditions | 29 |
| 4.1.1 | Artificial Differences | 29 |
| 4.1.2 | Real Differences | 29 |
| 4.2 | Artificial Differences Results | 29 |
| 4.2.1 | Size: Large | 30 |
| | Voxel-based Method | 30 |

| | | |
|-------|--------------------------------|----|
| | Tensor-based Method | 31 |
| 4.2.2 | Size: Medium | 32 |
| | Voxel-based Method | 33 |
| | Tensor-based Method | 34 |
| 4.2.3 | Size: Small | 35 |
| | Voxel-based Method | 35 |
| | Tensor-based Method | 36 |
| 4.3 | Real Differences Results | 38 |
| 4.3.1 | Patient 1 | 38 |
| | Voxel-based Method | 38 |
| | Tensor-based Method | 39 |
| 4.3.2 | Patient 2 | 40 |
| | Voxel-based Method | 40 |
| | Tensor-based Method | 42 |
| 4.3.3 | Patient 3 | 44 |
| | Voxel-based Method | 44 |
| | Tensor-based Method | 46 |
| 5 | Discussion | 49 |
| 5.1 | Voxel-based method | 49 |
| 5.2 | Tensor-based method | 49 |
| 6 | Applications | 51 |
| 7 | Conclusions | 52 |
| 7.1 | Future works | 52 |
| | References | 54 |

1. Introduction

1.1 Description of the problem

When a person suffers an accident that produces trauma to the head, one of the first things that should be performed is an MRI of the brain region in order to spot possible brain injuries. If the damage is large, a medical specialist can usually recognize which parts of the brain have been affected without automated help. However, if the damage is minor, it is generally much harder for the doctor to figure out what could be the consequences of the injury.

For this particular project we have the following conditions: A patient's brain is scanned, producing a first MRI just after receiving trauma to the head. The patient has not received extensive damage, and so the MRI comes out as expected from a medical specialist. After a period of three to twelve months later, a new MRI of the same patient is taken and is compared with the initial MRI.

It is assumed that if there are symptoms produced as a consequence of the accident, then there must be differences in the brain of the patient, even when the differences might be too small to see with the naked eye.

The goal of this project is to analyse and test different registration and morphometry methods, in order to compare this type of brain MRIs and produce information on the differences between them.

The final result of the project should be a tool that allows the medical specialist to do the comparison on his or her own.

1.2 Importance

Manual examination of MRI studies suffers from many problems. The study can be affected specially by acquisition related factors such as:

- One cannot assume a one-to-one correspondence between slices from one acquisition to the next in order to make side-by-side comparisons.
- A different scanner may be used in every examination, producing a scan which will have different signal characteristics.
- Often, scanning parameters are not the same form one acquisition to the next.

- Change can present itself in many ways. The radiologist is required to assimilate all this data before making a decision, which often is quite difficult.

A very important part of change detection is therefore not simply the detection of change but the separation of acquisition-related change from disease-related change. Also, methods that produce objective, reproducible and accurate metrics of disease course are of great interest since the change in appearance over time is essential to understanding disease course [10].

Also, it is not enough to be able to determine if there are differences, since from a clinical standpoint, knowing where and how changes have occurred is as important as knowing that they have occurred [10].

From a practical point of view, it is also very important for the patients to be able to obtain a correct diagnosis, specially if the symptoms they are suffering are hard to explain or subjective, as it is common with non-extensive brain injuries.

1.3 Related Works

Many related works focus on inter-subject studies, which means that they compare images acquired from different individuals in order to diagnose diseases or identify abnormalities. This generally implies that the first step in the analysis process is one of *spatial normalization* or *inter-subject registration*, in which the aim is to reduce the anatomical variability in the volumetric brain scans.

The works of [21, 13] use this technique combined with voxelwise group analysis of functional magnetic resonance imaging (fMRI), and the works of [1, 6] also use spatial normalization combined with diffusion tensor imaging in order to study brain white matter.

More similar works, related to the quantification of small changes in volume observed over time can be observed in [5, 12]. The results of these works are used to diagnose and evaluate disease progression and treatment.

Some other works that also use the specific techniques described later in this project are:

- Hajnal et al. in [4] and Lemieux et al. in [9] utilize subtraction after applying rigid registration on the volumes.
- Rey et al. in [12] introduce the Jacobian operator of the deformation field resulting from the registration as a measure of local volume variation.
- Pohl et al. in [11] describe a semiautomatic procedure targeted toward identifying difficult-to-detect changes in brain tumor imaging. The result of this study is also a module for *3D Slicer*, and was very useful for this project since its source code is open for other researchers to view.

More information about the module can be found on its webpage:
<http://www.slicer.org/slicerWiki/index.php/Documentation/4.1/Modules/ChangeTracker>

2. Methods

This chapter focuses on the theory necessary to understand the research and work done during this project.

It includes the concepts of *registration* and *morphometry*, which were used extensively throughout this work. Each section includes the theory on which the concept is based and descriptions of the specific techniques used for the analysis of volumes.

2.1 Registration

This section describes the concept of *registration*, this is the idea on which most of this project is based on, and it is essential in order to obtain a usable final result.

The following good definition of registration is given by [22]:

“Image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. It geometrically align two images –the reference and sensed images.”

“Image registration is a crucial step in all image analysis tasks in which the final information is gained from the combination of various data sources like in image fusion, change detection, and multichannel image restoration.”

The following images show a checkerboard comparison of two volumes taken of the same patient at different times. Figure 2.1 shows the image comparison before performing affine registration. Figure 2.2 shows the comparison of images after the registration has been done.

The second volume has been modified using affine registration so that the comparison between both volumes becomes much easier.

According to [22], the majority of the registration methods consist of the following steps:

1. *Feature detection*: Salient and distinctive objects are detected.
2. *Feature matching*: The correspondence between the features detected in the sensed image and those detected in the reference image is established.

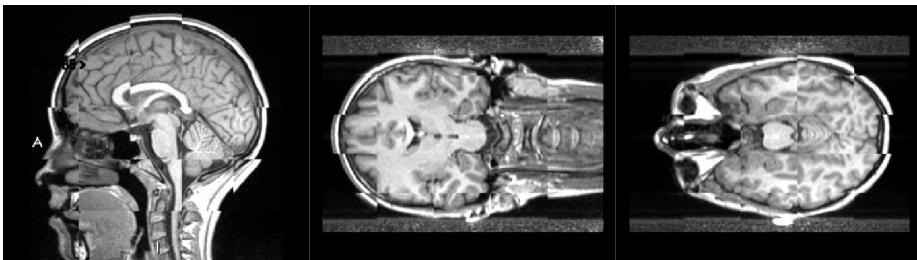


Figure 2.1. Comparison of volumes before registration

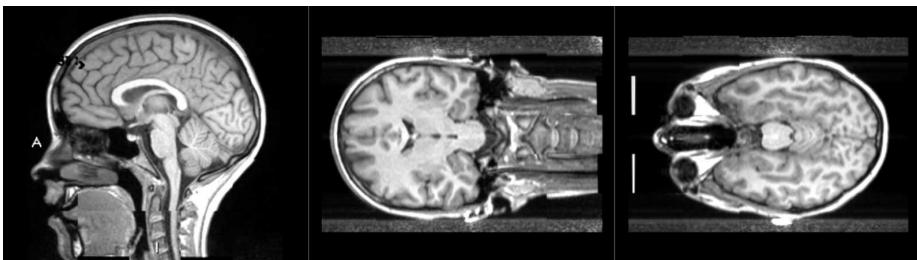


Figure 2.2. Comparison of volumes after registration

3. *Transform model estimation:* The type and parameters of the mapping functions are estimated. These functions align the sensed image with the reference image.
4. *Image resampling and transformation:* The sensed image is transformed by means of the mapping functions.

2.1.1 Registration Methods used

There are many different methods to accomplish registration between images or volumes. In this project only a few of these methods were used based on experiments performed with some of the available methods. The chosen procedures were the ones that behaved better under the specific conditions of this work.

Affine Registration

In geometry, an *affine transformation* or an *affinity* is a transformation which preserves straight lines (i.e., all points lying on a line initially still lie on a line after the transformation) and ratios of distances between points lying on a straight line. It does not necessarily preserve angles or lengths, but does have the property that sets of parallel lines will remain parallel to each other after

an affine transformation [16].

The implementation of affine registration used during this project is the one included in *3D Slicer* version 4.1.

The application is able to register two images together using an affine transform and mutual information, and it allows the user to modify quite a few parameters in order to obtain the expected result.

If you would like to know more about this implementation of affine registration please refer to the module webpage:

<http://wiki.slicer.org/slicerWiki/index.php/Documentation/4.1/Modules/AffineRegistration>

B-Spline Registration

In the mathematical subfield of numerical analysis, a *B-spline* is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition.

However, in computer graphics, the term *B-spline* frequently refers to a spline curve parametrized by spline functions that are expressed as linear combinations of *B-splines* (in the mathematical sense explained above) [17].

Just like in the case of affine registration, the implementation of B-Spline registration used during this project is the one included in *3D Slicer* version 4.1.

The application divides the volumes into a grid of user-defined size in which each line is a B-spline that will be modified by the registration to create a transform that aligns the two volumes.

If you would like to know more about this implementation of B-spline registration please refer to the module webpage:

<http://wiki.slicer.org/slicerWiki/index.php/Documentation/4.1/Modules/BSplineDeformableRegistration>

BRAINS Demon Warp Registration

As with the registration methods explained before, *BRAINS Demon Warp* registration is implemented as a module in *3D Slicer* version 4.1.

Before the registration takes place, there is an initial pre-processing step in which the intensity mismatch between the volumes is corrected by histogram matching. This prevents future errors in the registration, since a good result depends on the volumes intensities being similar.

The method's implementation in *3D Slicer* contains other interesting parameters that can be modified by the user. The ones that were found most useful for this research are:

- *Displacement field smoothing sigma*: A gaussian smoothing value to be applied to the deformation field at each iteration.
- *Update field smoothing*: Smoothing sigma for the update field at each iteration.

Some resulting deformation fields exemplifying the changes produced by modifying this values can be seen next:

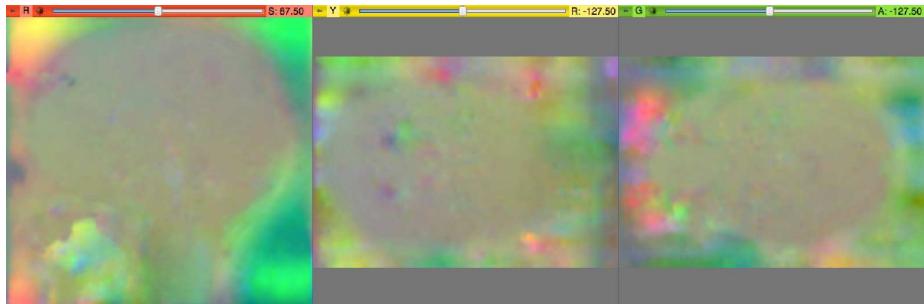


Figure 2.3. Displacement field smoothing sigma of 1.0 (default value)



Figure 2.4. Displacement field smoothing sigma of 2.5

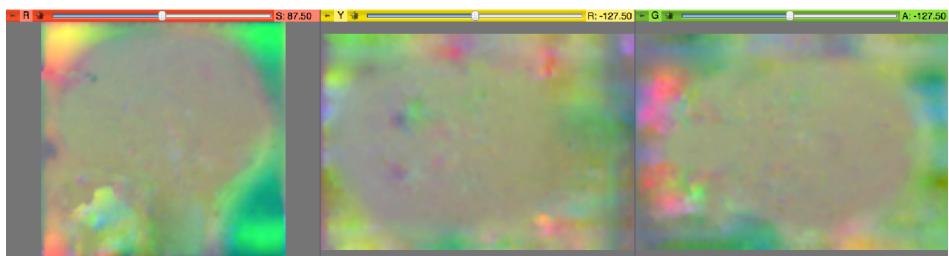


Figure 2.5. Update field smoothing of 0.0 (default)



Figure 2.6. Update field smoothing of 2.0

The actual registration step works by using the ITK filter based on *Thirion's Demons algorithm*, in which the main idea is to consider the objects boundaries in one image as semi-permeable membranes and to let the other image, considered as a deformable grid model, diffuse through these interfaces, by the action of effectors situated within the membranes [14].

An important characteristic of this implementation, that was particularly useful for this project, is that it can produce a *deformation field* as the output of the registration. A deformation field is a vector image in which each point is a vector that indicates the amount of deformation necessary at that point in order to align the volumes.

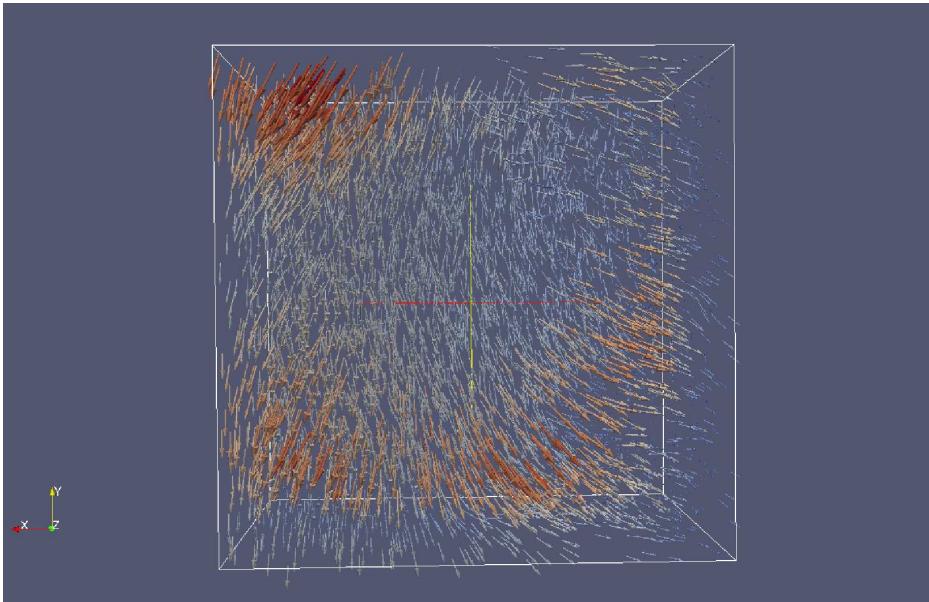


Figure 2.7. An example of a deformation field viewed with *ParaView*

If you would like to know more about the implementation of *BRAINS Demon Warp* registration please refer to the module webpage:

<http://wiki.slicer.org/slicerWiki/index.php/Documentation/4.1/Modules/BRAINSDemonWarp>

2.2 Morphometry

Morphometry refers to the measurement of external form. More specifically, *brain morphometry* is concerned with the measurement of brain structures and changes thereof during development, aging, learning, disease and evolution. Its goal is to derive specific information from noninvasive neuroimaging data of live brains, typically obtained from magnetic resonance imaging (MRI); and to quantify the anatomical features of the brain in terms of shape, mass and volume [18].

In general, brain morphometry can be divided into three different methods: *deformation-based*, *tensor-based* and *voxel-based* morphometry. Defined briefly as:

- *Deformation-based*: Uses deformation fields to identify differences in the relative positions of structures.
- *Tensor-based*: Uses deformation fields to identify differences in the local shape of brain structures.
- *Voxel-based*: Uses voxel-wise comparison of the local concentration of grey matter.

2.2.1 Morphometry methods used

Only *tensor-based* and *voxel-based* morphometry methods were used during this project, since their outputs were easier to adapt to our requirements.

Voxel-based Morphometry

A useful measure of structural difference among populations is derived from a comparison of the local composition of different brain tissue types (e.g., grey matter, white matter, etc). Voxel-based morphometry (VBM) has been designed to be sensitive to these differences, while discounting positional and other large scale volumetric differences in gross anatomy.

Since its inception, VBM has become an established tool in morphometry being used to detect cortical atrophy and differences in slender white matter tracts [3].

An objection to VBM is that it is sensitive to systematic shape differences attributable to misregistration. Many potential differences can arise as a result of movement or different positioning of the subject in the scanner, and

also there can be systematic differences in the relative intensity of grey matter voxels compared to white matter [3].

All these differences can be detected by VBM since they are all real differences among the data, even when they may not imply an increase or reduction in grey matter density.

When VBM is used to compare MRI data of many different subjects, as is the case many times, the process involves spatially normalizing all the images to the same stereotactic space, extracting the gray matter from the normalized images, smoothing, and finally performing a statistical analysis to localize, and make inferences about, group differences. The output from the method is a statistical parametric map showing regions where gray matter concentration differs significantly between groups [2].

For a detailed step-by-step description of VBM, please refer to [2].

Tensor-based Morphometry

The goal of tensor-based morphometry (TBM) is to determine regional shape differences.

A deformation field that maps one image to another can be considered a discrete vector field. By taking the gradients at each element of the field, a *Jacobian matrix* field is obtained, in which each element is a tensor describing the relative positions of the neighboring elements. Morphometric measures derived from this tensor field can be used to locate regions with different shapes. The field obtained by taking the determinants at each point gives a map of the structure volumes relative to those of a reference image [2].

In principle, the *Jacobian matrices* of the deformations (a 2nd order tensor field relating to the spatial derivatives of the transformation) should be more reliable indicators of brain shape than absolute deformations. Absolute deformations represent positions of brain structures, rather than local shape, and need to be quantified relative to some arbitrary reference position [3].

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

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} \\ \frac{\partial y_3}{\partial x_1} & \frac{\partial y_3}{\partial x_2} & \frac{\partial y_3}{\partial x_3} \end{bmatrix}$$

The form of TBM that was used in this project involves comparing relative volumes of different brain structures, where the volumes are derived from

Jacobian determinants at each point. According to [3], this type of morphometry is useful for studies that have specific questions about whether growth or volume loss has occurred, and so is appropriate for our problem.

The *Jacobian determinant* is the determinant of the *Jacobian matrix* and is sometimes simply called “the Jacobian”.

For a differentiable function f (a function whose derivative exists at each point in its domain), the *Jacobian determinant* of f ’s *Jacobian matrix* at a given point gives important information about the behavior of f near that point.

For instance, if the *Jacobian determinant* at a point p is positive, then f preserves orientation near p ; if it is negative, f reverses orientation. The absolute value of the *Jacobian determinant* at p gives us the factor by which the function f expands or shrinks volumes near p [19].

3. Proposed Solutions

During the course of this project two different methods where implemented, both taking into account previous works about morphometry and quantification of small changes in volumes.

This chapter contains a description of all the elements used to develop the methods presented, and also the technical details necessary to understand them.

3.1 Tools used

3.1.1 3D Slicer

Slicer, or **3D Slicer**, is a free, open source software package for visualization and image analysis. It is natively designed to be available on multiple platforms, including Windows, Linux and Mac Os X.

3D Slicer was chosen among a series of other tools because of its type of architecture, which allows new developers to write new modules that can be included in the program with relative simplicity. Also, its code is open source, which allows the programmer to analyse any part of the code before implementing his own.

[8] gives a list of other available software technologies, presented in the following table:

| Name | Distribution license | Brief list of features |
|-----------|---|--|
| BrainVISA | Open source software. CeCILL License. | - scientific visualization - signal and image processing |
| CMGUI | Open source software. Mozilla Public License. | - scientific visualization - scripting capabilities - model personalization - signal and image processing |
| MedINRIA | Free for non-commercial use. | - scientific visualization - model personalization - signal and image processing |
| MeVisLab | Basic version free for non-commercial use. SDK version requires a commercial license. | - scientific visualization - scripting capabilities - plug-in architecture - image processing |
| OpenMAF | Open source software. BSD-style License. | - generic application framework - scientific visualization - model personalization - image processing |
| ParaView | Open source software. BSD-style License. | - scientific visualization - scripting capabilities - plug-in architecture |
| Slicer 3D | Open source software. BSD-style License. | - scientific visualization - plug-in architecture - image processing |

Figure 3.1. List of current software projects focused on medical image processing and end-user software for model personalization

3D Slicer provides image registration, processing of DTI (diffusion tractography), an interface to external devices for image guidance support, and GPU-enabled volume rendering, among other capabilities. 3D Slicer has a modular organization that allows the easy addition of new functionality and provides a number of generic features not available in competing tools.

3D Slicer is built on VTK, a pipeline-based graphical library that is widely used in scientific visualization. In version 4, the core application is implemented in C++, and the API is available through a Python wrapper to facilitate rapid, iterative development and visualization in the included Python console. The user interface is implemented in Qt, and may be extended using either C++ or Python.

Slicer supports several types of modular development. Fully interactive, custom interfaces may be written in C++ or Python. Command-line programs in any language may be wrapped using a light-weight XML specification, from which a graphical interface is automatically generated [15].

For more information on this tool please refer to its official webpage:

<http://www.slicer.org/>

3.1.2 ITK

ITK stands for **Insight Segmentation and Registration Toolkit**, it's a cross-platform, open-source application development framework widely used for the development of image segmentation and image registration programs.

ITK is implemented in C++ and it is wrapped for Tcl, Python and Java. This enables developers to create software using a variety of programming languages.

ITK's code is highly efficient, which means that many software problems are discovered at compile-time, rather than at run-time during program execution. It also enables ITK to work on two, three, four or more dimensions.

For more information on this tool please refer to its official webpage:

<http://www.itk.org/>

3.1.3 Other Tools

Programming Languages

The programming languages chosen during this project are **C++** and **Python**, mainly because they are main languages in which 3D Slicer is written, which means that it was easier to communicate with 3D Slicer by using them.

MATLAB

MATLAB is a numerical computing environment and programming language. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran [20].

MATLAB was used during this project specifically to create and quickly manipulate MRI volume files.

Official website for this tool: <http://www.mathworks.com/>

ParaView

ParaView is an open-source, multi-platform data analysis and visualization application. ParaView users can quickly build visualizations to analyze their data using qualitative and quantitative techniques.

ParaView was used during this project to visualize the deformation fields produced after the registration of two volumes.

Official website for this tool: <http://www.paraview.org/>

3.2 Implemented Methods

Both methods are implemented as modules of *3D Slicer* and their functionalities can be called in two ways: Through the user interface of *3D Slicer*, or as a *CLI* (Command Line Interface) from any other module implemented for *3D Slicer*.

This allows the modules to be usable even from a newly developed *3D Slicer* module.

3.2.1 Voxel-based method

This method was implemented as a *3D Slicer* module with the following steps:

1. The user selects the base and follow-up volumes to be compared.
2. The user selects the registration method to be used and applies it on the volumes.
3. The module subtracts the base volume and the volume resulting from the registration and shows the resulting differences as colored layer on top of the base volume.

The registration methods available are: *Affine registration* (default), *B-Spline deformable registration* and *BRAINS Demon Warp registration*; all of them available as already existing modules in *3D Slicer*.

For more information on the registration methods, please refer to section 2.1.1.

The subtraction is done pixel-by-pixel and the result produces a label volume that shows the differences in color over the original base volume. The chosen color table for the label volume is “PET-Heat”, directly available in *3D Slicer*, since it seemed to produce a volume that was brighter and with easier to spot differences.

Example

The following is an example of a volume without major differences that has been modified in order to add some obvious differences in the shape of circles in both of the original volumes.

A real user of the application would go through the following steps:

1. The user selects the *MRIChangeDetectorModule* in *3D Slicer*.

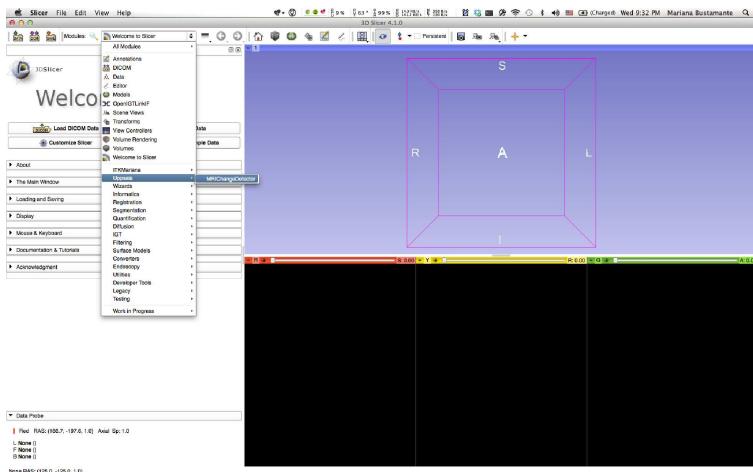


Figure 3.2. Step 0: Module selection

2. The user adds the volumes to be analysed.

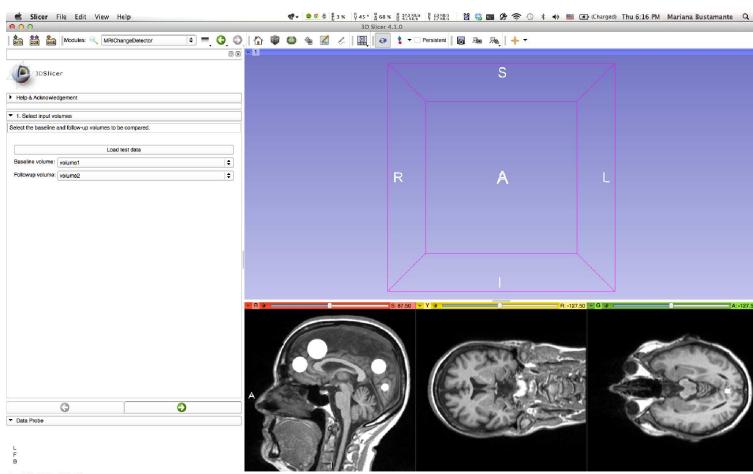


Figure 3.3. Step 1: Adding volumes

3. The user chooses the registration method to be applied.

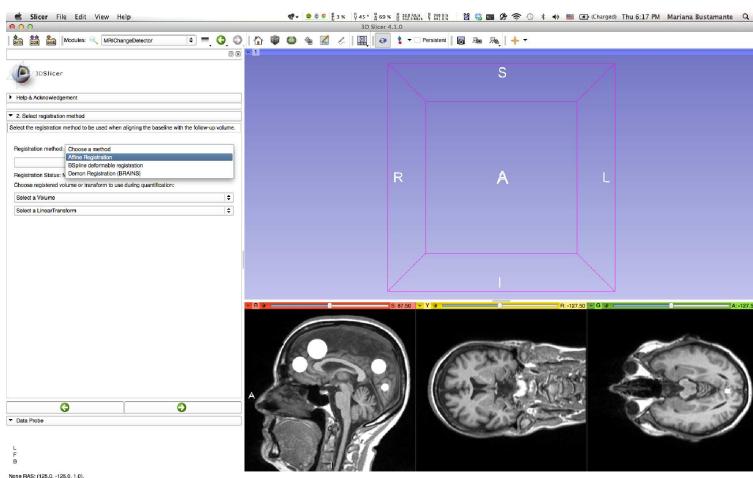


Figure 3.4. Step 2: Registration method

4. The user clicks the button “Run Quantification” and the program runs the subtraction and creation of label volume.

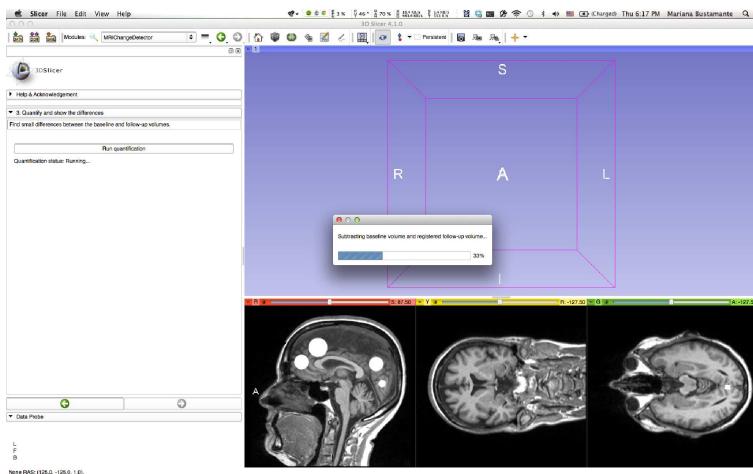


Figure 3.5. Step 3: Running quantification

5. The program shows the resulting label volume.

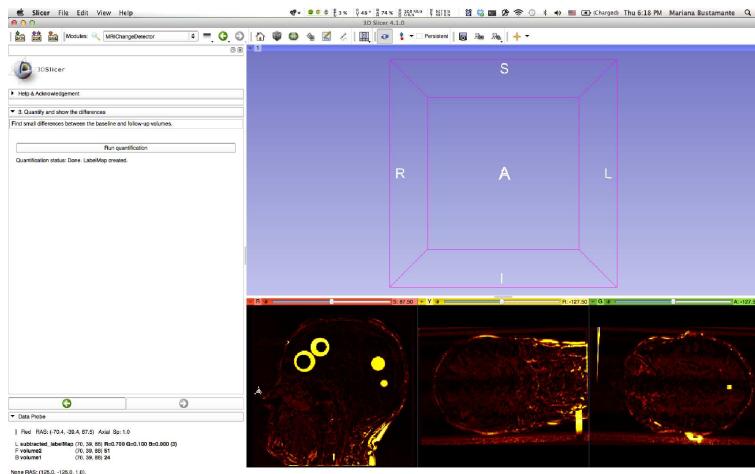


Figure 3.6. Step 4: Quantification result

Note that the resulting differences look like rings. This is the expected result because both of the original volumes were modified by adding circles of distinct sizes.

6. The user can now watch the label volume on top of the original volume and move the planes as he wishes.

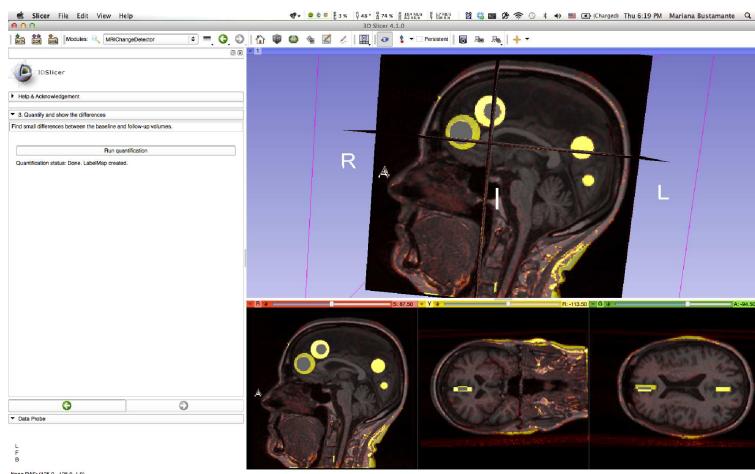


Figure 3.7. Step 5: User visualization

Technical Details

The user interface of the module is written in *Python* using some libraries from *Qt* and *CTK*, being *ctkWorkflowWidgetStep* from *CTK* the most important since it allows the creation of a “step-by-step” wizard.

The part of the module in charge of the subtraction of volumes is written in C++ using *ITK*.

The subtraction is done using the ITK filter *AbsoluteValueDifferenceImageFilter*, which computes the difference between each two pixels and then calculates the absolute value of the result. This allows the module to detect all the possible differences between the volumes, regardless of the sign of the resulting values.

3.2.2 Tensor-based method

This method was also implemented as a *3D Slicer* module with the following steps:

1. The user selects the base and follow-up volumes to be compared.
2. The user selects the displacement field smoothing sigma to use during the registration and runs it.
3. The user selects the percentage of growth and shrinkage that he would like to see and runs the measurements. The module shows the result as a colored layer on top of the base volume.

The registration method that is used in this case is *BRAINS Demon Warp Registration*, for more information on this method please refer to section 2.1.1.

The *displacement field smoothing sigma* is a Gaussian smoothing value to be applied to the output deformation field at each iteration of the registration process. Increasing this value produces deformation fields that are smoother and where the vector values are less prone to show big changes over very small movements in the field.

The default value for the smoothing sigma is 1.0; however, according to our experiments, to produce deformation fields that are smooth enough the value should be between 2.0 and 3.5.

The module also allows the user to choose the percentage of the values of the *Jacobian determinants* that will be shown in the final result. Since there are two types of values (corresponding to growing and shrinking of the original volume), there is also two values to select. For example, if the user chooses to view 50% of the growth values, only half of the values that represent growth in the deformation field will actually be shown on the resulting label map.

Once the user has seen the results with certain percentages, the module can be run again with a new set of parameters in order to get better results.

Example

The tensor-based method will be applied to the same volumes used in the previous example. A real user of the application would go through the following steps:

1. The user selects the *TensorMRIChangeDetectorModule* in *3D Slicer*.

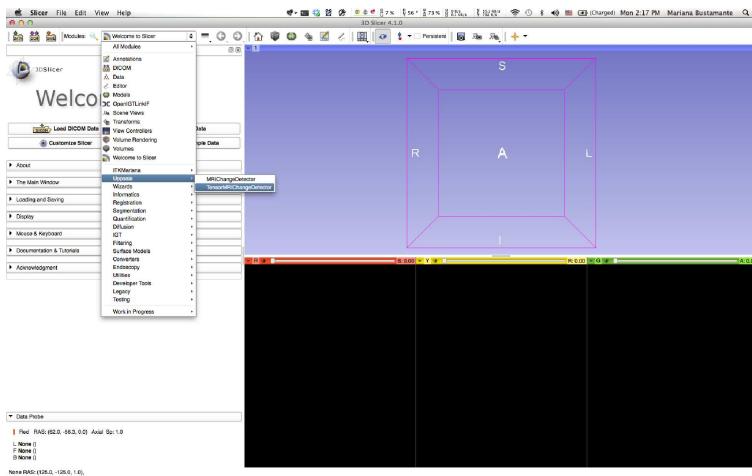


Figure 3.8. Step 0: Module selection

2. The user adds the volumes to be analysed.

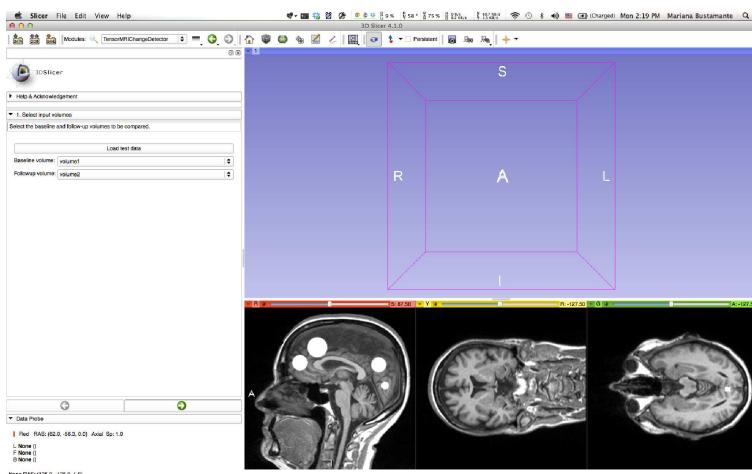


Figure 3.9. Step 1: Adding volumes

3. The user chooses the deformation field smoothing sigma and begins the registration.

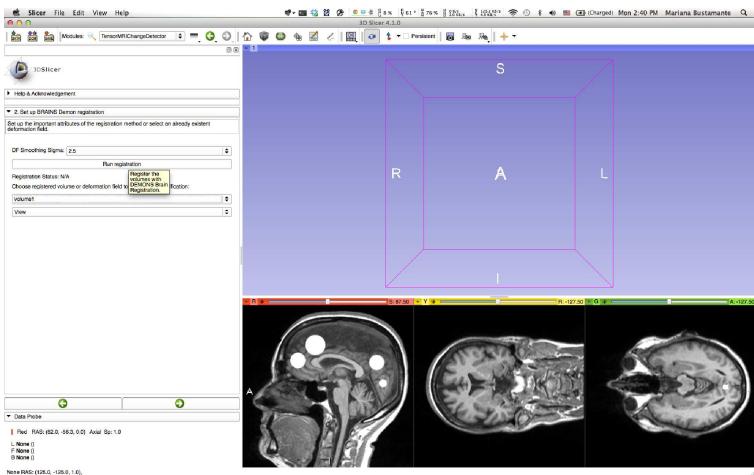


Figure 3.10. Step 2: DEMONS Warp Registration

4. The user chooses the percentage of growth and shrinkage to use in the output label map and clicks the button “Run Tensor measurements” to initiate the tensor calculations.

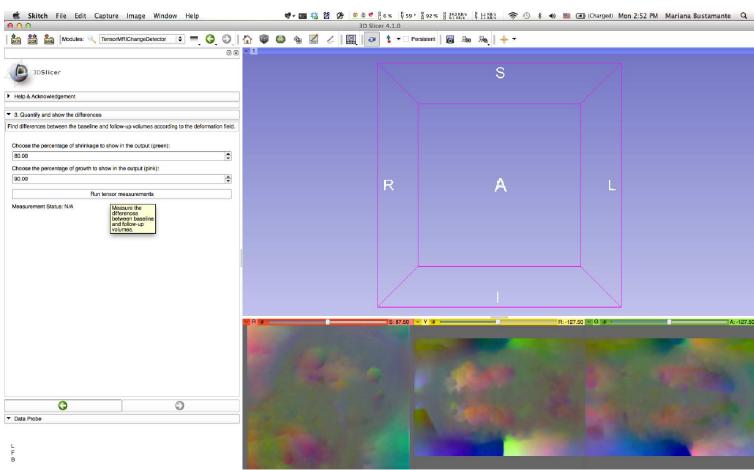


Figure 3.11. Step 3: Running tensor measurements

The volume that can be observed in the image is the deformation field produced by the registration from the previous step.

5. The program shows the resulting label volume directly on top of the original volume.

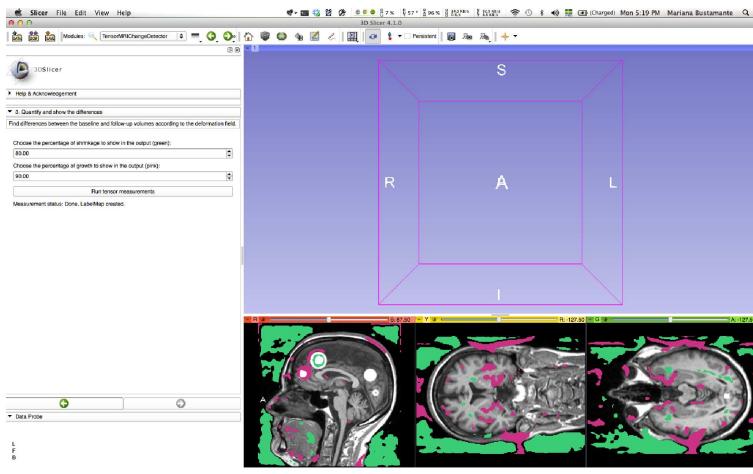


Figure 3.12. Step 4: Quantification result

6. The user can now move the planes and visualise the volumes as desired.

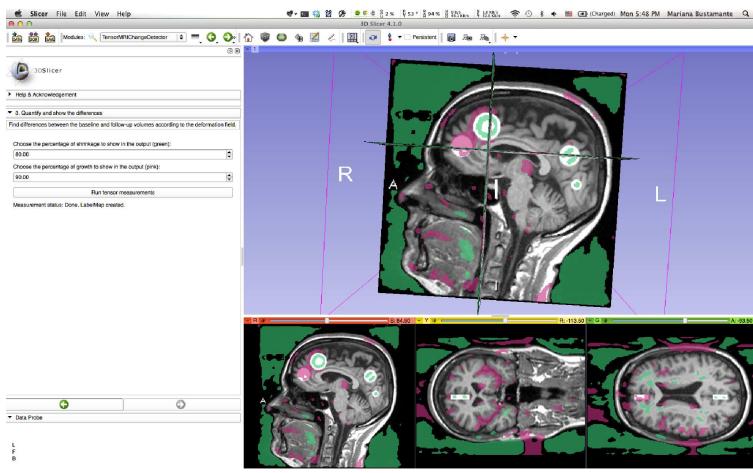


Figure 3.13. Step 5: User visualization

7. The user can change the percentage numbers depending on what he or she wants to see and re-run the module by clicking the button once again.



Figure 3.14. Step 6: Interactive growth/shrinkage values

Technical Details

Like the voxel-based method, the user interface of the module is written in *Python* using some libraries from *Qt* and *CTK*, being *ctkWorkflowWidgetStep* from *CTK* the most important since it allows the creation of a “step-by-step” wizard.

The tensor measurements are written in *C++* using *ITK*. It is the most important part of the method and it includes the following functionalities:

- Computation of the *Jacobian determinant* at each point in the deformation field by using the filter *DisplacementFieldJacobianDeterminantFilter*.
- Initially the Jacobian values are centered on 1.0, this means that when the values are close to 1.0 the change in the volume is nonexistent or small enough to be ignored.
In order to move the “no-change-zone” from 1.0 to 0.0, a *logarithm* filter is applied on the Jacobian values.
- Computation of the *Jacobian determinant* values that will be shown on the final result according to the percentages chosen by the user.
- Creation of a label map with two colors that represent growth and shrinkage only showing the amount of values specified by the user.
- Creation of a volume with all the values of the *Jacobian determinant* for comparison with the previously described label map.

4. Experiments

This section shows the results of the experiments done with the implemented solutions.

The methods were tried with volumes to which artificial differences of distinct sizes and shapes were added; and also with real patient's volumes obtained from the Uppsala University Hospital.

4.1 Experiments Conditions

4.1.1 Artificial Differences

For this experiments, the MRIs used belong to a healthy patient and the time difference between each exam is very short, so it is expected of them to have no changes.

The methods were used on three distinct sizes of difference: large, medium and small. The considered sizes are based on the comparisons done with real patients and the sizes of the differences presented on those cases.

4.1.2 Real Differences

For this experiment, the MRIs from three real patients with known differences were compared using both methods.

Two of them are patients of the study for which this project was started, and so they have very small differences that are hard to detect with the naked eye.

The third patient has larger known differences produced by a medical condition and its subsequent surgery.

4.2 Artificial Differences Results

The initial baseline volume without modifications used for all of the artificial tests can be seen in the following image:



Figure 4.1. Artificial Differences: Baseline volume

4.2.1 Size: Large

Cubes of large size were deleted from the follow-up volume to represent areas of volume loss. The following volume was obtained:



Figure 4.2. Large Differences: Modified Follow-up volume

Voxel-based Method

The result obtained with this method and large differences is pretty good. All the differences are found and easy to see in the result.

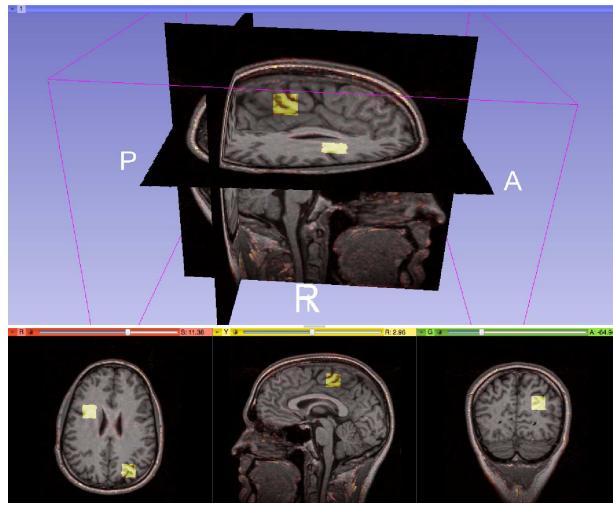


Figure 4.3. Artificial Large: Voxel-base method

Another angle of the result:

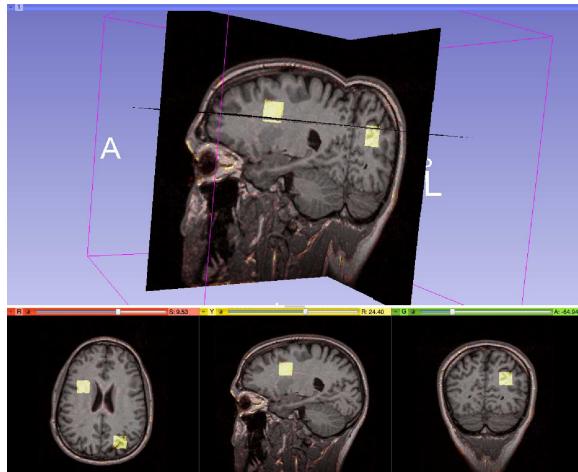


Figure 4.4. Artificial Large: Voxel-base method

Tensor-based Method

In the result obtained with this method the differences are easy to find, but their borders are not properly defined. The method gives the position of the differences but not their exact shape.

The parameters used to obtain this result are:

Deformation field smoothing sigma: 2.5

Shrinkage percentage: 80

Growth percentage: 65

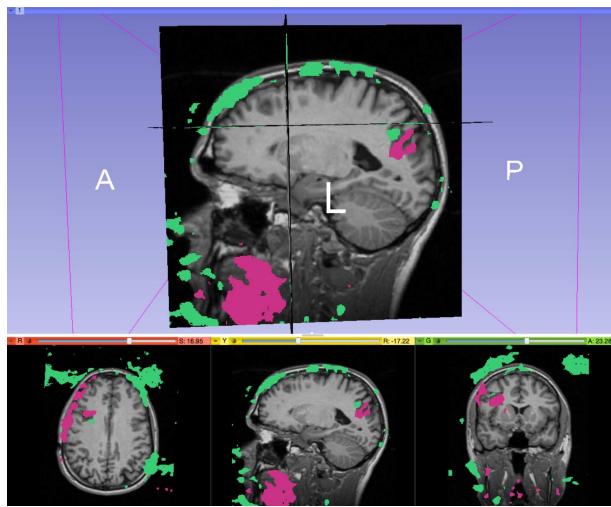


Figure 4.5. Artificial Large: Tensor-base method

Another angle of the result:

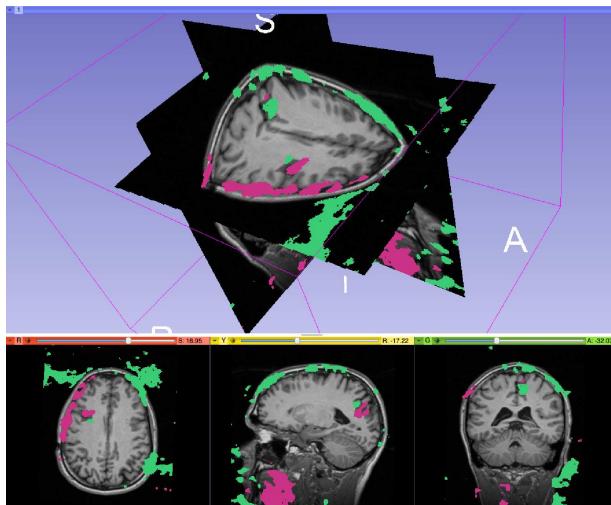


Figure 4.6. Artificial Large: Tensor-base method

4.2.2 Size: Medium

Cubes of medium size were deleted from the follow-up volume to represent areas of volume loss. The following volume was obtained:



Figure 4.7. Medium Differences: Modified Follow-up volume

Voxel-based Method

The result obtained with this method and medium differences is as good as with larger differences. All the deleted volume cubes are found and easy to see in the result.

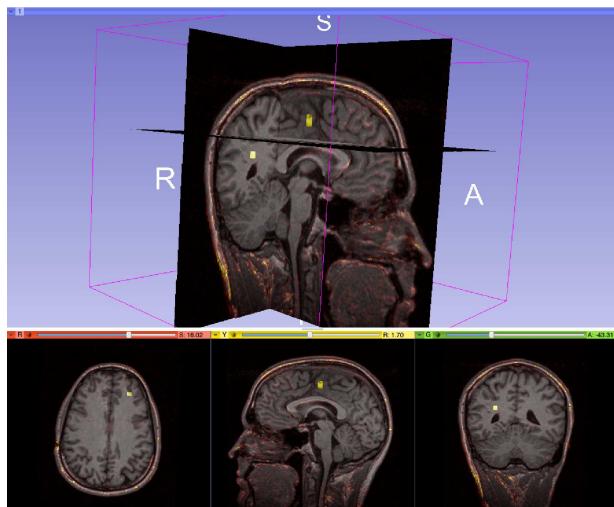


Figure 4.8. Artificial Medium: Voxel-base method

Another angle of the result:

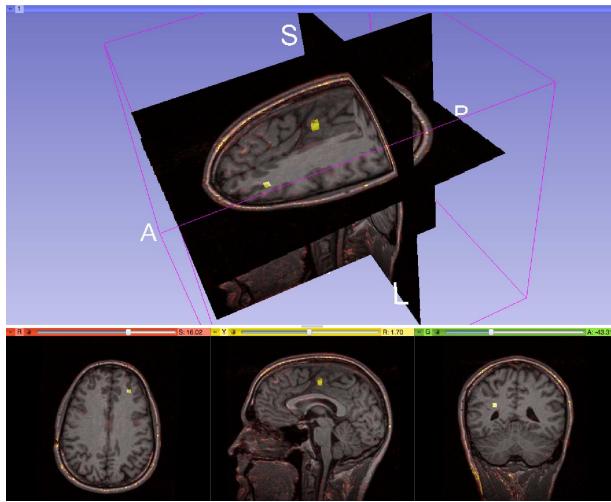


Figure 4.9. Artificial Medium: Voxel-base method

Tensor-based Method

The result is a bit worse than for bigger differences. The volume losses are still found by the application, but their borders are not defined and the method also finds other zones with differences that are hard to differentiate from the added ones.

The parameters used to obtain this result are:

Deformation field smoothing sigma: 2.5

Shrinkage percentage: 80

Growth percentage: 76

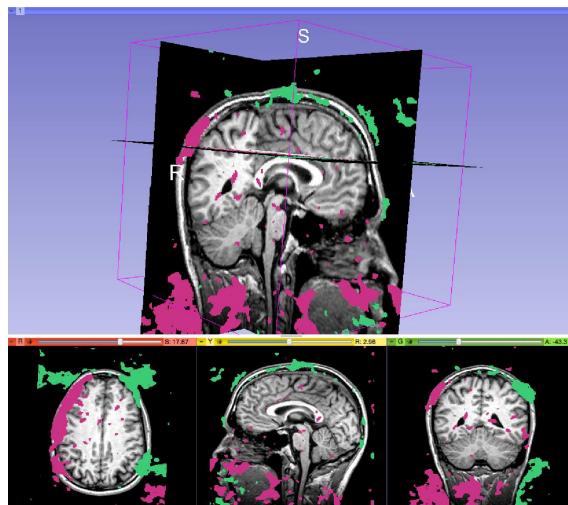


Figure 4.10. Artificial Medium: Tensor-base method

Another angle of the result:

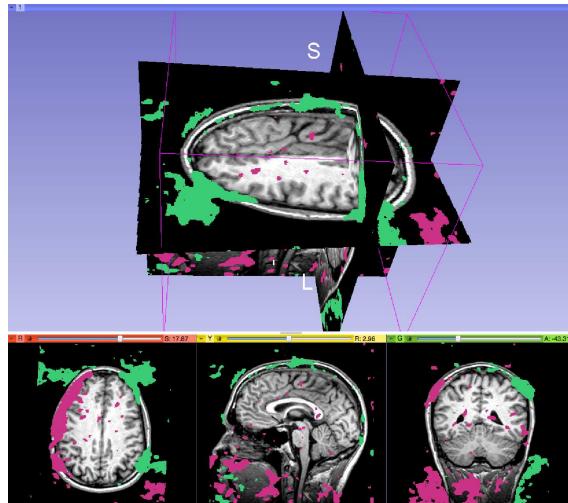


Figure 4.11. Artificial Medium: Tensor-base method

4.2.3 Size: Small

Rectangles of very small size were deleted from the follow-up volume to represent areas of volume loss. Rectangles were selected instead of cubes in this case in order to be able to view the result more simply.

The following volume was obtained. In this image, the small rectangles are marked in red just for the purpose of being more visible in this report. The original image does not include the red circles:



Figure 4.12. Small Differences: Modified Follow-up volume

Voxel-based Method

The result obtained with this method and small differences is also very good. All the deleted volume rectangles are found in the result, although in this case

the differences are so small that could be hard to see, even when they are highlighted in the resulting volume.

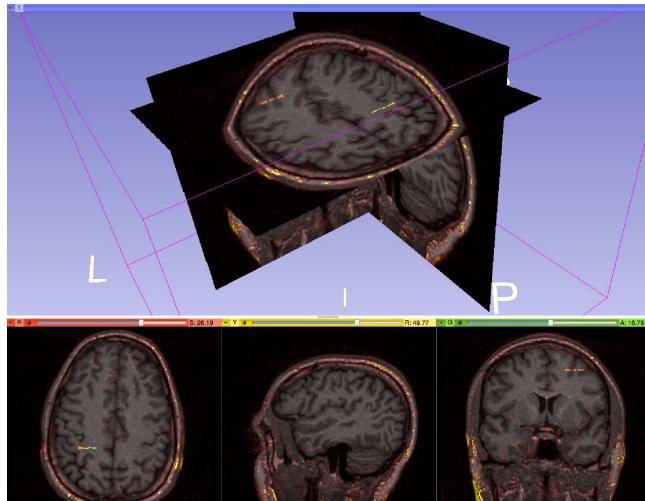


Figure 4.13. Artificial Small: Voxel-base method

Another angle of the result:

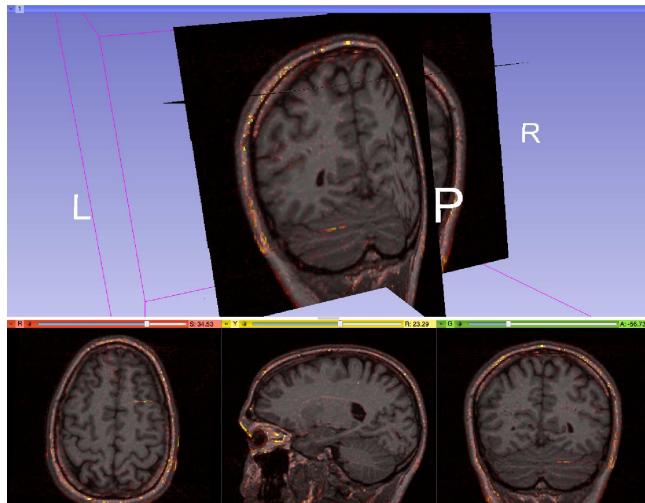


Figure 4.14. Artificial Small: Voxel-base method

Tensor-based Method

The result obtained is not very useful. It may be possible to find some of the differences since we know their position in this experiment. However, this would not be the case with real volumes as it would be impossible to discern between real differences and those produced by imperfections in the method.

The parameters used to obtain this result are:

Deformation field smoothing sigma: 3.5

Shrinkage percentage: 70

Growth percentage: 80

Note that in this case the parameter for deformation field smoothing sigma is higher. This produced a smoother deformation field after the registration, which improved the results to some extent; even though the final result is still not good enough.

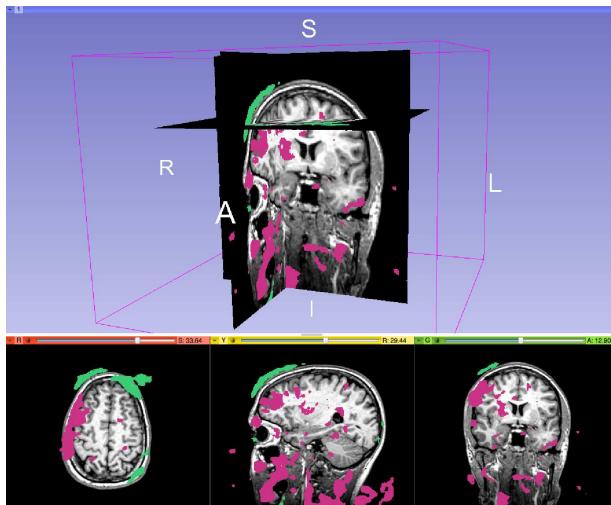


Figure 4.15. Artificial Small: Tensor-base method

Another angle of the result:

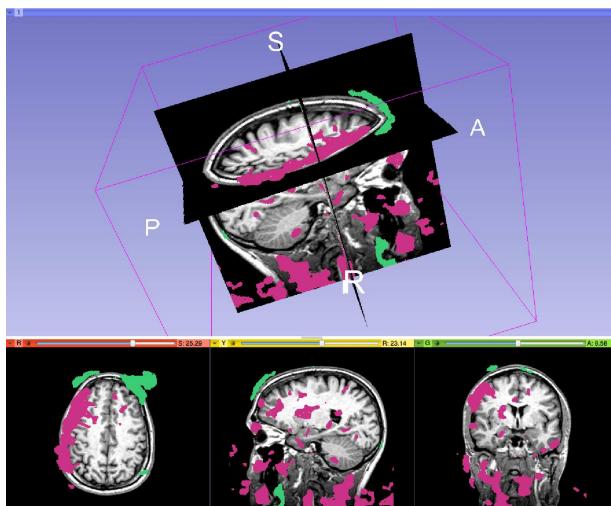


Figure 4.16. Artificial Small: Tensor-base method

4.3 Real Differences Results

4.3.1 Patient 1

This patient presents small differences in the parietal lobe, the differences can be seen as red lines in the voxel-based method's result and pink or green areas in the tensor-based method's result.

Voxel-based Method

The registration method used was *Affine registration*.

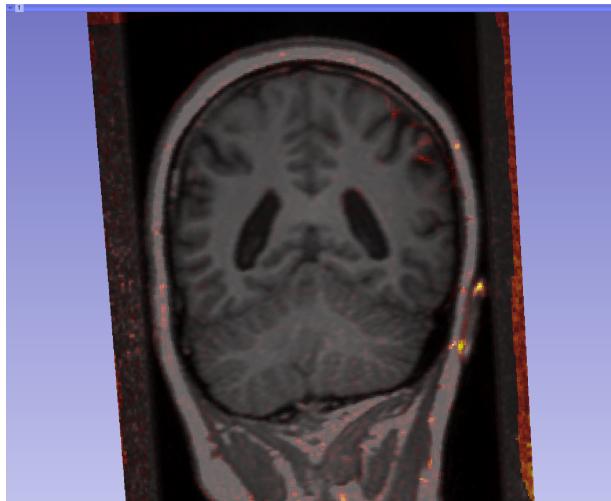


Figure 4.17. Voxel-based method. Patient 1: Coronal plane

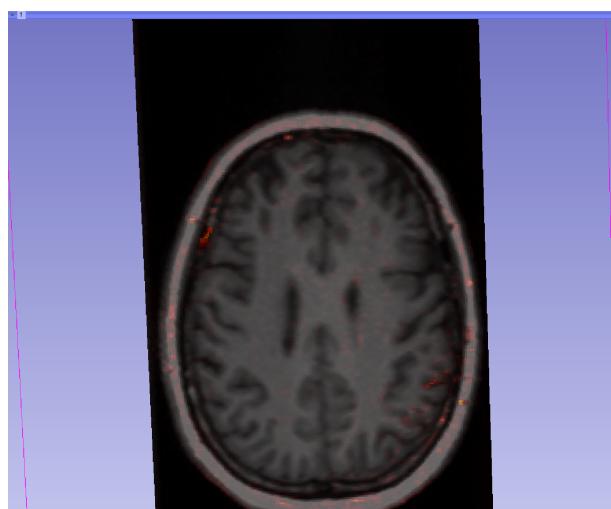


Figure 4.18. Voxel-based method. Patient 1: Traversal plane

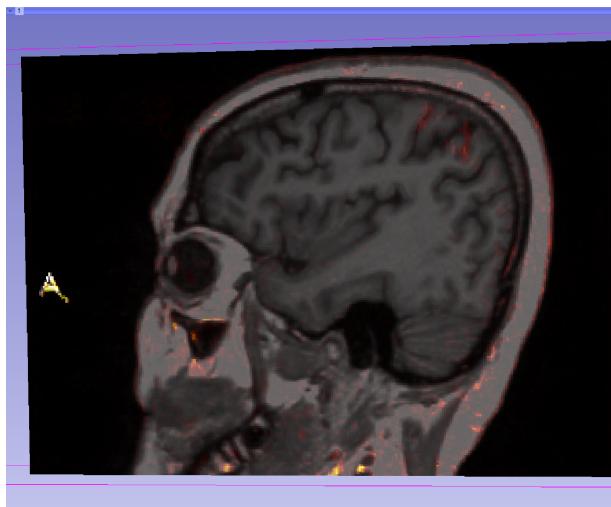


Figure 4.19. Voxel-based method. Patient 1: Sagittal plane

Tensor-based Method

Parameters used:

Deformation field smoothing sigma: 2.5

Shrinkage percentage: 70

Growth percentage: 75

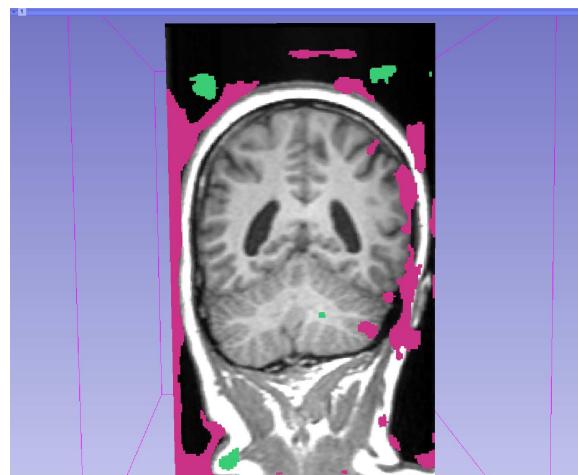


Figure 4.20. Tensor-based method. Patient 1: Coronal plane



Figure 4.21. Tensor-based method. Patient 1: Traversal plane

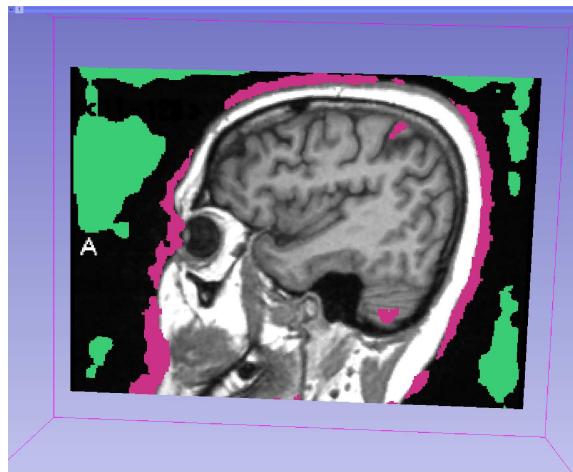


Figure 4.22. Tensor-based method. Patient 1: Sagittal plane

4.3.2 Patient 2

The differences in this patient, if actually present, are really small.

Voxel-based Method

The voxel-based method shows small differences near the corpus callosum on the three planes; these differences, according to the medical expert, might be real because this is a very common area affected by trauma.

The registration method used was *Affine registration*.

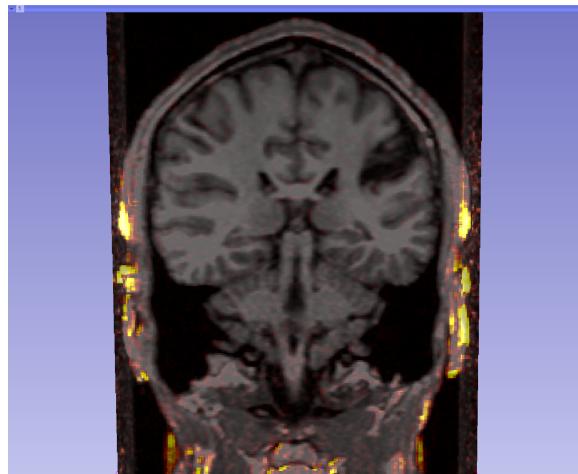


Figure 4.23. Voxel-based method. Patient 2: Coronal plane

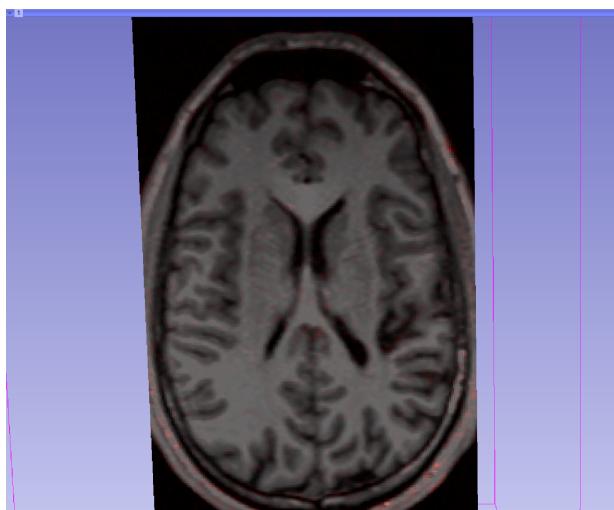


Figure 4.24. Voxel-based method. Patient 2: Traversal plane

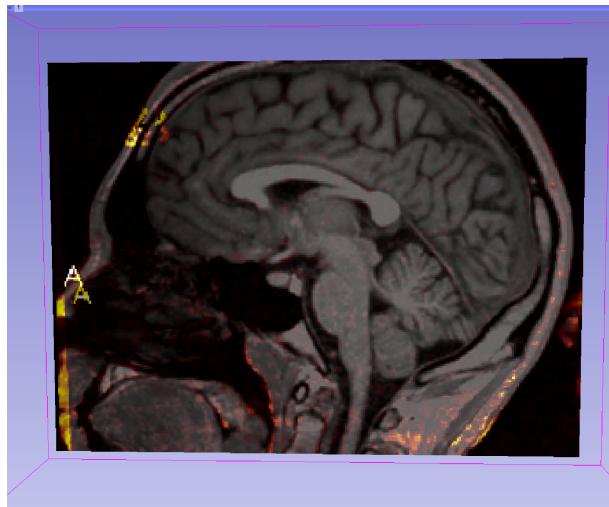


Figure 4.25. Voxel-based method. Patient 2: Sagittal plane

Tensor-based Method

The tensor-base method doesn't find the same differences in the corpus callosum as the previous method. With the usual percentage values (from 70% to 80% for both growth and shrinkage), the method almost doesn't find any differences.

In order to show some of the possible places where the method shows some type of differences, the value of the shrinkage percentage was increased until 88%. Given this, the pink areas shown are not necessarily real differences.

Parameters used:

Deformation field smoothing sigma: 2.5

Shrinkage percentage: 80

Growth percentage: 88

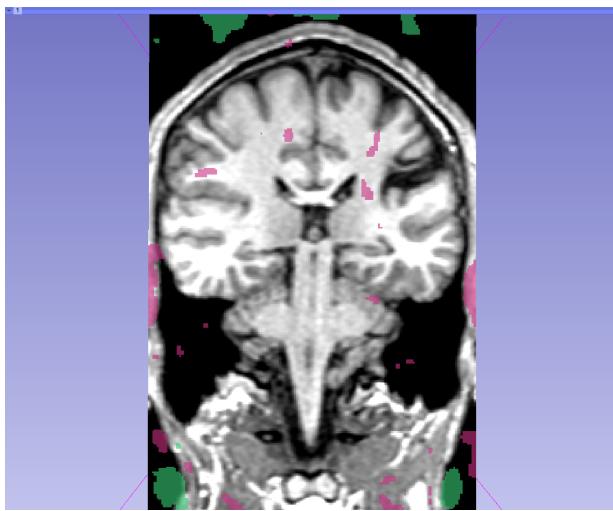


Figure 4.26. Tensor-based method. Patient 2: Coronal plane



Figure 4.27. Tensor-based method. Patient 2: Traversal plane

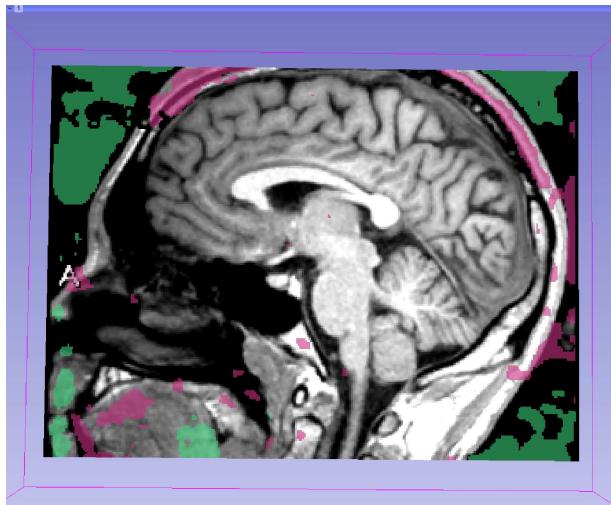


Figure 4.28. Tensor-based method. Patient 2: Sagittal plane

4.3.3 Patient 3

This patient had a medical condition for which it has had two surgeries performed. A tumor, located on the right hemisphere of the frontal lobe, was removed during the first surgery. The MRIs used during this experiment were taken before and after the second surgery, in which a second growth was removed located the same area.

Voxel-based Method

The method shows the expected differences in the right hemisphere of the frontal lobe of the brain. It also shows some size differences especially in the lower parietal lobe, which can be observed in image 4.30.

The registration method used was *Affine registration*.

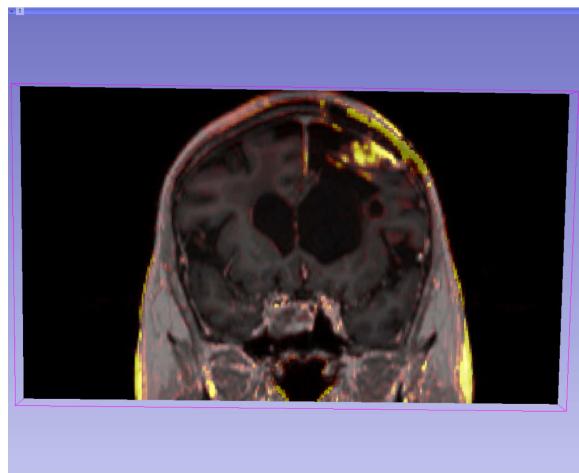


Figure 4.29. Voxel-based method. Patient 3: Coronal plane

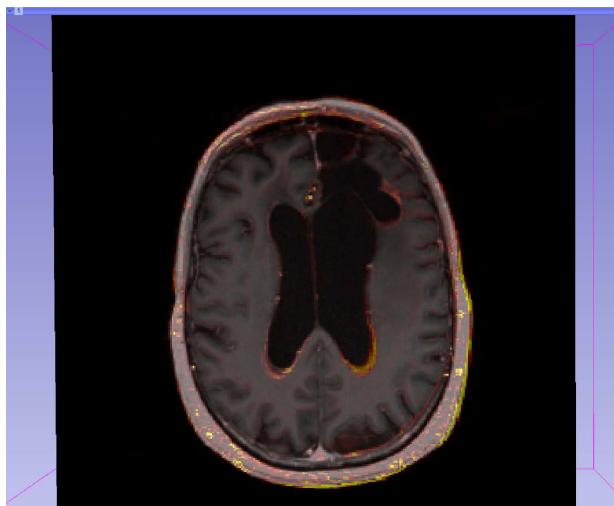


Figure 4.30. Voxel-based method. Patient 3: Traversal plane

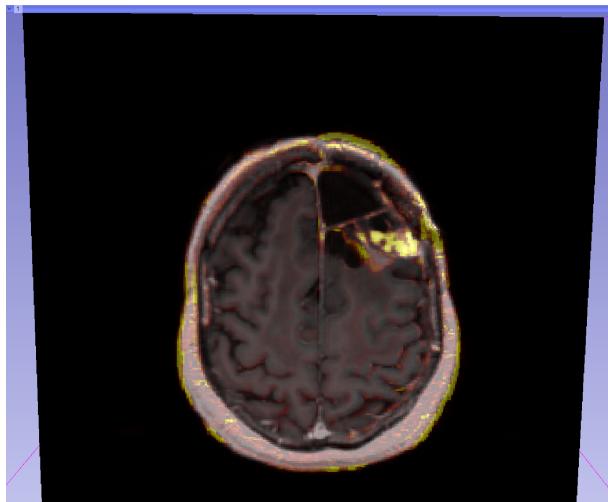


Figure 4.31. Voxel-based method. Patient 3: Upper traversal plane

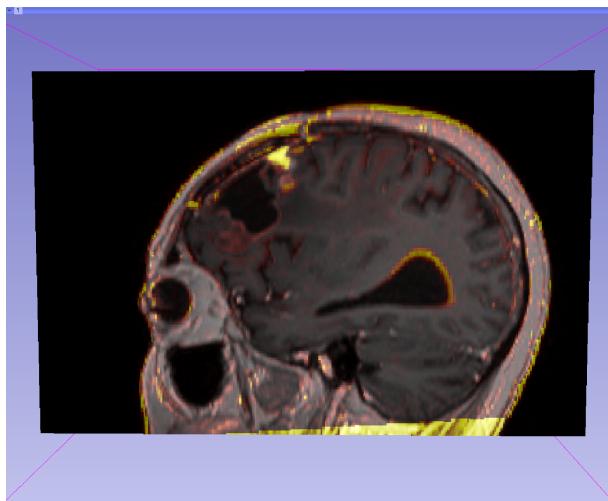


Figure 4.32. Voxel-based method. Patient 3: Sagittal plane

Tensor-based Method

This method also shows the expected difference due to the surgery (correctly expressed as shrinkage, in green) which can be seen in the images 4.33, 4.35 and 4.36.

The image 4.34 is showed for comparison with the differences shown in image 4.30 in the previous method. The tensor-based result doesn't show the exact same result; however, some growth (in pink) can be seen. This could be attributed to either inaccuracy of the method or to movement in the brain

consistent with the surgery.

Parameters used:

Deformation field smoothing sigma: 2.5

Shrinkage percentage: 60

Growth percentage: 50

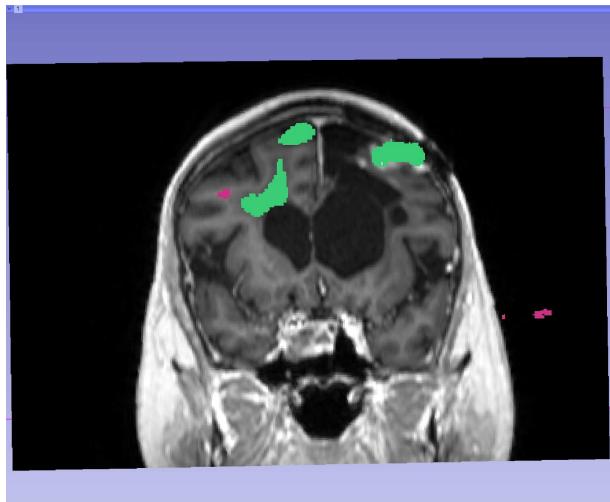


Figure 4.33. Tensor-based method. Patient 3: Coronal plane

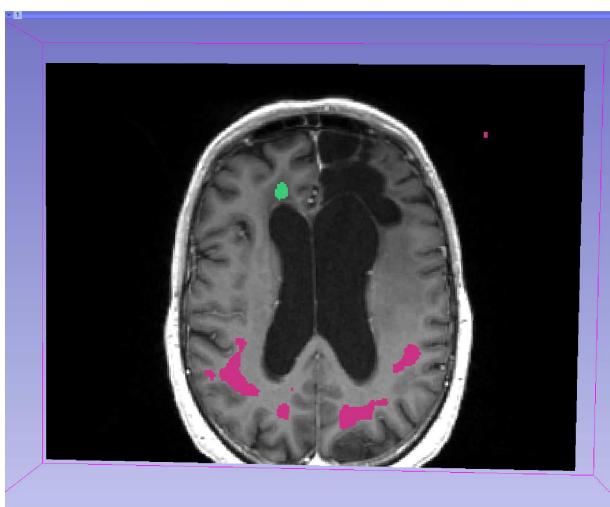


Figure 4.34. Tensor-based method. Patient 3: Traversal plane

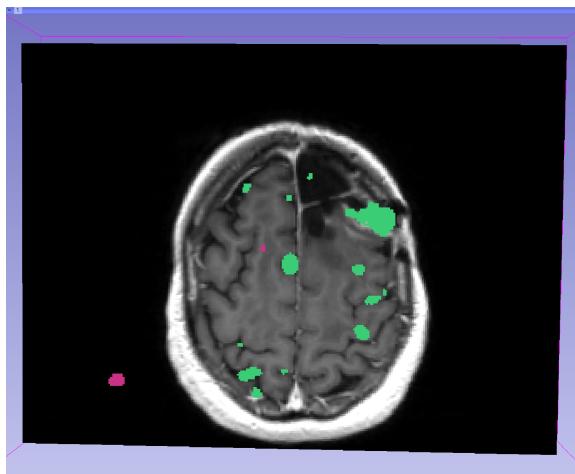


Figure 4.35. Tensor-based method. Patient 3: Upper traversal plane

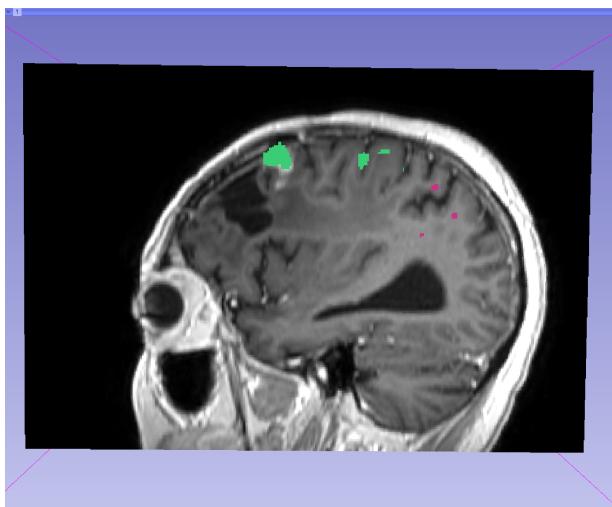


Figure 4.36. Tensor-based method. Patient 3: Sagittal plane

5. Discussion

This chapter contains an analysis of the strengths and weaknesses of each of the proposed methods based on the results obtained after several experiments with patient containing real and artificially added differences.

5.1 Voxel-based method

This method performs especially well in cases of volume loss, since this condition implies larger differences in intensity between both volumes. According to the experiments performed, the size of these differences can be really small and the method may still produce useful results.

The method depends a lot on the registration results obtained in the second step. If the registration result is poor, the program will produce “ghosts” or false differences that may confuse the user.

Note that a poor registration result might not necessarily be a direct consequence of the registration method chosen, it may also be due to problems with the volumes; for example, if the patient’s position changes a lot from one volume to the next, or if the MRI machine has very distinct settings in each examination.

Since in our specific case we deal with differences that are quite small, sometimes it is still hard to see them in the results; specially if we are looking at a screenshot of the application, rather than the actual application where we can see all of the frames in the volume.

5.2 Tensor-based method

This method has a lot of potential; since in theory, by using the deformation field resulting from the registration, we have all the information to be able to identify all the differences between the volumes.

In spite of its potential, further analysis of the method results in the following challenges:

1. We need to generate a “useful” deformation field; which in this case means a deformation field that is not too noisy, and from which we can obtain real differences.

2. We also need to figure out “good values” for the percentages of growth and shrinkage that we are going to use for the final result. This point is particularly complicated, since the *Jacobian determinant* values are usually not distributed in the same way for different volumes.

The module attempts to help the user find the right values by making the most relevant numbers interactive in the user interface. However, it is sometimes very hard to distinguish between actual differences and noise caused by a non smooth deformation field.

Also, the *Jacobian determinant* values are not regularly distributed around zero; this means that in most cases, the range of the values representing growth (located above zero), is much larger than the range of values that represent shrinkage (values below zero). As a consequence, the percentage of shrinkage is much more sensitive to small changes than the percentage of growth.

6. Applications

The original idea for the project came out of the necessity for a program that could be used to find small differences between MRIs of trauma patients at the Uppsala University Hospital.

The main doctor interested in the project is PhD. Raili Raininko, from the Department of Radiology at the hospital, who collaborated with her experience and comments during the entire course of this project.

So far, the application has been used in a study made with the collaboration of the Department of Neuroscience of Uppsala University and the Department of Radiology of the Uppsala University Hospital.

During the mentioned study, the MRIs of nineteen patients who presented mild traumatic brain injuries were compared; with the first MRI taken 2 or 3 days after the injury, and the second 3 to 7 months after.

The application created during this project was used to corroborate the results obtained after visual analysis of the MRIs by an expert physician.

The study concludes that loss of brain volume may be a feasible marker of brain pathology after mild traumatic brain injuries.

Volume comparison, and specifically medical image comparison, continues to be a very challenging problem. The applications developed during this project should be well received by the medical audience, even if they do not guarantee a complete automatization of the comparison process.

As described later on the *Future Works* section, it would be very useful to make the modules implemented available to the public by making them a part of a new release of *3D Slicer*.

In this way, the modules could be used, commented on, and even possibly improved, by anyone who would be willing to use them or by a member of the *3D Slicer* developer community.

7. Conclusions

The goal of detecting and locating small differences between images and volumes presents a very interesting problem, not only limited to the medical field. It is still a big challenge today for which a complete solution is required.

Even though in our specific case we deal with patients that have received mild trauma, it can still mean a lot for the quality of life of a person if he or she receives help when needed, and if the physician is able to locate the exact area affected by the injury.

The applications presented in this project were created as modules of a bigger and already existent application called *3D Slicer*, which contains many other functionalities for medical image analysis and manipulation.

In this way, the applications are not only given an interface that is easy to use, but also allow the user to utilize any of the other modules already present within *3D Slicer*.

The applications attempt to solve the problem for MRI volumes with two different techniques, both obtained from analysis of previous research done in the area.

The results obtained with the voxel-based method are quite good, even when the artificial differences added were too small to be found with the naked eye. During this research, this method was successfully used in a medical study with real patients performed by the Uppsala University Hospital.

The tensor-based method also produced useful results for differences of large and medium size. Even though the results were not as expected for smaller differences, the method could be the start for a very interesting future development possibly using the *Jacobian matrix*.

Finally, it is important to highlight that the results of both methods as they are now, are just a suggestion of the size and location of the differences between the volumes; as a consequence, the procedure can not be completely automatic, since the applications still need some human help in order to define the differences more exactly and possibly rule out errors.

7.1 Future works

The tensor-based method needs to be improved in order to obtain more exact results, and for it to be able to detect smaller differences.

The first thing that should be done is either find a better way of figuring out which range of values of the *Jacobian determinant* are actually useful to detect differences, or directly using the *Jacobian matrix* to calculate the changes in volume between the MRIs.

Manipulating the *Jacobian matrix* is harder than using the *Jacobian determinant* since, according to my research, there is no *ITK* function or library already implemented to use it. Some theoretical pointers on how to use the matrix can be found in [3].

The voxel-based method seems to work quite well in our experiments with both real and artificial differences between volumes. However, it would be very interesting to be able to compare its results against another tool with the same goals.

We would like the modules to be available for all the users of *3D Slicer* and for the public in general. To achieve this goal, a few specific conditions must be fulfilled in order for the module to be accepted as part of the *3D Slicer* code.

This would probably produce some criticism from the *3D Slicer* developer community, which could contain useful comments on the current implementation of the modules and how to improve it.

References

- [1] B.A. Ardekani, J. Nierenberg, M.J. Hoptman, D.C. Javitt, and K.O. Lim. Mri study of white matter diffusion anisotropy in schizophrenia. *Neuroreport*, 16:947–23, 2003.
- [2] John Ashburner and Karl J. Friston. Voxel-based morphometry—the methods. *NeuroImage*, 11:805–821, 2000.
- [3] John Ashburner, Karl J. Friston, and W. Penny. *Human Brain Function*. Section Editors, 12 Queen Square, London, UK, 2 edition.
- [4] J.V. Haynal, N. Saeed, A. Oatridge, E.J. Williams, I.R. Young, and G. Bydder. Detection of subtle brain changes using subvoxel registration and subtraction of serial mr images. *Journal of Computer Assisted Tomography*, 19(5), 1995.
- [5] M. Holden, J.A. Schnabel, and D.L. Hill. Quantification of small cerebral ventricular volume changes in treated growth hormone patients using nonrigid registration. *IEEE Trans Med Imaging*, 21:1292–301, 2002.
- [6] D.K. Jones, L.D. Griffin, D.C. Alexander, M. Catani, M.A. Horsfield, R. Howard, and S.C. Williams. Spatial normalization and averaging of diffusion tensor mri data sets. *Neuroimage*, 17:592–617, 2002.
- [7] Kitware. Itk, 2012. [Online; accessed 15-September-2012].
- [8] Ignacio Larrabide, Pedro Omedas, Yves Martelli, Xavier Planes, Maarten Nieber, Juan Moya, Constantine Butakoff, Rafael Sebastián, Oscar Camara, Mathieu De Craene, Bart Bijnens, and Alejandro Frangi. Gimias: An open source framework for efficient development of research tools and clinical prototypes. In Nicholas Ayache, Hervé Delingette, and Maxime Sermesant, editors, *Functional Imaging and Modeling of the Heart*, Lecture Notes in Computer Science, pages 417–426. Springer Berlin / Heidelberg, 2009.
- [9] L. Lemieux, U.C. Wiesmann, N.F. Moran, D.R. Fish, and S.D. Shorvon. The detection and significance of subtle changes in mixed-signal brain lesions by serial mri scan matching and spatial normalization. *Medical Image Analysis*, 3(2), 1998.
- [10] Julia Patriarche and Bradley Erickson. A review of the automated detection of change in serial imaging studies of the brain. *Journal of Digital Imaging*, 17(3):158–174, Sep 2004.
- [11] K.M. Pohl, E. Konukoglu, S. Novellas, N. Ayache, A. Fedorov, I-F. Talos, A. Golby, W.M. Wells III, R. Kikinis, and P.M. Black. A new metric for detecting change in slowly evolving brain tumors: Validation in meningioma patients. *NeuroSurgery*, 68:225–33, 2011.
- [12] D. Rey, G. Subsol, H. Delingette, and N. Ayache. Automatic detection and segmentation of evolving processes in 3d medical images: Application to multiple sclerosis. *Medical Image Analysis*, 6:163–179, 2002.
- [13] M. Svensen, F. Kruggel, and H. Benali. Ica of fmri group study data. *Neuroimage*, 16:551–563, 2002.

- [14] J.-P Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.
- [15] Wikipedia. 3dslicer, 2012. [Online; accessed 15-September-2012].
- [16] Wikipedia. Affine transformation, 2012. [Online; accessed 25-September-2012].
- [17] Wikipedia. B-spline, 2012. [Online; accessed 25-September-2012].
- [18] Wikipedia. Brain morphometry, 2012. [Online; accessed 10-September-2012].
- [19] Wikipedia. Jacobian matrix and determinant, 2012. [Online; accessed 26-September-2012].
- [20] Wikipedia. Matlab, 2012. [Online; accessed 15-September-2012].
- [21] T.A. Zeffiro, G.F. Eden, R.P. Woods, and J.W. VanMeter. Intersubject analysis of fmri data using spatial normalization. *Adv Exp Med Biol*, 413:235–40, 1997.
- [22] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.