

# **Automatic Construction of Statistical Children Brain Atlas with MRI**

Project ANIM

Christian Wachinger, Patric Somlo  
Celine Hudelot, Isabelle Bloch

December 19, 2005

# Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Project Objectives</b>	<b>3</b>
2.1. Groupwise Atlas Approach . . . . .	3
2.2. Pairwise Atlas Approach . . . . .	4
2.3. Choice of our Model . . . . .	5
2.4. Global Overview of our Approach . . . . .	5
<b>3. Pre-Processing</b>	<b>5</b>
3.1. Image Resizing . . . . .	8
3.2. Segmentation . . . . .	8
3.2.1. Method for Segmentation . . . . .	8
3.2.2. Evaluation of the Segmentation . . . . .	8
3.3. Centering and View correction . . . . .	8
<b>4. Registration</b>	<b>9</b>
4.1. Affine Registration . . . . .	10
4.2. Non-rigid Registration . . . . .	10
4.3. Software Environment . . . . .	11
<b>5. Atlas Construction</b>	<b>12</b>
5.1. Construction of an average intensity atlas . . . . .	13
5.2. Calculation of the Average Transformation . . . . .	13
5.2.1. Parameterized Linear Model . . . . .	15
5.2.2. Applying PCA and Occurred Problem . . . . .	15
5.2.3. Alternative Approach . . . . .	16
5.3. Applying the Average Transformation on the Average Intensity Atlas . .	17
<b>6. Conclusion</b>	<b>17</b>
<b>A. Results of Transformation</b>	<b>19</b>
<b>B. How To Use</b>	<b>21</b>
<b>C. Scripts</b>	<b>22</b>
<b>D. Parameter files</b>	<b>26</b>
<b>E. MATLAB files</b>	<b>27</b>

# 1. Introduction

Image atlases have a high importance in medical image treatment especially atlases of the human brain. They are used for the automatic segmentation of brain structures, which is important in the analysis of the brain anatomy and brain morphometry. Normally this segmentation is done manually by trained experts but it is too time consuming and susceptible to intra- and inter-rater subjectivity. The fully automated segmentation and labelling can be achieved with atlas-based methods, which use warping techniques to calculate the spatial transformation between the atlas model and the subject brain. The crucial part to get high quality segmentation is to have a good atlas.

The construction of the statistical atlas was realised within the scope of a project for the class "ANIM" (Analysis of Images) at the Ecole Normale Supérieure de Télécommunication (ENST) Paris.

We would like to thank Celine Hudelot for introducing us in the problem of non-rigid registration and for taking that much time to support us with our various problems. We would also thank Prof. Isabelle Bloch.

## 2. Project Objectives

The task of this project was to study and propose a general method for the construction of a statistical atlas for the human brain based on Magnetic Resonance Imaging (MRI). In particular, this method should be applied to build an atlas for children. In the scope of the project for ANIM we will apply this method to the head of children having an age between 8 and 12. The magnetic resonance images that are used to build this atlas were taken by the university hospital in Paris and the CHU of Nimes. The construction of the atlas, bases on the non-rigid registration between the different subjects. The non-rigid registration utilizes a global transformation (affine) and a local transformation (free form). Due to the limited duration of our project we decided to integrate existing implementations as far as possible.

There are several different approaches for the construction of the brain atlases. In two of them we were interested in detail. Firstly a groupwise approach that is described in [1] and secondly a pairwise approach that is proposed in [2]. Further information about the two different methods is given in the following two sections.

### 2.1. Groupwise Atlas Approach

A simultaneous or groupwise registration will be done on all subjects to a yet unknown reference that represents the average shape of the population (Figure 1). To find the average shape (reference image) the sum of all the deformations has to be zero while maximizing the similarity of all images. This approach has the advantage that no reference image has to be chosen in advance that normally strongly affects the atlas. For the registration the algorithm described in section 4 is used with the additional condition

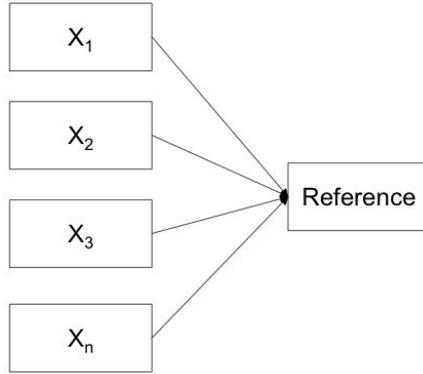


Figure 1: registration on all images to a average shape image

that the sum of all deformations has to be equal to zero.

$$\sum_{i=1}^n d_i(x) = 0 \quad \forall x \in \Sigma$$

To measure the similarity of the  $n$  images a  $n$ -dimensional histogram, which for large numbers of images would become computationally infeasible, can be used. Instead an arbitrary image is selected as an intensity (but not an anatomical) reference  $X_{ir}$ . The similarity to the intensity image can be calculated

$$S(X_{ir}, X) = \frac{H(X_{ir}) + H(X)}{H(X_{ir}, X)}$$

where  $H(X_{ir})$  represents the marginal entropy of intensity reference,  $H(X)$  represents the marginal entropy of the combined set of images and  $H(X_{ir}, X)$  denotes their joint entropy.

The constrained optimization problem

$$\max S(d) \quad \text{s.t.} \quad \sum_{i=1}^n d_i(x) = 0$$

is solved by an iterative Rosen's Gradient Projection Method [3]. Which is comparable to the method of steepest descent for unconditional strained optimization.

## 2.2. Pairwise Atlas Approach

In the pairwise atlas approach one reference image has to be chosen. Performing pairwise registrations between each image and the reference image, the average intensity image and also the average transformation using statistical methods like the Principal Component Analysis (PCA) are calculated. Applying the average transformation to the average images results in the atlas. Further information can be found in section 5.

### 2.3. Choice of our Model

To use the groupwise method for atlas construction the Rueckert's non-rigid registration toolkit has to be modified to implement the side condition  $\sum_{i=1}^n d_i(x) = 0$  in the optimization process. Since we only possess binaries for non-rigid registration this approach would be out of the scope of this project.

The statistical approach is more convenient because we can use the existing non-rigid transformation toolkit and implement the statistical analysis on our own. The interface between the two systems is composed by the output deformation files. For this reason we chose to implement the latter approach.

### 2.4. Global Overview of our Approach

In this paragraph we want to give a global overview of the construction of the atlas that can also be seen in Figure 2. Each step is later described in more detail. First of all, a pre-processing has to be applied to the different images. This is necessary to firstly standardize the image and also to facilitate the later calculation and therefore to achieve better results. In a second step the registration between the segmented brains has to be done. The registration is split up in an affine and a non-rigid part. The affine part is firstly calculated and is more or less a mapping from one image into the coordinate system of the other image. So, in the affine part there is no information concerning the inter brain variability. The later coming non-rigid registration is in contrast used to model the variability between the brains. Therefore this part is also the interesting one for the succeeding statistical analysis.

After having calculated the transformation from each image to the reference image and also the inverse transformation from the reference image to all the others the atlas can be constructed. The first transformation applied to the images is used to build the image with the averaged intensity. The inverse transformation serves as input for statistical model to build the average transformation. Concretely, it is used as input for the PCA which makes up the core of the statistical model. Having the average intensity image and also the average transformation the atlas is easily constructed by applying it to the image. The process, beginning at the registrations and ending up at the atlas can be seen in Figure 3.

## 3. Pre-Processing

Before registering the images a pre-treatment has to be applied to them to achieve good results and also to exclude errors in the following steps of the atlas construction. The problem when making the implementation not completely by yourself, but instead using already existing software solutions, is that you do not know in detail what is exactly going on. So, in order to exclude possible sources of errors, the images should be standardized as much as possible.

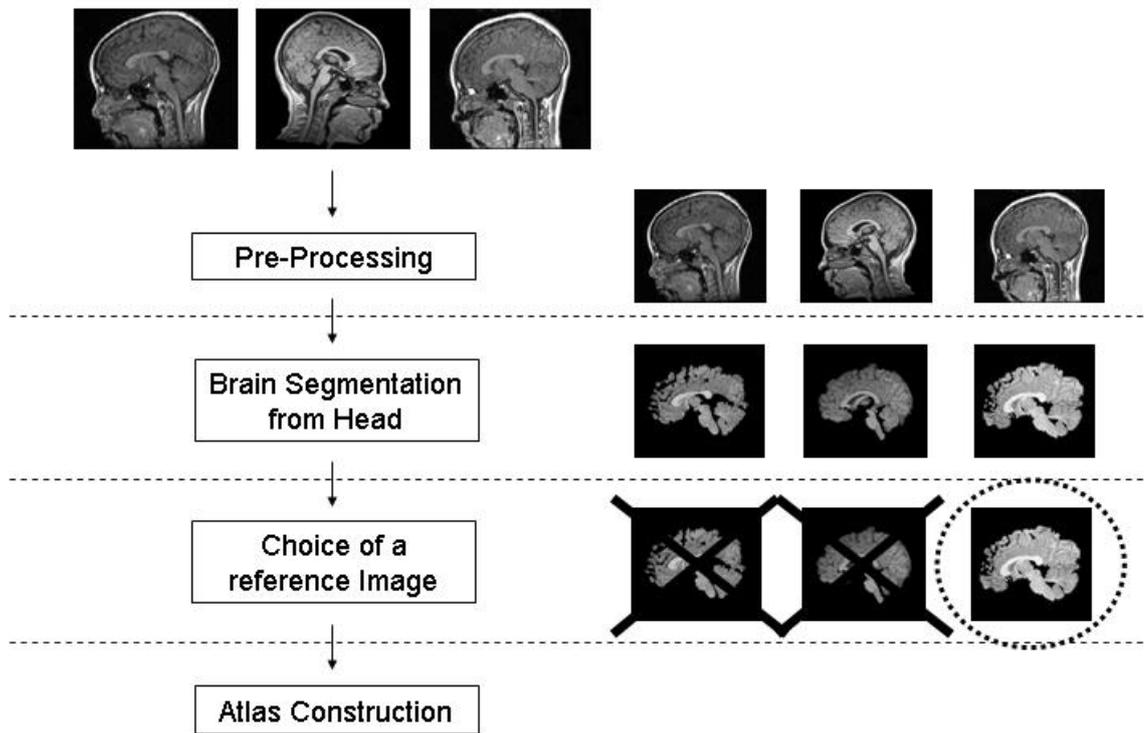


Figure 2: Overview of our Approach

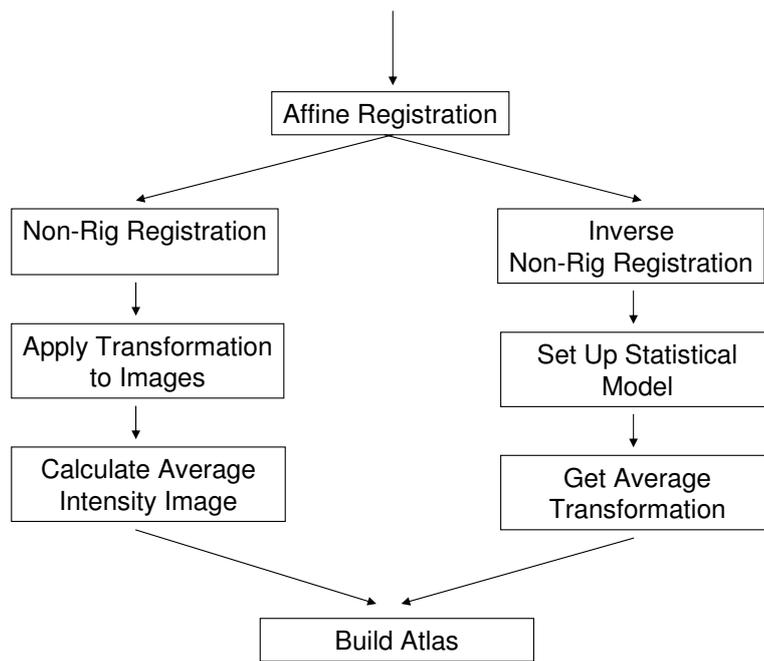


Figure 3: Process to Calculate the Atlas

### **3.1. Image Resizing**

First of all the volumes should have the same size. For our images we were enlarging them to  $256*256*256$ . The resizing of each image can be done with the "t\_resize" command of TIVOLI but we were using Matlab. The images were read in with the help of a dim2mat function that was written by the ENST. Unfortunately the file was just made to read in images with two dimensions. Our extension of the function to three dimensions can be found in the appendices. In Matlab was a matrix with the target size created and the original image was embedded. Afterwards the matrix was written back to the dim-format with mat2dim function that we also had to extend to three dimensions and is also listed in the appendices.

### **3.2. Segmentation**

In a second step the brains were segmented out of the head images. This was done because in the case of our project, we have limited our work to the construction of a brain atlas. The construction of a head atlas is even more complex because the surrounding bony tissue negatively influences the registration. Moreover, the heads are cut on different positions so that a registration would have to be done on data that not even shows the same information. But is also clear that some information will be lost by the segmentation, as although there are excellent algorithms for the brain segmentation, the exact extraction of the brain is not possible.

#### **3.2.1. Method for Segmentation**

The methods that we were using to extract the brain are the "t\_brain" from the TIVOLI library and "VipGetBrain" from Anatomist/BrainVISA. In the t\_brain function a threshold value for the segmentation has to be specified. The VipGetBrain method works more automatic but also there exist several parameters to fine tune the output.

#### **3.2.2. Evaluation of the Segmentation**

The segmentation of the brain using the above described is an iterative process, where the user has to constantly evaluate the results of the segmentation and repeat it until the best performance is achieved. In our case we were doing the evaluation visually with Brainvisa. The program offers a very good possibility to visualise the original image and the calculated segmentation so that an evaluation can be easily done. Examples of a good and a bad segmentation, visulaised with Brainvisa, can be seen in Figure 4 and 5.

### **3.3. Centering and View correction**

When segmenting out the brains one has the unpleasant result that it is not centered but almost touches the upper bound. We had some problems that the brains were cut off if the images were not centered. It is sufficient to center the reference image because after the affine registration all the other images are also centered. Moreover the view of

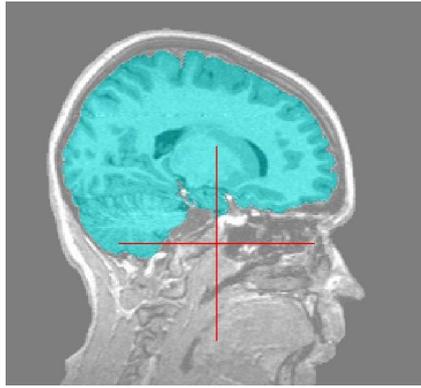


Figure 4: Good Segmentation of the Brain

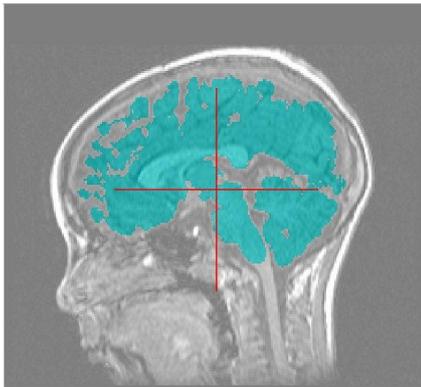


Figure 5: Bad Segmentation of the Brain

the heads in the images has to be standardized. In our case some images were turned 180 along the z axis so that they were looking in exactly opposite directions (see Figure 4 and 5). The affine registration was not able to correct this problem since the overlay of the both images was just too well and so the registration got stuck in a local minimum.

Normally, the centering and view correction should be possible with the functions "rview" and "areg" of the ITK, that is described in section 4.3. When using "rview" it is possible to modify the affine transformation and simultaneously visualize it. So it is easy to find the right transformation for the view correction. The saved transformation parameters can also be manually edited. This is necessary for the centering as in the GUI the translation value is restricted. After finding the right transformation it has to be applied to the image with the "areg" command of ITK.

## 4. Registration

For the construction of the atlas a non-rigid registration is used because the variability of anatomical structures across different individuals has to be modeled. A usable alignment between different brains is not possible with rigid registrations. The transformation has

to be calculated in two different ways. First, the mapping from each image to the reference image was done. Afterwards the transformations were applied to the images and the intensity atlas was calculated in which all the intensities are averaged. In a second step the transformation from the reference to each image was computed. The transformation files were used as input for the later statistical analysis.

## 4.1. Affine Registration

The affine registration has 12 degrees of freedom and is used before the non-rigid registration is applied. It is used to move overlay two volumes as well as possible by using translation, rotation, scaling and shearing. The affine transformation model has the following form:

$$T_{affine}(x, y, z) = \begin{pmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \Phi_{14} \\ \Phi_{24} \\ \Phi_{34} \end{pmatrix}$$

## 4.2. Non-rigid Registration

The non-rigid registration consists of three different parts: a transformation, a non-linear minimization and a cost function. For the nonlinear minimization the gradient descent method was chosen. The cost function is made up of a voxel-based similarity measure and regularization terms which constraints the transformation to be smooth. For the similarity measure the Normalized Mutual Information (NMI), which are very frequently used, are chosen. The most important part of the registration is the way the transformation is modeled. It is clear that a normal rigid transformation is not sufficient for the registration since an affine transformation does not account for the inter-subject variations. Instead, a combination of a affine transformation (global) and a non-rigid transformation (local) is applied. This can be written as

$$T(x) = T_{affine}(x) + T_{non-rigid}(x)$$

with  $x$  being a point in the anatomy. There exist several approaches to model the local transformation like thinplate-splines and B-splines or some physical model like elastic and fluid registration. In the proposed algorithm the B-splines were chosen. The basic principal is to deform an object by manipulating an underlying mesh of control points. These points resemble a grid with uniform spacing. So after calculating the non-rigid transformation between two subjects one receives a set of 3-D points that describe the performed free form deformation (FFD). The non-rigid transformation model can be written as the 3-D tensor product of the one-dimensional cubic B-splines in the anatomy

$$T_{non-rigid}(x) = \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 B_l(u)B_m(v)B_n(w)c_{i+l,j+m,k+n}$$

with

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

where  $c$  denotes a  $m_x \times m_y \times m_z$  lattice of control points which parameterize the FFD,  $i, j, k$  denote the indexes of the control points and  $u, v, w$  correspond to the relative position of  $x$  in the lattice coordinates.

For the further analysis the rigid part of the transformation is not of interest as it just describes the different position, orientation and overall size of each subject's anatomy. The non-rigid part is the important one as it shows the anatomical variability of the different subjects. Therefore the dependency of the local transformation on the global transformation has to be removed.

### 4.3. Software Environment

Like stated above the idea is to use already existing implementations for the registration. On the one hand because it is a common problem and several researchers have already developed solutions. On the other hand it is not so easy to implement and would fill a whole project on itself. The software environments which calculate the nonrigid registration and on those we were looking on greater detail are the Insight Registration Toolkit, the Image Registration Toolkit (ITK) and the VTK CISC Registration Toolkit. After trying out all of them the VTK and ITK seemed to be most promising. Unfortunately there were some problems with finding the right program on the right operating system that took us some time. Finally, we worked with the ITK under Linux because it showed the best performance and was also the most convenient way to manage the high computational load. The non-registration between two images took more than 1,000 minutes.

The commands in ITK that we were using are "areg" for the affine registration, "nreg" for the non-rigid registration and "rview" for the visualization of the images. We were passing the following parameters to the "nreg" method:

- Input Image
- Input transformation (normally the affine registration)
- Output transformation
- Parameter file
- Mesh spacing

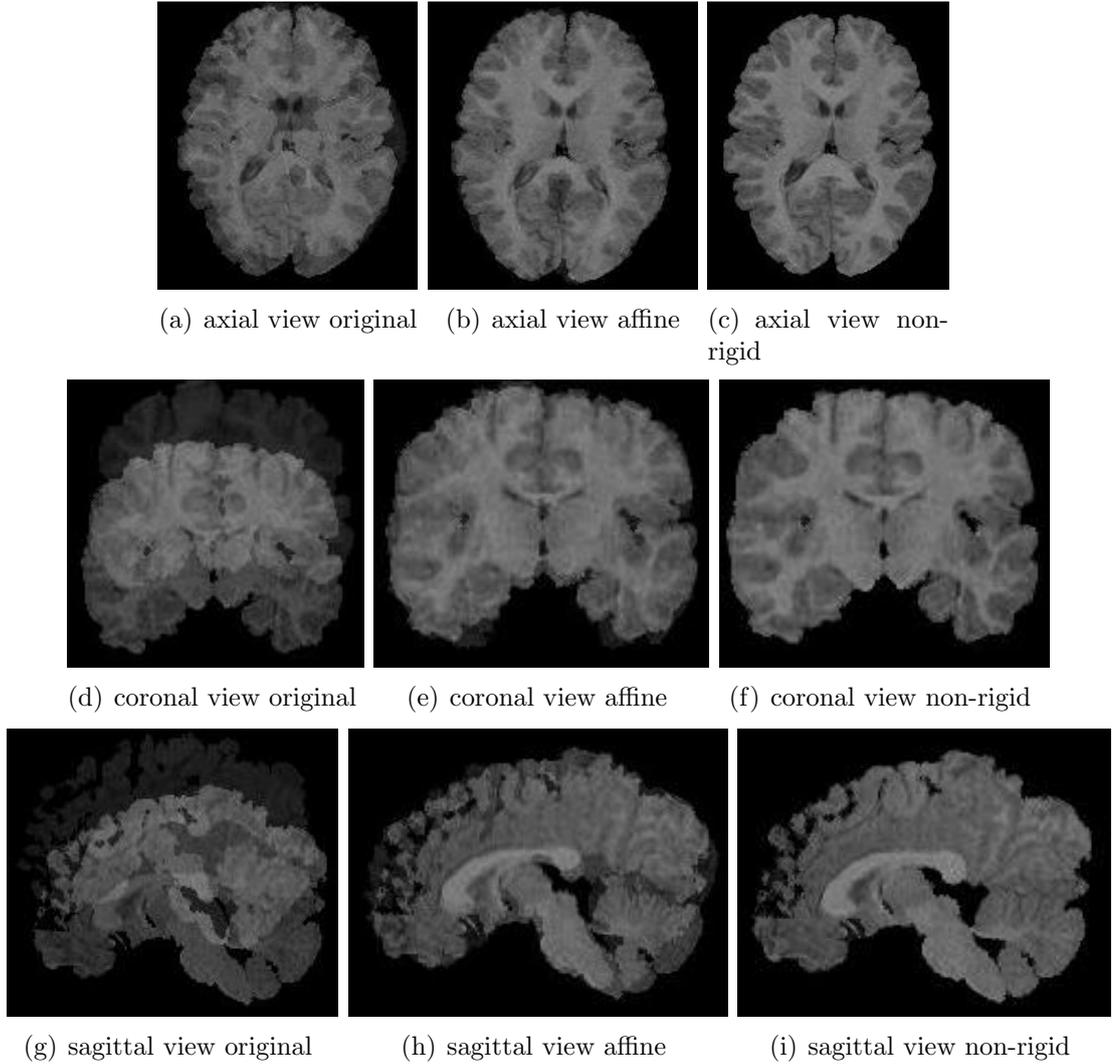


Figure 6: Image  $\mathcal{S}_1$  before affine, after affine and after non-rigid registration compared with the reference image  $\mathcal{S}_r$ .

In order to facilitate the usage of the transformation we wrote several script files in which all the commands are collected. The scripts are listed in Appendix C and the parameter files in Appendix D. In figure 6 the effect of the affine and non-rigid transformation can be observed in a superposition view. The result of the other transformation can be found in the Appendix A

## 5. Atlas Construction

To construct an atlas out of the different affine and non-rigid registrations we use the statistical shape model presented by Rueckert in [2]. The construction of the atlas consists of three parts:

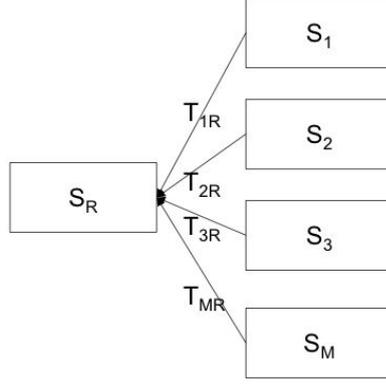


Figure 7: FFD of  $\mathcal{S}_i$  on  $\mathcal{S}_r$

1. construction of an Average Intensity Atlas
2. calculation of the Average Transformation
3. applying the Average Transformation on the Average Intensity Atlas

The three steps, the corresponding problems and our results will be presented in the following sections.

## 5.1. Construction of an average intensity atlas

To construct an average intensity atlas out of our  $m + 1$  images one has to be chosen randomly as our reference image  $\mathcal{S}_r$ . A free form deformation (FFD)  $T_{iR}: \mathcal{S}_i \rightarrow \mathcal{S}_r$  will be applied on all the remaining  $m$  images (see Figure 7).

After the FFD, the average intensity atlas  $\mathcal{S}_{avg}$  between the  $m$  transformed images over all voxels is calculated:

$$\mathcal{S}_{avg} = \frac{1}{m} \sum_{i=1}^m \mathcal{S}_i$$

The intensity atlas of our three images  $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$  each registered to the reference image  $\mathcal{S}_r$  can be seen in Figure 8. Except of some internal structure, the form of the intensity average atlas is very similar to the reference image.

## 5.2. Calculation of the Average Transformation

This section is a key section and describes how to construct out of the many transformations  $T_{Ri}: \mathcal{S}_r \rightarrow \mathcal{S}_i$  the average transformation  $C$  with the help of a parameterized linear model, see section 5.2.1. The transformation  $C$  is applied on the average intensity atlas constructed in section 5.1. Instead of calculating just the mean Rueckert suggests in [2] to use a PCA to build the linear parameterized model.

PCA was originated by Pearson in 1901 [4] and later developed by Hotelling 1933 [5]. Excellent statistical treatment of principal components can also be found in [6, 7].

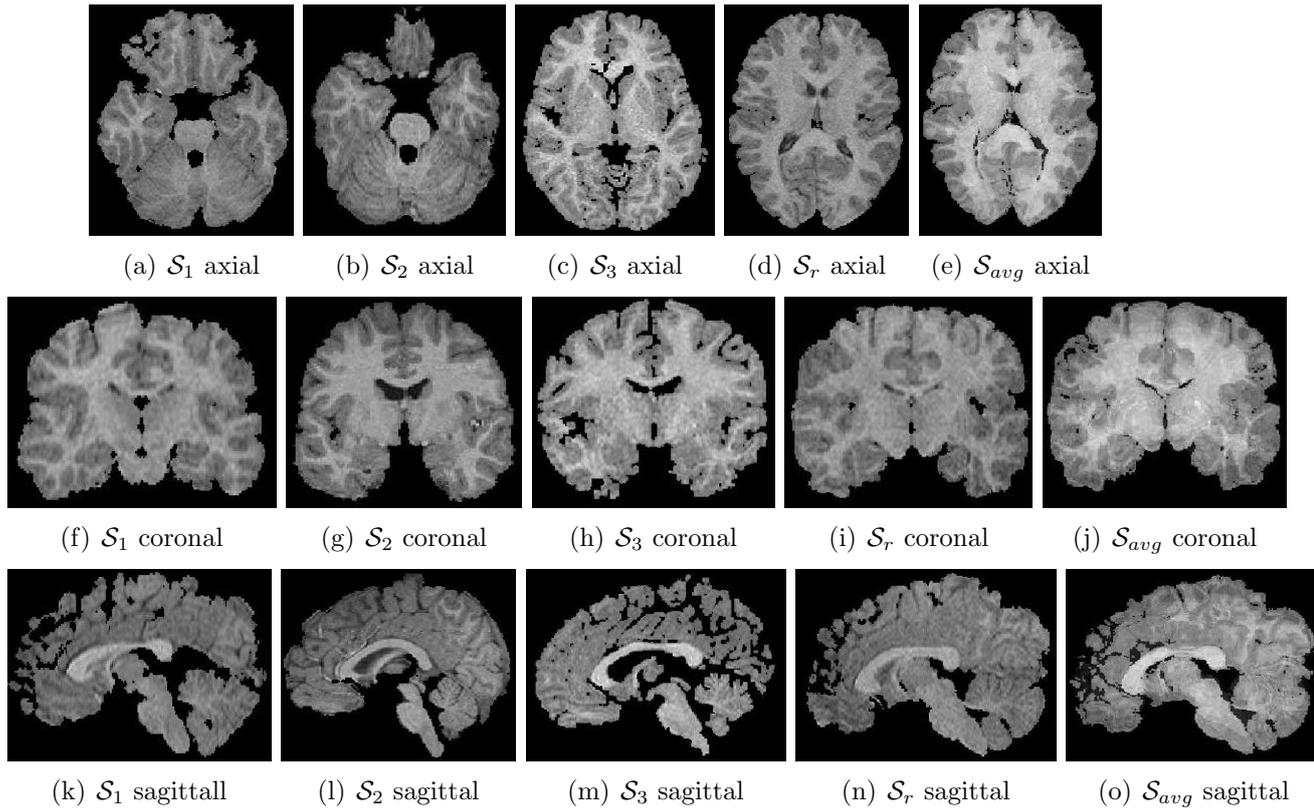


Figure 8: Slices of the original images  $\mathcal{S}_1$ ,  $\mathcal{S}_2$ ,  $\mathcal{S}_3$ ,  $\mathcal{S}_R$  and the calculated atlas  $\mathcal{S}_R$ . This shows that the atlas is very similar to the reference image  $\mathcal{S}_R$ .

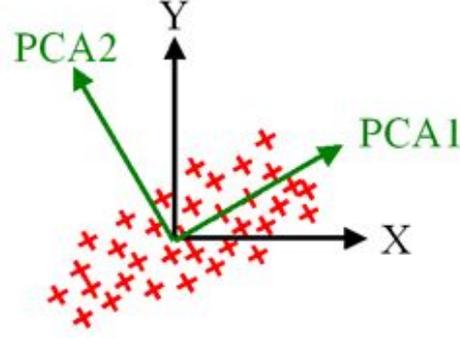


Figure 9: PCA with two principal components

### 5.2.1. Parameterized Linear Model

Supposing that we have  $m$  FFDs described as control point vectors  $C_1, \dots, C_m$ , where the control points  $C_i$  correspond to the transformation  $T_{Ri}$ , mapping from the anatomy to the reference subject  $\mathcal{S}_r$  to the anatomy of all other subjects  $\mathcal{S}_i$ . Each  $C_i$  consists of  $m_x \times m_y \times m_z$  3-D control points. The goal is to approximate the distribution of  $C^*$  using the following parameterized linear model

$$C^* = \hat{C} + \Phi b \quad \text{with} \quad \hat{C} = \frac{1}{m} \sum_{i=1}^m C_i$$

and  $b$  being the model parameter vector. The columns of the matrix  $\Phi$  are formed by the principal components of the covariance matrix  $S$

$$S = \frac{1}{m-1} \sum_{i=1}^m (C_i - \hat{C})(C_i - \hat{C})^T$$

To calculate  $\Phi$  we need the eigenvectors  $\Phi_i$  and corresponding eigenvalues  $\lambda_i$  (sorted by  $\lambda_i \geq \lambda_{i+1}$ ) of  $S$ .  $\Phi$  consists of the  $t$  first eigenvalues  $\Phi = (\Phi_1 | \Phi_2 | \dots | \Phi_t)$ . The idea is to approximate any anatomy within the population group using the model parameter  $b$  which has just a reduced number of dimension. But although this reduction of dimensionality and therefore also loss of data the most significant information is kept as the eigenvectors corresponding to the highest eigenvalues are chosen. The idea of the PCA is shown in Figure 9 where the first eigenvector "PCA1" conserves the maximum variation of the points.

### 5.2.2. Applying PCA and Occurred Problem

For reasons of time constraints we chose to implement the PCA in Matlab. As described in Figure 10 we have  $M = 5$  different transformation  $T_{R1}, T_{R2}, \dots, T_{RM}$  from our reference image  $\mathcal{S}_r$  to the other images  $\mathcal{S}_i$  which contain  $N = 150,000$  control coordinates.

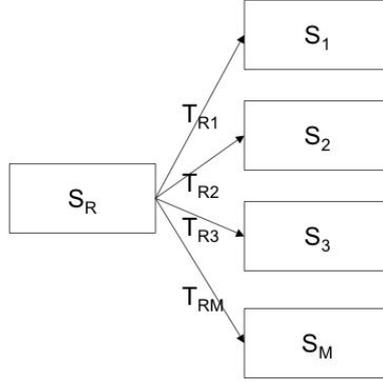


Figure 10: Non-rigid transformation from reference

We used Matlab to read the ASCII version of our free FFD and constructed the Matrix of our 3-D control points  $C$ :

$$C = \begin{pmatrix} T_{R1}(1) & T_{R2}(1) & \cdots & T_{RM}(1) \\ T_{R1}(2) & T_{R2}(2) & \cdots & T_{RM}(2) \\ \vdots & \vdots & \ddots & \vdots \\ T_{R1}(n) & T_{R2}(n) & \cdots & T_{RM}(n) \end{pmatrix}$$

Where  $T_{Rj}(i)$  is the  $i$ th point of our free form transformation from  $S_r$  to  $S_j$ .  $C_i$  is now the  $i$ th column of matrix  $C$ . Unfortunately the covariance matrix  $S$  is too big ( $150,000 \times 150,000$ ) and it cannot be calculated. Therefore the Rueckert approach cannot be done in Matlab and since there is not enough time to implement the PCA in C or C++, alternative solutions have to be considered.

### 5.2.3. Alternative Approach

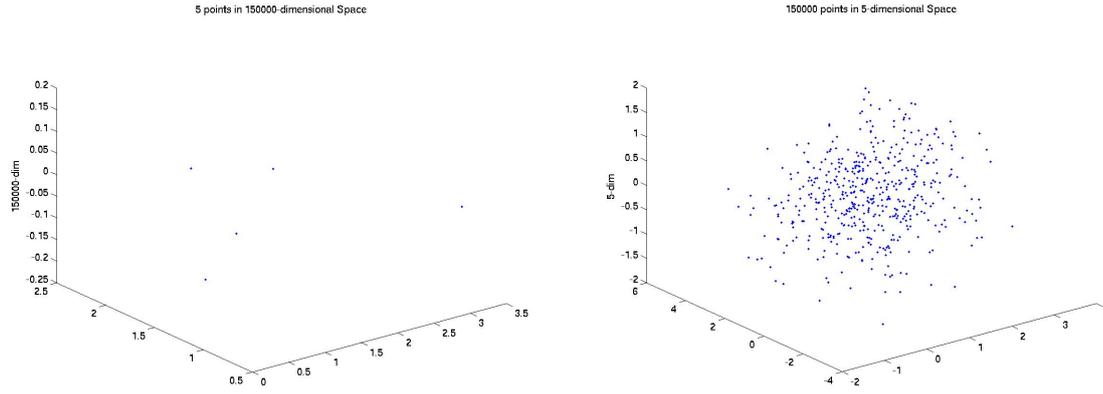
Since we have not enough memory to build the big covariance matrix  $S = \frac{1}{m-1} \sum_{i=1}^m (C_i - \hat{C})(C_i - \hat{C})^T$  we need to look for alternative solutions to construct our medium transformation.

Instead of applying a PCA transformation on  $N$ -dimensional space where each feature vector is a whole transformation we apply the PCA on a much lower  $M$ -dimensional space  $A = C^T$  wich was in similar way done in [8]:

$$A = \begin{pmatrix} T_{R1}(1) & T_{R1}(2) & \cdots & T_{R1}(n) \\ T_{R2}(1) & T_{R2}(2) & \cdots & T_{R2}(n) \\ \vdots & \vdots & \ddots & \vdots \\ T_{RM}(1) & T_{RM}(2) & \cdots & T_{RM}(n) \end{pmatrix}$$

A plot of  $A_i$  and  $C_i$  can be found in Figure 11.

If we do the eigenvalue decomposition of the covariance matrix  $S' = \frac{1}{n-1} \sum_{i=1}^n (A_i - \hat{A})(A_i - \hat{A})^T$  the first principal component  $\Phi'_1$  points into the direction of the maximal



(a) high dimensional space with vector  $C_i$

(b) low dimensional space with vector  $A_i$

Figure 11: high dimensional space plots  $C_i$  for  $i = 1 \dots M$ ; low dimensional space plots  $A_i$  for  $i = 1 \dots N$

variance between our 5 transformations. Since we want to maximize the robustness of the system we will scale  $\Phi_1$  corresponding to its  $L_1$ -norm and compute a weighted average transformation. The algorithm 1 can be used to construct the final transformation.

---

**Algorithm 1** Algorithm to calculate our average transformation

---

- 1:  $S' = \frac{1}{n-1} \sum_{i=1}^n (A_i - \hat{A})(A_i - \hat{A})^T$
  - 2:  $[\Phi, \Lambda] = eig(S')$
  - 3:  $\Phi_1 := \Phi_1 / |\Phi_1|_1$
  - 4:  $A = \{A_1 | A_2 | \dots | A_M\} * \Phi_1$
- 

Instead of using the linear model  $C = \hat{C} + \Phi b$  to map one anatomy to another we can only map the difference weighted averages between the transformations which correspond to a linear combination of the eigenvectors of the small covariance matrix  $S'$ .

### 5.3. Applying the Average Transformation on the Average Intensity Atlas

In a final step we apply our average transformation we calculated with algorithm 1 on our average intensity atlas built in section 5.1. The output is than the atlas of the population.

## 6. Conclusion

Unfortunately we were not able to finally calculate the atlas. The reason was that the data produced by the last registration series was not usable to build the statistical model on it. The vectors  $C_i$  describing the FFD had a different size. Due to the near dead line

of our project it is also not possible to make the calculation once again as the registration for one brain in our population takes about one day. Which means that the calculations for building an atlas from 5 images take approximately 10 days.

The process of atlas construction is a very complex task. We provided some scripts in the appendix to do certain tasks automatically. A human observation of the result is still needed to evaluate e.g. the quality of the segmentation and changes the according parameter to improve the results. Further the user has to deal with various different file types, which have to be converted using either Tivoli or Anatomist. The registration software is not yet stable and returns sometimes mysterious error messages. For this reason it is not yet possible to provide a fully automated program to do the atlas construction like it is proposed in this report and the user has to do the different steps manually.

Nevertheless, the obtained results are very promising. The average intensity atlas contains the inner form of all images in the shape of the reference image. The non-rigid registration led to a very exact superposition of the different images.

To build a more realistic atlas for children more images are needed. Supposing that the image data is given in a standardized format the construction of the atlas is just a question of computational complexity since the framework to build an atlas is now set up. Interesting would also be to examine the difference between the results of the small and large covariance matrix.

The project was very interesting and we learnt a lot about the process of how to build an atlas and the current available software implementations, which support the construction. Also, we got a good insight into non-rigid image registration and the general complexity of this domain.

## A. Results of Transformation

In Figure 12 the image  $\mathcal{S}_2$  and the reference image  $\mathcal{S}_r$  are compared by a difference image. In Figure 13 the image  $\mathcal{S}_3$  and the reference image are compared in a vertical view where the left side is the reference image.

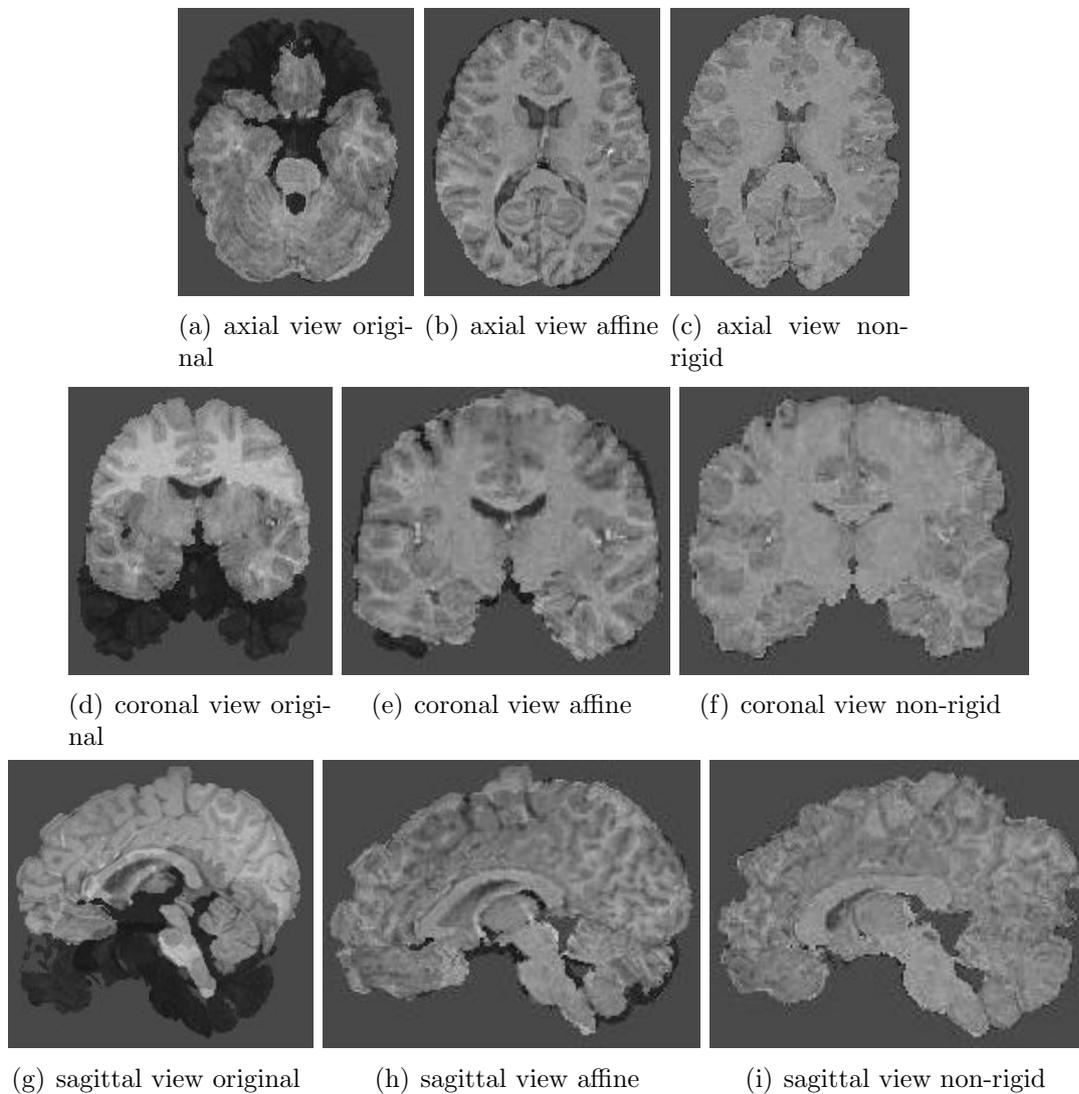
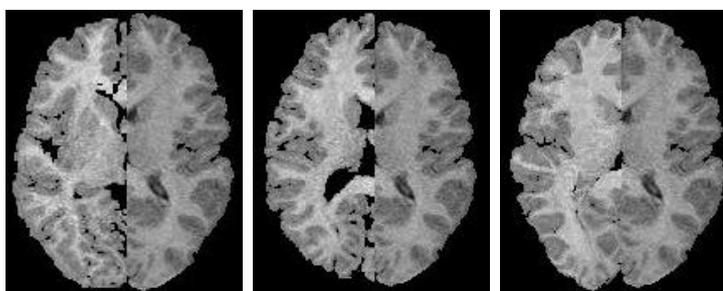
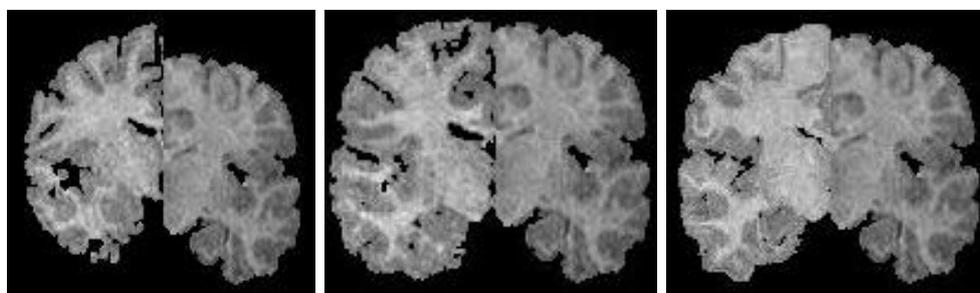


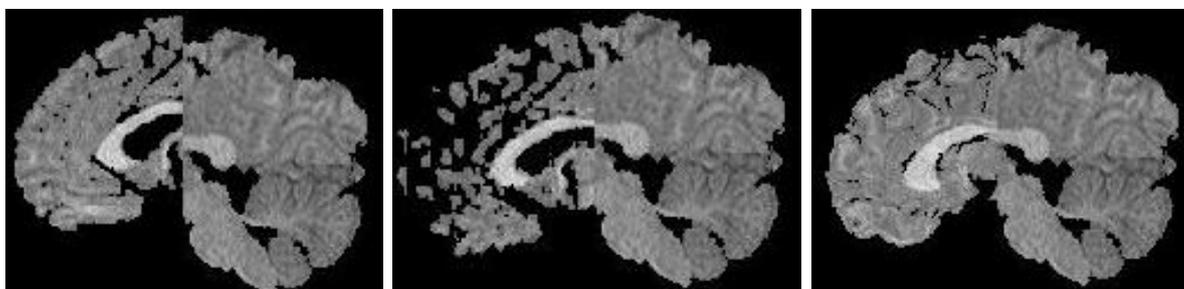
Figure 12: Image  $\mathcal{S}_2$  before affine, after affine and after non-rigid registration in comparison to the reference image  $\mathcal{S}_R$  in difference view



(a) axial view original (b) axial view affine (c) axial view non-rigid



(d) coronal view original (e) coronal view affine (f) coronal view non-rigid



(g) sagittal view original (h) sagittal view affine (i) sagittal view non-rigid

Figure 13: Image  $\mathcal{S}_3$  before affine, after affine and after non-rigid registration in comparison to the reference image  $\mathcal{S}_R$  in vertical view

## B. How To Use

In this section of the appendix we want to give a short explanation how to use the scripts and files that are listed in the following appendices.

1. The calculation of the affine and non-rigid registrations has to started (Listing 2).
2. After completing the affine registrations, the calculated transformation can be applied to the images (Listing 4).
3. Having the affine transformed images, the inverse non-rigid registration can be started (Listing 3).
4. When the non-rigid registration is finished the transformation can be applied to the images (Listing 5).
5. Having the correctly registered images the atlas function can be applied to get the average intensity image (Listing 6).
6. When the inverse non-rigid registration is finished the .dof files that are in a binary format have to converted to ASCII. These files can be read into Matlab and the statistcial can be applied (see Appendix E). The output transformation can then be applied onto the created average intensity image.

## C. Scripts

Listing 1: shell script for Non-Rigid Registration

```
#!/bin/sh
# files have to be processed and oriented in the same
# reference has to be adjusted to reference image not int
# current directory

# constants: can be modified

ref=../PK9/PK_out17.hdr
param_rigid=../param_rigid.txt
param_nonrigid=../param_nonrigid.txt

x1=30
x2=220
y1=30
y2=220
z1=30
z2=220

for file in .//*.hdr
do
let $[ i = 1]

do
echo $file
# affine transformation
areg $ref $file -dofout ${file}_aff.dof -p12 -parameter $param_rigid.txt

# transform image
transformation $file ${file}_aff.hdr -dofin ${file}_aff.dof

# inverse non-rigid transformation
nreg ${file}_aff.hdr $ref -dofout ${file}_inv_non.dof -ds 2.5 -parameter $param_rigid.txt

# non-rigid transformation
nreg $ref ${file}_aff.hdr -dofout ${file}_non.dof -ds 2.5 -parameter $param_rigid.txt

# transform
transformation ${file}_aff.hdr ${file}_non.hdr -dofin ${file}_non.dof -Rx1 $x1 -Ry1 $y1 -Rz1 $z1
rm ${file}_aff.*
let $[ i = i + 1]

done

atlas atlas.hdr $i *.aff.hdr
```

Listing 2: Script for Applying the Non-Rigid Registration

```
# This Script calculates the non-rigid Transformation.
# It maps all images to the reference image PK.
```

```

# Affine Part
areg PK9/PK_cube.hdr MN8A/MN_cube.hdr -dofout dof/P_M_aff.dof
-p12 -parameter param_rigid.txt
areg PK9/PK_cube.hdr GA8/GA_cube.hdr -dofout dof/P_G_aff.dof
-p12 -parameter param_rigid.txt
areg PK9/PK_cube.hdr CM12/CM12_out23_cube_trans.hdr
-dofout dof/P_C_aff.dof
-p9 -parameter param_rigid.txt
areg PK9/PK_cube.hdr AB10/AB_out17_cube.hdr -dofout dof/P_A_aff.dof
-p12 -parameter param_rigid.txt

# Non-Rigid Part:
#
# Region of interest:
x1=30 x2=220
y1=30 y2=220
z1=30 z2=220

nreg PK9/PK_cube.hdr MN8A/MN_cube.hdr -dofin dof/P_M_aff.dof
-dofout dof/P_M_non.dof -ds 5 -parameter param_nonrigid.txt
-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

nreg PK9/PK_cube.hdr GA8/GA_cube.hdr -dofin dof/P_G_aff.dof
-dofout dof/P_G_non.dof -ds 5 -parameter param_nonrigid.txt
-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

nreg PK9/PK_cube.hdr CM12/CM12_out23_cube_trans.hdr -dofin dof/P_C_aff.dof
-dofout dof/P_C_non.dof -ds 5 -parameter param_nonrigid.txt
-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

nreg PK9/PK_cube.hdr AB10/AB_out17_cube.hdr -dofin dof/P_A_aff.dof
-dofout dof/P_A_non.dof -ds 5 -parameter param_nonrigid.txt
-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

```

Listing 3: Script for Applying the Non-Rigid Registration in the Inverse Direction

```

# This Script calculates the inverse non-rigid Transformation from the
# reference image PK to all other images.
# The affine transformatin between the images was already calculated
# so that the affine transfomated images can be used as input.

# Non-Rigid Part:
#
# Region of interest:
x1=30 x2=220
y1=30 y2=220
z1=30 z2=220

nreg MN8A/MN_cube_aff.hdr PK9/PK_cube.hdr -dofout dofinv/M_P_non.dof
-ds 5 -parameter param_nonrigid.txt

```

```

-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

nreg GA8/GA_cube.aff.hdr PK9/PK_cube.hdr -dofout dofinv/G_P_non.dof
-ds 5 -parameter param_nonrigid.txt
-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

nreg CM12/CM_cube.aff.hdr PK9/PK_cube.hdr -dofout dofinv/C_P_non.dof
-ds 5 -parameter param_nonrigid.txt
-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

nreg AB10/AB_cube.aff.hdr PK9/PK_cube.hdr -dofout dofinv/A_P_non.dof
-ds 5 -parameter param_nonrigid.txt
-Tx1 $x1 -Tx2 $x2 -Ty1 $y1 -Ty2 $y2 -Tz1 $z1 -Tz2 $z2

```

Listing 4: Apply the calculated Affine Transformations to the Images

```

# script to transform images with the calculated non-rigid transformation

transformation MN8A/MN_cube.hdr MN8A/MN_cube.aff.hdr
-dofin dof/P_M.aff.dof
transformation GA8/GA_cube.hdr GA8/GA_cube.aff.hdr -dofin dof/P_G.aff.dof
transformation CM12/CM12_out23_cube_trans.hdr CM12/CM_cube.aff.hdr
-dofin dof/P_C.aff.dof
transformation AB10/AB_out17_cube.hdr AB10/AB_cube.aff.hdr
-dofin dof/P_A.aff.dof

```

Listing 5: Apply the calculated Non-Rigid Transformations to the Images

```

# script to transform images with the calculated non-rigid transformation

# Non-Rigid:
#
# Region of Interest:
x1=30 x2=220
y1=30 y2=220
z1=30 z2=220

transformation MN8A/MN_cube.hdr MN8A/MN_cube_non.hdr -dofin
dof/P_M_non.dof -Rx1 $x1 -Rx2 $x2 -Ry1 $y1 -Ry2 $y2 -Rz1 $z1 -Rz2 $z2

transformation GA8/GA_cube.hdr GA8/GA_cube_non.hdr -dofin dof/P_G_non.dof
-Rx1 $x1 -Rx2 $x2 -Ry1 $y1 -Ry2 $y2 -Rz1 $z1 -Rz2 $z2

transformation CM12/CM12_out23_cube_trans.hdr CM12/CM_cube_non.hdr -dofin
dof/P_C_non.dof -Rx1 $x1 -Rx2 $x2 -Ry1 $y1 -Ry2 $y2 -Rz1 $z1 -Rz2 $z2

transformation AB10/AB_out17_cube.hdr AB10/AB_cube_non.hdr -dofin
dof/P_A_non.dof -Rx1 $x1 -Rx2 $x2 -Ry1 $y1 -Ry2 $y2 -Rz1 $z1 -Rz2 $z2

```

Listing 6: Construction of Intensity Atlas

```
# script to transform images with the calculated non-rigid transformation  
  
atlas atlas/atlas.hdr 4 AB10/AB_cube_non.hdr CM12/CM_cube_non.hdr  
GAS/GA_cube_non.hdr MN8A/MN_cube_non.hdr
```

## D. Parameter files

Listing 7: Parameter Non-Rigid

```
Target blurring (in mm) = 0
Target resolution (in mm) = 0
#
# Source image parameters
#
Source blurring (in mm) = 0
Source resolution (in mm) = 0
#
# Registration parameters
#
No. of resolution levels = 3
No. of bins = 64
No. of iterations = 20
No. of steps = 4
Length of steps = 5
Similarity measure = NMI
Lambda = 0
```

Listing 8: Parameter Non-Rigid

```
Target blurring (in mm) = 0
Target resolution (in mm) = 0
#
# Source image parameters
#
Source blurring (in mm) = 0
Source resolution (in mm) = 0
#
# Registration parameters
#
No. of resolution levels = 1
No. of bins = 64
No. of iterations = 10
No. of steps = 4
Length of steps = 20
Similarity measure = NMI
Lambda = 0
```

## E. MATLAB files

Listing 9: modified PCA

```
function imout=pca(imMatrix,n)
% performs a principal component analysis with the
%'n' principal component of the imMatrix matrix.
%
% input: n*m matrix
% output:
% imout: matrix of 'n' principal component images
% information: number of information in principal component
%
[N,M]=size(imMatrix);
mu=mean(imMatrix);
%sigma=std(imMatrix)
for i=1:M
    imnormalized(:,i)=(imMatrix(:,i)-mu(i));
end;
[V,D]=eig(cov(imnormalized))
% backtransformation vector
Vprime=V(:,(M-n+1):M);
% normalisation to 1
Vprime=Vprime./sum(Vprime)
% transform
imout=(imMatrix*Vprime)';
```

## List of Figures

1.	registration on all images to a average shape image . . . . .	4
2.	Overview of our Approach . . . . .	6
3.	Process to Calculate the Atlas . . . . .	7
4.	Good Segmentation of the Brain . . . . .	9
5.	Bad Segmentation of the Brain . . . . .	9
6.	Image $\mathcal{S}_1$ before affine, after affine and after non-rigid registration compared with the reference image $\mathcal{S}_r$ . . . . .	12
7.	FFD of $\mathcal{S}_i$ on $\mathcal{S}_r$ . . . . .	13
8.	Slices of the original images $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_R$ and the calculated atlas $\mathcal{S}_R$ . This shows that the atlas is very similar to the reference image $\mathcal{S}_R$ . . . . .	14
9.	PCA with two principal components . . . . .	15
10.	Non-rigid transformation from reference . . . . .	16
11.	high dimensional space plots $C_i$ for $i = 1 \cdots M$ ; low dimensional space plots $A_i$ for $i = 1 \cdots N$ . . . . .	17
12.	Image $\mathcal{S}_2$ before affine, after affine and after non-rigid registration in comparison to the reference image $\mathcal{S}_R$ in difference view . . . . .	19
13.	Image $\mathcal{S}_3$ before affine, after affine and after non-rigid registration in comparison to the reference image $\mathcal{S}_R$ in vertical view . . . . .	20

## References

- [1] K. K. Bhatia, J. V. Hajnal, B. K. Puri, A. D. Edwards, and D. Rueckert. Consistent groupwise non-rigid registration for atlas construction. *Proc. IEEE Intl. Symp. Biomedical Imaging*, page 908, 2004.
- [2] Daniel Rueckert, Alejandro F. Frangi, and Julia A. Schnabel. Automatic construction of 3D statistical deformation models of the brain using non-rigid registration. *IEEE Trans. Med. Imaging*, 22(8):1014–1025, 2003.
- [3] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, New York, NY, USA, 1973.
- [4] K. Pearson. On lines and planes of closest fit to points in space. *Philos. Mag.*, 2:559–572, 1901.
- [5] H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.*, 24, 1933.
- [6] D. F. Morrison. *Multivariate statistical methods*. 1976.
- [7] Anant M. Kshirsagar. *Multivariate Analysis*, volume 2 of *Statistics: Textbooks and Monographs*. Marcel Dekker, Inc., New York, 1972.
- [8] T. F. Cootes and C. J. Taylor. Active shape models: Smart snakes. In *British Machine Vision Conference*, pages 267–275, 1992.