Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Fachbereich Elektrotechnik und Angewandte Naturwissenschaften
Abteilung Molekulare Biologie

Registrierung von MRT und MEG Daten

Registration of MRI and MEG data

Bachelor Thesis

zur Erlangung des akademischen Grades
Bachelor of Sciences (B.Sc.)
im Studiengang Molekulare Biologie
im Schwerpunkt Bioinformatik

vorgelegt von
Marie Theiß
aus Krefeld

Recklinghausen, 04. August 2014

Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit von mir selbst und nur unter Verwendung der angegebenen Hilfsmittel angefertigt wurde. Eingereicht wird die Bachelorthesis in dreifacher Ausführung.
Alle Abbildungen wurden selbst erstellt, und es wurde keine inhaltsverändernde Bildbearbeitung vorgenommen.

Die Arbeit wurde an der westfälischen Wilhelms-Universität im Institut für Biomagnetismus und Biosignalanalyse unter der Anleitung von Priv.-Doz. Dr. Carsten Wolters durchgeführt

Recklinghausen, 04. August 2014

Marie Theiß

Betreuer:

1. Gutachter: Prof. Dr. Heinrich Brinck

2. Gutachter: Prof. Dr. Achim Zielesny

Westfälische Hochschule
Fachbereich Elektrotechnik und Angewandte Naturwissenschaften
August-Schmidt-Ring 10
45665 Recklinghausen

Danksagung

Zu allererst möchte ich mich bei Bima für ihre große Geduld, die zahlreichen Anregungen und ganz besonders das Korrekturlesen bedanken.

Weiterhin gilt mein Dank Familie, Freunden, sowie den Professoren und Mitarbeitern der Hochschule, die mich nicht nur während der Bachelorphase, sondern im Verlauf des gesamten Studiums unterstützt haben. Besonders hervorheben sollte ich an dieser Stelle meinen Freund, der alle meine Launen ertragen hat.

Zuletzt noch ein Dank an alle Personen, die ihre Daten für diese Arbeit zur Verfügung gestellt haben, und an die Mitarbeiter des Instituts für Biomagnetismus in Münster.

Table of contents

1 Summary

In this thesis, a function was implemented which can be used to compute a transformation matrix for the registration of MR images and MEG data. In a break from tradition, this registration procedure is independent from external anatomical landmarks. Consequently, it is no longer necessary to position anatomical markers (*fiducials*, Gadolinium makers) on the MR recordings. Instead, a 3D point cloud is used for the registration. The function pre-aligns a segmented head surface and the 3D point cloud based on the MRI coordinate system. Subsequently, the ICP algorithm minimises the mean square error between the two data sets. Finally, it is possible to perform a visual check and a manual correction on the registration.

The whole procedure has been developed to optimise or replace the registration performed according to the established pipeline. For this purpose, results from the two different registration procedures were compared based on the mean square error.

Besides, an analysis of different point clouds was executed. The results suggest that the registration is most effective if the point cloud includes the parietal, temporal and frontal head region as well as the bridge of the nose. Additionally, points of the orbital rim can be recorded. In contrast to this, points of soft facial structures and the aural regions should be omitted.

However, the results also show that it is not possible to achieve a mean square error of zero. This is caused by a deformation of the point clouds. The patient should stir as little as possible during the point cloud recordings because an optimal registration is only possible if the deformation is kept to a minimum.

Zusammenfassung

Für diese Bachelorarbeit wurde eine Funktion implementiert, welche eine Transformationsmatrix für die Co-registrierung von MRT- und MEG-Daten berechnet. Die neue Registrierungsmethode funktioniert ohne externe anatomische Marker. Dementsprechend ist es nicht mehr notwendig, die Positionen der anatomischen Marker durch Gadolinium in den MRT Aufnahmen kenntlich zu machen. Statt der Marker soll eine 3D Punktwolke für die Registrierung verwendet werden. Die Vorregistrierung wird durch die Ausrichtung der Koordinatensysteme durchgeführt. Anschließend minimiert der ICP-Algorithmus die mittlere quadratische Abweichung zwischen der 3D Punktwolke und der segmentierten Kopfoberfläche. Bei Bedarf kann abschließend eine visuelle Kontrolle, sowie eine manuelle Korrektur der Registrierung durchgeführt werden.

Das Registrierungsverfahren soll dazu dienen, das etablierte Verfahren zu optimieren bzw. zu ersetzen. Aus diesem Grund wurden die Ergebnisse der beiden Verfahren mittels der mittleren quadratischen Abweichung miteinander verglichen.

Zudem wurde eine Analyse der unterschiedlichen Punktwolken durchgeführt. Aufgrund der Ergebnisse wird angenommen, dass die Registrierung am effektivsten verläuft, wenn die Punktwolke sowohl die parietale, temporale und frontale Kopfregion als auch den Nasenrücken beinhaltet. Zusätzlich können noch die knöchernen Strukturen um die Augenhöhlen mit einbezogen werden. Im Gegensatz dazu sollten weiche Gesichtsbereiche und die Region um die Ohren nicht in der Punktwolke enthalten sein.

Insgesamt zeigten die Ergebnisse, dass es aufgrund der Deformation der Punktwolke nicht möglich ist, die beiden Datensätze ohne einen verbleibenden mittleren quadratischen Abstand miteinander zu registrieren. Um die Deformationen gering zu halten, sollte sich der Patient während der Aufnahme, der Punktwolke möglichst wenig bewegen. Nur so kann eine bestmögliche Registrierung erreicht werden.

2 Introduction

This part briefly describes the different measurement methods, data formats and coordinate systems. Moreover, it explains the current standard for the matching of MEG and MRI data, as well as the pipeline which was developed during the ‘Praxisphase’.

2.1 *Magnetic resonance imaging (MRI)*

Magnetic resonance imaging (nuclear magnetic resonance imaging) is an imaging technique in medicine. Especially soft body parts (brain, viscera, tumours) are easy to visualize.

How well a tissue is distinguished from others largely depends on its hydrogen content. Tissues with very high hydrogen content look much brighter (white) than tissues with low hydrogen content (black).

Normally, the magnetic moments of the hydrogen protons are rotated in different directions, but the magnetic field created by an MR tomography temporarily aligns them in the same direction (*longitudinal magnetisation Mz*) and the same phase.

A high frequency (HF) pulse is used to change Mz orientation by about 90 degrees. The *transversal magnetisation (M_{XY})* can be measured as the MR signal. The maximum measurable signal depends on the density of protons in a volume unit.

The transversal magnetisation and MR signal can abate through longitudinal or transversal relaxation. Different relaxation times are visible as contrast in MRI scans.

Initiated by longitudinal relaxation, the spin reverts to longitudinal magnetisation. How fast this process occurs depends on the strength of the external magnetic field and the movement of molecules in the tissue. The time required is referred to as T1 time and determines when the spins are irritable again.

Transversal relaxation describes the decay of the transversal magnetisation. Activated Spins do not evolve their energy to the environment. Instead, there is an energy exchange between the spins. As a consequence, they are not in the same phase any more. T2 is the time which is required for transversal relaxation and depends on the tissue, but it is not determined by the external magnetic field. This time determines how fast the MR signal decays after excitation.

The relaxations are not influenced by each other.[1]

All slices (tomographic images) are joined together to create a three-dimensional image.

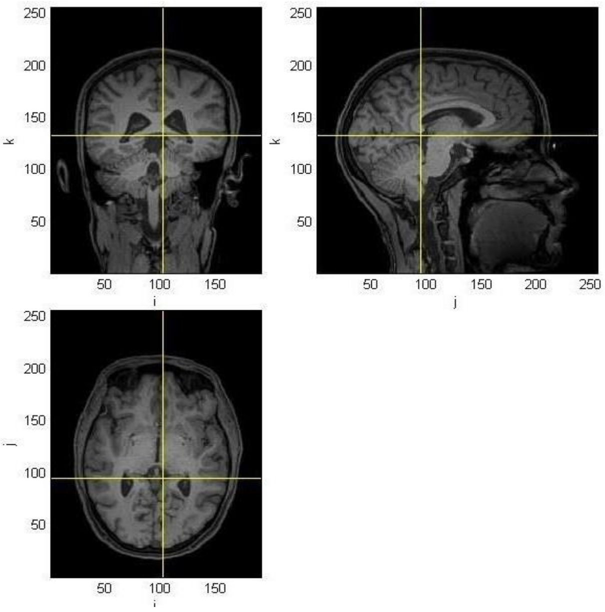

Figure 2.1| Example of a T1 weighted MRI scan.

Data of imaging methods are usually represented in a "voxel" coordinate system which does not specify the physical dimensions. It is necessary to define a head coordinate system to specify how the head relates to the voxel indices. A 4x4 homogenous transformation matrix is used to describe the head coordinate system. Three anatomical landmarks or external *fiducials*, which are easy to mark in the images, are needed to create this matrix. Depending on the position of the origin and the used landmarks, it is possible to define different coordinate systems. The *Talairach-Tournoux* coordinate system is defined using landmarks inside the brain (anterior and posterior commissure). It is helpful to map the location of brain structures by scaling them to a uniform brain size and matching them to the *Talairach-Tournoux atlas*.

The *CTF coordinate system* is defined with the help of external landmarks (nasion, left and right preauricular points) and can be used to register different data sets.[2]

[1] **Wei**

[2] **FiT**

2.2 *Magnetoencephalography (MEG)*

The magnetoencephalography is a non-invasive electrophysiological technique for recording brain activity with a very high time resolution. It is based on measuring the magnetic field outside the head using an array of superconducting sensors[3] (SQUID, superconducting quantum interference device). Three localization coils are used to determine the head's position during the measurements.

Thereby MEG contains complementary information to EEG.

Brain activity can be determined by the signal transduction between neurons. A stimulus triggers an *action potential* at the axon hillock. This expands along the axon up to the synapse. It changes the membrane potential due to charge shifting.

Neurotransmitters are submitted by the synapse. They transmit the impulse from the presynaptic to the postsynaptic neuron (formation*: postsynaptic potential*). The postsynaptic neuron can be irritated by different neurons at the same time.

If a certain threshold value is exceeded by the incoming (inhibiting and exhibiting) signals, a new *action potential* will be formed at the axon hillock.

Signals which do not exceed the threshold value will not trigger a new *action potential* and the signal transduction will stop.

The electrical transmission of impulses can be used to determine the neuronal activity.

The measurements are based on the signal transduction between neurons. Electrical signals between the neurons generate weak magnetic fields, about one billion times smaller than the earth's field.

If a neuron is excited simultaneously at several dendrites, these signals add up and cause a magnetic field (*postsynaptic potential)* which can be measured by superconducting sensors. The MEG senses magnetic fields generated by the brain and is very sensitive to those localised in the cortex.[4]

It measures predominantly quasi-tangential neural generators, while EEG measures quasi-radial signals.

[3] SQUID: can detect magnetic fields generated by neurons.

[4] iwe

The MEG uses two internal coordinate systems to locate the position of the patient's head relative to the MEG channels.

One is the *MEG dewar coordinate system*, which fixes the sensor positions, and the other is the MEG *head coordinate* system, fixing the anatomical markers. Three localization coils are fixed on the patient's head (nasion, right and left preauricular). The MEG measures the coils and localises the patient´s head relative to the (fixed) *MEG dewar coordinate system.*

The *MEG head coordinate system (CTF coordinate system)* which is defined relative to the patient´s head is then derived from the three localization coils. This can then be used to register the MEG dataset with MRI or *Polhemus Digitizer* data.[5,6]

2.3 *Polhemus FASTRAK® 3D digitizing system*

A *3D Digitizer* is used to digitise the location of EEG electrodes and the patient's head shape. It is possible to mark the desired positions of the head with the stylus, after the three anatomical landmarks for MEG/MRI registration have been digitised. The system stores all points (point clouds) in the *CTF coordinate system* (see 2.1.2 and 2.2.3).[6,7]

2.4 *MRI/MEG registration*

To localise the MEG signals with a high spatial resolution, the data need to be registered with the MR images. This is achieved by a transformation of the sensor positions. The MEG sensors are fixed in a helmet and are not in contact with the patient´s head. Therefore, the sensor positions cannot be projected onto the head's surface. Instead, it is necessary to know the head position within the MEG helmet.

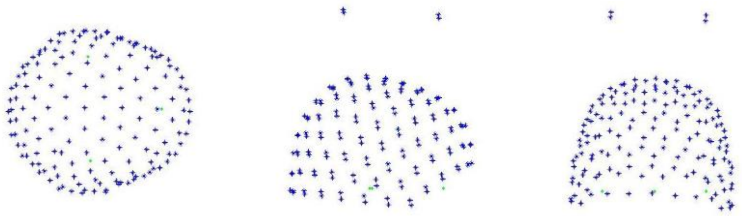

Figure 2.2| The sensor positions and anatomical markers are represented in the MEG head coordinate system.

[5] **FiT**
[6] **HLG**
[7] **FTB**

Three anatomical markers (nasion, right and left preauricular) are needed to derive the *CTF head coordinate* system. Their positions have been measured by the MRT with help of Gadolinium[8] markers (*fiducials*). In order to prevent the markers (inside the ears) from shifting during the MR recordings, they are stabilized with headphones. The *fiducials* generate bright pixels, which is visible in the slices.

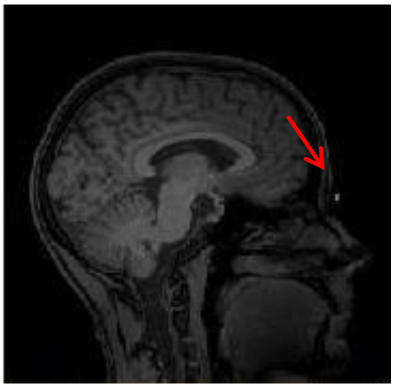

Figure 2.3| The Gadolinium markers cause bright spots within the MR images, so the anatomical structures are easy to identify.

First of all, the markers are manually defined in the MR images (guided by the bright pixels). Subsequently, a transformation matrix for the registration of MEG and MRT data can be calculated. The transformation matrix is used to compute the *CTF coordinates* for each point within the MR images. Thus, after source reconstruction, each signal measured by MEG can be localised within the MR images.

The registration of the data contains several possible sources of error. The anatomical markers which are required for the registration need to be positioned manually. It is not certain that exactly the same point is always found, although the *fiducials* are used for guidance. The markers can slip out of place during MR recordings. Moreover, the MEG does not measure the three localization coils continuously, with the result that not every movement of the patient is detected.

In order to optimise the registration of MEG and MRI data a *3D Digitizer* is used to record additional head points (3D point cloud). These points are saved in the *CTF coordinate system* and may be projected onto a segmented head surface.

[8] Gadolinium: chemical element, element category: lanthanide

The *iterative closest point algorithm* is adapted to the point cloud to optimise their registration with the surface. For this purpose, the algorithm generates a transformation matrix. To represent the patient’s head position in the MEG, the sensor positions need to be transformed by means of this matrix.

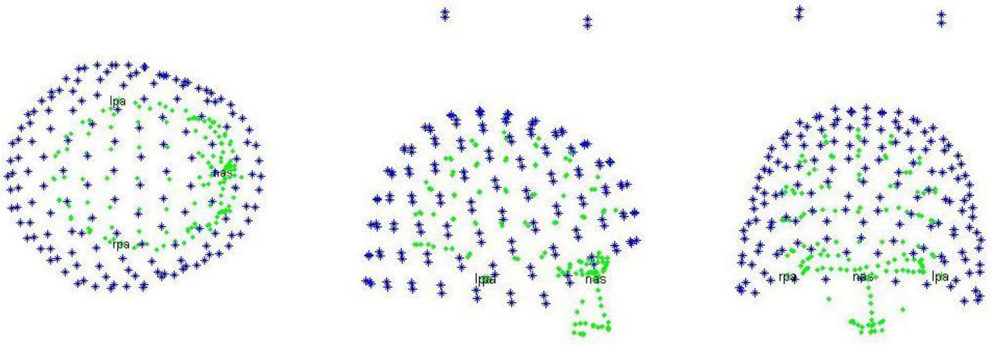

Figure 2.4| For the optimised pipeline, a 3D point cloud is stored in addition to the sensor positions and anatomical markers.

2.5 *Task*

The iterative closest point algorithm minimises the error function until it finds a local minimum (mean square distance between the point clouds remains constant). A good registration is only obtainable, if the data are aligned to each other before starting the algorithm. The optimised pipeline uses the initial registration according to the established pipeline. Thus it is still necessary to define the three anatomical landmarks within MR images.

If another way for the initial registration can be found, the *fiducials* will become obsolete. In that case, the MEG sensor positions could be aligned into the “voxel” coordinate system without modifying the MR images.

In this thesis, we try to implement a function which is able to do the initial registration without the three anatomical markers. Nevertheless, the registration by use of the iterative closest point algorithm should produce at least the same results as with the optimised pipeline. Considering that various point clouds could generate different results depending on pre-registration, it is necessary to find out which head points should be digitised.

3 Material and methods

The following section describes the used methods as well as the available software and materials.

3.1 *FieldTrip Toolbox*[9]

The *Donders Institute for Brain, Cognition and Behaviour* (Radboud University Nijmegen) developed a Matlab software toolbox for MEG and EEG analysis. It offers different algorithms for the analysis of electrophysiological data, which are used to create an effective pipeline for MEG and EEG registration. The algorithms are implemented in FieldTrip functions. They can usually be executed with a configuration structure and data structure as inputs from the command window. A configuration structure contains parameters which specify how the function behaves.

3.2 *Iterative Closest Point algorithm*

Up to this point, the optimised pipeline is equivalent to the established pipeline. The *iterative closest point algorithm* (ICP) is the essential step in this pipeline. The algorithm minimises the distance between the points of the point cloud and their nearest neighbour on the head surface.

The transformation is represented by *quaternions*. These are enhancements of the complex numbers. In contrast to the complex number (representing 2D vectors) the *quaternions* (representing 3-D rotations) contain two additional imaginary dimensions. A *quaternion* is educible in different ways.[10]

Firstly, with real numbers *(x, y, z, w)* and three imaginary components *(i, j, k),*

Equation 1| *Quaternion* with four real numbers and three imaginary components

$$\mathrm{q} = \mathrm{w} + \mathrm{x}i + \mathrm{y}j + \mathrm{z}k$$

[9] **FiT**

[10] **eds**

where *i, j*, *k* are the imaginary part $\left(=\sqrt{-1}\right)$ and *x, y, z, w* are real numbers. So it contains one real and three imaginary dimensions. *x, y* and *z* represent the axis about which the rotation occurs and *w* represents the scale of rotation.[11]

Secondly, in terms of axis angle:

Equation 2| *Quaternio*ns represented with axis-angle

$$\mathrm{q} = \cos\left(\frac{\mathrm{a}}{2}\right) + \mathrm{i}\left(\mathrm{x} * \sin\left(\frac{\mathrm{a}}{2}\right)\right) + \mathrm{j}\left(\mathrm{y} * \sin\left(\frac{\mathrm{a}}{2}\right)\right) + \mathrm{k}\left(\mathrm{z} * \sin\left(\frac{\mathrm{a}}{2}\right)\right)$$

In this formula, *a* is the angle of rotation and *x, y, z* are vectors representing the axis of rotation (*i, j*, *k* are still the imaginary part $\left(\sqrt{-1}\right)$).[12]

The algorithm uses the following methods to compute a transformation matrix for the registration.

First of all, the algorithm denotes the minimal distance metric d between each point $\vec{p}$ (of the point cloud) and one (the nearest) point (x_i) on the surface mesh X.

Equation 3| Euclidean distance

$$d(\vec{p}, A) = \min_{i \in \{1,\ldots,N_x\}} d(\vec{p}, x_i)$$

Subsequently, the mean square objective function needs to be minimised:

Equation 4| Error function

$$f(q) = \frac{1}{n}\sum_{i=1}^{n} \left|\left|\vec{x_i} - R(q_R)\vec{p_i} - q_T\right|\right|^2$$

It means in fact that a rotation matrix and a translation vector needs to be calculated. They should transform the points of the point cloud so much that the mean square error becomes minimal.

Equation 5| Rotation matrix generated by *quaternions*

$$R(q_R) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}$$

[11] **cPg**

[12] **eds**

The rotation matrix $R(q_R)$ is generated by a unit rotation *quaternion* which is computed as follows:

For the given data (point cloud and surface mesh), the cross-covariance matrix Σ_{px} is needed (^{t} implies vector transpose).

Equation 6| Cross-covariance matrix

$$\Sigma_{px} = \frac{1}{N_p} \sum_{i=1}^{N_p} (\overrightarrow{p_i} - \vec{\mu}_P)\,(\overrightarrow{x_i} - \vec{\mu}_x)^t = \frac{1}{N_p} \sum_{i=1}^{N_p} (\overrightarrow{p_i} x_i^T - \vec{\mu}_P \vec{\mu}_x^t)$$

The “centre of mass“ is generated before by:

Equation 7| Barycentre

$$\vec{\mu}_P = \frac{1}{N_P} \sum_{i=1}^{N_P} \overrightarrow{p_i} \qquad \vec{\mu}_x = \frac{1}{N_x} \sum_{i=1}^{N_x} \overrightarrow{x_i}.$$

The next step is to form the column vector $\Delta = [A_{23} A_{31} A_{12}]^T$. For that reason, the cyclic components of the anti-symmetric matrix $A_{ij} = \left(\Sigma_{px} - \Sigma_{px}^t\right)_{ij}$ are needed.

Afterwards, the symmetric 4x4 matrix $Q\left(\Sigma_{px}\right)$ is generated from the column vector and the cross-covariance matrix (*tr* implies trace function[13], I_3 is the 3x3 identity matrix).

Equation 8| Symmetric 4x4 matrix

$$Q\left(\Sigma_{px}\right) = \begin{bmatrix} tr\left(\Sigma_{px}\right) & \Delta^T \\ \Delta & \Sigma_{px} + \Sigma_{px}^T - tr\left(\Sigma_{px}\right) I_3 \end{bmatrix}$$

The optimal rotation vector is the unit eigenvector $q_R = [q_0 q_1 q_2 q_3]^T$ associated with the maximum eigenvalue of the matrix $Q\left(\Sigma_{px}\right)$.

Afterwards it is possible to generate the optimal translation vector by means of the following function:

Equation 9| Translation vector

$$q_T = \mu_x - R(q_R)\mu_p$$

[13] trace function: $tr\left(\Sigma_{px}\right) = \sum_{j=1}^{\mathrm{n}} (\Sigma_{px})_{jj}$

The ICP algorithm creates a transformation matrix which registers the point cloud with the head surface.[14]

3.3 *Optimised registration*

Figure 3.1 depicts the pipeline for pre-processing and registration of the available data.

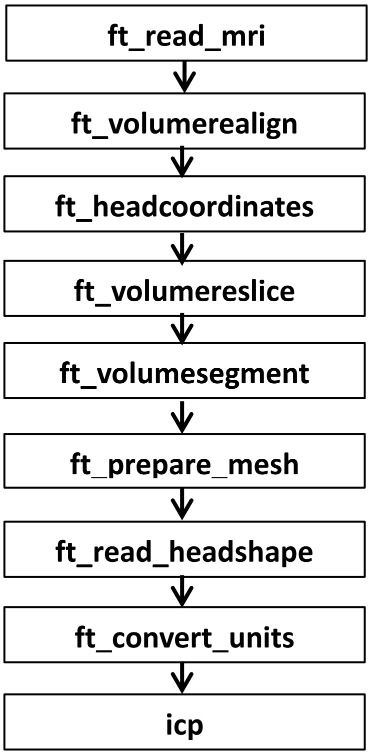

Figure 3.1| Pipeline of optimised registration.

After the data have been converted to the *head coordinate system*, the ICP algorithm is used to optimise the registration, but the *fiducials* are still needed for this pipeline. All functions are implemented in the Matlab software toolbox FieldTrip.

[14] **Bes**

3.3.1 Available data

For this thesis, data of six people are available - both MR data and 3D point clouds. All data have been acquired and provided by the institute for bio magnetism and bio signal analysis in Münster[15]. The propositi have different physical characteristics in sex, age, hair, weight and size to make sure that the pipeline is applicable to people of all kinds.
For the analysis, the data need to be pre-processed in different ways.

3.3.2 Pre-processing MR images[16]

FieldTrip supports different data formats. A list of the data formats that are supported by FieldTrip is present on the website. Our anatomical MRI data are saved as nifti (*.nii/*.nii.gz) files, which is supported by the following function:

Function 1| Loading MRI to Matlab.

```
mri = ft_read_mri('mri_data.nii');
```

For registration, the skin surface is required. Therefore, it is necessary to follow the procedure below, after loading the data.

Initially, it is necessary to align the MRI to the head coordinate system. For that reason, the three anatomical markers need to be set interactively in the MR images.

Function 2| Setting anatomical markers interactively.

```
cfg           = [];
cfg.methode   = 'interactive'
cfg           = ft_volumerealign(cfg,mri)
```

Subsequently, the transformation matrix which describes the voxel position in the *head coordinate system* is computed.

[15] **BMM**
[16] **FiT**

Function 3| Computing transformation matrix.

```
voxel2ctf         = ft_headcoordinates(nas, lpa, rpa, 'ctf');
mri.transform     = voxel2ctf;
```

It is possible to do a visual check on the coordinate system.

Function 4| Checking coordinate system interactively.

```
mri = ft_determine_coordsys(mri, 'interactive', 'yes');
```

For this a figure pops up, which allows screening the anatomical labels of the coordinate system axes. The additional input argument ‘axisscale’ enables the user to also change the scaling factor for the reference axes.

The next step is to align the anatomical MRI to the *head coordinate system*. This could be done with the following function.

Function 5| Aligning MRI to the head coordinate system.

```
cfg = [];
mri = ft_volumereslice(cfg,mri);
```

In this way, the anatomical MRI is defined in the same coordinate system as the MEG sensor positions, and the voxels are isotropic. Additionally, the function may be used to change the resolution (voxel size), or to zoom in on a part of the volume.

Each voxel of the anatomical MRI needs to be separated into the different tissues. It is possible to define which compartments (skull, scalp) and brain areas (cerebro-spinal fluid, white and grey matter) should be segmented. Results of the segmentation are probabilistic tissue maps of the compartments.

Function 6| Segmenting of the anatomical MRI.

```
cfg              = [];
cfg.output       = {'scalp'};
segmentedmri     = ft_volumesegment(cfg, mri);
```

The next task is to generate surfaces at the borders of the tissue types (meshes) and head models (volume conduction models) based on anatomical information. Each mesh is represented by points (vertices) connected in a triangular way. The volume conduction model is required for source analysis after the registration. It contains the meshes for different tissue types, as well as the conductivity value for each compartment and a matrix used for the volume conduction model with additional information. The model determines how a source is visible within the brain on the MEG sensors, and it is not needed for the registration.

The following invocation creates a mesh for the scalp (the mesh represents the head surface) with 200000 vertices from the segmented MRI.

Function 7| Preparing mesh for the segmented tissues.

```
cfg              = [];
cfg.tissue       = {'scalp'};
cfg.numvertices  = [200000];
bnd              = ft_prepare_mesh(cfg,segmentedmri)
```

3.3.3 Loading 3D point clouds

For each proband, a few point clouds (containing different head shape points) are available. The person was asked to maintain his/her position, since movements cause deformations of the point cloud. [17]

All point clouds are saved in the same coordinate system as the MEG sensor positions. The following figures show examples of different head points collected from the bony parts of the head.

[17] **FTB**

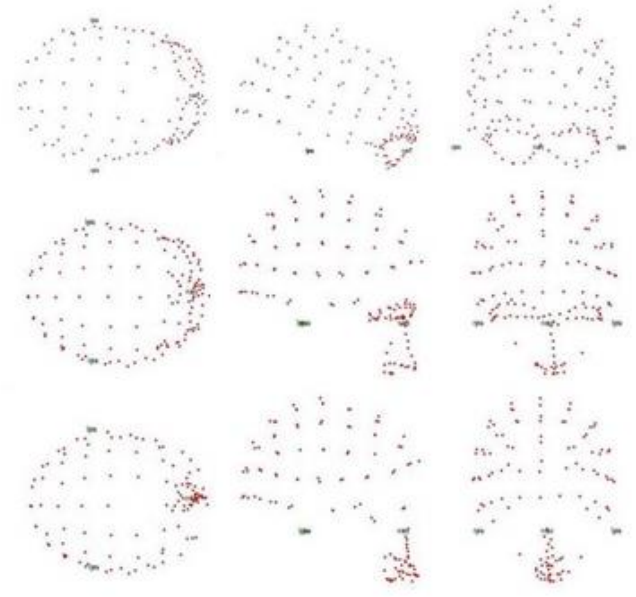

Figure 3.2| Examples of different point clouds. Top view (left column), right side of the face (middle column), frontal (right column). Point cloud f (top row), Point cloud b (middle row), point cloud c (bottom row).

The function

Function 8| Loading 3D head point to Matlab.

```
pol = ft read headshape('pol data.pos');
```

loads the head points from different file formats (*.gii, *.pos). After reading in the data, it is necessary to check their unit and change the geometrical dimensions to the same as the anatomical MRI (mm).

Function 9| Converting units.

```
pol = ft_convert_units(pol, 'mm');
```

The registration may be improved by using the iterative closest point algorithm after loading the point clouds and pre-processing the anatomical MRI. [18]

[18] **FiT**

3.3.4 Optimising registration by ICP

After pre-processing all data, the ICP algorithm is used to compute a transformation matrix which optimises the registration. An implementation of the ICP algorithm (by Martin Kjer und Jakob Wilm from the Technical University of Denmark, 2012) can be found on the *mathworks* (central, file exchange) website.[19]

The function is called with the following command.

Function 10| Using ICP algorithm.

```
[TR, TT, ER, t, info] = icp(bnd.pnt',pol_data.pnt', iter);
```

The input parameters are $3xN$ arrays for the point clouds and a numeric value for the number of iterations. A translation vector, rotation matrix, and additional information are the output parameters of this function. For example, ‘ER’ provides information about the mean square distance between the point clouds. Equation 8[20] shows how the distance between the point clouds is calculated.

Equation 10| Mean square error

$$mse = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_{i,x} - x_{i,x})^2 + \frac{1}{N}\sum_{i=1}^{N}(p_{i,y} - x_{i,y})^2 + \frac{1}{N}\sum_{i=1}^{N}(p_{i,z} - x_{i,z})^2}$$

[19] **MaL**

[20] **cnx**

4 Results

This chapter firstly presents the implemented function and secondly summarises the different results of registration (by use of ICP) resulting from different pre-registrations.

4.1 *Implemented function*

The implemented function uses the segmented head surface (stored as a mesh represented by points connected in a triangular way) and a 3D point cloud (representing the patient's head shape) as well as a configuration field as input parameters. First of all, the function checks whether a coordinate system has been defined for the head surface. The function presupposes that the 3D points are stored in the *CTF head coordinate system* (default for Polhemus FASTRAK® 3D digitizing system). If no 'bnd.coordsys' structure is found, a 'figure' (window) pops up that allows a visual check on the units and anatomical labels for the coordinate system axes. The orientation of the coordinate system is defined on the basis of the anatomical labels.

Afterwards, the function computes a rotation matrix (to be applied to the point cloud) that aligns the two datasets to the same direction. Furthermore, a translation vector is calculated which moves the barycentre of the point cloud to the barycentre of the surface mesh. In this way, the different data sets are pre-aligned and the ICP is applicable to the point cloud.

After the pre-alignment the function calls the ICP to optimise the registration. Finally, a complete transformation matrix is calculated. It includes the pre- and optimizing transformation matrices.

Equation 11| Total rotation matrix

$$Rot_{total} = Rot_{ICP} * Rot_{pre}$$

Equation 12| Total translation vector

$$Trans_{total} = Rot_{ICP} * Trans_{pre} + Trans_{ICP}$$

The configuration can contain options to specify the method and the number of iterations (for further information see appendix). It is possible to check the registration after optimisation. For this purpose, an interface enables the user to perform a visual check on the alignment after the ICP has been executed. Furthermore, the user can edit the registration manually. This option is realised through a function from the FieldTrip software. The function is called with a configuration structure which contains the template (surface mesh) and the individual head shape (3D head points).

Function 11| Interactive checking and modification of registration.

```
cfg.template.headshape        = bnd;
cfg.individual.headshape      = polhemus;
cfg                           = ft interactiverealign(cfg);
```

Afterwards, the ICP algorithm starts again with 100 iterations to optimise the manual registration.

The implemented function is called in the following way:

Function 12| Implemented function for registration.

```
[matrix, info] = ft_registrate_headshape(bnd, polhemus, cfg);
```

The output variables comprise the total transformation matrix and some diagnostic information (‘info’ variable). The latter contains the indices to the matching points of the point clouds and the positions before and after registration. Finally, the matching results are plotted.

The main code is shown below. Additional functions were used but are not described in the ‘Material and Methods’ chapter. They can be found in the appendix.

Programming code 1| Code of the implemented function.

```
function [matrix,info] =
ft_registrate_headshape(bnd,polhemus,cfg)

%% This function is used to register Polhemus data from ctf
coordinate system to the MRI voxel coordinate system

polhemus = ft_convert_units (polhemus, 'mm');

% Checking the input parameter and set defaults
iterations = isfield (cfg, 'iterations');
    if iterations == 0
        cfg.iterations = 200;
    end
method = isfield (cfg, 'method');
    if method == 0
        cfg.method = 'automatic';
    end
coordsys = isfield (bnd,'coordsys');
    if coordsys == 0
        bnd = ft_determine_coordsys(bnd, 'interactive', 'yes');
    end

%Calculate transformation matrix for pre-alignment and move head
points
r1 = alignaxes(bnd);
s_polhemus = barycentre(polhemus.pnt);
s_bnd = barycentre(bnd.pnt);
t1 = s_bnd - s_polhemus;
matrix = r1;
matrix(:,4) = t1;
pol = moveheadshape(matrix, polhemus);

%Using the ICP algorithm
[r2, t2, ER, t, info] = icp(bnd.pnt', pol.pnt',
cfg.iterations,'Matching', 'kDtree');

%Save the output parameters and move head points
info.ER = ER;
t1 = t1';
matrix = r2*r1;
matrix(:,4) = r2*t1+t2;
polhemus = moveheadshape(matrix,polhemus);

%Using the defined method
switch cfg.method
    case  'interactiv'
        fprintf('interactiv check')
    %Do a interactiv check and move head points with manually
generated

    %transformation matrix
    cfg.template.headshape = bnd;
    cfg.individual.headshape = polhemus;
    [cfg] = ft_interactiverealign(cfg);
    polhemus = moveheadshape(cfg.m,polhemus);

    %Calculate new output matrix
    matrix(1:3,1:3) = cfg.m(1:3,1:3)*matrix(1:3,1:3);
    matrix(:,4) = cfg.m(1:3,1:3)*matrix(:,4)+cfg.m(1:3,4);

    %Optimise manual registration by ICP algorithm
    [r_interactiv, t_interactiv, interactiv.ER, interactiv.t,
interactiv.info] = icp(bnd.pnt', polhemus.pnt', 100 ,'Matching',
'kDtree');

    interactiv_martix = r_interactiv;
    interactiv_martix(:,4) = t_interactiv;
    %Calculate new output matrix
    matrix(1:3,1:3) = r_interactiv*matrix(1:3,1:3);
    matrix(:,4) = r_interactiv*matrix(:,4)+t_interactiv;

   %Save the output parameters and move head points
   info.interactiv = interactiv.info;
   info.interactiv.ER = interactiv.ER;
   polhemus = moveheadshape(interactiv_martix,polhemus);

   %Mean square error after registration
   Error = interactiv.ER(101,1);

    case 'automatic'
        fprintf('registration is done')
    %Mean square error after registration
    Error = info.ER(cfg.iterations+1,1);
end
    figure; hold on;
    %Plot the results

ft_plot_mesh(bnd,'facecolor','skin','vertexcolor','none','edgeco
lor','none','facealpha',0.9,'edgealpha',0.9);

plot3(polhemus.pnt(:,1),polhemus.pnt(:,2),polhemus.pnt(:,3),'+')

    %Print the mean square error after registration
        fprintf('Mean sqaure error after registration %f mm',
Error)

end
```

4.2 *Function test*

The function was tested for reliable results after implementation. For this purpose, a comparison was drawn between the results deduced from the optimised pipeline and those from the implemented function.

4.2.1 Test data

For two of the six subjects, more than one point cloud was available. The results are based mainly on these subjects. The segmentation and meshes for these persons have been performed with CURRY[21] by Ümit Aydin[22]. The software offers a better segmentation quality. However, it consumes more time than the FieldTrip software. Hence, the surfaces for the other subjects were created with FieldTrip. The following tables give an overview of the different surfaces and point clouds.

Subject	vertices[23]	faces[24]
A 1457	25,335	50,666
A 1458	26,914	53,824
A 1468	200,000	399,996
A 1469	200,000	399,996
A 1470	200,000	399,996
A 1472	200,000	399,996

Table 4.1| Prepared surfaces for the subjects.

	No.	Parietal, temporal, frontal bones	Nose	Orbital		Points
A1457	a	EEG electrode positions				80
	b	EEG electrode positions	bridge			119
	c	EEG electrode positions	bridge		supra	138
	d	Standard[25]				101
	e	Standard	bridge			124
A1458	f	EEG electrode positions		sub	supra	125
	g	Standard				114
	h	Standard	bridge			144
A1468	i	EEG electrode positions	bridge	sub	supra	150
A1469	j	EEG electrode positions	bridge	sub	supra	142
A1670	k	EEG electrode positions	bridge	sub	supra	150
A1472	l	EEG electrode positions	bridge	sub	supra	138

Table 4.2| Available point clouds of the different subjects.

[21] Neuroimaging Suite software, **NeS**

[22] employee at the institute for bio magnetism and bio signal analysis in Münster , **BMM**

[23] vertices: number of points/corners of the triangles

[24] faces: number of triangular surfaces

[25] standard: point cloud used at the institute so far

4.2.2 Registration results

The prepared surfaces and head points were registered by the different pre-register procedures. As the following pictures show, the results varied widely after the pre-registrations.

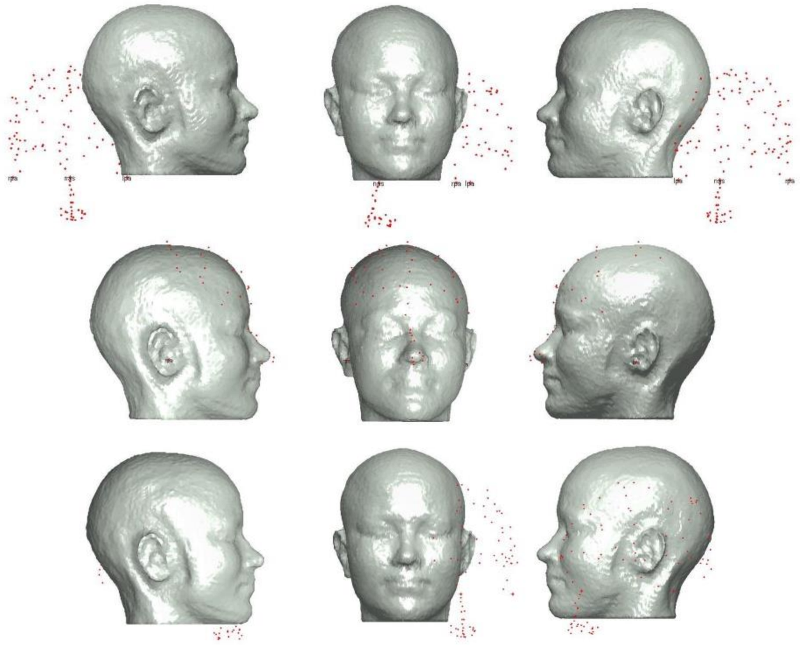

Figure 4.2| Pre-registration for point cloud e. Right side of face (left column), frontal (middle column) left side of face (right column). Before pre-registration (top row), after *fiducial* pre-registration (middle row), after CSB pre-registration (bottom row).

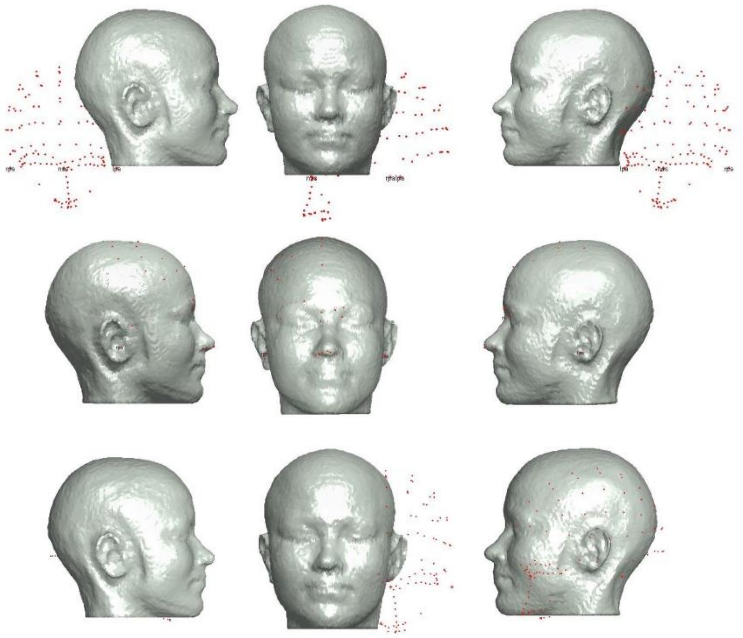

Figure 4.1| Pre-registration for point cloud c. Right side of face (left column), frontal (middle column) left side of face (right column). Before pre-registration (top row), after *fiducial* pre-registration (middle row), after CSB pre-registration (bottom row).

For each pre-registration procedure, the achieved mean square error and the required number of iterations for reaching the minimum distance with the ICP algorithm were recorded. Usually, the function was terminated after the automatic registration. A manual correction of the alignment was conducted only if the matching was obviously wrong.

No.	Pre-matching: *fiducials*	Iterations	Pre-matching: CSB[26]	Iterations	manual control	Iterations
a	3.4038	23	3.4845	141	#[27]	
b	3.3691	24	3.3726	156	#	
c	3.7260	35	3.7720	102	#	
d	4.2390	11	5.2641	80	4.1682	12
e	3.7500	33	5.5426	144	3.8331	16
f	4.4151	23	4.4172	103	#	
g	2.5840	33	4.1078	67	2.7255	16
h	3.7160	22	6.3893	90	3.6974	35
i	4.0545	46	4.0676	131	#	
j	3.0676	29	3.0446	79	#	
k	2.7340	79	6.9427	64	2.7803	51
l	3.3188	46	3.3340	107	#	

Table 4.3| Results after registration with the different procedures.

The results showed that it is possible to reach similar mean square derivations with the different ways of pre-registration. Although there were large differences between the numbers of iterations, the ICP algorithm was not very time-consuming during the automatic registration. The search for the nearest neighbour was done with a k-d tree[28] which is a time-saving method.

The new registration procedure was corrected by the user for five point clouds. Following this, it was possible to get a similar mean square distance as by *fiducial* pre-registration. The user had to correct all registrations which were achieved with the standard head points. Except for point cloud k, all head points were registered quite well after applying the new pre-matching method.

In addition, the results showed that the matching quality was not influenced by the number of points in the point cloud.

[26] CSB: Coordinate system and barycentre based pre-registration.

[27] #: no value determined

[28] k-d tree: k-dimensional tree, organises points in a k-dimensional space and can be used to do nearest neighbour searches quite fast.

Even though the deviation between the mean square errors was only slight, there were visible differences between the results of the different pre-registrations. This became clear by the following figures.

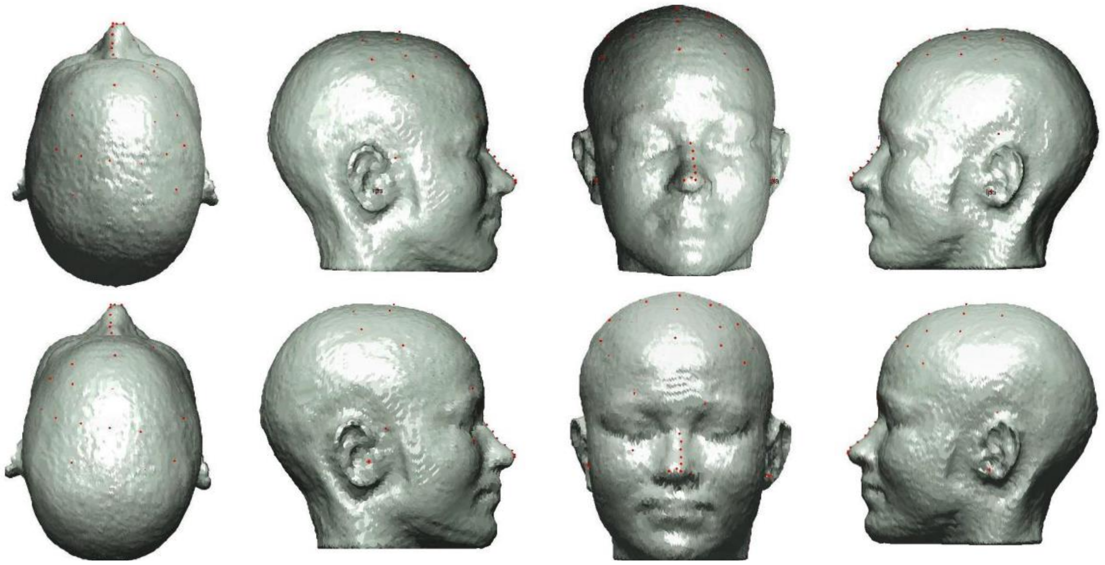

Figure 4.3| Registration results for point cloud c. Top view (first column), right side of face (second column), frontal (third column) left side of face (fourth column). After *fiducial* pre-registration (top row), after CSB pre-registration (bottom row).

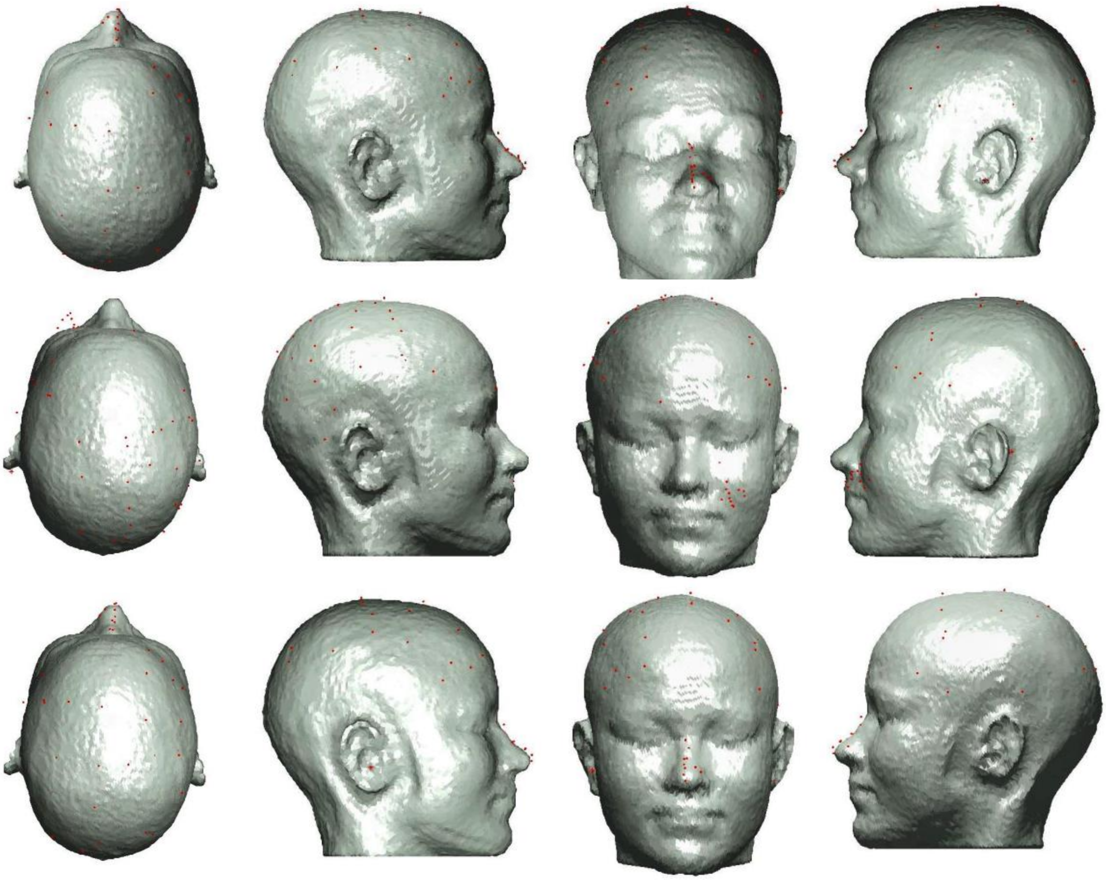

Figure 4.4|Registration results for point cloud d. Top view (first column), right side of face (second column), frontal (third column) left side of face (fourth column). After *fiducial* pre-registration (top row), after CSB pre-registration (middle row), after manual correction (bottom row).

5 Discussion

The optimised pipeline uses the *fiducials* to pre-align the data sets. A pre-alignment is essential to generate a good registration with the ICP. The algorithm iteratively minimises the mean square error between the head surface and the point cloud. This implies that the algorithm terminates immediately when a local minimum is reached. In this context it does not matter if the points are registered anatomically correctly.

One aim of this thesis was to implement a function that aligns the data without the *fiducial* positions. This means that a new pre-alignment procedure should create an acceptable result without the three anatomical markers.

The implemented function can be used with the 3D point cloud and a head surface. It is no longer necessary to mark the anatomical landmarks in the MR images for the pre-alignment. Consequently, it is unnecessary to perform MR recordings with the Gadolinium markers. Since the function reviews the MR coordinate system, it is possible to use differently represented MR data (acquired with varying recording systems). With reference to this, the task has been fulfilled.

However, the results obtained with the different pre-alignments show marked differences. By use of the pre-registration with the *fiducials,* a good alignment is achievable. In contrast to this, the CSB registration generates bad results. After CSB pre-registration, the mean square error between the point cloud and the surface is very large.

No.	Pre-matching: *fiducials*	Pre-matching: CSB
a	4,2873	24,5240
b	4,6287	33,7649
c	4,0497	36,0743
d	5,1607	34,9926
e	5,7521	29,4464
f	5,7946	33,8517
g	4,7443	23,4239
h	5,0402	23,0723
i	5,4964	39,2371
j	4,3050	32,7691
k	3,6545	43,1077
l	4,8999	35,8907

Table 5.1| Mean square error after pre-registration with the different procedures.

But it is important to distinguish the pre-alignment procedures in view of their purpose. Up to now, the *fiducial* matching has beed used to register the data sets, whereas the CSB registration is needed only to align the data for the following procedure. The figures show that the CSB registration aligns the datasets to the same direction, which is the condition for the optimisation by means of the ICP algorithm.

Not only the CSB registration, but also the *fiducial*-based pre-registration has been improved by the ICP algorithm. Thus, the mean square error has shown low differences for the two pre-alignments after the optimisation. Nevertheless, it was not possible to achieve a perfect registration. Especially those point clouds which were recorded with the standard procedure[29] could not be registered well with the ICP algorithm after the CSB pre-registration. Maybe the points are not homogenously distributed above the parietal, temporal and frontal bones. The irregular distances between the points could cause difficulties during the iteration procedure. Therefore the user should perform at least a visual check on the registration. If the manual registration is performed well, the ICP algorithm generates a similar result as with the *fiducial* pre-registration.

Generally, the standard point clouds could be used for the developed pipeline, but this generates additional labour. For that reason, the point clouds which include the EEG electrode positions seem to be more suitable for the registration. Moreover, these point clouds are also useful for the additional registration with EEG data.

The results showed that registrations with head points of the parietal, temporal and frontal bones generate small mean square errors between the point cloud and the head surface. But in this context, it should be noted that these point clouds are approximately hemispherical. Since the head surface is also hemispherical in some parts, the point clouds can slip out of place. In this case, a small mean square error is computed although the point cloud is registered wrongly.

Therefore, additional head points on the bridge of the nose are useful for guidance in the matching. Furthermore, this anatomical characteristic is segmented well by

[29] parietal, temporal and frontal head region

the MR images. That is also why it is suitable for the registration. Another part of the face which is clearly defined after segmentation is the region around the eyes. Especially the *arcus superciliaris*[30] is visible on the head's surface. This is a benefit for the registration of the 3D point cloud with the head surface. But the results showed that it is not enough to include only the orbital region. Point cloud f which described the parietal, temporal and frontal head region as well as the orbital region generated the worst matching quality, whereas the registration with those points clouds which additionally included the bridge of the nose was of a high quality. This fact shows that not the orbital region, but rather the bridge of the nose is suitable for stabilising the registration.

It should be noted that the head surface is segmented inaccurately where the headphones are positioned. According to this, it is not useful to place head points in this region. But the incorporation of points from this region may be helpful if the headphones could be omitted. For that purpose, the headphones need to be substituted by earplugs as soon as the *fiducials* are no longer needed for pre-registration.

After the registration, an error of 2.5 to 4.5 mm was left for every point cloud, independently of the pre-registration method. It was not possible to get a better result than that. This is due to different aspects, not only the alignment procedure. A deformation of the point clouds also prevents an optimal adjustment.

First of all the *Polhemus FASTRAK® 3D digitizing system* has a static accuracy of 0.76 mm RMS[31] (for the X, Y, or Z position) and 0.15º RMS for the receiver orientation[32]. So the point cloud is deformed already due to the system.

Furthermore, the proband moves when a head point is marked. Each contact with the stylus pushes the person's head away whereby a deformation of the point cloud is generated.

[30] supraorbital arch

[31] RMS: root mean square, quadratic mean

[32] **FTB**

A third aspect which causes the differences between the point cloud and the head surface could also be found in the recording procedure. The soft touch by the stylus provokes not only a head movement but also a dent of the skin. This appears most notably in soft facial regions (cheek, lips, nasal wings). Thus, these regions should not be included in the head points.

The point clouds which were used for this thesis included predominantly points of the parietal, temporal and frontal head region, the orbital rim and the bridge of the nose. But they also contained points of the nasal wings and the tip of the nose. In this region a deformation of the point clouds is possible. For further tests it makes sense to omit these regions from the point cloud.

Moreover, the proband has different postures during the data recording. The person lies on the back during the MRI and MEG recordings. In contrast, the point clouds are acquired while the person sits on a chair. The problem is that the shape of the face is changed slightly due to a postural change.

Consequently, there must be minor deviations between the facial structures during the MRI/MEG and the point clouds recordings. That is an additional reason for the remaining mean square deviation after the optimisation.

For the stated reasons, a point cloud could never fit perfectly to the head's surface.

Another important aspect is the evaluation of the matching. The function prints the mean square error to the command window after registration. For the given data it was up to a value of 4.4172 mm (for point cloud f) after the CSB pre-matching. But this value says nothing about the accuracy of the registration as such. This becomes clear from point cloud g. After the consecutive CSB pre-alignment and ICP optimisation, an acceptable mean square error of 4.1078 mm was calculated. Nonetheless, the visual check revealed that the registration looked bad.

All in all, the implemented function is usable for the registration of the head surface with the 3D head points. The different pre-registrations influence the results minimally with regard to the mean square error. It is anything but certain which of the final outcomes is the best. This is mainly because there are so many error sources during the data collection. It is inadvisable to pay too much attention to the mean square deviation. Ultimately it is only a statistical value and not the critical factor. Therefore, a visual control should always be a step in the alignment process, because this is the only way to find out whether the registration proceeded well.

To further develop the procedure, it would be useful to ascertain the errors caused during the point cloud recording. Thus a threshold for the mean square error could be defined. Subsequently, it would be an asset to perform physiological experiments. They would serve to get test data which could be used to compare the new registration method with the established pipeline. For this purpose, a source localisation should be performed with the different matching procedures.

6 List of references

[Bes] Besl P.J., McKay N.D.: A method for Registration of 3D-Shapes, IEEE Trans PAMI, 14 (2), 1992

[BMM] Medical Faculty Münster, Univ.-Prof. Dr. med. Dr. h.c. W. Schmitz, Univ.-Prof. Dr. Christo Pantev, Münster, Germany**,** http://campus.uni-muenster.de/biomag.html (11th July 2014, 10:28 am)

[cnx] William & Flora Hewlett Foundation, Bill & Melinda Gates Foundation, 20 Million Minds Foundation, Maxfield Foundation, Open Society Foundations, and Rice University. http://www.cnx.org/
(10th July.2014, 12:41 am)

[cPg] Alex Allain, Alex Hoffer, Michael Kern, Cprogramming.com, http://www.cprogramming.com/ (11th July 2014, 8:30 am)

[eds] Martin John Baker, euclideanspace.com, http://www.euclideanspace.com/ (11th July 2014, 9:08 am)

[FiT] FieldTrip, Donders Institute for Brain, Cognition and Behaviour, http://fieldtrip.fcdonders.nl/

[FTB] Fastrak Brochure, Polhemus (Rev. June 2012), Vermont, US and Canada, http://www.polhemus.com/ (4th July.2014, 5:17 pm)

[HLG] VSM MedTech Ltd., medical advances through technology, Canada, *Head Localization Guide; CTF; MEG Software,* Revision 3.2, 03 August 2006, http://www.vsmmedtech.com/

[iwe] Institute for Learning & Brain Sciences, University of Washington, Seattle, US, http://ilabs.washington.edu/ (21th June 2014, 3:44 pm)

[MaL] The MathWorks, Inc, Natick, Massachusetts (US), http://www.mathworks.com/ (30th June 2014, 11:32 am)

[NeS] Lumenbrite, http://www.neuroscan.com/ (19th June 2014, 9:54 am)

[Wan] Yutao Wang, University of British Columbia http://taylorwang.wordpress.com/ (16th June 2014, 4:13 pm)

[Wei] Dominik Weishaupt, Victor D. Köchli, Borut Marincek (2006*) Wie funktioniert MRI?: Eine Einführung in Physik und Funktionsweise der Magnetresonanzbildgebung*, Springer

7 Appendix

7.1 *Software and devices*

Software	Version	Manufacturers
MATLAB	R2013b	MathWorks, Natick, Massachusetts (USA)
FieldTrip	fieldtrip-20131211	Donders Centre for Cognitive Neuroimaging, Nijmegen (Netherlands)

Table 7.1| Used software.

Device	Modell	Manufacturers
MRI-Scanner	MAGNETOM Prisma	SIEMENS, München (Germany)
3-D Digitizer	Fasttrack	Polhemus, Colchester, Vermont (USA/Kanada)
Gadolinium marker		Assembling shop, Uniklinikum Münster (Germany)

Table 7.2| Used Devices.

7.2 *Point clouds*

Subject	No.	Head	Nose	Orbital	Altogether
A1457	a	80			80
	b	80	29		119
	c	80	21	37	138
	d	101			101
	e	98	26		124
A1458	f	79		46	125
	g	114			114
	h	105	39		144
A1468	i	80	35	35	150
A1469	j	80	27	35	142
A1670	k	80	35	35	150
A1472	l	80	23	35	138

Table 7.3| Number of points representing the different parts of the head/face.

7.3 *Lists*

List of abbreviations

ICP: Iterative Closest Point

MEG: Magnetoencephalography

MRI: Magnetic resonance imaging

MRT: Magnetic resonance tomography

CSB: Coordinate system- and barycentre-based pre-registration

Appendix

List of equations

List of figures

List of functions

List of programming code

List of tables

7.4 *Helper functions and configuration*

Function 13| Configuration options for the implemented function.

```
cfg.method       =    different methods to align the volume
                           'automatic' (default = 'automatic'),
                           'interactive'
cfg.iterations        =    number of iterations
                           value (default = 200)

When cfg.method = 'automatic', the ICP algorithm starts after the
datasets have been pre-aligned with the defined number of
iterations.
When cfg.method = 'interactive', a user interface allows a visual
check on the alignment and a manual adjustment. Afterwards the ICP
algorithm starts again with 100 Iterations.
```

Programming code 2| Calculation of the rotation matrix for the initial alignment.

```
function [r1] = alignaxes(bnd)
% This function creats a rotation matrix based on the MRI
coordinate system

switch bnd.coordsys
    case 'ras'
        r1 = [0,-1,0;1,0,0;0,0,1];
    case 'lps'
        r1 = [0,-1,0;1,0,0;0,0,1];
    case 'itab'
        r1 = [0,-1,0;1,0,0;0,0,1];
    case 'neuromag'
        r1 = [0,-1,0;1,0,0;0,0,1];
    case 'mni'
        r1 = [0,-1,0;1,0,0;0,0,1];
    case 'ctf'
        r1 = [1,0,0;0,1,0;0,0,1];
    case 'als'
        r1 = [1,0,0;0,1,0;0,0,1];
    case '4d'
        r1 = [1,0,0;0,1,0;0,0,1];
    case 'bti'
        r1 = [1,0,0;0,1,0;0,0,1];
    case 'las'
        r1 = [0,1,0;-1,0,0;0,0,1];
    case 'spm'
        r1 = [0,1,0;-1,0,0;0,0,1];
end
end
```

Programming code 3| Moving the head shape with a given transformation matrix.

```
function [Po] = moveheadshape(M,headshape)
% This function transforms a headshape with fiducials

Po.pnt=zeros(size(headshape.pnt));
Po.pnt(:,1)=headshape.pnt(:,1)*M(1,1)+headshape.pnt(:,2)*
M(1,2)+headshape.pnt(:,3)*M(1,3)+M(1,4);
Po.pnt(:,2)=headshape.pnt(:,1)*M(2,1)+headshape.pnt(:,2)*
M(2,2)+headshape.pnt(:,3)*M(2,3)+M(2,4);
Po.pnt(:,3)=headshape.pnt(:,1)*M(3,1)+headshape.pnt(:,2)*
M(3,2)+headshape.pnt(:,3)*M(3,3)+M(3,4);

Po.fid.pnt=zeros(size(headshape.fid.pnt));
Po.fid.pnt(:,1)=headshape.fid.pnt(:,1)*M(1,1)+headshape.f
id.pnt(:,2)*M(1,2)+headshape.fid.pnt(:,3)*M(1,3)+M(1,4);
Po.fid.pnt(:,2)=headshape.fid.pnt(:,1)*M(2,1)+headshape.f
id.pnt(:,2)*M(2,2)+headshape.fid.pnt(:,3)*M(2,3)+M(2,4);
Po.fid.pnt(:,3)=headshape.fid.pnt(:,1)*M(3,1)+headshape.f
id.pnt(:,2)*M(3,2)+headshape.fid.pnt(:,3)*M(3,3)+M(3,4);
end
```

Programming code 4| Calculation of the barycentre of the point clouds.

```
function [ s ] = barycentre( pointcloud )
% This function computes the barycentre for a point cloud

x = pointcloud(:,1);
y = pointcloud(:,2);
z = pointcloud(:,3);
s(1,1)=sum(x)/length(x);
s(1,2)=sum(y)/length(y);
s(1,3)=sum(z)/length(z);
end
```

7.5 *Output data*

Each output variable contains different fields, which are described in this part.

7.5.1 Optimised pipeline

```
disp(mri)
          dim: [192 256 256]
      anatomy: [192x256x256 double]
          hdr: [1x1 struct]
    transform: [4x4 double]
     coordsys: 'ras'
         unit: 'mm'
        cfg: [1x1 struct]
```

dim: Dimension, number of voxels

anatomy: Matrix with anatomical information, anatomical images

hdr: File header

transform: Transformation matrix, transforms voxel to *head coordinate system*

coordsys: Name of *head coordinate system*

unit: Unit of *head coordinate system*

cfg: Configurations of the used function

```
disp(pol)
     pnt: [124x3 double]
     fid: [1x1 struct]
    unit: 'cm'
```

pnt: Cartesian coordinates of the head points

fid: Cartesian coordinates of the *fiducials* and their labels

```
disp(segmentedmri)
        dim: [256 256 256]
    transform: [4x4 double]
     coordsys: 'ctf'
         unit: 'mm'
        scalp: [256x256x256 logical]
          cfg: [1x1 struct]
```

scalp: Binary structure describing the head surface

```
disp (bnd)
     pnt: [200000x3 double]
     tri: [399996x3 double]
    unit: 'mm'
     cfg: [1x1 struct]
```

tri: Indices of points describing a triangle

```
disp (icp_output)
      TR: [3x3 double]
      TT: [1x3 double]
      ER: [101x1 double]
       t: [101x1 double]
    info: [1x1 struct]
```

TR: Rotation matrix

TT: Translation vector

ER: Mean square error for each iteration

t: Time

info: Additional information

7.5.2 Implemented function

The implemented function generates two output variables.

```
disp (matrix)
    0.0355   -0.9993   -0.0106   92.0913
    0.9036    0.0367   -0.4269  109.7019
    0.4270    0.0056    0.9042   77.7283
```

The first variable (matrix) contains the transformation matrix (as 4x4 matrix), the last column describes the translation while the others describe the rotation. The matrix can be used to adjust the MEG sensors to the MRI coordinate system.

```
disp (info)
          q_idx: [1x80 double]
          p_idx: [1x80 double]
           qout: [3x80 double]
           pout: [3x80 double]
            pin: [3x80 double]
    distanceout: [1x80 double]
     distancein: [1x80 double]
             ER: [201x1 double]
     interactiv: [1x1 struct]
```

The second variable contains diagnostic information

q_idx: Indices to the matching points of the head surface after registration

p_idx: Indices to the matching points of the point cloud after registration

qout: Positions of the matching points of the head surface

pout: Positions of the matching points of the point cloud after registration

pin: Positions of the matching points of the point cloud prior to registration

distanceout: Distance between the matching points after registration

distancein: Distance between the matching points prior to registration

ER: Mean square error for each iteration

interactive: Contains the same information (indicies, positions and distances) for an interactive registration

7.6 *Fiducial positions within the MR images*

The positions for the fiducials are stated for the *voxel coordinate system.*

	Proband A1457		
rpa	173	106	77
lpa	16	118	73
nas	103	217	122

	Proband A1458		
rpa	177	118	83
lpa	13	121	91
nas	100	222	136

	Proband A1468		
rpa	149	149	181
lpa	143	142	9
nas	37	91	100

	Proband A1469		
rpa	157	153	176
lpa	158	150	19
nas	55	120	99

	Proband A1470		
rpa	144	176	181
lpa	147	179	13
nas	32	136	98

	Proband A1472		
rpa	153	150	180
lpa	155	158	15
nas	48	128	98