

Creation of an Anatomical Artefact Atlas Based on HArtMuT

Bachelor Thesis

Moritz Wilhelm Steffin
403573

09.02.2023

Supervisor:
Prof. Dr. Benjamin Blankertz

Assistant Supervisor:
Prof. Dr. Stefan Haufe



Technische Universität Berlin
School of Electrical Engineering and Computer Science
Institute of Software Engineering and Theoretical Computer Science
Neurotechnology

Abstract

In the field of neuroscience, the task of source localization, that is, for a recorded electrical signal, determining the source within the human brain is one of a multitude of research problems.

Parcellation atlases have been found to be a useful tool when trying to map the result of source localization, a position in space, to the anatomical label at that exact position. They segment the human brain, into anatomical areas, allowing for easy determination of the anatomical object at a specific point in space. They therefore offer great usability in the task of further processing the results of source localization.

Such parcellation atlases have been produced a multitude of times and are in current use for source retrieval of the human cortical regions. However, as of now, there has not been a parcellation atlas including non-cortical head and neck artefacts. The inclusion of said artefacts, which usually consist of unwanted muscle and eye contributions in EEG recordings, offers a variety of benefits for Electroencephalogram (EEG) based source localization.

This thesis proposes the creation of such a parcellation atlas for non-cortical sources within the human head, using anatomical artefacts provided by the Head Artefact Model Using Tripoles (HArt-MuT) [Harmening et al., 2022]. The atlas' labelling is based on inside-outside segmentation using a ray casting approach offered by the vtk toolbox¹. Its performance is verified in the task of labelling source coordinates by comparing the provided results with labeled dataset.

The results show the atlas to offer a sufficiently good performance in the task at hand. The atlas achieved a matching rate of 90.8% when compared with the reference data set. Nonetheless, it is still being limited by the boundaries of parcellation atlases in general. The chosen implementation introduces some additional uncertainty, which however does not significantly impact the atlas' performance. As source localization on experimental EEG data already suffers from various outside sources of uncertainty, the precision and results delivered by the created atlas were found to be sufficiently precise in this context.

¹<https://vtk.org/>, accessed: 04.09.22

Kurzfassung

Die Aufgabe der sog. Quellenlokalisierung, das heißt, für ein aufgenommenes Signal den Ursprung eben jenes Signals im menschlichen Körper zu bestimmen, ist im Feld der Neurowissenschaften eine zentrale Forschungsfrage.

Sogenannten Parzellierungsatlanten sind ein hilfreiches Werkzeug für den Versuch, dem Ergebnis von Quellenlokalisierung, also einer Position im Raum, ein bestimmtes anatomisches Areal zuzuweisen. Sie segmentieren das menschliche Gehirn, in anatomische Areale, und sind dementsprechend perfekt anwendbar auf die oben genannte Aufgabe.

Solche Parzellierungsatlanten existieren bereits in Vielzahl für das menschliche Gehirn, doch Stand heute gibt es keinen solchen Atlas für die Parzellierung von anatomischen, nicht-kortikalen Artefakten im Augen-, Nacken-, und allgemeinem Kopfbereich. Die Einbindung solcher Artefakte, welche generell ungewollte EEG Daten aus Muskel- und Augenbeteiligungen sind, bietet eine breite Palette an Vorteilen für Quellenlokalisierung basierend auf Elektroenzephalogramm (EEG) Daten.

In dieser Ausarbeitung erstelle ich einen ebensolchen Parzellierungsatlas für nicht-kortikale Quellen im menschlichem Kopf, unter Benutzung der anatomischen Artefakte des “Head Artefact Model using Tripoles” (HArtMuT) [Harmening et al., 2022]. Die Segmentierung des erzeugten Atlas basiert auf einem auf Raycasting aufgebautem Algorithmus aus der vtk Toolbox². Die Performance des Atlas’ wird für die Aufgabe der Zuweisung anatomischer Areale für bereitgestellte Koordinaten durch einen Vergleich mit bereits existierenden Referenzdaten überprüft.

Der erzeugte Atlas zeigt eine gute Qualität im Bereich der ihm gestellten Aufgabe. Gleichzeitig kann er diese Aufgabe nur im Rahmen der für Parzellierungsatlanten generell geltenden Einschränkungen bzgl. ihrer Genauigkeit erfüllen. Durch die gewählte Umsetzung existieren weitere kleinere Ungenauigkeiten, die allerdings die Gesamtqualität des Atlas nicht signifikant negativ beeinflussen. Da Quellenlokalisierung auf experimentellen EEG Daten bereits in der Natur der Sache nicht exakt ist, und durch Ungenauigkeiten anderer Ursachen beeinträchtigt wird, ist die Leistung des erzeugten Atlas’ zur Anwendung vollkommen hinreichend.

²<https://vtk.org/>, accessed: 04.09.22

Contents

| | |
|--|-----------|
| 1. Introduction | 1 |
| 1.1. Neuroelectric Imaging and Source Dissociation | 1 |
| 1.2. HArtMuT | 1 |
| 1.3. Usage of (Brain) Parcellation Atlases | 2 |
| 1.4. Anatomical Artefact Atlas Based on HArtMuT | 2 |
| 1.5. Contributions | 2 |
| 1.6. Provision of the Atlas and Code Repository | 3 |
| 2. Fundamentals | 4 |
| 2.1. File Formats | 4 |
| 2.1.1. NIFTI File Format | 4 |
| 2.1.2. TRI File Format | 4 |
| 2.1.3. STL File Format | 4 |
| 2.2. Physiological EEG Artefacts | 5 |
| 2.3. Anatomical Spaces | 5 |
| 2.3.1. MNI Space | 5 |
| 2.3.2. Voxel and Voxel Space | 5 |
| 2.4. Biomagnetic Inverse Problem | 6 |
| 2.5. Source Localization in EEG | 6 |
| 3. Methods | 7 |
| 3.1. Working with NIFTI Files | 7 |
| 3.1.1. Usage of 'NiBabel' | 7 |
| 3.1.2. Data in NIFTI Files | 7 |
| 3.1.3. Voxel Space to MNI Space Transformation | 7 |
| 3.2. Labelling the Atlas | 8 |
| 3.2.1. Initial Approach | 8 |
| 3.2.2. Improved Approach | 13 |
| 3.2.3. Flow Chart | 17 |
| 3.3. Point in 3D-mesh Functionality | 19 |
| 4. Results | 21 |
| 4.1. Reference Data | 21 |
| 4.1.1. Labelling Comparison | 22 |
| 4.2. Integration into Existing Pipelines - 'atlasreader' | 22 |
| 4.3. Outcome | 23 |
| 4.3.1. General Performance - Match and Mismatch Rate | 23 |
| 4.3.2. Analysis and Correction of Reference Data | 23 |

| | | |
|-----------|---|-----------|
| 4.3.3. | General Performance - Match and Mismatch Rate after Reference Data Correction | 24 |
| 5. | Discussion | 25 |
| 5.1. | Interpretation of the Results | 25 |
| 5.1.1. | General Labelling | 25 |
| 5.1.2. | 'Deliberate' Mismatches | 25 |
| 5.1.3. | 'Unwanted' Mismatches | 25 |
| 5.1.4. | Reference Data | 26 |
| 5.2. | Limitations | 26 |
| 5.2.1. | Sources of Uncertainty | 26 |
| 5.2.2. | Parcellation Atlases in General | 27 |
| 5.2.3. | Implementation | 27 |
| 5.2.4. | Reference Data Issues | 29 |
| 6. | Conclusion | 30 |
| A. | Artefact-Integer-Mapping | 34 |

List of Figures

| | | |
|-------|--|----|
| 3.1. | General affine transformation as provided by the NIFTI1 file format | 8 |
| 3.2. | Transformation matrix of the NIFTI1 file in use for this parcellation atlas | 8 |
| 3.3. | Algorithm 1: basic labelling approach | 10 |
| 3.4. | Algorithm 2: findLabelForWorldSpaceCoord functionality | 10 |
| 3.5. | Mesh of the right Platysma muscle, with atlas voxels whose center points were determined to be inside the mesh by the basic labelling Algorithm 1 | 11 |
| 3.6. | Schematic depiction of an artefact mesh too thin to be properly displayed by the atlas created via Algorithm 1 | 12 |
| 3.7. | Mesh of the left Inferior Rectus muscle, with atlas voxels whose center points were determined to be inside the mesh by the basic labelling Algorithm 1 | 12 |
| 3.8. | Algorithm 3: advanced labelling approach | 15 |
| 3.9. | Algorithm 4: reiteration of Algorithm 3 | 16 |
| 3.10. | Mesh of the left Inferior Rectus muscle, with voxels determined to be labelled with its corresponding label by the improved labelling Algorithm 4 | 16 |
| 3.11. | Mesh of the left Inferior Rectus muscle, with voxels with either their center point, or one of their corner points found to be within the mesh, determined by Algorithm 1 and the extension provided by Algorithm 4 respectively | 17 |
| 3.12. | Flowchart presenting the implementation and interactions of the four presented algorithms creating the anatomical artefact atlas | 18 |
| 4.1. | The atlas' NIFTI file displayed via 'Brain Viewer' | 21 |
| A.1. | Artefact-Integer-Mapping introduced by the labels CSV file provided with this atlas | 34 |

List of Tables

| | |
|--|----|
| 4.1. Comparison of the atlas' labelling with the reference data labelling, depicting the number of total points considered, both the total amount of matches and mismatches, and match and mismatch rate respectively | 23 |
| 4.2. Analysis of the provided reference data, depicting the total amount of data points to be considered, and the total amount of data points found to be labelled correct and incorrect, with their respective rates, found by a manual review. | 23 |
| 4.3. Comparison of the atlas' labelling with the now corrected reference data labelling, depicting the number of total points considered, both the total amount of matches and mismatches, and match and mismatch rate respectively | 24 |

1. Introduction

1.1. Neuroelectric Imaging and Source Dissociation

Two of the most widely used methods in the field of neuroimaging are Magnetoencephalography (MEG) and Electroencephalography (EEG). Both of these methods can be used to non-invasively measure and map the magnetic field generated by the electrical activity of neurons. The signal recorded by EEG and MEG alike is a mixture of a multitude of different sources contributing to the recorded data [Harmening et al., 2022]. As of today, there exist a variety of techniques used to dissociate this data by individual sources [Makeig et al., 1995]. However, the mixture of recorded sources does, in general, not solely consist of signals originating from cortical sources, but further contains signals from sources beyond the brain. These interfering sources manifest in electrical (noise from other devices [Hassani and Karami, 2015]), mechanical (cable sway in EEG [Symeonidou et al., 2018]), as well as physiological forms [Harmening et al., 2022]. Physiological non-brain sources consist of, but are not limited to, eye movement and general facial and neck muscle activity outside the brain. Mechanical and electrical sources can often be identified by specific invariant patterns, that can be dissociated without much difficulty [Harmening et al., 2022]. However, the same is not true for eye movement and facial and neck muscle activity. Due to their location on the outside of the skull, and thus their proximity to the sensors used, at least in comparison with cortical structures shielded by the low-conductive skull, the signals generated by these sources can be magnitudes larger than those generated within the brain, and hence contribute strongly to an otherwise diminutive EEG/MEG signal [Harmening et al., 2022]. These non-brain sources often account for a significant part of the electrical activity picked up by an EEG/MEG recording [Joyce et al., 2004], and thus pose great difficulty when trying to extract signals originating from the brain. As of today, activity resulting from non-brain sources is generally considered to be artefactual, that is, considered to be an unwanted, faulty part of the recording, caused by the nature of the recording method at hand [Jiang et al., 2019]. They are therefore usually rejected from the recorded data by a variety of different artefact rejection methods [Kanoga and Mitsukura, 2017] proposed till date.

1.2. HArtMuT

Despite the general consideration of physiological non-brain activity to be artefactual, considering such data not as ill-posed recordings, but rather as explorable information, offers a variety of advantages. Facial and eye muscle activity has been found to be closely connected to cognitive processes, and could thus help evaluating brain activity [König et al., 2016]. Furthermore, the less strict rejection of trial data contaminated by artefactual noise poses the chance for less restricting experimental designs. The closer consideration of non-brain signal sources also improves the ability to perform source localization on MEG/EEG data in general [Harmening et al., 2022].

1.3. USAGE OF (BRAIN) PARCELLATION ATLASES

With said benefits in mind, the Head Artefact Model using Tripoles (HArtMuT) has been proposed as a model suitable for eye and head muscle simulation. And to additionally include eyes and facial and neck muscles in head model leadfields, simulating forward EEG signals. [Harmening et al., 2022].

1.3. Usage of (Brain) Parcellation Atlases

In the context of neuroimaging, brain parcellation atlases are often employed to resolve the results of source localization, a position in space, to an anatomical area. Such atlases usually consist of a data file, mainly in the NIFTI1/NIFTI2 format, generally containing a set of 3-dimensional data representing the human head, and a label file, containing a mapping of integers to anatomical structures, assigned to individual cells of the data, which represent anatomical structures. If the source localization has been performed in the same standard anatomical space, these atlases can be used to identify the anatomical region associated with the provided coordinate.

1.4. Anatomical Artefact Atlas Based on HArtMuT

In this thesis, I implemented such a parcellation atlas, not focussing on the parcellation of the brain, as it has been done a multitude of times before [Fan et al., 2016] [Grabner et al., 2006], but instead on the non-cortical anatomical artefacts provided by HArtMuT. In doing so, I hope to create an easy, low-level approach to source retrieval, that is for coordinates provided by source localization, determining the anatomical structure at that point in space, for non-brain sources using parcellation atlases in EEG data, providing the aforementioned benefits.

The following thesis will provide an overview of the creation of said anatomical artefact parcellation atlas based on HArtMuT, look into the theoretical background, and test its performance in the task of source retrieval. The testing is done by comparing the atlas' results to a data set of already labelled coordinates.

1.5. Contributions

My contributions are as follows:

1. **Creation of an anatomical artefact parcellation atlas based on HArtMuT**, that offers the possibility to automatically label points in space for non-cortical artefacts, provided by HArtMuT
2. **Verification of the goodness of the created atlas**, by comparing labelling results provided by the atlas, with a set of previously labelled data points
3. **Testing the atlas' compatibility with the standard module 'atlasreader'**, by employing said module in the process of testing the anatomical artefact atlas

1.6. Provision of the Atlas and Code Repository

With this thesis, I provide the created parcellation atlas consisting of a NIFTI1 file, and a CSV file specifying the integer artefact mapping used for the atlas. Both of these can be found in the repository delivered with this thesis¹. I also provide the codebase used for the creation of the atlas, and a variety of additional files used in both the creation and testing of this atlas, including the artefact meshes used for the atlas' parcellation and the set of test data used for the verification of this atlas' labelling.

¹Repository: <https://github.com/Moritz209/atlascreation>

2. Fundamentals

Before further looking at the creation of the artefact atlas based on HArtMuT, I will lay some theoretical groundwork, and introduce some basic concepts and definitions reoccurring throughout this thesis.

2.1. File Formats

A variety of different file formats are employed in the process of creating the anatomical artefact atlas, which will be introduced in the following.

2.1.1. NIFTI File Format

The NIFTI (Neuroimaging Informatics Technology Initiative) file format is a widespread format used for neuroimaging, and hence the storage of medical images. It has been introduced in the early 2000s by the name-giving research group at the 'National Institutes of Health' in the USA, with the goal of providing a modern approach to data formats in neuroimaging. It builds upon its predecessor the 'Analyze' file format, by expanding it with additional fields, mainly focused on the image orientation in space [Knipe and Moore, 2019, Winkler, 2012]. There currently exist two versions of the NIFTI file format, NIFTI1 and NIFTI2. NIFTI2 offers the same amount of information as the NIFTI1 file format, with some minor advancements provided in the area of multi-dimensional data points. In this thesis, whenever a non-specified NIFTI file is referenced, it shall be assumed to be in NIFTI1 file format.

2.1.2. TRI File Format

A file format containing polygonal meshes for the human head. Usually consists of a header line, a vertex section, and a triangle section where the meshes individual triangular faces are defined based on the vertices in the vertex section [Bourke, 2001]. The anatomical artefacts provided by HArtMuT are in TRI format.

2.1.3. STL File Format

A standard interface usually used for 3D printing and similar modes of construction. It is used to describe the surface of polyhedrons via triangular faces. In this thesis, it is used as an input format for some third party python modules in use during the creation of the atlas, as it is more widespread than the aforementioned TRI file format, in the context of 3D imaging.

2.2. Physiological EEG Artefacts

Physiological EEG artefacts are generally considered to be unwanted, and faulty parts of a recording, being caused by the specific means of EEG recordings. They are usually subject to filtering using a set of versatile artefact rejection methods [Kanoga and Mitsukura, 2017].

In the context of this thesis however, anatomical artefacts shall denote the 63 non-cortical artefacts provided by HArtMuT. These 63 artefacts are made up of a variety of anatomical structures located within the human head and neck. They mainly consist of muscular artefacts (mostly head and neck muscles), and are complemented by some additional non-muscular anatomical structures (mainly parts of the human eye). More precisely, anatomical artefacts provided by HArtMuT shall refer to the TRI files representing 3-dimensional meshes of said 63 artefacts. The artefact meshes are located within the standardized Montreal Neurological Institute (MNI) space.

A comprehensive list of the HArtMuT artefacts used throughout the creation of this parcellation atlas may be found in Appendix A (also including an integer-artefact-mapping used in the creation of the atlas). The aforementioned corresponding TRI files are also provided within this thesis' repository.

2.3. Anatomical Spaces

2.3.1. MNI Space

In neuroimaging, there exist multiple predefined standard coordinate spaces. These standard systems are used to access the location of brain structures independently from individual anatomies leading to small differences in size and shape of every human brain. One of these coordinate systems is the MNI space. It has been introduced by the 'Montreal Neurological Institute' (MNI) as an average of a multitude of Magnetic Resonance Imaging (MRI) scans, with the goal of reflecting the average human neuroanatomy. The MNI space defines boundaries around the brain, whilst also providing locations of landmarks, from a fixed point of origin [Brett et al., 2002]. When creating a parcellation atlas it is generally advisable to create it based on one of the standard coordinate spaces of neuroimaging. This provides the benefit of lower effort usability, due to not requiring any transformation into another arbitrary system when querying the atlas. As the artefacts provided by HArtMuT are defined in MNI space, the atlas based on them will also be in MNI space. Therefore, MNI space coordinates may directly be queried without necessitating any transformation.

2.3.2. Voxel and Voxel Space

2.3.2.1. Voxel

A voxel is the representation of a value on a regular grid in 3-dimensional space¹. It is the 3-dimensional equivalent to a pixel in 2-dimensional images. Voxels are usually referenced by a 3-tuple, providing coordinates for all three dimensions. Such a referencing 3-tuple shall be denoted as (i,j,k). The significant difference between a voxel and a 3-dimensional coordinate is the fact that a voxel possesses a volume, and hence, does not specify an individual point in space, but rather a specific polyhedron.

¹<https://en.wikipedia.org/wiki/Voxel>, accessed: 01.09.2002

2.4. BIOMAGNETIC INVERSE PROBLEM

2.3.2.2. Voxel Space

Voxel space shall refer to the regular grid provided by a NIFTI file's data. This data is generally realized by 3-dimensional arrays, and can be accessed by a 3-tuple of coordinates as described in Section 2.3.2.1. The volume of each specific voxel is also predefined within a NIFTI files' header. A 3-tuple used to access a NIFTI files' data shall be denoted as a voxel space coordinate.

2.4. Biomagnetic Inverse Problem

Bodily activity such as processes in nerves, and in this specific case, neurons, generates a minute electrical activity. This electrical activity and the resulting electromagnetic fields may be measured using a variety of different techniques. For the brain, such measurements are often done using EEG. As EEG is a non-invasive method, it can only directly measure electrical activity on the surface of the human scalp. The reconstruction of the electrical activity within the body, from the recordings on the surface of it, is called the 'Biomagnetic Inverse Problem' [Lucka, 2009]. As EEG is one of the key data sources in the field of neuroimaging, and the atlas will mainly be used to resolve source locations on EEG data, this shall also be denoted as the 'Inverse EEG Problem'.

2.5. Source Localization in EEG

Source localization on EEG recordings is the attempt of identifying a specific location or point within the brain where a recorded EEG signal originated. To further elaborate, it is the localization of the specific neural activity within the human brain, that is measurable by sensors placed on the human head and recorded via EEG. Reliable source localization offers a variety of benefits including improved treatment capabilities for neurological illnesses originating in the brain (e.g. epilepsy), and a generally improved understanding of information flow, and the brain itself [Michel and He, 2019].

3. Methods

3.1. Working with NIFTI Files

3.1.1. Usage of 'NiBabel'

Reading from, and writing to NIFTI files is a task that is not unique to this atlas' creation. Therefore it has previously been implemented. The 'NiBabel' python package provides extended functionality in reading and writing medical and neuroimaging file formats [Brett et al., 2022]. For this thesis, it is mainly used for reading and writing NIFTI1 files, which constitute the data part of the anatomical artefact atlas. They contain the data representing the space in which the human head is located, and which is modified and labelled according to the given HArtMuT artefacts.

3.1.2. Data in NIFTI Files

As previously mentioned, a NIFTI files' data is, dependent on the task at hand, most often realized as a 3-dimensional data array. This data array is the voxel space of the atlas, with voxels referenced to by the aforementioned (i,j,k) triplet of voxel coordinates. Exact specifications for this data, like the data type and shape may be found in the header of a NIFTI file.

3.1.3. Voxel Space to MNI Space Transformation

While voxel coordinates provide a unique identifier of a voxel within the 3-dimensional data of a NIFTI file, these coordinates are only of use in reference to a specific NIFTI files' data.

In medical- and neuroimaging, recorded images may occur in a multitude of different world coordinate spaces. These coordinate spaces may be based on the specific scanner the image was recorded on, aligned to another file, or aligned to one of the multiple standard spaces, like the Talairach or MNI space. To be able to transfer such recorded images into the voxel coordinate space, and hence the data of a NIFTI file, is crucial for the creation of this atlas.

The NIFTI file format provides multiple methods for transforming between voxel and world coordinate space. The applicability of these methods is based on the origin of the world coordinate space of the data provided. The method mainly used for the creation of this anatomical artefact atlas, is based on a full affine transformation. It is the method provided by the NIFTI file specification [Knipe and Moore, 2019] to be used for transforming to some standard world space, in the atlas' case, to MNI world coordinates.

The NIFTI file specific affine transformation matrix is provided within the NIFTI files' header, in the *srow_x*, *srow_y* and *srow_z* fields. The transformation consists of a standard matrix-vector-multiplication.

3.2. LABELLING THE ATLAS

As per convention, (x,y,z) shall denote world coordinates, and (i,j,k) shall denote voxel coordinates respectively.

$$\begin{aligned}x &= srow_x[0]*i + srow_x[1]*j + srow_x[2]*k + srow_x[3] \\y &= srow_y[0]*i + srow_y[1]*j + srow_y[2]*k + srow_y[3] \\z &= srow_z[0]*i + srow_z[1]*j + srow_z[2]*k + srow_z[3]\end{aligned}$$

Figure 3.1.: Transformation matrix as provided by the NIFTI1 file format specification, with (i,j,k) denoting voxel space coordinates, and (x,y,z) denoting world space coordinates
From 'THE NIFTI-1 DATA FORMAT' by Robert H. Cox, 2004 [Cox, 2004]

In general, most NIFTI transformation matrices will provide non-zero values for $srow_x[0]$, $srow_y[1]$ and $srow_z[2]$.

$$\begin{bmatrix} 1.5 & 0 & 0 & 90 \\ 0 & 1.5 & 0 & -126 \\ 0 & 0 & 1.5 & -190 \end{bmatrix}$$

Figure 3.2.: Transformation matrix of the NIFTI file in use to transform from voxel space to world space for the creation of this anatomical artefact atlas, realized via the “to_world_coordinate_space” method in the repository

3.2. Labelling the Atlas

The most important step in creating an anatomical artefact atlas is the labelling of every data cell within the atlas’s NIFTI file with a specific label, indicating either the presence of a specific or the absence of any artefact at this point in space.

To store a corresponding label to a specific artefact within the atlas, I have created a very simple mapping of artefacts to integer values. This mapping is provided as the second part of the atlas (completed by the NIFTI file), in a labels CSV file, and maps all 63 HArtMuT artefacts, and the additional 'empty Space' label, indicating the absence of any of the 63 artefacts at a specific point in space, to an integer value. The exact integer-artefact-mapping of the atlas created for this thesis may be found in Fig. A.1 of appendix A.

The labelling of the atlas was done via two different approaches, the second one building upon the first one. The approach employed initially uses a very simple and straightforward method to label the atlas’ voxels. The improved approach is based on this initial approach, but fixes some of the problems arising with the basic approach.

3.2.1. Initial Approach

The pseudo code for my initial labelling approach is provided in Algorithm 1 with line numbering referenced in this written explanation. The line numbered pseudo code for the 'findLabelForVoxelSpaceCoord' functionality referenced in Algorithm 1, is also provided in Algorithm 2.

To perform the labelling, an empty array with the desired atlas shape of (121, 145, 199) (provided by the chosen base NIFTI file) is iterated (Alg. 1, ll. 1-3). Every single data cell of said array is

3.2. LABELLING THE ATLAS

uniquely identified by a triplet (i,j,k) . This triplet denotes the center point of an individual voxel, as defined by the NIFTI1 file format definition [Knipe and Moore, 2019]. As the array is the atlas' voxel space, this (i,j,k) triplet shall be denoted as the voxel space coordinate. For each of these triplets, the corresponding world space coordinate triplet (x,y,z) is calculated, using the transformation matrix provided by the atlas' NIFTI file (see Fig. 3.2) (Alg. 1, ll. 4). Next, the aforementioned 'findLabelForVoxelSpaceCoord' functionality is called, determining the correct label for a specific world space coordinate (Alg. 1, ll. 5).

Algorithm 2 then, being provided a single coordinate in world coordinate space, iterates through all anatomical artefacts (Alg. 2, ll. 1). For each artefact, a bounding box is calculated, and it is verified that the coordinate provided lies within this bounding box (Alg. 2, ll. 2). If this is the case, the algorithm proceeds with this specific artefact, otherwise it continues with the next provided artefact. Using this approach managed to reduce the times actually needed to check if a coordinate is located within an artefact mesh, from about 220 million (3.5 million voxels \cdot 63 Artefacts), to roughly 3.4 million (the sum of the total voxels contained in each artefact bounding box) calculations, leading to a significant decrease in the required computational power and time to create this atlas. If the algorithm proceeds with the artefact, it then determines whether the coordinate lies within the considered artefact (Alg. 2, ll. 3). This specific calculation will be looked at in more detail in Section 3.3. If the coordinate was found to not be inside the specific artefact mesh, the algorithm continues with the next provided artefact. If the coordinate is not found to be within any of the provided artefacts, the label 0 is returned, indicating 'empty Space', as per our mapping. On the other hand, if the coordinate was found to be within the specific artefact mesh, the specific integer label referencing to the specific artefact is retrieved (Alg. 2, ll. 4), and then returned (Alg. 2, ll. 5).

3.2. LABELLING THE ATLAS

Algorithm 1: Labelling the atlas data

Result: 3-dimensional, labelled atlas data array

```
1 for  $i$  in range ( $atlasDataShape[0]$ ) do
2   | for  $j$  in range ( $atlasDataShape[1]$ ) do
3   |   | for  $k$  in range ( $atlasDataShape[2]$ ) do
4   |   |   | worldSpaceCoord = worldSpaceFromVoxelCoords(( $i,j,k$ ))
5   |   |   | atlasData[ $i$ ][ $j$ ][ $k$ ] = findLabelForWorldSpaceCoord(worldSpaceCoord)
6   |   | end
7   | end
8 end
```

Figure 3.3.: Labelling the atlas' data array by iterating it, transforming voxel space coordinates to world space coordinates, and determining their specific label.

Algorithm 2: Assigning a specific label to a coordinate

Result: label for input world space coordinate (x,y,z)

```
1 for  $artefact$  in artefactFiles do
2   | if  $inArtefactBoundingBox((x,y,z), artefact)$  then
3   |   | if  $isCoordInArtefactMesh((x,y,z), artefact)$  then
4   |   |   | label = getLabelForArtefact(artefact)
5   |   |   | return label
6   |   | end
7   | end
8 end
9 return 0
```

Figure 3.4.: For a specific coordinate (x,y,z) in world space, determine the corresponding label by iterating all artefacts, checking if the coordinate lies within the artefact's bounding box, and then further checking if the coordinate is within the artefact mesh. If a match is found, return the artefact specific label. If no label could be found after iterating all artefacts, return 0, the 'empty Space' label.

3.2. LABELLING THE ATLAS

3.2.1.1. Issues with the Initial Approach

An obvious problem arises, when trying to label a voxel, hence a 3-dimensional construct with a non-zero volume in 3-dimensional space, by checking whether its center point (the (i,j,k) voxel coordinate) lies within any of the artefact meshes provided by HArtMuT. A common occurrence, especially near the surface of the artefact meshes, is a voxel whose center point is not within any artefact mesh, but whose total volume may very well intersect with one or multiple meshes.

Impact on 'large' artefact meshes

This is a minor problem for large, compact artefact meshes, as they will intersect with a significant amount of voxel center points, and will therefore be represented relatively well within the created atlas. This is shown exemplary in Fig. 3.5.

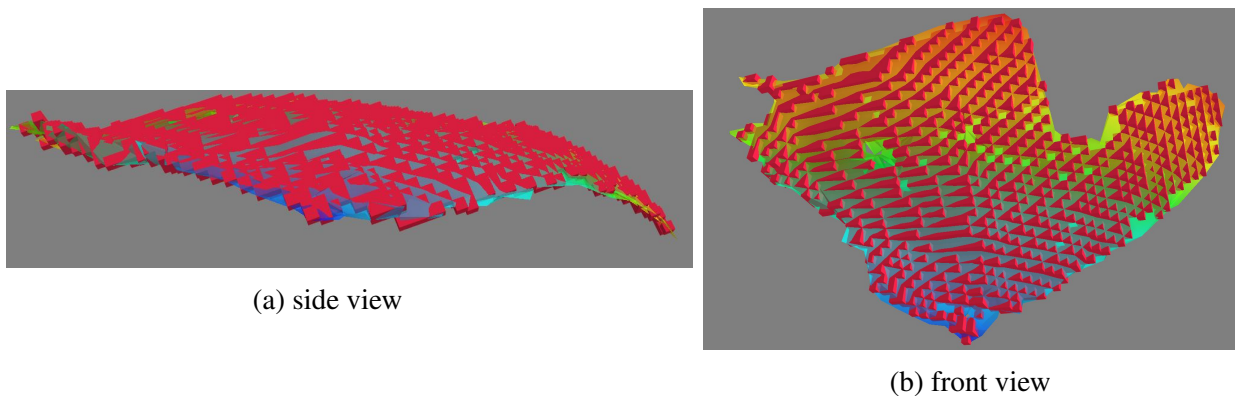


Figure 3.5.: Mesh of the right Platysma muscle (green-blue) in side (a) and front (b) view compared to the atlas' voxels (red boxes) whose center points were determined to be within the mesh by Algorithm 1. It can be seen that with the exception of minor parts, the artefact mesh is well covered by corresponding atlas voxels.

It should be noted, that the figure below has been created in an individually created atlas not depicting any other anatomical artefacts, to increase the descriptiveness of the figure. Plotting done via Mayavi (3D visualization toolkit) [Ramachandran and Varoquaux, 2011]

It can be seen that whilst there are still some minor parts of the artefact mesh visible without any voxel coverage, in general it is quite neatly covered in (red) atlas voxels labelled accordingly by Algorithm 1. The depicted voxels accurately display the shape of the artefact mesh.

Impact on 'small' artefact meshes

For smaller, thinner artefact meshes however, the aforementioned problem poses a significant issue. An artefact thinner than a voxel's edge length, may very well be located right between two planes of voxels (as depicted in Fig. 3.6), and therefore will not accurately be represented in the atlas created by Algorithm 1.

3.2. LABELLING THE ATLAS

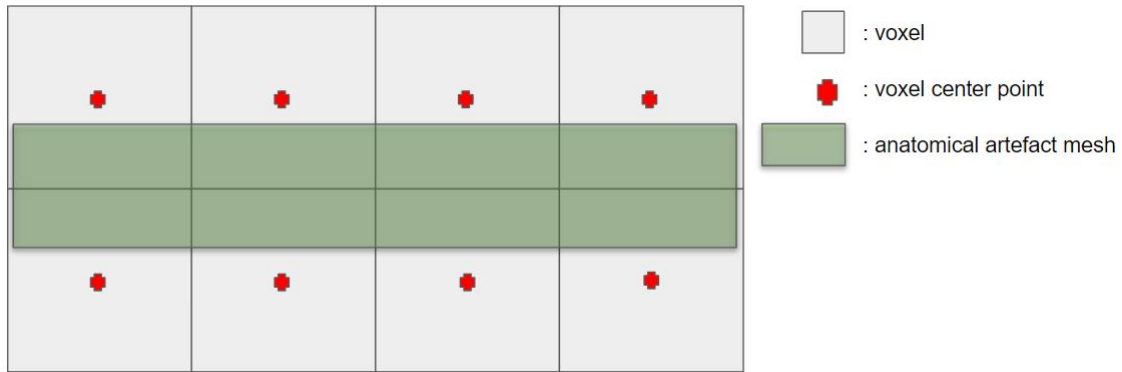


Figure 3.6.: Schematic 2-dimensional depiction of a thin artefact mesh, without proper 'representation' in the atlas, due to no voxel center point being located within the mesh, and therefore no voxels being labelled with the specific artefact label as per Algorithm 1

In an hypothetical worst case scenario, this might lead an entire artefact to not be included in the atlas. This would be the case, if an artefact is so thin and unfortunately located in space, that no single voxel's center point is located within its mesh. However, the majority of the artefacts used are large enough to not significantly suffer from this issue. Nonetheless, the more filigree and thin artefacts, especially small anatomical structures located around the eyes, are significantly negatively impacted by this issue regarding their display within the atlas. An example of this may be seen in Figure 3.7.

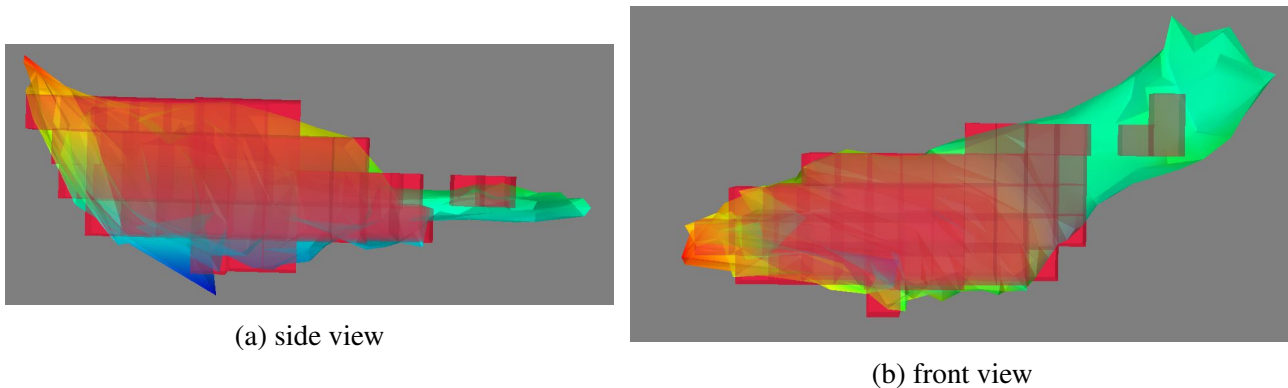


Figure 3.7.: Mesh of the left Inferior Rectus muscle (green-blue) in side (a) and front (b) view compared to the atlas' voxels (red boxes) whose center points were determined to be within the mesh by Algorithm 1. It can be seen, that larger parts of the artefact mesh, especially near the thinner right hand side of the mesh, are not sufficiently covered by corresponding atlas voxels.

It should be noted, that the figure below has been created in an individually created atlas not depicting any other anatomical artefacts, to increase the descriptiveness of the figure. Plotting done via Mayavi (3D visualization toolkit) [Ramachandran and Varoquaux, 2011]

It can be seen that the thin right hand side of the left muscle Inferior Rectus mesh is only sparsely populated by voxels whose center points were found to be within the mesh. This is due to the issue depicted in Fig. 3.6. Whilst the artefact mesh is still partially covered in (red) atlas voxels, the voxels

3.2. LABELLING THE ATLAS

display a significantly altered shape, as they do not accurately cover the right hand side of the artefact mesh.

3.2.2. Improved Approach

There exist multiple different approaches suitable to tackle the labelling problem presented in Section 3.2.1.1. The most precise method would require the calculation of every intersection of every voxel with each individual anatomical artefact, and then further proceeding to compare the volumes of the bodies created by said intersections. The voxel would then be labelled as the artefact whose mesh makes up most of the voxels non-empty volume. If no intersection would be found the voxel would be labelled 'empty Space'. As the calculation of an intersection between a voxel's mesh and an irregular artefact mesh is a non-trivial task, significantly increasing both the computational requirements, and the atlas' creation time, I have decided to limit this thesis' complexity, by settling with a less complicated, heuristic approach. I will later on discuss whether the precision provided by said approach is appropriate for the task at hand. Nonetheless, it should be noted, that when desiring completely correct labelling in all cases (not that a definition for correct is not trivial for the creation of a parcellation atlas), an implementation of the aforementioned voxel-mesh-intersection approach would be required.

Chosen approach

To get a more precise representation of the artefact meshes within the atlas, especially near the artefact mesh surfaces, and for thin, small artefacts, an advanced labelling approach is proposed. It is based on not only labelling voxels whose center points were found to be within an artefact mesh, but to also take into account corner points within meshes for voxels located next to voxels assigned to an artefact (and not empty space) label by the basic Algorithm 1.

The pseudo code for this advanced labelling approach is provided in Algorithm 3 with line numbering references in this written explanation. This algorithm is supposed to be run on the atlas' NIFTI file after the initial labelling via Algorithm 1 is done, and therefore advances the initial labelling, instead of fully replacing it.

The advanced labelling approach reiterates the 3-dimensional, now labelled, atlas data array (Alg. 3, ll. 1-3). For every single triplet (i,j,k) , it is checked whether the data cell of the array specified by this triplet, contains the value 0 (Alg. 3, ll. 4). This value, per the integer-artefact-mapping depicted in Appendix A, denotes empty space, hence, the specific voxel's center point was found to not be within any of the HArtMuT provided artefacts. If this is not the case, the algorithm proceeds with the next data cell, otherwise, for the voxel specified by the (i,j,k) triplet, (i',j',k') triplets are calculated, referencing all surrounding voxels within the atlas's data (Alg. 3, ll. 5). The algorithm further iterates through all these surrounding voxels found (Alg. 3, ll. 7), and for each one, a tuple (label, count = 0) is added to a previously created set (Alg. 3, ll. 6). The label in this tuple shall be the label of the specific surrounding voxel. This set is used to keep track of all different labels found in voxels neighbouring the initial (i,j,k) voxel.

Next, all corner points for the initial (i,j,k) voxel are calculated, and then transformed from voxel into world space. The algorithm proceeds to iterate them (Alg. 3, ll. 10). For each corner point, all artefacts previously added to the artefact set as tuples are iterated (Alg. 3, ll. 12). It is checked if the

3.2. LABELLING THE ATLAS

current corner point is located within the specific artefact (Alg. 3, ll. 13), and if that is the case, the artefact's tuple count value is increased by one and the loop is exited. This is done for all artefacts in the set. After iterating through all artefacts, for all corner points, the algorithm proceeds with the actual labelling. From the previously created artefact set, containing tuples with distinct artefact labels and a count value denoting often the specific artefact mesh contained one of the (i,j,k) voxel's corner points, a list is extracted containing all those tuples with a maximized count value. That is, all those artefact labels are extracted, found most often to contain a corner point within their respective meshes (Alg. 3, ll. 20). This method returns an empty list, if no artefact label has a count value larger than zero. In such a case, the algorithm continues with the next data cell (Alg. 3, ll. 21-22). If the length of this list is one (Alg. 3, ll. 24), hence, a simple majority of corner points for the (i,j,k) voxel were found to be within one specific artefact, the labelling is obvious, and the (i,j,k) voxel is labelled accordingly (Alg. 3, ll. 25). If there are multiple entries in the list, the label whose respective artefact is the smallest volume wise is chosen, and the (i,j,k) voxel labelled accordingly (Alg. 3, ll. 27). This is done to improve the representation of smaller artefacts in the atlas, as they suffer the most from the issue depicted in Fig. 3.6.

Reiteration of Algorithm 3

To achieve the best results, the reiteration described by Algorithm 3 is repeated multiple times. This is necessary, as the focus on voxels neighbouring on already labelled voxels restricts the algorithm from labelling all voxels with corner points within the mesh, if they are not neighbouring a labelled voxel after the initial Algorithm 1. These iterations are repeated, until the labelling no longer changes. The process of this repetition is shown in Algorithm 4.

Algorithm 3 is run once initially (Alg. 4, ll. 2), after storing the atlas' data in a temporary variable (Alg. 4, ll. 1). After this initial run, it is checked whether there have been changes in the atlas' data (Alg. 4, ll. 3), before repeating the process (Alg. 4, ll. 4-5).

It is important to note, that for an artefact not picked up by the initial labelling performed via Algorithm 1, that is, if no voxel center points were determined to be within the artefact mesh, the advanced labelling, due to the focus on empty voxels neighbouring already labelled voxels, will not include this artefact either. This is however, only the case for extremely small artefacts, and can therefore be deemed irrelevant. A further optimized version not implemented by this thesis could calculate corner point intersections with the artefacts not picked up by the initial labelling for all atlas voxels, however, it would need to be researched how this would affect the atlas' creation time.

3.2. LABELLING THE ATLAS

Algorithm 3: Reiterate atlas data to label voxels with corner points in artefact meshes

```

Result: 3-dimensional, labelled atlas data
1 for  $i$  in range ( $atlasDataShape[0]$ ) do
2   for  $j$  in range ( $atlasDataShape[1]$ ) do
3     for  $k$  in range ( $atlasDataShape[2]$ ) do
4       if  $atlasData[i][j][k] == 'emptySpace'$  then
5          $surroundingVoxel = getSurroundingVoxel((i,j,k))$ 
6          $artefactSet = set()$ 
7         for  $v$  in  $surroundingVoxel$  do
8            $artefactSet.append((v.label, 0))$ 
9         end
10        for  $p$  in  $worldSpaceFromVoxelCoordsList(cornerPoints((i,j,k)))$  do
11           $count = 0$ 
12          for  $artefact$  in  $artefactSet$  do
13            if  $pointInArtefactMesh(p, artefact)$  then
14               $artefactSet[count][1] = artefactSet[count][1] + 1$ 
15              break
16            end
17             $count = count + 1$ 
18          end
19          end
20           $maxArtefactList = getMaxFromArtefacts(artefactSet)$ 
21          if  $len(maxArtefactList) == 0$  then
22            continue
23          end
24          if  $len(maxArtefactList) == 1$  then
25             $atlasData[i][j][k] = getLabelForArtefact(maxArtefactList[0])$ 
26          else
27             $atlasData[i][j][k] = getLabelForSmallestArtefact(maxArtefactList)$ 
28          end
29        end
30      end
31    end
32  end

```

Figure 3.8.: Reiterating the atlas' data to label 'empty Space' voxels neighbouring already labelled voxels, if one of their corner points is within an artefact mesh. If corner points are contained in different artefact meshes, choose the artefact containing most. If there is no artefact containing most, choose the smallest artefact volume wise.

3.2. LABELLING THE ATLAS

Algorithm 4: Repetition of Algorithm 3 until it produces no more change

Result: 3-dimensional, labelled atlas data

```
1 atlasDataTmp = atlasData
2 Algorithm3(atlasData)
3 while atlasDataTmp != atlasData do
4   | atlasDataTmp = atlasData
5   | Algorithm3(atlasData)
6 end
```

Figure 3.9.: Repetition of the advanced labelling approach depicted in Algorithm 3 until it produces no more change, that is, until no more labels are changed in the atlas' data.

Figure 3.10 shows the result of the improved labelling Algorithm 4 on the example of the left Inferior Rectus muscle. For this depiction, only voxels not initially labelled with the specific artefact's label by Algorithm 1 are plotted, to further emphasize the gain in representation.

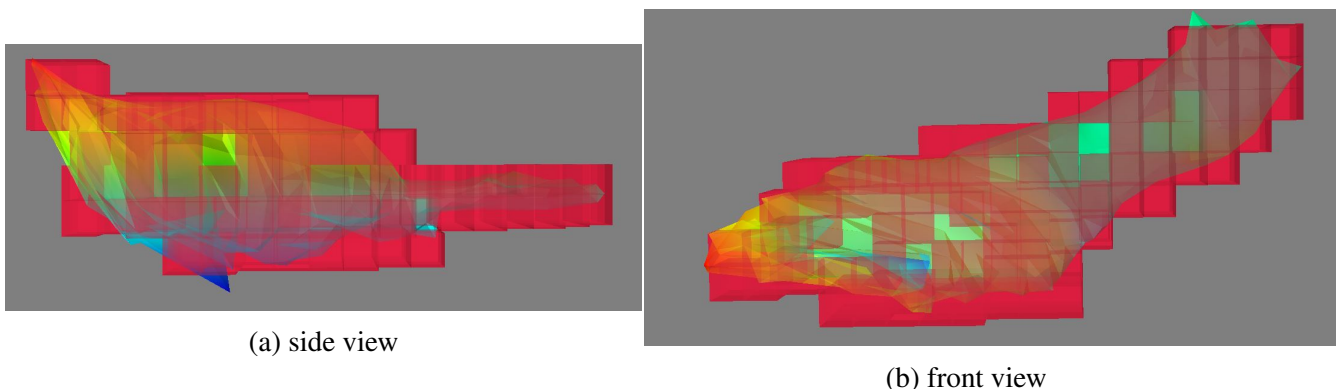


Figure 3.10.: Mesh of the left Inferior Rectus muscle (green-blue) in side (a) and front (b) view compared to the atlas' voxels (red boxes) whose corner points were determined to be within the mesh by Algorithm 4.

A significant increase in the artefact coverage by atlas voxels can be seen, when compared to the result provided by Algorithm 1 (see Fig. 3.7).

The above depiction shows only voxels determined to have corner points in the artefact mesh by Algorithm 4, excluding voxels already determined to be within the mesh by the basic labelling done via Algorithm 1.

It should be noted, that the figure below has been created in an individually created atlas not depicting any other anatomical artefacts, to increase the descriptiveness of the improvements made.

Plotting done via Mayavi (3D visualization toolkit) [Ramachandran and Varoquaux, 2011]

The figure above clearly shows the improved representation of the muscle artefact to be seen within the created atlas. This is especially obvious at the thin right hand side of the artefact depicted, when comparing Fig. 3.10a with the initial labelling approach depicted in Fig. 3.7a. Whilst Fig. 3.7a

3.2. LABELLING THE ATLAS

significantly lacks voxels with center points found to be within the mesh near the thinner right-hand part of the artefact, the improved approach extends the few voxels already labelled by Algorithm 1 by 2 planes of voxels whose corner points very clearly intersect with the mesh.

The merging of the results from Fig. 3.7 and Fig. 3.10 achieves a relatively high quality atlas voxel representation of the depicted artefact mesh, using the presented low effort, heuristic approach. The result can be seen in Fig. 3.11.

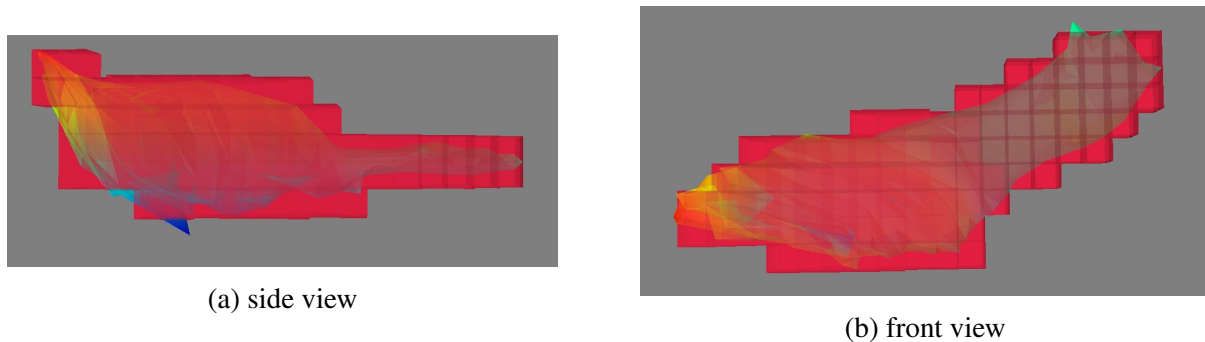


Figure 3.11.: Mesh of the left Inferior Rectus muscle (green-blue) in side (a) and front (b) view compared to the atlas' voxels (red boxes) determined to be labelled accordingly by either the basic labelling Algorithm 1, or the improved labelling Algorithm 4. The artefact mesh is almost fully covered by atlas voxels, with some small details still not being fully contained.

It should be noted, that the figure below has been created in an individually created atlas not depicting any other anatomical artefacts, to increase the descriptiveness of the improvements made.

Plotting done via Mayavi (3D visualization toolkit) [[Ramachandran and Varoquaux, 2011](#)]

From the above depiction it is obvious that the mesh is not fully covered by voxels even after merging both Algorithm 1 and Algorithm 4, and hence, is not fully and perfectly depicted by the atlas' voxels. This was to be expected by using a non-perfect, heuristic approach. However, the offered accuracy turned out to be sufficient for the task at hand. It will be further discussed why a certain imperfection is not a significant problem in Chapter 5.

3.2.3. Flow Chart

To offer a better understanding of the interaction of the various presented algorithms, and to get a final overview of the general process at hand for creating the anatomical artefact atlas, a flow chart is provided unifying the aforementioned algorithms, on a slightly higher lever of abstraction compared to the previously shown pseudo code.

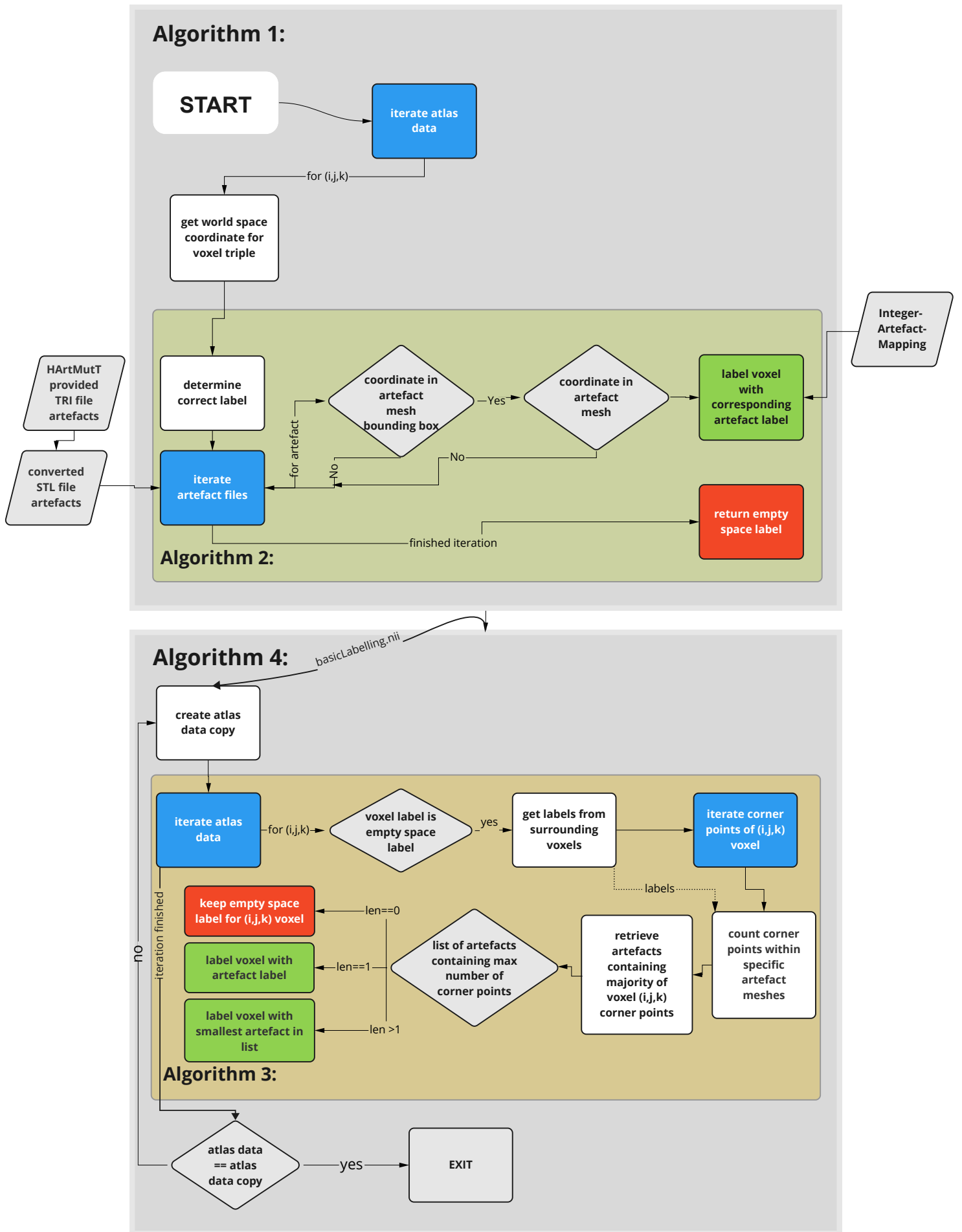


Figure 3.12.: Flowchart presenting the implementation and interactions of the four aforementioned algorithms creating the anatomical artefact atlas

3.3. Point in 3D-mesh Functionality

A core aspect of creating the anatomical artefact atlas based on HArtMuT, consists of transferring the provided triangular artefact meshes into the atlas' voxel space. The key part of this lies within the functionality determining whether a specific point lies within a provided artefact mesh. This is a 3-dimensional generalization of the Point-in-Polygon (PiP) problem. There exist two general approaches to tackle this problem in 2-dimensional space. The first approach being the 'even-odd' concept, realized by a ray casting approach, and the second approach being the non-zero winding number concept [Hormann and Agathos, 2001]. Whilst a ray casting approach can be transferred into 3-dimensional space more directly, the winding number approach requires some more extensive generalization to be used for inside-outside segmentation [Jacobson et al., 2013].

The following will shortly look into both these concept's general idea, and then further explain the decision made for the implementation in this thesis.

Ray Casting Approach

Ray Casting for inside-outside segmentation on a polyhedron P for a point to test T employs the simple idea that when casting a ray R from T into any arbitrary direction, and further counting the intersects between the ray R and the faces of the polyhedron P , the point T is located within the polyhedron exactly if the number of intersects is odd. This is based on the observation that for every intersect of the ray R with the polyhedrons' faces the location of T toggles between being inside and outside of the polyhedron P , with the ray R either just entering (even intersects) or just leaving (odd intersects) the polyhedron.

The actual implementation of this approach, for both 2-dimensional and 3-dimensional applications, needs to tackle multiple edge cases arising, like the question of how to count an intersection of the ray R directly with a vertex of the polygon/polyhedron.

A more in depth approach on the application of ray casting for a point-in-polyhedron test and optimizations regarding computational complexity has been published by D. Horvat in 2012 [Horvat, 2012].

Winding Numbers Approach

In their work on the 3-dimensional generalization of winding numbers, a concept mentioned as early as the 18th century [Grünbaum and Shephard, 1990], for inside-outside segmentation, Jacobson et al. in 2013 [Jacobson et al., 2013] defined the winding number as the 'signed length of the projection of C [an oriented, closed Lipschitz curve] onto the unit circle around p [the point to test] divided by 2π ' (p. 3). To provide a better intuition, the winding number may be interpreted as the number of full rotations the curve C , or an imaginary object travelling on C in the direction of the provided orientation, travels around p . Full clockwise rotations decrease this count, whilst full counter-clockwise rotations increase it¹.

Applied on the case of inside-outside segmentation, a point in 2-dimensional space is within a polygon defined by a closed curve C , if the winding number of p with respect to C is not equal to zero. Or with the intuitive description, the curve C rotates around the point p at least once in either clockwise or counter-clockwise orientation (theoretically however, if the number of clockwise and counter-clockwise rotations is the same, the point p is outside of the polygon defined by the curve C).

As mentioned above, more details on the generalization into 3-dimensional space for robust inside-outside segmentation may be found in Jacobson et al., 2013 [Jacobson et al., 2013].

¹https://en.wikipedia.org/wiki/Winding_number, accessed: 09.09.22

3.3. POINT IN 3D-MESH FUNCTIONALITY

The striking benefit of the generalized winding number approach is its robustness. However it is also quite slow. For a point in space and a triangular mesh defined by n triangular faces, the time complexity of the generalized winding number approach is $O(n)$ [Devert, 2018], with n being the number of vertices in the mesh. When dealing with anatomical artefact meshes, partially containing tens of thousands of triangular faces, and whilst having to do an abundance of point in 3D mesh checks, this turned out to be too slow for the purposes of this thesis.

An earlier attempted implementation of the generalized winding number approach resulted in an estimated average calculation time per point-in-artefact check of roughly two seconds, when parallelised on four CPU cores. This average is not exact by any means, but rather a low estimate based on the observations made. When having to perform 3.4 million (see Section 3.2.1) point-in-artefact checks, the expected creation time of the atlas was roughly 78 days. This was way too slow to be feasible in the context of this thesis, even if further sped up by multiprocessing on more CPU cores.

Therefore, I have decided to go with the first option: ray casting. Ray casting for inside-outside segmentation, when performed on the appropriate data structures, can be sped up to a time complexity of $O(\log n)$, for n triangular faces [Horvat, 2012]. Whilst possible to reimplement a ray casting inside-outside segmentation approach from scratch, it seemed advisable to proceed with one of the multiple already existing toolkits implementing this. This thesis uses the 'vtkSelectEnclosedPoints' module, provided by vtk, an open source visualization toolkit developed by Kitware. The module implements the desired inside-outside segmentation via ray casting, and has shown to be sufficiently fast for my needs.

Usage of `vtkSelectEnclosedPoints`

The module performs ray casting based inside-outside segmentation on 'vtkPolyData' objects. These vtk specific objects represent geometric structures consisting of vertices, lines, and polygons with their specific attributes². This is implemented, by creating said 'vtkPolyData' objects from the artefact STL files, which were in turn generated by transforming the initially provided artefact TRI files.

²<https://vtk.org/doc/nightly/html/classvtkPolyData.html>, accessed: 01.09.2022

4. Results

The following chapter will evaluate the quality of the previously created anatomical artefact atlas in terms of its precision in labelling coordinates in MNI space with the corresponding HArtMuT artefacts. A plot of the final atlas is shown in Fig. 4.1.

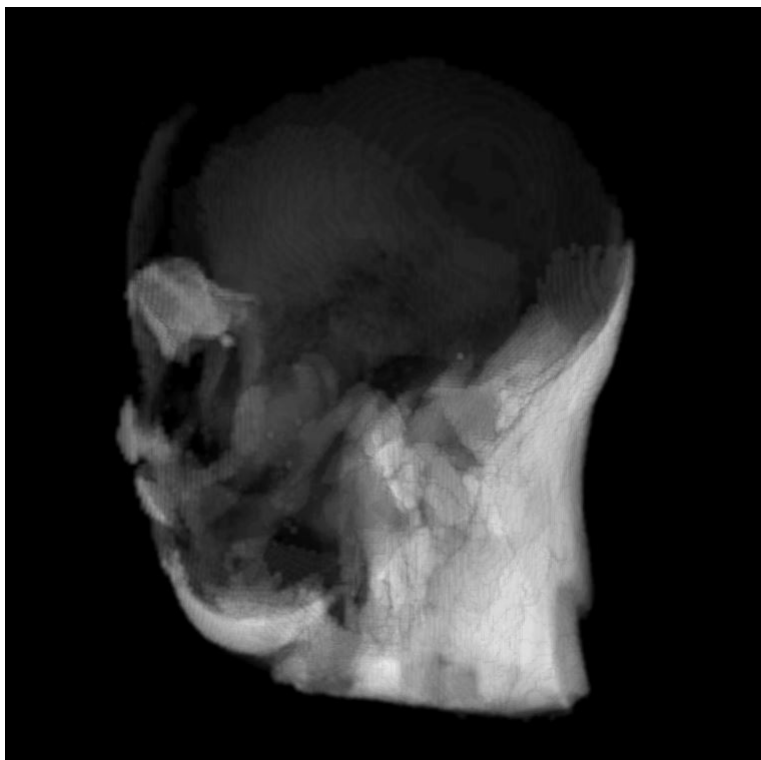


Figure 4.1.: Presentation of the created atlas' final NIFTI file, displayed via 'Brain Viewer'¹ with the atlas' artefacts visible in grayscale according to their artefact integer label.

Note: For better visibility in the context of this depiction, the integer artefact label of the muscle General has been altered from one to sixty-four.

4.1. Reference Data

To test the atlas, it will be queried for a data set of 1446 coordinates located within MNI space, for which I have been provided with the corresponding HArtMuT artefact labelling by the authors of

¹<https://socr.umich.edu/HTML5/BrainViewer/>, accessed: 05.10.2022

4.2. INTEGRATION INTO EXISTING PIPELINES - 'ATLASREADER'

Harmening et al 2022. It contains the localized artefact sources of 19 subjects in an mobile EEG experiment. This data was labelled using the 'is_inside_naive'-functionality of the 'marmakoide/inside-3D-mesh'² repository, implementing inside-outside segmentation on 3-dimensional meshes using the aforementioned generalized winding number approach [Jacobson et al., 2013]. The reference data labelling was also modified by hand by Nils Harmening, to adjust for some occurring errors, and to increase labelling accuracy. These labelling results are compared with those provided by the created atlas.

It is important to note, that the reference data at hand does not employ the exact same artefacts when labelling coordinates. Whilst 57 of the artefacts are the same, the atlas' labelling is extended by more detailed artefacts within the human eye, to allow for more precise labelling. This obviously introduces a source of mismatching labels, which will need to be taken into account when evaluating the quality of the atlases labelling. The artefacts included within the atlas, but not the reference data are: left and right Eye Aqueous, left and right Eye Lens, and left and right Eye Vitreous.

As the data provided to me included some data points that were either duplicates, or outside of the atlas' coordinate space, they were removed from the initial data set before testing. This process resulted in the aforementioned 1446 data points to be taken into consideration.

The cleaned reference data used for this comparison is provided with this thesis' repository as 'referenceDataOriginal.txt'.

4.1.1. Labelling Comparison

Whilst the atlas' data has been labelled with the integer labels presented in Appendix A, the reference data has been labelled not via said integer labels, but rather the artefact names directly. Therefore, when talking about a matching labelling, this shall denote a match between the artefact provided by the integer-artefact-mapping via the atlas' integer label, and the artefact label of the reference data.

4.2. Integration into Existing Pipelines - 'atlasreader'

To show usability within standard tools existing for working on parcellation atlases, and to make use of such tools for the task at hand of comparing the atlas' labelling results with the provided reference data, the python package 'atlasreader' [Notter et al., 2019] is used for the retrieval of the atlas' labelling. This package has been introduced as a means for generating informative figures and region labels from statistical MRI images, and offers wide spread functionality in working with parcellation atlases.

²<https://github.com/marmakoide/inside-3D-mesh>, accessed: 22.09.22

4.3. Outcome

4.3.1. General Performance - Match and Mismatch Rate

Tab. 4.1 shows the general performance of the atlas' labelling by displaying the number of matches and mismatches between the atlas', and the reference data's labelling.

| Total Data Points | Matches | Match Rate | Mismatches | Mismatch Rate |
|-------------------|---------|------------|------------|---------------|
| 1446 | 1184 | 81.9% | 262 | 18.1% |

Table 4.1.: Comparison of the atlas' labelling with the reference data labelling, depicting the number of total points considered, both the total amount of matches and mismatches, and match and mismatch rate respectively

It can be seen that, out of the total of 1446 data points, the atlas' and reference provided labels are matching in a total of 1184 cases. This corresponds to a matching rate of 81.9%. There has been a mismatch in labelling in a total of 262 cases, a mismatch rate of 18.1%.

4.3.2. Analysis and Correction of Reference Data

In the process of evaluating the created atlas' labelling, during the investigation of mismatches, the reference data was found to be faulty in some cases. Therefore, and to later on allow for better interpretation and a higher validity of the atlas' results, I have decided to review the reference data to check for potential incorrect labelling, and to correct these incorrect data points. This is done by individually plotting coordinates and the relevant artefact meshes, and then verifying the correctness of the provided data by hand. The reference data is corrected wherever it is found to be incorrect. This correction includes labelling coordinates with the 6 labels mentioned before in Section 4.1, previously not used in the reference data's labelling.

I have verified the correctness for all data points included in the original reference data to the best of my knowledge. This lead to the following results depicted in Tab. 4.2.

| Total Data Points | Corr. Reference Data | Corr. Rate | Incorr. Reference Data | Incorr. Rate |
|-------------------|----------------------|------------|------------------------|--------------|
| 1446 | 1200 | 83% | 246 | 17% |

Table 4.2.: Analysis of the provided reference data, depicting the total amount of data points to be considered, and the total amount of data points found to be labelled correct and incorrect, with their respective rates, found by a manual review.

The reference data was found to be correct for 1200 of the 1446 data points. This corresponds to a rate of 83%. However, it was also found to be incorrect for a total of 246 data points, corresponding to a rate of incorrect data points of 17%.

Most coordinates found labelled incorrectly in the initially provided reference data were found to be either just outside of an artefact mesh and labelled as inside, or just inside, and labelled as outside. The majority of falsely labelled coordinates in the reference data stems from the muscle General, the

4.3. OUTCOME

left and right muscle Temporalis Temporoparietalis and the muscle Splenius Capitis, the 4 largest artefacts used in the creation of the atlas, volume wise.

For those data points found to be incorrect, I have corrected them to the best of my knowledge.

The corrected reference data is provided within this thesis' repository as 'referenceDataCorrected.txt'.

4.3.3. General Performance - Match and Mismatch Rate after Reference Data Correction

As the experiment testing the atlas' labelling depicted in Tab. 4.1 can not be deemed conclusive due to the not insignificant amount of wrongfully labelled data points found in the original reference data (see Tab. 4.2), a second test of the atlas' labelling has been conducted, with the now corrected reference data.

| Total Data Point | Matches | Match Rate | Mismatches | Mismatch Rate |
|------------------|---------|------------|------------|---------------|
| 1446 | 1313 | 90.8% | 133 | 9.2% |

Table 4.3.: Comparison of the atlas' labelling with the now corrected reference data labelling, depicting the number of total points considered, both the total amount of matches and mismatches, and match and mismatch rate respectively

Out of the 1446 data points provided by the corrected reference data, the data's and the atlas' labelling match in a total of 1313 cases. This corresponds to a matching rate of 90.8%. There has been a labelling mismatch in a total of 133 cases. This corresponds to a mismatch rate of 9.2%.

4.3.3.1. Mismatches

Whilst Tab. 4.3 shows matching labelling in 90.8%, the atlas has nonetheless also produced a total of 133 mismatches in comparison to the corrected reference data, corresponding to a mismatch rate of 9.2%.

When looking into the faulty labels introduced by the atlas, it was found that all but one of the occurring mistakes where coordinates labelled with an artefact label, despite the coordinate point being just slightly outside (distance of one or less voxel diameter in world space) the artefact mesh.

Out of the total of 133 mismatches, only one was found were the atlas labelled the point in question with an artefact label, despite the coordinate being clearly outside any of the HArtMuT artefact meshes with some significant distance (distance of two or more voxel diameters in world space) to the artefact it was labelled according to. This one faulty coordinate was labelled with the muscle General label in the atlas.

5. Discussion

5.1. Interpretation of the Results

5.1.1. General Labelling

The previous finding indicate that the created atlas is generally well suited for the task of assigning HArtMuT artefact labels for provided coordinates. However, whilst the created atlas, tested against a data set of previously labelled coordinates in the previous section performed generally well on the aforementioned task (90.8% matching rate, see Tab. 4.3), it still introduced false labels for 133 out of the total of 1446 data points (9.2% mismatching rate, see Tab. 4.3). Whilst this was partially to be expected, due to the imperfect, heuristic approach chosen for the implementation in this thesis (see. Section 3.2.2), it is still important, to judge these faulty labels by whether they have been introduced by the chosen implementation, or another source of error negatively impacting the atlas' performance, which would be significantly limiting its usefulness.

5.1.2. 'Deliberate' Mismatches

As described in Section 4.3.3.1, all but one data point labelled incorrectly by the created atlas were found to be coordinates located closely outside of the artefact meshes, with the atlas still labelling them to be within a specific artefact mesh. These faulty labelled points were at most one voxel diameter in voxel space away from the closest artefact mesh, which contributes strongly to the finding of these mistakes to be originating from the chosen labelling approach. The approach depicted in Section 3.2.2 introduced such a source of error, by labelling voxels according to an artefact, if just one of their corner points is found to be contained in the mesh. Section 5.2.3.1 look into the source and impact of these faulty labels in more detail. Preempting this, it can be said, that these mistakes in general do not pose a huge loss in quality for the atlas' labelling, as the task of source localization in general suffers from multiple outside sources of uncertainty which will be looked at closer in Section 5.2. This general uncertainty renders minor mistakes such as the aforementioned ones negligible. As these mistakes do also generally increase the displayment of the HArtMuT artefacts in the atlas (as more voxels receive a non-empty space label), which increase the likelihood of determining an artefact for a specific input source, these inaccuracies are deemed to be insignificant and tolerable.

5.1.3. 'Unwanted' Mismatches

Out of the 133 data points found to be faultily labelled, only one could not be attributed to the chosen implementation. Whilst the exact source of this mistake can not be determined with full certainty in the scope of this thesis, the experience gained throughout the creation of this atlas indicates a calculation error occurring during the inside-outside segmentation using the 'vtkSelectEnclosedPoints' module as the most likely explanation. This assumption is supported by the fact that the faulty label

5.2. LIMITATIONS

was determined to be the muscle General, by far the largest artefact mesh used in the creation of this atlas, which significantly increases the calculation requirements required for inside-outside segmentation, and which might have introduced some sort of inaccuracy which lead to the false label. A more in depth look into uncertainty introduced by the usage of the 'vtkSelectEnclosedPoints' module is found in Section 5.2.3.2. It can be said however, that even though the testing of the atlas has been done on a relatively small data set of size 1446, the occurrence rate of such mistakes, not explainable by the chosen approach to labelling, is only roughly 0.07% (1 occurrence in 1446 data points). This is deemed low enough to be considered negligible, when compared to the general uncertainty in the sources provided by source localization, and the tool of parcellation atlases in general. Both of which will be looked at more closely in the following section.

5.1.4. Reference Data

Finally, my findings regarding the initially falsely labelled reference data indicates, that the chosen method of segmentation [Devert, 2018] for said reference data performed poorly on the artefact meshes employed during the creation of this atlas. As mentioned in Section 4.3.2, most data points falsely labelled were found to stem from the four largest artefacts used in the creation of this atlas. This might indicate, that the chosen method suffered a loss of accuracy for more complex artefact meshes requiring more, and more complex, calculations to be performed. However, this can not be said with full certainty, as for a randomly distributed set of reference data coordinates, it is only naturally to find most mistakes to be made for the largest artefact meshes making up most of the non-empty space.

Finally, it can not be said with full certainty, whether the issues shown in the original reference data originated from a problem with the provided implementation in the 'marmakoide/inside-3D-mesh' repository, or if an error occurred in the application of the implemented concept on the HArtMuT artefacts.

5.2. Limitations

Even though I have succeeded in creating an anatomical artefact atlas based on HArtMuT, which provides relatively high quality source labelling as depicted in Tab. 4.3, this thesis and the atlas' capability is still limited by various different factors, which will be looked into in the following subsection.

5.2.1. Sources of Uncertainty

Before looking into the limitations bounding the created parcellation atlas and its implementation in specific, it is important to look into the general limitations existing within source localization and the provided artefacts. The following list of sources of uncertainty is not exhaustive, but helps to put the atlas' limitations into perspective.

5.2.1.1. Provided Artefacts

The anatomical artefacts provided to create this atlas have been created from the manual segmentation of a high resolution Magnetic Resonance Imaging (MRI) scan of a single person by a group of

5.2. LIMITATIONS

experts [Iacono et al., 2015]. Such a segmentation introduces a variety of uncertainties. Even when performed by professionals, there is no guarantee for a perfect segmentation, and the limitation to a single person introduces uncertainty by focusing on an specific individual muscle anatomy.

5.2.1.2. Processed Data

Source Localization generally deals with preprocessed EEG data. Such preprocessing adds another layer of uncertainty within the provided sources, as it will not provide perfect components of electrical activity to localize. Additionally, when trying to solve the Inverse EEG Problem as presented in Section 2.4, there are a multitude of assumptions and simplifications to be made¹, which further increase general uncertainty.

5.2.2. Parcellation Atlases in General

Whilst the usage of parcellation atlases is widespread for the task of source retrieval, the approach and its usability in general, and therefore of the implementation presented in this thesis as well, is limited by some existing boundaries.

As with all data structures, the created atlas can only offer resolution up to a specific precision. Whilst theoretically possible to increase this precision by reducing voxel size, this also significantly increases the computation time and power required to create the atlas. As mentioned before in Section 5.2.1, sources provided by source localization deal with significant uncertainty from other factors beyond the parcellation atlas. Therefore a higher resolution atlas is often not deemed necessary. For simplicity, in this thesis, I chose to use the NIFTI file provided by the 'extended Tissue Probability Map' (eTPM), developed by Christopher Rorden [Huang et al., 2013]. This map fulfilled the requirements of including the neck structure of the human head, and offered a sufficiently detailed resolution. Its resolution was deemed to be sufficient in comparison to other brain parcellation atlases, which were found to be in the general vicinity resolution wise.

Another aspect limiting the precision of parcellation atlases in general is the fact that they are usually defined for some standard anatomical space. For the created atlas, this space is the MNI space. Such anatomical spaces are usually created as the average of a multitude of scans, and are therefore not perfectly applicable to an individual person.

Such a perfect applicability can never be achieved with a non-individual atlas, as the human head anatomy differs ever so slightly from person to person.

5.2.3. Implementation

As previously described, some of the aforementioned limitations come naturally with the means of an anatomical parcellation atlas. However, the chosen specific implementation introduced by this thesis also contributed to these limitations, and therefore, the following section will address the two key aspects of the atlas' implementation that restrict and limit its quality of labelling.

¹<https://sapienlabs.org/lab-talk/the-inverse-problem-in-eeeg/>, accessed: 04.09.2022

5.2. LIMITATIONS

5.2.3.1. Labelling Approach

Algorithm 1 and Algorithm 4 introduced the basic concepts of the atlas' labelling. Initially, voxels were labelled if their center point were found to be within a specific artefact mesh. Later on, this labelling approach was improved, by checking, for voxels previously labelled as 'empty Space' which were found to be neighbouring voxels labelled with an artefact label, if one of their corner points was found to be within an artefact mesh. The combination of these approaches provides relatively good voxel coverage of the artefact meshes within the created atlas (see Fig. 3.11), but introduces some potentially faulty labelling.

As voxels have a volume, determining that their center or corner points are within an artefact mesh is not sufficient to guarantee that all points within the voxel are within the artefact mesh. Therefore, and due to the limited resolution of the created atlas, a perfect labelling is not guaranteed for all inputs. However, due to the previously established fact that source localization suffers from multiple outside sources of uncertainty, this approach was deemed to be highly suitably for the task at hand.

An analysis of the mismatches between the atlas' and the reference data's labelling shows that most of the atlas' false labels are found for coordinate points very closely, but barely outside of an artefact mesh. This is found to be consistent, as the chosen approach introduces a source of error for input coordinates very closely outside of an artefact mesh. The labelling might indicate them to be within an individual artefact, even if they are actually just very close to it (more precisely, at most, one voxel diameter away from the mesh). Whilst theoretically incorrect, such a labelling is not obstructive, but rather helpful in the context of source retrieval, due to the fact that it provides additional information. As sources for source localization are per default not exact, this approach rather introduces an error tolerance, allowing for the retrieval of the anatomical source for a provided location, even if said location is not perfectly exact.

If a more precise labelling than the one created by this thesis is required, it is necessary to implement a more complicated, computation-heavy approach to the atlas' labelling.

5.2.3.2. Inside-Outside Segmentation

Section 3.3 introduced the means for determining if a coordinate in space is located within one of the artefact meshes. The functionality offered by *vtk* has been found to be of high quality and precision. Nevertheless, few individual cases have been found, in which the used '*vtkSelectEnclosedPoints*' functionality provides unexpected results, that is, for coordinates clearly outside of an artefact mesh, determining them to be within the mesh. Such errors mainly occurred for the artefact mesh of the muscle General, which is by far the largest of the HArtMuT provided meshes. It can only be assumed that its size contributed to individual errors in the functionalities calculations. However, a viewing of the atlas in a software specified in the display of NIFTI files² showed these issues to be extremely rare. Whilst the exact quantity of these occurrences can not be specified, based on a visual inspection I assume there to be less than one hundred in total. Compared to a grand total of roughly 3.5 million voxels, having in mind that sources provided by source localization are not exact anyway, and in regards of the performed experiment showing sufficient labelling accuracy of the created atlas, these errors can be deemed negligible.

²<https://socr.umich.edu/HTML5/BrainViewer/>, accessed: 05.10.2022

5.2. LIMITATIONS

5.2.4. Reference Data Issues

As depicted in Tab. 4.2, the reference data at hand proved to be at least partially faulty. This was verified by individually checking the provided data points by plotting the respective coordinates and artefact meshes. The data points deemed incorrect were corrected. However, as this was done by hand, and the provided artefact meshes are highly complex, I can not guarantee absolute correctness for all data points. Therefore, for future verification of the created atlas, there should be a new set of reference data put to use, created by another means of segmentation.

6. Conclusion

This thesis proposed the creation of an anatomical artefact parcellation atlas, based on the non-cortical head artefacts provided by HArtMuT. This atlas is to be used on the results of source localization, by performing source retrieval for anatomical HArtMuT artefacts, and therefore, to allow for easy determination of the specific artefact at a sources' position in space.

I implemented such an artefact atlas, building up the artefact segmentation on the vtk provided 'vtkSelectEnclosedPoints' module, which performs inside-outside segmentation for coordinates and 3-dimensional meshes using a ray casting approach, which managed to sufficiently reduce the atlas' creation time to be feasible in the context of this thesis.

The atlas was found to perform sufficiently good in the task at hand. In comparison to the corrected reference data the atlas provides the correct result in 90.8% of cases. Most incorrect labels retrieved by the atlas were found to be negligible, when compared to the uncertainty introduced by source localization in general. Most faulty labels were deemed to be a result of the design choice for the atlas' implementation, which introduced a small error rate when labelling coordinates very closely to an artefact mesh's surface.

Alongside this thesis' main aspect of creating and testing the HArtMuT based artefact atlas, I have found the initially provided reference data to be faulty, corrected it to the best of my knowledge, and provide the corrected data with this thesis.

As a parcellation atlas based on non-cortical HArtMuT artefacts has not been previously introduced, I managed to implement this atlas as the first easy solution to source position labelling for said HArtMuT artefacts, significantly reducing the time and computation power required to do source retrieval when compared to a manual approach.

Additionally, the compatibility with the standard parcellation atlas toolbox 'atlasreader' has been shown, by using it in the process of testing the created atlas.

Nonetheless, due to the issues depicted with the initially provided reference data, a future verification of this atlas with an alternative data set created by a different means of segmentation is recommended.

Bibliography

- [Bourke, 2001] Bourke, P. (2001). Tri file format.
- [Brett et al., 2002] Brett, M., Johnsrude, I., and Owen, A. (2002). The problem of functional localization in the human brain. *Nature reviews. Neuroscience*, 3:243–9.
- [Brett et al., 2022] Brett, M., Markiewicz, C. J., Hanke, M., Côté, M.-A., Cipollini, B., McCarthy, P., Jarecka, D., Cheng, C. P., Halchenko, Y. O., Cottaar, M., and et al. (2022). nipy/nibabel: 3.2.2.
- [Cox, 2004] Cox, R. W. (2004). The nifti-1 data format. *Data Format Working Group (DFWG)*, page 13.
- [Devert, 2018] Devert, A. (2018). inside-3d-mesh. <https://github.com/marmakoide/inside-3d-mesh>. Accessed: 01.09.2022.
- [Fan et al., 2016] Fan, L., Li, H., Zhuo, J., Zhang, Y., Wang, J., Chen, L., Yang, Z., Chu, C., Xie, S., Laird, A. R., Fox, P. T., Eickhoff, S. B., Yu, C., and Jiang, T. (2016). The Human Brainnetome Atlas: A New Brain Atlas Based on Connectional Architecture. *Cerebral Cortex*, 26(8):3508–3526.
- [Grabner et al., 2006] Grabner, G., Janke, A. L., Budge, M. M., Smith, D., Pruessner, J., and Collins, D. L. (2006). Symmetric atlasing and model based segmentation: An application to the hippocampus in older adults. In Larsen, R., Nielsen, M., and Sporring, J., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, pages 58–66, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Grünbaum and Shephard, 1990] Grünbaum, B. and Shephard, G. C. (1990). Rotation and winding numbers for planar polygons and curves. *Transactions of the American Mathematical Society*, 322:169.
- [Harmening et al., 2022] Harmening, N., Klug, M., Gramann, K., and Miklody, D. (2022). Hartmut - modeling eye and muscle contributors in neuroelectric imaging. *bioRxiv*.
- [Hassani and Karami, 2015] Hassani, M. and Karami, M. R. (2015). Noise estimation in electroencephalogram signal by using volterra series coefficients. *Journal of Medical Signals and Sensors*, 5:192 – 200.
- [Hormann and Agathos, 2001] Hormann, K. and Agathos, A. (2001). The point in polygon problem for arbitrary polygons. *Computational Geometry*, 20(3):131–133.
- [Horvat, 2012] Horvat, D. (2012). Ray-casting point-in-polyhedron test.

Bibliography

- [Huang et al., 2013] Huang, Y., Dmochowski, J., Su, Y., Datta, A., Rorden, C., and Parra, L. (2013). Automated mri segmentation for individualized modeling of current flow in the human head. *Journal of neural engineering*, 10:066004.
- [Iacono et al., 2015] Iacono, M., Neufeld, E., Akinnagbe, E., Bower, K., Wolf, J., Vogiatzis Oikonomidis, I., Sharma, D., Lloyd, B., Wilm, B., Wyss, M., Pruessmann, K., Jakab, A., Makris, N., Cohen, E., Kuster, N., Kainz, W., and Angelone, L. (2015). Mida: A multimodal imaging-based detailed anatomical model of the human head and neck. *PLoS ONE*, 10:e0124126.
- [Jacobson et al., 2013] Jacobson, A., Kavan, L., and Sorkine-Hornung, O. (2013). Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (TOG)*, 32:1–4.
- [Jiang et al., 2019] Jiang, X., Bian, G.-B., and Tian, Z. (2019). Removal of artifacts from eeg signals: A review. *Sensors*, 19:987.
- [Joyce et al., 2004] Joyce, C., Gorodnitsky, I., and Kutas, M. (2004). Automatic removal of eye movement and blink artifacts from eeg data using blind component separation. *Psychophysiology*, 41:313–25.
- [Kanoga and Mitsukura, 2017] Kanoga, S. and Mitsukura, Y. (2017). *Review of Artifact Rejection Methods for Electroencephalographic Systems*, pages 71–72.
- [Knipe and Moore, 2019] Knipe, H. and Moore, C. (2019). Nifti (file format). *Radiopaedia.org*.
- [König et al., 2016] König, P., Wilming, N., Kietzmann, T. C., Ossandón, J. P., Onat, S., Ehinger, B. V., Gameiro, R. R., and Kaspar, K. (2016). Eye movements as a window to cognitive processes. *Journal of Eye Movement Research*, 9(5).
- [Lucka, 2009] Lucka, F. (2009). Inverse Probleme in Medizinischer Bildgebung und Bioelektromagnetismus. 2009, Universität Münster, Seminar zu Differenzialgleichungen in der Biomedizin, URL: https://www.uni-muenster.de/AMM/num/Vorlesungen/Seminar_DglBioMed_SS09/InverseProblems.pdf
Accessed: 13.09.2022.
- [Makeig et al., 1995] Makeig, S., Bell, A., Jung, T.-P., and Sejnowski, T. J. (1995). Independent component analysis of electroencephalographic data. In Touretzky, D., Mozer, M., and Hasselmo, M., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 145–151. MIT Press.
- [Michel and He, 2019] Michel, C. M. and He, B. (2019). Chapter 6 - eeg source localization. In Levin, K. H. and Chauvel, P., editors, *Clinical Neurophysiology: Basis and Technical Aspects*, volume 160 of *Handbook of Clinical Neurology*, pages 85–101. Elsevier.
- [Notter et al., 2019] Notter, M. P., Gale, D., Herholz, P., Markello, R., Notter-Bielser, M.-L., and Whitaker, K. (2019). Atlasreader: A python package to generate coordinate tables, region labels, and informative figures from statistical mri images. *Journal of Open Source Software*, 4(34):1257.
- [Ramachandran and Varoquaux, 2011] Ramachandran, P. and Varoquaux, G. (2011). Mayavi: 3d visualization of scientific data. *Computing in Science & Engineering*, 13:40 – 51.

Bibliography

- [Symeonidou et al., 2018] Symeonidou, E.-R., Nordin, A., Hairston, W. D., and Ferris, D. (2018). Effects of cable sway, electrode surface area, and electrode mass on electroencephalography signal quality during motion. *Sensors (Basel, Switzerland)*, 18.
- [Winkler, 2012] Winkler, A. M. (2012). The nifti file format. <https://brainder.org/2012/09/23/the-nifti-file-format/>. Accessed: 01.09.2022.

A. Artefact-Integer-Mapping

```
name, index
Empty_Space, 0
Muscle_General, 1
Muscle_LateralRectus_right, 2
EyeLens_right, 3
Muscle_MedialPterygoid_left, 4
Muscle_Masseter_left, 5
EyeVitreous_left, 6
Muscle_LateralPterygoid_right, 7
Muscle_InferiorRectus_right, 8
Muscle_ZygomaticusMinor_right, 9
Muscle_Masseter_right, 10
Muscle_InferiorOblique_left, 11
Muscle_ZygomaticusMinor_left, 12
Muscles_Risorius_left, 13
Muscle_DepressorAnguliOris_left, 14
Muscle_Occipitiofrontalis_FrontalBelly, 15
Muscle_DepressorLabii_right, 16
Muscle_MedialRectus_left, 17
Muscle_Platysma_right, 18
Muscle_Mentalis_right, 19
Muscle_Sternocleidomastoid_left, 20
Muscle_LevatorLabiiSuperioris_right, 21
EyeCornea_right, 22
Muscle_Temporalis_Temporoparietalis_left, 23
Muscle_MedialPterygoid_right, 24
Muscle_LevatorScapulae_left, 25
EyeCornea_left, 26
Muscle_LateralPterygoid_left, 27
Muscle_Buccinator_left, 28
Muscle_SuperiorRectus_left, 29
Muscle_Temporalis_Temporoparietalis_right, 30
Muscle_SuperiorOblique_left, 31
EyeVitreous_right, 32
EyeAqueous_left, 33
EyeAqueous_right, 34
Muscles_Risorius_right, 35
Muscle_ZygomaticusMajor_left, 36
Muscle_SpleniusCapitis_left, 37
Muscle_Buccinator_right, 38
Muscle_OrbicularisOculi_right, 39
Muscle_LevatorLabiiSuperioris_left, 40
EyeLens_left, 41
Muscle_LateralRectus_left, 42
Muscle_Nasalis, 43
Muscle_SpleniusCapitis_right, 44
Muscle_Mentalis_left, 45
Muscle_InferiorRectus_left, 46
Muscle_Trapezius, 47
Muscle_Occipitiofrontalis_OccipitalBelly, 48
Muscle_DepressorLabii_left, 49
EyeRetina_Choroid_Sclera_left, 50
Muscle_SuperiorRectus_right, 51
Muscle_Sternocleidomastoid_right, 52
Muscle_OrbicularisOculi_left, 53
Muscle_Platysma_left, 54
Muscle_InferiorOblique_right, 55
Muscle_MedialRectus_right, 56
EyeRetina_Choroid_Sclera_right, 57
Muscle_LevatorScapulae_right, 58
Muscle_DepressorAnguliOris_right, 59
Muscle_SuperiorOblique_right, 60
Muscle_OrbicularisOris, 61
Muscles_Procerus, 62
Muscle_ZygomaticusMajor_right, 63
```

Figure A.1.: Artefact-Integer-Mapping introduced by the labels CSV file provided with this atlas