# Predicting facial hard tissue shape from soft tissue shape with a dense autoencoder and principal component analysis

## A Proof of Concept

Master thesis

Technical Medicine - Medical Imaging and Interventions

by

Ilja Westra

4th of April, 2023

GRADUATION COMMITTEE

| | |
|---|---|
| Chairman | Prof. Dr. I. Sechopoulos |
| Medical supervisor | Drs. B.A. Meijer |
| Technical supervisors | |
| Radboudumc | Dr. G. de Jong |
| University of Twente | Prof. Dr. I. Sechopoulos |
| Process supervisor | Drs. P.A. van Katwijk |
| External member | J.K. van Zandwijk, MSc |
| Extra member | Prof. Dr. T.J.J. Maal |

UNIVERSITY OF TWENTE.

Amsterdam UMC Universitair Medische Centra

3DLab Radboudumc

# Abstract

**Introduction:** Facial feminization surgery (FFS) is an underdeveloped surgical field in need of objective methods to plan, execute, predict and evaluate FFS and its outcomes. If the desired postoperative facial shape is known, a method to translate that soft tissue shape into matching hard tissue shape will make objective surgical planning possible. A proof of concept for translating facial soft tissue shape towards matching hard tissue shape is provided in this research.

**Methods:** Principal component analysis was performed on 252 facial soft and hard tissue meshes with 1:1 vertex correspondence. The soft and hard tissue principal component scores were used as input and output, respectively, for a dense autoencoder. The training, validation and test sets consisted of 200, 31, and 21 subjects respectively. The predicted hard tissue principal component scores were reconstructed to meshes and compared to the true meshes.

**Results:** The reconstructed meshes had a median error in the x, y and z coordinates below 1.5 mm with median absolute deviation below 1.2 mm. The regions with errors above the third quartile corresponded with inaccurately predicted principal components.

**Conclusion:** This study provides a proof of concept for the predictive value of facial soft tissue shape for facial hard tissue shape through a combination of PCA and a dense autoencoder applied to 3D meshes. Along with more data of subjects with comparable demographics and a stricter exclusion on image quality, the results should improve and the true possibilities for objective FFS methods should become clear.

# Contents

**Appendices**

# List of acronyms

| | |
|---|---|
| **CBCT** | Cone Beam Computed Tomography |
| **FFS** | facial feminization surgery |
| **HT** | hard tissue |
| **MtF** | male to female |
| **OMFS** | oral and maxillofacial surgery |
| **PCA** | principal component analysis |
| **PCs** | principal components |
| **ST** | soft tissue |

# Chapter 1

# Introduction

In recent decades, the number of people seeking professional help for gender identity related issues has increased tremendously [1]–[3]. A perceived incongruity between a person's gender identity and their gender assigned at birth is defined as gender dysphoria, which often causes emotional and mental distress [4], [5]. Treatment of gender dysphoria is individualized and may or may not involve hormonal and surgical interventions. Gender confirming surgical interventions include bottom surgeries (genital reconstruction or removal of sex organs), top surgeries (breast removal or augmentation) and facial surgery. In male to female (MtF) individuals, otherwise known as transgender women, facial gender confirming surgery has been found to generally be the surgery of highest priority for transitioning [6], where it is also called facial feminization surgery (FFS).

In FFS, masculine facial features are feminized to reduce dysphoria and instances of misgendering. In the Netherlands, most oral and maxillofacial surgery (OMFS) training programs have little to no exposure to FFS. The relative novelty along with a lack of evidence-based protocols and guidelines are the main reasons why FFS is an underdeveloped surgical discipline that is difficult to execute systematically. The knowledge to perform FFS is acquired mainly through experience, and at the time of writing there is no gold standard of pre-operative planning. This makes it difficult to train new surgeons as the precise adjustments depend a subjective analysis of an individual patients' facial features. [5], [7], [8]

Furthermore, patients do not know what they will look like after surgery as there is no accurate way of visualizing the post-operative changes in facial features of individual patients [9], [10]. This can lead to unrealistic expectations and patient dissatisfaction. Meanwhile, evaluation of FFS results are currently based on such subjective measures as patient satisfaction or the surgeon's perception of femininity. In short, there is a need for objective methods to plan, execute, predict and evaluate FFS and its outcomes to ensure clinical reproducibility of procedures, predictable outcomes with realistic patient expectations and improved training of surgeons.

## 1.1 Framework

This research has been conducted at the OMFS department of Amsterdam UMC and the 3D lab of Radboudumc in the Netherlands and proposes a method for prediction of facial hard tissue shape based on facial soft tissue shape. Knowledge of the predictive properties of these tissue shapes can be deployed towards surgical planning of FFS and other OMFS procedures to improve patient care.

## 1.2 Research aim

Due to the aesthetic component of FFS, surgical planning is based around the postoperative facial shape. To accurately and objectively plan FFS, an approximation of surgical hard tissue adjustments that result in the desired facial shape is needed. The necessary hard tissue adjustments can be obtained by analyzing the difference between the preoperative and postoperative hard tissue shape. Therefore, if the desired postoperative facial shape is known, a method to translate that soft tissue shape into matching hard tissue shape will make objective surgical planning possible. Such a method for translating facial soft tissue shape towards matching hard tissue shape is explored in this research.

This thesis aimed to provide a proof of concept for the method of using principal component analysis in combination with a dense autoencoder to predict facial hard tissue shape based on soft tissue shape in adult individuals of the general population.

## 1.3 Outline of thesis

This thesis describes the process of the proposed method for facial hard tissue prediction and provides early results obtained with this method. In Chapter 2, a clinical and technical background will be given. The clinical background provides insight into FFS and the current clinical workflow and its limitations. In the technical background the principles of the Meshmonk algorithm, autoencoder models and principal component analysis (PCA) are explained. Chapter 3 states the methods executed in this research, divided in data preparation and processing. Chapter 4 contains the obtained results which are later discussed in Chapter 5. Finally, in Chapter 6, conclusions and recommendations are given.

Chapter 2

# Background

This chapter contains background information which will provide clinical context and explanation of principles behind technical aspects of this thesis. This chapter is divided in a clinical and technical section.

## 2.1 Clinical background

In this section, a general introduction to FFS is given and literature on the relation between soft and hard tissue is reviewed.

### 2.1.1 Facial feminization surgery

Successful FFS provides patients with the ability to align their facial appearance and gender expression with their gender identity, thus decreasing their gender dysphoria and related mental health issues. [11] A number of different procedures and approaches are possible in FFS, which should always be tailored to the individual patient. These procedures include hard tissue alterations such as frontal bone cranioplasty, zygoma osteotomy, mandibular reduction, and genioplasty. Soft tissue alterations are also an important part of FFS and include rhinoplasty, brow lifts, hairline lateration, lip augmentation, Adam's apple contouring and face and neck lifts. [8] [12] Hard and soft tissue altering procedures are often combined into the treatment of the patient.

Patient satisfaction is one of the most important factors of treatment, which denotes whether patients are content with the received health care. This measure is affected by patient expectations, independent of quality of care, with realistic expectations resulting in a higher patient satisfaction. [13]–[15] Unfortunately, realistic expectations are difficult to manage in facial surgeries such as FFS due to the aesthetic aspects and inability to simulate the surgical outcome pre-operatively. [9], [10] A surgical outcome simulation would help manage expectations and is expected to result in more realistic expectations and a higher patient satisfaction.

Value in surgical care is maximized when the desired outcome is highly likely to be realised, and the incidence of adverse outcomes is low. [16] Through implementation of surgical planning, the accuracy and reproducibility of FFS will increase which is expected to result in an increase in value in surgical care.

### 2.1.2 Relation between soft and hard tissue

The shape of the face and its features is determined by hard tissue, which is comprised of the cranial and facial bones, and soft tissue such as skin, muscle, and fat. The relationship between facial hard and soft tissue has been the subject of numerous studies in the field of OMFS, with the aim to be able to predict postoperative facial appearance changes. When looking at the predictive value between facial hard and soft tissue, most research has been conducted within orthognathic surgery and has been focused on soft tissue prediction [17]. Not unlike FFS, the objective of orthognathic surgery is twofold: a functional and aesthetic result. Research is focused on postoperative soft tissue changes, which is a generally unpredictable component. Proposed methods for prediction of soft tissue changes include cephalometric analysis [18] [19], mass tensor models (MTM) [20], finite element models (FEM) [21] [22], machine learning (ML) [23] and deep learning (DL) [24].

Cephalometric analysis uses angles and distances between facial soft and hard tissue landmarks. This method lacks in three dimensional accuracy [21], especially in the upper lip and nose area [25]. The MTM and FEM algorithms are volumetric models and have provided promising results, but studies still have their limitations such as a small number of patients [26] [22] [21]. The trained DL model was compared to and more accurate than MTM, but this was performed exclusively on the lower face [24].

No notable research has been conducted the other way around, on predicting hard tissue shape based on soft tissue shape. As orthognathic surgery prioritizes functionality over aesthetics, surgical planning is generally built around occlusion. Taking the desired soft tissue shape as starting point and working back towards hard tissue changes is opposite to current orthognathic planning methods and may not result in the desired functionality. However, considering the main purpose of FFS is to achieve a specific aesthetic outcome, this could prove to be a suitable method for surgical planning for FFS.

## 2.2 Technical background

In this section, a background is provided on the technical aspects on this thesis. The MeshMonk toolbox, autoencoder network and principal component analysis are explained.

### 2.2.1 MeshMonk

MeshMonk is an open-source customizable toolbox created for phenotyping of 3D meshes. A facial soft tissue template is provided for analysis of facial images, through registration of the template on target images. The MeshMonk registration method has been validated for 19 facial landmarks on n=41 facial 3D images, with an average error of 1.26mm when compared to manual observers. [27], [28]

A readily available validated facial soft tissue registration method combined with supported customisation makes MeshMonk well suited for the purpose of this thesis. Facial soft and hard tissue can be registered in a similar way which allows for analysis of the predictive value of facial shape.

The facial soft tissue template consists of 7 160 vertices, or quasi-landmarks, each defining three coordinates. This template is fitted to the target's shape through two registration steps: a scaled rigid and a non-rigid transformation. During registration, the vertex coordinates form towards the accompanying landmarks of the target face. As each target is registered in the same way by the MeshMonk algorithm, the resulting meshes, or MeshMonk models, have a 1:1 vertex correspondence. The vertex number which denotes the nasion landmark in one target, will also denote the coordinates for the nasion in the other targets.

A facial hard tissue template needed to be created specifically for this research, to be applied to hard tissue targets. Afterwards the resulting hard tissue MeshMonk models will have a 1:1 vertex correspondence as well. The vertex correspondence in addition to applying generalized procrustes analysis (GPA) with scaling, makes it possible to analyse soft and hard tissue model shapes.

### 2.2.2 Autoencoder

Autoencoders are a type of neural network that can be used for unsupervised learning, where the network learns to represent the input data in a latent, often lower-dimensional, space and to reconstruct the data from the latent space. This can be useful for reducing the dimensionality of high-dimensional data, data compression or checking relations between data. One type of autoencoder is the dense autoencoder, which consists of a series of fully connected layers.

For this thesis, a dense autoencoder was created. The principals of this type of autoencoder and the way it was used will be explained in this chapter.

Autoencoders are made up of two parts: an encoder and a decoder, which enable self-supervised learning. The encoder takes the input data and maps it to a lower-dimensional representation, the latent space, while the decoder takes this representation and reconstructs the original input. The latent space serves as a bottleneck for information flow between the encoder and decoder, forcing the network to learn a compressed representation of the input. Dense layers are fully connected layers, meaning that every input neuron of the layer is connected to every output neuron.

One way to use a dense autoencoder to explore relationships between different types of data is to first pre-process the data using Principal Component Analysis (PCA) to reduce the dimensionality of the data.

The pre-processed data can then be fed into a dense autoencoder, which can map the data to a higher-dimensional space, allowing for the exploration of relationships between different types of data. The main benefit of using PCA in this way is that it can help to remove noise and irrelevant information from the input data, which can improve the performance of the autoencoder in smaller data sets. In essence, by letting the autoencoder train only on important information the chances of it recognizing relationships specific to the smaller data set which are not necessarily true for the general population will be significantly reduced. Additionally, by feeding the autoencoder a low dimensional input and increasing the dimensionality in the encoder to the latent space, it can capture more complex relationships between different types of data.

Training the autoencoder involves minimizing the difference between the desired output and the output of the decoder. This is done by defining a loss function that measures the difference between the input and output, and then optimizing this loss function using backpropagation. An often used loss function is the mean squared error (MSE), where larger errors are punished more harshly than smaller errors. On contrary, the mean squared logarithmic error (MSLE) function does not punish larger errors as harshly as MSE and is more robust to outliers.

The gradients of the loss function with respect to the weights of the network are then computed using backpropagation, and the weights are updated using an optimization algorithm, such as adaptive moment estimation (Adam). The advantage of Adam is that it computes individual adaptive learning rates for different parameters and updates them. [29]

Batch normalization can be used to normalize layer inputs within the autoencoder, to counter the changes in distribution of inputs during training. This allows for a higher learning rate and faster learning. [30] Another way to improve the performance is dropout, where neurons of the layer on which it is applied are randomly omitted with a set probability. This is a method designed to reduce overfitting. [31] Another way to limit overfitting is to add Gaussian noise to the input of layers as a regularizing method. These methods were all used in this thesis to improve the performance of the autoencoder.

### 2.2.3 Principal Component Analysis

Principal Component Analysis (PCA) is a method to extract dominant patterns, or important information, from multivariate datasets that are difficult to interpret. The main objectives of PCA are to retrieve important information from the data and to reduce the size of the data set by omitting the less important data. [32], [33]

The extracted information is represented as a set of new, orthogonal, variables: principal components (PCs). These principal components are linear combinations of the original variables. The first principal component has the largest possible variance, while the second principal component is computed orthogonal to the first component with the largest possible variance. All following components are computed in the same way. The result is that each principal component will always be orthogonal to, and account for a lower percentage of the variance than the previous components. The values of these new variables, the principal components, are called scores. These scores describe the projections of the observations on the principal components. [32]

Datasets should be sorted into rows of observations and columns of variables before performing PCA. The principal components can be computed with the covariance method, which is used when variables have similar scale. As the number of observations is smaller than the number of variables in this thesis, the singular value decomposition (SVD) algorithm is used. The following steps give an example of computing the principal components of dataset **A** with dimensions $n \times p$, containing $n$ observations and $p$ variables. [34] [35]

1. Subtract the empirical mean **u** of each column of **A** and store the mean-subtracted data in $n \times p$ matrix **X**

2. Calculate the $n$ x $n$ left singular vector matrix **U**, the $n \times p$ diagonal singular values matrix **S** and the $p \times p$ right singular vector matrix **V** through SVD of matrix **X**:

$$\mathbf{X} = \mathbf{USV}^T \tag{2.1}$$

3. Sort the diagonal singular values matrix **S** in descending order, and permute the matrices **U** and **V** correspondingly

4. Determine the percentage of variance explained by each component, and return these values in vector **e**:

$$\mathbf{e} = \frac{100 * (\mathbf{S}^2/(n-1))}{\Sigma(\mathbf{S}^2/(n-1))} \tag{2.2}$$

5. Determine amount $L$ of principal components needed to account for a desired percentage of the variance through cumulative summing of **e**

6. Let **Q** = $\mathbf{V}^T$ and only keep the first $L$ columns, to obtain $p$ × $L$ matrix **Q**

7. Compute $n$ × $L$ scores matrix **S** by multiplying the matrices **X** and **Q**:

$$\mathbf{S} = \mathbf{XQ} \tag{2.3}$$

In matrix **S**, the first column corresponds to the first principal component, the second column corresponds to the second principal component and so on. These components can be plotted against each other in 2D or 3D plots to show the distribution of the data. Every value of the principal components represents one observation. Therefore, when plotting the components, each dot corresponds with one observation. When differences between classes of the observations within the data account for variances, it will also be visible which components represent these variances as the classes will cluster or be distributed differently. [32] [34]

Matrix **Q** is also called the coefficient matrix. The coefficients of the matrix are used to compute the scores matrix **S**. By design, this matrix can also be used to compute scores of observations not included in the PCA. The new observations $\mathbf{X}_{new}$ need to be centered in the same way as the original observations included in the PCA (Step 1), after which they can be projected on the principal components by multiplication with **Q** to obtain the scores $\mathbf{S}_{new}$:

$$\mathbf{S}_{new} = \mathbf{X}_{new}\mathbf{Q} \tag{2.4}$$

One of the main advantages of PCA is the possibility of reversing the analysis to obtain the original data. By reversing the mathematical computations listed before, the original data can be reconstructed provided all principal components are included. However, when using less principal components, or when using all principal components in an analysis with a larger number of variables than observations, the reconstructed original data will not be exactly the same [32]. This can also be used to reverse the autoencoder output in this thesis, which will consist of scores matrices, to be able to compare the output to the original data. Here it is important to reconstruct the output as well as the original data, with the same amount of principal components, to be able to compare the two.

In this thesis, existing functions within Matlab (Mathworks) will be used to calculate PCA, instead of manually coding the steps explained above.

Chapter 3

# Methods

This chapter states the methods used to explore the predictive value of facial soft tissue (ST) shape for hard tissue (HT) shape. The methods are divided in data preparation and processing and analysis of the results.

As this study focused on the shape of the facial tissues, 3D meshes of soft and hard tissue were used. To scale and align all meshes, the MeshMonk algorithm was used. MeshMonk is centered around facial soft tissue shape, which means that a hard tissue template needed to be created for the application of the algorithm on hard tissue meshes. After alignment all facial meshes were split through the median plane, mirroring the left side to the right. The vertex indices were updated on the mirrored half to ensure vertex correspondence between each ST mesh, and between each HT mesh. This eliminates profound asymmetrical components from influencing the PCA and increased the amount of single subjects for the study.

Before implementing the autoencoder model, PCA was applied. Usage of autoencoder model was chosen due to the ability to recognize non-linear relations between data, however it may be overpowered for use on limited data set size. To prevent it from learning relations which are incidentally true for the data set but not necessarily true for the actual general population, PCA was added to extract the most important shape information. The scores of the principal components (PCs) were used as input and output for the autoencoder. After the autoencoder predicted HT PCA scores to match the input ST PCA scores, the HT mesh was reconstructed from the predicted PCA scores and compared to the real HT mesh.

## 3.1 Data preparation

The data consists of facial 3D photos (.TSB) and Cone Beam Computed Tomography (CBCT) scans (DICOM), which were retrospectively retrieved from the database of the OMFS department of Radboudumc. The data was preprocessed with 3DMedX (Radboudumc) and MATLAB (MathWorks), a HT template needed for processing with MeshMonk was created with Materialise Mimics (Materialise) and Meshmixer (Autodesk).

### 3.1.1 Data selection

The data set was compiled of matching facial ST and HT data (i.e. ST and HT of the same subject) acquired within a one year time frame. The criteria were chosen to obtain a data set which was balanced between the number of subjects and homogeneity between them. Male and female adult Caucasian individuals with a healthy BMI ($\geq$ 18 and $\leq$ 28) and matching ST and HT data were included. Individuals with genetic or acquired craniofacial disfigurements, ST images made within one year of OMFS and low quality images were excluded.

In total, 105 individuals with a mean age of 27.4 years and standard deviation of 7.9 were included. Of these individuals, 26 were included twice and 9 were included three times due to more than one ST and HT set being available; one preoperative and (multiple) postoperative. A total of 149 ST and HT images were included, of which one was used to create the HT template, leaving 148 matching sets.

### 3.1.2 Creation and simplification of 3D meshes

The data was converted to 3D meshes in Wavefront .obj format. ST data was converted using a .TSB to .obj converter from the 3D lab of Radboudumc. The DICOM files were loaded into 3DMedX to create meshes which were then exported as .obj files.

Unlike the ST .TSB files which were already meshes, the required information still had to be extracted from the DICOM files. One by one, the DICOM files were imported and the right tissue was visualized by setting the Hounsfield Unit (HU) lower threshold to approximately -600 for soft tissue and to 276 for hard tissue.

As discussed in 2.2.1, MeshMonk requires five facial landmark coordinates for the initial rough alignment of meshes. In order to align the ST and HT meshes in the same way, additional ST meshes were created from the DICOM files. These meshes were used in the next step to determine the facial landmarks, since they share the same orientation as the HT meshes.

#### Facial landmarks

Five facial landmarks were determined on all ST images from both the 3D photos and the DICOM files: left and right endocanthion, nasal tip and left and right cheilion. This was done in 3DMedX by importing the meshes and assigning the landmarks manually by using the MeshMonk batch landmark workflow. The landmark coordinates were saved in .csv files.

### Pre-processing hard tissue 3D meshes

Unlike the 3D photos, the hard tissue meshes created from DICOM files were complex meshes comprised of multiple surfaces. The DICOM files also had differing values for the Image Position Patient variable, meaning the meshes were scattered around the 3D space. To perform MeshMonk, meshes need to be in vicinity of each other to be processed and preferably have a simple surface. To improve the resulting MeshMonk models the hard tissue meshes were pre-processed.

In MeshLab, Laplacian smoothing was applied and the meshes were reduced to 30 000 faces through quadratic edge decimation with preservation of normals.

Then, in Matlab, an ellipsoid volume was removed from the inner structures of all HT meshes. Hard tissue meshes and corresponding soft tissue landmarks were loaded into Matlab. To ensure ellipsoid location and dimensions which would work for a large number of meshes, the ellipsoid was defined based on the mesh dimensions. The center and axes sizes were found through trial and error, an example can be found in figure 3.1. The ellipsoid center coordinates needed to be posterior on the median plane and were computed as follows:

1. Calculate 3D midpoint between endocanthion landmarks
2. Subtract 0.2 times total length of the mesh in the Y-axis to the midpoint Y-coordinate

The size of the X and Y semi-axes were defined as being 0.3 times the total length of the mesh in the X and Y axes, respectively. The Z semi-axis was defined as 0.8 times the total length of the mesh in the Z-axis. Once the ellipsoid was defined, the vertices and corresponding faces within the ellipsoid volume needed to be identified and removed. To identify these vertices and faces, the standard ellipsoid equation was used:

$$(\frac{x}{a})^2 + (\frac{y}{b})^2 + (\frac{z}{c})^2 = 1 \tag{3.1}$$

where x, y and z are the coordinates of the vertices minus the ellipsoid center coordinates and a, b, and c are the lengths of the semi-axes. For every vertex where the equation is smaller than 1, the corresponding faces and vertices were removed and the remaining indices updated.

By removing most of the inner vertices and faces not needed for the surface mesh, MeshMonk was less likely to wrongfully match surface points to points of the inner skull structures.

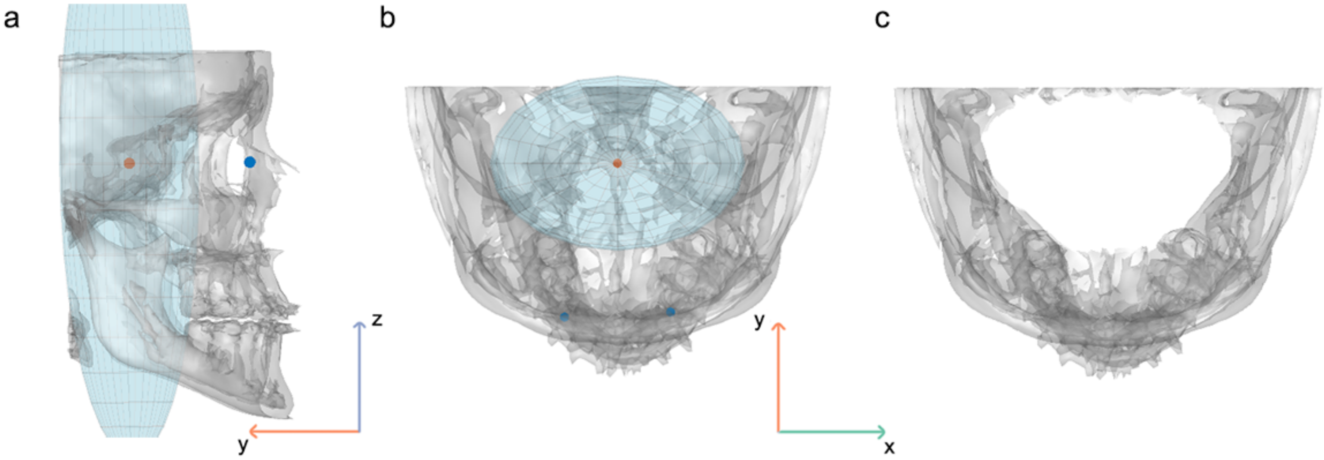


**Figure 3.1:** Visualisation of an ellipsoid within a hard tissue mesh. Red dot indicates the center of the ellipsoid, blue dots indicate the endocanthion landmarks. (**a**) lateral view of the hard tissue mesh and ellipsoid. (**b**) Cranial view of the hard tissue mesh and ellipsoid. (**c**) Cranial view of the hard tissue mesh with ellipsoid volume removed.

### 3.1.3 Hard tissue template

In order to use MeshMonk on the HT data, a HT template was needed which is not provided by MeshMonk. From the dataset, one HT image with a symmetrical shape and good image quality was selected to serve as template. In Materialise Mimics, the hard tissue was segmented and the inner skull structures and dentures removed. Meshmixer was used to remove double surfaces and fill surface holes in the maxilla of the mesh. In MeshLab, a Laplacian smoothing filter was applied to reduce unevenness and the mesh was simplified to 12 000 faces using quadratic edge decimation with preservation of normals.

Finally, in Meshmixer the template was mirrored in the median plane, resulting in a symmetrical template with 6 785 vertices and 12 724 faces depicted in Figure 3.2. In comparison, the ST template provided by MeshMonk (Figure 3.3 consists of 7 160 vertices and 14 050 faces.

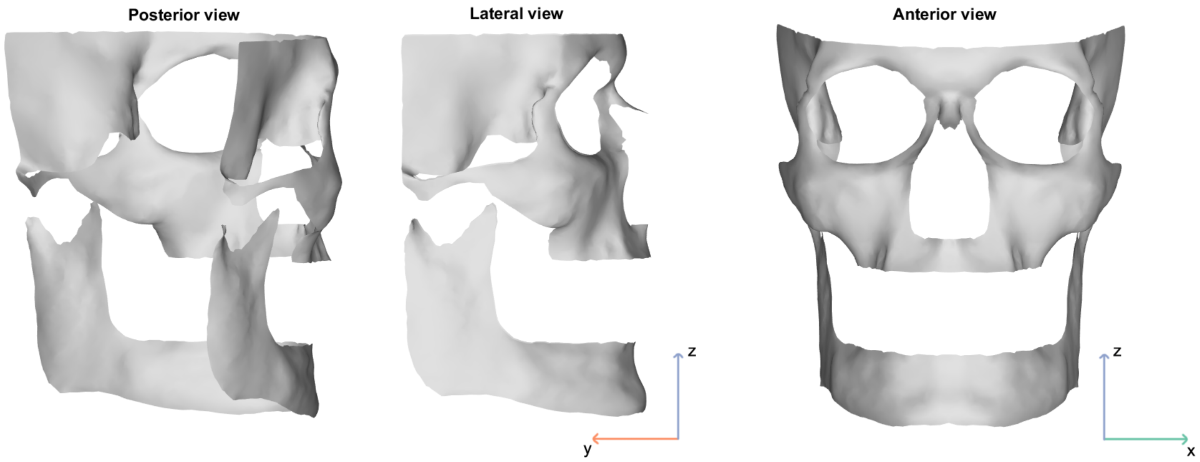


**Figure 3.2:** Posterior, lateral and anterior view of hard tissue template created for use on CBCT data with MeshMonk.

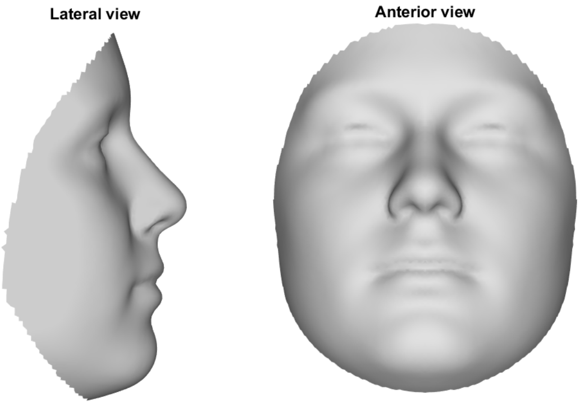


**Figure 3.3:** Lateral and anterior view of the soft tissue template from MeshMonk.

### 3.1.4 MeshMonk

The MeshMonk algorithm was applied in Matlab, using the available open-source code which was edited for batch processing and for application on hard tissue meshes. The meshes were translated towards the origin (0,0,0) by applying a translation to the mesh vertices. The translation vector was calculated by subtracting the Image Position Patient value, defined in the DICOM info, from the origin.

Then, MeshMonk was performed on ST and HT meshes with the ST and HT template, respectively. The initial rigid transformation was set at 30 iterations, the non-rigid transformation was set at 400 iterations. Both rigid and non-rigid transformation had scaling and use of surface normal direction to determine outliers enabled. The full parameter settings can be found in the used Matlab script in **??**.

### 3.1.5 Split meshes and final exclusion

To prevent asymmetry from being recognized as the most important component by PCA, the ST and HT meshes were split in two through the median plane. Of the resulting left and right half meshes, the left side was mirrored. The vertex indices were then updated to guarantee vertex correspondence between meshes of the same tissue type. This resulted in a uniform data set of right half ST and HT facial meshes, which enlarged the data set size and eliminated the asymmetrical component.

The soft and hard tissue meshes were split separately in Matlab. Defining the left and right side was determined by the templates, since these are symmetrical the median plane is exactly defined and vertices belonging to either side can be easily identified. Through vertex correspondence between meshes due to MeshMonk, left and right sides of all meshes could then be extracted and saved separately.

First, the vertex corresponding with the nasion landmark was identified in the templates. The vertices were translated so that the nasion vertex corresponded with the origin (0,0,0), ensuring the median plane was defined as the YZ-plane through x = 0. Vertices with x-coordinates below or equal to 0 corresponded with the right half. By identifying and removing faces and their corresponding vertices with x-coordinates higher than 0, the right half was calculated. By default, the leftover faces and their corresponding vertices defined the left halves.

The left template halves were mirrored by multiplying the X-coordinates of the vertices with -1. The left half vertices coordinates were then identical to the right half vertices, through nearest neighbor search function in Matlab the matching vertex indices were found and adjusted to guarantee vertex correspondence. Finally, after copying the faces from the right half to the mirrored left half, the meshes were identical. The soft tissue meshes had 3633 vertices, compared to 3420 for the hard tissue meshes. The ST and HT templates which were cut in half through the median can be found in figures 3.4 and 3.5.

The vertex indices were saved and used to mirror all left half meshes. Each half mesh was individually visually inspected for large errors of the Meshmonk algorithm, only meshes with errors that would reasonably not show up in PCA were saved and used further on. The final data set consisted of 252 right half matching soft (3363 x 3 x 252) and hard (3420 x 3 x 252) tissue mesh sets, 133 preoperative and 119 postoperative sets of a total of 104 individuals.

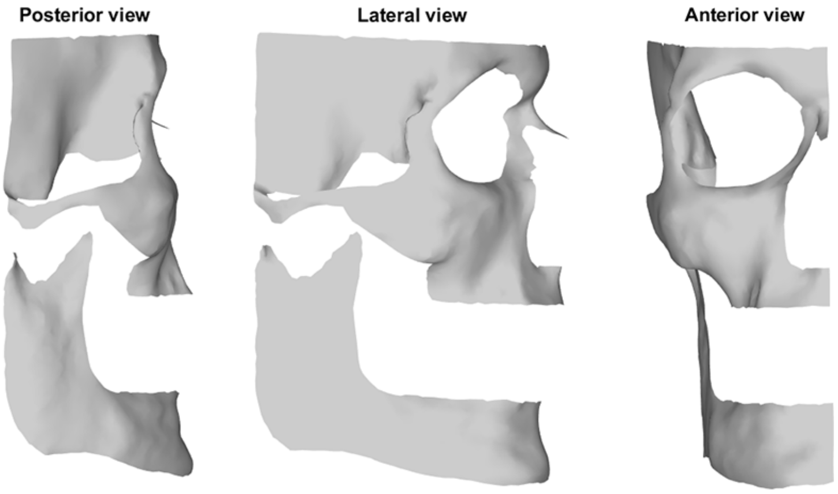


**Figure 3.4:** Posterior, lateral and anterior view of the median split hard tissue template.

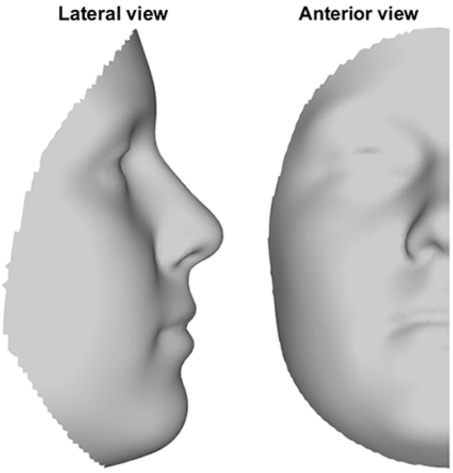


**Figure 3.5:** Lateral and anterior view of the median split soft tissue template.

## 3.2 Data processing

A PCA was performed on hard and soft tissue meshes to obtain the scores matrices. These were normalized and then used to train, validate and test the autoencoder. The resulting predicted hard tissue scores of the test set were then reconstructed to hard tissue meshes and compared to the original meshes. To measure the performance of the autoencoder and not of the PCA, the original meshes were reconstructed with the same number of principal components as were used as input and output of the autoencoder. The general methods of the data processing are visualized in Figure 3.6, with a more detailed workflow included in Figure 3.7 on page 23.

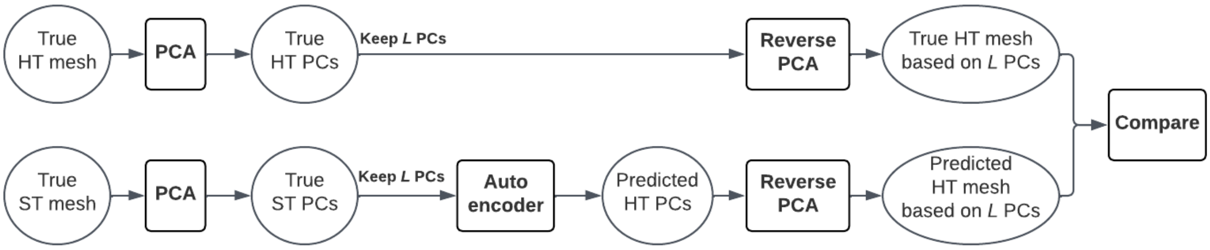


**Figure 3.6:** Data processing workflow. With *L* the chosen number of principal components for the lower dimensional representation of the data (see 2.2.3).

### 3.2.1 Principal component analysis

The PCA was performed in Matlab according to the principles explained in section 2.2.3 to calculate the scores matrices. The methods for ST and HT meshes were similar, but not identical and will be elaborated upon separately. The supporting scripts can be found in Appendix A.1.

#### Hard tissue

The input for the analysis were the vertex coordinates of the meshes, stored in 3420 x 3 x 252 data matrix **A**. This gives a total of 3420 vertices (*v*) each with three dimensions (*d*) for 252 subjects (*n*). First, a generalized procrustes analysis (GPA) was performed and the transformation matrix from the GPA was used to transform the meshes towards the mean of **A** over all subjects *n*, resulting in 3420 x 3 x 252 data matrix **X**. The 1 x 252 scaling factor **f** for each subject was saved, to be able to scale the vertices back to their original size later. This was necessary to be able to approximate the error in mm, as 1 unit in the 3D photos and 1 unit in the CBCT images equals 1 mm.

The PCA itself was based on preoperative meshes, to reduce influence of implants in the postoperative meshes on the analysis. First, matrix **X** was reshaped to the size of 133 x 10260, where each row *n* represented one subject and each column *j* represented one dimension of one vertex. The mean *v* of every column of **X** was calculated and subtracted from **X** and returned in matrix **Y**.

$$\mathbf{v}_j = \frac{1}{n}\sum_{n=1}^{133}(\mathbf{X}_{n,j}) \tag{3.2}$$

$$\mathbf{Y}_{n,j} = \mathbf{X}_{n,j} - \mathbf{h}\mathbf{v}_j \tag{3.3}$$

With **h** a *n* x 1 column of ones. Vector **v** was saved for PCA of postoperative meshes and reconstruction later on. With the PCA function in Matlab, the analysis was performed on **Y**. For the exact computation used in this function, see 2.2.3. The 10260 x 132 coefficient matrix **C**, the 133 x 132 scores matrix $\mathbf{S}_{preop}$ and vector **e** with the percentage of variance explained per component with length 132 were saved.

Through cumulative summing of **e**, the number of principal components needed to account for 80$\%$ of the variance was found to be seventeen. The amount of variance explained per component has dropped to 1.2$\%$ in the seventeenth component. Higher components will each explain a decreasing amount of variance and will add less and less information. The first seventeen principal components were therefore chosen to represent the soft and hard tissue models in the autoencoder.

The scores of the 119 postoperative meshes were calculated using **C**. The 3420 x 3 x 119 data matrix **M** of postoperative meshes was reshaped to the size of 119 x 10260, where each row *m* represented one subject and each column *j* represented one dimension of one vertex. Vector **v** was subtracted from **M** and returned in vector **Z**.

$$\mathbf{Z}_{m,j} = \mathbf{M}_{m,j} - \mathbf{h}\mathbf{v}_j \tag{3.4}$$

With **h** a *n* x 1 column of ones. Then, the 119 x 132 scores matrix $\mathbf{S}_{postop}$ was calculated by multiplying **Z** with **C**,

$$\mathbf{S}_{postop} = \mathbf{Z}_{m,j} * \mathbf{C}_{j,n-1}{}^{1} \tag{3.5}$$

Before using the scores matrices as input for the autoencoder, the preoperative and postoperative scores matrices were concatenated to *m*+*n* x *n*-1, or 252 x 132, matrix $\mathbf{S}_{orig}$ and normalized by subtracting the 1 x 132 column mean $\mathbf{s}_{mu}$ and then dividing by the 1 x 132 column standard deviation vector $\mathbf{s}_s$:

$$\mathbf{s}_{mu} = \frac{1}{n-1}\sum_{n=1}^{132}(\mathbf{S}_{orig}) \tag{3.6}$$

$$\mathbf{s}_s = \sqrt{\frac{\sum(\mathbf{S}_{orig} - \mathbf{s}_{mu})}{n}} \tag{3.7}$$

$$\mathbf{S}_{orig} = \frac{\mathbf{S}_{orig} - \mathbf{s}_{mu}}{\mathbf{s}_s} \tag{3.8}$$

All rows and the first seventeen columns of $\mathbf{S}_{orig}$, $\mathbf{s}_{mu}$ and $\mathbf{s}_s$ were saved to use with the autoencoder and the reconstruction of data later on.

[1] Note that **C** has *n*-1 columns, as there are more variables (*v***d* = 10 260) than observations (*n* = 133) included in the PCA

### Soft tissue

The PCA for the soft tissue meshes followed the same method, with the exception of the scaling factor of the GPA. The scaling factor of the soft tissue mesh of each subject equalled the scaling factor of the matching hard tissue mesh.

The number of dimensions and subjects was identical, but the number of soft tissue vertices *v* was 3633. The analysis was performed on 133 preoperative subjects, the 119 postoperative subjects' scores were calculated through multiplication of the mean-centered data matrix with the coefficient matrix of the soft tissue PCA. The preoperative and postoperative scores matrices were concatenated and normalized through subtraction of the mean and division by the standard deviation. To create a dataset with equal dimensions between soft and hard tissue data, the first seventeen columns of this matrix were saved to use with the autoencoder.

## 3.2.2 Autoencoder

The autoencoder was created in Python (version 3.9.13, Python Software Foundation) using Keras (version 2.12.0, F. Chollet et al). The Python scripts can be found in A.2. Table 3.1 provides an overview of the demographic variables of the included subjects.

**Table 3.1:** Demographic variables of included subjects.

| | Number (mean ± SD) | |
|---|---|---|
| Variable | All subjects | Test subjects |
| Female | 77 | 13 |
| Male | 27 | 8 |
| Age at time of 3D image | 27.4 ± 7.9 | 25.5 ± 8.5 |
| Included meshes per subject | 2 ± 1 | 1 |

Two PCA scores matrices, 252 x 17 ST and 252 x 17 HT, with rows of subjects and columns of principal components were used as input and output of the autoencoder. ST scores were used as input and HT scores as output, to force the network to learn the relation between the two. The data sets were divided into training, validation and test sets of 200 x 17, 31 x 17 and 21 x 17, respectively. The test set was chosen from individuals which had only one half face mesh included, so no resembling other halves were present in the training or validation set. The architecture of the encoder and decoder were as follows:

- **Input layer**, specifying the input shape as a vector of length 17

- **Encoder**, using several dense layers to transform the input data into a higher-dimensional representation
    1. Dense layer with 51 neurons and ReLu activation function
        - Batch normalization with momentum of 0.8
        - Dropout rate of 0.35
    2. Dense layer with 51 neurons and ReLu activation function
        - Add Gaussian noise with standard deviation of 1
        - Batch normalization with momentum of 0.75
        - Dropout rate of 0.3
    3. Dense layer with 68 neurons and ReLu activation function
        - Add Gaussian noise with standard deviation of 1
- **Decoder**, using two layers to transform the encoded data back to the input shape
    1. Dense layer with 68 neurons and ReLu activation function
        - Batch normalization with momentum of 0.75
        - Dropout rate of 0.3
    2. Dense layer with 51 neurons and ReLu activation function
        - Batch normalization with momentum of 0.75
- **Output layer**, returns vector of length 17 with linear activation function which gives the transformed input data

The autoencoder was trained with early stopping with a patience of 30 and a maximum of 1000 epochs. The class weights were set equal to the variance explained by the corresponding principal component. The batch size was set at 16. An adaptive moment estimation (Adam) optimizer was used with a learning rate of 0.001 and a $\beta_1$ and $\beta_2$ of 0.9 and 0.999, respectively. The loss function was set as mean squared logarithmic error:

$$\textit{Loss} = \frac{1}{N}\sum_{n=1}^{N}(log(y_i + 1) - log(\hat{y}_i + 1))^2 \tag{3.9}$$

With *y* the real score value, *ŷ* the predicted score value and *N* the number of subjects. The training and validation losses were tracked and plotted. After training, the autoencoder predicted the output for the test set, the hard tissue scores matrix, which was saved.

A 5-fold cross validation was performed to ensure that the performance of the autoencoder did not depend on the selected validation subset.

### 3.2.3 Reverse principal component analysis

With the predicted hard tissue score matrix of the autoencoder available, this needed to be transformed back to vertex coordinates to be able to compare it to the true coordinates, which were the desired result from the PCA and autoencoder combination. The PCA was reversed by performing the inverse of the steps performed in 3.2.1, in reverse order. The reconstructed meshes from the autoencoder generated PCA scores were compared to the original 17 PC reconstructed meshes by calculating the mean absolute error between every vertex coordinate in the x, y and z axes. Let *n* be 252 total subjects, *v* 3420 vertices, *d* three dimensions, *x* 21 test subjects, *k* 17 principal component columns and *j* 10260 variables.

1. Load variables
   - Autoencoder predicted 21 x 17 hard tissue scores matrix $\mathbf{S}_{pred,x,k}$
   - Vertex coordinates mean 1 x 10260 vector $\mathbf{v}_j$
   - PCA computed 10260 x 17 hard tissue coefficient matrix $\mathbf{C}_{j,k}$
   - PCA computed 252 x 17 hard tissue scores matrix $\mathbf{S}_{true,n,k}$ and 1x17 mean vector $\mathbf{s}_{mu,k}$ and 1 x 17 standard deviation vector $\mathbf{s}_{s,k}$
   - GPA computed 1 x 21 scaling vector $\mathbf{f}_x$

2. Reverse normalization to obtain the reconstructed 21 x 17 hard tissue scores matrix $\mathbf{S}_{rec}$

$$\mathbf{S}_{rec,x,k} = \mathbf{S}_{pred,x,k} * \mathbf{s}_{s,k} + \mathbf{s}_{mu,k} \tag{3.10}$$

3. Calculate the mean absolute error (MAE) and mean error (ME) between every test subject for each PC

$$MAE_k = \frac{\sum_{x=1}^{X} |\mathbf{S}_{true,x,k} - \mathbf{S}_{rec,x,k}|}{X} \tag{3.11}$$

$$ME_k = \frac{\sum_{x=1}^{X} \mathbf{S}_{true,x,k} - \mathbf{S}_{rec,x,k}}{X} \tag{3.12}$$

   with *X* the total number of test subjects, 21.

4. Approximate 21 x 10260 data matrix $\mathbf{X}_{rec,x,j}$ and add mean vector $\mathbf{v}_j$:

$$\mathbf{X}_{rec,x,j} = \mathbf{S}_{rec,x,k} * \mathbf{C}'_{j,k} \tag{3.13}$$

$$\mathbf{X}_{rec,x,j} = \mathbf{X}_{rec,x,j} + \mathbf{hv}_j \tag{3.14}$$

5. Reshape $\mathbf{X}_{rec,x,j}$ to 3420 x 3 x 21 and reverse GPA scaling by dividing by $\mathbf{f}$:

$$\mathbf{X}_{rec,v,d,x} = \frac{\mathbf{X}_{rec,v,d,x}}{\mathbf{f}_x} \tag{3.15}$$

6. Reconstruct 3420 x 3 x 21 true mesh data matrix $\mathbf{X}_{true,v,d,x}$ with 17 principal components by repeating steps 3 and 4 with $\mathbf{S}_{true,x,k}$

$$\mathbf{X}_{true,x,j} = \mathbf{S}_{true,x,k} * \mathbf{C}'_{j,k} \tag{3.16}$$

$$\mathbf{X}_{true,x,j} = \mathbf{X}_{true,x,j} + \mathbf{hv}_j \tag{3.17}$$

$$\mathbf{X}_{true,v,d,x} = \frac{\mathbf{X}_{true,v,d,x}}{\mathbf{f}_x} \tag{3.18}$$

7. Calculate the mean absolute error (MAE) and mean error (ME) between every vertex coordinate in the x, y and z axes for each test subject *x*

$$MAE_{d,x} = \frac{\sum_{v=1}^{V} |\mathbf{X}_{true,v,d,x} - \mathbf{X}_{rec,v,d,x}|}{V} \tag{3.19}$$

$$ME_{d,x} = \frac{\sum_{v=1}^{V} \mathbf{X}_{true,v,d,x} - \mathbf{X}_{rec,v,d,x}}{V} \tag{3.20}$$

with *V* the total number of vertices, 3420.

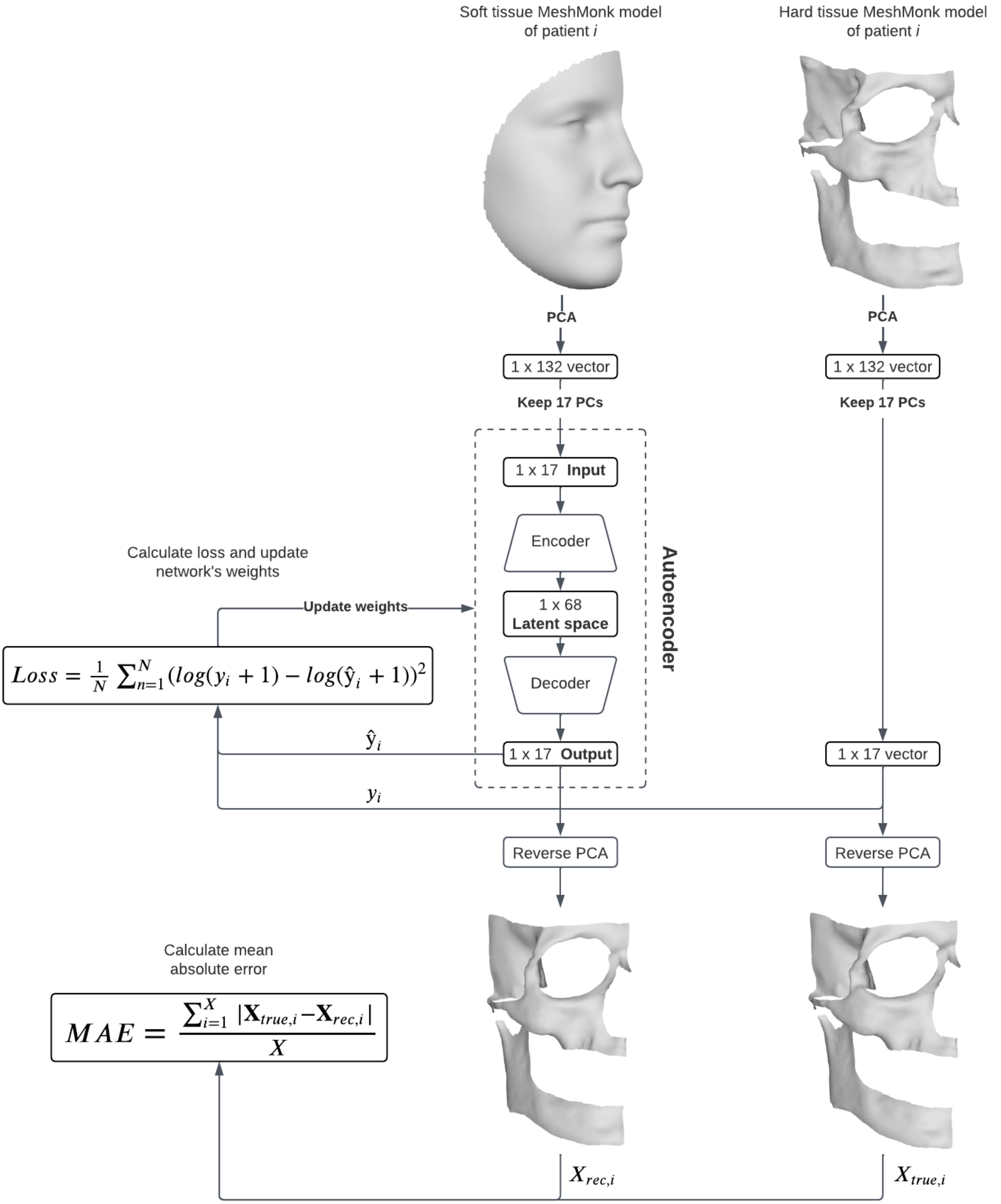

Soft tissue MeshMonk model of patient *i*
Hard tissue MeshMonk model of patient *i*
PCA
PCA
1 x 132 vector
1 x 132 vector
Keep 17 PCs
Keep 17 PCs
1 x 17 Input
Encoder
1 x 68 Latent space
Decoder
1 x 17 Output
Autoencoder
1 x 17 vector
Calculate loss and update network's weights
Update weights
$Loss = \frac{1}{N}\sum_{n=1}^{N}(log(y_i + 1) - log(\hat{y}_i + 1))^2$
$\hat{y}_i$
$y_i$
Reverse PCA
Reverse PCA
Calculate mean absolute error
$MAE = \frac{\sum_{i=1}^{X} |\mathbf{X}_{true,i} - \mathbf{X}_{rec,i}|}{X}$
$X_{rec,i}$
$X_{true,i}$

**Figure 3.7:** Data processing workflow.

## Chapter 4

# Results

In this chapter, results of the PCA, the autoencoder model and the final hard tissue reconstruction are stated.

## 4.1 Principal component analysis

Table 4.1 shows the percentages and cumulative percentages of variance explained by the first twenty soft and hard tissue PCs. The values of the seventeenth PC, the last PC included, is shown in bold. Figure 4.1 shows the plotted cumulative percentages of Table 4.1.

**Table 4.1:** Percentages and their cumulative of variance explained by the first twenty PCs of the PCA on soft and hard tissue vertices vertices.

| | Percentage (cumulative) | | | | | Percentage (cumulative) | | | |
|---|---|---|---|---|---|---|---|---|---|
| PC | Soft tissue | | Hard tissue | | PC | Soft tissue | | Hard tissue | |
| 1 | 38.7 | (38.7) | 17.4 | (17.4) | 11 | 1.1 | (90.3) | 2.0 | (71.6) |
| 2 | 21.6 | (60.3) | 13.9 | (31.3) | 12 | 1.0 | (91.3) | 1.9 | (73.6) |
| 3 | 8.5 | (68.8) | 9.7 | (41.0) | 13 | 0.7 | (92.1) | 1.7 | (75.2) |
| 4 | 5.6 | (74.4) | 6.8 | (47.8) | 14 | 0.7 | (92.7) | 1.5 | (76.7) |
| 5 | 4.2 | (78.6) | 5.1 | (53.0) | 15 | 0.5 | (93.3) | 1.4 | (78.1) |
| 6 | 3.3 | (81.9) | 4.7 | (57.6) | 16 | 0.5 | (93.8) | 1.2 | (79.3) |
| 7 | 2.9 | (84.8) | 4.2 | (61.7) | **17** | **0.4** | **(94.2)** | **1.2** | **(80.5)** |
| 8 | 1.8 | (86.6) | 3.1 | (64.8) | 18 | 0.4 | (94.6) | 1.1 | (81.6) |
| 9 | 1.4 | (88.0) | 2.9 | (67.6) | 19 | 0.3 | (94.9) | 1.0 | (82.5) |
| 10 | 1.2 | (89.2) | 2.0 | (69.7) | 20 | 0.3 | (95.2) | 0.8 | (83.4) |

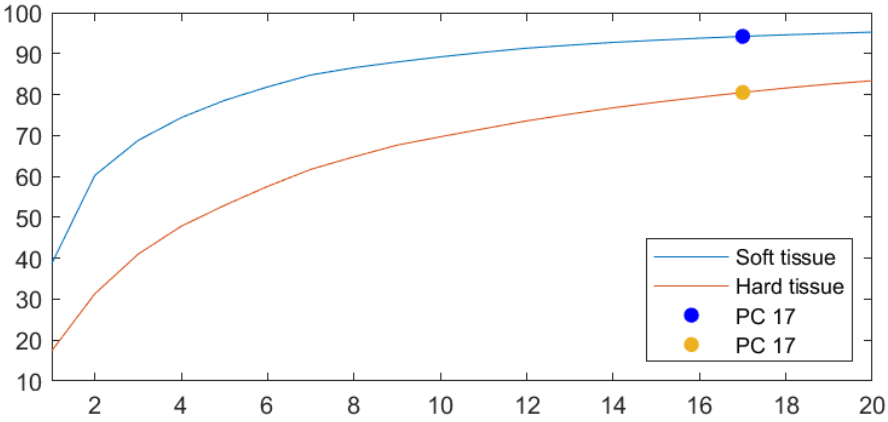


**Figure 4.1:** Plot of the cumulative percentage of variance explained per principal component.

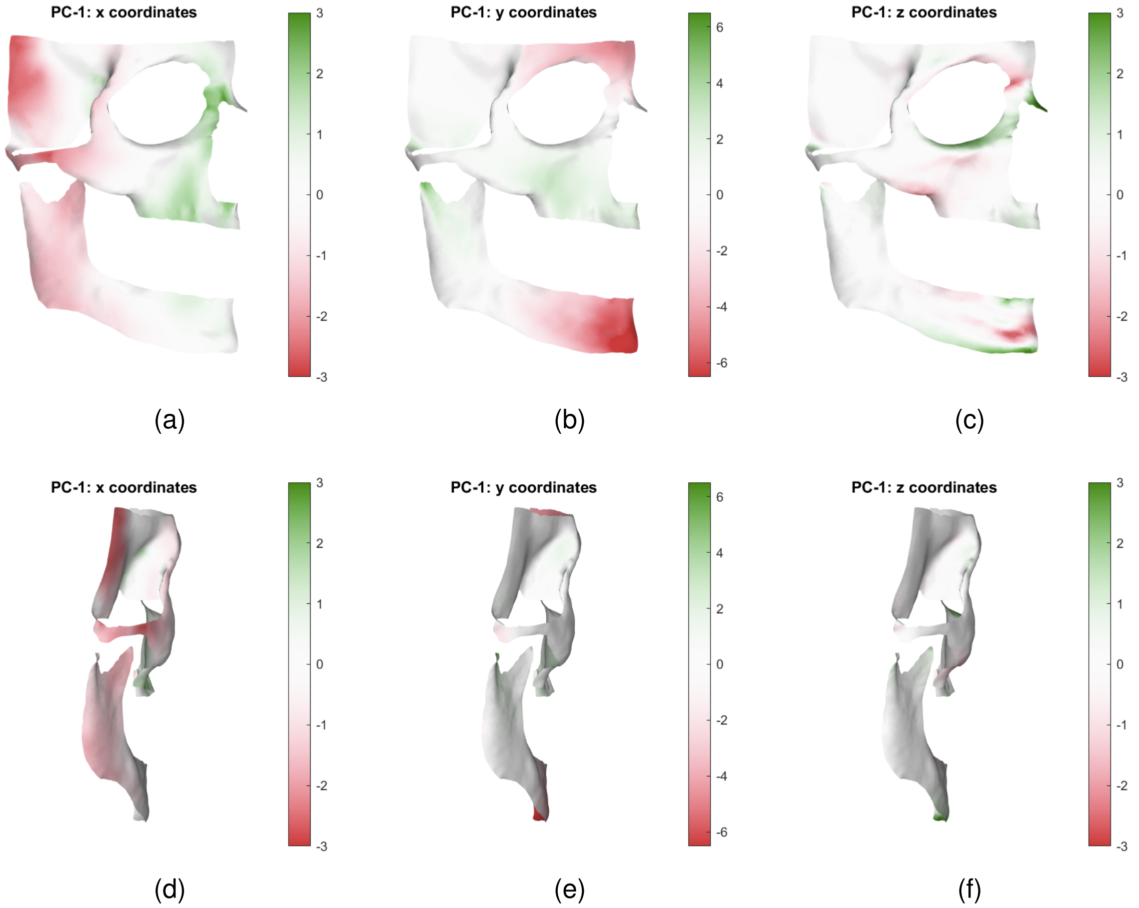
Table 4.2 shows the standard deviation of the true PC scores, next to the mean absolute error and the mean error of the predicted PC scores per PC, which are also expressed as percentage of the standard deviation.

**Table 4.2:** PC scores standard deviation and error summary measures, expressed as mean absolute error, mean error and as percentage of the PC scores standard deviation.

| PC | Scores SD | MAE | (% SD) | ME | (% SD) | PC | Scores SD | MAE | (% SD) | ME | (% SD) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ±82.9 | 73.8 | (89) | 5.6 | (7) | 10 | ±29.5 | 28.8 | (98) | -9.1 | (31) |
| 2 | ±72.6 | 68.3 | (94) | -23.5 | (32) | 11 | ±26.7 | 27.2 | (102) | -3.9 | (14) |
| 3 | ±64.3 | 31.6 | (49) | 2.3 | (4) | 12 | ±27.9 | 22.2 | (79) | -9.5 | (34) |
| 4 | ±50.1 | 49.6 | (99) | 8.6 | (17) | 13 | ±29.9 | 24.6 | (82) | -5.6 | (19) |
| 5 | ±45.2 | 43.9 | (97) | 6.4 | (14) | 14 | ±25.0 | 22.4 | (90) | 7.6 | (30) |
| 6 | ±45.8 | 30.5 | (66) | 8.4 | (18) | 15 | ±24.4 | 16.2 | (66) | -4.7 | (19) |
| 7 | ±42.6 | 41.2 | (97) | -14.2 | (33) | 16 | ±21.6 | 22.1 | (102) | -14.9 | (69) |
| 8 | ±34.9 | 30.7 | (88) | -11.5 | (33) | 17 | ±23.9 | 22.9 | (96) | -8.3 | (35) |
| 9 | ±33.8 | 24.0 | (71) | -2.7 | (8) | | | | | | |

Figures 4.2 through 4.7 show colormaps of the component of the three SD scaled transformation of the x, y and z coordinates in the direction of the vertex normals for PCs 1, 2, 4, 5, 7 and 9. These colormaps are displayed on a mesh of the mean vertex coordinate of all subjects.

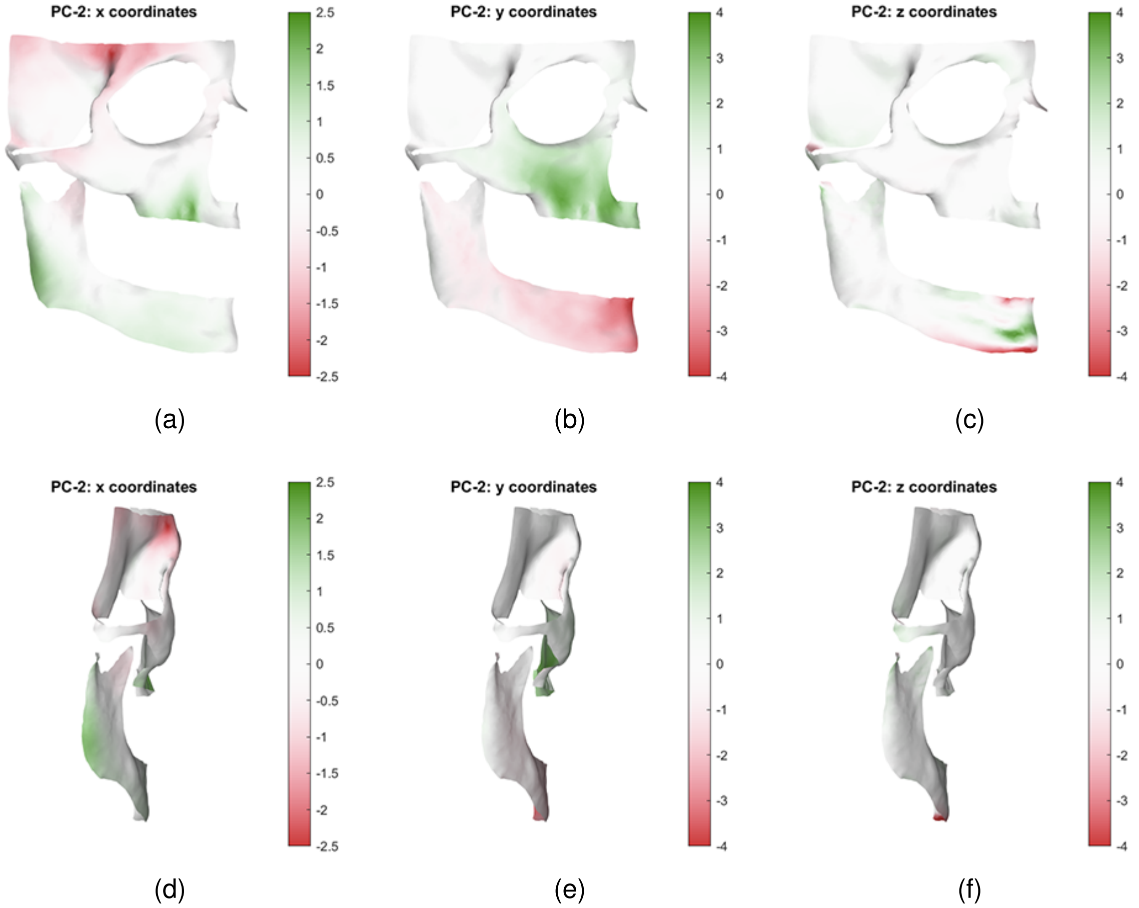
These figures were deemed the most relevant to be displayed here, the figures for all first twelve principal components can be found in Appendix A.1.



(a) (b) (c)

(d) (e) (f)

**Figure 4.2:** Transformation of x, y and z in PC 1 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

**Figure 4.3:** Transformation of x, y and z in PC 2 in the direction of the vertex normals.

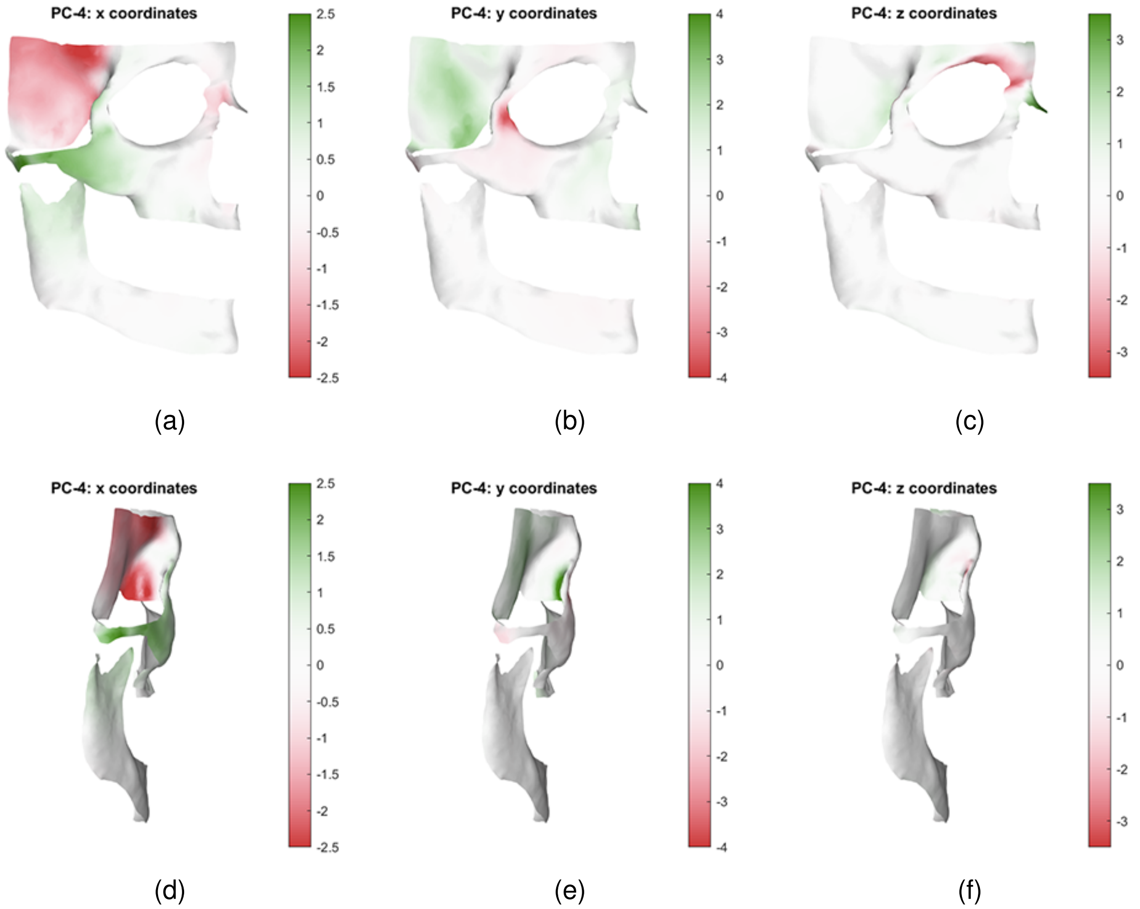


**Figure 4.4:** Transformation of x, y and z in PC 4 in the direction of the vertex normals.

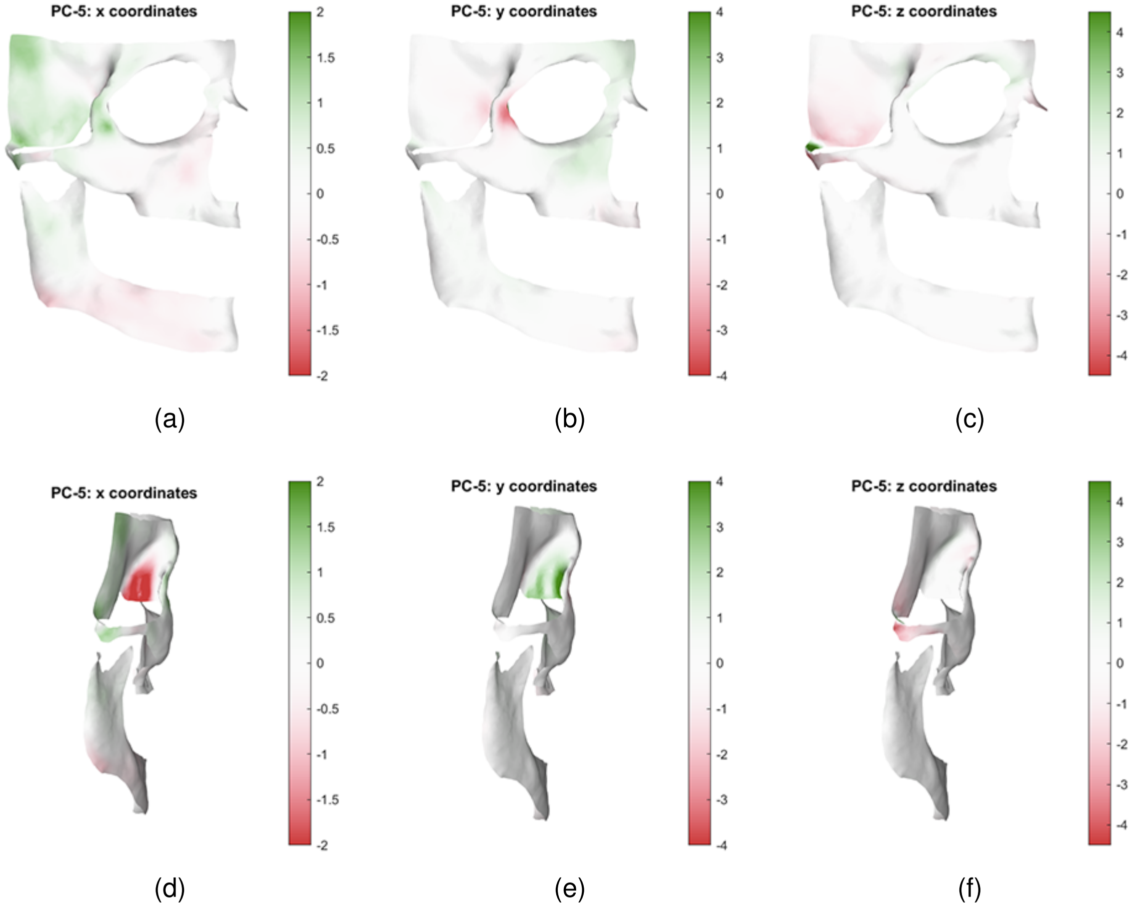


**Figure 4.5:** Transformation of x, y and z in PC 5 in the direction of the vertex normals.

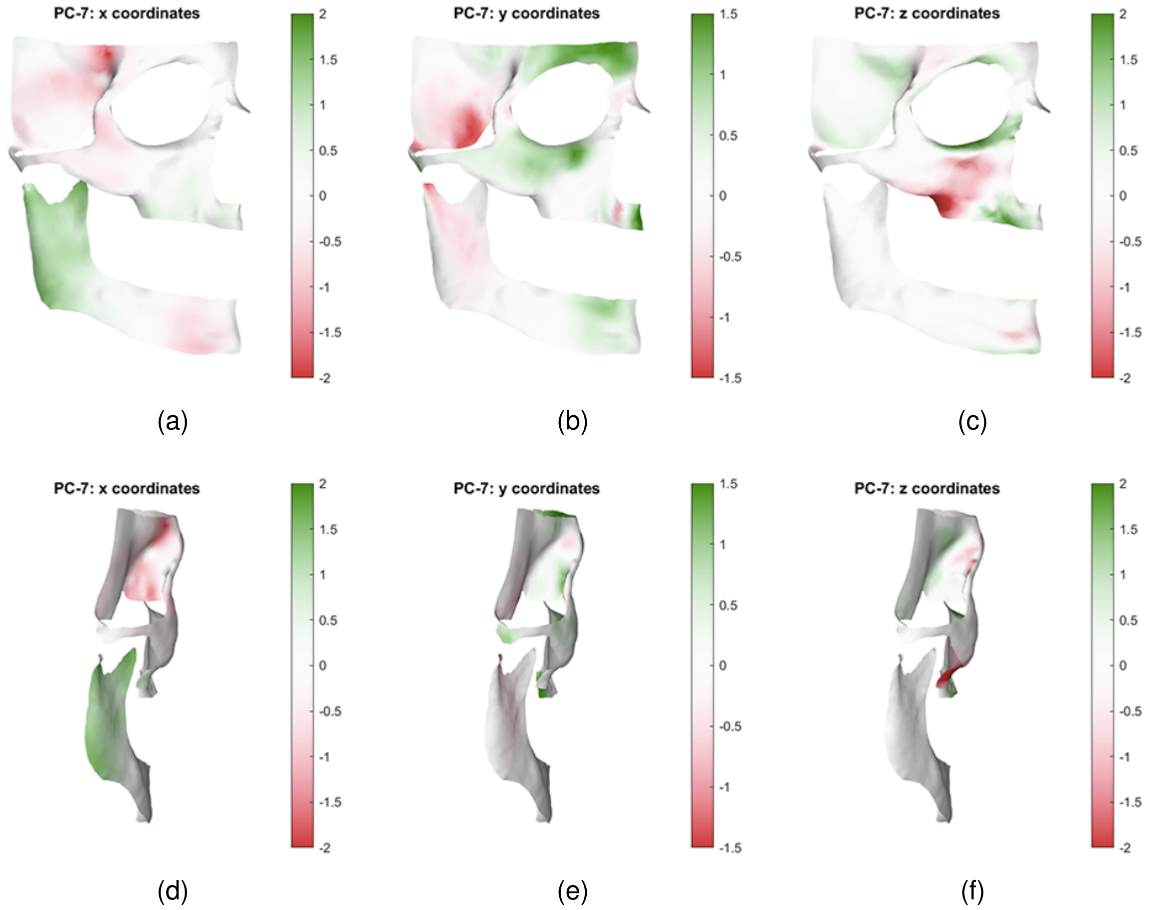


**Figure 4.6:** Transformation of x, y and z in PC 7 in the direction of the vertex normals.

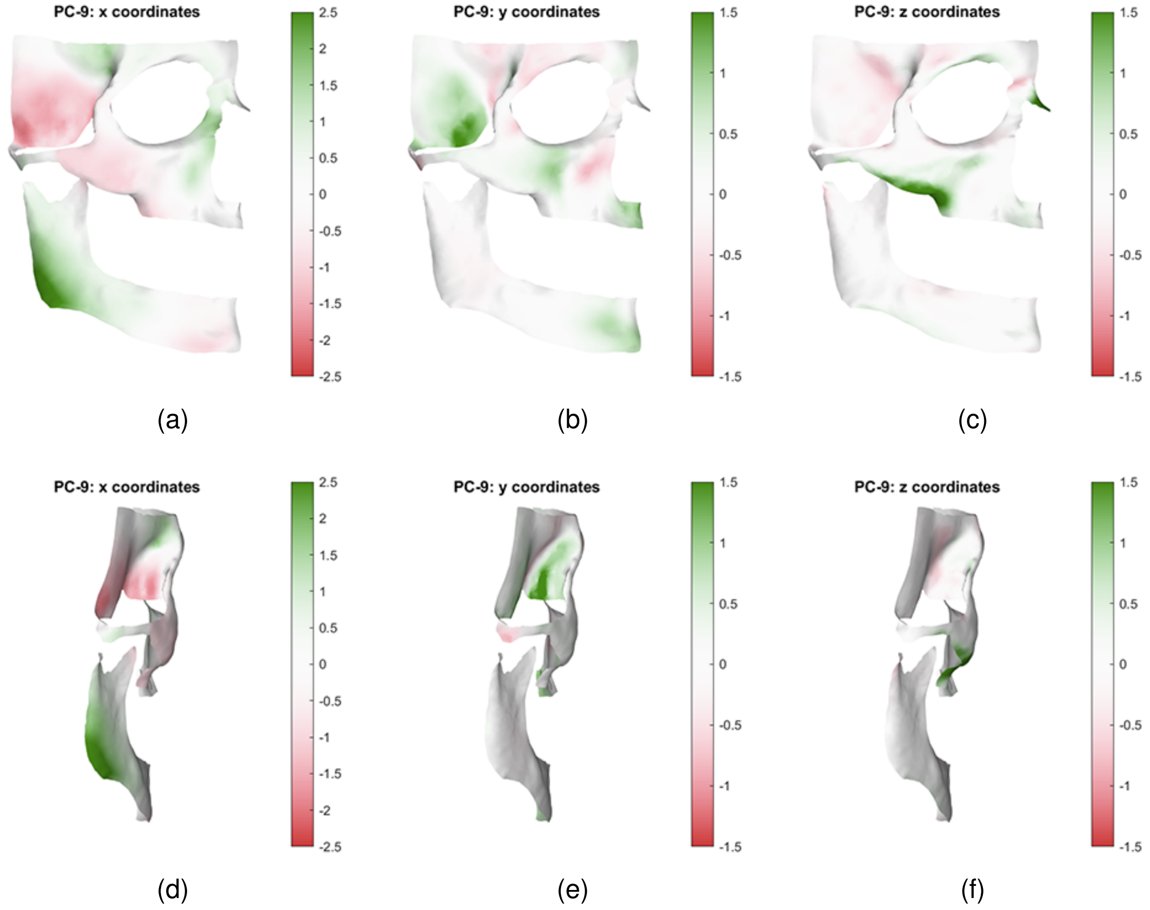


**Figure 4.7:** Transformation of x, y and z in PC 9 in the direction of the vertex normals.

## 4.2 Autoencoder

The loss plot displaying the training and validation loss can be found in 4.8. Training was stopped after 687 epochs by the early stopping callback. The validation losses of the 5-fold cross validation had a mean of 0.12 with a standard deviation of 0.01 at the end of training.

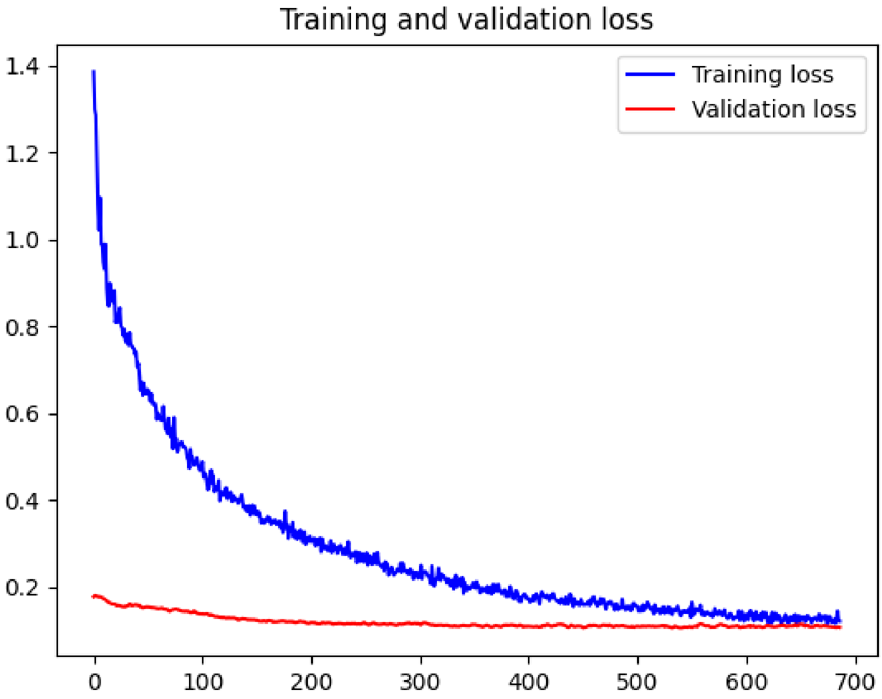


**Figure 4.8:** Training and validation loss plot of the autoencoder.

## 4.3 Reconstruction

The orientation of the axes with x in the lateral direction, y in the coronal direction and z in the vertical direction is depicted in Figure 4.9:

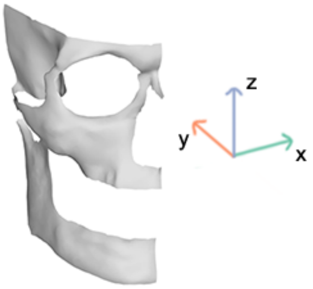


**Figure 4.9:** Hard tissue template with the direction of the coordinate axes.

Figure 4.10 shows the histograms of the absolute errors in three dimensions of the from PCA scores reconstructed vertices of 21 test subjects. Summary metrics of these absolute errors are provided in Table 4.3. The mean errors approximated zero in every dimension for each test subject.

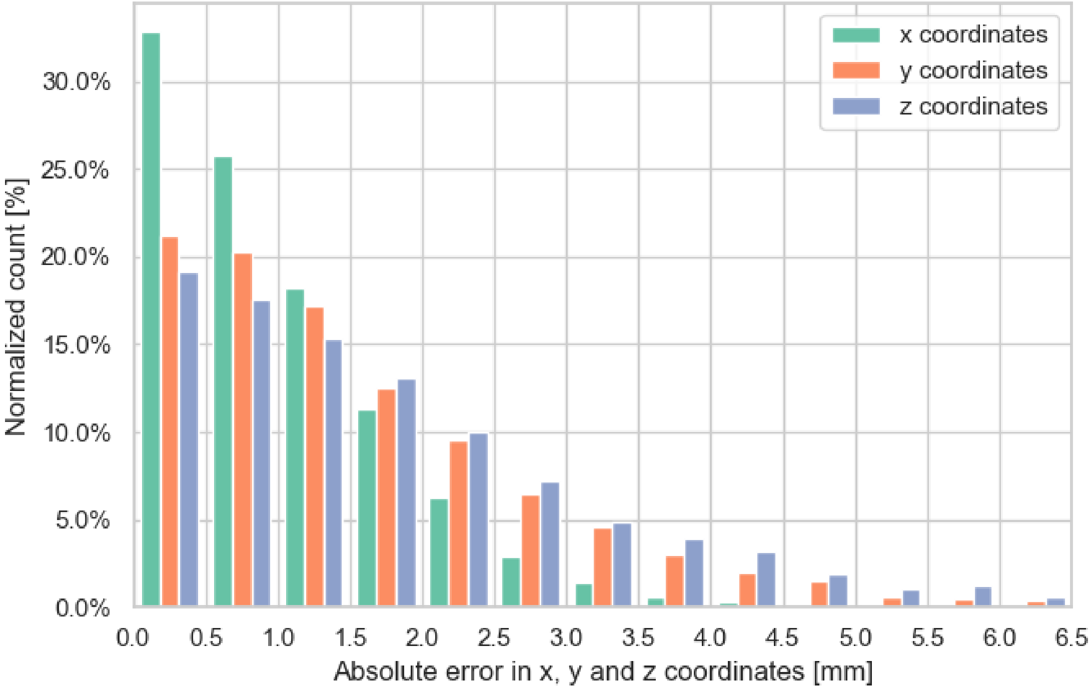


**Figure 4.10:** Histograms of the mean absolute errors in the vertex coordinates of 21 test subjects.

**Table 4.3:** Summary metrics of absolute errors in vertex coordinates of 21 reconstructed hard tissue images in millimeters.

| Variable | x-coordinates (mm) | y-coordinates (mm) | z-coordinates (mm) |
|---|---|---|---|
| Median | 0.8 | 1.2 | 1.4 |
| Median absolute deviation | 0.6 | 1.0 | 1.1 |
| Third quartile | 1.4 | 2.2 | 2.4 |

Distance maps of the mean absolute error of every vertex over each subject in the x, y and z coordinates can be found in figures 4.11, 4.13 and 4.15. Additionally, the mean absolute error of coordinates that showed absolute errors above the third quartile in ten or more reconstructed meshes are highlighted in figures 4.12, 4.14 and 4.16. These mean absolute error values include a total of 342, 367 and 414 errors in the x, y and z coordinates, respectively.

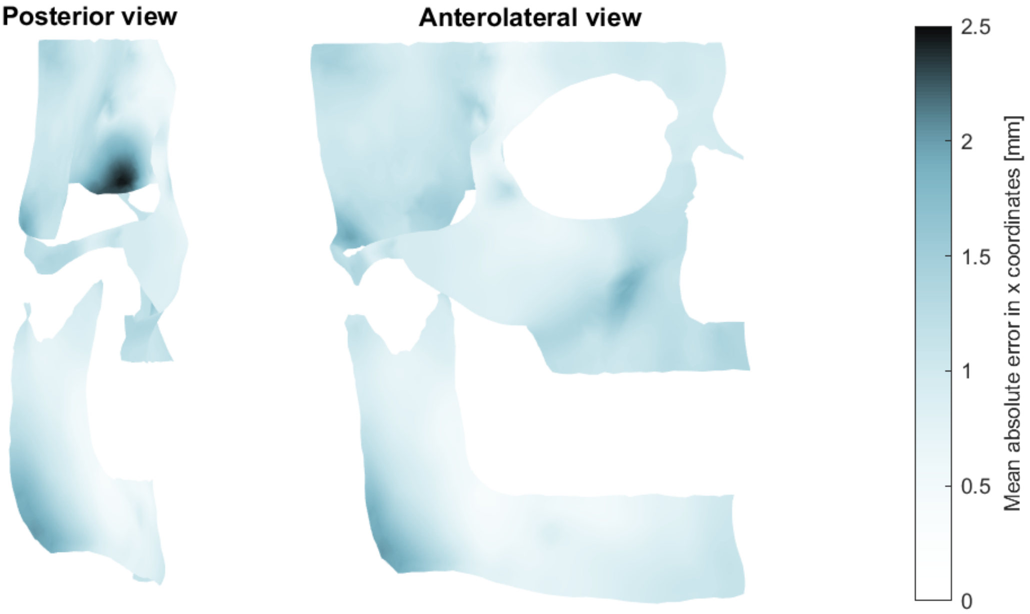


**Figure 4.11:** Distance map of the mean absolute error in the x coordinates.

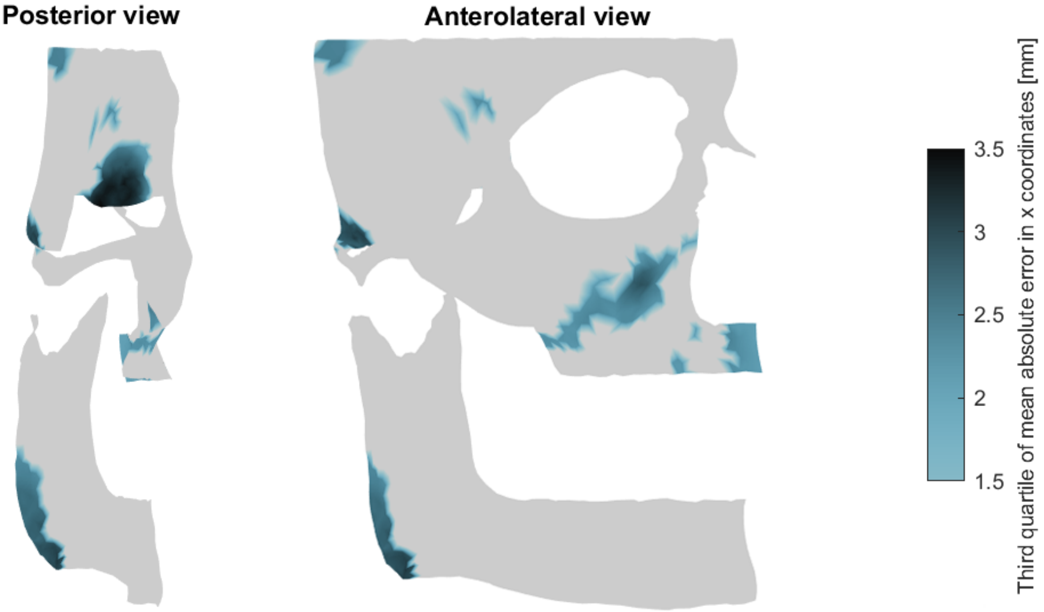


**Figure 4.12:** Distance map of mean absolute error above third quartile in the x coordinates of at least 10 test subjects.

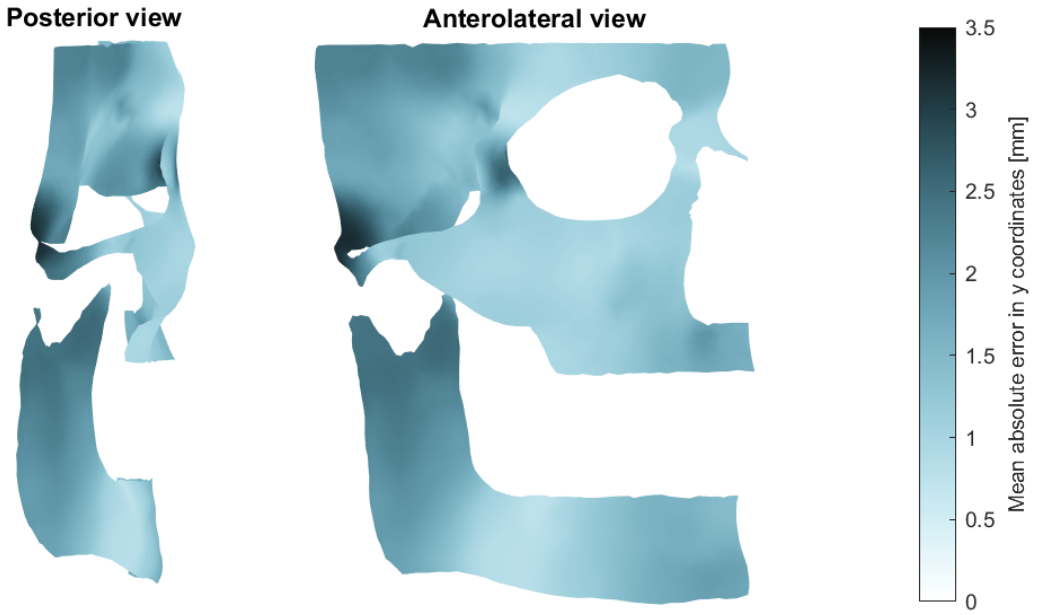


**Figure 4.13:** Distance map of the mean absolute error of all test subjects in the y coordinates.

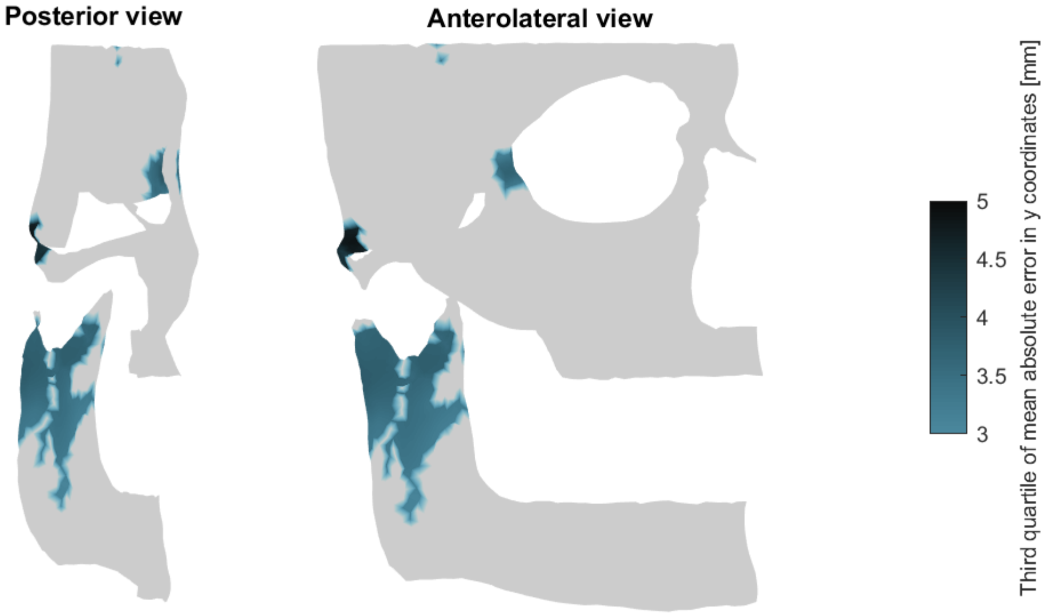


**Figure 4.14:** Distance map of mean absolute error above third quartile in the y coordinates of at least 10 test subjects.

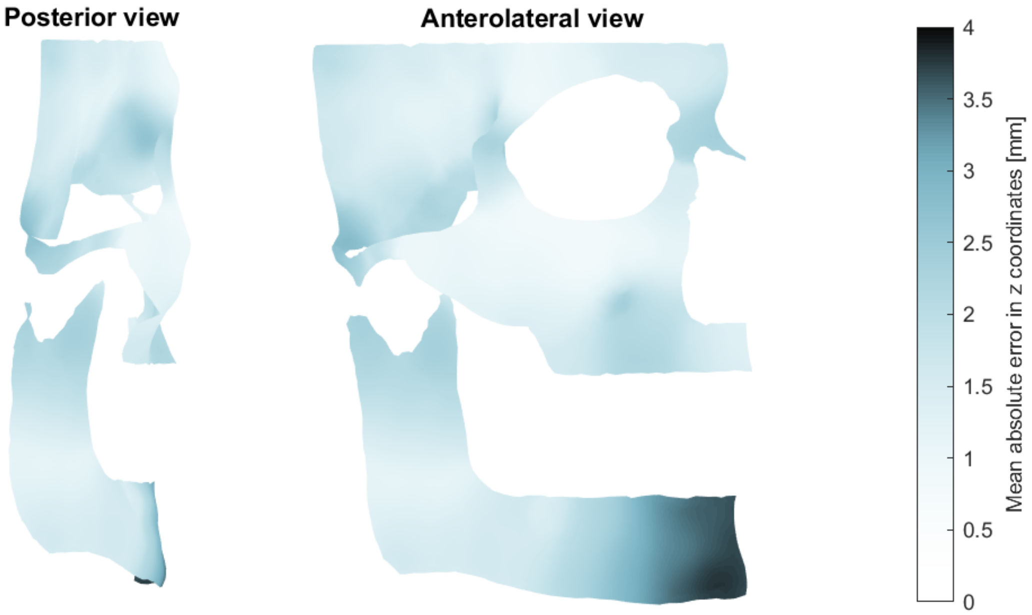


**Figure 4.15:** Distance map of the mean absolute error of all test subjects in the z coordinates.

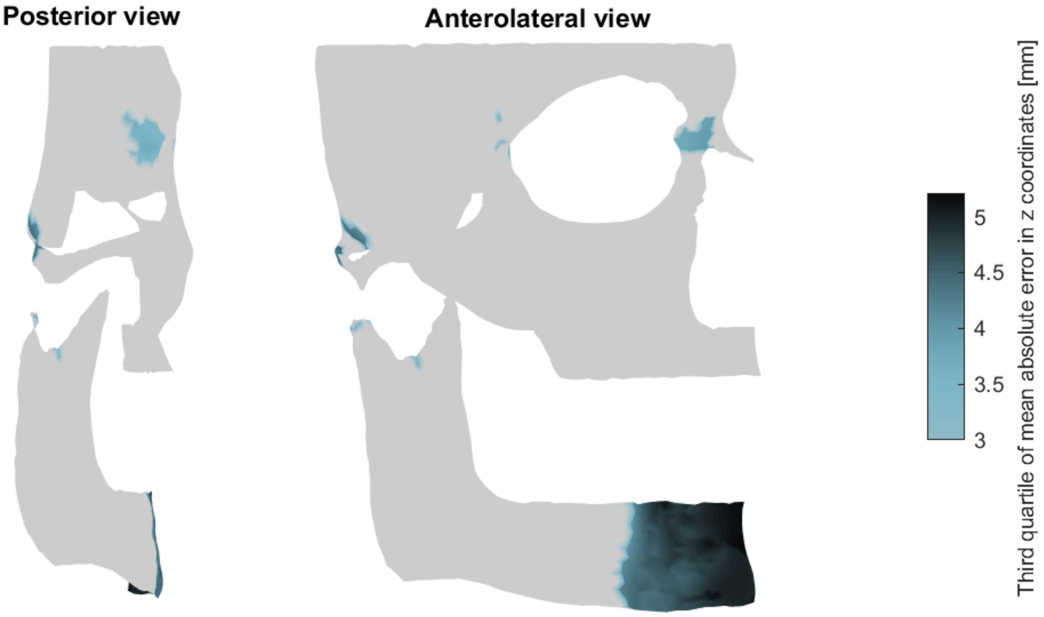
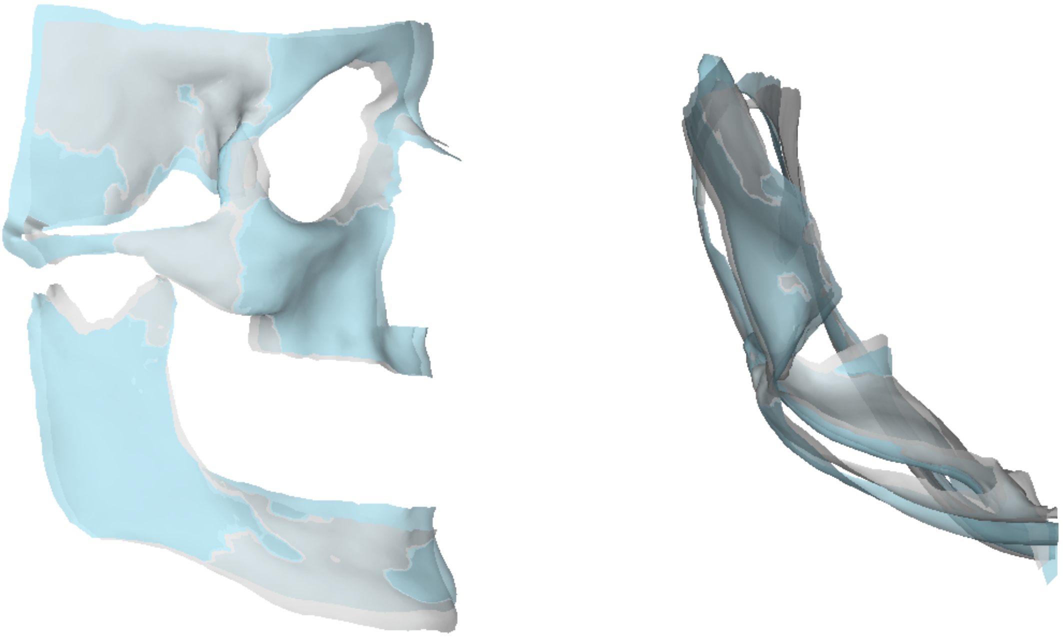


**Figure 4.16:** Distance map of mean absolute error above third quartile in the z coordinates of at least 10 test subjects.

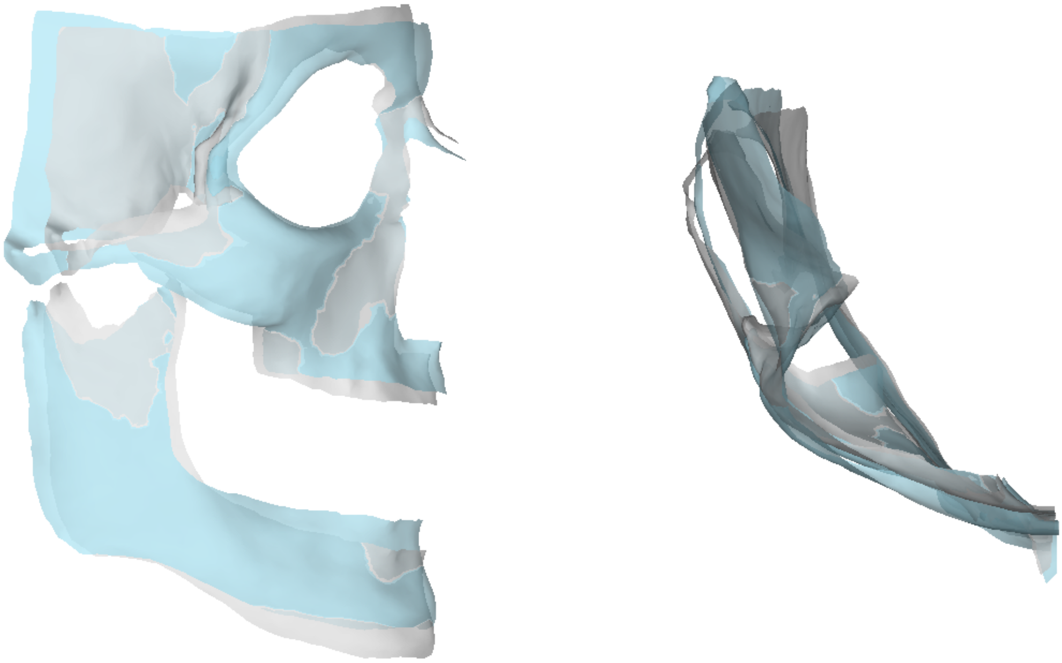
Figures 4.17 through 4.19 show the difference in shape between the reconstructed and true hard tissue meshes of three patients with average, high and low MAE values, respectively. Patient 1 had an average MAE value of 1.5 mm over all vertices and dimensions, patient 2 had the highest MAE value of 2.2 mm and patient 3 had the lowest MAE value of 0.8 mm.



(a) Lateral view

(b) Cranial view

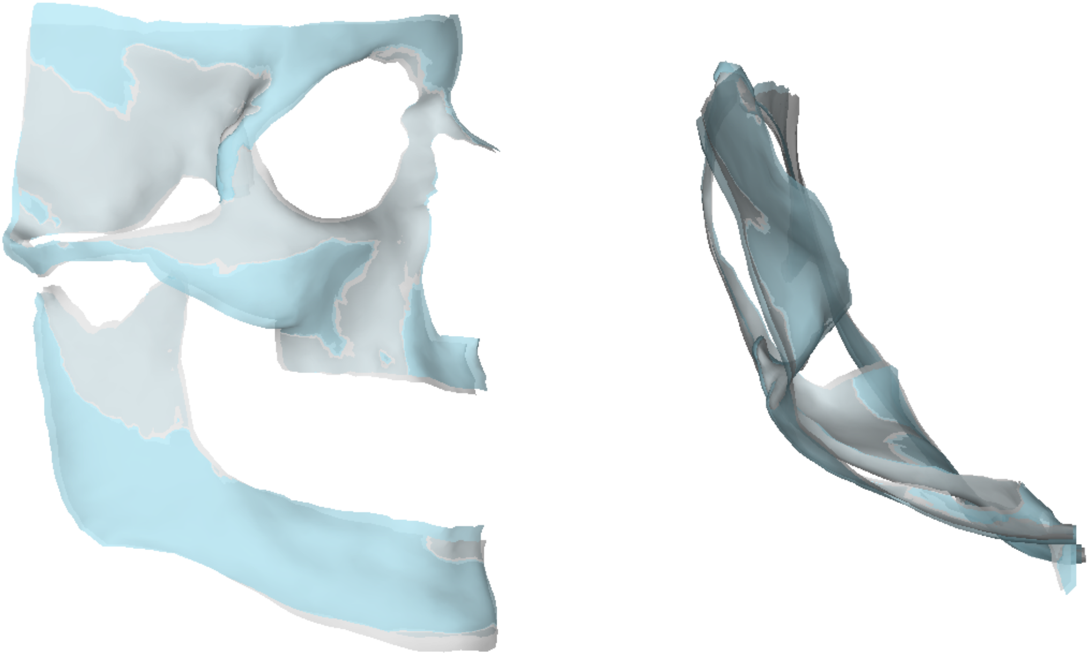
**Figure 4.17:** Overlay of reconstructed mesh in blue over true mesh in gray, of patient number 1 with average MAE values.



(a) Lateral view

(b) Cranial view

**Figure 4.18:** Overlay of reconstructed mesh in blue over true mesh in gray, of patient number 2 with highest MAE values.



(a) Lateral view

(b) Cranial view

**Figure 4.19:** Overlay of reconstructed mesh in blue over true mesh in gray, of patient number 3 with lowest MAE values.

Chapter 5

# Discussion

In this study a combination of PCA and autoencoder model was used to research a proof of concept for the possibility to predict facial hard tissue shape based on facial soft tissue shape. Accurate predictions of facial hard tissue shape would aim to assist OMFS surgeons in surgical planning of FFS.

## 5.1 Interpretation of the results

In this section, the results of the previous chapter will be reviewed.

### 5.1.1 Principal component analysis

In Table 4.1 the cumulative percentages of the variance explained by each principal component for both the soft and hard tissue PCA are shown. It is apparent that the variance of the soft tissue data is less complex and the same amount of variance can be represented by a lower amount of PCs. Where seventeen PCs are needed to explain 80 percent of the hard tissue data variance, only six are needed for the soft tissue data. It was chosen to include seventeen PCs of both groups, ST and HT, to provide symmetrical input and output for the autoencoder. However, the information represented by these principal components is asymmetrical, as there is more information enclosed in the soft tissue input. It is difficult to assess the exact implications of this asymmetry without a comparison to a model trained on six ST PCs and 17 HT PCs. Having more information in the input may make it easier for the model to train if the information is relevant, however if the information is not relevant for the prediction of the hard tissue PCs it will add noise to the data which will make training harder for the autoencoder.

Table 4.2 shows the MAE and ME values expressed as percentage of the SD of the predicted hard tissue PC scores. The MAE values are generally higher in the lower PCs, but as the SD of the scores is also higher in lower PCs, this is not unexpected. When comparing the MAE expressed as percentage of PC scores SD, the errors are comparable in the higher and lower PCs.

The ME values show a tendency of the autoencoder to underestimate the scores of PCs 1 and 3 through 6, and to overestimate PCs 2 and 7 through 17. The chosen loss function of mean squared logarithmic error could have been on influence on this, as this function punishes underestimates more harshly than overestimates. It is also important to note that the earlier PCs contain the most information and are therefore the most important to predict correctly. A symmetrical loss function, mean squared error for example, could correct the overestimates but may simultaneously worsen the underestimates in five of the first six PCs.

Figures 4.2 through 4.7 depict the areas and transformation of three SDs in mm. The component of the transformation in the direction of the vertex normals is shown as opposed to the transformation in the direction of the axes, as the components of the three coordinates cannot be calculated separately. The PCA is performed on all vertex coordinates at once, and the variance in the data is a combination of the three dimensions. These figures will be reviewed in section 5.1.3 alongside the reconstructed meshes.

### 5.1.2 Autoencoder

The loss plot in 4.8 shows a lower validation loss at the start of training than the training loss. As the number of epochs increases, the training loss converges towards the validation loss, indicating that the model is capable of learning on the provided data. As the input consists of soft tissue PC scores and the output of hard tissue PC scores, this shows that soft tissue shape has a predictive value for hard tissue shape. Due to the combination of PCA and an autoencoder, the exact relations between soft and hard tissue are difficult to assess. In the future, with a larger dataset and an improved MeshMonk algorithm, it should be possible to train a model on original data of full ST and HT meshes instead of on PC scores of half ST and HT meshes. This will provide the possibility to train on both linear and non-linear relations between the data, instead of being limited to the linear relations due to the linear dimensionality reduction of PCA.

### 5.1.3 Reconstruction

The histograms in figure 4.10 show a right-skewed distribution of the absolute errors. As the errors were not normally distributed, the median and median absolute deviation were calculated for each dimension. The errors in the x coordinates were significantly lower than in the y and z coordinates, with the z coordinates displaying the highest errors. As seen in 4.3, the median of the errors is lowest for x coordinates with 0.88 ± 0.56 mm, highest in y coordinates with 1.51 ± 0.87 mm and 1.35

± 0.88 for z coordinates.

As the z axis has the largest dispersion and the x axis the lowest due to half facial mehes being used, it seems logical that the errors would generally be higher in the z coordinates and lower in the x coordinates. More insightful is, however, to look at where in the meshes the higher errors occur.

The top 25% of errors in each coordinate dimensions occurred in specific areas in at least ten of the 21 test subjects. These areas of a total of 342 vertices for the x coordinates can be seen in Figure 4.12, and are located in the mandibular angle, temporal bone and fossa, zygomatic process of the temporal bone, alveolar and zygomatic process of the maxilla and the biggest errors in the sphenoid bone area. In Figure 4.2a and 4.2d, a transformation in the x coordinates in the maxilla and zygomatic process of temporal bone can be seen. With the MAE of the first PC representing a high percentage of the scores SD (89), this would have been of influence. In figures 4.3a and 4.3d, the influence of the second PC on the x coordinates of the mandibular angle can be seen. The ninth PC in figures 4.7a and 4.7d also has an influence on the mandibular angle, however as the variance explained by this PC is smaller, the influence will also be smaller. Figures 4.4a, 4.4d, 4.5a and 4.5d show that the variance of the sphenoid in the x direction is largely determined by the fourth and fifth PC. These PCs all have a large error percentage as displayed in Table 4.2.

For the y coordinates, 367 vertices had errors above the third quartile in ten or more subjects. Figure 4.14 shows these vertices grouped together in the zygomatic process of the temporal bone, sphenoid bone, frontal process of zygomatic bone and mandibular ramus. The mandibular ramus is influenced by the first, second and seventh PC in figures 4.1b, 4.1e, 4.2b, 4.2e, 4.6b and 4.6e. The zygomatic process of the temporal bone, the sphenoid bone and the frontal process of the zygomatic bone are all represented in the fourth PC in figures 4.4b and 4.4e. These last two areas are also determined by the fifth PC as seen in figures 4.5b and 4.5e. The zygomatic process of the temporal bone and the sphenoid bone are also represented by the seventh and ninth PC in figures 4.6b, 4.6e, 4.7b and 4.7e.

For the z coordinates in Figure 4.16, a large part of total of the 414 erroneous vertices is located in the chin, with smaller areas in the mandibula, zygomatic process of the temporal bone, sphenoid bone and frontal process of the maxilla. The first and second PC as depicted in figures 4.1c, 4.1f, 4.2c and 4.2f show variance in the chin region, with the frontal process of the maxilla also being represented by the first and fourth (figures 4.4c and 4.4f) PC. The zygomatic process of the temporal bone is influenced by multiple PCs; 1, 2, and 5 in figures 4.1c, 4.1f, 4.2c, 4.2f, 4.5c and 4.5f.

The overlay images of the reconstructed meshes of three patients in figures 4.17 through 4.19 shows how the errors in the vertex coordinates influence the overall hard tissue shape. While the reconstructed mesh in Figure 4.19 follows the true shape rather well, it is obvious that the reconstructed mesh in Figure 4.18 is not an accurate depiction of the actual shape.

## 5.2 Strengths and limitations

To the best of our knowledge, this is the first study in which the predictive value of soft tissue shape for hard tissue shape is explored. The reverse, hard to soft tissue shape, has been researched with main focus on predicting orthognathic surgery outcome. However, the combination of PCA with a dense autoencoder like in this study has not been proposed.

The main advantage of combining PCA with autoencoder is the possibility to extend the dimensionality in the autoencoder instead of reducing it. PCA ensures a low dimensional input which does contain the most important information, which allows an increase in dimensionality towards the latent space without compromising training time. By creating this higher dimensional latent space, non-linear complex relations between the input and output can be trained more effectively. Simultaneously, the PCA protects the process by omitting less relevant, or subject specific, information. This prevents the autoencoder from learning connections that may be true for the train set, but may not be true for the general population. This is especially useful for smaller data sets such as that used in this study.

Furthermore, the proposed combined method has promising feasibility. A trained autoencoder model can be used to create a prediction within minutes and does not require specific hardware or software to be purchased, and is easy to understand and use with little training needed. The part that will require the most time and effort is the data preparation steps, which requires each observation to be pre-processed in the exact same way, but this can be done efficiently with the use of scripts.

### 5.2.1 MeshMonk

MeshMonk was used in this study to create standard sized meshes with vertex correspondence between the meshes. The vertex correspondence allows for comparison of separate quasi-landmarks between multiple meshes and of face and skull shapes. A soft tissue template is provided by MeshMonk, which was used to create the soft tissue models based on 3D photo images. A hard tissue template was not available, and was created in this study by selecting a clean CBCT image from the data set and simplifying it towards a template. This template is based on a single patient's hard tissue shape, while the MeshMonk soft tissue template is an average of over 8000 facial scans. The validation of the MeshMonk toolbox with the soft tissue template over 19 landmarks on 41 facial images was described as a 1.26 mm average error. [27] The accuracy of MeshMonk in combination with the hard tissue template is not measured in this study. It is expected that due to the hard tissue template not being averaged over multiple scans, the performance of the template will be suboptimal when applied to scans with highly differing shapes from the template. This needs to be validated, however, and could be improved by averaging the template shape over as many scans as available in multiple iterations.

A comparison of the lateral view of the soft and hard tissue template in 5.1 provides a possible explanation for some areas displaying larger errors as stated in **??**.

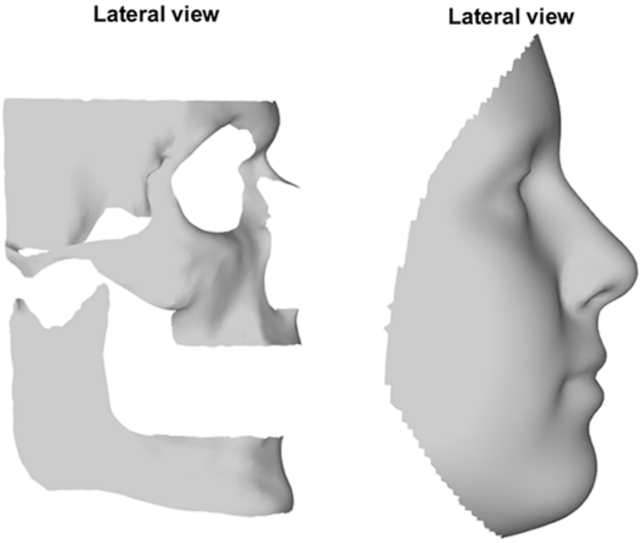


**Figure 5.1:** Lateral view of the hard and soft tissue templates.

The hard tissue areas of the mandibular angle, ramus and condyle as well as the posterior part of the zygomatic arch and the temporal bone and fossa are missing their overlying soft tissue areas in the template. Since the PCA extracts the most important information regarding general shape and variance in shape of the supplied subjects and the autoencoder was trained on these scores, the general shape of these areas could be produced. However, since there is some of the information of the soft tissue shape of these areas missing, this could explain the generally larger errors.

The zygomatic process of the temporal bone and temporal fossa that display larger errors in the x and y directions and the chin with errors in all dimensions are regions that often showed errors in MeshMonk. The template was not applied correctly to these areas in multiple subjects, resulting in incorrect correspondences being found. While this was known to happen in other areas of the skull as well, it was rarely in the exact same place and could be filtered out by PCA. It is expected that the chin, zygomatic process and temporal fossa had incorrect template application in too many subjects. This would be returned in the PCA scores matrices and would influence the training of the autoencoder trying to discern relation between soft and hard tissue scores. These faulty meshes were chosen to include as the importance of more data was deemed greater than the influence of these errors on the PCA. With a larger data set available, meshes being incorrectly processed by MeshMonk could be more strictly evaluated and removed to improve the quality of the data provided to PCA and the autoencoder.

### 5.2.2 Principal component analysis

The PCA was based on preoperative scans only, to reduce the influence of implants in the postoperative meshes on the analysis. The scores of the postoperative subjects was calculated with the coefficient matrix of the preoperative PCA. Ideally, a large number of healthy non-operated scans would be used to determine the variance of hard tissue shape in the general population. As CBCT images require radiation and medical indication to be made, nearly all available scans are of subjects who had an indication for OMFS.

### 5.2.3 Autoencoder

The autoencoder was evaluated using a 5-fold cross-validation procedure to ensure that its performance did not depend on the selected validation subset. The validation losses at the end of training had a mean of 0.12 with a standard deviation of 0.01, indicating that the autoencoder performed well in learning the relationship between the ST and HT scores. There was no analysis performed on the latent features generated by the autoencoder. The use of PCA scores as input results in a more abstract and difficult to interpret representation, but knowing what connections are learned by the autoencoder could provide more insight.

This dense autoencoder worked reasonably well with the data it was provided, but with more available data other autoencoder models may be more suited. The use of a variational autoencoder (VAE) with a probabilistic distribution in the latent space may be more effective to discern complex non-linear relationships than a dense autoencoder. The probabilistic nature would allow for modeling of uncertainty in the data that would give insight in which specific areas may have more complex relationships between soft and hard tissue shape.

### 5.2.4 Demographics

A large number of variables are of importance in facial morphology and the relation between soft and hard tissue shape. The most important variables are BMI, age, race and sex. Fat distribution, influenced by BMI and hormones, changes the facial shape and relation between soft and hard tissue shape. With increasing age, soft tissue firmness decreases and changes shape. [36] In this study, a trade off had to be made between data set size and homogeneity. All subjects were of Caucasian race, but age ranged from 18 to 55 years. BMI was not accurately documented and not all 3D photos were made according to the correct workflow. Good quality matching soft and hard tissue images of healthy individuals with comparable demographics are needed in a large number to be able to properly train models on the relation between tissue shapes.

## 5.3 Clinical relevance

Errors within the range of 2 mm are generally deemed acceptable [21], [37], [38], the MAE of 19 of the 21 reconstructed meshes fall within this range. However, it is important to note that the comparison of the vertex coordinates is done by calculating the MAE between the reconstructed vertices from autoencoder generated PCA scores with 17 PCs and the true vertex coordinates that were also reconstructed with 17 PCs. The true error to the full mesh cannot be calculated because of the use of PCA, and will need to be evaluated on a model which is trained on whole meshes without any limiting pre-processing such as PCA. It is however very likely that the true errors will be higher than represented in this paper.

At the time of writing, a perfectly working model that can accurately predict facial hard tissue shape based on soft tissue shape would not make an immediate impact on the workflow of FFS. It is important to realize that to be able to use such a model for preoperative planning and postoperative evaluation, a method to simulate desired postoperative soft tissue shape needs to be in place. This may very well be realised in the time it will take to further research the prediction of facial HT shape from facial ST shape, but it is an important aspect for the clinical usefulness of such a predictive model.

## Chapter 6

# Conclusion

This study provides a proof of concept for the use of principal component analysis in combination with a dense autoencoder to predict facial hard tissue shape based on soft tissue shape. The autoencoder mean squared logarithmic error loss approximated a value of 0.1 after training and median errors below 1.5 mm were found in each vertex dimension of the reconstructed hard tissue meshes, which proves the predictive value of facial soft tissue shape. The combination of multiple processing steps makes it hard to pinpoint the origins of errors, and we suggest each step should be individually evaluated on accuracy to be able to understand and improve the process further. Along with more data of subjects with comparable demographics and a stricter exclusion on image quality, the results should improve and the true possibilities for objective FFS methods should become clear.

# Bibliography

[1] Wiepjes CM, Nota NM, de Blok CJM, Klaver M, de Vries ALC, et al, “The Amsterdam Cohort of Gender Dysphoria Study (1972–2015): Trends in Prevalence, Treatment, and Regrets,” *The Journal of Sexual Medicine*, vol. 15, no. 4, pp. 582–590, 04 2018.

[2] Wood H, Sasaki S, Bradley SJ, Singh D, Fantus S, et al, “Patterns of Referral to a Gender Identity Service for Children and Adolescents (1976–2011): Age, Sex Ratio, and Sexual Orientation,” *Journal of Sex Marital Therapy*, vol. 39, pp. 1–6, 2013.

[3] Chen M, Fuqua J, Eugster EA, “Characteristics of Referrals for Gender Dysphoria Over a 13-Year Period,” *Journal of Adolescent Health*, vol. 58, no. 3, pp. 369–371, 03 2016.

[4] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition*, 2013.

[5] Coleman E, Bockting W, Botzer M, Cohen-Kettenis P, DeCuypere G, et al, *Standards of Care for the Health of Transsexual, Transgender, and Gender Nonconforming People*. World Professional Association for Transgender Health, 7th Version, 2012.

[6] Ginsberg BA, Calderon M, Seminara NM, Day D, “A potential role for the dermatologist in the physical transformation of transgender people: A survey of attitudes and practices within the transgender community,” *Journal of the American Academy of Dermatology*, vol. 74, no. 2, pp. 303–308, 02 2016.

[7] Juszczak HM, Fridirici Z, Knott PD, Park AM, Seth R, “An update in facial gender confirming surgery,” *Current Opinion in Otolaryngology Head and Neck Surgery*, vol. 27, no. 4, pp. 243–252, 08 2019.

[8] Capitán L, Gutiérrez Santamaría J, Simon D, Coon D, Bailón C, et al, “Facial Gender Confirmation Surgery: A Protocol for Diagnosis, Surgical Planning, and Postoperative Management,” *Plastic and Reconstructive Surgery*, vol. 145, no. 4, pp. 818–828, 04 2020.

[9] Eyssel J, Koehler A, Dekker A, Sehner S, Nieder TO, “Needs and concerns of transgender individuals regarding interdisciplinary transgender healthcare: A non-clinical online survey,” *PLoS ONE*, vol. 12, no. 8, 08 2017.

[10] Ching AH, Hirschman A, Lu X, Maniskas S, Forte AJ et al, “Perception of femininity and attractiveness in Facial Feminization Surgery,” *Annals of Translational Medicine*, vol. 9, no. 7, p. 602, 08 2021.

[11] Ainsworth TA, Spiegel JH, “Quality of life of individuals with and without facial feminization surgery or gender reassignment surgery,” *Quality of Life Research*, vol. 19, no. 7, pp. 1019–1024, 05 2010.

[12] Deschamps-Braly JC, “Facial Gender Confirmation Surgery,” *Clinics in Plastic Surgery*, vol. 45, no. 3, pp. 323–331, 2018.

[13] Blank NR, Giordano CN, Dusza S, Nehal K, Lee EH, “Patient Expectations Influence Postoperative Facial Satisfaction Measured by the FACE-Q Skin Cancer Module: A Pilot Study,” *Dermatologic surgery*, vol. 46, no. 8, p. 1113, 2020.

[14] Cadogan J, Bennun I, “Face value: an exploration of the psychological impact of orthognathic surgery,” *British Journal of Oral and Maxillofacial Surgery*, vol. 49, no. 5, pp. 376–380, 2011.

[15] Bockting W, Robinson B, Benner A, Scheltema K, “Patient satisfaction with transgender health services,” *Journal of sex and marital therapy*, vol. 30, no. 4, pp. 277–294, 2004.

[16] Hager GD, Maier-Hein L, Swaroop Vedula S, *Handbook of Medical Image Computing and Computer Assisted Intervention*.

[17] Olivetti EC, Nicotera S, Marcolin F, Vezzetti E, Sotong JPA, Zavattero E, et al, “3D Soft-Tissue Prediction Methodologies for Orthognathic Surgery—A Literature Review,” *Applied Sciences*, vol. 9, no. 21, p. 4550, 10 2019.

[18] Gupta RJ, Schendel S, *Soft Tissue Changes and Prediction with Orthognathic Surgery*.

[19] Naveed N, Sabapathy K, “An overview of the methods of prediction planning for orthognathic surgery using cephalometrics,” *European Journal of Molecular Clinical Medicine*, vol. 7, no. 4, p. 2020, 2020.

[20] Mollemans W, Schutyser F, Nadjmi N, Maes F, Suetens P, “Predicting soft tissue deformations for a maxillofacial surgery planning system: From computational strategies to a complete clinical validation,” *Medical Image Analysis*, vol. 11, no. 3, pp. 282–301, 2007.

[21] Knoops PGM, Borghi A, Breakey RWF, Ong J, Jeelani NUO, Bruun R, et al, “Three-dimensional soft tissue prediction in orthognathic surgery: a clinical comparison of Dolphin, ProPlan CMF, and probabilistic finite element modelling,” *International Journal of Oral and Maxillofacial Surgery*, vol. 48, no. 4, pp. 511–518, 2019.

[22] Kim D, Kuang T, Rodrigues YL, Gateno J, Shen SGF, Wang X, et al, “A novel incremental simulation of facial changes following orthognathic surgery using FEM with realistic lip sliding effect,” *Medical Image Analysis*, vol. 72, pp. 511–518, 08 2021.

[23] Knoops PGM, Papaioannou A, Borghi A, Breakey RWF, Wilson AT, et al, “A machine learning framework for automated diagnosis and computer-assisted planning in plastic and reconstructive surgery,” *Scientific reports*, vol. 9, no. 1, pp. 1–12, 08 2019.

[24] ter Horst R, van Weert H, Loonen T, Bergé S, Vinayahalingam S, et al, “Three-dimensional virtual planning in mandibular advancement surgery: Soft tissue prediction based on deep learning,” *Journal of Cranio-Maxillofacial Surgery*, vol. 49, no. 9, pp. 775–782, 2021.

[25] Elshebiny T, Morcos S, Mohammad A, Quereshy F, Valiathan M, “Accuracy of Three-Dimensional Soft Tissue Prediction in Orthognathic Cases Using Dolphin Three-Dimensional Software,” *Journal of Craniofacial Surgery*, vol. 30, no. 2, pp. 525–528, 04 2019.

[26] Liebregts J, Xi T, Timmermans M, de Koning M, Bergé S, Hoppenreijs T, Maal T, “Accuracy of three-dimensional soft tissue simulation in bimaxillary osteotomies,” *Journal of Cranio-Maxillo-Facial Surgery*, vol. 43, pp. 329–335, 2015.

[27] White JD, Ortega-Castrillón AO, Matthews H, Zaidi AA, Ekrami O, Snyders J, et al, “MeshMonk: Open-source large-scale intensive 3D phenotyping,” *Scientific Reports*, vol. 9, no. 6085, 04 2019.

[28] Webmonks. The webmonks meshmonk github repository, retrieved from https://github.com/thewebmonks/meshmonk on 16.02.2023.

[29] Kingma DP, Ba J, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[30] Ioffe S, Szegedy C, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *Proceedings of Machine Learning Re-*

*search*, vol. International conference on machine learning, pp. 448–456, 06 2015.

[31] Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 07 2012.

[32] Abdi H, Williams LJ, “Principal component analysis,” *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[33] Wold S, Esbensen K, Geladi P, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[34] Jollife I, Everitt BS, Howell, DC, *Principal component analysis. Encyclopedia of statistics in behavioral science*, 2005.

[35] Smith LI, *A tutorial on Principal Components Analysis*, 2002.

[36] Zhuang Z, Landsittel D, Benson S, Roberge R, Shaffer R, “Facial Anthropometric Differences among Gender, Ethnicity, and Age Groups,” *The Annals of Occupational Hygiene*, vol. 54, no. 4, p. 391–402, 2010.

[37] Mundluru T, Almukhtar A, Ju X, Ayoub A, “The accuracy of three-dimensional prediction of soft tissue changes following the surgical correction of facial asymmetry: An innovative concept,” *International Journal of Oral and Maxillofacial Surgery*, vol. 46, no. 11, pp. 1517–24, 2017.

[38] Cunha HS, da Costa Moraes CA, de Faria Valle Dornelles R, da Rosa EL, “Accuracy of three-dimensional virtual simulation of the soft tissues of the face in OrtogOnBlender for correction of class II dentofacial deformities: an uncontrolled experimental case-series study,” *Oral and Maxillofacial Surgery*, vol. 25, pp. 319–35, 2021.

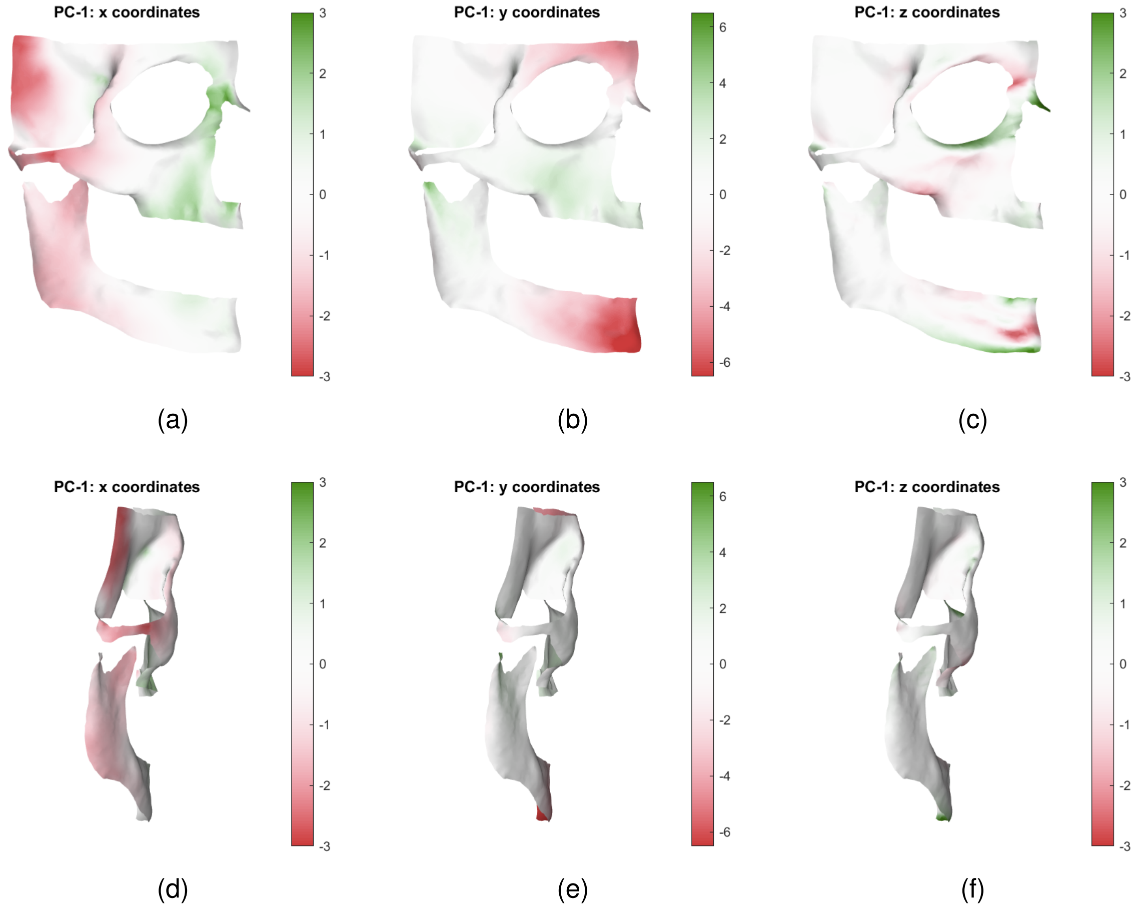
# Appendix A

# Appendix

The principal component visualizations and the script used to create the autoencoder model are included in this appendix.

## A.1 PC visualizations

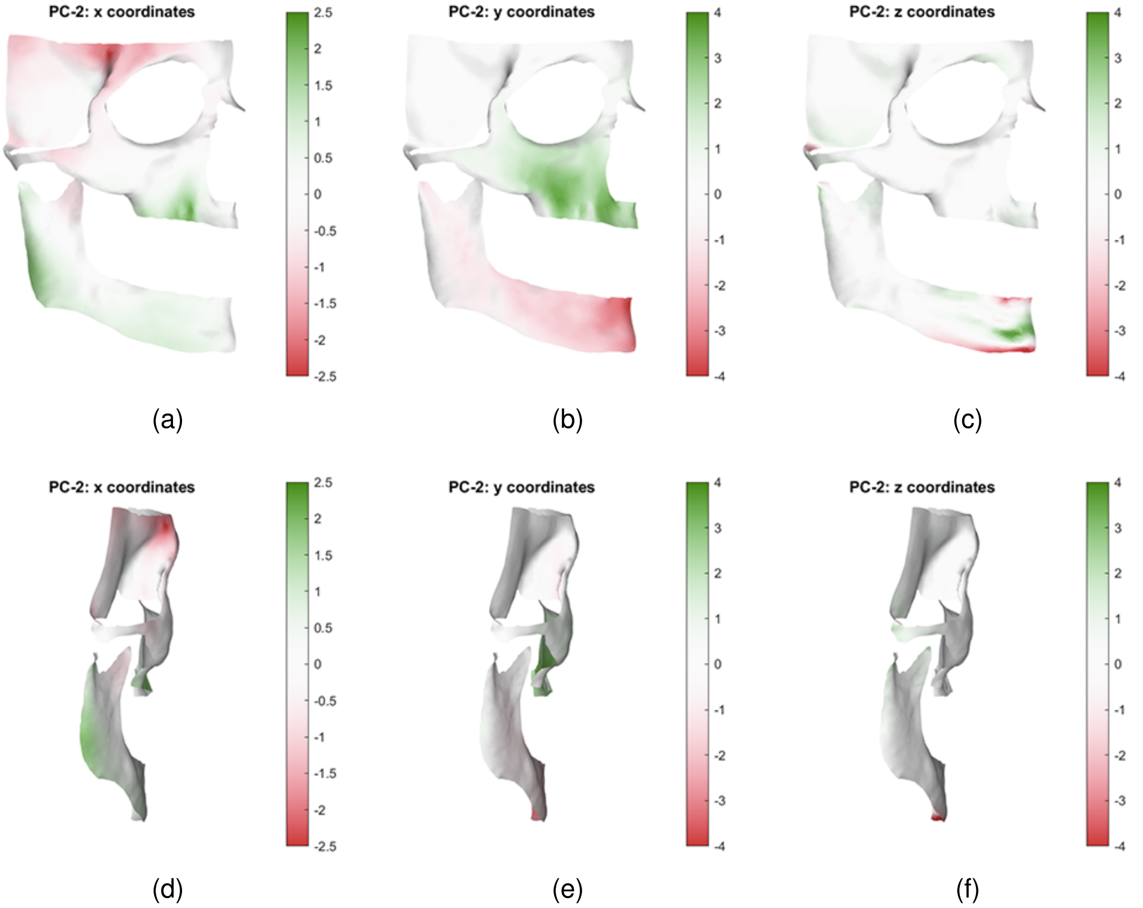
Figures A.1 through A.12 show the three standard deviation sized transformation of the x, y and z coordinates in PCs 1 through 12, respectively.



(a) (b) (c)

(d) (e) (f)

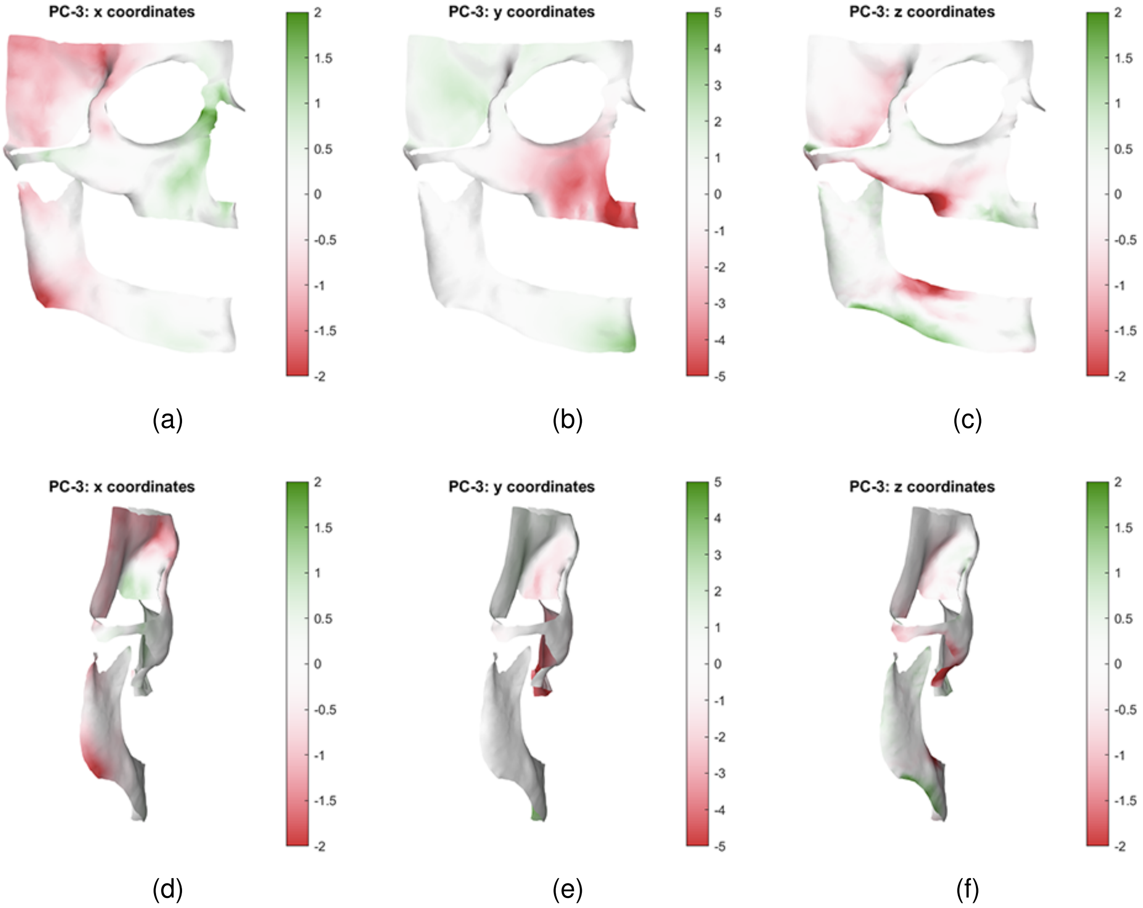
**Figure A.1:** Transformation of x, y and z in PC 1 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

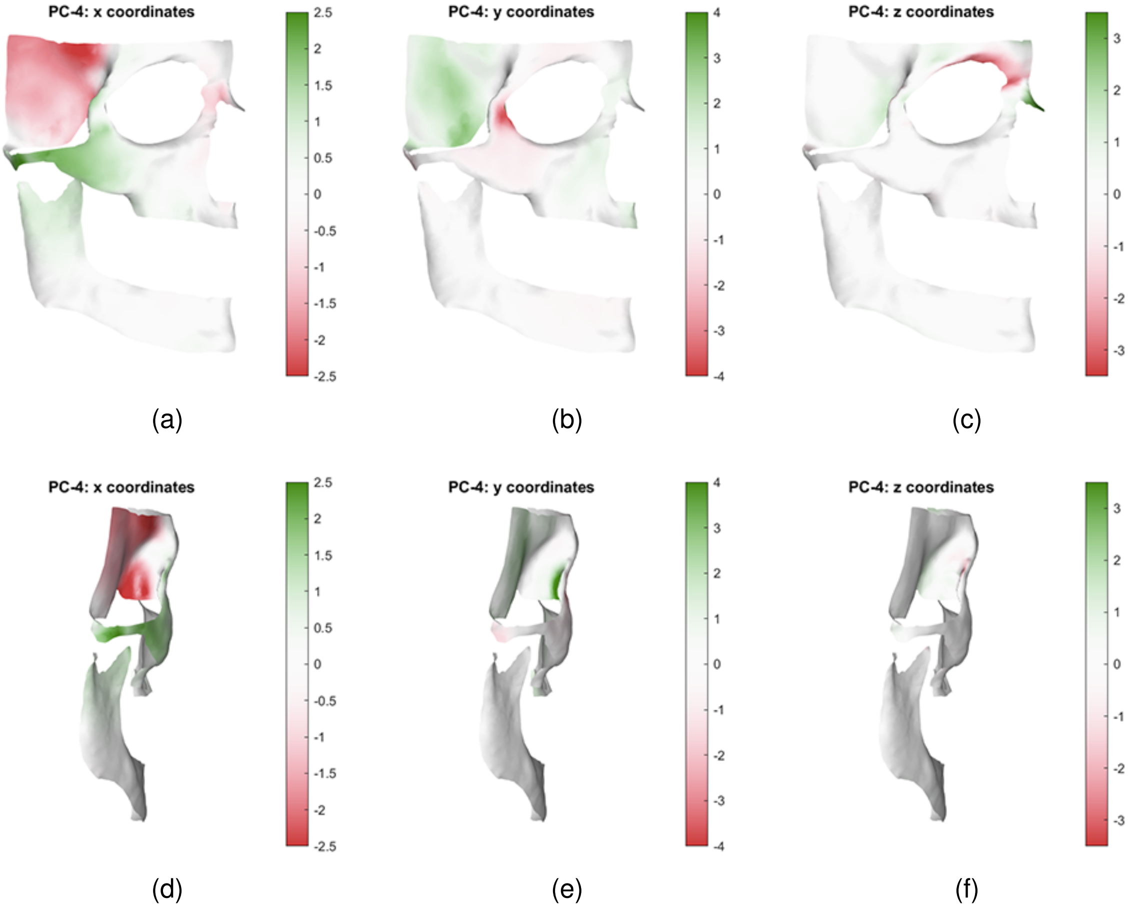
**Figure A.2:** Transformation of x, y and z in PC 2 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

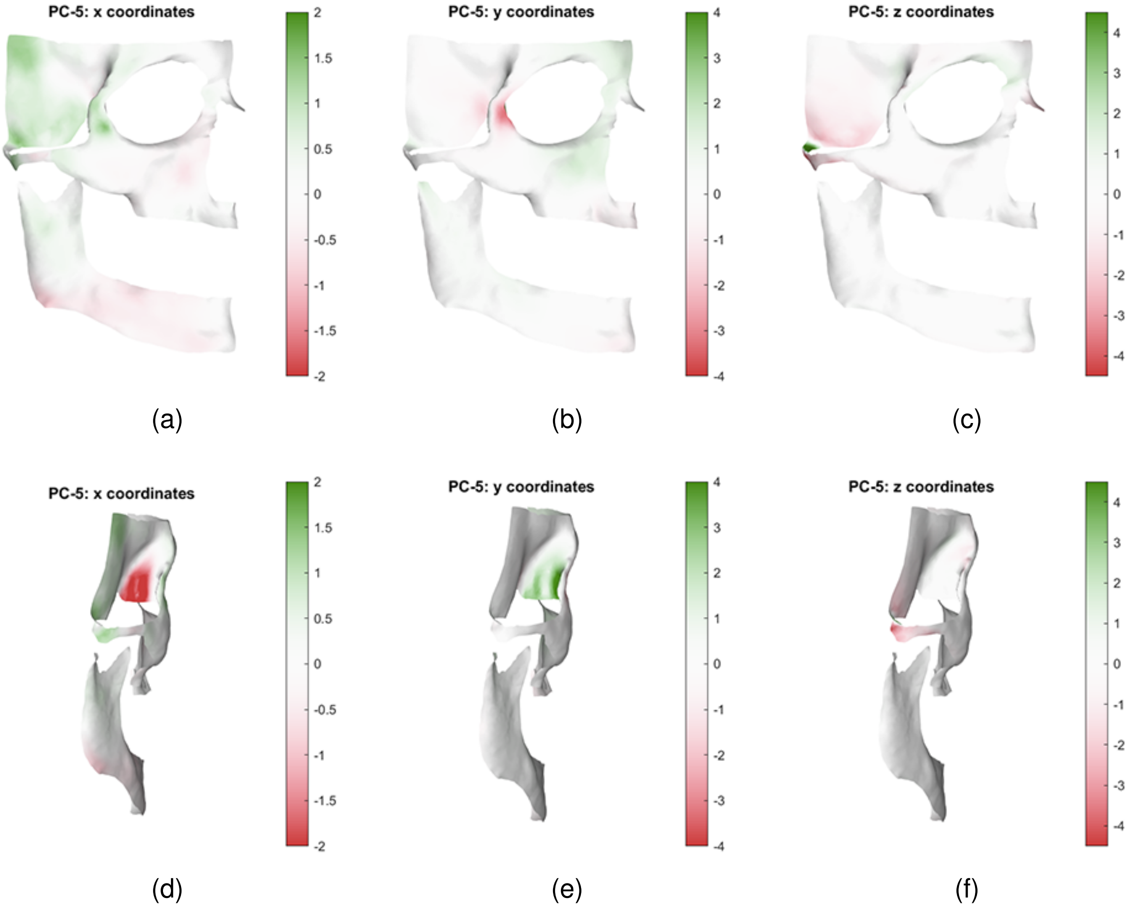
**Figure A.3:** Transformation of x, y and z in PC 3 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

**Figure A.4:** Transformation of x, y and z in PC 4 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

**Figure A.5:** Transformation of x, y and z in PC 5 in the direction of the vertex normals.

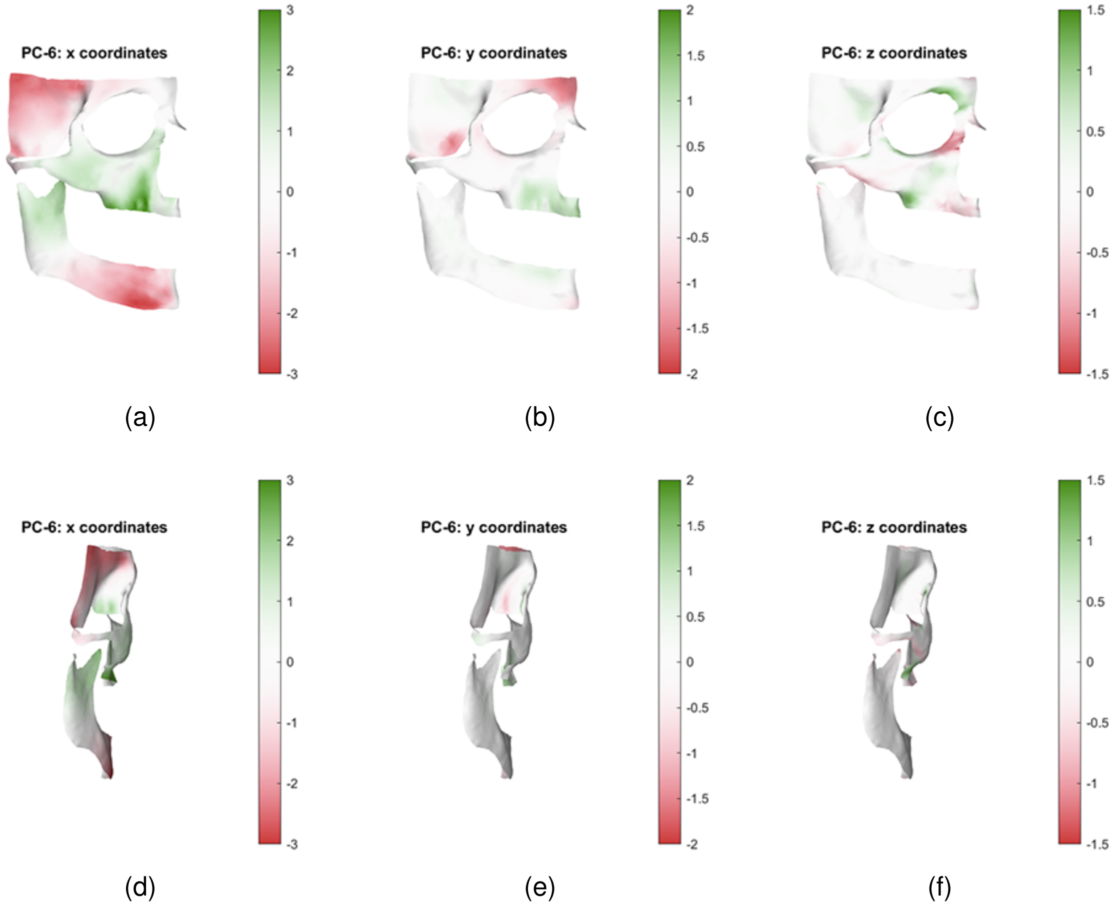


**Figure A.6:** Transformation of x, y and z in PC 6 in the direction of the vertex normals.

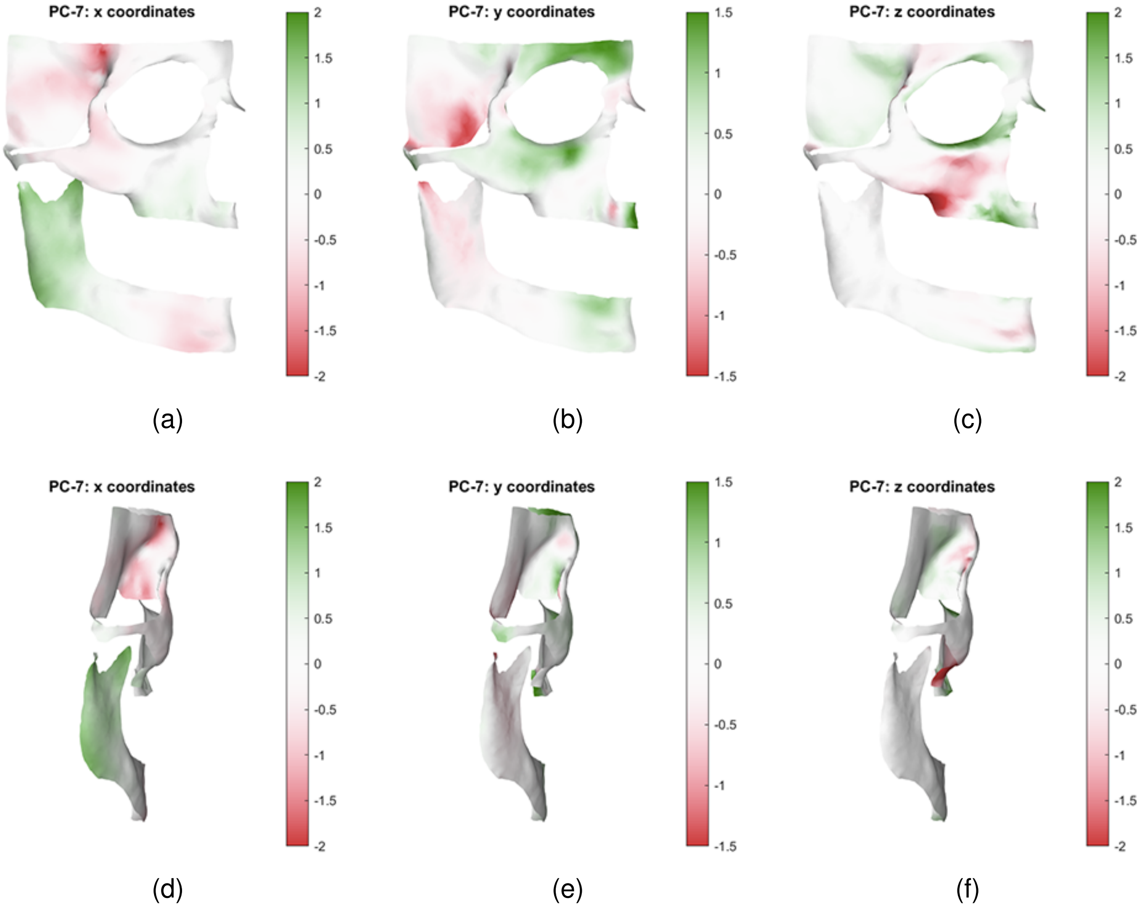


**Figure A.7:** Transformation of x, y and z in PC 7 in the direction of the vertex normals.

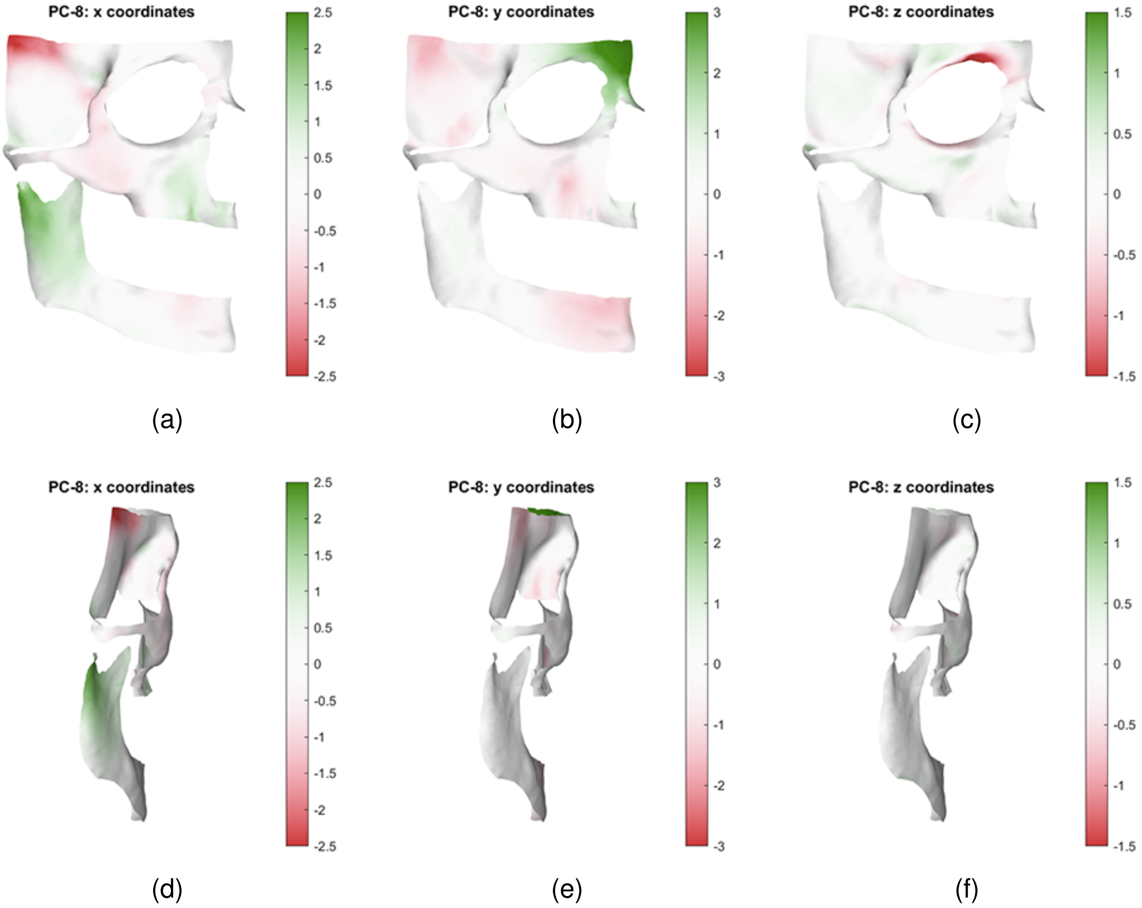


**Figure A.8:** Transformation of x, y and z in PC 8 in the direction of the vertex normals.

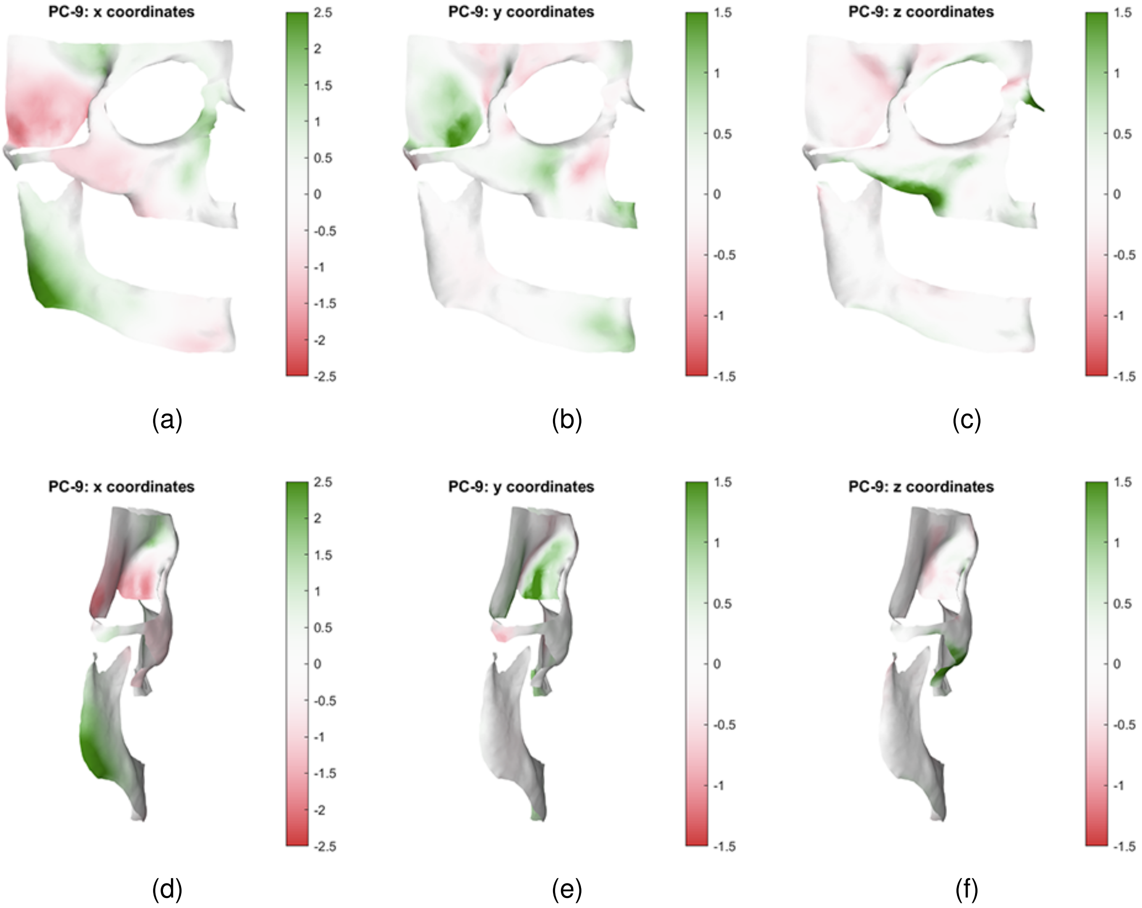
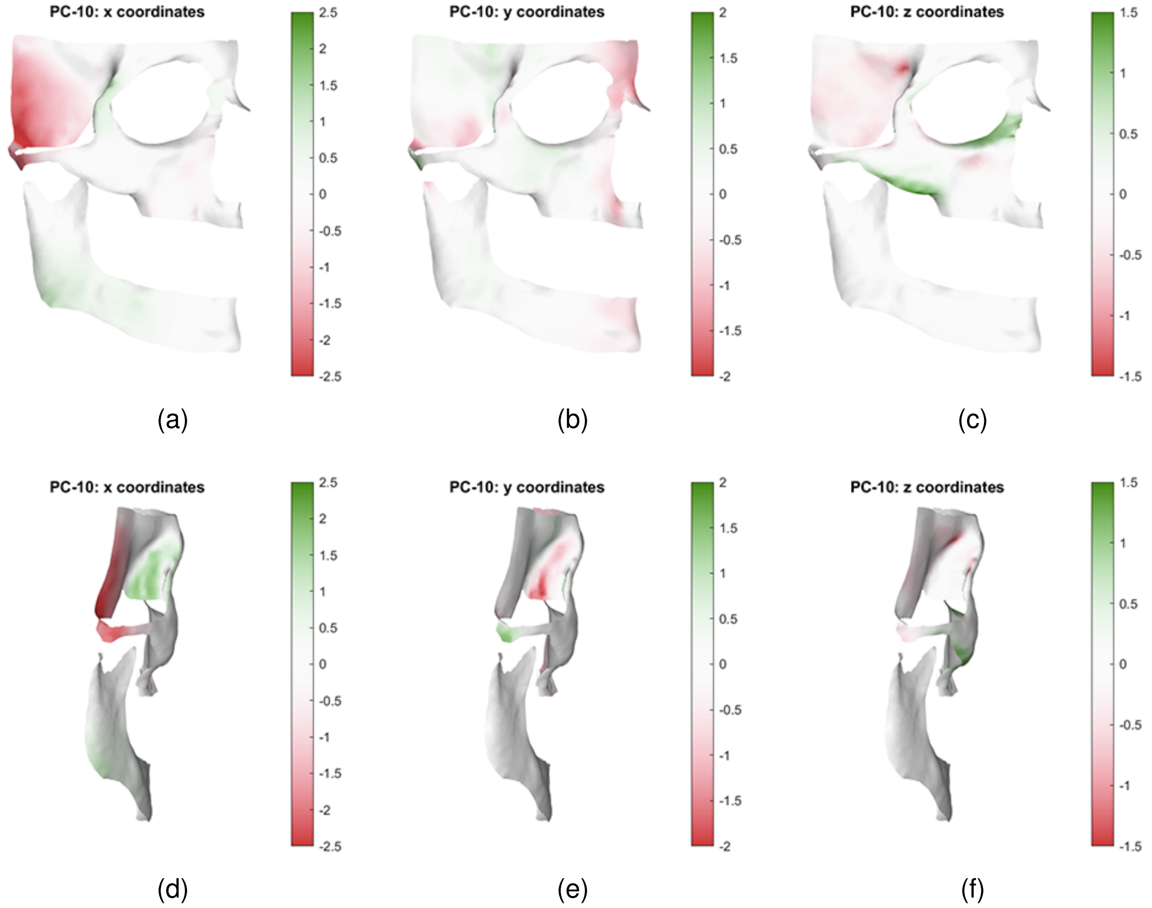


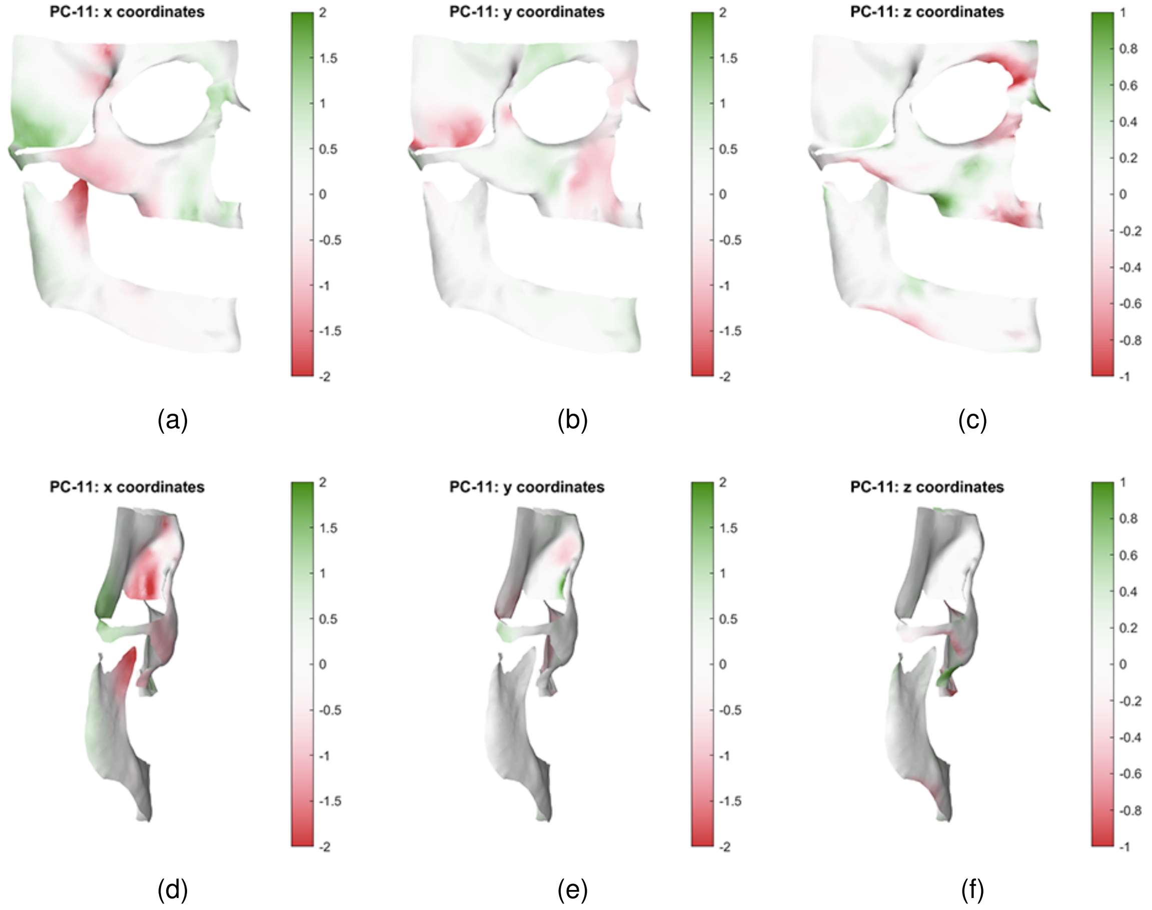
**Figure A.9:** Transformation of x, y and z in PC 9 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

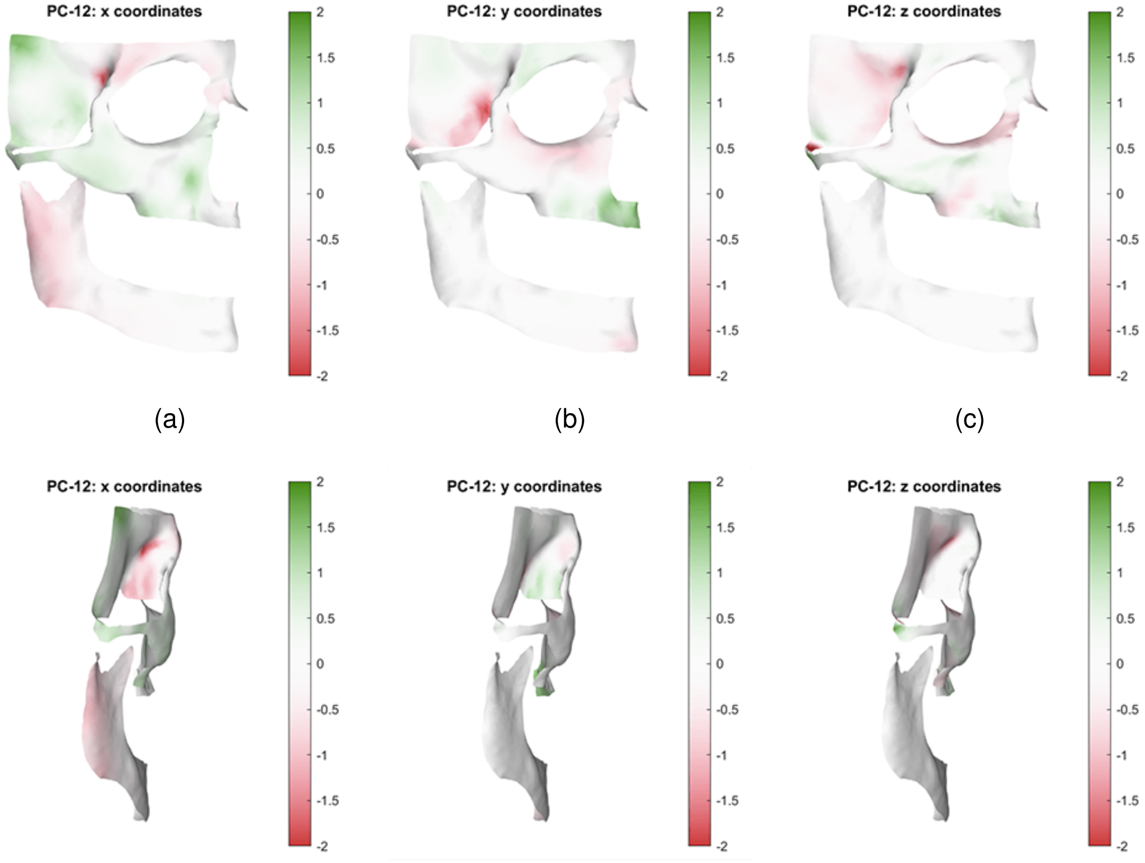
**Figure A.10:** Transformation of x, y and z in PC 10 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

**Figure A.11:** Transformation of x, y and z in PC 11 in the direction of the vertex normals.



(a) (b) (c)

(d) (e) (f)

**Figure A.12:** Transformation of x, y and z in PC 12 in the direction of the vertex normals.

## A.2 Autoencoder script

```
# import needed libraries and functions
import keras
from keras import layers
import tensorflow as tf
import tensorflow.keras as keras
from tensorflow.keras.layers import Dense, Input, Flatten,\
                                    Reshape, LeakyReLU as LR,\
                                    Activation, Dropout
from tensorflow.keras.models import Model, Sequential
from matplotlib import pyplot as plt

from sklearn.model_selection import KFold, StratifiedKFold
import numpy as np
import matplotlib.pyplot as plt
import scipy.io

# Load hard and soft tissue scores matrices from .mat files
scores_skull = scipy.io.loadmat(r'D:\M3 data\Final\PCA\Scores_Skull_samen_norm.mat',
↪  struct_as_record=False)
scores_skull = scores_skull['N']
scores_softtissue = scipy.io.loadmat(r'D:\M3
↪  data\Final\PCA\Scores_ST_samen_norm.mat', struct_as_record=False)
scores_softtissue = scores_softtissue['N']

# Use only the first 17 principal components, columns 1 through 17
scores_skull = scores_skull[:,0:17]
scores_softtissue = scores_softtissue[:,0:17]

# Divide into train, validation and test subsets
trainInput = scores_softtissue[0:200,:]     # train on soft tissue input of 200
↪  subjects
valInput = scores_softtissue[200:231,:]     # validate on 31 subjects

trainDesOutput = scores_skull[0:200,:]      # train on desired output of hard tissue
↪  scores of 200 subjects
valDesOutput = scores_skull[200:231,:]

testInput = scores_softtissue[231:252,:]    # test input of subject numbers 231
↪  through 252

# Define architecture of the autoencoder

input = keras.Input(shape=(17,))
encoded = layers.Dense(51, activation='relu')(input,)
encoded = layers.BatchNormalization(momentum=0.99)(encoded)
encoded = layers.Dropout(rate=0.1)(encoded)
encoded = layers.Dense(51, activation='relu')(encoded)
encoded = layers.GaussianNoise(1)(encoded)
encoded = layers.BatchNormalization(momentum=0.99)(encoded)
encoded = layers.Dense(68, activation='relu')(encoded)

decoded = layers.Dense(68, activation='relu')(encoded)
decoded = layers.BatchNormalization(momentum=0.99)(decoded)
decoded = layers.Dense(51, activation='relu')(decoded)
decoded = layers.BatchNormalization(momentum=0.99)(decoded)
decoded = layers.Dense(17, activation='linear')(decoded)

# Create the autoencoder
autoencoder = keras.Model(input, decoded)
callback = keras.callbacks.EarlyStopping(monitor='loss', patience=40)

# Define the optimizer
step = tf.Variable(0, trainable = False)
boundaries = [399, 699]
values = [0.001, 0.001*0.2, 0.001*0.1]
learning_rate_fn = keras.optimizers.schedules.PiecewiseConstantDecay(boundaries,
↪  values)
learning_rate = learning_rate_fn(step)

opt = keras.optimizers.Adam(learning_rate= learning_rate, beta_1=0.9)
autoencoder.compile(optimizer=opt, loss = 'mean_squared_logarithmic_error')

# Load column weights from .mat file
weights = scipy.io.loadmat(r'D:\M3 data\Final\PCA\pctExplained_Skull_preop.mat',
↪  struct_as_record=False)
weights = weights['pctExplained']

# Define weights per column
column_weight = {0: weights[0], 1: weights[1], 2: weights[2], 3: weights[3],
↪  4:weights[4], 5: weights[5], 6: weights[6], 7: weights[7], 8: weights[8], 9:
↪  weights[9], 10: weights[10], 11: weights[11], 12: weights[12], 13: weights[13],
↪  14: weights[14], 15: weights[15], 16: weights[16], 17: weights[17]} #18:
↪  weights[19], 19: weights[20], 20: weights[21], 21: weights[22]

# Set number of epochs for training and train the autoencoder
n_epochs = 2000
autoencoder_train = autoencoder.fit(trainInput, trainDesOutput, epochs = n_epochs,
↪  batch_size = 16, class_weight = column_weight, shuffle = True,
↪  validation_data=(valInput,valDesOutput), callbacks = [callback], verbose = 1)

# Track and plot the training and validation loss
loss = autoencoder_train.history['loss']
```

```
val_loss = autoencoder_train.history['val_loss']

plt.figure()
plt.plot(loss, "b", label='Training loss')
plt.plot(val_loss, "r", label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.savefig('loss.png')
plt.show()

val_loss = {'val_loss':val_loss}
loss = {'val_loss':loss}
# Use autoencoder to predict outcome of test set
decoded = autoencoder.predict(testInput)
decoded = {'decoded':decoded}

# Save prediction to .mat file
scipy.io.savemat("decoded.mat",decoded)
scipy.io.savemat("val_loss.mat",val_loss)
scipy.io.savemat("loss.mat",loss)
```