Technical University of Denmark

Master Thesis

Automated smile analysis using fast 4D facial surface sequence

Xiaohan Lyu - s182354

supervised by:
Vedrana Andersen Dahl
Tron Andre Darvann
Nuno Vibe Hermann

March 26, 2021

Approval

This thesis has been prepared over six months at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark in the collaboration with the 3D Craniofacial Image Research Laboratory. This thesis fulfils the requirement for the degree Master of Electrical Engineering.

Xiaohan Lyu - s182354

.....

Signature

.....

Date

Abstract

The face of another human being is what you see when you meet him or her. A lot of important communication and interaction between people takes place by decoding of the facial appearance and facial expression. Natural variation of facial shape and expression defines what a normal face is. In addition, social norms dictate what an acceptable facial appearance and facial expression is. Children born with an abnormal face or inability to properly move the face into different expressions have often problems in communication with other people and may be teased in school etc. One of the goals of the treatment of e.g. individuals with cleft lip and palate or facial paralysis is to restore the appearance and movement of the face. Qualitative check of treatment progress can be done by looking at the patient or photos or videos of the patient. Better quantitative measurement of progress can be done in 3D scans taken at the time of treatment and again after some treatment. Nowadays 3D surface scanners (3D photo) are available and used by clinicians to compare "snapshots" of the face, e.g. at a particular facial expression (neutral is usually used). However, in some cases, it would be better to evaluate the fast change in expression. Is the child able to smile normally? Is the smile symmetric? In order to measure this, it is possible to use a 3D scanner that is fast enough to obtain surfaces at a rate of 60 frames per second. Such a system is called a 4D surface scanner. This thesis investigates possible ways to handle the large amounts of data coming from such a system (a 3dMD 4D stereophotogrammetric system located at the 3D Craniofacial Image Research Laboratory, School of Dentistry, University of Copenhagen). As the smile is an important expression, we have focused on smile analysis. Having a quick, reliable smile analysis method could make the clinical treatment easier. Important tasks that we deal with are:

- Selection of surfaces from a smile sequence
- Automatic anatomical landmark point identification
- Estimation of errors and performance of the automatic method
- Analysis of single faces in terms of shape and asymmetry
- Analysis of smile sequences

The thesis presents a method that allows a fully automatic smile analysis that by use of smile trajectories in PCA space can characterize and quantify a smile.

Acknowledgement

I would like to take this opportunity to express my gratefulness to my supervisor Vedrana Andersen Dahl, Tron Darvann and Nuno Vibe Hermann. Thanks for their engagement and guidance in the project. This project is my first project working on 3D image analysis. Their patience has greatly encouraged me.

Besides, I would also like to thanks the volunteer for providing the dynamic smile sequence. In this special period, they provide the data with me so that I can continue this project.

Lastly I would like to express my deepest gratitude for my family, for the unconditional support, for allowing me to pursue my dream indefinitely, also for the company and motivation when I am low in energy. I am forever indebted.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and motivation

The facial expression analysis is essential due to the effect of facial analysis has on medicine that is beyond image analysis itself. A powerful facial expression analysis method provides clinicians with more helpful in the treatment of injuries and conditions affecting a face. The application of surface scanner has developed the 3D face analysis as an area of interest for clinical researchers and companies. In the last decade, there has been an increasing number of researches works on 3D face analysis.

This master thesis is motivated by the demand from clinicans working with the 3D Craniofacial Image Research Laboratory at the University of Copenhagen. The 3D Craniofacial Image Research Laboratory cooperates with the school of dentistry at University of Copenhagen, Copenhagen university hospital Rigshospitalet and Technical University of Denmark. The laboratory is committed to surveying the medical cases of craniofacial illness. For example, the children with congenital craniofacial malformations such as syndromes and cleft lip and palates, as well as the craniofacial anomalous children and adolescents that have acquired a trauma. Craniofacial malformation often affects the face and sometimes the facial expression. So the analysis of facial expression is helpful for the treatment of craniofacial illness. A most common method for analyzing the facial expression has been based on a static image or a single 3D model. But the static face cannot convey a facial expression completely. In the last decade, there has been a few researches that carry out the analysis of facial expression using a dynamic 4D method. The dynamic facial expression recorded in 3D at a high temporal resolution conveys a much more complete representation of the variation in an individual's expression compared with a static image (or a 3D surface). The 4D facial expression acquisition monitors the changes of facial expression with time, which can provide the information about before versus after treatment. It is significant to develop such methods for analyzing the dynamics of facial expressions. This project explores a method to analyze the dynamic of facial expression. This method has three particular benefits:

- The method is capable of handling data in 4 dimensions (3D surface + time).
- The method can be used for analyzing a large data set.
- The method includes automatic landmark placement thereby decreasing the time spent on quantifying smile sequences.

This thesis evaluates the feasibility of a method that analyzes dynamic facial expressions based on a large data set.

1.2 Literature review

The literature review is fundamental to tackle our problems. There is a very limited number of scientific papers written on the subject. A previous paper[11] studies the smile of children recorded by video. They label the children's face using 884 landmarks, and uses software to track these landmarks over time. Subsequently, statistical analysis is carried out on the landmarks. The core of the procedure is that the motion of facial muscles is tracked by landmarks.

The rapid development of deep learning broadens the way in the image analysis field. The convolutional neural network enables the analysis both on 2D and 3D data. A multi-view consensus CNN architecture described in [3] provides an accurate and robust method to place landmarks. The 3D surfaces are rendered with different views and the rendering results are fed to the convolutional neural network which is a stacked hourglass network based on the residual blocks. The final predicted result is given by fusing the distribution of the landmark location in the space after applying the outlier-robust method. The method as described in [3] is tested and applied for our purposes of smile analysis in this thesis. The number and location of facial landmarks to place and analyze varies according to the purpose. It is therefore necessary to be able to easily re-train the network for new landmarks. Using the method developed in [3], it is easy to realize the re-training procedure on a selected training set.

Following the identification of a sparse set landmarks a validation is carried out on these. In addition to the use of a sparse set of landmarks a method inspired by the work of Hutton et al. [9] uses TPS and closest-point deformation with a deformable template face ("atlas") to obtain detailed point correspondence between faces. A pointwise method to quantify the shape asymmetry is introduced in [12] based on the use of a symmetric atlas face. The asymmetry is calculated in every point in the face as the difference in localization of the same anatomical point on the left and right side of the face. This method was applied in this thesis.

In [13], a set of shapes is subjected to a principal components analysis (PCA). Shapes are annotated with landmarks and then all landmarks are input to PCA. Authors experiment with the principal components analysis in the human hands and the resistors, both of

them perform a good result. The principal components analysis can use in this project to analyze the dynamics of facial expression.

In [9], a method of smoothing a trajectory in the PC space is presented. Correspondingly, the method can optimize the result of PCA in this project.

1.3 Thesis outline

The objective of this thesis is to explore a method for analyzing the dynamics of smiles in a large data set. The objective of this thesis is to implement a convolutional neural network for automatic landmarking and analyze the smiles according to these landmarks.

Chapter 2 is an overview and timeline of this thesis.

Chapter 3 includes the theory related to the project. The theory in this chapter is general and can be used for other projects that have a similar purpose.

Chapter 4 delivers a detailed description of the data set used such as the data format and attributes.

Chapter 5 presents the methodology of this thesis. The method in this chapter is specific and tailored for this particular project.

Chapter 6 presents the results of the project. Detailed discussion about the results are also presented in this chapter.

Chapter 7 presents a general discussion.

Chapter 8 provides a conclusion and some thoughts on future directions.

Chapter 2

Project Overview

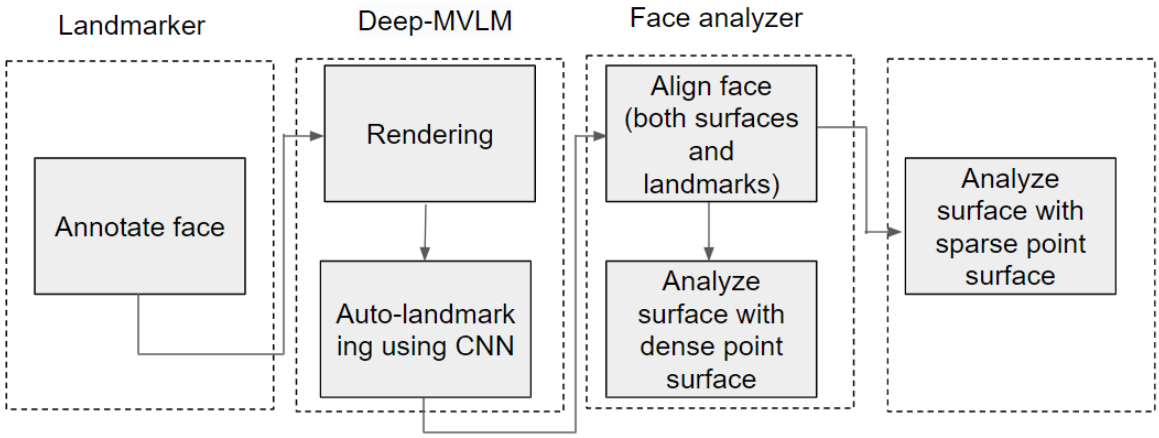


Figure 2.1: Flowchart of the entire method presented in this thesis. From left to right: Training sets are annotated (landmarked) using the software landmarker [1]. Deep-MVLM software [2] developed by Rasmus Paulsen at the Technical University of Denmark [3] based on multi view consensus CNNs (convolutional neural networks) is used to train and carry out automatic facial landmark placement. Analysis may be carried out in two different modes: using the whole face surface (dense point surface) or the landmarks directly identified by the neural network (sparse point surface). The analysis of dense point surface is carried out on Face Analyzer software based on landmarker [1]

2.1 Our method

The purpose of this project is to explore a method to investigate the dynamics of facial expression in 4D (3D surface + time). Figure 2.1 provides an overview of the entire method presented in this master thesis. The thesis deals with smile analysis in 4D which

is focused on the practical application to treatment of children and adolescents with facial anomalies.

Until now, the procedure is that landmarks are placed manually on each 3D face surface and subsequently using the landmarks to quantify the motion of a smile. This manual method is very time consuming (placement of 20 landmarks take up to 5-10 minutes) and this is not practically feasible for analysis of long duration high temporal resolution (60 frames per second) recordings of smiles. It has therefore almost solely been limited to static 3D surfaces or a few 3D surfaces (three or four) taken during a smile. Having the task of analyzing static surfaces or very short 4D sequences is feasible to analyze the full face (all points in the face) and not only the limited number of landmarks (typically 20) identified. Our task is much more drastic and calls for analyzing a very large number of surfaces (typically 500 per smile) to identify motion paths of anatomical points in the face over time. Automatic landmark identification can make the analysis of 4D smile sequences feasible. However, since the computational resources to analyze each surface at full spatial resolution are currently relatively large (20 seconds for each 3D surface to obtain detailed point correpondence) and carrying out a PCA on the full time series at full spatial resolution runs into memory issues, the current thesis has mostly limited itself to analysis of the sparse set of original landmarks.

This thesis achieves a not time-consuming method for 4D analysis. Convolutional neural network (CNN) is introduced providing a fully automatic method. The general idea is to quantify a smile and be able to compare it to a reference smile. The reference smile could for instance be a smile calculated as the average of several relevant individuals (e.g. normal individuals). As an example, the smile of a child with cleft lip and palate is different from the smile of a normal child. The method should be able to characterize a smile relative to a reference smile or relative to the smile of the same individual recorded at a different time (e.g. before and after treatment). A particular advantage of our method is that it is automatic and therefore less subjective. Some subjectiveness is nevertheless introduced by the procedure of training the CNN which requires manual input. The "opinion" of the CNN will reflect that of the human operator during the training process. It is therefore important to carefully design the training procedure. One of the risks of manual landmarking is that the accuracy of landmarks depends on the operator. A professional operator labels landmarks with a high accuracy, while a layman may bring some errors when he places landmarks. Apart from this, placing landmark one by one is a time-costly work.

In this thesis the number of facial landmarks is reduced to a suitable value. The number of anatomical landmarks was reduced to seventeen. This landmark structure can represent the whole face with a smaller number of landmarks. More information about the chosen landmarks is presented in chapter 5. It is foreseen that more landmarks will be added in the future in order to be able to track smaller spatial detail of the smile. The convolutional neural network we use is designed to analyze 2D images, so the 3D

face needs to be converted 2D images. Firstly, we set landmarks on each 3D face in the training set. Secondly, the 3D face is converted to the 2D image using rendering. Rendering results are fed to the convolutional neural network as described by Paulsen et.al [3]. Finally, the convolutional neural network train with the ground truth which we set. The trained network is used for placing landmarks on the 3D face. When the landmarks are automatically placed on all the surfaces in a smile sequence, the smile may be quantified.

There are two approaches to analyzing the smile. One of the approaches is where properties (e.g. asymmetry) of each surface (each time instant) is extracted and plotted over time. For the analysis of asymmetry we use a method based on dense point surfaces. In this approach, the dense point correspondence between faces is built to calculate the surface difference and surface asymmetry in every point across the face.

Another approach is based on the sparse set of landmarks directly identified by the CNN. The sparse surface is the face described with the landmarks that were defined in the training set. Our idea is that important characteristics of a smile may be reflected and quantified by its trajectory in PC space. Each landmark on the face can be seen as a feature of the smile. Principal components analysis can extract useful features (a new set of variables) from a set of smiles in a population of individiuals. This is the main smile analysis method that we used in this thesis. The theory behind the selected approaches will be explained in the next chapter. Chapter 5 describes the detailed methodology. The methods will appear in the same order they are applied.

2.2 Software

Some libraries and software are used in this project. I list all of the needed software and its functionality for the reader to understand the method more clearly.

The VTK framework is throughout this project. The Visualization Toolkit (VTK) is open source software for manipulating and displaying scientific data. It comes with state-of-the-art tools for 3D rendering, a suite of widgets for 3D interaction, and extensive 2D plotting capability [14].

Meshconv [15] is used to convert the format of face file.

The software named Landmarker[1] is based on the VTK framework used in this thesis. Landmarker is used for annotating landmarks on the face and visualizing the surface of the face.

Face Analyzer is an extension to Landmarker [1] that develops detailed point correspondence between faces using a deformable atlas method. This software is used for the surface alignment.

Deep-MVLM [3] is the packaged python code that predict the position of the landmark on the face by the convolutional neural network. This software is developed by Paulsen et al [3]. They package the python code so that we can use it by calling the command in

the command prompt.

The calculation of principal components analysis is based on the module named Numpy.

2.3 Hardware

Table 2.1 shows the technical specifications of the laptop used for this project. All of above software are installed on this computer.

Table 2.1: The hardware information of the laptop used for this project.

Name	Laptop
Model	Dell Precision 7730
Processer	Intel(R) Core(TM) i5-8400h CPU @2.5GHZ 2.50GHZ
RAM	32.0GB
Operating system	Windows 10 pro
System type	64-bit operating system, x64-based processor

2.4 Project timeline

Figure 2.2 represent the timeline of this project.

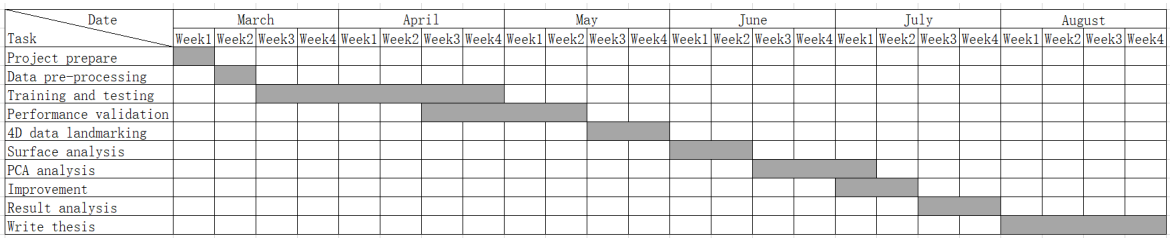


Figure 2.2: Project timeline

Chapter 3

Theory

This chapter explains the theory of the selected approach. An overview of this chapter shows as following:

- Section 3.1 introduces the rendering. The camera model and OpenGL pipeline are presented for describing the rendering process.
- Section 3.2 explains the method that using CNN to place landmarks on the 3D face automatically. The theory about convolutional neural network is introduced to describe how the CNN predict the landmarks on a 3D surface.
- The method of using sparse landmarks for describing a surface is illustrated in section 3.3. The sparse point surface is used for the principal components analysis.
- Section 3.4 introduces the procedure of building the dense point corresponding between surfaces. The dense point surface prepares for the surface pointwise analysis. Besides, landmarks are aligned in this procedure.
- Sections 3.5 and 3.6 define the surface pointwise difference and the surface asymmetry in this project, respectively.
- Section 3.7 presents the principal components analysis. It is used for analyzing the dynamics of the smile.
- Section 3.8 explains the kernel smoothing method, which uses for the improvement of the PCA result.

3.1 Camera model and viewing transformation

In chapter 5 the convolutional neural network use for placing landmarks on a face. The results of rendering are the input of network. This section illustrates the rendering process based on the OpenGL pipeline.

In the scene file, 3D models represent a physical body using a collection of points (also called vertices) in 3D space, connected by the geometric entities such as triangles. The pipeline accepts vertices of 3D models as input and produces a 2D array of color-values which the screen can shows. The procedure of the pipeline shows in figure 3.1.

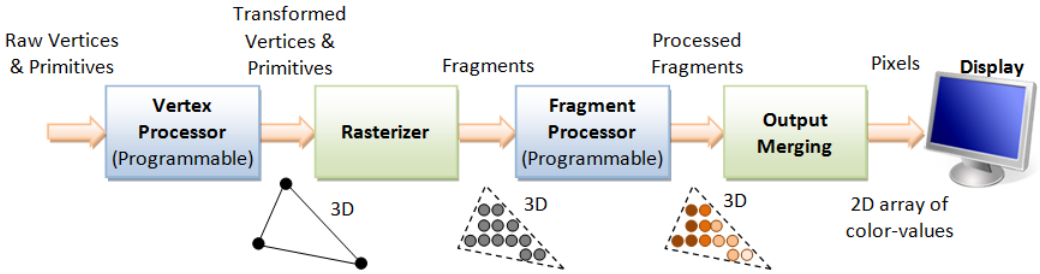


Figure 3.1: 3D Graphics Rendering Pipeline [4]

The pipeline process these vertices that transform the 3D model from real-space to a 2D image that the screen can show. If needed, the pipeline will transform texture information from a 3D model into a 2D image. The coordinate transformation is an important part of this process.

Due to the handedness of coordinate systems, images can be rotated, scaled, translated by the coordinate transformation. Besides, the geometric meaning of linear algebra makes the combination of these transformations are easy to calculate. A combination of transformations can compute using the multiplication of matrix. Some coordinate transformations are needed when we want to render a 3D model to a 2D image. The 3D model is located in the object coordinate system at the beginning. Different model has different object coordinate. To process a set of models easily, each model needs to transform into a common coordinate system, which is named the world coordinate system. Then models are transformed from 3D space to 2D image and then convert to the pixel coordinate system. This is the whole procedure of the rendering pipeline. The transformation process illustrates in figure 3.2.

As seen in the figure, the process of the transformation from 3D into 2D experiences two times transformation: camera transformation and projection transformation. The reason for using these two transformations is that only computing coordinate transformation and then discarding depth information directly cannot provides enough information about models.

Suppose there is a model in real space, the different viewports will result in the different observations of this model. As seen in figure3.3, a camera is located in real-world space. The near clip plane is in front of the camera. The cube represents a part of the viewing pyramid. The clip plane represents the image observed from this viewport. The green ball that out of the cone is clipped. The red ball that has a larger distance from the viewpoint (viewpoint here means the location of the camera) shows a smaller size than

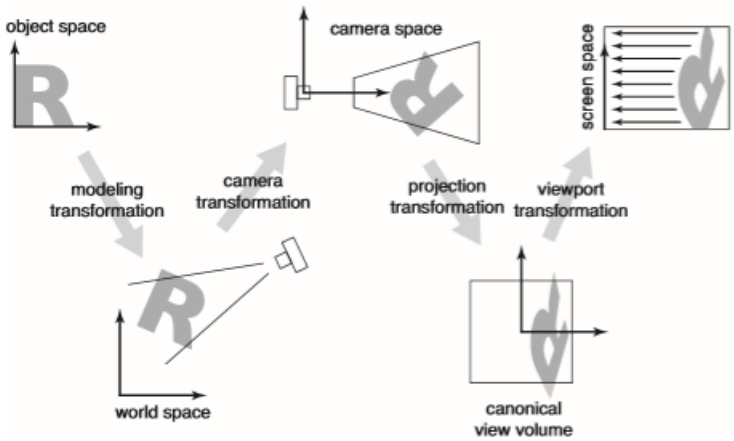


Figure 3.2: The coordinate transformation procedure of rendering pipeline. [5]

the yellow ball in the near clip plane. But if we set a camera at another position, the near clip plane will show a different result.

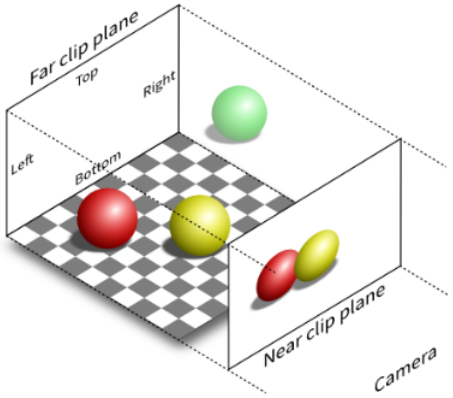


Figure 3.3: camera model [6]

To render the object located at 3D space correctly, we need to set a virtual camera to 'observe' the model in 3D space. Once a viewpoint is set, depth information can provide the right transformation relationship of this view. Setting the viewport can discard depth information correctly. The different viewports provide different rendering results. Setting different viewports of a model can get the desired number of 2D rendering images.

3.2 Landmarking using CNN

Figure 3.4 illustrates the overall system of the 3D landmark predicted procedure. The convolutional neural network is commonly used for 2D image analysis. For example, CNN maybe trained that, given an image of the face, output a heat map of the 2D landmark.

To expend this method to a 3D case, CNN is combined with a multi-view approach to identify landmark position on 3D face surfaces. For this, each face is rendered to multiple views, then feed rendering results to the CNN module. CNN will generate one heat map for each landmark. A landmark has one 3D view ray in each view. The 3D view rays of one landmark will intersect to a point in real-world space. Using the voting process (RANSAC and least squares estimate(LSQ)) removes outliers. The position of landmark in real-world space is the intersection point of its view rays.

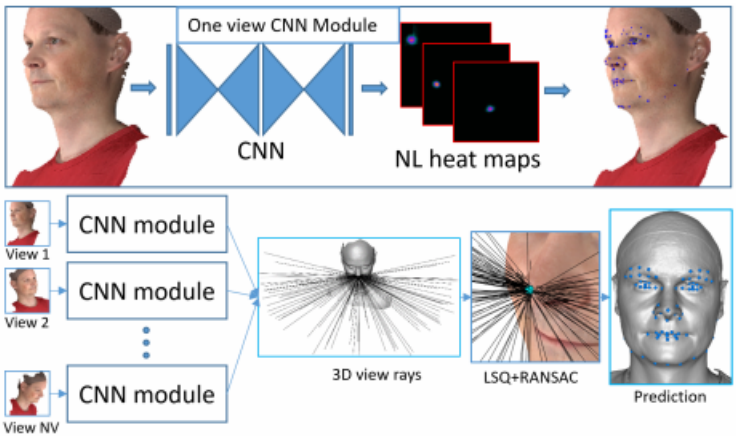


Figure 3.4: Illustration of landmarking method [3]. NV is the number of the view. NL is the number of the landmark.

The network architecture in the figure is the stacked hourglass model based on residual blocks. More details about network architecture are discussed in [3].

3.3 Sparse landmarks for describing a surface

The principal components analysis is described in chapter 5. The input of principal components analysis is the faces represented by a set of sparse landmarks.

The facial landmarks are the points that have the anatomical feature on the face. These points are easy to place manually and define distinct facial features. Figure 3.5 shows an example of facial landmarks. In the figure, red landmarks define the anatomical features, and blue landmarks are pesudo-landmarks that define the curves and width of eyes, lips etc. In this thesis, a less number of landmarks are used for describing a face in principal components analysis. We called it sparse landmarks. The sparse landmark structure used in this project is presented in chapter 5.

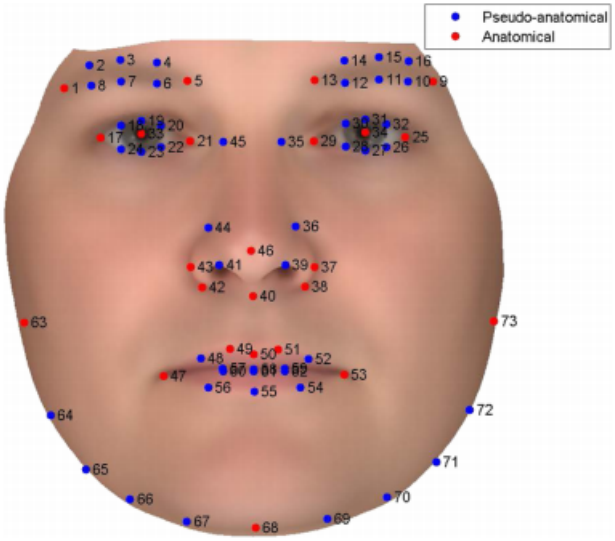


Figure 3.5: Example of manual annotation of landmarks[7]

3.4 Build dense point correspondence between surfaces

The goal of building the dense point correspondence between surfaces is to compute the pointwise relationship between faces. In this project, we use the dense surface model for measuring the facial asymmetry and the difference between facial expressions. The 3D surface model, as illustrated in section 3.1, represent a physical body using a collection of vertices. The goal of building the dense point correspondence between surfaces is to transform all faces in a data set to minimize the distance of the corresponding vertices between faces. The method of building dense point correspondence between surfaces differs from authors. The following sections present one of the methods, which is inspired by [9]. Landmarker [1] utilize this method. The starting point of this method is to align hand-placed landmarks at points that have high reproducibility of biologically homologous. Then apply deformation to make a more accurate model. The procedure shows in figure 3.6.

3.4.1 Standardization of head orientation

The orientation means that the surface is translated and rotated to a same location and orientation. The method normally uses for the analysis of a skull. This method standardizes the orientation of the face according to the Frankfort horizontal plane (FHP) and the mid-sagittal plane (MSP). The mean horizontal plane and mean sagittal in figure 3.7 are FHP and MSP, respectively. The FHP is the plane defined by three points on the skull: the upper margin of the opening of the left and right external auditory canals and the lowest point on the lower margin of the left orbit. The mid-sagittal plane (MSP) divides a normal head into left and right halves.

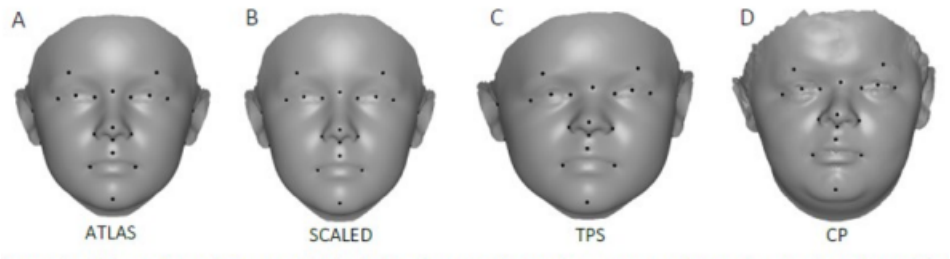


Figure 3.6: Illustration of the steps included in the transformation process of the template face. A: the template face. B: the template face after scaling. C: the template face after thin-plate-splines transformation. D: the matched template face after closest point deformation.

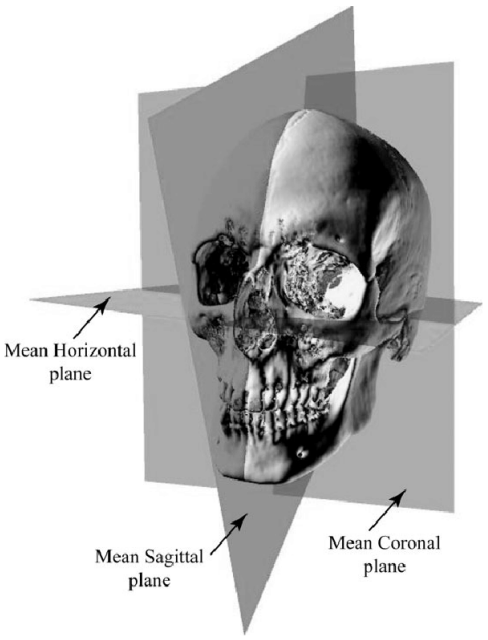


Figure 3.7: Reference plane

The aim of the standard orientation is that transforms the skull to the coordinate system defined by FHP and MSP. Besides, the skull is facing in the x-axis of the Cartesian coordinate system. The procedure of orientation shows in figure 3.8.

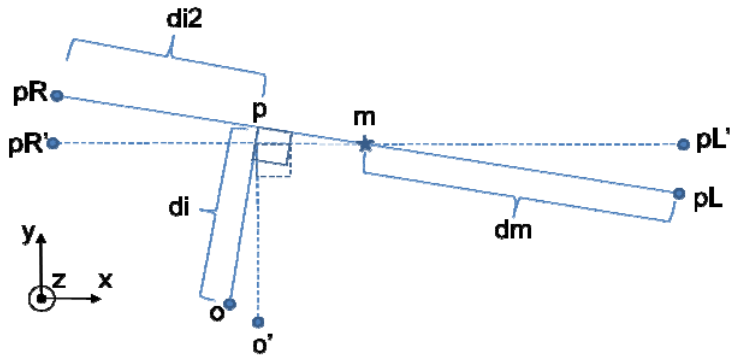


Figure 3.8: The illustration of orientation

In the figure, the pR and pL is the upper margin of the opening of the left and right external auditorycanals. O is the lowest point on the lower margin of the left orbit. The defination of pR, pL and o can change with the structure of landmarks. The following steps carry out the construction of the target coordinate:

- Find mid-point m between pR and pL and then compute the distance from m to pL as dm.
- $pR' = [m_x - dm, m_y, m_z]$, $p'_L = [m_x + dm, m_y, m_z]$.
- The perpendicular through point o to the line between pR and pL is constructed and intersects to point p. di is the distance between point o and the line between pR and pL.
- $di2$ is the distance between pR and p.
- $o' = [pR_x + di2, m_y, m_z]$, which is the origin of target coordinate system.

Orientation contains rotation and translation, which are all of the rigid transformations.

3.4.2 Thin plate spline interpolation

Thin plate spline (TPS) transformation is a commonly used interpolation method to deform the non-rigid shape. In this project, the TPS is for transforming the template face to an individual face. Using TPS aligns the landmarks between the template face and an individual face. Then the following deformation will be more accurate. Most of the biological deformation can be estimated by TPS. The name thin plate spline refers to a physical analogy involving the bending of a thin sheet of metal. Just as the metal has

rigidity, the TPS fit resists bending also, implying a penalty involving the smoothness of the fitted surface [16]. TPS computes a transformation that has the minimized energy of bending. It makes the distortion of the surface minimum. The purpose of TPS transformation is to deform one shape to another shape to make them has similarity.

Figure 3.9 shows a graphic portrayal of TPS transformation between (A) and (B). There are two sets of landmarks in (A) and (B). arrows in (C) is the displacement vectors of landmark, which shows intuitively the direction of each landmark transformed.

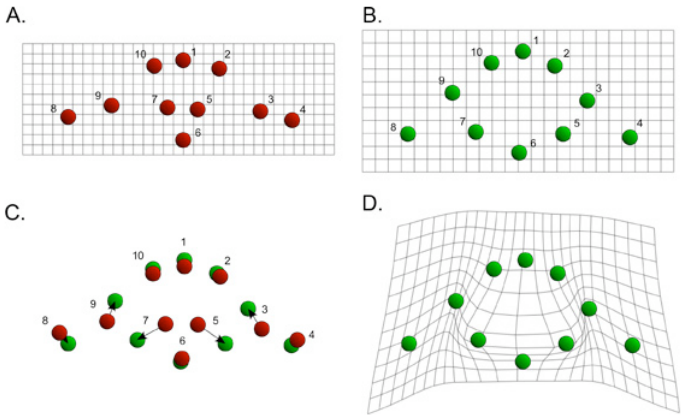


Figure 3.9: Thin plate spline [8]

As seen in (D), displacement of landmarks 5 and 7 is larger than other landmarks. Accordingly, the TPS representation of this deformation has a strong changing of grid lines in the region of these landmarks. Also, this deformation costs minimal energy.

Using the VTK framework can realize thin palate spline interpolation easily. It has a related function to describe the deformation by a set of source landmark and a set of target landmarks. The source landmark moves to the position of the target landmark with minimal energy.

3.4.3 Closest point deformation

This method makes that two surfaces has a same biological shape. The method of closest point deformation is quite easy: move each point in the source point to the point in the target point set that has the closest distance with this point (figure 3.10). The VTK framework provides functions to deform surface by closest point deformation.

3.5 Facial pointwise differences

Having the dense point corresponding between surfaces makes the calculation of facial pointwise differences easily. The surface pointwise differences measure the shape similar-

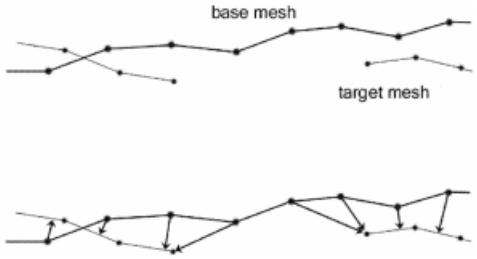


Figure 3.10: Closest point of two sets of point [9]

ity. Recall the rendering part, a collection of triangles represent a 3D model, like figure 3.11.

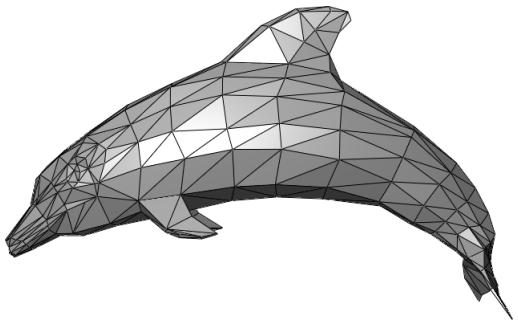


Figure 3.11: Dolphin triangle meshes [10]

The surface pointwise differences are calculated by making a geometric comparison between triangles of both two surfaces. The differences between the two surfaces are the Euclidean distance of corresponding triangle centers. Each triangle has one value to represent the difference. The distance of the correspondence triangle center defines the distribution of facial pointwise differences.

3.6 Facial asymmetry

Figure 3.12 illustrates the method to compute asymmetric face according to the MSP plane and FHP plane. This method is based on the dense point surface.

t and s represent the transverse and sagittal directions, respectively. The blue line draws the profile of a face. The green line is the mirror profile according to the MSP plane. The vector between p and p'_{mirr} defines the asymmetric value A. This process needs that p and p'_{mirr} are anatomically corresponding points on the left and right side of MSP, respectively.

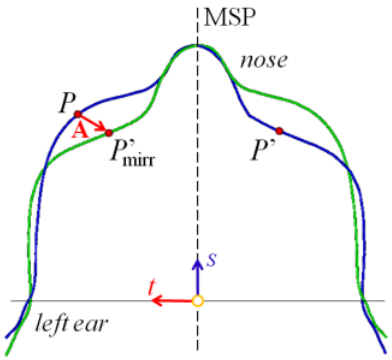


Figure 3.12: Schematic of computing asymmetric face

3.7 Principal components analysis

Principal components analysis (PCA) is a used tool to reduce the dimensionality of data. PCA reduces dimension by orthogonal transformation, which will convert a set of linearly correlated data into a set of linearly independent data to keep more essential features. Taking the features of the smiley face as the input, we can get principal components of the smile.

3.7.1 Eigen decomposition

Given N features of each data $p_1, p_2, \ldots, p_N \in R$, there are number of S data in the data set,

$$x_i = [p_1, p_2, \ldots p_N] \tag{3.1}$$

where $i \in [1, S]$. The mean of this data set is given by

$$\bar{x} = \frac{1}{S} \sum_{i=1}^{S} x_i \tag{3.2}$$

Then Zero-centered the whole data

$$X = \begin{bmatrix} (x_1 - \bar{x}) \\ \ldots \\ (x_N - \bar{x}) \end{bmatrix} \tag{3.3}$$

Compute the covariance matrix

$$CovMat = \frac{1}{s-1} X^T X \tag{3.4}$$

The parameters we are interested in are eigenvalues and eigenvectors of each feature. The eigenvector shows how the original data transform. The eigenvalue is the variance of data

in the corresponding dimension. Define the eigenvectors matrix as $\phi = [\phi_1|\phi_2|\dots|\phi_S]$ and eigenvalues of every feature as $\lambda_1, \lambda_2, \dots, \lambda_N$. Because of the large variance indicates a more scattered data, the eigenvectors descend sort in the eigenvectors matrix according to the corresponding eigenvalue. Define the data after reducing dimension as $b = [b_1, b_2 \dots b_k]$, b is given by

$$b = \phi_k X \tag{3.5}$$

where k is the dimensionality of data after reducing, ϕ_k is the eigenvectors matrix with first k eigenvectors.

3.7.2 Data reconstruction

Except for the dimensionality of data reducing, sometimes we want to reconstruct data. According to equations 3.5 and 3.3, the data can transformed from a low dimension back to data in a high dimension. The reconstructed data x' which located in point b of the low dimension is given by

$$x' = \bar{x} + \phi_k^T b \tag{3.6}$$

3.8 Kernel smoothing method

The kernel smoothing method is a statistical technique to estimate the source function. It uses a weighted average of neighboring value to observe data and get the smoothed estimated function. This method gets the regression function by fitting each point. In this project, the kernel smoothing method is used to smooth the smile trajectories in the PC space. The smoothness can set by the kernel parameter. The kernel function is defined by

$$K_h(X_0, X) = D(\frac{\|X - X0\|}{h(X0)}) \tag{3.7}$$

where $X, X_0 \in R^P$, $h(X0)$ is the kernel radius that changes the smoothness of estimated function, $D(t)$ is a positive real-valued function that decreases (or no increases) according to the increasing of the distance between the X and X_0. Supposing there is a function $Y(X)$, for each $X_0 \in R^p$, the estimated function is defined by

$$\hat{Y}(X_0) = \frac{\sum_{i=1}^{N} K_h(X_0, X_i) Y(X_i)}{\sum_{i=1}^{N} K(X_0, X_i)} \tag{3.8}$$

Figure 3.13 shows some of kernel functional graphs.

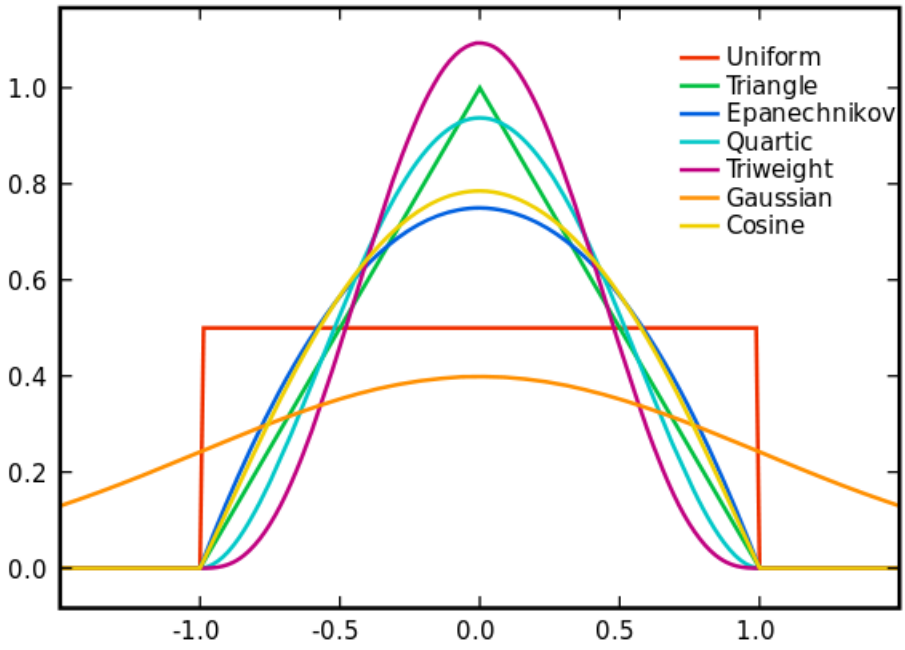


Figure 3.13: Some kernel

Chapter 4

Data description

The data we use in this thesis is from two data sets. One is the BU-3DFE data set from Binghampton University[17] that provides different facial expressions from a hundred normal adults. The other is from the 3D Craniofacial Image Research Laboratory. Five normal adult volunteers performed a smile while being recorded by the 3D scanner. BU-3DFE data set is used as training set and also for testing the performance of the convolutional neural network. The data from the 3D Craniofacial Image Research Laboratory is for developing the method of analyzing the dynamics of smiles.

4.1 BU-3DFE

The BU-3DFE data set comprises a total of 100 subjects (56 females, 44 males). It describes seven types of facial expressions, and each facial expression consists of intensities of four levels [17]. Each facial expression is documented as a single 3D surface captured when the subject is instructed to make that particular expression. we called this the static data set. The 3D surfaces in this data set are triangle mesh surfaces. The file format is WRL. A subset of the BU-3DFE data was chosen as the training set for the convolutional network: a smiling face with intensity of 2 and a smiling face with intensity of 4, and a neutral face (seen in figure 4.1).

A first training set contained all the neutral faces and smiling faces with intensity of 4. To improve the performance of the network, a second training set was created by adding 30 smiling faces of intensity 2 to the first training set. Thus the training set is composed of a hundred neutral faces, a hundred smiling faces of intensity 4, and thirty smiling faces with the intensity of 2. There are a total of 230 subject in the largest training set that we used. Several experiments were carried out with different sizes of training sets in this thesis. Details about the inputs to the convolutional neural network are described in the next chapter.

(a) (b) (c)

Figure 4.1: BU-3DFE data set. (a) Smile with intensity of 2. (b) Smile with intensity of 4. (c) Neutral

4.2 Dynamic smile sequence

This data set is produced by 3D Craniofacial Image Research Laboratory at the University of Copenhagen. The data set consists of facial 3D scans acquired by the 3dMD surface scanner (figure 4.2). This system has five camera units and can acquire full head geometry and color at a temporal resolution of 60 frames per second. Dynamic facial expression can be recorded as a time sequence for a subject sitting in the system chair centered between the cameras. The maximum recording time is 15 seconds. We call this the dynamic data set. The surfaces provided by this scanner are represented as triangle meshes. This scanner produces an OBJ file containing geometry for each facial expression, an MTL file which contains extra information about properties, and a JPG file which contains texture information.

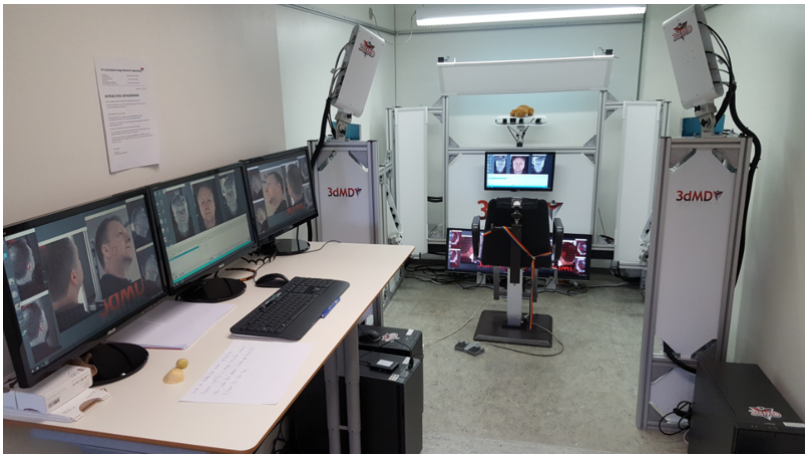


Figure 4.2: 3dMD scanner

Four volunteers recorded their smile using this scanner. Some of volunteer have more than 1 smile sequence. The volunteers were scanned with 60 frames per second time resolution. In general, there are hundreds of surface files in a smile sequence. We need to pre-process the data before carrying on analysis, e.g. principal components analysis. The method of data pre-processing will be described in the next chapter.

Chapter 5

Methodology

This chapter introduces the methodology according to how it uses the theory described in the previous chapter. The data preprocessing process is described in the first section. The details of the network training process and the performance evaluation of different networks are presented next. Then a description of how to build dense point correspondence between surfaces. The calculation of point-wise difference between faces and facial asymmetry is described, and the PCA analysis as applied in this thesis is illustrated. Finally, the kernel smoothing method for trajectories in PC space used in this thesis is introduced.

5.1 Data pre-processing

In general, data pre-processing is an important step not just in deep learning, but also in the context of most experiments where data analysis is needed. In relation to the BU-3DFE data set we need to transform the 3D mesh surfaces into a form that the neural network can take as input. For the dynamic smile sequence, we need to remove the redundant information from 3D mesh surfaces.

5.1.1 BU-3DFE

Each face in the BU-3DFE data set has been annotated with 83 landmarks as figure 5.1. In [3], the network train with these 83 landmarks.

In this thesis, we have decided to focus on a smaller number of landmarks that traditionally have been used in facial analysis. Using a smaller number of landmarks also has the advantage of reducing computation time. The principle behind choosing new landmarks is that they are easy to identify by a human so that the manually placed landmarks has a high accuracy and precision. Landmarks should also be distributed over the whole face in order to capture motion in the whole face. The new landmark configuration is shown in 5.2.

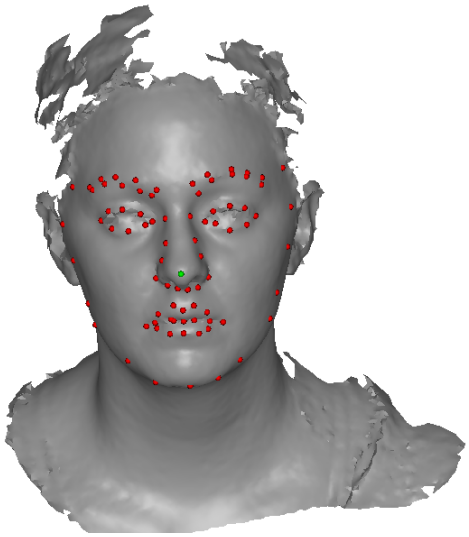


Figure 5.1: Structure of the original landmark

Figure 5.2: Structure of the new landmark

Two software programs were used to set the new landmark: Landmarker [1] and meshconv [15]. The process of manually placing the landmarks consists of two steps:

- Changing data format. Sometimes the software will specify a certain file format as input. Landmarker cannot read WRL files, so the transformation between different file formats is needed. File format transformation extracts useful information from the source file and rewrites it to the target file. Our purpose is to transfer the WRL file to the OBJ file. This process is done using Meshconv [15].
- Setting landmarks using landmarker[1]. The landmark sequence is shown in appendix A.

5.1.2 Rendering for automated landmarking

The 3D surface is transformed into the 2D images using rendering. In [3], the 3D object is rendered using the VTK framework. The method provided by Rasmus Paulsen's homepage[2] is used to render data with new ground truth. Paulsen randomly chooses the camera positions in a certain range in his work. Then the 3D object is rendered by the different viewport. We continue to use his parameter of camera location in this thesis. The range of angle of view of the x-axis is from -40 to 40. The range of angle of view of the y-axis is from -80 to 80. The range of angle of view of the z-axis is from -20 to 20. One object is rendered to 96 views. The configuration of the rendering is presented in appendix B.

Figure 5.3 shows one of the rendering results. As seen in the figure, landmarks transform to its corresponding 2D position with a correct way. The depth information is discarded. The rendering result of Palsaun's method has three image types: depth image, RGB image and geometry image. In this work, the depth image is used as the input of the neural network. It should be mentioned that the color information was discarded. The reason is that the training part is the first step in this thesis. At that time, I am not sure what kind of data set I will get in the future. I prefer to select the image that contains the information as less as possible so that I will not have trouble with the supplementary file such as MTL file. The color information could be included in the future.

5.1.3 Time normalization for dynamic smile sequence

Four volunteers had their smiles scanned in the 3dMD scanner. Each person had between one and three smile sequences recorded at 60 frames per second. It was decided to select only a subset of the surfaces for analysis in order to speed up the pre-processing and analysis. Furthermore, a time normalization of each smile sequence was carried out in order to approximately match sequences from different subjects. Although each subject was instructed to change expression from a neutral face to a maximum smile and back to neutral again, there were differences in duration of the smile sequences of different

Figure 5.3: Rendering result of the depth image

subjects. Matching the sequences in terms of time would make it possible to e.g. calculate a mean smile or a difference between smiles.

In practice, the time normalization consisted of selecting 100 frames from one sequence and forming a new sequence from these. The sequences were designed such that the first fifteen frames are neutral face before the smile begins, frames from no.16 to no.40 represent the process from beginning the smile to max smile, frames from no.41 to no.60 are at or close to the maximum smiling face, frames from no.61 to no.85 represent the process of going from the maximum smile back to a neutral face. The last fifteen frames represent the neutral face after a smile. Figure 5.4 shows frames from two smile sequences acquired with the 3dMD scanner. As seen in the figure, smile has similar strength in corresponding frames in two different subjects.

5.2 Landmarking automatically

It is very easy to re-train the network according to the guide in [2]. The details about the configuration of the training are presented in Appendix B.

Table 5.1 shows some important training parameters. The MSE loss is mean square error. The equation is given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y_i})^2 \tag{5.1}$$

Adam is an optimization algorithm that can adaptive changes the learning rate in the training process. Drop out is an regularization method for reducing the overfitting.

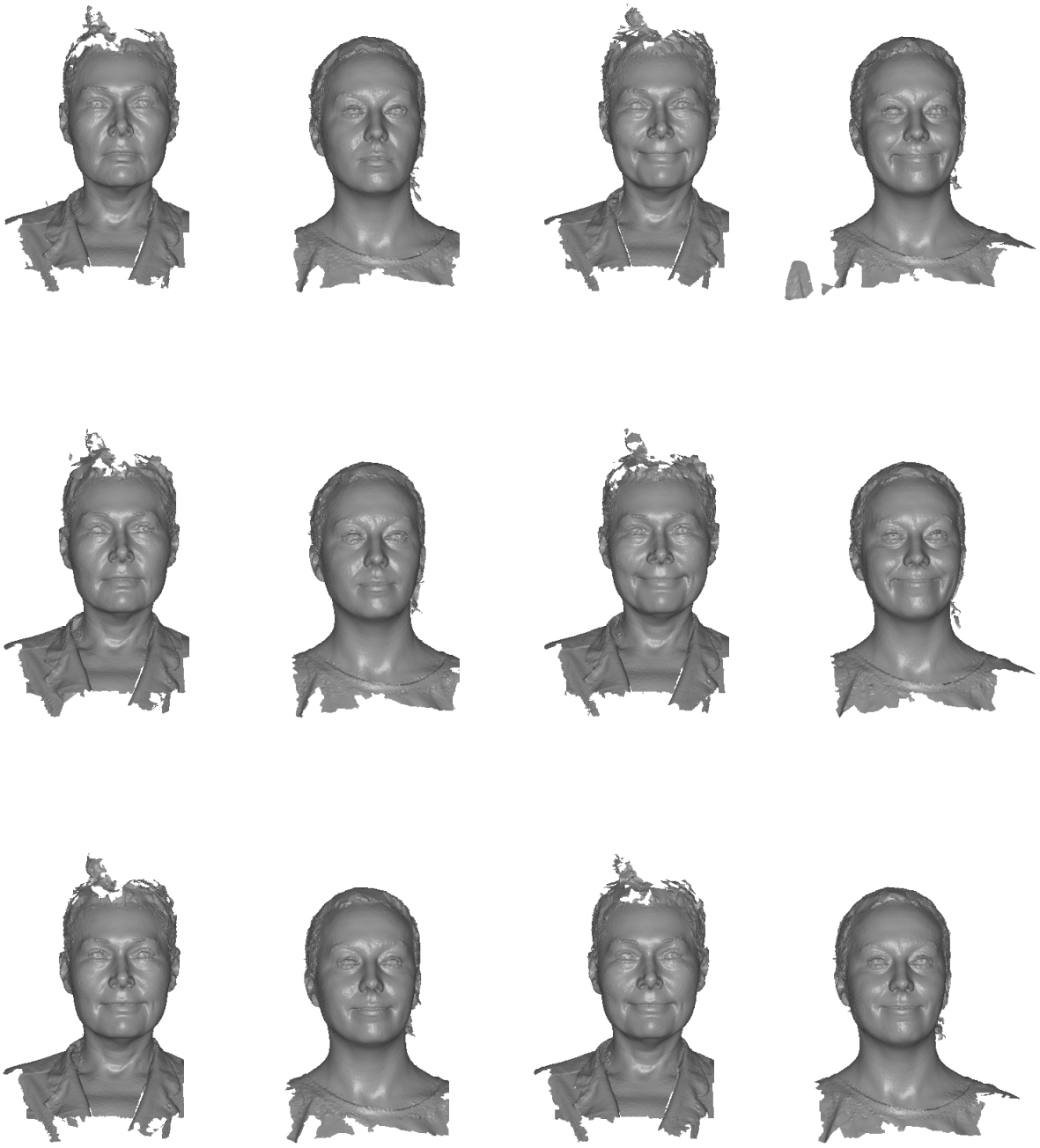


Figure 5.4: Data from 3dMD scanner. The first and third columns represent a smile sequence of a subject. The first and second columns represent the facial expression of the two subjects in corresponding frames.

Table 5.1: Training parameter

item	parameter
batch size	8
optimizer	Adam
learning rate	0.001
loss function	MSE loss
drop rate	0.2

Several experiments were carried out using different sizes of training data sets. Table 5.2 shows five different sizes of training sets that were used in this project. The network that performed the best was chosen for the task of automatic landmark placement.

Table 5.2: The design of different training sets. Smile4 is the smiling face with the intensity of 4. Smile2 is the smiling face with the intensity of 2. Neutral is the neutral face.

Network	No. of object	Facial expression	No. of input image
A	100	50 smile4 + 50 neutral	8640
B	130	65 smile4 + 65 neutral	11232
C	160	80 smile4 + 80 neutral	13284
D	200	100 smile4 + 100 neutral	17280
E	230	100 smile4 + 100 neutral + 30 smile2	19872

5.3 Benchmark test of the trained networks

There are a total of 100 subjects in BU-3DFE data set, which means that each facial expression has a limited number of faces representing it. All the smiling faces with the intensity of 4 and neutral faces are used for training the five networks. Three test sets, as described in the following, were used for testing the performance of the five networks.

- Test set 1. This test set contains 30 faces that were not included in the training sets of the first three networks while the training sets of the last two networks include it.
- Test set 2. This test set contains 20 smiling faces with the intensity of 2.
- Test set 3. This test set contains 20 smiling faces with the intensity of 3.

The ground truth of test set 1 is consists of landmarks that I annotated at the beginning of the project period, about 5 months before. The ground truth of test set 2 and 3 is

consists of landmarks that I annotated at the end of the project. The test set 1 was used for testing the change of performance with the increasing number of scans the training sets. Test set 2 was used for testing the accuracy of automatic landmark identification in faces with expressions in the range between the neutral face and maximum smile. The smiling face in the test set 3 is similar to the smiling face with the intensity of 4 in the training set. This is validated in Section 6.5. So this test set is used to test the performance of all five networks. The predicted landmarks are compared with the annotated landmarks. Euclidean distance between ground truth landmarks and predicted landmarks measures the error.

5.4 Intra-test of manual landmarks

As explained in section 5.3, the ground truths in the three test sets are obtained twice by duplicate landmarking. The intra-test of manual landmarks investigates if the double measurement is consistent. Euclidean distance between the first and second landmark position is calculated and used as a representation of intra-observer error. The following parameters are calculated for each landmark:

- Mean of error.
- Standard deviation of error.
- Root mean square of error.
- s is a value used by clinicians. If it is below 1 mm it is acceptable. It is defined in the article by Houston [18], and given by an equation termed Dahberg's formula:

$$s(i) = \sqrt{\frac{\sum d^2}{2n}} \tag{5.2}$$

  where d is the distance. n is the number of the subjects.

- p is the p-value of a Student's t-test for testing for statistically significant means in the location of landmarks of the two rounds of landmark placement. A value of p that indicates a significant difference raises a flag that could indicate a systematic error.

5.5 Alignment of surfaces

For the analysis based on dense point surfaces, it is important to align all scans according to the same coordinate system. It is a necessary pre-processing step before computing the surface pointwise difference and asymmetry after building the dense point correspondence

between surfaces. Due to the fact that the poses of people are different when they are being scanned, the location and direction of heads are different (figure 5.5). Also, the resolution of the surface may be different according to the scanner setting. Higher resolution means that the surface is represented by more vertices. To make the analysis easier, we need to remove the redundancy information of surfaces.

(a) scan 1 (b) scan 2

Figure 5.5: Different scan

For the analysis based on sparse points, the alignment is also important, for instance when carrying out principal components analysis. Alignment of surfaces and landmarks and the process of determining detailed point correspondence between surfaces is carried out in the software named face analyzer. Face analyzer follows several steps shown in Figure 3.6. The details about building detailed point correspondence between surfaces will be described in the following sections.

5.5.1 Orientation of surface

The facial orientation is standardized to make the surface facing the same direction and reset the origin in space at which the scanned object is located. We modify the definition of the FHP plane and MSP plane to fit the structure of landmarks in this project. In figure 3.8, pR and pL is defined as the landmark of the right eye outer corner and the left eye outer corner, respectively. The origin o is the mid-point of nasion and nose tip. All of the surfaces are oriented in the same way. Figure 5.6 shows one of the standardized surfaces.

(a) (c)

Figure 5.6: Example of orientation. (a) The surface before orientation. (b) The surface after orientation

5.5.2 Scaling

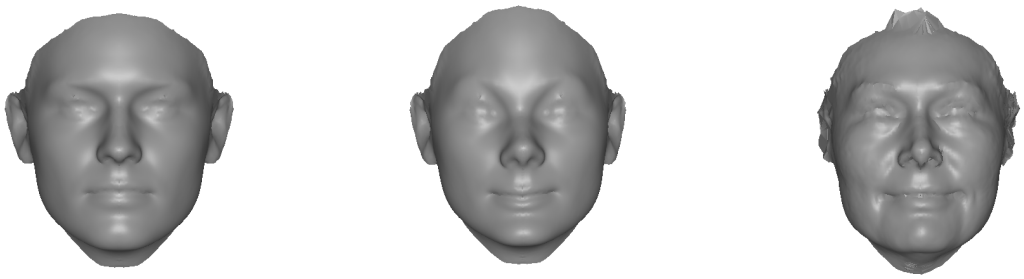
Scaling means changing the size of a 3D surface. The purpose of scaling is to filter out size in order to be able to focus on shape information. The scaling step is omitted in this work in order to be able to estimate true dimensions.

5.5.3 Non-rigid deformation for getting the shape similarity

In order to obtain detailed point correspondence between surfaces, a symmetric template face is deformed to take on the shape of each scanned face. Thin-plate-spline (TPS) is used to warp the template surface onto each surface, which results in all the surfaces being represented by the same number of vertices. A template face after making TPS transformation is shown in figure 5.7(b). The template face in figure 5.7(a) is that after making orientation.

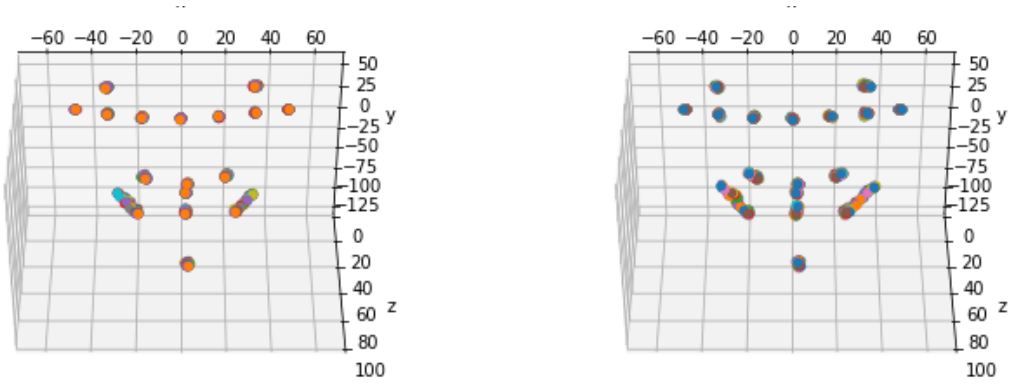
All surfaces are brought into close alignment after TPS transformation. Finally, the dense point correspondence between surfaces is built by closest point deformation. This step makes each template surface that is TPS-warped has a specific biological shape. Figure 5.7 (c) shows a result after closest point deformation.

The landmarks on the face are also aligned through above procedure. Figure 5.8 shows two examples of the aligned landmarks. Figure 5.8 (a) and figure 5.8 (b) represent the smile process of two different subjects.



(a) (b) (c)

Figure 5.7: Example of TPS + CP. (a) The template surface. (b) The template surface after TPS transformation. (c) The surface in (b) after closest point deformation.



(a) (b)

Figure 5.8: The smile process process of two different subjects described with aligned landmarks.

5.6 Pointwise difference and asymmetry

After detailed point correspondence has been achieved, it is a simple process to calculate pointwise differences between faces or calculating averages of several surfaces. The pointwise difference between surfaces may be used to analyze the changes caused by an individual smiling. For example, the difference between a smiling face and a neutral face of an individual shows the movement of facial muscles in the smile process. There are several smile sequences of each person in the dynamic smile sequence. For an individual, the average surface of each facial expression is computed. The average surface is used to calculate the surface pointwise difference between a smiling face and a neutral face. For each individual, the average surface of three facial expressions is computed: the neutral face which is before the smile, the maximum smiling face, and the neutral face after the smile. The pointwise difference between these three facial expressions is computed.

The same facial expression is used to calculate the asymmetry of the face. The asymmetric value of each facial expression is calculated by the average surface of each facial expression.

5.7 Principal components analysis for smile analysis

To analyze the smiles of an individual or even a large population of individuals, it is necessary to analyze a large number of surfaces. Working with the large size of data is computationally heavy and pushes us to find a solution that can represent data in a lower dimension. PCA was chosen in order to extract important information about the smile sequences. The data set is high dimensional and morphologically the human smile process has similar topology. These topologies can be represented in a lower dimension. For example, people generally move their mouth corners up and down during their smiles. It would be expected that PCA is suitable for our purpose.

Recalling the data pre-processing section, one hundred frames record the smile process. The face is described by 17 landmarks, as shown in figure 5.8. The input of PCA is a matrix. Each row of the matrix contains features of each frame. Each column of the matrix contains the same feature of whole frames. The dimensionality of data reduces to 3D for visualization in principal components space (PC space). So the eigenvectors matrix in equation 3.5 is composed of the first three eigenvectors in ϕ ($k = 3$). After each frame transforms to a point, points in the same smile sequence can connect by time. Then the smile trajectory can be shown in PC space. This trajectory represents a smile. In this project, we calculate the average smile trajectory of each individual and then use these average smile trajectories to compute the mean average smile trajectory for a group of individuals.

Except for reducing the dimensionality of data, dimensional recovery is also important. It is very useful that a point in principal components space transforms back into the

real-world space to synthesize a new face for the smile analysis. A 3D point in principal components space can be reconstructed to the face in the real-space according to equation 3.6.

In this thesis, equation 3.6 is also used in the survey of how each principal component impacts a facial expression. The b represents a point in the PC space. The first parameter in b describe the first principal component, the second parameter in b describe the second principal component, and so on. The influence of each parameter can be visualized by setting the related parameter in a certain range and other parameters as zero.

5.8 Smoothing smile trajectory in the PC space.

There are two reasons for using smoothing method. On the one hand, the input of PCA is a set of smile sequence changes with time. Each frame represents a time point in the sequence. Crucially, these frames cannot represent a continuous-time sequence. It is impossible to express a continuous-time by a bunch of frames. The kernel serves to interpolate between two neighboring frames to produce a continuous smile sequence that changes with time. On the other hand, errors in landmark localization adds noise to the smile trajectory. Smoothing method is effective for removing this noise.

In this project, I test two kernels for smoothing trajectory: triangular and Gaussian. The triangular kernel is used to obtain the trajectories that change with time finally. The Gaussian kernel produces a similar result. The trajectory function $s(t)$ is

$$s(t) = \frac{\sum_{i=1}^{n} K(s_i, t) b_i}{\sum_{i=1}^{n} K(s_i, t)} \tag{5.3}$$

where n is the size of data set, s_i and t are, respectively, the time and smile location in shape-space of ith subject. b_i has the same meaning as equation 3.5.

5.8.1 Triangular kernel

The function of the triangular kernel is

$$K(u) = \max(1 - \frac{|1-u|}{w}, 0), \quad u = s_i - t \tag{5.4}$$

where t is the time length. w is the width that can change the smoothness of the curve. A large value makes the curve more smooth, while a small value tends to introduce noise because the curve is influenced by individual data.

Figure 5.9 shows the smoothness of trajectory changes with different widths. When the width equals to 0, the original trajectory and smooth trajectory are overlapping. When the width increase to 10, the original trajectory is efficiently smoothed.

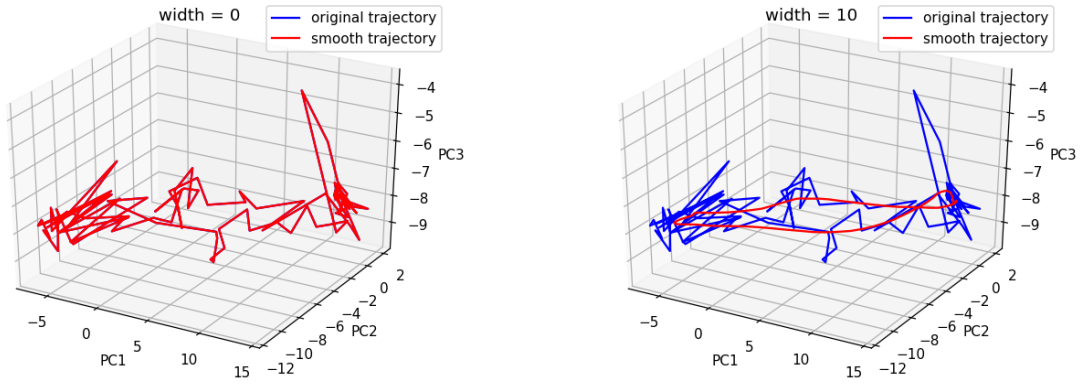


Figure 5.9: Smoothness with the different width of triangular kernel

5.8.2 Gaussian kernel

The function of Gaussian kernel is

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2w}u^2}, \quad u = s_i - t \tag{5.5}$$

As seen in figure 5.10, the Gaussian kernel shows a more smooth result than the triangle kernel when the width equals to 0, while it performs a similar result when the width equals to 10.

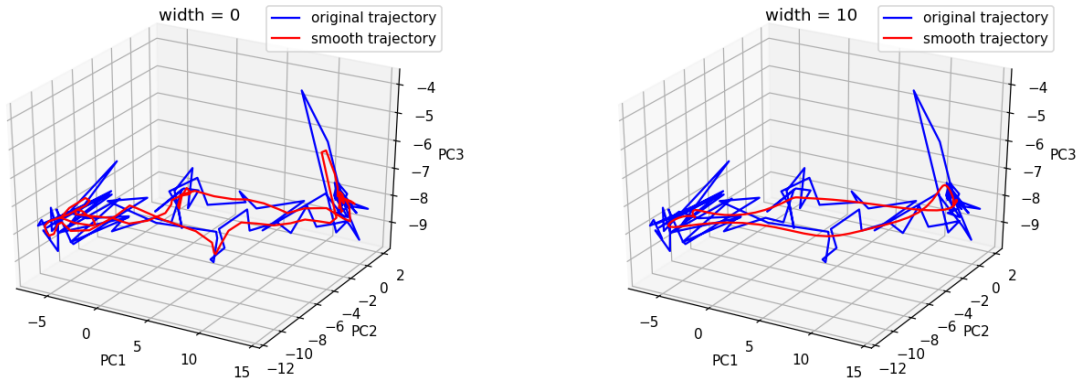


Figure 5.10: Smoothness with the different width of Gaussian kernel

Chapter 6

Results

This chapter lists the result of each step in this project. An explanation of the results is given to make the presentation of results more readable. An overview of the results is given in the following:

- Performance comparison of network using the three test sets.
- Results of pointwise difference between facial expressions.
- Results of face asymmetry.
- Smile trajectory in principal component (PC) space. The trajectories before versus after smoothing are shown. The similarity of trajectories of different individuals is discussed.
- A survey of the test set based on PCA.

To make the expression simpler in the figure of this chapter, we named the facial expression in a short name. The neutral face that before the smile is neutral 1. The neutral face that after the smile is neutral 2. The maximum smiling face is smile.

6.1 The performance of network

Three test sets are used for the testing performance of networks (Refer to section 5.4). All of the results shown in this section relate to the performance test. The labels in the figures of this section are name of the network that are explained in table 5.2.

6.1.1 Result of test set 1

As seen in figure 6.1, the error of predicted results decreases with the increasing size of the training set in the first three networks. This result is expected because the performance of network will increase according to the increasing size of the training set.

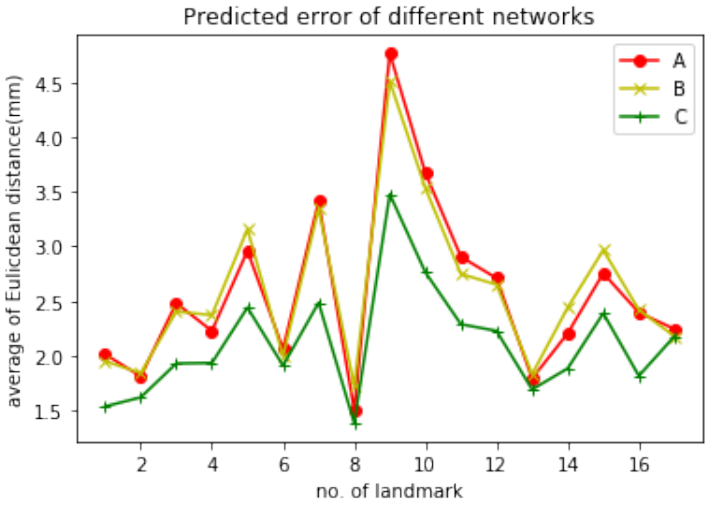


Figure 6.1: The landmark localization error with the test set 1 in different networks

6.1.2 Result of test set 2

Figure 6.2 shows the error of smile with the intensity of 2 in different networks. The performance of network E is best while the rest networks show similar performance. This result is expected because the training sets of the first three networks didn't include the smile with the intensity of 2.

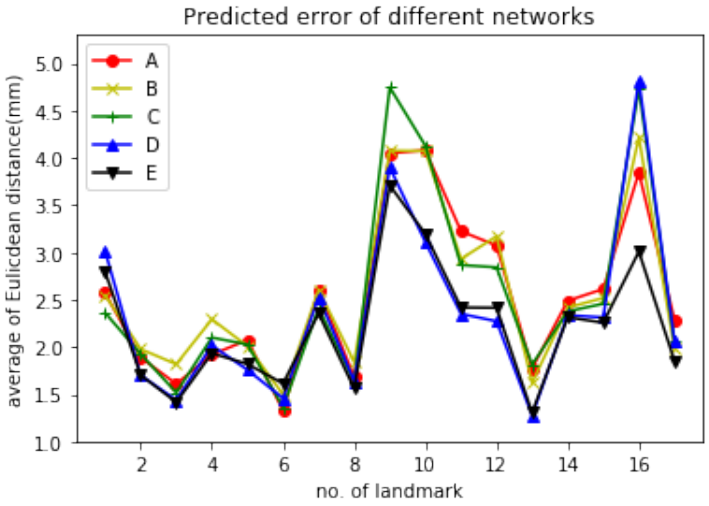


Figure 6.2: The landmark localization error with the test set 2 in different networks

6.1.3 Result of test set 3

In figure 6.3, the five networks show similar performance. Comparing to figure 6.1, the results show a positive correlation between the size of the training set and predicted

accuracy. This figure shows a contradictory result when comparing to the result shown in figure 6.1. The possible reason for this contradictory result is provided in the next section.

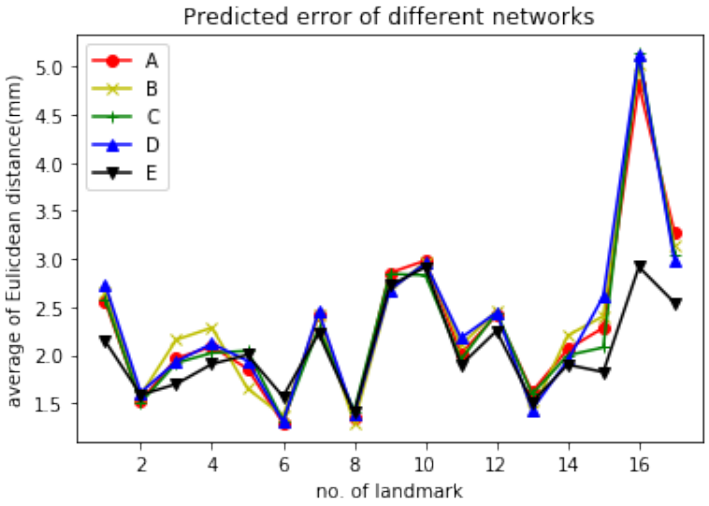


Figure 6.3: The landmark localization error with the test set 3 in different networks

6.1.4 Variation of intra-observer

Figure 6.4 shows the result of an intra-observer duplicate landmarking test. Landmarks were annotated twice by the same operator (observer) and the distance between the two landmarks was calculated for each landmark. As seen in the figure, the maximum distance reaches 1 mm. This distance results in a covariate shift. Covariate shift refers to the change in the distribution of the input variables present in the training and the test data [19].

An example result for one landmark (chin) is shown in figure 6.5. In the figure, the distance between landmark positions at first and second landmarking is plotted against subject (scan) number. Resules for other landmarks are shown in appendix C.

6.1.5 Robustness of the selected network

Although the covariate shift influences the test of network performance, the training process is not affected. Because none of the data that were labeled in the second landmarking round were included in the training set. We still can trust the predicted accuracy of the network. Network E is selected as the best for carrying out automatic placement of landmarks because its performance is better than others even in the presence of the manual landmarking error.

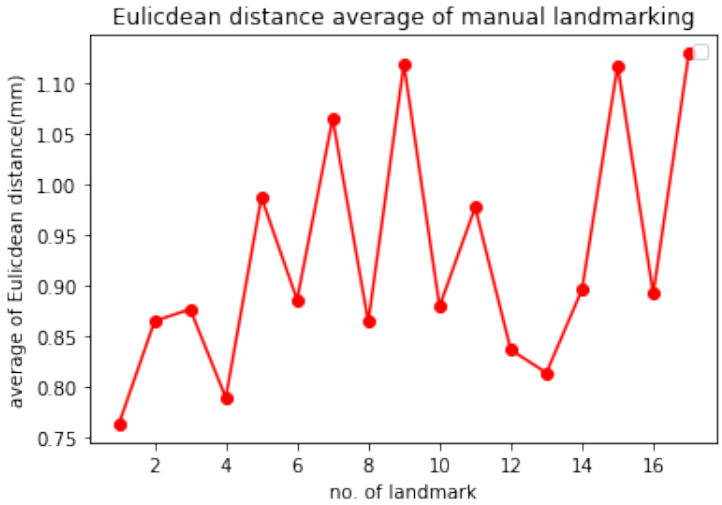


Figure 6.4: The mean distance between landmarks placed twice by the same operator.

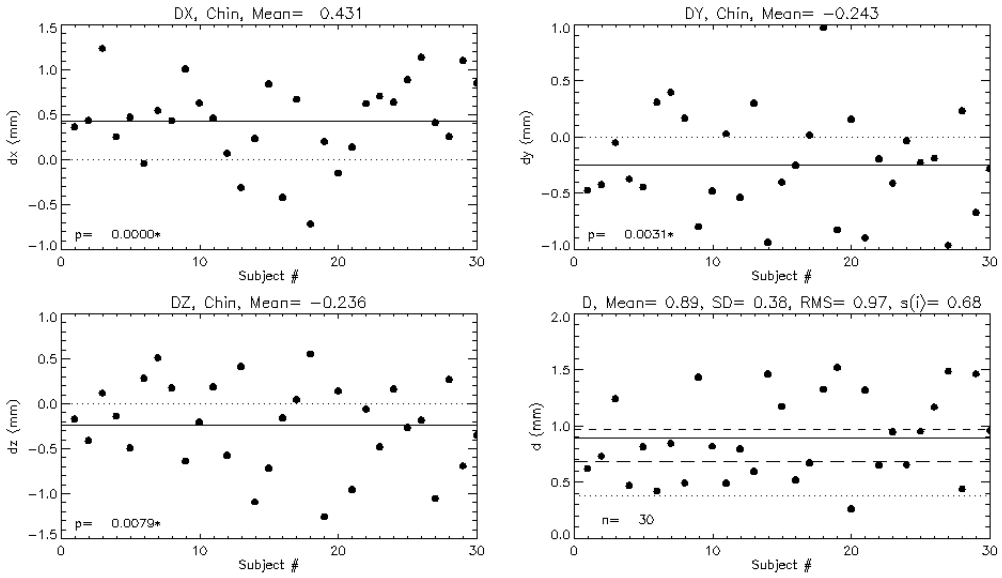


Figure 6.5: The distance (mm) between first and second manual landmarking session for the chin landmark. The Euclidian distance D is shown in the lower right plot, while the Cartesian components (DX, DY, DZ) are plotted in the other three plots, as indicated. In the figure, the SD is standard deviation. The RMS is root mean square. Mean is the mean error. The i in s(i) is the number of the landmark in the figure.

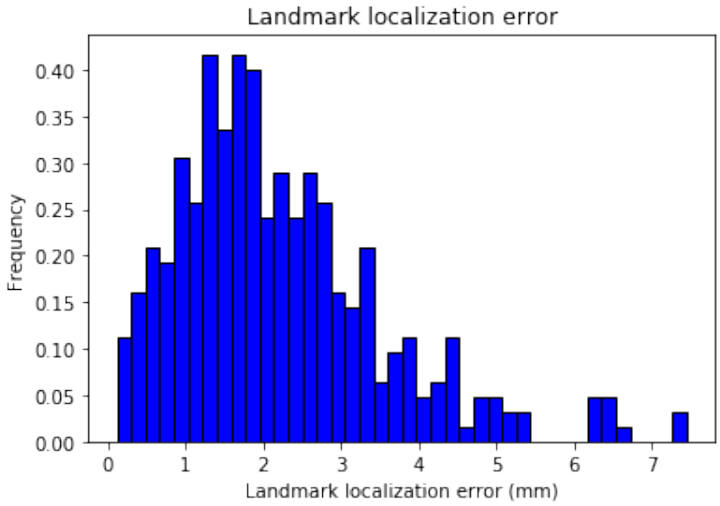


Figure 6.6: The predicted error histogram of selected networks. The Y-axis represents the frequency. The metric of the x-axis is millimeter.

Distribution of predicted error was calculated in order to investigate errors further, and this is shown in figure 6.6. Most of the errors are within 3 mm. There are few errors larger than 5 mm.

The face with maximum predicted error on each landmark shows in appendix C.

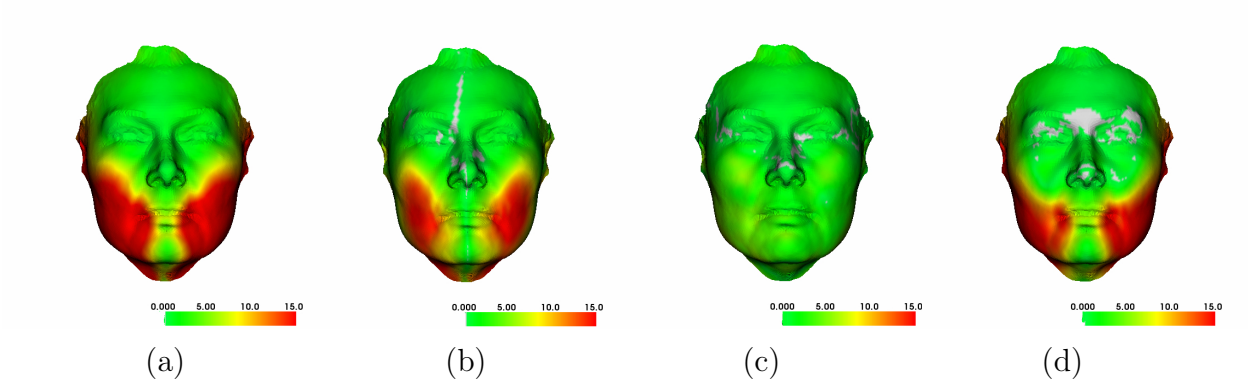
6.2 The pointwise difference between facial expressions

In this section, the pointwise difference between faces are shown as color-coded surface.

Figure 6.7 shows the pointwise difference between neutral 1 (the neutral face before smile sequence starts) and maximum smile. Figure 6.7 (a) is the overall pointwise difference. The red area is the area that the face muscle moves in the process of facial expression change from neutral 1 to maximum smile. Figure 6.7 (b), (c) and (d) show more details about the movement of face muscle in this process. Figure 6.7 (b) indicates the protrusion of the cheek. Figure 6.7 (d) indicates that the mouth corner moves up. Figure 6.7 (c) indicates that the face muscle didn't have any movement in the horizontal direction.

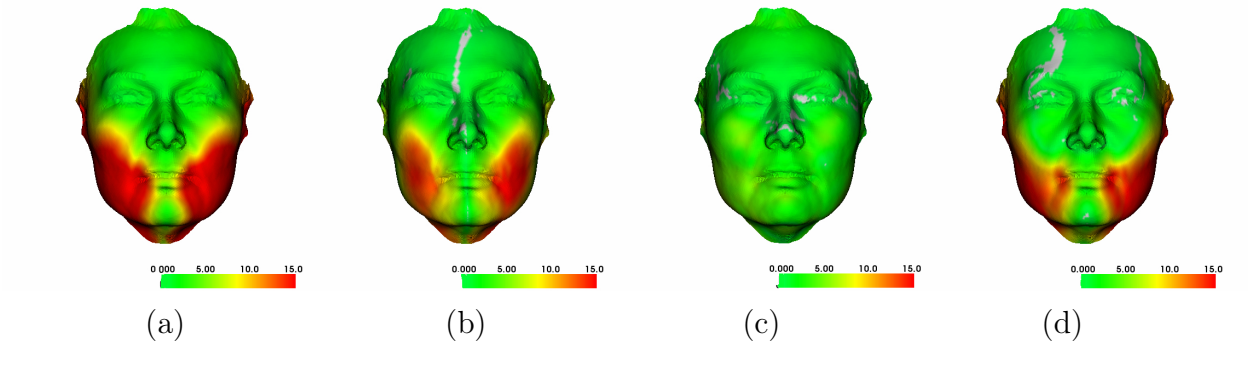
Figure 6.8 shows the pointwise difference between smile and neutral 2 (neutral face after the smile sequence). As seen in the figure, the movement of face muscle is almost the same as in figure 6.7 which is a reasonable result. People will move their muscles on the face during a smile. When a smile finishes, the moved muscle will go back to the original area. In addition to this, smile expression affects the cheek and mouth corner. Besides, the smile has almost no effect on the forehead and the area around the eyes.

Figure 6.9 shows the pointwise difference between neutral 1 and neutral 2. In the figure, it maybe seen that there is a difference between the two neutral faces. It indicates that



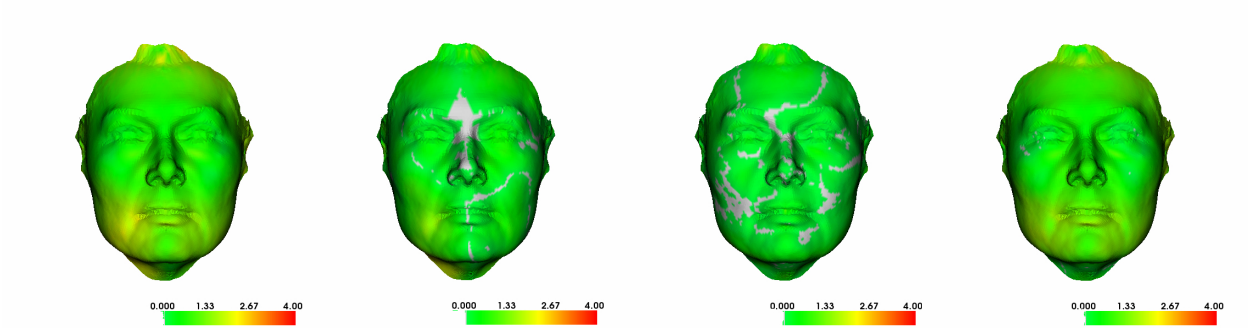
(a) (b) (c) (d)

Figure 6.7: Face surface color-coded according to the difference between neutral 1 and maximum smile in an example subject. (a) The overall pointwise difference. (b) The pointwise difference along the x-axis (transverse direction). (c) The pointwise difference along the y-axis (vertical direction). (d) The pointwise difference along the z-axis (sagittal direction). The face shown is a mean neutral face as acquired before the smile sequence started. The grey area on the face means that the difference in this area is close to zero.



(a) (b) (c) (d)

Figure 6.8: Face surface color-coded according to the difference between maximum smile and neutral 2 and in an example subject. (a) The overall pointwise difference. (b) The pointwise difference along the x-axis (transverse direction). (c) The pointwise difference along the y-axis (vertical direction). (d) The pointwise difference along the z-axis (sagittal direction). The face shown is a mean neutral face as acquired before the smile sequence started.



(a) (b) (c) (d)

Figure 6.9: Face surface color-coded according to the difference between neutral 1 and neutral 2 in an example subject. (a) The overall pointwise difference. (b) The pointwise difference along the x-axis (transverse direction). (c) The pointwise difference along the y-axis (vertical direction). (d) The pointwise difference along the z-axis (sagittal direction). The face shown is a mean neutral face as acquired before the smile sequence started.

facial expression before the smile and after the smile is different even for the same person.

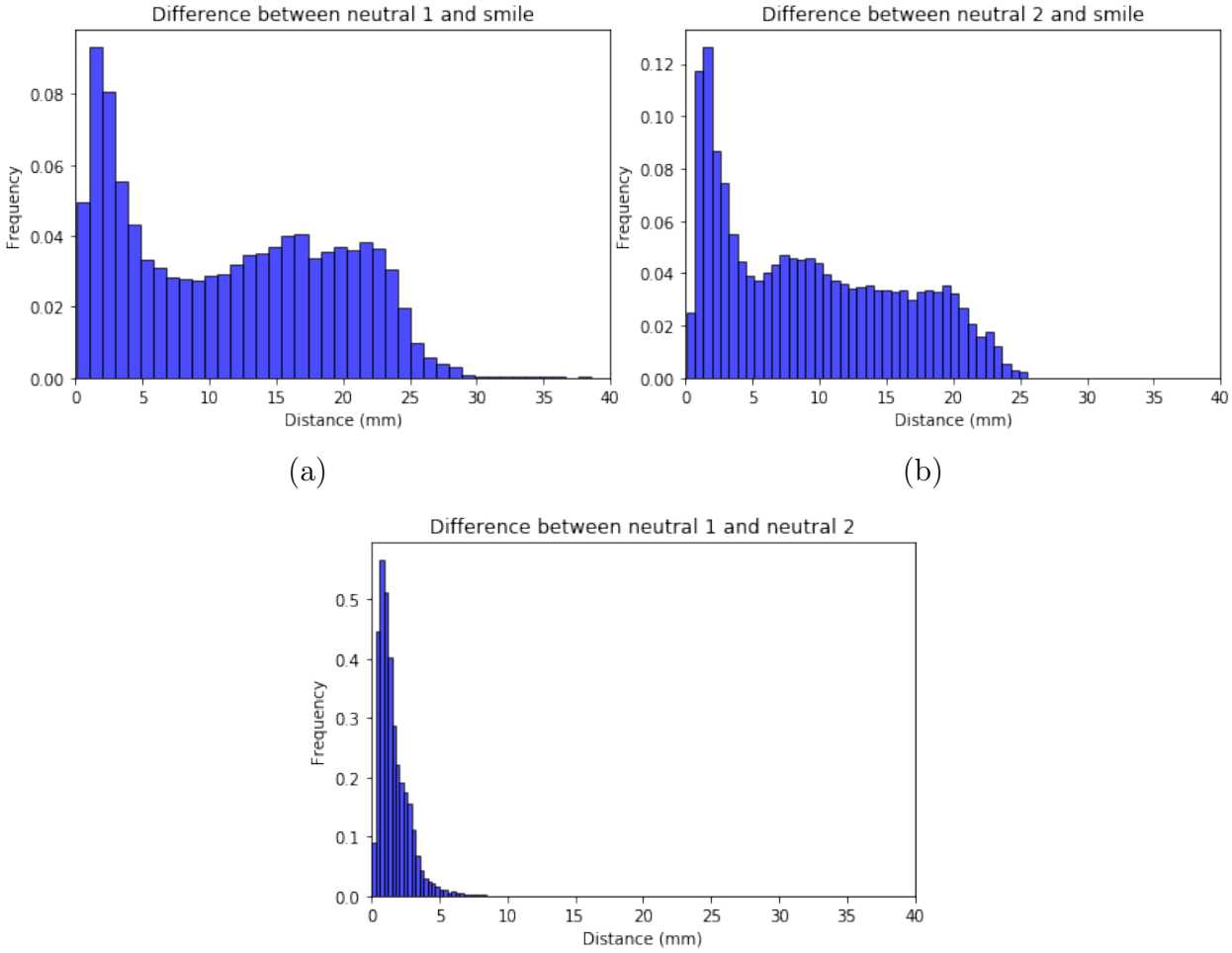
More results show in appendix D.

Figure 6.10 shows histograms of the pointwise difference in the dynamic data. As seen in the figure, pointwise difference between a smile and the two neutral expressions, respectively, shows a similar distribution. Differences between the two neutral faces are very small. Most of the differences are within $5mm$. This result is not contradictory to the results shown by the color-coded surface. The difference between a smile and a neutral face is larger than the difference of a neutral face. The difference between a smile and a neutral face is larger than the difference between neutral faces.

The highest bin in three histograms are all within 5 mm. The frequency of the highest bin in the histogram 6.10 (a) and (b) are both around 0.1. This result evidences a truth that a part of the face still has a slight movement. e.g. the forehead and area around the eyes. When an individual smiles, the displacement of some muscles reaches more than 25 mm. In (c), the frequency of the highest bin is larger than 0.5. This result indicates that half of the differences between the two neutral faces are less than 5 mm. Although there are differences between the two neutral faces, the differences are very small.

6.3 Asymmetric smile

Figure 6.11 shows how the facial asymmetry of an individual changes with time during a smile sequence. The amount and localization of asymmetry are presented in the figure. The first and last frame is the neutral face before smile and after smile, respectively. The face is shown to be asymmetric in the neutral expression both before and after the smile sequence. The spatial distribution of asymmetry is similar before and after the smile, but



(a) (b)

(c)

Figure 6.10: The difference histogram of whole dynamic data set. (a) The difference between neutral 1 and smile. (b) The difference between neutral 2 and smile. (c) The difference between neutral 1 and neutral 2.

the amount of asymmetry is slightly smaller after the smile. More results are shown in Appendix D.

It can also be seen that, in the example subject shown, the amount of asymmetry increases with the extent (strength) of the smile. This result means that the smile is not symmetrical. In the figure, the asymmetry of a smile is larger than the asymmetry of the neutral faces. There are two possible reasons. The first is that the asymmetry of the smiling face is large caused by an asymmetric motion during the smile process. The second reason is the additive effect of an asymmetric neutral face and a smile process that magnifies the asymmetry that was present in the neutral face.

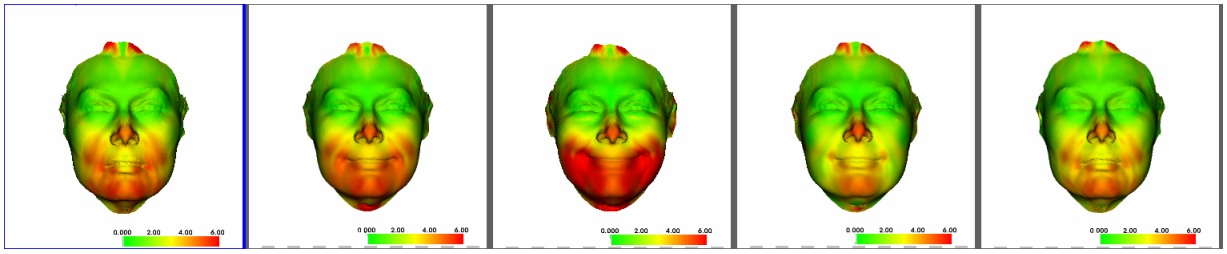


Figure 6.11: Asymmetric values change with time

Figure 6.12 shows the trajectories of the landmark on the mouth corner in physical space. The blue curve represents the trajectory of the landmark of the mouth corner during motion from a neutral face to a smiling face. The white curve represents the trajectory during the opposite motion. The red point labels the starting point of the landmark on the mouth corner. It is seen that the two mouth corner move up with different trajectories. This result shows that the individual smile is asymmetric. In addition, the trajectories of two mouth corners are not smooth. Mouth corners did not move with a smooth trajectory during a smile. The measured trajectory of the mouth corner is like an oscillating curve.

Figure 6.13 shows the histogram of asymmetry values in the data set. The asymmetric values in three facial expressions (neutral face before the smile, smiling face, and neutral face after smile) have a similar distribution. For a neutral face, most of the asymmetric values are within 5 mm. For a smiling face, most of the asymmetric values are below 10 mm. The result is reasonable because the smile is asymmetric. The additive effect of asymmetry on neutral and smile results in a larger asymmetry on the smiling face.

This "synergy of asymmetry" may be separated to analyze the asymmetry of the smile. The asymmetry of neutral face is removed from the smiling face. As seen in figure 6.14, more than 60 % of asymmetric values close to zero. Besides, almost all asymmetric values are within 5 mm. This result indicates that the amount of asymmetry of the smile is smaller than the amount of asymmetry of the neutral face.

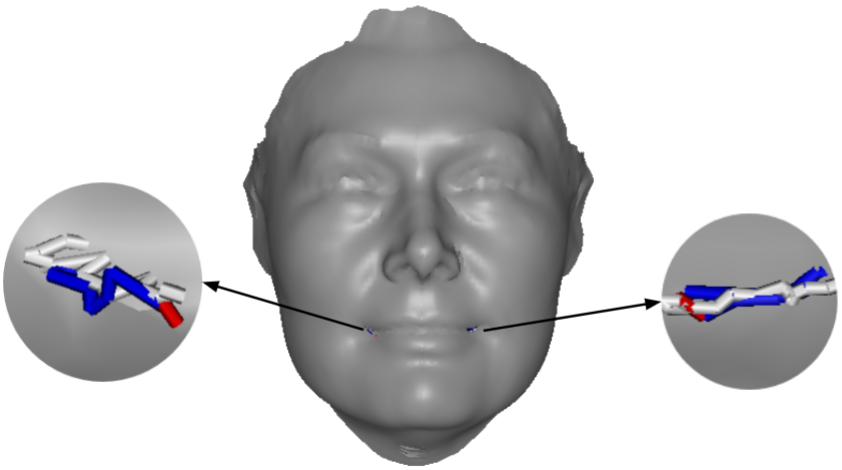


Figure 6.12: Trajectories of mouth corner during a smile. The landmark locates the mouth corner.

6.4 Result of PCA

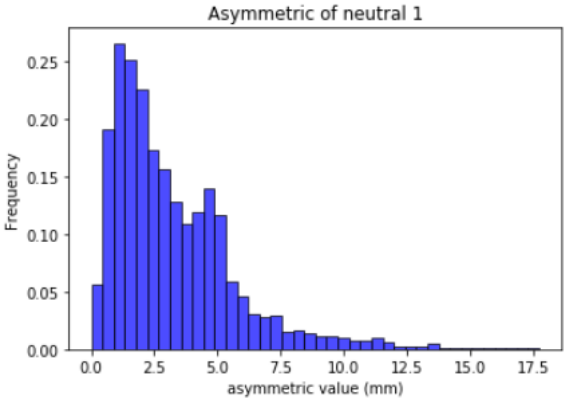
6.4.1 Examining the model of smile sequence

All the eigenvalues of the covariance matrix are shown in figure 6.15. The first five eigenvalues are significantly higher than others while the rest of the eigenvalues are almost equal. The significant eigenvalues of the covariance matrix are shown in table 7.1. The ratio of the sum of the first three eigenvalues is 88.38%, which is large enough to describe the principal components. In addition, 3-dimension is the largest number of dimensions that humans can visualize. Consequently we use first three principal components to analyze a smile.

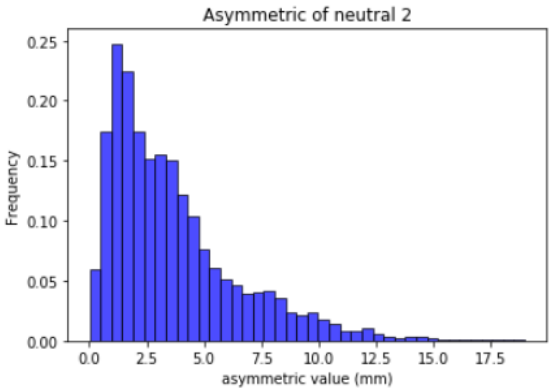
Figure 6.16 shows the correlation of each principal component with times. As we recall from section 5.1.3, 100 frames are used for representing the process of a smile. The y-axis in the figure is the frame number.

The curves in (a) and (b) have similar trends. In the first twenty frames, the value of PC stands at a similar value for each individual. However, over the following twenty frames, there is a sharp fall. The value of PC keeps a certain value during the next twenty frames. The value of PC then rises to the original value which at the beginning of a curve. It remains stable for the following frames. The two curves represent the process of smile. First twenty frames are the neutral face, then the facial expression changes from neutral face to maximum smile. The facial expression changes back to neutral face from the sixtieth frame.

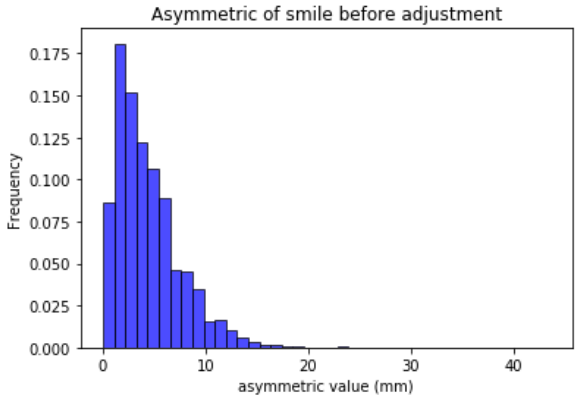
According to table 7.1, the eigenvalue of the third principal component occupies 20 % of



(a)



(b)



(c)

Figure 6.13: The asymmetric histogram of the whole dynamic data set. (a) Asymmetric value of neutral 1. (b) Asymmetric value of neutral 2. (c) Asymmetric value of smile with additive effect.

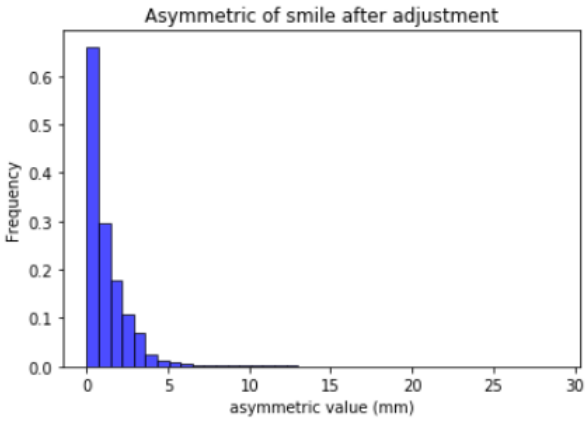


Figure 6.14: The asymmetric value of smile without additive effect.

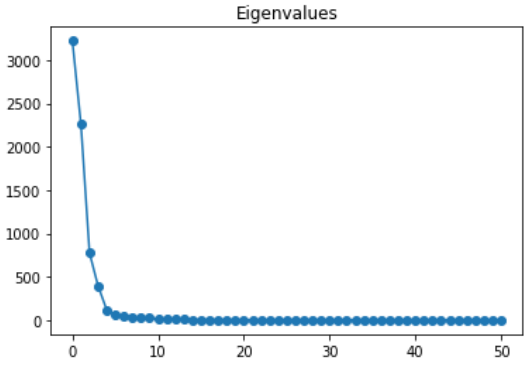
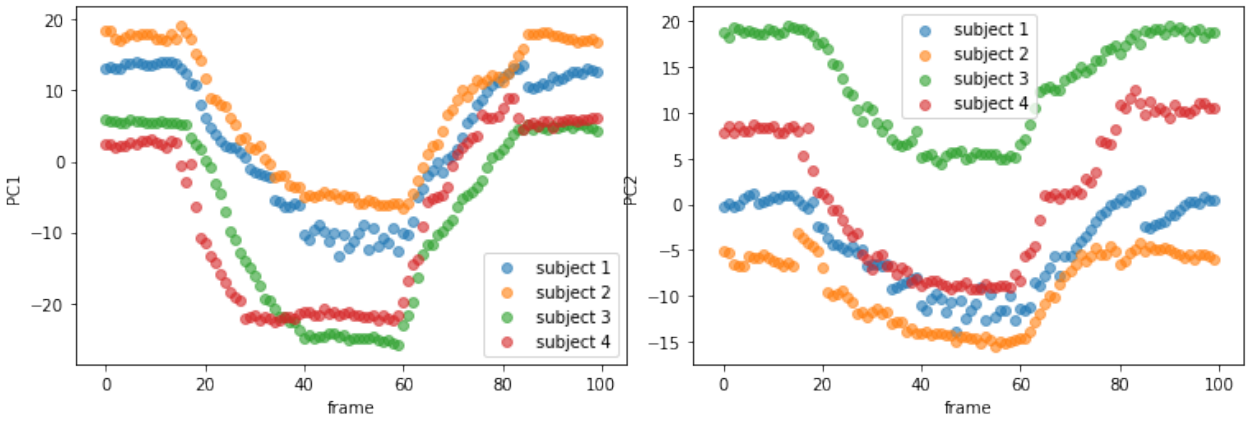


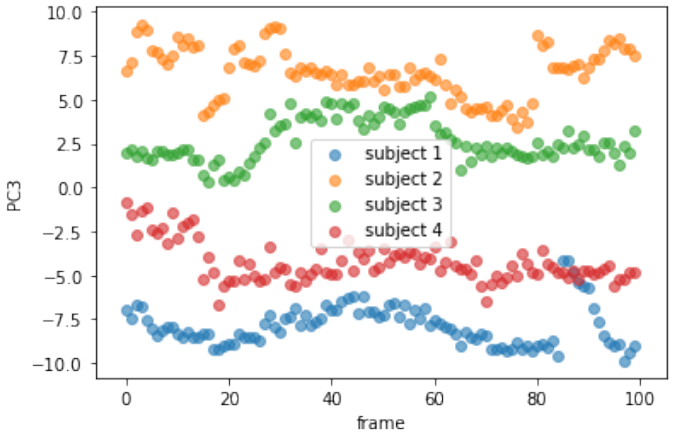
Figure 6.15: Eigenvalues

Table 6.1: Eigenvalues of covariance matrix drived from the dynamic data set.

Eigenvalue	$\frac{\lambda_i}{\lambda_T} \times 100\%$
λ_1	45.54%
λ_2	31.89%
λ_3	10.95%
λ_4	5.55%
λ_5	1.69%



(a) (b)



(c)

Figure 6.16: Principal component plotted against the time of each person. (a) The first principal component plotted against the frames. (b) The second principal component plotted against the frames. (c) The third principal component plotted against the frames.

all eigenvalues. It is an important impact factor of a smile. However, the third principal component almost has no correlation between time and smile in (c). The first frame to the last frame shows a fluctuation in the value of PC3. To show the impact of each principal component on a smile more intuitively, we generate the new face by varying the first three parameters of b in equation 3.6 separately. Each parameter in b represents a mode variation of shape and can illustrate how each principal component affects the smile. We connect the landmarks in the figure to visualize the synthesized face more easily.

Figure 6.17 illustrates the varying face with the change of first principal component. As seen in the figure, the facial expression changes from an smiling face to a sad face (mouth corner moves down, which is the common facial expression of sad). Refer to figure 6.16 (a), the value of PC1 of smiling face is lower than the neutral face. So figure 6.17 shows a consistent result.

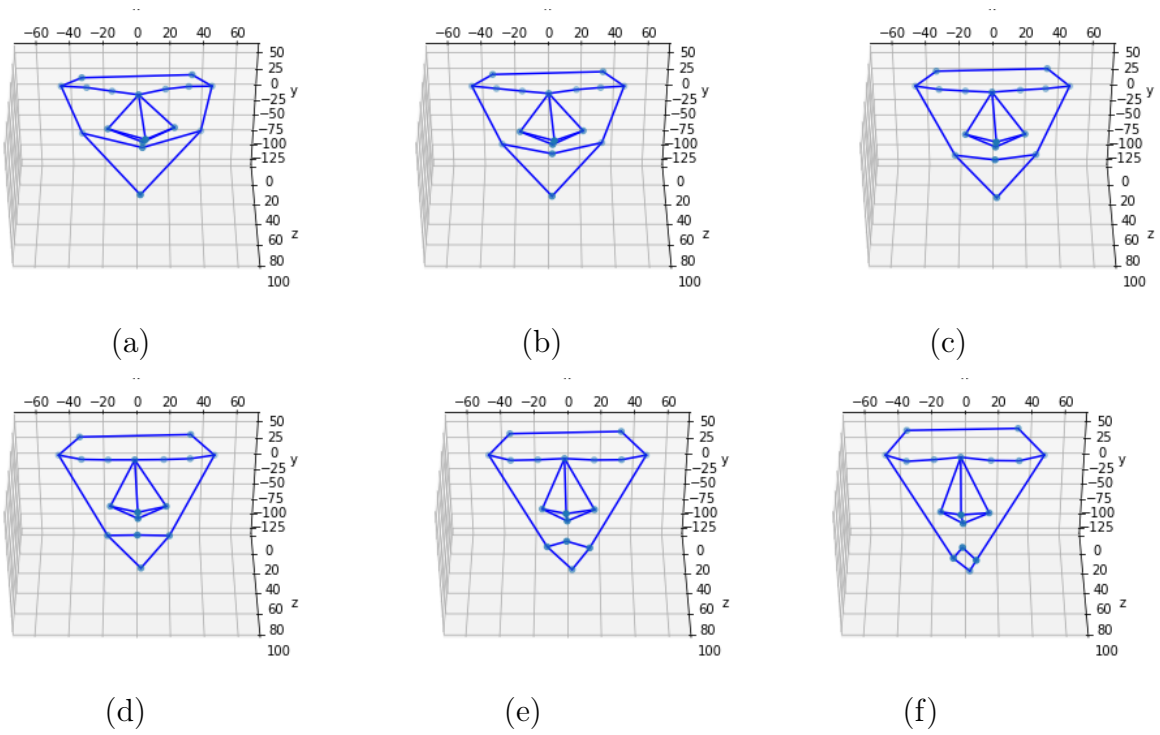


Figure 6.17: First mode with the parameter changes from - $\sqrt{\lambda_1}$ to $\sqrt{\lambda_1}$ (from (a) to (f)).

Figure 6.18 shows the varying face with the change of the second principal component. As similar to figure 6.17, the mouth corner moves from up to down. The facial expression changes from a smiley face to a sad face with the increasing value of PC2. The facial expression changes from a smiling face to a sad face with the increasing value of PC2 (6.16 (b)). Except for the mouth corners, the position of the nose has movement in the figure. This result indicates that the second principal component also has a correlation with the position of the nose.

Figure 6.19 illustrates the varying face with the change of the third principal compo-

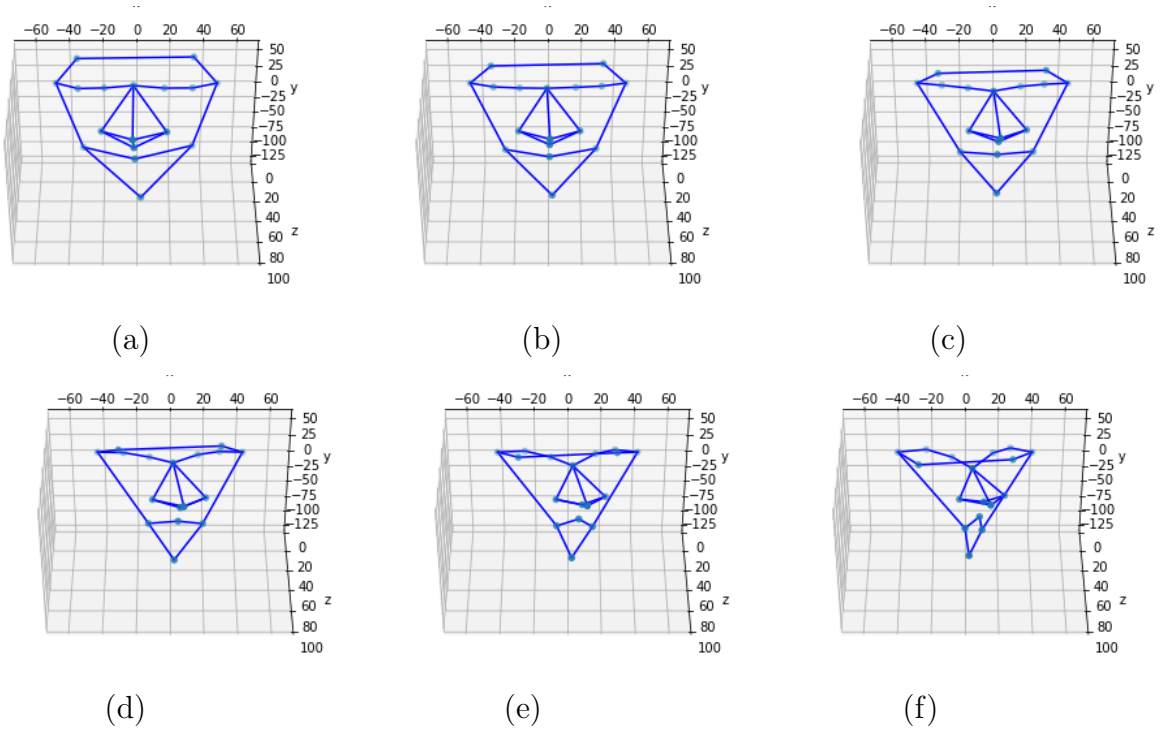


Figure 6.18: Second mode with the parameter changes from -$\sqrt{\lambda_2}$ to $\sqrt{\lambda_2}$ (from (a) to(f)).

nent. Two mouth corners almost have no movement. This result indicates that the third principal component has a weak correlation with the smile. Meanwhile, the third principal component correlates with the size of a face. The changes are not as clear as the nose changes with the second parameter, but it exists.

As seen in the above figures, the synthesized faces contain the size information of the face and the direction of the face. To further investigate it, landmarks that changes with each principal component are plotted together in figure 6.20. The red and green points represent the landmark in the corners of mouth. In 6.20 (a), the eyebrows have a large variation while the position of eyes has a slight variation. In figure 6.20(b), the movement of the mouth corner is more horizontally compared with figure (a). Other landmarks has larger movement than figure (a). In figure (c), the movement of landmarks is small than figure (a) and figure (b). There may be two reasons why the landmarks except to mouth corners also have a large variation. Some people move facial features during a smile. For example, the position of the eyebrow will move upwards when someone smiles. Another possible reason is that there still are some differences in size and orientation in the input data even though an alignment of the scans was carried out.

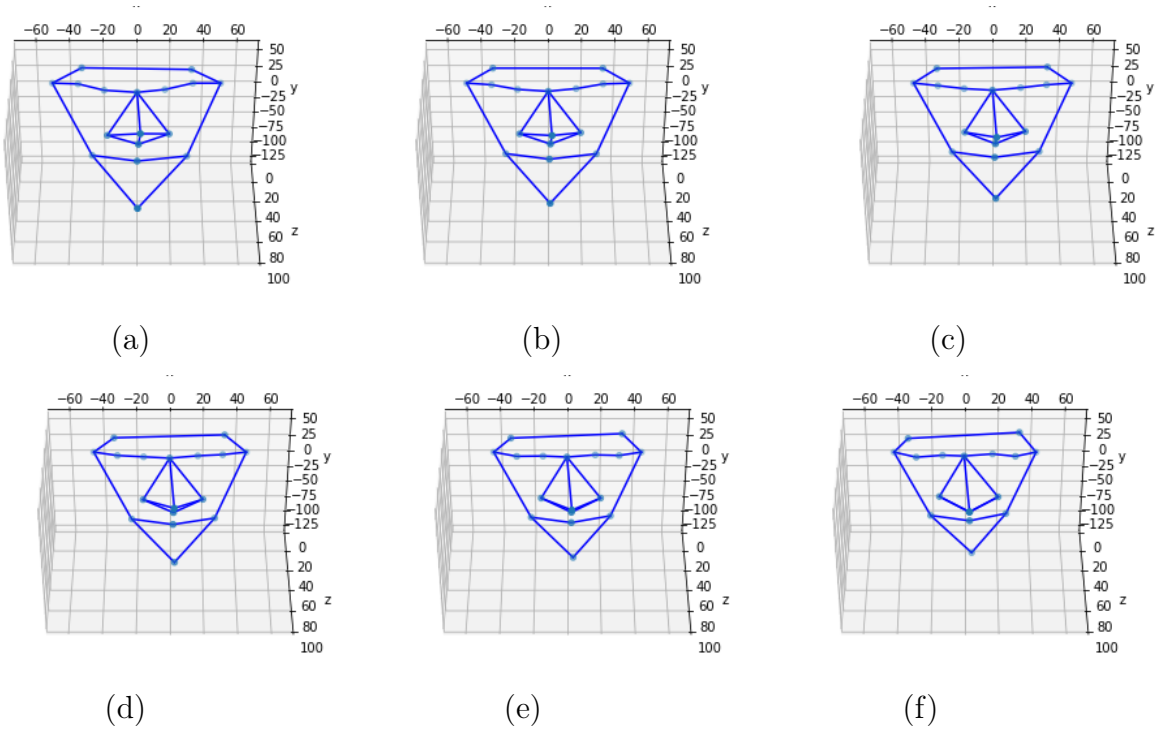


Figure 6.19: Third mode with the parameter changes from - $\sqrt{\lambda_3}$ to + $\sqrt{\lambda_3}$ (from (a) to(f)).

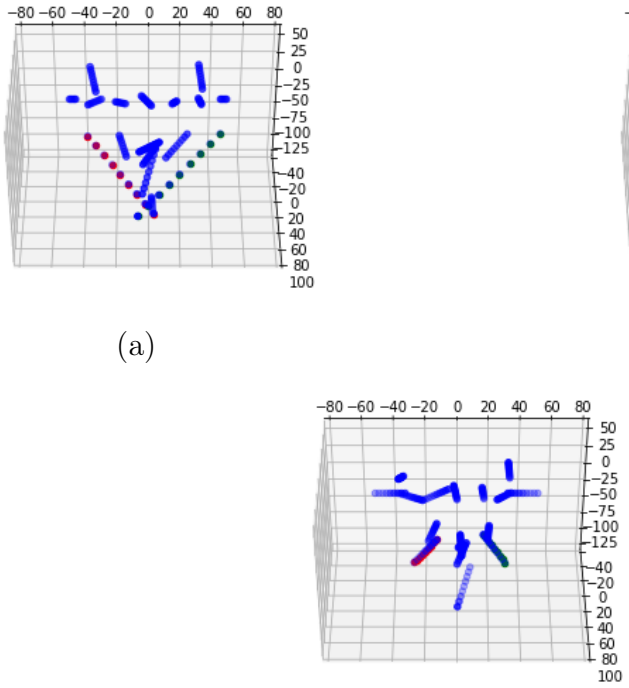
6.4.2 Smile trajectory in PC space

Figure 6.21 (a) presents the data which is reduced to the 3D in the principal component space. Each point represent a face. The points are connected according to the time to get the smile trajectory in the PC space. Figure 6.21 (b) shows the average smile trajectories of each person in the PC space and the mean smile trajectory of the dynamic data set. In the figure, the subject 1 to 4 means the average smile trajectory of each individual, the mean trajectory means the average trajectory of all of these subjects. As seen in the figure, smile trajectories fluctuate up and down in PC space but go in the same direction. Much of the fluctuation of trajectories is believed to be due to noise.

6.4.3 Smile trajectory after smoothing

Figure 6.22 shows the smoothed trajectories.

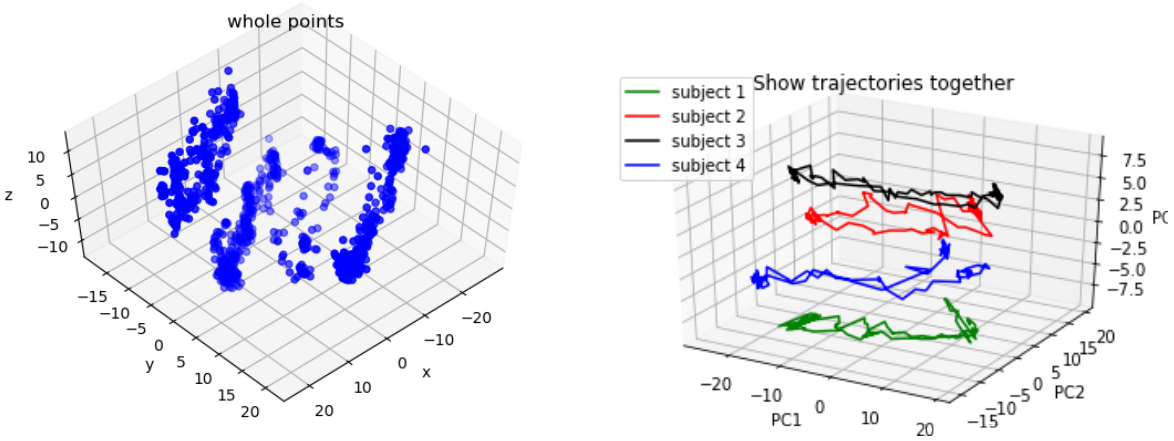
As it can be clearly seen in the figure, trajectories of different individuals start at a different point, then go in the same direction but arrive at different points. Finally, the trajectories end at a location that is close, but not identical to, the start point. These trajectories describe the process of smile clearly: the facial expression is the neutral face at the beginning, then the facial expression arises from a slight smile to the maximum smile. Finally, facial expression changes from a maximum smile back to a neutral face



(a) (b)

(c)

Figure 6.20: The face variation with each principal component. (a) Mode 1. (b) Mode 2. (c) Mode 3.



(a) (b)

Figure 6.21: (a) Data that after reducing dimension. (b) Smile trajectories in PC space.

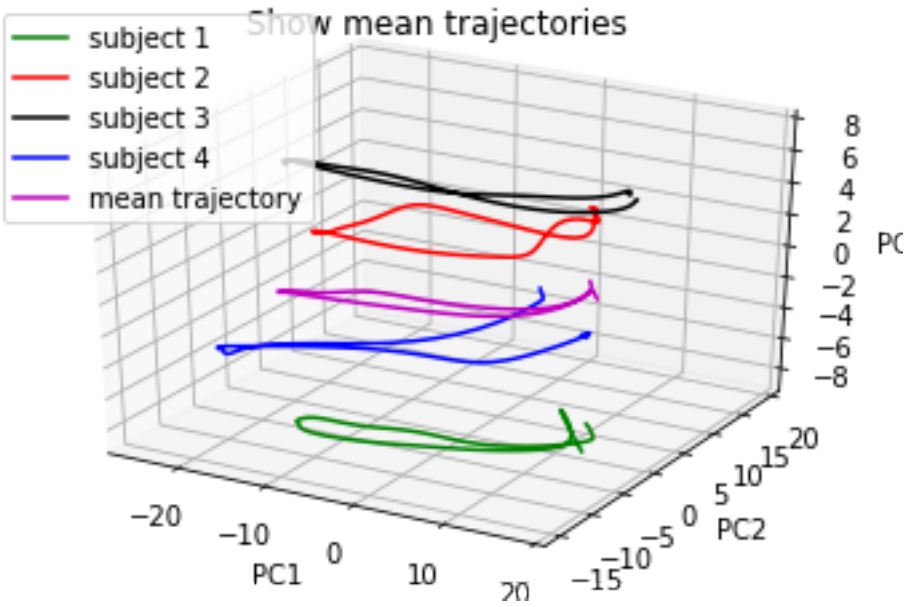


Figure 6.22: Smile trajectories after smoothing in PC space.

again. This result shows that the smile of a population has commonality.

A sequence of synthetic smiles may be generated using the mean trajectory. It may be used to validate the correctness of the mean trajectory. Figure 6.23 shows the smile sequence that is reconstructed from the mean smile trajectory in PC space. The point in the figure means that the face is synthesized from this point in the principal component (PC) space.

6.5 A supplementary survey: test sets in PC space

In this section we explore properties of the input BU-3DFE dataset. Recall from Section 5.3 that three test sets are used to test the performance of the network. To survey the facial expressions in the test set, all faces in the three test sets are analyzed using PCA. The landmarker [1] computes PCA of test sets using dense surface. The distributions of the test sets in the PC space shows the difference between test sets.

The results show on figure 6.24. The blue points and yellow points are fully separated, which means the neutral face is different from the maximum smile. The position of green points mostly is between the blue points and yellow points. It makes sense because of the smile intensity of green points is smaller than the yellow points while larger than blue points. The red points and yellow points are not very well separated. So the smile with the intensity of 3 and the smile with the intensity 4 are not very different. Thus, the smiling face of the intensity 3 can be used to test the predicted accuracy of a maximum smile (smiling face of the intensity 4).

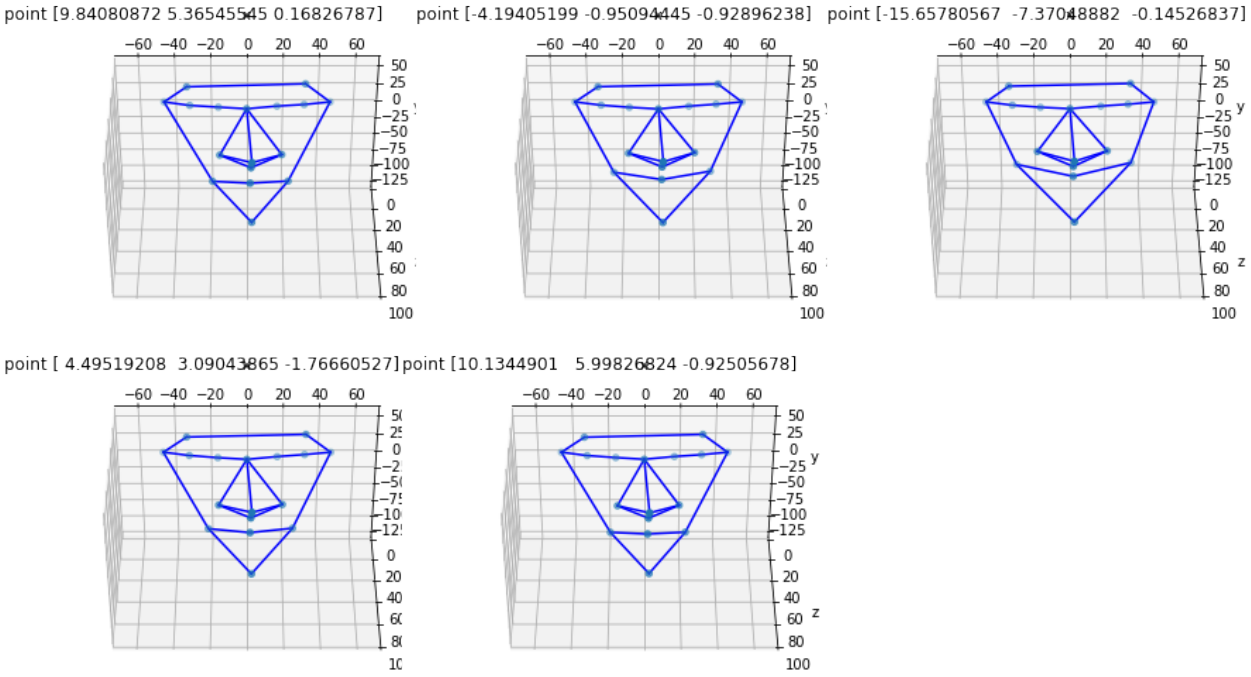


Figure 6.23: A smile sequence generated by the mean trajectory

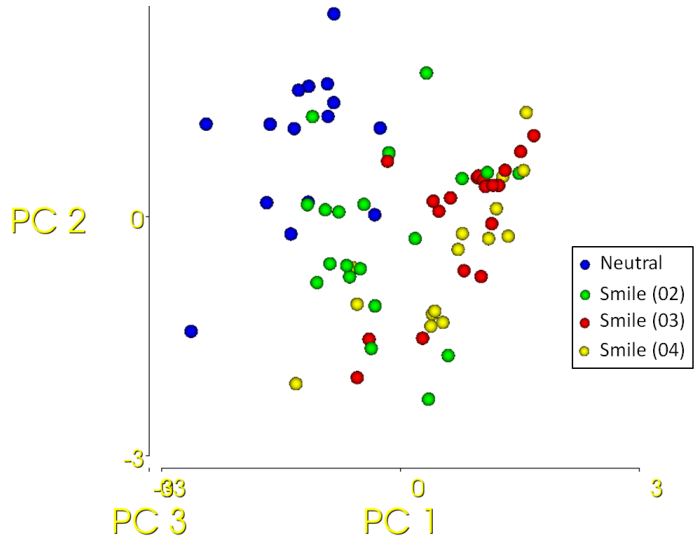


Figure 6.24: Faces of three test sets in the PC space. The smile(02) is the smiling face with the intensity of 2, the smile(03) is the smiling face with the intensity of 3, and so on.

As seen in the figure, some points with the different colors are overlapping a lot. To investigate the reason for this, figure 6.25 illustrates a few of those surfaces that are located in unexpected places in the PC space. A smile is not necessarily a smile in the test data set. This is an imperfection of the input data set that could be expected to contribute to lessening the sensitivity of the method in terms of discerning between different strengths of smiles.

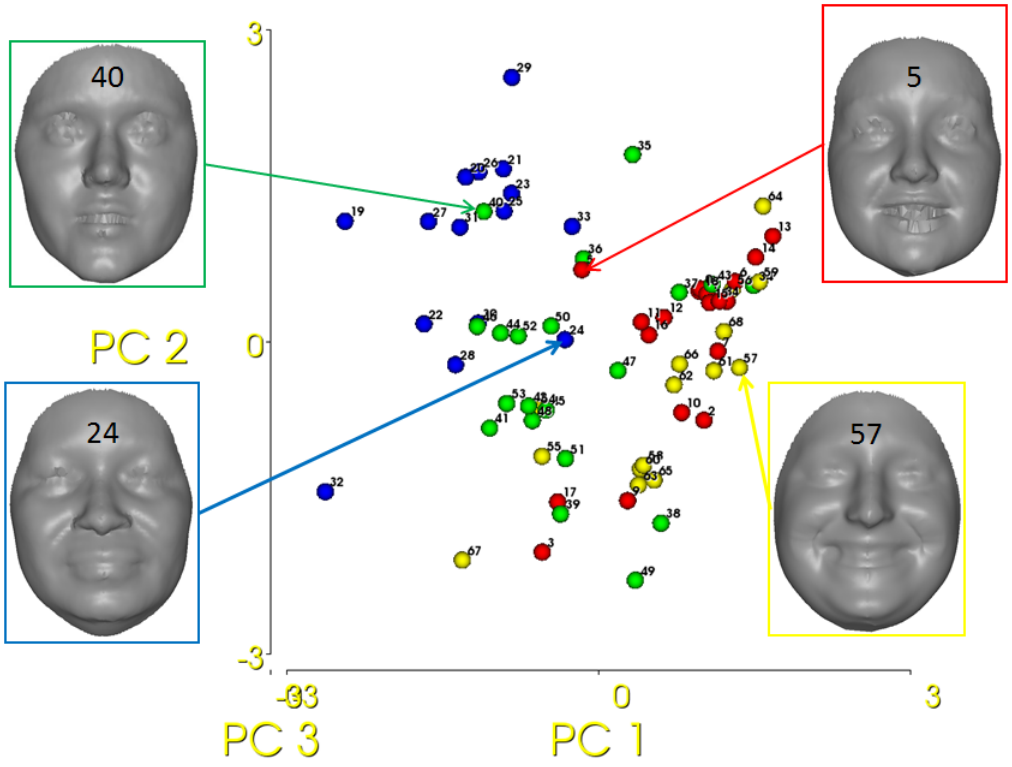


Figure 6.25: Surfaces that are located in unexpected locations in PC space.

Chapter 7

Discussion

This chapter gives a general discussion about the result.

7.1 Accuracy of CNN

The performance of CNN has a significant effect on the smile analysis. CNN predicts the position of landmarks based on the ground truth. Consequently, the accuracy of manual annotation therefore has an important influence on the auto-landmarking result.

Theoretically, the size of the training set has a positive correlation with the performance of CNN. But in reality, we need to consider the errors. If the same person labels the ground truth twice, the distribution of data will be changed due to the varies intra-observer (seen figure 6.5). A solution of it is that mixes all of the training set together and sets the mixed training set as the input of training. In this project, the distribution of ground truth is different from the training set and test set. So the test result is influenced by this. Therefore, it's better to annotate the ground truth of the data set one time and use this ground truth train and test the network.

It is a fact that the error of annotated ground truth cannot be eliminated. The distribution of the landmark also differs between different landmarking sessions. A comparison between automatic and manual landmark placement was done in two cases of juvenile idiopathic arthritis. Both individuals had been diagnosed with arthritis. One had the affection of one of the mandibular joints (unilateral affection), while the other had no affection of the mandibular joints (no affection). The landmark were placed by clinicians. The comparison results are shown in appendix F.

Some landmarks always show worse performance than others (seen the figures in 6.2). These landmarks are on the chin and eyebrow. The landmark that is hard to put manually has a higher error than landmarks that has strong anatomical cues. This result is consistent with the inference of Paulsen.

7.2 Computation time

The GPU card using in this thesis is NVIDIA Quadro P5200. All of the calculation are based on this GPU. The calculation in this thesis contains network training and evaluation, the computation of dense surface model, and analysis of results.

Table 7.1: Computation times of each procedure.

Procedure	Time (s)
Rendering in the data pre-processing (per surface)	10
Training time (per batch)	1
Prediction time (per surface)	15
Building the dense point corresponding between surfaces (per surface)	25
PCA (400 surfaces)	≈ 0
synthesizing face (per face)	≈ 0

Chapter 8

Conclusion and further work

8.1 Conclusion

In this thesis a method for analysis of human smiles as recorded in 4D with high temporal resolution has been developed and tested on smile sequences from healthy volunteers. The method has been seen to be fully automatic and fast and is based on automatic landmark identification using convolutional neural networks (CNNs) on triangular mesh representations of faces. The accuracy of the method has been investigated and shows promise, although the accuracy of a human operator carrying out manual landmarks was not reached. However, it was demonstrated that the method was able to successfully quantify facial asymmetry in individuals with juvenile idiopathic arthritis with very little discrepancy from the result obtained by manual landmarking by the clinician. Asymmetry calculations are less sensitive to landmark identification errors provided that the landmarking errors are symmetrical in the face. Several future improvements are proposed, see below.

Apart from demonstrating the performance of automatic landmark placement in smile sequences, new methods of analyzing properties of a smile and a population of smiles were introduced in this thesis. The methods were based either on the dense surfaces created by a method of building detailed point correspondence by atlas deformation, or on the sparse set of landmarks that were directly identified by the CNN.

The former method provided means of quantification of amount and direction of spatially dense shape change between two given time points in a smile sequence. Furthermore, it provided a measure of the facial asymmetry at given time points and hence the spatial and temporal development of the asymmetry of a smile could be quantified. It was demonstrated that smiles were not completely symmetric in the normal individuals included in the study. The asymmetry was seen to increase with the strength of the smile; the possible reason being that an asymmetry in the neutral face was magnified during the smile process. Also, the individuals were not able to return to the same neutral face after the smile process.

The latter method was particularly suited for a principal components analysis (PCA) of the sparse set of landmarks representing the smile sequence. It was demonstrated that smiles form loops in PC space and that the direction in PC space of going from a neutral towards a smiling face was very similar for all the subjects included in the study. It was also demonstrated that the first PC was strongly correlated to the amount of smile. The second PC was related to differences in facial morphology between the subjects. Noise in the smile trajectories was dealt with by smoothing in PC space. By averaging temporally normalized smile trajectories in PC space it was possible to create a mean smile trajectory. The mean smile trajectory could serve as a reference smile trajectory for the analysis of smile trajectories from non-healthy individuals in the future. Abnormal smiles could be identified and quantified in terms of a distance from the reference smile. A limitation in the present work is the limited number of smile sequences available. More smile sequences should be included in order to obtain a representative description of mean and variation of a normal smile in a population of normal individuals.

The methods presented here are feasible on a higher end laptop computer with a relatively powerful GPU, but can be realized on even smaller hardware. Using more powerful computers would be able to speed up the method considerably.

8.2 Possible researches in the future

The automatic landmark placement was seen to be feasible and closing in on the accuracy of manual landmarking. In order to fully reach the performance of manual landmarking by an expert, it is proposed to include texture information in addition to the geometry information used in this thesis. Another possible improvement would use a better (more representative) training data set, ideally created using the same scanner as the one used in the clinical studies and better selected according to age and gender. Also, the training set could be a dynamic training set instead of the static training (only including a few static instances of the smile) set currently used. It is also possible that including more expressions in the training set (sad, angry, etc.) would improve the performance of the network even on smiling faces.

A possible alternative direction to explore in the future would be to use tracking methods, e.g. based on optical flow, to follow facial features across frames in a time sequence. The starting point of the tracking could be provided by the CNN, as well as possible tracking drift could be corrected by the CNN.

In the future it is proposed to use a better time sampling for the time normalization used in order to temporally match different smile sequences. An interpolation scheme could be used instead of the manual selection of frames as done so far. It is an obvious future improvement to increase the temporal resolution of the frames of the smile sequence that is actually used. Currently 100 frames were used, but this can easily be increased, although an investigation should be carried out determining the necessity of a higher time

resolution. The necessary temporal resolution would depend on the speed of the facial movements we wish to measure. Some movements around the eyes that are important to monitor in some types of facial paralysis are very fast and known to vary on time scales that need the full temporal resolution of the acquisition system (60 frames per second).

Another possible improvement is the result visualization. In this thesis, the reconstructed facial expression is the position of landmarks. It is an intuitive way to visualize the result but could be improved. A more intuitive way to show the result is to reconstruct the dense point face according to the position of landmarks. A possible method to realize it is making a deformation of an average face according to the landmarks.

Finally, more details of the smile could be picked up by adding more landmarks.

Appendix A

Landmark sequence

Table A.1: Landmark Sequence

No.	Name	No.	Name
1	nasion	10	left eyebrow
2	nose tip	11	nose right
3	right eye center	12	nose left
4	left eye center	13	lip middle
5	right eye outer corner	14	right mouthcorner
6	right eye inner corner	15	left mouthcorner
7	left eye outer corner	16	chin
8	left eye inner corner	17	nasolabial point
9	right eyebrow		

Appendix B

Configuration of Deep-MVLM

B.1 Modules installation

Listing B.1 shows all the module needed for running the Deep-MVLM. The environment named pytorch1.2 is created. All of command will run under this environment.

Listing B.1 All the needed module loadings for running the Deep-MVLM.

```
conda create -n pytorch1.2
conda activate pytorch1.2
conda install pytorch torchvision cudatoolkit=10.0 -c pytorch
conda install -c anaconda vtk
conda install -c conda-forge libnetcdf=4.7.1
conda install -c anaconda imageio
conda install -c conda-forge matplotlib
conda install -c anaconda scipy
conda install -c conda-forge tensorboard
conda install -c conda-forge scikit-image
conda install -c anaconda absl-py
```

B.2 Prepare the file for checking input

Listing B.2 shows a function in Jupyter Notebook. This function make an document to check the input file of rendering. This document is necessary to run the rendering program. It need to put in the folder of training set. The input of this function is the path of training set. We can get a text document file that has all of name of objects in the training set.

Listing B.2 Making document for checking the training set.

```
import os
```

```
import shutil
names=[]
file_names = []
dir_names=[]
def get_no_problems_list(file_dir):
    f = open(file_dir+'/'+'BU_3DFE_base_filelist_noproblems.txt', 'w')
    for root, dirs, file in os.walk(file_dir):
        for i in range(len(file)):
            file_name = os.path.splitext(file[i])[0]
            file_exten = os.path.splitext(file[i])[1]
            if  file_name.endswith('RAW') and file_exten.endswith('wrl'):
                file_names.append(file_name)
                f.write(str(file_name[0:5]+'/'+file_name[0:12]+'\n'))
    f.close()
get_no_problems_list("E:/retraining_set/BU_3DFE/")
```

Listing B.3 shows the json file to configure the rendering and training parameter. The module of preparedata contains the path of input file and the path of rendering result. The raw data dir in the module of preparedata is the file location of the raw data set. The processed data dir is the file location of the rendering result. The module of process 3d is to setting the camera position. The data dir in the module of data loader is the file location of the input of CNN. It should have the same location as the rendering result. The training parameters show with its name intuitive in this json file. We can therefore change it easily.

Listing B.3 Configuration of the json file

```
{
    "name": "MVLMModel_BU_3DFE",
    "n_gpu": 1,

    "arch": {
        "type": "MVLMModel",
        "args": {
            "n_landmarks": 17,
            "n_features": 256,
            "dropout_rate": 0.2,
            "image_channels": "depth"
        }
    },
    "data_loader": {
        "type": "FaceDataLoader",
        "args":{
```

```
        "data_dir": "F:/TrainData/BU_3DFE_processed/",
        "heatmap_size": 256,
        "image_size": 256,
        "image_channels": "depth",
        "n_views": 96,
        "batch_size": 8,
        "shuffle": true,
        "validation_split": 0.1,
        "num_workers": 8
    }
},
"optimizer": {
    "type": "Adam",
    "args":{
        "lr": 0.001,
        "weight_decay": 0,
        "amsgrad": true
    }
},
"loss": "mse_loss",
"metrics": [
    "my_metric", "my_metric2"
],
"lr_scheduler": {
    "type": "StepLR",
    "args": {
        "step_size": 50,
        "gamma": 0.1
    }
},
"trainer": {
    "epochs":15,

    "save_dir": "F:/saved/",
    "save_period": 1,
    "verbosity": 2,

    "monitor": "min val_loss",
    "early_stop": 10,

    "tensorboard": true
},
"process_3d": {
```

```
        "filter_view_lines": "quantile",
        "heatmap_max_quantile": 0.5,
        "heatmap_abs_threshold": 0.5,
        "write_renderings": false,
        "off_screen_rendering": true,
        "min_x_angle": -40,
        "max_x_angle": 40,
        "min_y_angle": -80,
        "max_y_angle": 80,
        "min_z_angle": -20,
        "max_z_angle": 20
    },
    "preparedata": {
        "raw_data_dir": "F:/TrainData/BU_3DFE/",
        "processed_data_dir": "F:/TrainData/BU_3DFE_processed/",
        "off_screen_rendering": true
    },
   "pre-align": {
      "align_center_of_mass" : false,
      "rot_x": 0,
      "rot_y": 0,
      "rot_z": 0,
      "scale": 1,
      "write_pre_aligned": false
   }
}
```

B.3 Rendering and Training

The Deep-MVLM set the whole data set as the input of training. If we want to reduce the size of training set, the parameter need to change in the function of the file named pareparedata as listing B.4

Listing B.4 split data into train and test

```
def split_data_into_train_and_test(base_file_names, output_dir):
    train_file = output_dir + "dataset_train.txt"
    test_file = output_dir + "dataset_test.txt"
    f1 = open(train_file, 'w')
    f2 = open(test_file, 'w')

    new_set = []
```

```
    for name in base_file_names:
        n1 = os.path.dirname(name)
        if n1.find('F') > -1:
            n1 = n1.replace('F', '')
            num = int(n1)
            if num < 46:
                new_set.append(name)
                f1.write(name + '\n')
            else:
                f2.write(name + '\n')
        if n1.find('M') > -1:
            n1 = n1.replace('M', '')
            num = int(n1)
            if num < 45:
                new_set.append(name)
                f1.write(name + '\n')
            else:
                f2.write(name + '\n')
    f1.close()
    f2.close()
    return new_set
```

Run the rendering program by calling:

```
python preparedata --c configs/BU_3DFE-depth.json
```

When the rendering program is done, run the training program by calling:

```
python train --c configs/BU_3DFE-depth.json
```

Visualize the training loss and validation loss by calling:

```
tensorboard --logdir F:/saved/log/MVLMModel_BU_3DFE
```

B.4 Landmark prediction

A trained network will got after training. From the command line, run :

```
python -m http.server
```

This is for Python 3.x. In a browser, open:

```
http://localhost:8000/
```

Then navigate to the trained network file. A URL can get for loading the network. In the folder named deepmvlm, the network is set in the python file named api. Listing B.5 shows the file that set the network model with URL.

Listing B.5 Setting the network for prediction.

```
models_urls = {
    'MVLMModel_BU_3DFE-depth':
        'http://localhost:8000/FinalThesis/NetworkModel/MVLMModel_BU_3DFE_depth_
        100Object_40epoch-9a77e50a202d9cbe554d10f2306e5059ce4a839d234ef8b503407e
        39a09fec31.pth',
  }
```

After setting the network model, the prediction process of landmark run by calling:

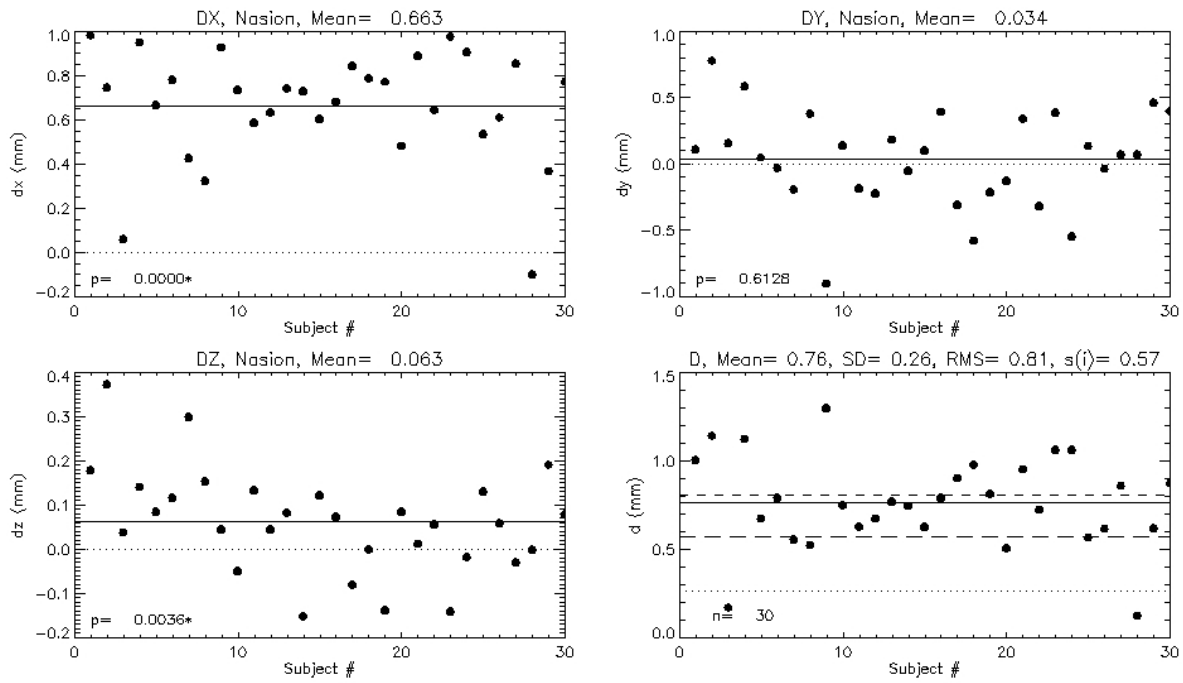
```
python predict.py --c configs/BU_3DFE_depth.json --n yourdirectory
```

Appendix C

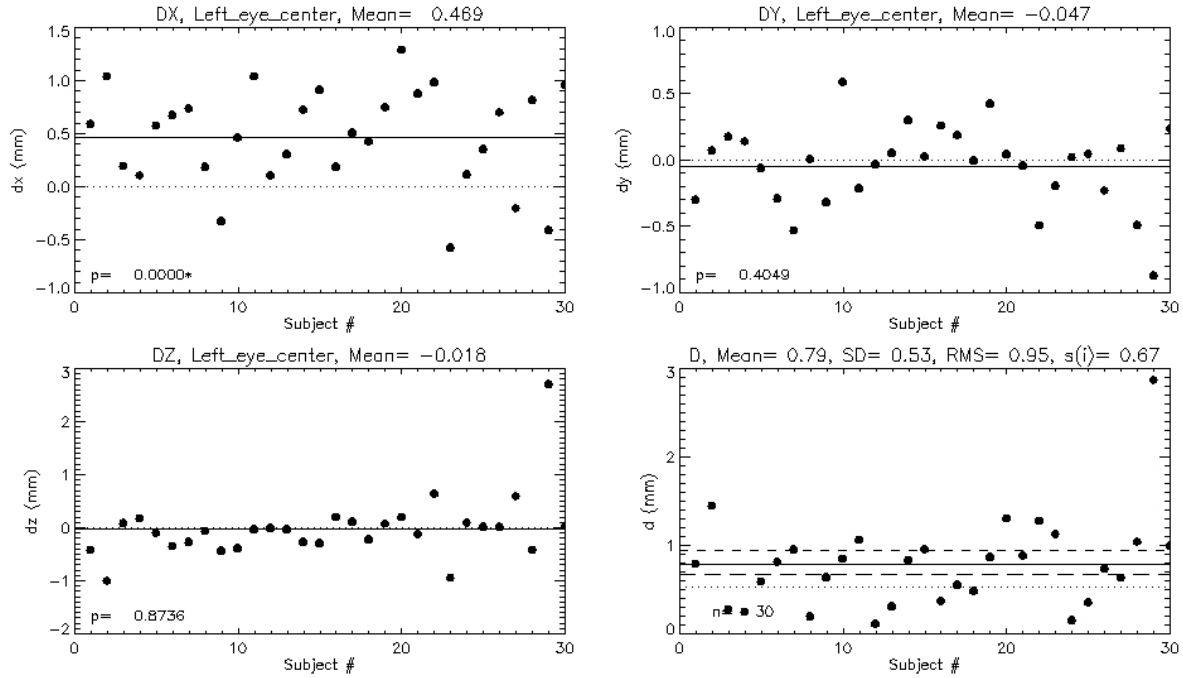
Network related results

C.1 Results of intra-test

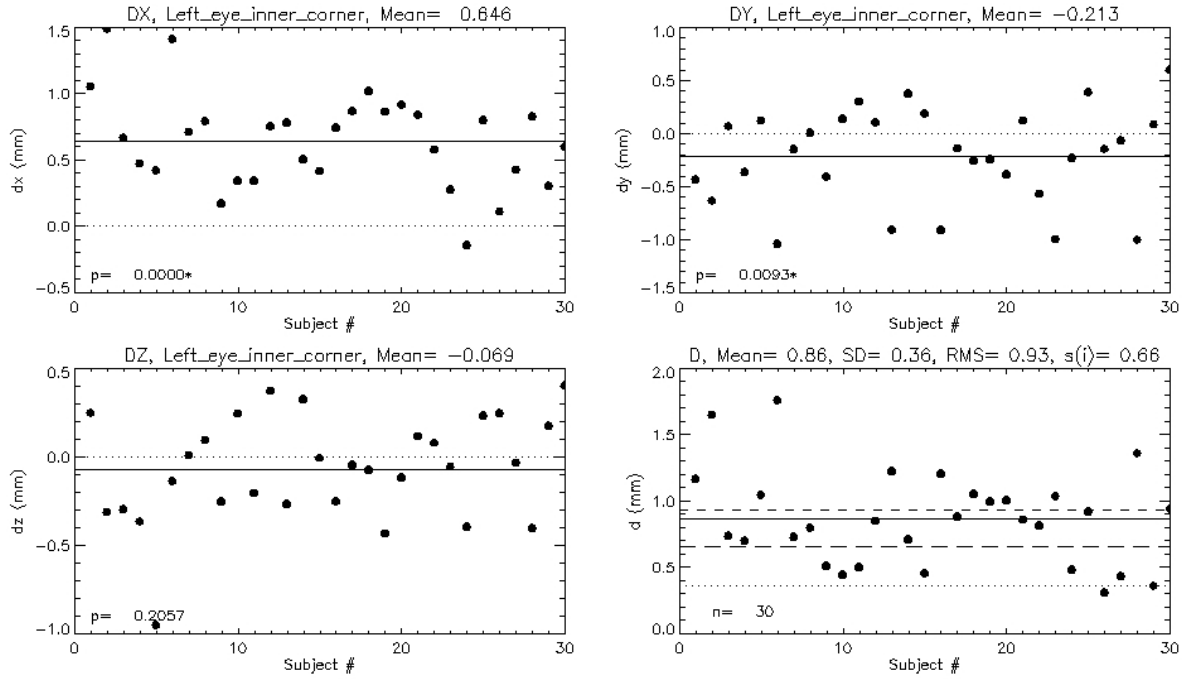
Figure C.1 illustrates the error of each landmark.



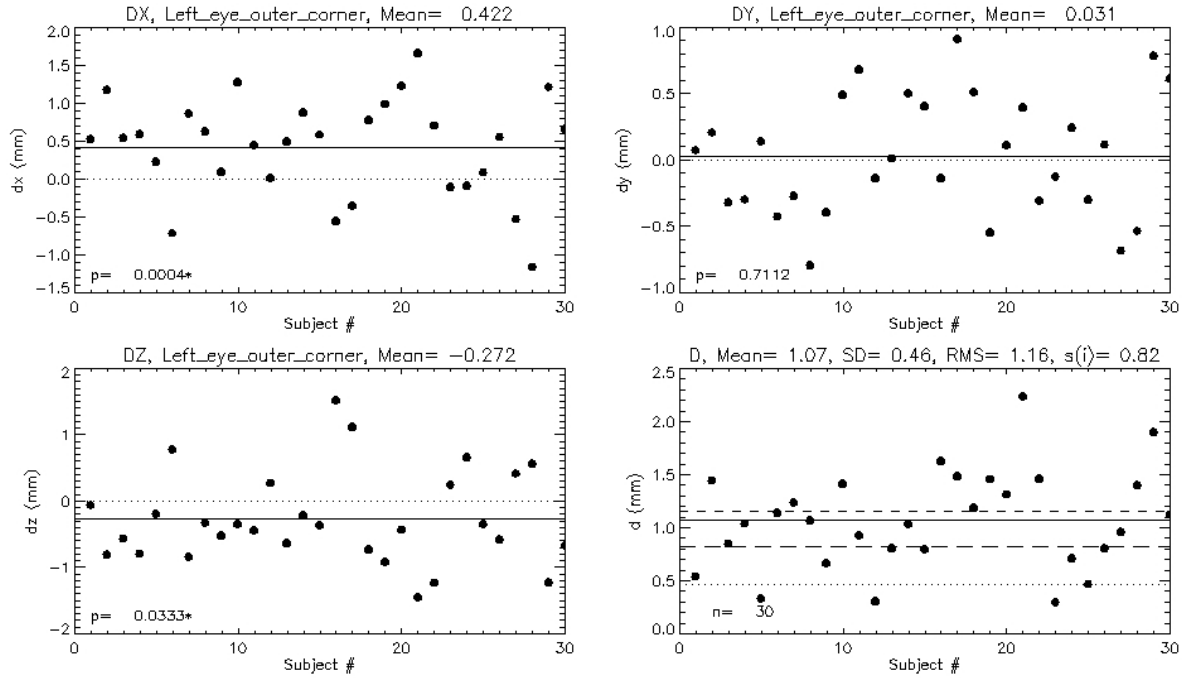
(a) The error of nasion



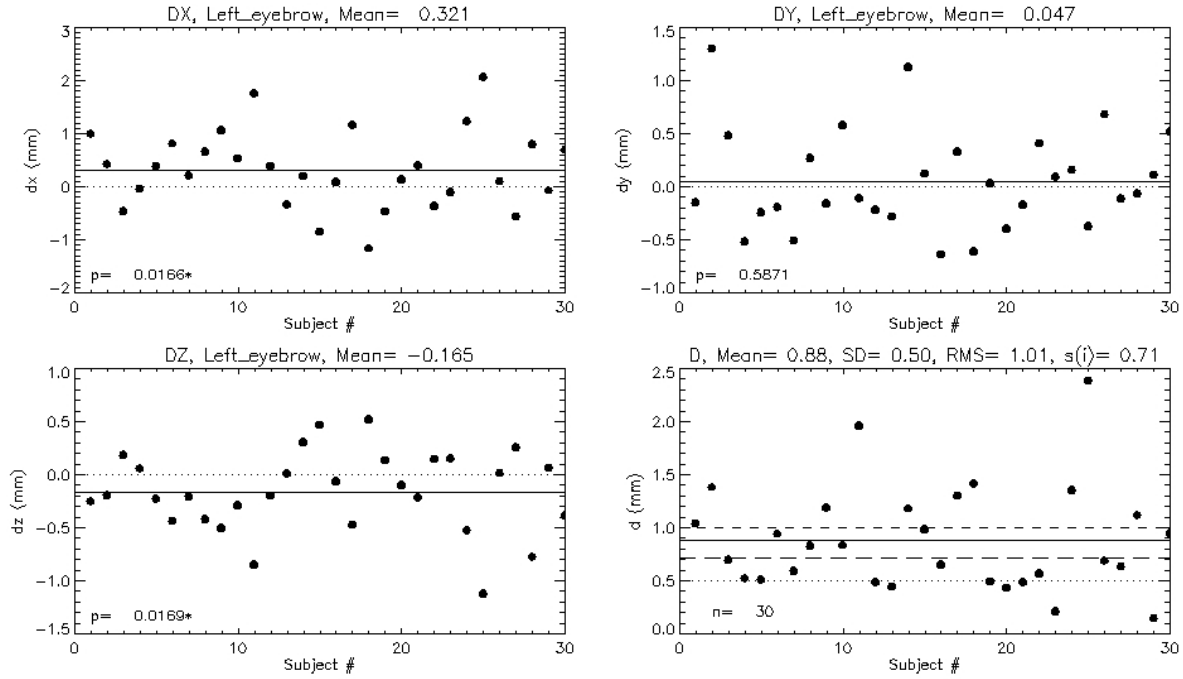
(b) The error of left eye center



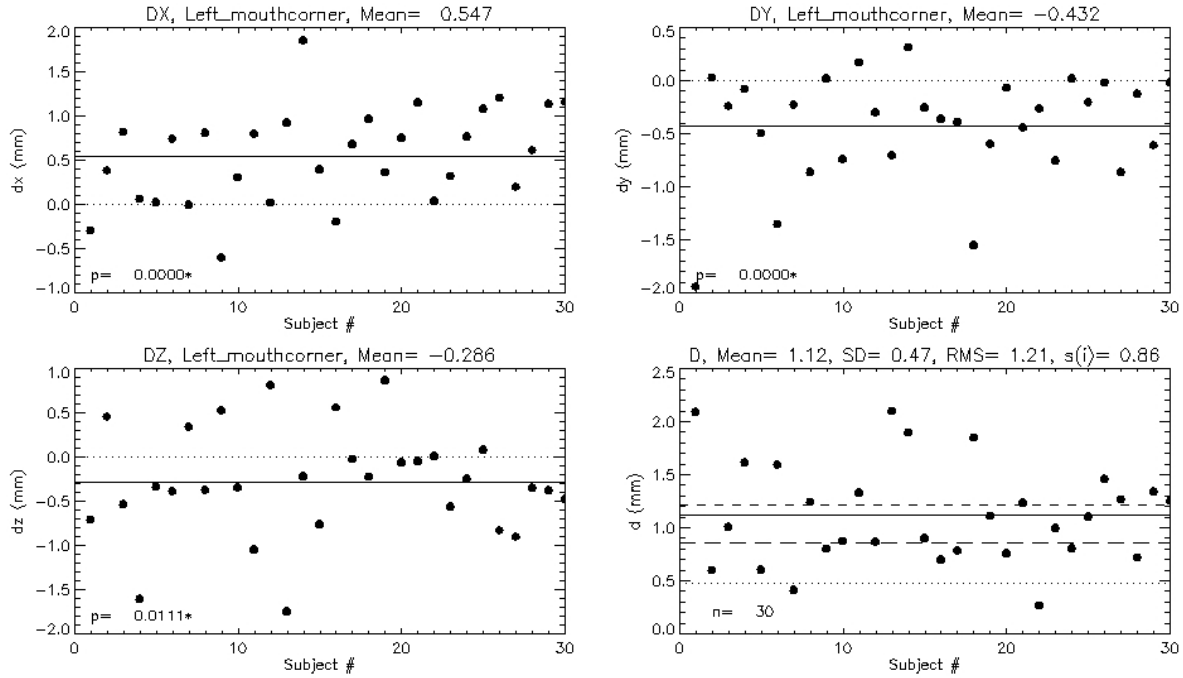
(c) The error of left eye inner corner



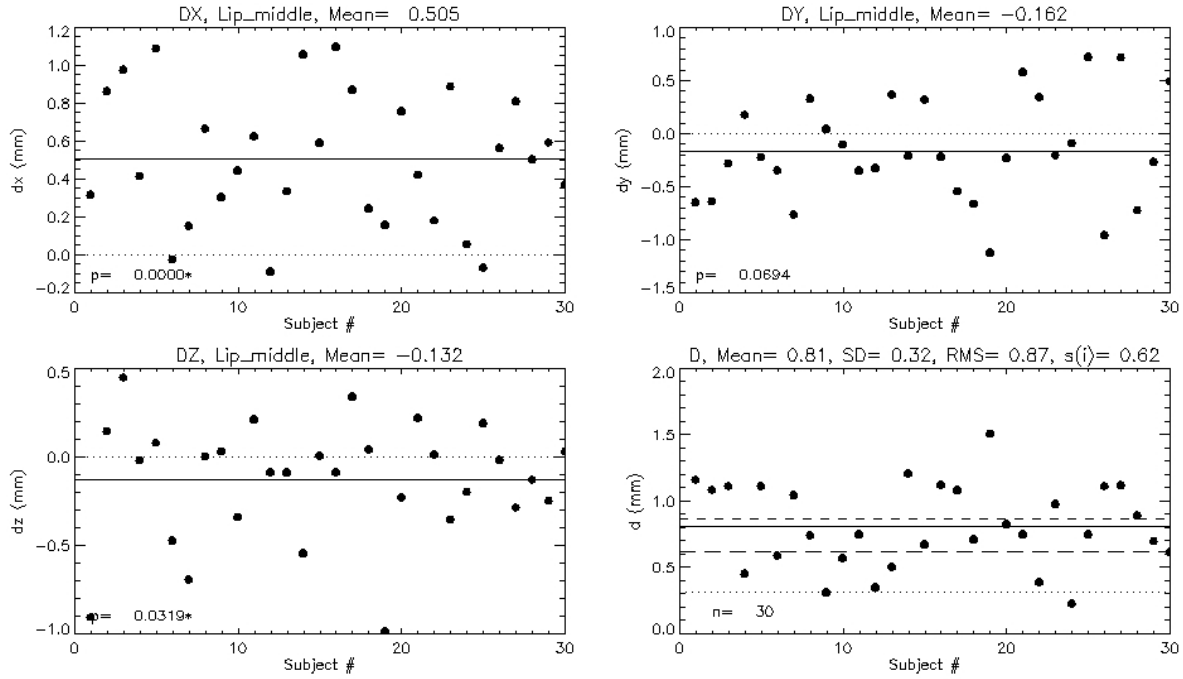
(d) The error of left eye outer corner



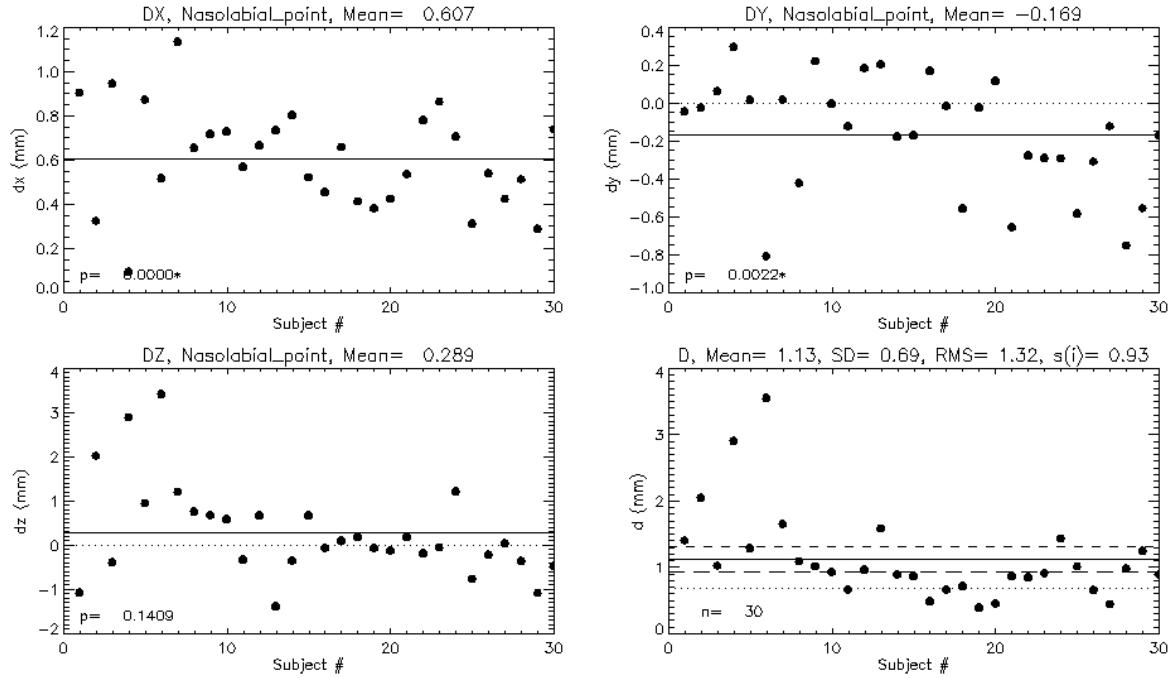
(e) The error of left eyebrow



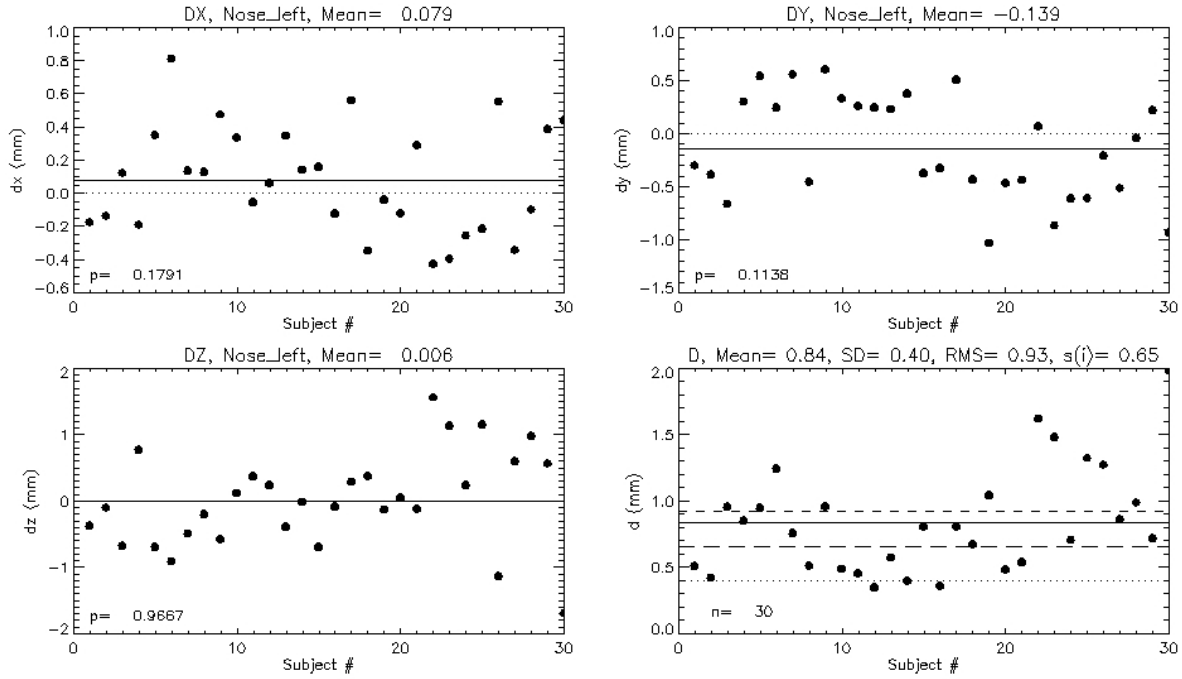
(f) The error of left mouth corner



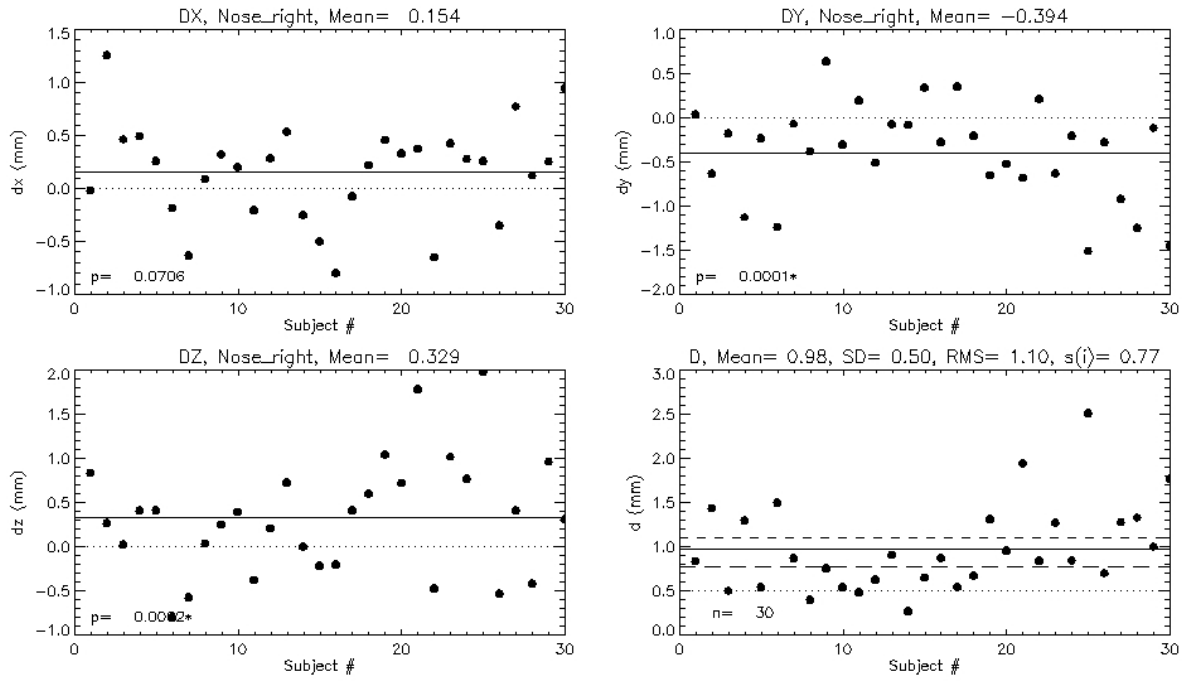
(g) The error of lip middle



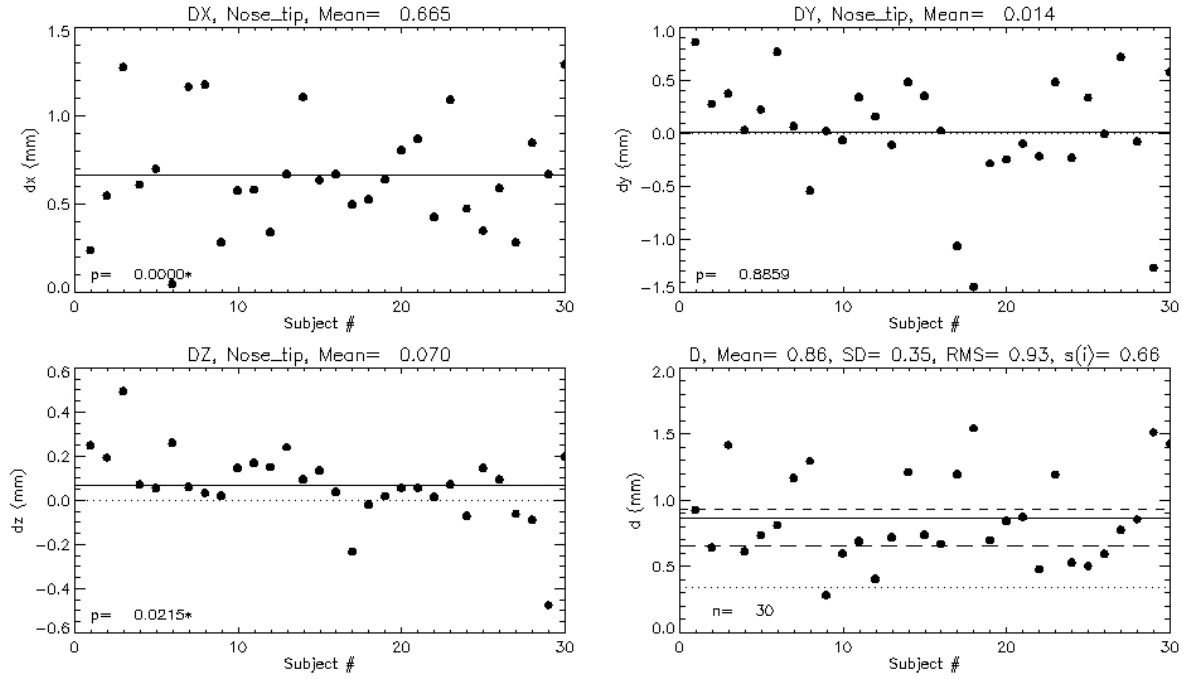
(h) The error of nasolabial point



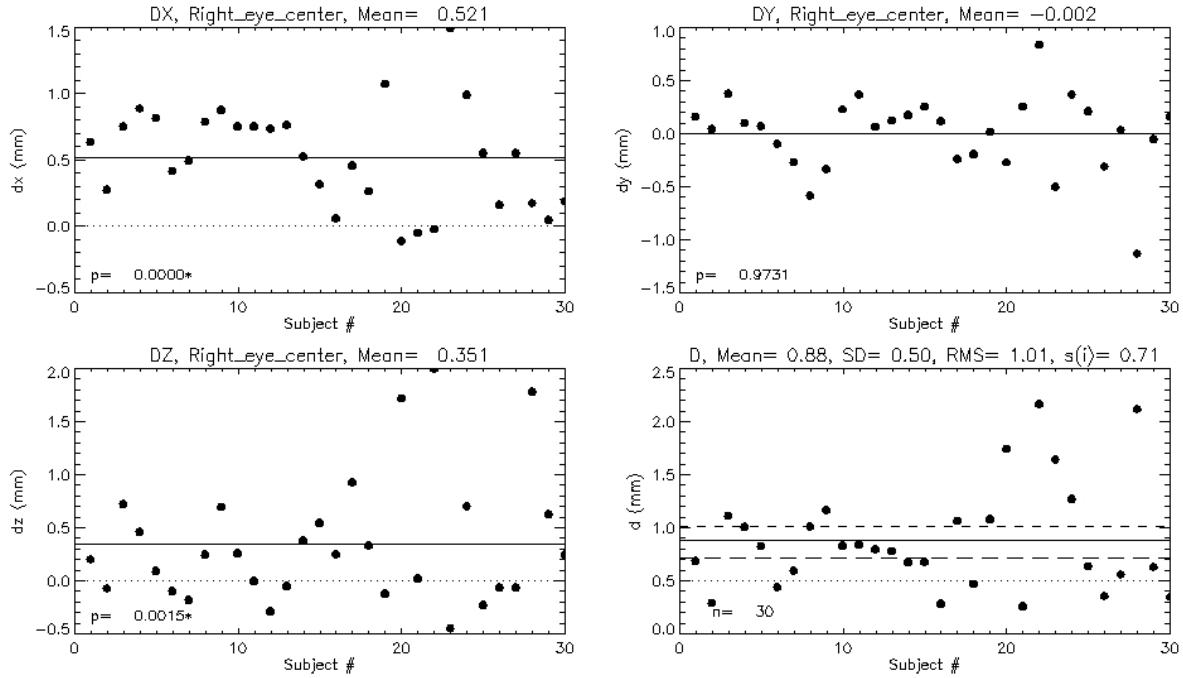
(i) The error of nose left



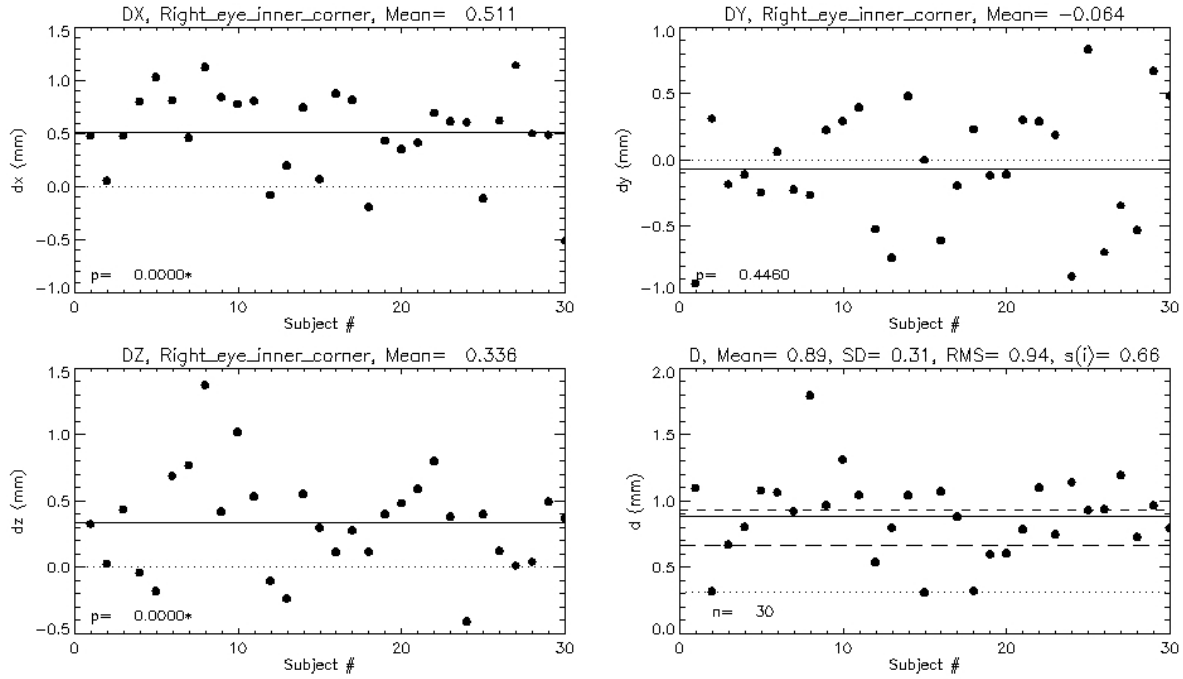
(j) The error of nose right



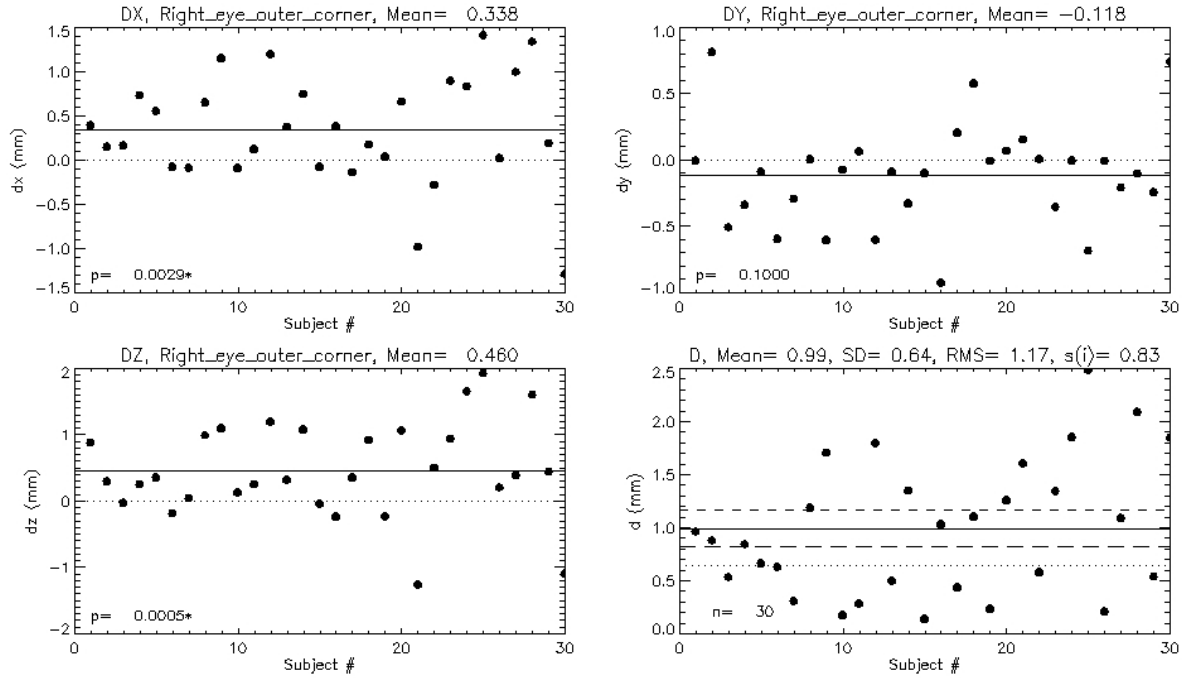
(k) The error of nose tip



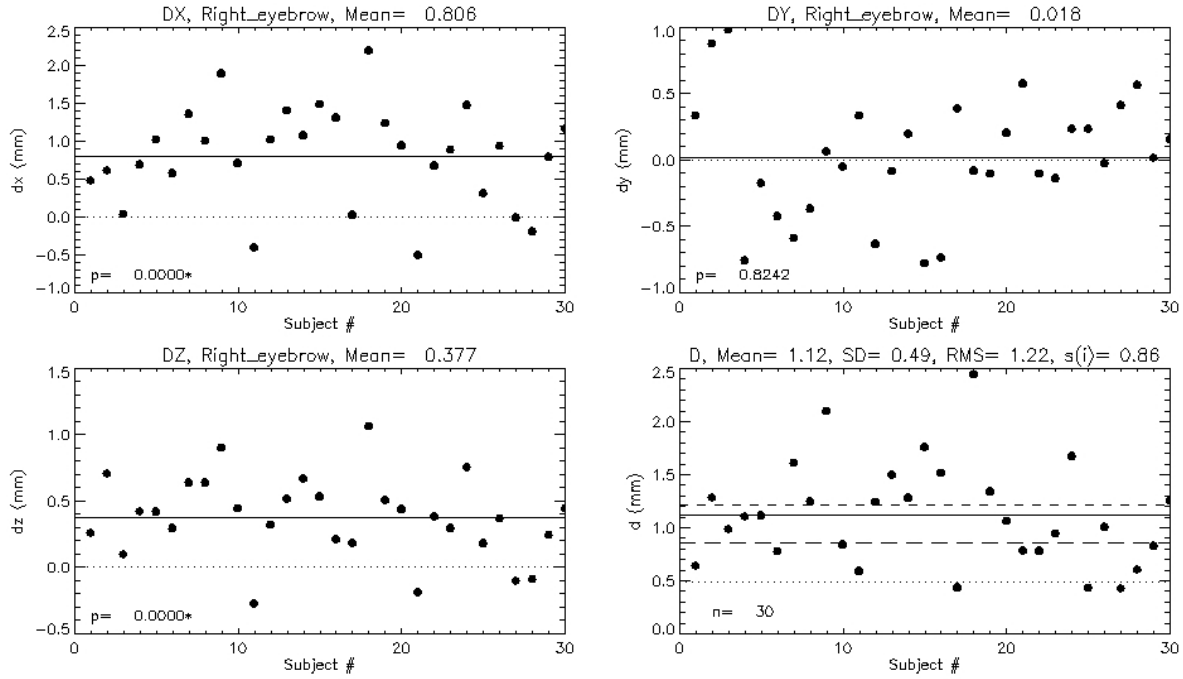
(l) The error of right eye center



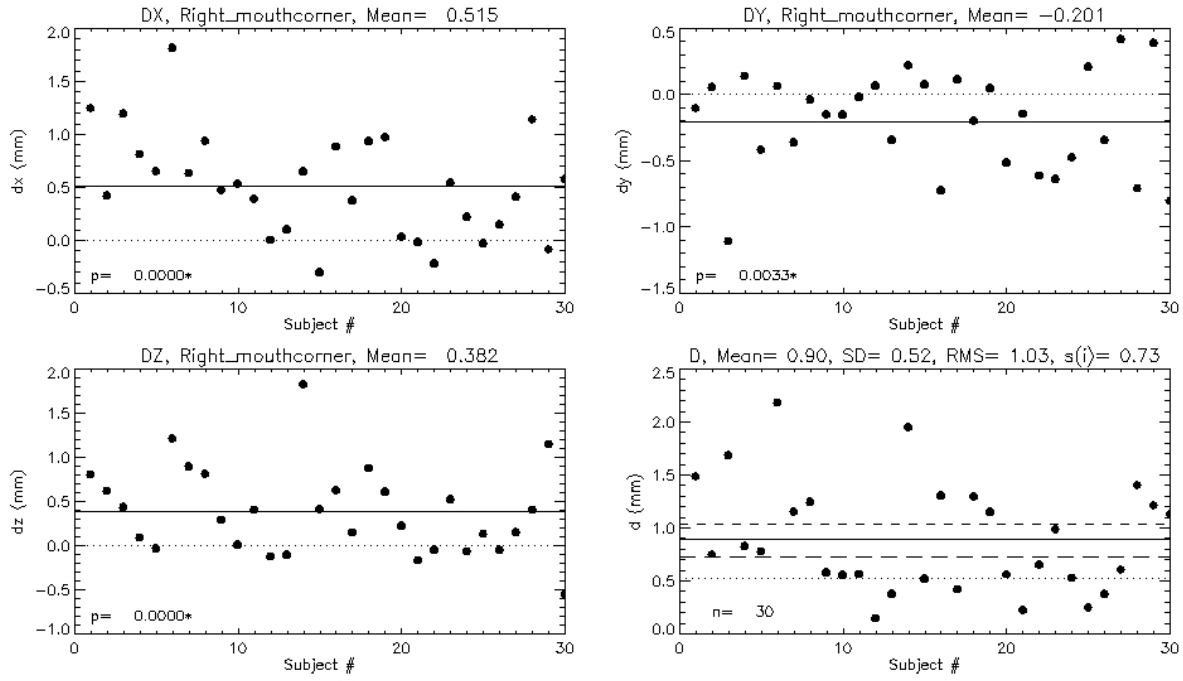
(m) The error of right eye inner corner



(n) The error of right eye outer corner



(o) The error of right eyebrow



(p) The error of right mouth corner

Figure C.1: The manual error of each landmark

C.2 Max error of the face

The following table shows the face which has max error in the test set.

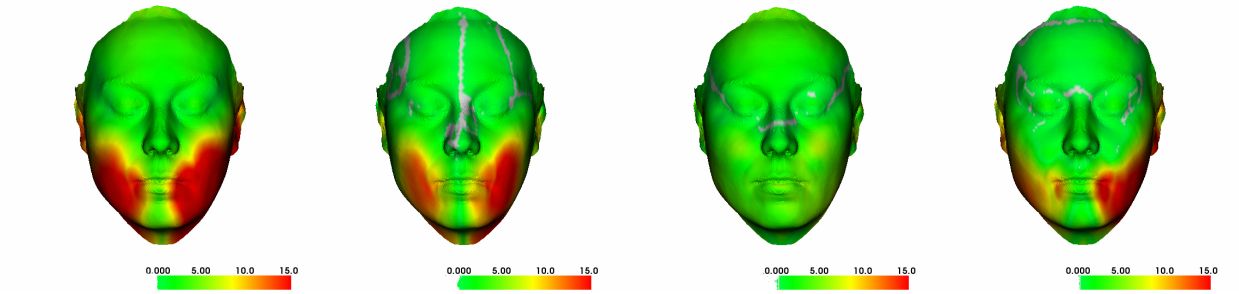
No. of LM	Max error	No. of face
1	4.9354	20
2	2.6928	14
3	5.8720	14
4	2.9726	14
5	4.8134	17
6	3.1784	8
7	5.3470	15
8	3.2619	1
9	7.6680	19
10	6.6345	5
11	4.5558	16
12	4.3591	7
13	4.2072	1
14	5.0912	10
15	3.3892	10
16	6.0083	1
17	5.1484	10

Table C.1: The face which has the max predicted error.

Appendix D

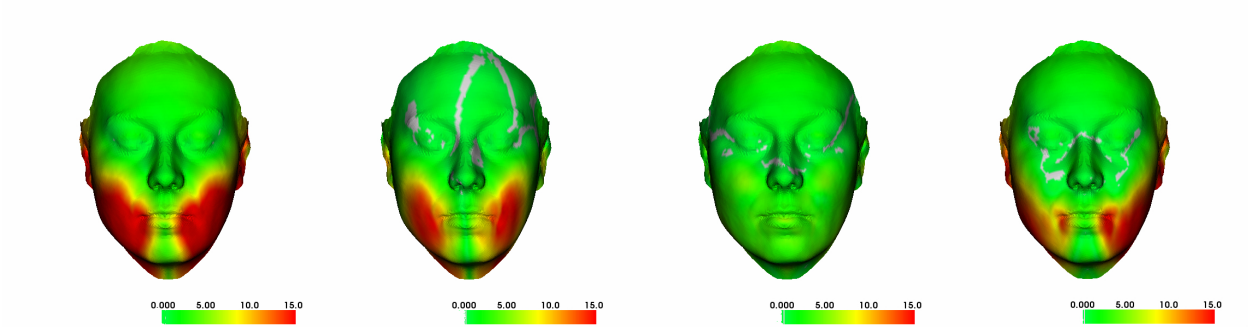
Additional results

D.1 Results of the point-wise difference



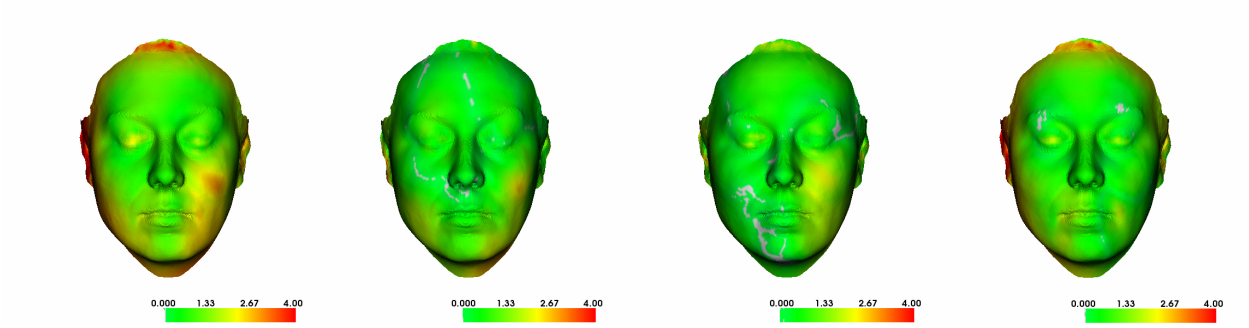
(a) (b) (c) (d)

Figure D.1: Face surface color-coded according to the difference between neutral 1 and maximum smile in an example subject. (a) The overall pointwise difference. (b) The pointwise difference along the x-axis (transverse direction). (c) The pointwise difference along the y-axis (vertical direction). (d) The pointwise difference along the z-axis (sagittal direction). The face shown is a mean neutral face as acquired before the smile sequence started. The grey area on the face means that the difference in this area is close to zero.



(a) (b) (c) (d)

Figure D.2: Face surface color-coded according to the difference between maximum smile and neutral 2 and in an example subject. (a) The overall pointwise difference. (b) The pointwise difference along the x-axis (transverse direction). (c) The pointwise difference along the y-axis (vertical direction). (d) The pointwise difference along the z-axis (sagittal direction). The face shown is a mean neutral face as acquired before the smile sequence started.



(a) (b) (c) (d)

Figure D.3: Face surface color-coded according to the difference between neutral 1 and neutral 2 in an example subject. (a) The overall pointwise difference. (b) The pointwise difference along the x-axis (transverse direction). (c) The pointwise difference along the y-axis (vertical direction). (d) The pointwise difference along the z-axis (sagittal direction). The face shown is a mean neutral face as acquired before the smile sequence started.

D.2 Results of the face asymmetry

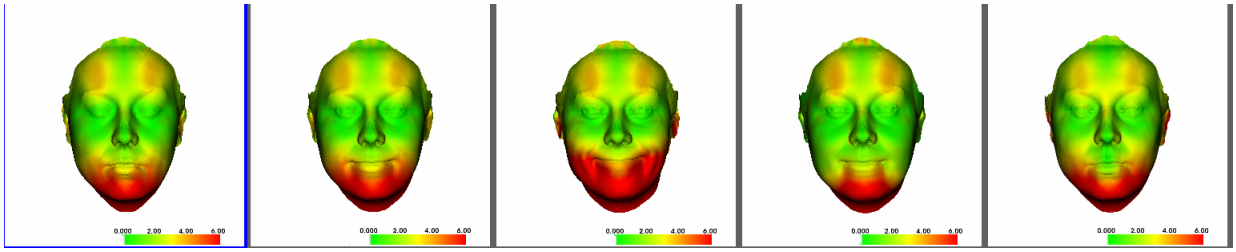


Figure D.4: Asymmetric values change with time

Appendix E

PCA code

E.1 Modules installation

Listing E.1 shows the modules needed for computing PCA.

Listing E.1 Calculation of PCA.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import math
```

E.2 Calculation of PCA

Listing E.2 is the function for computing the PCA. The input is a matrix that every columns contain the same feature of each face and every rows contain features in the same face. This function return three values: the eigenvectors, the eigenvalues and the data after reducing the dimension.

Listing E.2 Calculation of PCA.

```
def pca(dataMat,k):
    newData = dataMat - mean_shape
    covMat =
        np.matmul((dataMat-mean_shape).T,dataMat-mean_shape)/(dataMat.shape[1]-1)
    eigVals,eigVects=np.linalg.eigh(np.mat(covMat))
    eigValIndice=np.argsort(eigVals)
    k_eigValIndice=eigValIndice[-1:-(k+1):-1]
    #get first k vlaues of eigvectors
    k_eigVect=eigVects[:,k_eigValIndice]
    lowDDataMat=np.array(newData*k_eigVect)
```

```
    return eigVals,eigVects,lowDDataMat
```

E.3 Kernel smoothing

Listing E.3 shows the function that smooth the trajectories in the PC space. The input of kernel smooth is the trajectory before smoothing and the range of the time. The input of get smooth trajectory has same meaning as input of kernel smooth. The output of get smooth trajectory is the trajectory after smoothed.

Listing E.3 Kernel smoothing.

```
def kernel_smooth(inputs,t):
    num = []
    den = []
    for i in range(len(inputs)):
        kernel = max(1-abs(i-t)/10,0) #triangular
     #   kernel = 1/np.sqrt(2*math.pi)*np.exp(-0.5*(np.square(i-t)/10)) #Gaussian
        T_i = kernel * inputs[i]
        num.append(T_i)
        den.append(kernel)
    nums = sum(num)
    dens = sum(den)
    return nums/dens

def get_smooth_trajectory(smooth_in):
    T =[]
    for i in range(100):
        Ti = kernel_smooth(smooth_in,i)
        T.append(Ti)
    T=np.array(T)
    fig = plt.figure()
    ax = fig.gca(projection='3d')
    figure = ax.plot(smooth_in[:,0], smooth_in[:,1],smooth_in[:,2],
        c='b',label='original trajectory')
    figure = ax.plot(T[:,0],T[:,1],T[:,2], c='r',label='smooth trajectory')
    ax.set_xlabel('PC1')
    ax.set_ylabel('PC2')
    ax.set_zlabel('PC3')
    ax.legend()
    return T
```

E.4 synthesizing face

Listing E.4 shows how to synthesize a face from a point in the PC space. The input of this function is a point in PC space. This function returns the landmark position of the synthesized face on the real space.

Listing E.4 synthesizing face.

```
def get_manual_face(points):
    temp = flip_eigVects[:,0:3].T
    re_build = np.matmul(points,temp)
    re_build = mean_shape + re_build
    re_build=re_build.reshape(51,)
    for s in range(len(points)):
        ax.scatter(re_build[s,0::3], re_build[s,1::3], re_build[s,2::3],
            depthshade = False)
        ax.view_init(elev=130, azim=-90)
    plt.show()
    return re_build
```

Appendix F

The comparison between auto-landmarking and manual landmarking

Figure F.1 shows the landmarks placed by a clinician as well by means of the automatic software.

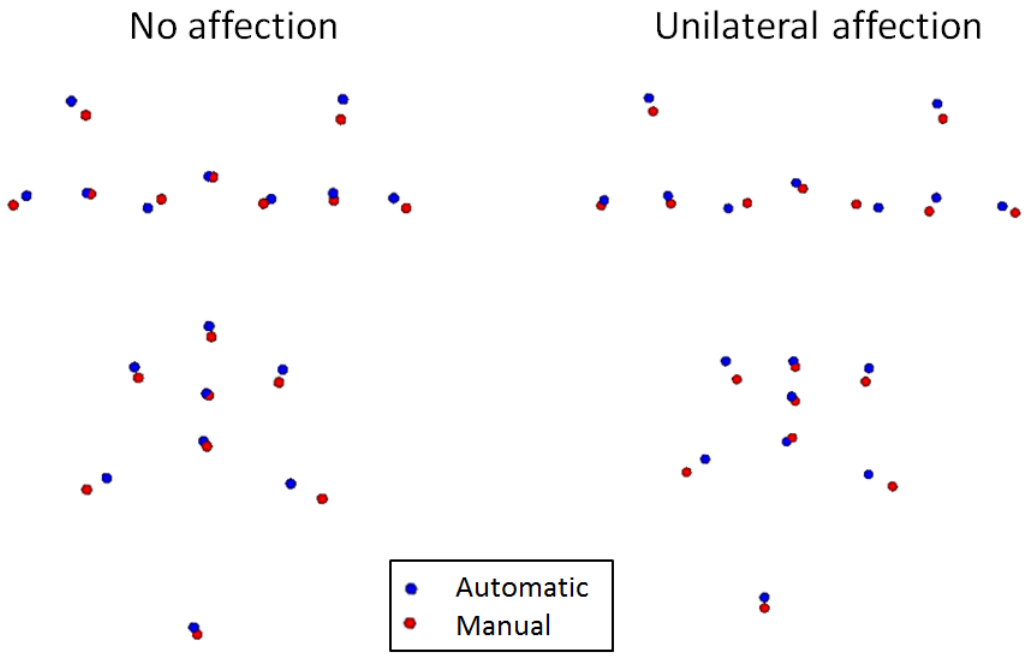
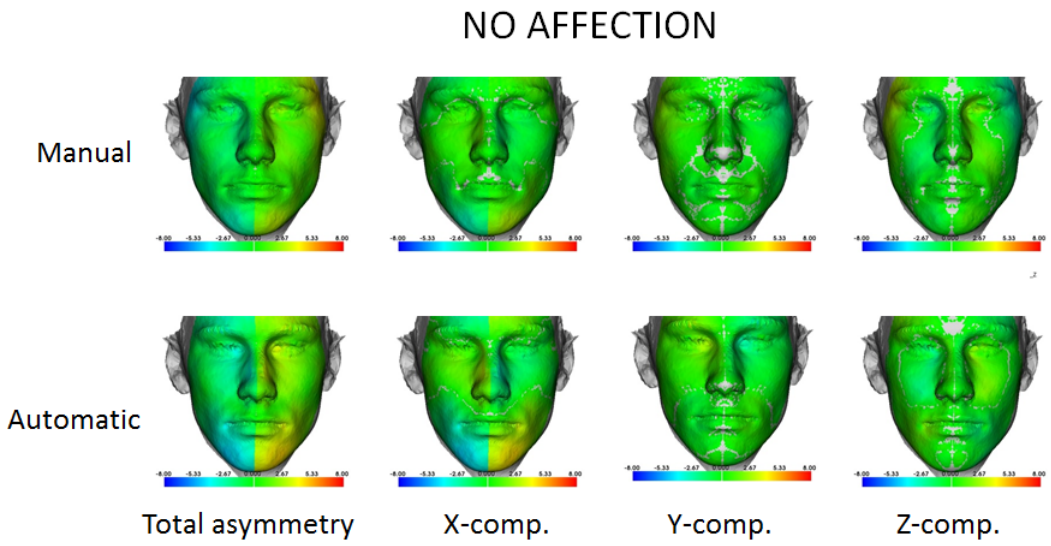
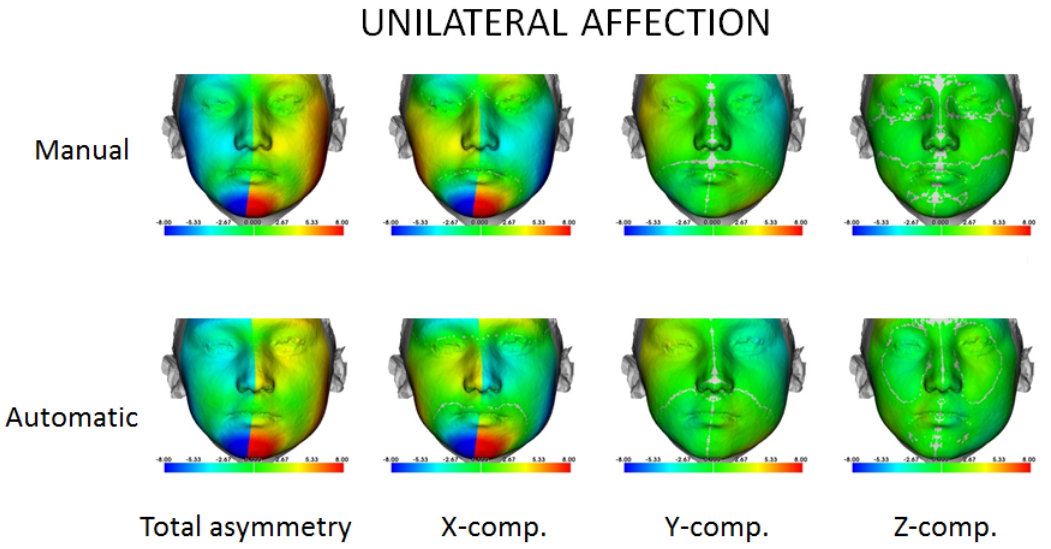


Figure F.1: Landmarks places by an experienced clinician (red spheres) and by the automatic method (blue spheres) in two cases of subjects with juvenile idiopathic arthritis.

Figure F.2 shows the corresponding results of computation of facial asymmetry based on the landmarks.



a



b

Figure F.2: Illustration of amount of asymmetry in two subjects with juvenile idiopathic arthritis. a (upper figure): Individual with no affection of the mandibular joints. Upper row shows the results by using manually places landmarks while bottom row shows results of using automatically placed landmarks. The four columns represents the total amount of asymmetry (leftmost column) and the amount of asymmetry in the lateral (sideways), vertical (up-down) and sagittal (front-back) directions in the face. Color bar ranges from -8mm to 8mm. b (lower figure): Same as a, but for a case with unilateral affection.

Bibliography

[1] Tron A Darvann. Landmarker: a vtk-based tool for landmarking of polygonal surfaces. *In silico dentistry-the evolution of computational oral health science. Osaka, Japan: Medigit*, pages 160–2, 2008.

[2] Ranmus. Deep-mvlm repository. `https://github.com/RasmusRPaulsen/Deep-MVLM`.

[3] Rasmus R Paulsen, Kristine Aavild Juhl, Thilde Marie Haspang, Thomas Hansen, Melanie Ganz, and Gudmundur Einarsson. Multi-view consensus cnn for 3d facial landmark placement. In *Asian Conference on Computer Vision*, pages 706–719. Springer, 2018.

[4] Chua Hock-Chuan. 3d graphics with opengl basic theory. `https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html`.

[5] Steve Marschner and Peter Shirley. *Fundamentals of computer graphics*. CRC Press, 2015.

[6] glumpy. Transformations. `https://glumpy.github.io/modern-gl.html`.

[7] Jens Fagertun, Stine Harder, Anders Rosengren, Christian Moeller, Thomas Werge, Rasmus R Paulsen, and Thomas F Hansen. 3d facial landmarks: Inter-operator variability of manual annotation. *BMC medical imaging*, 14(1):35, 2014.

[8] Norm MacLeod. Palaeomath: Part 20 - principal and partial warps. `https://www.palass.org/publications/newsletter/palaeomath-101/palaeomath-part-20-principal-and-partial-warps`.

[9] Tim J Hutton, Bernard F Buxton, Peter Hammond, and Henry WW Potts. Estimating average growth trajectories in shape-space using kernel smoothing. *IEEE transactions on medical imaging*, 22(6):747–753, 2003.

[10] wikipedia. Polygon mesh. `https://en.wikipedia.org/wiki/Polygon_mesh`.

[11] Rami R Hallac, Nikhitha Thrikutam, Pang-Yun Chou, Rong Huang, James R Seaward, and Alex A Kane. Kinematic analysis of smiles in the healthy pediatric population using 3-dimensional motion capture. *The Cleft Palate-Craniofacial Journal*, 57(4):430–437, 2020.

[12] Hildur Ólafsdóttir, Stephanie Lanche, Tron A Darvann, Nuno V Hermann, Rasmus Larsen, Bjarne K Ersbøll, Estanislao Oubel, Alejandro F Frangi, Per Larsen, Chad A Perlyn, et al. A point-wise quantification of asymmetry using deformation fields: application to the study of the crouzon mouse model. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 452–459. Springer, 2007.

[13] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[14] VTK. vtk homepage. `https://vtk.org/`.

[15] Patrick Min. meshconv. `http://www.patrickmin.com/meshconv` or `https://www.google.com/search?q=meshconv`, 1997 - 2019. Accessed: 2020-03-15.

[16] wikipedia. Thin plate spline. `https://en.wikipedia.org/wiki/Thin_plate_spline`.

[17] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006.

[18] WJB Houston. The analysis of errors in orthodontic measurements. *American journal of orthodontics*, 83(5):382–390, 1983.

[19] Klaus-Robert Müller. Machine learning in non-stationary environments.