3D Face Recognition
Data Processing: Registration and Deformation

M.M.J. Gerlach & C.T. Rooijers
3D Face Recognition
Data Processing: Registration and Deformation

BACHELOR OF SCIENCE THESIS

For the degree of Bachelor of Science in Electrical Engineering at Delft University of Technology

M.M.J. Gerlach & C.T. Rooijers

September 7, 2013

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) · Delft University of Technology
A 3D face recognition algorithm has been developed for the Microsoft Kinect in the scope of the final bachelor project at the TU Delft in 2013. The aim of the project is to develop a prototype face recognition system.

The prototype system had to outperform the existing 2D face recognition system. The project is divided into three subgroups each designing a specific part of the system. The three subgroups are: the data acquisition group, the data processing group and the data comparison group. This thesis provides the data processing part of the system.

An overview of the different existing algorithms is made, followed by the requirements for the system. The algorithm should give reliable results in a reasonable time and should be pose, expression and illumination invariant. A 2D face recognition algorithm, PCA, was implemented first.

The 3D face recognition algorithm follows a morphable model approach. Several algorithms are used to align and fit the 3D face scan from the Kinect. To align the scan with a model, ICP and spin-images are used also referred to as registration. Deformation is done by a non-rigid ICP algorithm to fit the model with the scan. From the fitted model a geometry image and a normal image is generated. For prototyping reasons Matlab was used to implement and test the algorithm.

The developed algorithm is tested by several measurements. From the results of these measurements, it could be concluded that the algorithm is robust and reliable, but isn’t fast. The proposed alignment solution is able to deal with rotation between \(-\pi/2\) and \(\pi/2\) about any of the three axes and with translation in any of the three direction in the range of 1 till 10 cm. The proposed solution is able to improve up till 12% overall in the case of rotation compared to conventional spin-images and up till 43% in the case of translation. A speed-accuracy trade-off have been made. Furthermore, many optimization can still be made. More research will be needed to fulfill the system.
Table of Contents

<table>
<thead>
<tr>
<th>Preface</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Face Recognition Algorithm</td>
<td>3</td>
</tr>
<tr>
<td>2.1 2D Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>2.2 3D Algorithms</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Requirements</td>
<td>5</td>
</tr>
<tr>
<td>2.3.1 Requirements with regards to the 2D Face Recognition Algorithm</td>
<td>5</td>
</tr>
<tr>
<td>2.3.2 Requirements with regards to the 3D Face Recognition Algorithm</td>
<td>5</td>
</tr>
<tr>
<td>2.3.3 Requirements with regards to the (ecological) situation of the system in its surroundings</td>
<td>6</td>
</tr>
<tr>
<td>2.3.4 Requirements with regards to the production process</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Implemented Algorithm</td>
<td>6</td>
</tr>
<tr>
<td>2.4.1 2D System</td>
<td>6</td>
</tr>
<tr>
<td>2.4.2 3D System</td>
<td>6</td>
</tr>
<tr>
<td>3 Algorithm of alignment and fitting</td>
<td>9</td>
</tr>
<tr>
<td>3.1 System Overview</td>
<td>9</td>
</tr>
<tr>
<td>3.2 3D Face Model</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Alignment</td>
<td>10</td>
</tr>
<tr>
<td>3.3.1 Coarse alignment</td>
<td>12</td>
</tr>
<tr>
<td>3.3.2 Fine alignment</td>
<td>14</td>
</tr>
<tr>
<td>3.4 Fitting</td>
<td>15</td>
</tr>
<tr>
<td>3.4.1 The Problem</td>
<td>15</td>
</tr>
<tr>
<td>3.4.2 The Method</td>
<td>15</td>
</tr>
<tr>
<td>3.4.3 Optimization</td>
<td>16</td>
</tr>
<tr>
<td>3.5 Geometry and Normal Image</td>
<td>16</td>
</tr>
</tbody>
</table>
4 Measurements

4.1 Rotation test .................................................. 19
  4.1.1 Result of ICP ........................................... 20
  4.1.2 Result of spin-images .................................. 21
  4.1.3 Result of the proposed solution ..................... 22

4.2 Translation test ............................................... 23
  4.2.1 Result of ICP ........................................... 24
  4.2.2 Result of spin-images .................................. 25
  4.2.3 Result of the proposed solution ..................... 26

4.3 Fitting test .................................................. 27

5 Conclusion .................................................. 31

5.1 Evaluation of the System ................................... 31

5.2 Future Work ................................................ 31

Bibliography .................................................. 33

Glossary .................................................. 37

  List of Acronyms ........................................... 37
  List of Symbols ........................................... 37
Preface

The bachelor Electrical Engineering at the Delft University of Technology is completed with the "Bachelor Afstudeer Project" (BAP). As part of the BAP students write a thesis to show their academic capability. The last quarter of the final year was fully spent on this project. Our project group consist six Electrical Engineering students, working in pairs. The main goal was to develop a reliable 3D face recognition prototype system in nine weeks. This thesis is about the data processing part of the system.

Delft, University of Technology

September 7, 2013

M.M.J. Gerlach & C.T. Rooijers

Acknowledgments

We would like to thank our supervisors Shingo Mandai and Edoardo Charbon for their support during the project. We would like to acknowledge the Delft University of Technology for giving us the opportunity to write this thesis, especially Jaap Hoekstra, Ton Slats en Martin Schumacher for supplying our work room and opening the door for us every morning. Further the Circuits and Systems group deserves special thanks for the generous financial support to purchase a Microsoft Kinect and for supplying us with a "super" computer. We are grateful to Arie Stobbe for his help. Thanks also to our team members for the fine collaboration.
Chapter 1

Introduction

The field of face recognition has been around for many years. Faces contain biometric information which is convenient to read by a face recognition system instead of DNA, fingerprints etc. The first automated system utilizing face recognition was developed in 1977 [1]. Many applications for security access and identification followed [2,3] and face recognition has even been successfully applied at the 2008 Beijing Olympic Games [4].

Although a face is unique, a face recognition system is not always reliable, especially by varying circumstances such as: expression, pose and illumination. Creating a 2D picture of the face loses too much biometric information. New 3D camera technologies [5,6] could overcome these problems of face recognition.

That’s why recent research focuses on 3D face recognition solutions and try to tackle the challenging problems of 2D face recognition by developing a robust algorithm for 3D cameras, working under varying circumstances such as: expression, pose and illumination. The existing solutions rely on expensive high-resolution 3D cameras while the cheap low-resolution 3D cameras that have recently become available such as the Microsoft Kinect have been left fairly untouched.

Recent research [7] has shown a 3D face recognition algorithm working with low-resolution cameras although their system has some significant drawbacks. Sparse coding relies on the complete data requiring impractical amounts of storage for a real system. Therefore our goal is to develop a real prototype system for 3D face recognition with the Microsoft Kinect. The system should be better than existing 2D systems, overcome the expression, pose and illumination problems.

The system can divided into three tasks. These tasks are as listed below:

- Data acquisition from the Kinect and the extraction of the face from the data.
- Data processing for face recognition.
- Comparison of the face data with a created database.
The data processing and data comparison group together form the face recognition algorithm, the core for a face recognition system. It is the aim of this thesis to describe the data processing algorithm and the choices made.

In chapter 2 we studied the different existing 2D face recognition algorithms [8–19]. Afterwards we will get a feel for 2D face recognition by implementing an algorithm, according to the requirements, making use of the RGB data from the Kinect. We decided not to use the 2D algorithm for our final system, because the recognition rate will not benefit from the combination of two different algorithms separately for 2D and 3D. This only becomes more reliable when both have a clear match. Otherwise, some weights given to each algorithm helps to determine a positive match [20] or a sophisticated algorithm [21]. Instead we will use the 2D data to improve our 3D face recognition, done by data acquisition group.

Also in chapter 2 we will move on to the different existing 3D algorithms [7,22–29] and chose our approach according to the specifications of our system. In chapter 3 you can find our approach, the actually data processing algorithm. In chapter 4 the algorithm is measured by different tests. The conclusions of this thesis and some suggestions for further work are provided in chapter 5.
In this chapter a brief overview of the different 2D face recognition algorithms is given in section 2.1, followed by the 3D face recognition algorithms in section 2.2. The requirements for the 2D and 3D algorithm are given in section 2.3. Algorithm choices are made for 2D and 3D in section 2.4 using the overview and requirements.

2.1 2D Algorithms

Many different algorithms for face recognition exist. The first algorithms were based on local geometric features. This method proved to be unsuccessful for reliable face recognition. The next advancements were in holistic algorithms, which aim to describe the global features of the face by a linear approach. The first algorithm was Principal Component Analysis (PCA) which describes the faces as a linear combination of eigenfaces, a set of eigenvectors, which maximize variation across all faces [8].

The downside to eigenfaces is that variations in the training images of the same image tend to outweigh variations across faces. Furthermore nonlinear attributes of the faces are discarded. To overcome the first problem Linear Discriminant Analysis (LDA) was introduced which is based on fisherfaces which maximize the variation across the training images of the same faces [9]. Independent Component Analysis (ICA) followed, which make use of higher-order statistics to improve [10].

Many variations of the PCA, LDA and ICA algorithms have been made with varying results [11,12]. The holistic approaches have been studied thoroughly and comparative studies give different conclusions on results and performance depends largely on the application [13,14]. In holistic based approaches pre-processing in the form of normalization and alignment is important, especially when fewer training images are available [15].

Although this improvements, eigenfaces are still not really suitable for changes in illumination, expressions and pose. Elaborated research was done on local geometric features based algorithms to overcome these problems. Hidden Markov Model (HMM) were also applied to
the problem of face recognition [16]. They are sparsely used and their hidden nature make them hard to optimize.

Also other advancements were made in local feature base approaches such as Local Binary Patterns (LBP) [17], Gabbor wavelets and Elastic Bunch Graph Matching (EBGM) [18]. The last category is model based algorithms like Active Appearance Model (AAM) [19]. A comparison of these different 2D face recognition algorithms is made in Table 2.1.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>Easy to implement</td>
<td>Unreliable</td>
</tr>
<tr>
<td></td>
<td>Widely used</td>
<td>Hard to make reliable</td>
</tr>
<tr>
<td></td>
<td>Incorporated in Open Computer Vision Library (OpenCV)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fast, little memory</td>
<td></td>
</tr>
<tr>
<td>LDA</td>
<td>Optimization of PCA</td>
<td>Optimization of PCA</td>
</tr>
<tr>
<td></td>
<td>Idem PCA</td>
<td>Not always an improvement on PCA</td>
</tr>
<tr>
<td>ICA</td>
<td>Idem LDA</td>
<td>Not incorporated in OpenCV</td>
</tr>
<tr>
<td>Feature Matching</td>
<td>Reliable</td>
<td>Hard to pick robust features</td>
</tr>
<tr>
<td></td>
<td>Incorporated in OpenCV</td>
<td>Slow</td>
</tr>
<tr>
<td></td>
<td>Well documented</td>
<td>More data</td>
</tr>
<tr>
<td></td>
<td>Easy to convert to 3D</td>
<td></td>
</tr>
<tr>
<td>HMM</td>
<td>Different fields of applications</td>
<td>Not optimizable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sparsely used</td>
</tr>
<tr>
<td>EBGM</td>
<td>Reliable with frontal view</td>
<td>Difficult</td>
</tr>
<tr>
<td></td>
<td>Scalable to 3D</td>
<td>Unreliable with pose variation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not incorporated in OpenCV</td>
</tr>
<tr>
<td>AAM</td>
<td>Possible to convert to 3D morphable model</td>
<td>Not incorporated in OpenCV</td>
</tr>
<tr>
<td></td>
<td>Reliable under all conditions</td>
<td>Difficult, possibly slow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sensitive to misalignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not applied to Kinect data, is it possible?</td>
</tr>
</tbody>
</table>

Despite of all the research and many different algorithms, variation in expression and pose is still not overcome. The evolution in the 3D camera field, create new opportunities to overcome these still existing problems in face recognition. Obvious a 3D image contain more information compared with 2D.

### 2.2 3D Algorithms

In the beginning conventional 2D algorithms were applied to 3D. Such as PCA [22, 23] and LDA [24] variant for 3D, which increase performance. However it is still not reliable and suitable for low-resolution 3D cameras, such as the Kinect.

Also local feature based approaches [25] are done in 3D, benefits from the depth info. This makes features more unique, but needs high-resolution 3D cameras to extract the features well. Later new algorithms made especially for 3D came such as a combination of Annotated Face Model (AFM) and Average Regional Model (ARM) [26], which outperforms the conventional approach and are also more applicable to low-resolution 3D cameras.

In the past years many algorithms have been developed for high resolution and low noise 3D cameras. Much effort has gone to a Hausdorff distance approach [27]. Developing algorithms
2.3 Requirements

Predefined requirements are important to help choosing the right algorithm to implement in the face recognition system. Because of the 2D algorithm is only use for practice it has totally different requirements as the 3D algorithm.

2.3.1 Requirements with regards to the 2D Face Recognition Algorithm

The goal is to implement 2D face recognition algorithm fast and move on to 3D, because the 2D algorithm is not used in the final face recognition system. The requirements for 2D are:

- Easy to implement; the algorithm should be implemented fast, because the 2D algorithm won’t be part of the face recognition system. It is only done to get some feeling for facial recognition.
- Widely use (code already available); when code already is available, we don’t lose time by coding the algorithm.
- Reusable for data acquisition and comparison group; the data acquisition and comparison group can start immediately and reuse their work for the 3D system even if the algorithm itself won’t be used in the system.

2.3.2 Requirements with regards to the 3D Face Recognition Algorithm

The 3D algorithm should be an improvement of the existing 2D algorithms. This will be the core of the face recognition system. The requirements are:

- Pose, expression and illumination invariant; the algorithm should overcome the existing problems in 2D facial recognition.
- Robust, even with incomplete 3D data; the algorithm can work correctly with incomplete 3D data from the Microsoft Kinect.
- Reliable; the algorithm should observe small differences between people (even distinguish twins).
- Fast; the face recognition system should recognize a person in 2 seconds.
- c++ code; finally the algorithm should be translated to c++.
2.3.3 Requirements with regards to the (ecological) situation of the system in its surroundings

This system will only exist in a digital environment so it will not have an impact on the ecological surroundings. The only impact the system might have is that it uses a computer. This computer consumes energy and unless this energy is completely green, the energy consumption will have a small impact on the environment.

Privacy is an issue with data acquisition, because personal biometric data is registered. So the complete system must be an offline and secured system.

2.3.4 Requirements with regards to the production process

- Matlab; the algorithm will be tested and run using Matlab.

2.4 Implemented Algorithm

2.4.1 2D System

As mentioned above an easy to implement and widely use algorithm is desired. Because of the widely use of PCA, we decided to implement it with an existing Matlab code [30] optimized by the data acquisition and comparison groups, who can reuse code for the 3D algorithm. In this way we got a quick feel for face recognition and could move on to 3D. The knowledge of eigenfaces will help us to decide which algorithm we will use for our 3D system.

2.4.2 3D System

As mentioned in the requirements the goal is to develop a robust face recognition system under variation in pose, illumination and disguise. By applying a 3D morphable model framework, the system can compensate the variance in pose by proper alignment. Furthermore although the Kinect 3D capturing process might be influenced by illumination the 3D data itself is illumination invariant [7] and therefore it should allow us for robust recognition under varying illumination conditions.

This solution had been applied successfully to high-resolution data from 3D cameras, but to our best of knowledge has not been used with low-resolution data. The expensive 3D camera also have long data acquisition time while the Kinect data acquisition is real-time, as shown in Table 2.2.

Microsoft Kinect

The 3D camera used for our face recognition platform is the Microsoft Kinect. The main reasons to choose the Kinect is that it has both a 2D and 3D camera and it’s well developed Software Development Kit (SDK) and open source alternatives. Furthermore low cost make it widely available compared to the expensive alternatives.
Table 2.2: Comparison of 3D data acquisition devices [7].

<table>
<thead>
<tr>
<th>Device</th>
<th>Speed (sec)</th>
<th>Charge Time</th>
<th>Size (cm$^3$)</th>
<th>Price (USD)</th>
<th>Acc. (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3dMD</td>
<td>0.002</td>
<td>10 sec</td>
<td>N/A</td>
<td>&gt;$50,000</td>
<td>&lt;0.2</td>
</tr>
<tr>
<td>Minolta</td>
<td>2.5</td>
<td>no</td>
<td>23073</td>
<td>&gt;$50,000</td>
<td>0.1</td>
</tr>
<tr>
<td>Artec Eva</td>
<td>0.063</td>
<td>no</td>
<td>2635</td>
<td>&gt;$20,000</td>
<td>0.5</td>
</tr>
<tr>
<td>3D3 HDI R1</td>
<td>1.3</td>
<td>no</td>
<td>N/A</td>
<td>&gt;$10,000</td>
<td>&gt;0.3</td>
</tr>
<tr>
<td>SwissRanger</td>
<td>0.02</td>
<td>no</td>
<td>287</td>
<td>&gt;$5,000</td>
<td>10</td>
</tr>
<tr>
<td>DAVID SLS</td>
<td>2.4</td>
<td>no</td>
<td>N/A</td>
<td>&gt;$2,000</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Kinect</strong></td>
<td><strong>0.033</strong></td>
<td><strong>no</strong></td>
<td><strong>676</strong></td>
<td><strong>&lt;$200</strong></td>
<td><strong>1.5-50</strong></td>
</tr>
</tbody>
</table>

The Kinect is not based on the time of flight principle which most modern 3D cameras use but has a rather different approach to acquire the depth data. An infrared (IR) pattern is emitted and the Complementary Metal Oxide Semiconductor (CMOS) sensor detects the change in relative distance of the IR dots [6].

The data presented by the Kinect is a depth map, which is an image where the intensity represent the depth. The depth images is noisy and contains large holes. Recent research has shown that the low resolution noisy data from the Kinect can be tough to use but averaging and filtering has shown promise [31] or applying techniques like KinectFusion [32]. Large holes and noisy surfaces still remain however and conventional hole filling techniques fail to improve these. To avoid holes on arbitrary locations to influence the recognition result a surface resampling technique needs to be applied.

In our system we fit a morphable model to fill missing data and smooth the noise. This has already been done for high resolution cameras [28]. In the next chapter we will follow this approach with Kinect data.
Algorithm of alignment and fitting

The comparison of the different algorithms in chapter 2 shows the promise of a morphable model approach, even with the challenging Kinect data. In this chapter we propose a framework for alignment and fitting, based on the work in [28]. In section 3.1 an overview of the different steps in the algorithm is given. In section 3.2 the Basel Face Model (BFM), used by the algorithms, is introduced. The first step of the system is to align the scan, provided by the data acquisition group, with the model. The alignment consist of applying spin-images and Iterative Closest Point (ICP), as described in section 3.3. Next, fitting method is explained in section 3.4. In section 3.5 two images are generated and provided to the comparison group.

3.1 System Overview

A rough overview of the different steps in the algorithm is given in Figure 3.1. The data received from the data acquisition group is an unordered list of vertices, every vertex is a point in $\mathbb{R}^3: [x, y, z]$. From this point cloud a mesh is created by computing the triangles by using a Delaunay triangulation. The normals to these triangles are computed, which are used by the point to plane distance of ICP and with the creation of spin-images. Our proposed alignment consists of a pre-alignment based on the mean distance, a coarse alignment based on spin-images and a fine alignment based on two kinds of ICP.

Figure 3.1: System Overview

1A triangle is a closed set of three edges.


3.2 3D Face Model

Both the alignment and fitting stage need a generic face model to respectively find a common reference frame and fill in missing data. The 3D face scans obtained by the Kinect contains noise and many holes making it inapplicable for the creation of a face model.

The model we use for our 3D morphable framework is the BFM publicly available from the Computer Science department of the University of Basel [33] as shown in Figure 3.2(a). It is constructed from training sets with a wide variety in sex, age and weight [34] making it applicable to a wide variety of faces. Furthermore high quality 3D scans were used unlike the Kinect.

We removed the internal vertices and triangles that were causing problem in the fitting stage. These internal vertices from the model have no corresponding points in the scan. As discussed in section 3.4, this is not problematic in the first iterations with high $\alpha$, when local deformations are not allowed. It can be problematic in later iterations, where the $\alpha$ has been lowered and local deformation is allowed. The transformation of neighbour vertices will force the internal vertices forward. These internal vertices can be removed from the model without considering model detail since they will not be reached by the geometry and normal image.

The data acquisition could not supply faces with reliable ears. Therefore in a later stage the ears and neck were removed from the model by removing all points further than radius 10 centimetre (cm) from the nose. Figure 3.2(b) shows the final used model.

![Original](a) Original ![Selection](b) Selection

Figure 3.2: The Basel Face Model

3.3 Alignment

In the alignment stage the scan from the data acquisition group needs to be brought in correspondence with the face model. This is done by determining a transformation that
3.3 Alignment

optimally aligns the given points in the scan with the model. What makes this problem challenging is that the corresponding point in the model and scan are unknown a-priori [35]. Furthermore the noisy and incomplete Kinect data make it difficult to select robust matching points.

The rigid transformation allows for six degrees of freedom: translation and rotation about any of the three axis. The rotation is given by \( R(\theta, \phi, \gamma) \), followed by the translation \( t_x, t_y, t_z \). The order of the operations is critical, the transformation is done in the following order:

1. Roll, rotate around the z-axis by changing angle \( \gamma \)
2. Pitch, rotate around the x-axis by changing angle \( \phi \)
3. Yaw, rotate around the y-axis by changing angle \( \theta \)
4. Translate by \( t_x, t_y, t_z \)

Figure 3.3 shows the different rotations around the axis. The transformation problem can be translated to finding the transformation matrix \( T \)

\[
\begin{bmatrix}
\cos(\theta) \cos(\phi) & \cos(\theta) \sin(\phi) \sin(\gamma) - \sin(\theta) \cos(\gamma) & \cos(\theta) \sin(\phi) \cos(\gamma) + \sin(\theta) \sin(\gamma) & t_x \\
\sin(\theta) \cos(\phi) & \sin(\theta) \sin(\phi) \sin(\gamma) + \cos(\theta) \cos(\gamma) & \cos(\theta) \sin(\phi) \cos(\gamma) - \cos(\theta) \sin(\gamma) & t_y \\
-\sin(\phi) & \cos(\phi) \sin(\gamma) & \cos(\phi) \cos(\gamma) & t_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

that best resembles

\[
\begin{bmatrix}
X_{\text{model}} \\
Y_{\text{model}} \\
Z_{\text{model}} \\
1
\end{bmatrix} = T \ast \begin{bmatrix}
X_{\text{scan}} \\
Y_{\text{scan}} \\
Z_{\text{scan}} \\
1
\end{bmatrix}
\] \quad (3.1)

The relation in the above equation will never be fully reached. Instead this relation will be approximated by minimizing a norm function.

The computed transformation should be rigid, e.g. it should keep the distance between points constant. All non-rigid deformation, such as shearing, should result from the fitting stage.
As a pre-alignment step a rough translation is determined by looking at the mean x, y and z-values of the points in the model and the scan. The difference between these means is used as a rough initial translation.

### 3.3.1 Coarse alignment

The coarse alignment stage relies on spin-images to find corresponding points between the scan and the model and use groups of the corresponding points to determine a transformation, as show in Figure 3.4. Our implementation is based on the description and pseudo code from the thesis introducing spin-images [36].

#### Spin-image

The spin-images are created using a MATLAB script from [37], which is modified slightly to optimize for CPU-time. To create the spin-images two parameters are needed: the binsize and the image width. The bin size is set equal to the mesh resolution as suggested in [36]. The mesh resolution can be computed by taking the distance between each vertex and it’s neighbours and averaging over all vertices. The image width is set to fifteen as suggested in [36].

For every vertex of the model a spin-image is computed. Next a random vertex is selected from the scan and this is compared to every spin-image of the model by computing the similarity measure. These similarity measures are statistically evaluated to detect vertices with high similarity scores.

To detect which vertices have high similarity scores relative to the others, a histogram of the similarity measures is detected. Since the similarity score histogram typically has a single mode, a simple statistical measure for relative high values exists in the form of the fourth spread or interquartile range. With this measure outliers in the histogram are detected, corresponding to vertices pairs that are highly alike.

![Figure 3.4: Surface matching using spin-images between two range views of a toy robot [36].](image)
After this a list of corresponding points on the model and scan are known. This list needs to be translated to a transformation which aligns the scan to the model, however this can be properly done from just the correspondence. The procedure of finding corresponding points allows a single point on the scan to be matched to several points on the model. This might be due to the symmetry in the model, which is definitely the case for faces. Also point which lie close to each other might have high similarity scores. To get rid of these point, two filter are applied to the list.

The first filter remove correspondences from the group that are relatively small compared to the rest of the group. The second filter removes correspondences that are geometrically inconsistent. Correspondences are removed from the list which can’t be used later to compute a transformation. After filtering, correspondences remain that have high similarity scores and are geometrically consistent. These correspondences are grouped favoring points that are far apart.

For every group a transformation is determined. Transformation that are not rigid or rotate over non acute angles are removed. The transformation that reduces the distance to the model the most is picked as the final transformation. An overview of the surface matching process is shown in Figure 3.5.
Figure 3.5: Surface matching block diagram

3.3.2 Fine alignment

The fine alignment stage consists of two versions of ICP. The first version [38] relies on the point to plane distance while the second version relies on the point to point distance [39].

ICP

A widely used solution to the problem of 3D registration is the ICP algorithm. Although many variations of the ICP algorithm exist, the general idea can be summed up as follows. First corresponding points are found. Next a transformation is computed from these corresponding point and this process iterates until a certain error metric is reached. This solution works well for 3D shapes that are already roughly aligned. An overview of the ICP algorithm is given in Figure 3.6.
3.4 Fitting

3.4.1 The Problem

The fitting stage takes the points of the scan and model brought in correspondence by the alignment stage. Since the model and scan are now in aligned, an ICP algorithm can be applied to gradually deform the model towards the scan. We apply a nonrigid optimal step ICP framework \[40\]. This method finds a deformation for every point in the model instead of a general deformation which has been proven to work better \[41\]. This method is ideal because of it’s robustness against any residual error left by the alignment stage and it’s ability to handle incomplete data.

The idea of the fitting stage is to deform a model onto the target. This fitting stage should keep the local geometric feature of the original scan and at the same time fill missing data by the smooth deformed model data. This results in the regions begin filled in a sensible way.

3.4.2 The Method

Both the model and the scan are described by a set of vertices and edges. For every vertex in the scan a nearest neighbour is found on the model based on the Euclidean distance. We use the knnsearch\(^2\) implementation in Matlab because of it’s speed. Since our model generally has more vertices than the scan the search returns duplicate points. A second stage therefore makes sure that only the points with the smallest Euclidian distance remain.

We use a cost function that contains two terms. The first term is the distance term that tries to minimize the distance between the model and the scan mesh. The second term is the stiffness which penalizes different transformations for neighbour vertices. The algorithm minimizes the complete cost function.

\[
E(X) := E_{\text{distance}}(X) + \alpha E_{\text{stiffness}}(X)
\]

\[\text{Eq. 3.2}\]

\(^2\)A Matlab function to find nearest neighbours.
**Distance Term**

The first term is the distance term which brings corresponding points found by the nearest neighbour algorithm, with duplicate points removed, together. The duplicate points are removed by setting the corresponding elements of matrix W to zero. The points that are on the boundary of the mesh are removed in the same way.

\[ E_{\text{distance}}(X) = \| W(DX - U) \| \]  

(3.3)

**Stiffness Term**

The second term is the stiffness which penalizes different transformations for neighbour vertices. This can be written in matrix notation with the help of the arc-node incidence matrix.

\[ E_{\text{stiffness}}(X) = \| (M \bigotimes G)X \| \]  

(3.4)

### 3.4.3 Optimization

**Parameters**

The \( \alpha \) in Eq. (3.2) can be altered to weigh the effect of the stiffness term on the total cost term. A high \( \alpha \) will result into global fitting and will not allow for local deformation. This is especially useful when the fitting begins and the alignment of the model and scan is not perfect. A high \( \alpha \) will in this case allow the model to align better first before deformation starts. In the fitting process the \( \alpha \) is gradually lowered to allow for more local deformation.

The optimization is in finding the lowest value for \( \alpha \) that allows for the local features of the scan to show in the fitted model, while disallowing noise and bumps in the model to show in the fitted model. Furthermore an optimal value for \( \alpha \) is especially difficult when the level of detail for certain regions of the face, especially the ears, alters in between scans. A region with sparse data points and a low \( \alpha \) will result into undesirable deformation of the model.

**Speed**

The fitting algorithm takes a significant amount of iterations to deform the model reliably, for a result that has both the geometric features of the original data as well as the filling and smoothing properties of the model. Considerable speed can therefor be achieved by small speedups in the code.

### 3.5 Geometry and Normal Image

From the fitted 3D model we generate two images that are used by the comparison group. We use a geometry and normal image to sample the 3D surface of the model. These image offer a simple way of comparing faces while keeping the unique features of the original mesh [42].
To create the geometry image we compute an uniform grid between the minimum and maximum values of the x and y coordinates of the 3D model. Next for every point in the grid the corresponding Z value is computed. These Z values are then scaled to the range of the color scheme. An example of the resulting geometry image can be seen in Figure 3.7(a).

The z value is an explicit function of x and y, e.g. \( z = f(x, y) \). Now the normal to this plane can easily be computed by taking the gradient and normalizing. An example of the resulting normal image can be seen in Figure 3.7(b).

Figure 3.7: The two images that are created from the fitted model for a specific fitted scan. These images serve as an input for the data comparison group.
In this chapter the results of the conducted experiments are presented. The goal of these experiments is to measure the ability of the different algorithms explained in chapter 3 to properly align and fit the points of the scan and the model.

Due to the unavailability of Kinect data, a sample scan supplied with the Basel Face Model (BFM) is used. The supplied scan is already aligned with the model but was not used in creating the BFM. For all test the model and scan meshes where simplified using Qslim [43] to around 2200 faces which corresponds to about 1100 vertices. The aligned scan is misaligned by rotating or translating the scan by a fixed amount. Next the different algorithms are run to realign the scan with the model. The error metric is the 2-norm of the realigned scan points minus the original aligned scan points.

In section 4.1 the ICP algorithm, the algorithm based on spin-images and the proposed algorithm are tested by measure the rotation. In section 4.2 the alignment tests are done for translation. In section 4.3 the fitting algorithm is tested.

4.1 Rotation test

The scan points are rotated by an angle in between $-\pi/2$ and $\pi/2$. In this range nine angles are used, spaced $\pi/8$ apart. The error is computed for every axis as defined in Figure 3.3. This range was chosen since this is the expected rotation in our scan data. For rotation beyond this range, the missing data due to self-occlusion will be a greater problem than the actual rotation. One might argue that this is also the case for the outliers in the current range of rotation.
4.1.1 Result of ICP

**Figure 4.1:** The residual error against rotation for ICP. The algorithm iterates until the change in error is down to 0.1%

**Figure 4.2:** The number of iterations until termination is reached. The algorithm iterates until the change in error is down to 0.1%
4.1 Rotation test

The results of the experiment are in Figure 4.1. A first thing to notice is that even without rotation the error is not reduced to zero. This is accountable to the ICP algorithm converging in to a local minimum rather than a global minimum. For the given range of the angle the ICP algorithm is able to recover the rotation well. Only with the extreme rotation of \(-\pi/4\) and \(\pi/4\) in the case of roll and pitch, ICP fails to align the scan with the model. In Figure 4.2, it can be seen clearly that in the case of roll the number of iteration is low for the extremes. In this situation the ICP algorithm has clearly converged into a local minimum which is far from a correct alignment.

4.1.2 Result of spin-images

Figure 4.3: The residual error against rotation for spin-images. The spin-images were created with a bin size equal to the mesh resolution, an image width of 15 and 90 random points were selected.

The residual error with spin-images is in most cases similar or slightly larger compared to ICP. However the residual error is relatively constant across the rotation range and even with extreme rotation, spin-images are able to align the scan and data reasonably well. This ability make spin-image useful as a coarse alignment stage. Since the spin-images from the scan are selected randomly, this test varies slightly in each run. Still a clear trend can be seen in Figure 4.3: spin-images are always able to improve alignment, only the error differs.
4.1.3 Result of the proposed solution

Figure 4.4: The residual error against rotation for the proposed solution.

Figure 4.5: The reduction in the error achieved by the proposed solution compared to spin-images only against rotation.
4.2 Translation test

The proposed solution consist of spin-images, followed by ICP based on point to plane distance and finally ICP based on point to point distance. The two ICP algorithms where run after the spin-images run in Figure 4.3. This is critical to compare the error of spin-images with the error of the proposed in Figure 4.4, since the residual error is slightly random due to the random spin-image selection explained in section 3.3.1. The reduction in error of the proposed versus spin-images in Figure 4.5 shows a large reduction in some cases and a small increase in others. Overall the fitting stage will benefit from the large reduction while the small increase will be compensated by the first iterations with a high $\alpha$ of the fitting stage. The ICP after spin-images generally improves the alignment and stays clear from unwanted local minima. Furthermore the number of iterations the ICP algorithm does before the termination criterion is reached, is greatly reduced by a rough alignment with spin-images. Eventough the error is increased in some cases, overall the proposed solution is able to improve the error. In the case of yaw an overall reduction 12% is achieved, while pitch and roll are stuck at respectively 5.7% and 0.7%.

4.2 Translation test

In the translation test all scan points are translated by a value between 1 and 10 centimetre (cm). The residual error is computed for every translation in one of the possible three direction independently. This range is chosen as a realistic scenario for the translation in the real system. Translation that exceed this range will generally not occur due to the applied pre-processing, described in section 3.3.
4.2.1 Result of ICP

![Figure 4.6](image1.png)

**Figure 4.6:** The residual error against translation for ICP. The algorithm iterates until the change in error is down to 0.1%

![Figure 4.7](image2.png)

**Figure 4.7:** The number of iterations against translation for ICP. The algorithm iterates until the change in error is down to 0.1%
With translation the ICP algorithm clearly struggles to align the scan and model properly. In Figure 4.6 it can be seen that the residual error gets significantly worse for increasing translation in the x-direction and the y-direction. From Figure 4.7 it can be concluded that the number of iterations in the x-direction and the y-direction is large. In the z-direction the number of iterations is considerably smaller. The partial overlap with the x and y-translation clearly increases the chance of convergences into a local minimum.

### 4.2.2 Result of spin-images

![Graph showing error against translation for spin-images](image)

**Figure 4.8:** The residual error against translation for spin-images. The spin-images were created with a bin size equal to the mesh resolution, an image width of 15 and 90 random points were selected.

The residual error with spin-images is slightly larger for small translation compared to the ICP algorithm. With larger translation the residual error with spin-images is small compared to ICP. From Figure 4.8 it can be concluded that the resulting error is relatively constant with varying rotation. Especially when compared to the clear increasing trend in Figure 4.6.
4.2.3 Result of the proposed solution

Figure 4.9: The residual error against rotation for spin-images. Spin images created with bin size equal to the mesh resolution.

Figure 4.10: The reduction in the error achieved by the proposed solution compared to spin-images against translation.
The results of the proposed solution in Figure 4.9 show comparable results as spin-images, except for the outliers in residual error. The result of the reduction in the error in Figure 4.10, show a general improvement. For the outliers in residual error with spin-images a large improvement is achieved. The proposed solution clearly has the most robust result over the translation range. The overall reduction in error are: 4.8%, 5.9% and 43% for respectively the x, y and z-direction.

4.3 Fitting test

To test how well the fitting stage is able to improve the scan by fitting the model, the high-resolution scan from the BFM was used again. The original high-resolution scan serves as a reference result. The scan was altered by adding white Gaussian noise and creating holes to resemble a more realistic result as received by the Kinect. Noise was added until a signal-to-noise ratio of 50 decibel (dB) was reached and ten holes were created on random locations with a random radius between 1 and 10 millimetre (mm).

![Image](image)

(a) Original  (b) Noise and holes added  (c) Fitted model result

**Figure 4.11:** The original high-resolution scan together with the version altered with noise and holes. The fitted model shows a result close to the original. The holes are filled and the noise reduced. Some detail is lost around the mouth and chin area.

In Figure 4.11 the original scan, the altered scan and the fitted model are shown. The fitting stage is able to fill in the missing data in a proper way, as can be seen when comparing the original with the fitted model. Also the noisy surfaces of the altered model are smoothed by the fitting stage. Overall the fitting stage clearly improves on the altered scan although some details of the original scan are lost.
Figure 4.12: The error between nearest neighbour points on the model and the scan against the number of iterations. The $\alpha$ is lowered if the difference in error is lower than 0.3 mm for five consecutive iterations.

In Figure 4.12 the typical error against iteration curve for one particular run of the fitting stage is shown. The jumps in the error are due to the jumps in the $\alpha$. For a high $\alpha$ value the fitting stage only allows for global deformation. It is clear that the error is reduced rapidly in the first iterations but this reduction lowers for the following steps in the $\alpha$ parameter. For a low $\alpha$ value local deformation is allowed. Finding the optimal value for $\alpha$ is a trade off between a fitting stage that is able to fill holes and smooth noise and at the same time is able to produce enough detail in the fitted model.
4.3 Fitting test

Figure 4.13: An example of the different alignment algorithms in action with KinectFusion data. The ICP algorithm has converged into an unwanted local minimum. The spin-image based algorithm aligns the nose and the forehead well, while it fails to align the mouth and ears. The proposed algorithm aligns the frontal face and ears perfectly for the fitting stage.
In this chapter some conclusions from the work performed over the last nine weeks are drawn. The system is evaluated in section 5.1 with respect to its requirements. Finally, in section 5.2 suggestions for future work are given and explained.

5.1 Evaluation of the System

We use a system based on a morphable model based approach. The used approach has the ability to overcome some of the challenges in the face recognition field such as: variation in pose, illumination and expression. The problem with varying pose is overcome by a sophisticated alignment stage, which is able to compensate for this variation. The challenge in varying illumination is overcome by using 3D data which is in principle invariant under variation in illumination. Variation in expression is tackled by using a geometry and normal image to express the geometry of the entire faces rather than local feature which might vary with expression.

We proposed an algorithm for alignment and fitting. The proposed solution is able to improve up till 12% overall in the case of rotation compared to conventional spin-images and up till 43% in the case of translation. The proposed alignment algorithm is robust over a broad range of rotation (−π/2 till π/2) and translation (1 till 10 cm) as shown by the measurements. When compared to ICP, the proposed algorithm exchanges some accuracy for this robustness.

5.2 Future Work

The code used for spin-images can still be improved. The script used for the creation of spin-images is slow and is responsible for most of the computation time. Further research is needed to determine the optimal amount of random points to be selected before filtering and determination of transformation. This will be a trade off between computation time and reliable results which is far from trivial.
The descriptiveness of the geometry and normal image can be increased by increasing the quality of the point cloud supplied by the data acquisition group. One solution might be to improve on the existing averaging technique while another solution might be using Kinect in combination with a reliable method of detecting the nose. Nose detection in 3D data is still challenging, a solution might be using spin-image based detection using the spin-image of the nose model.

The current fitting result might benefit from more regularizers. The full model might be used if the ears in the scan improve or reliable ear landmarks are used. The current algorithm can be improved by more thorough research in suitable values for $\alpha$. This will be a trade off between a fitting stage that is able to average noise and fill missing data but still has enough detail in the fitted model.

To further overcome the problem of missing data, advances might be towards a system that exploits the symmetry of the faces [7] or fitting with partial scans and multiple cameras [44]. Challenges that still remain are occlusion and disguise.
Bibliography


[34] IEEE, *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, (Genova, Italy), 2009.


### List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAM</td>
<td>Active Appearance Model</td>
</tr>
<tr>
<td>AFM</td>
<td>Annotated Face Model</td>
</tr>
<tr>
<td>ARM</td>
<td>Average Regional Model</td>
</tr>
<tr>
<td>BFM</td>
<td>Basel Face Model</td>
</tr>
<tr>
<td>cm</td>
<td>centimetre</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>dB</td>
<td>decibel</td>
</tr>
<tr>
<td>EBGM</td>
<td>Elastic Bunch Graph Matching</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>ICA</td>
<td>Independent Component Analysis</td>
</tr>
<tr>
<td>ICP</td>
<td>Iterative Closest Point</td>
</tr>
<tr>
<td>IR</td>
<td>infrared</td>
</tr>
<tr>
<td>LBP</td>
<td>Local Binary Patterns</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
</tr>
<tr>
<td>mm</td>
<td>millimetre</td>
</tr>
<tr>
<td>OpenCV</td>
<td>Open Computer Vision Library</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
</tbody>
</table>
List of Symbols

\( \alpha \)   Weight coefficient of the stiffness term
\( \gamma \)   Angle of rotation about the z-axis
\( \phi \)     Angle of rotation about the x-axis
\( \theta \)   Angle of rotation about the y-axis
\( t_x \)     Translation in the x-direction
\( t_y \)     Translation in the y-direction
\( t_z \)     Translation in the z-direction