2012

3D face recognition based on a modified iterative closest point method

Kankan Zhao
University of Wollongong

3D Face Recognition Based on a Modified Iterative Closest Point Method

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Master of Engineering by Research

from

UNIVERSITY OF WOLLONGONG

by

Kankan Zhao

School of Electrical, Computer and Telecommunications Engineering

March 2012

Dedicated to my family

Declaration

I, Kankan Zhao, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Master of Engineering - Research, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Signature:______________________

Kankan Zhao

March 19, 2012

ABSTRACT

Face recognition has gained extensive attention recently, with many applications in a broad range of domains such as access control in security systems and picture tagging in social network web sites. This project builds a 3D face database and recognizes the unknown 3D face images in comparison with the 3D face database.

In 3D face images used in this thesis are acquired by a 3D data acquisition system based on Digital Fringe Projection Profilometry (DFPP). DFPP is an efficient 3D data acquisition system to capture 3D data, with its simple system structure, high resolution and low cost. The 3D database consists of thirty group images In each group, there are three images corresponding with three views with (i.e. left-side view, right-side view, and frontal view) at the same scale of the same subject. The scale is different from group to group.

To achieve 3D face recognition, there are two parts devised: image alignment and comparison. In order to implement efficient and accurate image alignment, two steps which are coarse alignment and fine alignment are implemented. In the coarse alignment step, two 3D images are roughly aligned into a same coordinates system and roughly aligned. After the coarse alignment step, the two face images will be aligned closer and an initial estimated value will be given for the fine alignment.

A modified partial Iterative Closest Point (ICP) method is proposed in the fine alignment step. The partial ICP method is an efficient alignment method for 3D data reconstruction and 3D face recognition. It iteratively aligns the two point sets based on repetitive calculation of the closest points as the corresponding points in each iteration. However, if two 3D face images with different scales are from the same person, the partial ICP method does not work. In this thesis, the scaling effect problem of 3D face recognition has been solved. A 3×3 diagonal matrix as the scale matrix in each iteration of the partial ICP has been well designed. The probing face image which is multiplied by the scale matrix will keep the similar scale with the reference face image. Therefore even if the scales of the probing image and the reference image are different, the corresponding points can be accurately determined. The mean square distance between the two face images are compared to recognize that whether the two face images are from the same person or not.

Based on the experiment results, the 3D face recognition can be achieved via the method proposed in this thesis. The mean square distance between two face images from the same person can reach to less than 0.05 while the two face images from the different persons can only keep 0.10 to 0.30.

ACKNOWLEDGEMENT

I would like to personally thank the following people who supported me with my sincere gratitude.

My deepest gratitude goes first and foremost to my supervisor Professor Jiangtao Xi, for his professional guidance, his useful instructions in supervisions. My grateful thanks also go to my co-supervisor, Dr. Yanguang Yu, for her kind encouragement and patience all through my research studies.

Likewise I am grateful to my parents, for their endless love and selfless support during these years.

I would extend my sincere thanks to all fellows of Optoelectronic Signal Processing Research Lab (OSPR), for their great support and helpful discussion.

Last, my thanks would go to express my heartfelt gratitude to the staffs of the School of Electrical, Computer and Telecommunications Engineering. With their technical supports and patience, I have spent a happy time during the research work in University of Wollongong.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATIONS

$s_0(x)$: input signal.

f: frequency of $s_0(x)$.

h: height value of one point.

L: distance between camera and reference plane

D: distance between the camera and the projector,

$\delta(x)$: fringe shift.

u_1: response function of the projector.

u_2: response function of the camera.

$d(x)$: deformed fringe pattern

f_0: fringe frequency.

$\mathcal{F}$: Euclidean distance between two point sets.

d^2: square Euclidean distance.

p_i: correspondences weight factor.

$\boldsymbol{x_i}, \boldsymbol{y_i}$: i-th point in the model and reference images.

D': reference image.

$\boldsymbol{R}$: rotation matrix.

$\boldsymbol{t}$: translation matrix.

$\boldsymbol{S}$: scale matrix.

l_m: number of correspondences.

D_{max}: distance threshold.

$\mathcal{D}$: expected average correspondence distance.

μ: real average correspondences distance.

σ: deviation of the correspondences distance

ξ: the distance threshold in very bad situation.

$\widehat{\boldsymbol{q}}$: dual number quaternion.

$\hat{q}$: dual number.

$\breve{\boldsymbol{q}}$: quaternion.

$\boldsymbol{q}$: 3*1 vector.

$\widehat{\mathbf{q}}$: 3*1 dual vector.

$\breve{\boldsymbol{n}_{\boldsymbol{\iota}_{\boldsymbol{m}}}}, \breve{\boldsymbol{x}}_{\boldsymbol{\iota}}$: quaternion of the model.

$\breve{\boldsymbol{n}}_{\boldsymbol{\iota}}, \breve{\boldsymbol{y}}_{\boldsymbol{\iota}}$: quaternion of transformed model.

$\breve{\boldsymbol{n}_{\boldsymbol{\iota}_{\mathbf{0}}}}, \breve{\boldsymbol{y}_{\boldsymbol{\iota}_{\mathbf{0}}}}$: quaternion of expected model.

ACRONYMS

ICP: Iterative Closest Point

DFPP: Digital Fringe Projection Profilometry

Industrial CT: Industrial Computed Tomography

MRI: Magnetic Resonance Imaging

PCA: Principal Component Analysis

EGI: Extended Gaussian Image

SIFT: Scale Invariant Feature Transform

SVD: Singular Value Decomposition

DQ: Dual Quaternions

DLP: Digital Light-processing Projects

LCD: Liquid Crystal Display

PSP: Phase Shifting Profilometry

FTP: Fourier Transform Profilometry

SLP: Structured Light Projection

SI: Shape Index

MSE: Mean Square Error

SICP: Scaling Iterative Closest Point

CHAPTER 1 INTRODUCTION

Face recognition is one of the biometric techniques used in access control systems, surveillance systems, credit card payment systems, etc. Face recognition based on 2D face images has already been maturely developed [1]. In order to achieve higher accuracy, researchers introduce face recognition techniques based on 3D data which appeared in the late 20-th century and has been utilized widely recently. Although 3D images are more complicated than 2D images, they are invariant in illumination and accurate in geometric information which provides extra precision for the object recognition. This project aims to achieve the face recognition based on 3D data by a coarse-to-fine alignment and the distance comparison method.

1.1 Background

This project is one of the applications for the Digital Fringe Projection Profilometry (DFPP) data acquisition system. The DFPP data acquisition system is effective since it is characterized with high resolution, low errors and fast acquisition speed. The 3D data acquisition system used in this thesis generates the range images. A range image (depth map) comprises a 2D matrix, and each element in the matrix reflects the distance between one point on the object surface to the camera. Range images provide plenty of geometric information for 3D face recognition and are invariant in different illuminative conditions and viewpoints.

The 3D face recognition technique in this thesis aims to compare an unknown face (probing face) with the database to identify and recognize the unknown face which is a one-to-many matching problem. The database used in this thesis is built by the means of the DFPP data acquisition system. The images in the database feature two characteristics that need to be well considered: scaling effect and the partially-overlapped problem. In this thesis, a method to deal with these problems is developed.

1.2 Contributions

The key contributions made to the field of face recognition are:

- ✧ A new 3D face database is built. It consists of 30 group images, whereby three images of a genuine person with three views—all with the same scale—are included in each group. In different groups, the scale of the three images is differ from other groups.

- ✧ A 3D face recognition procedure which consists of two parts—image alignment and distance comparison—was designed. The image alignment consists of coarse alignment and fine alignment. A feature points extraction based coarse alignment method is developed. In the fine alignment part, a modified Iterative Closest Point (ICP) method addresses the scaling problem and the partially-overlapped problem. Thus, after the alignment, the different distances between the two face images (probing face image and reference face image from the database) are obtained and the 3D face recognition can be implemented via the comparison of the different distances between the two face images.

1.3 Outline of the Thesis

This thesis consists of six chapters, organized as follows:

- ✧ Chapter 1 presents the introduction and contributions of this thesis.
- ✧ In Chapter 2, the literature pertaining to the 3D data acquisition and 3D face recognition methods is reviewed. In addition, feature-based, template-based and multi-model matching methods of 3D face recognition methods are described, addressing the issues inherent in each.
- ✧ In Chapter 3, the procedure used to acquire the 3D facial data is described and a description of the 3D face database is given.
- ✧ By comparing the face images from the database, a coarse-to-fine alignment and comparison strategy is designed in Chapter 4 and Chapter 5. A coarse alignment method is given in Chapter 4 to calculate a rough estimation motion based on the feature points between the probing face image and the reference image selected randomly from the database.
- ✧ In Chapter 5, the fine alignment method and recognition method are presented, including the detailed of the modified partial ICP method, which can be used based on the face images after coarse alignment in Chapter 4. With the initial estimation motion calculated from Chapter 4, a more accurate fine alignment

method can be generated from this chapter. The 3D face recognition can be achieved by comparing the different distances between the two face images which have been finely aligned.

✧ Chapter 6 provides the final discussion and conclusion of the thesis, as well as recommendations for the future work in this field.

CHAPTER 2 LITERATURE REVIEW

In this chapter is given an overview of recent techniques of 3D data acquisition and 3D face recognition. First of all, the overview of 3D profilometry techniques is given including the DFPP technique. Then, existing 3D face recognition methods are presented which can be divided into three categories: feature-based matching methods; template-based matching methods and multi-model matching methods. In the summary, these methods are concluded and compared with their merits and drawbacks.

2.1 3D Data Acquisition

The 3D data acquisition technologies can be divided into two categories: contact measurement and non-contact measurement [2]. With contact measurement techniques, a mechanical arm is utilized to touch the probing object. The height of the object surface can be retrieved by recording the traces of mechanical arms. Instead of using mechanical probe, other techniques using another probe are also applied in 3D measurement, such as inertial [3] (gyroscope, accelerometer), ultrasonic trackers [4] and magnetic trackers [5]. In non-contact techniques, two categories are divided as follows: transmissive and reflective. Transmissive includes industrial Computed Tomography (industrial CT), Magnetic Resonance Imaging (MRI) and ultrasound. Reflective technologies consist of non-optical and optical technologies. The categories of 3D image acquisition technologies are illustrated in Figure 2.1. The data acquisition system in this project is a system based on capture and analysis of the reflected optical fringe patterns, which belongs to the optical technology category. More detailed information will be introduced in Chapter 3.

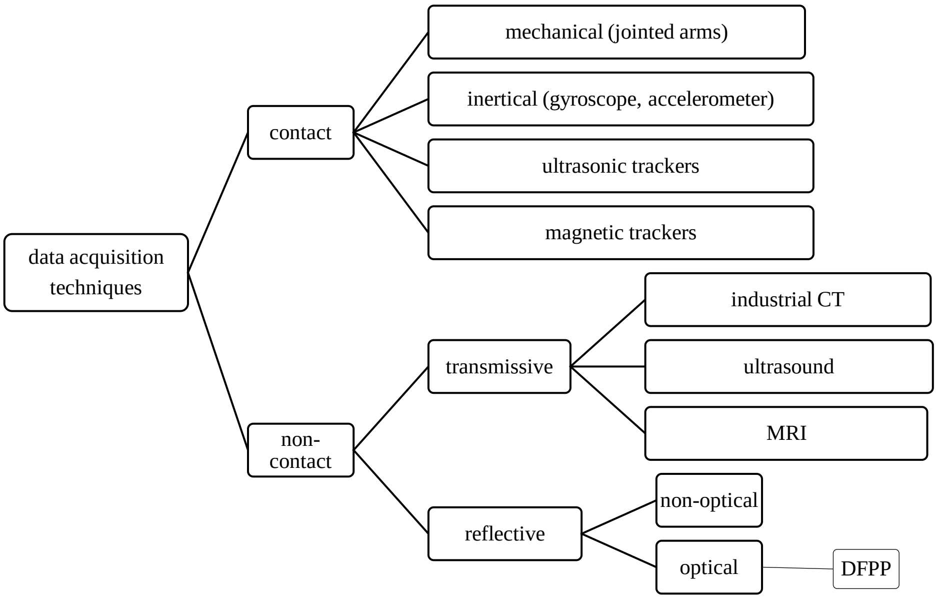


Figure 2.1: Range image acquisition techniques.

2.2 3D Face Recognition

2.2.1 Introduction

It is known that researchers[6-8] have already given the surveys of existing face recognition methods. Gökberk, *et al*[6] compared the 3D face recognition methods by using 3D point coordinates, surface normals, curvature-based descriptors, 2D depth images, and facial profile curves. A decision level fusion technique is analyzed to classify the feature descriptors. Their literature only focused on the methods of feature descriptors based methods while Bowyer, *et al*[7] listed the existing face recognition methods not only feature descriptors based methods but also template-based face recognition methods. The methods they mentioned are those of using multiple models: 2D intensity images and 3D face images. Smeets, *et al*[8] reviewed the 3D face recognition methods with emphasize on dealing with facial expressions.

This section gives the review of face recognition methods based on range images. The height field of a range image is given in Figure 2.2. A 3D surface is placed on the top of the reference plane. The arrows in this figure are denoted as the height value of the surface. Range images could be presented in grey scale as shown in Figure 2.3. The range image gives direct, explicit geometric information and invariant to variations in lighting and viewpoint [9, 10]. As a result, using range images to achieve the face recognition is better than 2D image.

The next three sections categorize the 3D face recognition methods based on feature-based matching methods, template-based matching methods and multi-modal matching methods.

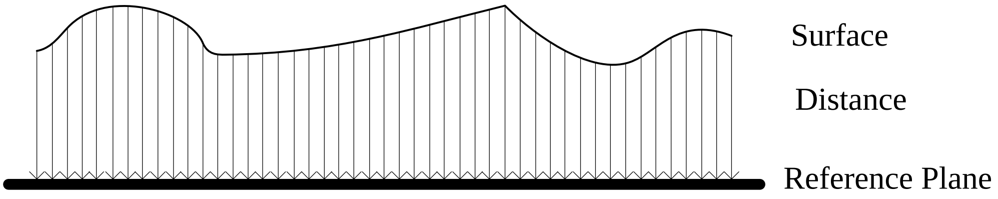


Figure 2.2: Height field of a range image[10].

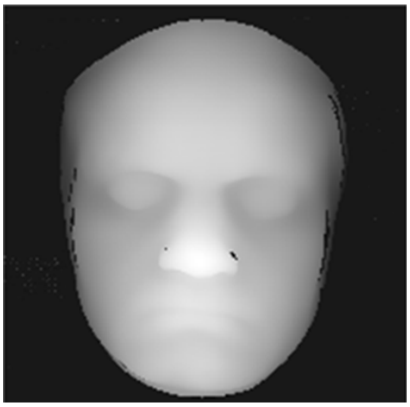


Figure 2.3: An example of a range image comes from the database (shows in grey scale).

2.2.2 Feature-Based Matching Methods

The methods of using features extracted from the face surface to achieve face recognition, are reviewed in this section. The shape information contained in one object surface can be highly compressive as features. The feature-based matching methods can be divided into two main categories: local feature-based methods and global-feature based methods.

Local features consist of surface curvatures, surface shape types, surface normal, and the angles between different surface normals. Researchers either base matching on one of these local features or combine them together so as to compare 3D face images. How to choose the local features is always a problem of ambiguity. The reason is that the local features are required to cover the uniqueness and completeness of the whole 3D image but avoid redundant information.

Global feature-based methods are compressing the information of the whole 3D face image. Principal Component Analysis (PCA) based methods and Extended Gaussian

Image (EGI) based methods belong to this category. PCA is a highly compressive method which reduces the dimension of the 3D face image, by applying PCA, one 3D face image can be described as a vector which will be easy to compare with other vectors. EGI based methods present the surface of 3D face images as surface normals, and the correlation between each 3D image is easy to compute.

1. Local Feature Based Methods

i. Curvature Based Methods

This section describes the face recognition methods of curvature based methods. A surface in a 3D space can be recognized by its intrinsic curvatures. Several kinds of curvatures, such as principal curvatures, mean curvature (H) and Gaussian curvature (K), could be used as a tool to realize the face recognition. The curvature based methods usually segment a face surface into several different curves. Besl[11] in 1986 categorized the free-form curves into eight different curves: peak, pit, ridge, valley, pat, minimal, saddle ridge and saddle valley surfaces. Whereas Dorai, *et al*[12], in 1997, segmented the curves into nine curves, which are spherical cap, dome, ridge, saddle ridge, saddle, saddle rut, rut, trough and spherical cup. This segmentation method has been utilized by many researchers to segment the surface type. Figure 2.4 illustrates the segmentation of several different curves by Dorai, *et al*[12].

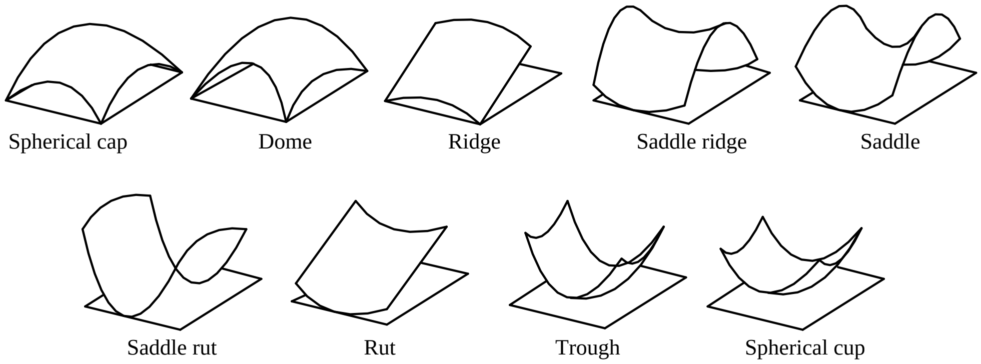


Figure 2.4: Nine types of shape [12].

Before reviewing the literature, some definitions are listed below:

- Normal curvature: denote C as a non-singular curve on a smooth surface I, denote T as the tangent vector of C at a point p, denote $\boldsymbol{n}$ as the surface normal. The normal curvature N at a point p is the curvature of the curve projected onto the plane containing the tangent T of the curve and the surface normal.

- Principal curvatures: point p on surface I will have many curvatures, the principal curvatures consist of the maximum and minimum curvatures.

- Mean curvature: the mean curvature H at p is the average curvature of the maximum curvature and minimum curvature.

- Gaussian curvature: the Gaussian curvature K at p is the product of the maximum curvature and minimum curvature.

Cartoux, *et al* in 1989[13] proposed a method based on principal curvatures to achieve face recognition and authentication. They used the property of quasi-symmetry in the human face to segment a face into two parts, and then found a bilateral symmetry plane.

The nearest neighbor rule was used to classify the different faces and from their report[13], we know that it is a robust method because the result they post achieved 100% classification. In 1990s, Lee, *et al*[14] also used Gaussian curvature to segment the original range image into a set of different local curvatures, and compared two sets of segmented curvatures rather than two whole face images. In 1992, Gordon[15] divided the face image into two level features: the high level features which were the salient parts in the face image contained eyes, nose and head, and the low level features which were the properties of the nose, eyes and head, for example, the eye width, the maximum Gaussian curvature on the nose bridge and the head width.

In 2001, Kim, *et al*[16] presented a method to extract features by using principal curvatures. They segmented the face image after real time normalization in order to find feature points of mouth, nose and eyes. In 2002, Campbell, *et al*[17] used the principal curvatures to determine the salient surface segments, then divided these segments into sub-regions[17], finally, the sub-regions set the unit of the surface. Moreno, *et al*[18] segmented pronounced curvatures based on HK curvatures (mean and Gaussian curvatures). According to the HK segmentation method[18], a point can be used as a sign for classification. In that paper[18], three kinds of points were used: Hyperbolic points ($K < 0$), Elliptical convex points ($H < 0$ and $K > 0$) and Elliptic concave points ($H > 0$ and $K > 0$). Bhanu, *et al*[19] used a scale-space filter to analyze the curvature based fiducial extraction and then set the face profile for the comparison. Sun, *et al*[20] used principal curvatures to classify the different surface types and improved the result of Principal Component Analysis (PCA) based methods by Hesher, *et al*[9]. In 2006, Chang, *et al*[21] proposed a method based on HK curvature which can achieve recognition under varying facial expression. They only used the region of nose to do the matching, while Colombo, *et al*[22] used HK curvature with the nose and eyes regions to do the face registration. Akagündüz, *et al*[23] extracted interest points based on HK curvatures in 2009. They improved the HK curvature obtained from scale space so that it was invariant to scale and orientation.

ii. Point Signature Based Methods

Point signature is a descriptor, which compresses the characters of a certain mass of 3D surface. The characters can be any information of the surface, for example, curvature, shape variation, angle difference at a specific point to the normal vector, and distance difference, etc. In order to enable the comparison of these characters more easily, researchers always represent the characters as a 1D histogram or 2D histogram. The researchers who use these feature descriptors to achieve the recognition are listed in this section.

It was first presented by Chua, *et al*[24] in 1997. The definition of the point signature is[24]: Given a point p at a 3D surface, known with normal vector $\boldsymbol{N}$, we could build a sphere with radius r and center p. The intersection of the sphere and the 3D surface is a 3D curve denoted as C. After setting a plane P which is perpendicular to the normal vector on point p and projecting the curve C onto plane P (Figure 2.5 (a)), we will get a new curve C' which is a circle on plane P and a reference vector $\boldsymbol{n_r}$, the projection distance from C' to plane P is a signed distance profile[25] (Figure 2.5 (b)). The corresponding distance to every angle is shown in Figure 2.5 (c). Each point on curve C can be described as two properties[25] (Figure 2.6):

- The signed distance from itself to the corresponding point on curve C'.
- A clockwise rotation angle θ about $\boldsymbol{N}$ from the reference direction $\boldsymbol{n_2}$.

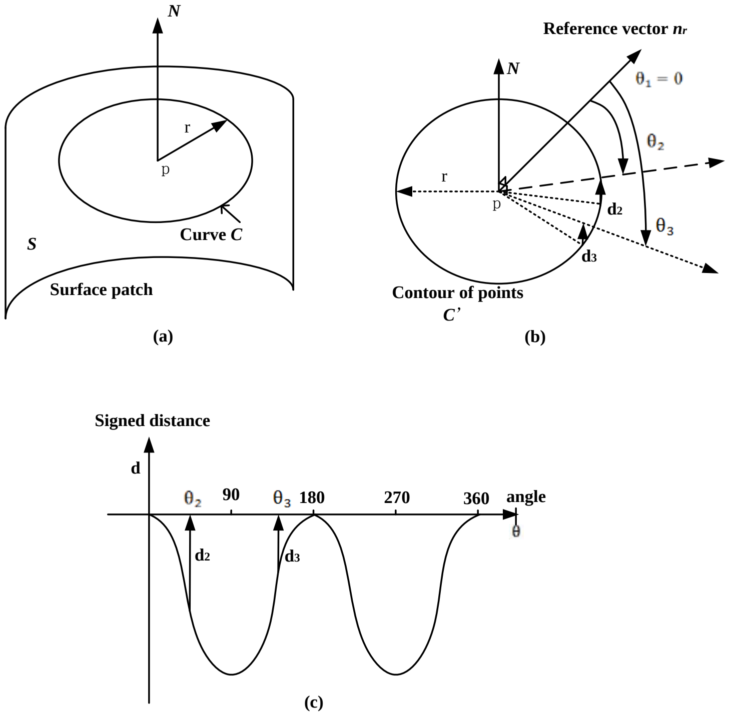


Figure 2.5: Point signatures: (a) contour of points at a fixed radius, (b) reference direction, (c) signature of distance profile from translated fitted plane[24].

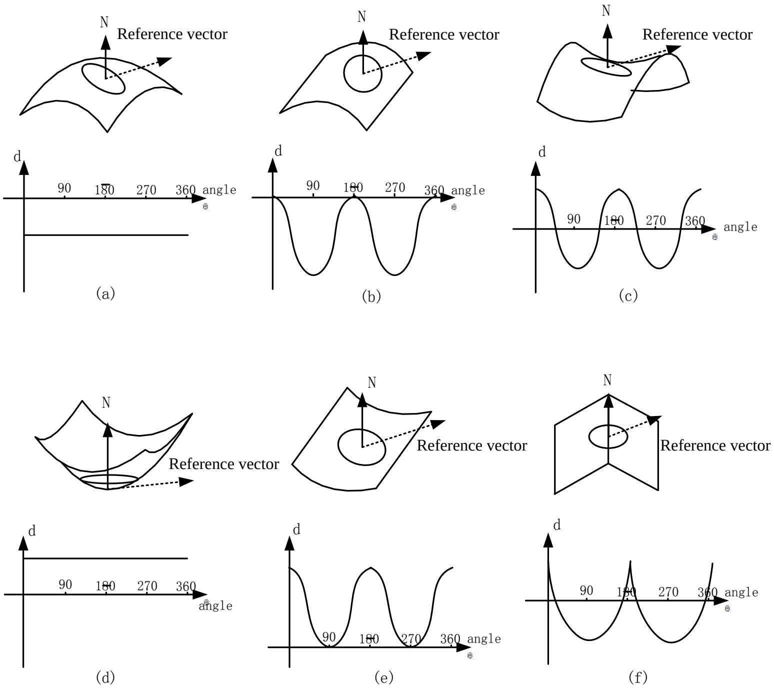


Figure 2.6: Examples of point signatures: (a) peak, (b) ridge, (c) saddle, (d) pit, (e) valley, (f) roof edge[24].

After the point signature presented in 1997, Chua, *et al*[25] proposed a method for 3D face recognition based on point signature. Because of the facial expressions, the whole face surface is a non-rigid surface, so they extracted the rigid parts from the whole face surface. In Figure 2.7, the face images of different facial expressions and rotations are given from Chua's research while the results of rigid parts are illustrated in Figure 2.8.

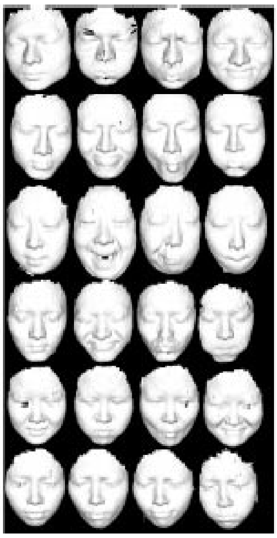


Figure 2.7: Face images of different facial expressions and rotations from Chua's research[25].

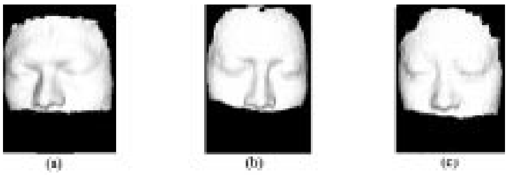


Figure 2.8: The lower faces (a)(b)(c) are the extracted rigid face regions of each of the first three persons in upper faces[25].

Lowe[26] presented a method that extracts distinctive invariant features from images, which can be used to perform reliable matching between different views of an object or scene in 2004. Lowe's method is called Scale Invariant Feature Transform (SIFT), as it is invariant to illumination and scale changing. SIFT is representing one key point by using a 16×16 window to represent the orientations around it, which can be treated as a feature descriptor. A Gaussian weight function is also used to control the orientations in different points around the key points, the nearer the key points, the higher the value of the weight factor. Figure 2.9 is shown as an example of the descriptor computation in one key point. In 2009, Lo [10] presented a method which is called 2.5D SIFT. This method adds two more elements (slant and tilt) to one SIFT descriptor which can be used in 3D images.

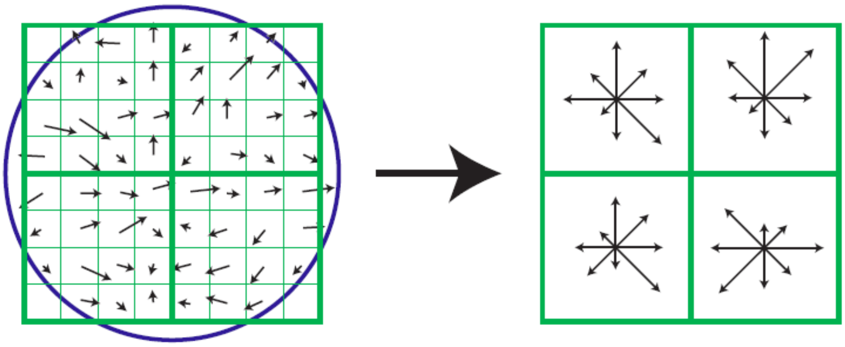


Figure 2.9: One keypoint descriptor. In the left side image, it shows the gradient magnitude and orientation around one keypoint. The circle is denoted as the Gaussian window. On the right side image, it shows that the gradient magnitude and orientation are summed up in every 4×4 region. This figure gives a 2×2 descriptor compressed 8×8 samples area. [26]

Other researchers used other information about the surfaces as the feature descriptors. Xu, *et al*[27] used the feature vector to describe shape variation information of the area in mouth, eyes and nose. Shan, *et al*[28] used shapeme histogram, which uses histogram of shape signature or prototypical shapes to do the recognition on partially observed query objects. Huang, *et al*[29] used the histogram proportion of depth differences to compare the various face images.

2. Global Feature Based Methods

i. Principal Component Analysis (PCA) Based Methods

The dimension reduction technology is used in some of 3D face recognition methods, for instance, Principal Component Analysis (PCA) method. PCA method is a vector dimension reduction method, which is to find out the principal component of a vector and remove the noise and redundancy, in order to simplify a complex dimensional data set into a low dimensional data set. PCA was first used in face recognition in the year 1991 by Turk, *et al*[30] in 2D images. The main idea of PCA can be concluded as follows:

Let the points be $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots, \boldsymbol{x_N}$, each of the point has the *m*-dimension, PCA aims to find out a vector which can be described with K variables, K < *m*, using the steps below:

✧ Find out the average vector:

The average vector can be calculated by: $\boldsymbol{\mu} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \boldsymbol{x_i}$,

✧ Find out the covariance matrix of the differences of mean:

The covariance matrix is $\boldsymbol{C} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^{\boldsymbol{T}}$, where $(\boldsymbol{x_i} - \boldsymbol{\mu})$ is the difference of mean for vector $\boldsymbol{x_i}$.

✧ Find out the eigenvectors and eigenvalues of the covariance matrix $\mathbf{C}$:

Note that a unit vector $\boldsymbol{v}$ as $\boldsymbol{v}(\boldsymbol{x})$, the value of $\boldsymbol{v}$ on the i-th data point can be represented as $\boldsymbol{v}(\boldsymbol{x_i}) = \boldsymbol{v}^{\boldsymbol{T}}(\boldsymbol{x_i} - \boldsymbol{\mu})$, the variance of v can be easily calculated by:

$$\mathrm{var}(\boldsymbol{v}) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}(\boldsymbol{x_i})\boldsymbol{v}(\boldsymbol{x_i})^{\boldsymbol{T}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}^{\boldsymbol{T}}(\boldsymbol{x_i} - \boldsymbol{\mu})\left(\boldsymbol{v}^{\boldsymbol{T}}(\boldsymbol{x_i} - \boldsymbol{\mu})\right)^{\boldsymbol{T}}$$

$$= \boldsymbol{v}^{\boldsymbol{T}}\{\textstyle\sum_{i=1}^{N}(\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^{\boldsymbol{T}}\}\boldsymbol{v} = \boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{C}\boldsymbol{v}, \qquad (2.1)$$

where the diagonal matrix of eigenvalues of $\mathbf{C}$ is denoted as $\mathrm{var}(\boldsymbol{v})$ and matrix $\boldsymbol{v}$

contains m eigenvectors of covariance matrix $\mathbf{C}$ and the length of the eigenvector is also m.

✧ Arrange the eigenvectors:

In this step, we need to put the eigenvectors as a decreasing eigenvalue order.

✧ Select appropriate value of K:

In order to select K as small as possible, the analysis of the distribution for each eigenvector is firstly required. We can use cumulative energy to decide the distribution for each eigenvector: $\mathrm{g}[m] = \sum_{q=1}^{m} Dis[q,q]$, and then, if we need the eigenvectors containing 90% information of the whole vector, we decide K from the equation: $\frac{\mathrm{g}[m=K]}{\sum_{q=1}^{m} Dis[q,q]} \geq 90\%$.

In 2003, Hesher, *et al*[9] presented the PCA method to reduce the dimensionality of range images, and used the nearest neighbor rule to realize the identification part, but this approach was not idealistically stable due to noise. The computational cost of PCA based methods is very low, nevertheless, the results of PCA are not accurate on the basis of the experiment results. Many researchers employ the PCA method in combination with other efficient methods to ensure accuracy. Blanz, *et al*[31] used a 3D based PCA method in order to reach pose invariant. Chang, *et al*[32] presented the PCA method based on 2D and 3D face images in 2003. Russ, *et al*[33] used 3D face alignment for PCA. Li, *et al*[34] proposed an approach in which PCA is used as a tool to achieve face normalization and the curve extraction, and after that, Iterative Closest Point (ICP) method is utilized to make the comparisons.

ii. Extended Gaussian Image (EGI) Based Methods

Gaussian Image is such an image that maps the surface normal for every point on the 3D surface into a unit sphere (Gaussian sphere), the tail of which lies in the center of the unit sphere whereas the head lies on the surface. Extended Gaussian Image (EGI) can be obtained by placing a mass at each point equal to the surface area of the corresponding face[35]. The EGI is described in Figure 2.10 as below.

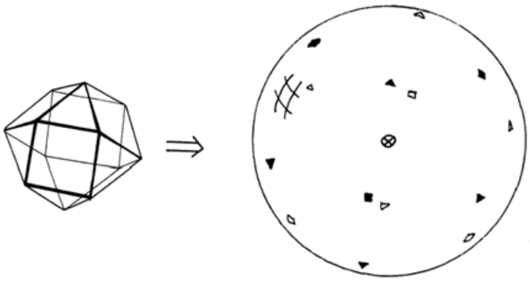


Figure 2.10: Extended Gaussian Image can be thought as a collection of point messes on the Gaussian sphere. Each mass is proportional to the area of the corresponding face. Point masses on the visible hemisphere are solid mark and while others are open ones. The center of mass must be the center of unit sphere which is described as a little cross in the figure. [35]

Lee, *et al*[14] is the first research group using EGI as the method for 3D face recognition. They utilized the mean and the Gaussian curvatures to segment the face into different convex regions. Then each convex region was represented onto the unit sphere which forms an Extended Gaussian Image was represented. They also interpolated the EGI since all of the face surface types cannot be included in the convex regions. At last, they used a graph matching algorithm based on the correlation matrix between different regions to realize the matching part. In 1998, Tanaka, *et al*[36] proposed a method based on principal curvatures using a 3D vector sets correlation approach[36]. This approach didn't require any local feature extraction or segmentation and the method of comparison was on the basis of Fisher's[37] spherical correlation on EGI. A problem existing in EGI based methods is that EGI is not sensitive to image scale invariance, so the two images will not be distinguished via EGI based methods if the two images contain the same object but different size[38].

2.2.3 Template-Based Matching Methods

This section reviews the face recognition methods which are based on the coordinates of 3D face images. 3D images have enough coordinates information so that researchers can use the coordinates to directly compare the difference. The algorithm of template matching methods is simple to understand, and widely applicable to different types of 3D images, for example, range images. Both Iterative Closest Point (ICP) based methods and Hausdorff distance based methods belong to this category.

1. Iterative Closest Point (ICP) Based Methods

Iterative Closest Point (ICP) is used for registration curves or free-form surfaces, as it is an efficient method for the 3D data reconstruction and the 3D face recognition. After giving a pre-estimated motion value (always the translation matrix $\boldsymbol{t}$ and rotation matrix $\boldsymbol{R}$ of the two point sets), the two point sets could be aligned within several iterations. The ICP method is an iterative algorithm that will converge monotonically to the nearest local minimum of a mean-square distance metric[39]. Several methods can be used in the ICP method to compute the translation and rotation between the probing image and the image in the database, such as Singular Value Decomposition (SVD)[40], orthonormal matrices[41], unit quaternion, and dual quaternion (DQ)[42]. The results of these four methods are almost the same. More information about the comparison of these four methods can be found in [43] proposed by Eggert in 1997.

The algorithm of ICP can be described like this:

The ***INPUT*** of ICP: two 3D point sets, one is model point set $\boldsymbol{x} = \{\boldsymbol{x_i}\}$, i=1,2,...l. l is the number of points in $\boldsymbol{x}$; the other is the reference point set D'; the initial value of rotation $\boldsymbol{R}^0$ and the translation $\boldsymbol{t}^0$.

The ***OUTPUT*** of ICP: an optimal motion consists of a rotation matrix $\boldsymbol{R}$ and a translation matrix $\boldsymbol{t}$ between the two point sets.

Iteration begins:(in I-th iteration)

Step 1: Find the closest points $\boldsymbol{y_i}^I$ $(i = 1,2, \ldots, l_m)$ in reference D' corresponding to every point from model $\boldsymbol{x_i}^I$.

Step 2: Compute the motion between two point sets (by using either of the four methods to compute $\boldsymbol{R}^{I}$ and $\boldsymbol{t}^{I}$).

Step 3: Apply the motion to the model.

The iteration ends if:

The condition of termination is to satisfy the termination constraint. For the terminate condition, if $\|\boldsymbol{R}^{I}-\boldsymbol{R}^{I-1}\|<\varepsilon_R$ and $\|\boldsymbol{t}^{I}-\boldsymbol{t}^{I-1}\|<\varepsilon_t$, the iteration ends, where $\varepsilon_R, \varepsilon_t$ are the thresholds for rotation and translation. For face recognition, the mean square distance between the two point sets should be compared in order to recognize whether the two point sets are from the same person or not. If the distance is smaller than a threshold, these two images are treated as the genuine face, otherwise these two face images are from different persons.

ICP is a method which is very popular and used in surface registration and object recognition. Besl, *et al*[39] created the ICP for surface registration in 1992. This method is using the distances between two points corresponding to the two point sets to realize the surface location and recognition, but the drawback of this method is that it is only available if the first surface is one part of the second surface or the two point sets are totally overlapped. In the year 1994, Zhengyou Zhang[44] improved Besl′s method so that it can be used no matter the first surface is a sub-surface of the second one or not.

Medioni, *et al*[45] built a 3D face database and tested their database to realize face recognition based on ICP method in 2003. The database they built was presented with 3D surfaces. Lu, *et al*[46] built a coarse-to-fine alignment strategy method by using Besl′s scheme to do the coarse alignment and Chen′s scheme as the fine alignment. Chang, *et al*[47] presented a new adaptive rigid multi-region selection method, which compared the independent multiple regions of the face surface. Then the results of the multiple matches were combined. Amor, *et al*[48] also used the coarse-to-fine strategy to do the recognition. They computed the rigid transformation of the two models and brought them together[48] for the first step, and in the second step they used the ICP method. In 2009, Tong, *et al*[49] used an anthropometric face model to estimate the face region which was a fast local region detection method, and an extension of ICP method was proposed to do the matching. They added an intensity coordinate i as the fourth-dimensional which formed a 4D point (x, y, z, i), the first three (x, y, z) coordinates were the spatial coordinates.

The traditional ICP method cannot, however, handle such a case as the scales of the probing image and the reference image are different. Due to this problem, researchers have proposed several methods to solve it. In 2000, Zha, *el al*[50] utilized the extended signature images to establish the correspondence between the two images no matter whether these two images were in the same scale or not. Then the scale parameter (a scalar) can be computed by corresponding mean curvatures. Zinßer, *et al*[51] in 2005 estimated a scale factor between the two point sets in every iteration. At the same time, Ko, *et al*[52] used the ratio of the normal curvatures on two point sets as the scale factor. Du, *et al*[53], in 2007, proposed a method where the scale was described as a 3×3 scale matrix, which had a boundary in order to avoid the phenomenon that the points of a set converge to a point of the other set. Du, *et al*[54] improved their method in order to estimate the initial parameters by using the eigenvalues and eigenvectors of covariance matrices of point sets in 2010.

2. Hausdorff Distance Based Methods

In comparison with ICP, in which the distance between point to point is used, it is known that Hausdorff distance is another distance calculation by which the distance between two point sets is calculated. In other words, Hausdorff distance does not need to compare the distance between point to point but point to plane.

Achermann, *et al*[55] proposed a method using an extension of Hausdorff distance matching in 1997 and the result they reported could get a 100% recognition rate in some situations. Lee, *et al*[56] built a system which was based on depth-weighted Hausdorff distance using the principle of mean and Gaussian curvatures in the year 2004. They combined the depth information and local curvature features together to do the person verification. Russ, *et al*[57] used an iterative method to update the two point sets, the constraint condition can be controlled by two variables: the ratio in set Y within a distance vale of point in set X, it is used to control the corresponding point, and the Hausdorff distance with the weight factor of the ratio as is used to fit the quality.

2.2.4 Multi-Modal Matching Methods

The 3D image processing techniques have become more and more popular and researchers have started to use 3D surface to do the face recognition since 1990. There is no doubt that the 3D face recognition could get a more accurate result because much more information is contained in 3D images compared with 2D images. Some researchers have proposed methods combined 2D face recognition methods and 3D face recognition methods together in order to get a more robust result since the 2D face recognition techniques have become mature these years.

Beumier, *et al*[58] used multi-modal recognition, they fused the 2D and 3D lateral profile and central profile together to increase the verification performance. Bronstein, *et al*[59] required a range image (geometry) and a 2D image (texture) to produce two sets of eigen decompositions to the flattened textures and the canonical images are employed to solve facial expressions. Chang, *et al*[32] used a PCA based method both in 2D and 3D images, based on their experiment results, sole 2D or 3D PCA based methods cannot give the accurate results while the combined 2D and 3D PCA method proved much better. They used a confidence-weighted variation of the sum-of-distance rule for the distances of 2D images and 3D face images. Tsalakanidou, *et al*[60] developed the approach based on range image and color information which is the color component (YUV) rather than intensity. Similarly with Chang, *et al*[32], they preferred the PCA method to do the face recognition. Godil, *et al*[61] also used a PCA based method to do color and 3D face recognition but the result was worse than Chang, *et al*[32] because the data in the database contained more grid points and different normalization method. In 2005, Chang, *et al*[47] concluded that multi-modal based method can give the best result than 2D modal alone or 3D, and the fusion of several 2D modals together can also produce a good result. Papatheodorou, *et al*[62] employed a 4D face data to do the face recognition. The 4D face data is a 3D geometry and a 2D texture map. They used ICP method with the 4D face data $(x, y, z, \alpha t)$, which α is the weighted factor implied in the contribution of the texture, so that the closest points were not only in the 3D images but also the textual difference. Mian, *et al*[63] at 2007 presented a method which is efficient and robust to facial expressions based on ICP method[63].

2.3 Summary

In consideration of this chapter in which the 3D face recognition methods have been categorized and reviewed, 3D face recognition can be divided into the following categories: feature-based matching methods; template-based matching methods and multi-modal matching methods.

Feature-based matching methods are the fast matching methods widely used by many researchers. For example, in Lo, *et al*[64] 2.5D SIFT method, only about thirty points are extracted from a face image to recognize, by which a lot of time has been saved for computation. However, on the other hand, only such a few points extracted cannot contain all the face information. It is not safe to extract only a few points to compare the face images. A good way to use feature based methods is to combine these methods with other methods in order to achieve a high accurate result. In this thesis, the curvatures are used to select the feature points. Although there are only certain types of surface by using the curvatures to classify, the typical feature areas on the face images can be used so that the curvatures information is enough to compare. In addition, since the curvatures are the basic methods for surface feature analysis, it is easy to realize and speed up the computation of the curvatures. More details will be given in Chapter 4.

Template-based matching methods are those where the whole face surface is matched without feature analysis. ICP is such a method that selected by many researchers as a baseline to compare with other methods. It is a very accurate method for image registration and object recognition. One drawback of traditional ICP[39, 44] is that it cannot handle the 3D images with different scales. Du, *et al*[54] proposed a method in which the scaling effects of the images have been considered. They estimated the scale factor by using the coordinates of the points on the face surface. However, they didn′t clarify that whether their method can deal with the partially-overlapped problem or not.

Multi-model matching methods are the most accurate methods, but more resources are required of these methods compared with others. The algorithms of these methods will be more complex than other 3D image based methods since these methods need both 2D image and 3D image to compare the face images.

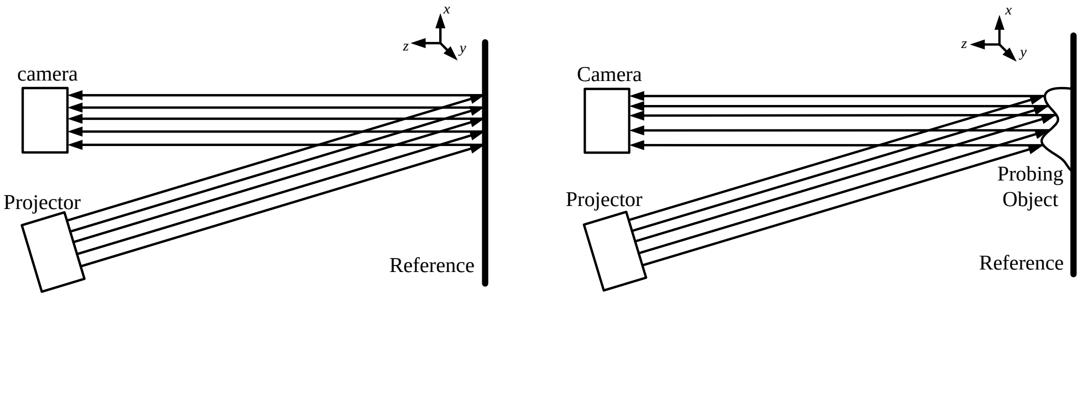
CHAPTER 3 DATA ACQUISITION SYSTEM AND DATABASE

A 3D data acquisition system based on Digital Fringe Projection Profilometry (DFPP) is described in this chapter. It is an optic technique to obtain 3D images. A 3D face database which consists of 90 3D face images obtained from this DFPP system is built up. This 3D face database is used for 3D face recognition. The whole structure and procedure of the DFPP system are given in Section 3.1. The 3D face database is described in Section 3.2.

3.1 DFPP System

3.1.1 Introduction

DFPP based 3D data acquisition system is an effective system of the non-contact 3D shape measurement. A DFPP based system consists of a CCD camera, a digital video projector and a computer, which is a simple 3D data acquisition system. The structure of the DFPP based system is illustrated in Figure 3.1.

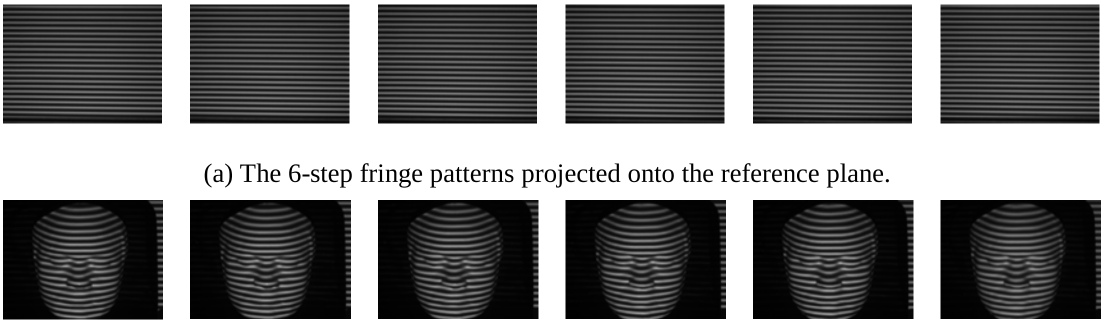


(a) The fringe patterns projected onto the reference plane.

(b) The fringe patterns projected onto the object.

Figure 3.1: Structure of DFPP based system.

A DFPP based system projects pre-designed multiple fringe patterns onto a reference plane and then projects onto the probing object placed on the reference plane, both the fringe patterns on the reference plane and the deformed fringe patterns on the probing object can be collected by the camera. After the analysis of the two different groups of multiple fringe patterns, the height of the probing object surface to the reference plane can be retrieved. Because of the recent advanced developed digital projection technology, DFPP based systems are widely used because their structure is simple and the cost is low[65]. Figure 3.2 shows the six-step fringe patterns projection. Figure 3.2(a) shows the six steps fringe patterns projected onto the reference plane and Figure 3.2 (b) shows the same six-step fringe patterns projected onto the object. The reconstruction result is indicated in Figure 3.3 and the light effect is added to enhance the face image clearly. The reconstruction result gives the precise details of the face by using the data acquisition system.



(a) The 6-step fringe patterns projected onto the reference plane.

(b) The 6-step fringe patterns projected onto the probing object.

Figure 3.2: Six steps fringe patterns projection.

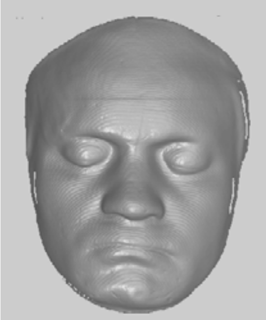


Figure 3.3: Reconstruction result (by adding the light effect).

3.1.2 Procedure of the DFPP System

The procedure of the DFPP system can be concluded as three steps: fringe pattern projection and acquisition, fringe pattern analysis and calibration.

1. Fringe Pattern Projection and Acquisition

To obtain the surface of one object via the DFPP based data acquisition system, a group of fringe patterns to project have to be well-designed. The optic and mechanical systems can both create the fringe patterns in 1990s, such as a slide projector or interference of two laser beams, and the patterns are either Ronchi or sinusoidal gratings[66]. A lot of patterns can be generated and implemented in 2000s due to the software-driven nature associated with DFPP, and the fringe patterns have various selections such as trapezoidal patterns[67], triangular patterns[68, 69] and saw-tooth patterns[70].

The projector also needs to be determined. A projector which has a high contrast ratio, less screen door effect in order to reach the high measurement resolution is suitable for the DFPP system. The projector also requires a fast switching capability. There are two kinds of popular projectors available in the market, Digital Light-processing Projects (DLP) projector and Liquid Crystal Display (LCD) projector. DLP projector performs better than LCD projector, with 0.0032 higher average accuracy and 0.0060 smaller standard deviation, proposed by Yen, *et al*[71]. Gong, *et al*[72] reported that an off-the-shelf DLP projector could reach fast image switching of 120 frames per second in 3D shape measurement.

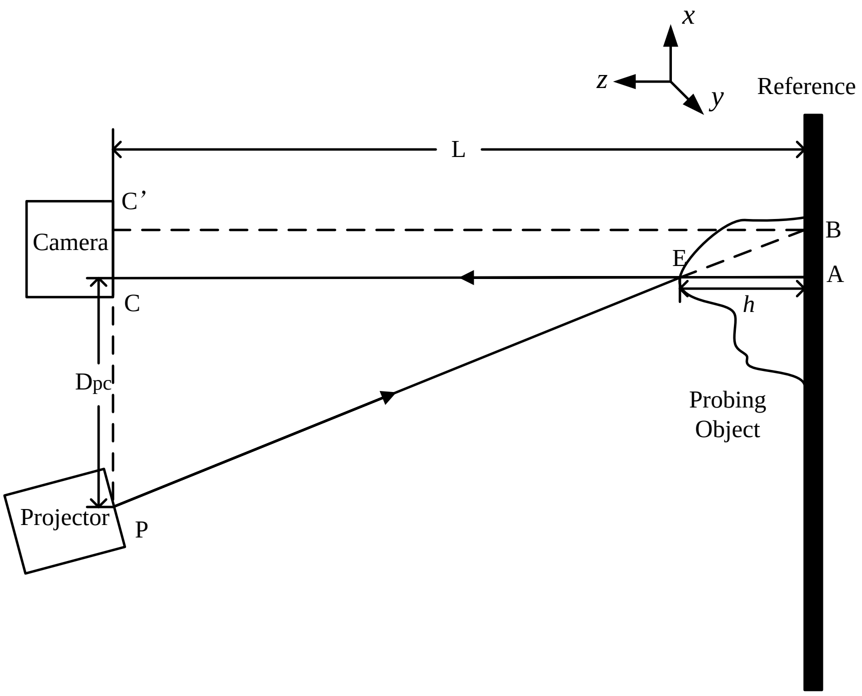


Figure 3.4: The ideal optical model of DFPP based system.

The ideal optical model of the DFPP based system is demonstrated in Figure 3.4. It assumes that the light beams projected from the projector and collected in the camera are all parallel light beams, and the responses from the camera and projector are linear. x axis is along the direction from the projector to the camera and z axis is vertical to the reference plane and point to the camera in Figure 3.4. The fringe patterns from the point P are projected to one point E on the object, and these fringe patterns will be collected by the camera at point C. The projector generates the fringe patterns with an input signal $s_0(x)$. $s_0(x)$ is a 2D signal. $s_0(x)$ is projected onto the object and reflected at point E in Figure 3.4. The fringe patterns shift along the *x* direction. The reflected fringe patterns of $s_0(x)$ will be collected at point C on the camera. If no object is placed on the reference, the reflected fringe patterns will be collected at point C′. The shift of the image from camera is $\overline{CC'}$ and the fringe shift is $\overline{AB}$. L is the distance between camera and reference plane, D_{pc} is the distance between the camera and the projector, h is the height of at point E on the object.

The sinusoidal signal is used as the input signal in this thesis,

$$s_0(x) = cos(2\pi f x), \tag{3.1}$$

where f denotes as the frequency of $s_0(x)$.

From Figure 3.4 it is easy to get that $\triangle ABE \sim \triangle CPE$. We get the relationship below:

$$\frac{h}{\overline{AB}} = \frac{L-h}{D_{pc}}. \tag{3.2}$$

Hence, the height value h is:

$$h = \frac{\overline{AB} \cdot L}{D_{pc} + \overline{AB}}. \tag{3.3}$$

We can extend $\overline{AB}$ and h to all field along x, rewrite Equation (3.3), we will have

$$h(x) = \frac{\delta(x) \cdot L}{D_{pc} + \delta(x)}, \tag{3.4}$$

where $\delta(x)$ is the fringe shift replaced from $\overline{AB}$ and $h(x)$ is the height value replaced from h.

Deformation of the fringe patterns happens when the object is placed onto the reference plane. In the next step, we discuss how to analyze the deformed fringe patterns to determine the height value of the object.

2. Fringe Pattern Analysis

In this step, the method of how to calculate the height values of the 3D surface based on the fringe pattern deformation is presented. The output from the projector can be denoted as:

$$I_1(x) = u_1[s_0(x)]. \tag{3.5}$$

Here, a response function of the projector is obtained which is denoted as u_1, since the assumption that the response of the projector is linear, u_1 can be denoted as:

$$u_1(x) = a_1 x + b_1. \tag{3.6}$$

After the lights propagation, the lights on the object surface are:

$$I_2(x) = a_2(x) I_1(\alpha x) + b_2(x), \tag{3.7}$$

where $a_2(x)$ is the light propagation attenuation, $b_2(x)$ is the background light between the projector and the surface.

After the reflection on the surface, the light becomes:

$$I_3(x) = r(x)I_2(x), \tag{3.8}$$

where $r(x)$ is the reflection factor.

Another light propagation attenuation happens between the reflected lights and the camera which is denoted as follows:

$$I_4(x) = a_4(x)I_3(x) + b_4(x). \tag{3.9}$$

The output signal obtained from camera can be written as:

$$s(x) = u_2\big(I_4(x)\big), \tag{3.10}$$

where u_2 is the response of the camera which is written as:

$$u_2 = a_5 x + b_5. \tag{3.11}$$

Combining the equations from (3.5) to (3.11), we have

$$\begin{aligned} s(x) &= a_1 \cdot a_2(x) \cdot r(x) \cdot a_4(x) \cdot a_5 \cdot s_0(\alpha x) + b_1 + b_2(x) + b_4(x) + b_5 \\ &= A(x)g(x) + B(x), \end{aligned} \tag{3.12}$$

Where $g(x) = s_0(\alpha x)$, $A(x) = a_1 \cdot a_2(x) \cdot r(x) \cdot a_4(x) \cdot a_5$, and $B(x) = b_1 + b_2(x) + b_4(x) + b_5$.

The deformed fringe pattern can be calculated by:

$$s_d(x) = A'(x)g\big(x - \delta(x)\big) + B'(x). \tag{3.13}$$

The symbol $'$ means the real illumination of the acquiring environment reflectivity $A(x)$ and background $B(x)$ will be changed.

The output signal (the signal from the camera) is:

$$s(x) = A(x)\,cos\,\phi(x) + \ B(x). \tag{3.14}$$

The deformed fringe pattern is:

$$s_d(x) = A'(x)\,cos\left(\phi\big(x - \delta(x)\big)\right) + \ B'(x). \tag{3.15}$$

The phase of the sinusoidal signal is denoted as $\phi(x)$ which can be written as:

$$\phi(x) = 2\pi f \alpha x = 2\pi f_0 x, \tag{3.16}$$

where f_0 is the fringe frequency on the reference. The phase difference between $s(x)$ and $s_d(x)$ is:

$$\triangle\,\phi(x) = \phi(x) - \phi\big(x - \delta(x)\big). \tag{3.17}$$

The phase shift is featured with a relationship with the phase difference:

$$\delta(x) = \frac{\triangle\phi(x)}{2\pi f_0}. \tag{3.18}$$

The height is:

$$h = \frac{\delta(x)\cdot L}{D_{pc}+\delta(x)} = \frac{\triangle\phi(x)L}{2\pi f_0 D_{pc}+\triangle\phi(x)}. \tag{3.19}$$

The key to determine the surface height of the object is $\delta(x)$.

The methods used to analyze the fringe patterns and reconstruct the 3D shape based on phase detection can be divided as: Fourier Transform Profilometry (FTP), Phase-Shifting Profilometry (PSP) and Phase Unwrapping.

However, the sinusoidal or periodic input signals are required as the fringe patterns on the basis of phase detection methods, it is difficult to realize in practice because nonlinear intensity distortion inherent to digital video projectors. The shift detection based methods can solve this problem such as the double three-step phase-shifting algorithm[73] and gamma curve estimation look up table[74]. Shift detection based method utilizes the spatial shift maps rather than phase maps of the fringe patterns, and can avoid the nonlinear distortion. As a result, it does not require the pure sinusoidal or periodic fringe patterns, so a wide use range of light patterns can be used such as sinusoidal, triangular, and saw-tooth with or without distortions.

Here, a method which is used to extract fringe phase with multiple fringe patterns is given. Phase Shifting Profilometry (PSP) method is used in the acquisition system. It was first proposed by Srinivasan [75].

The multiple fringe patterns projected from the projector can be described like this:

$$s_n = A(x)\,cos\left(\frac{2\pi n}{N_{step}} + \phi(x)\right) + B(x), n = 1,2,3, \ldots, N_{step}, \qquad (3.20)$$

where N_{step} is the number of the fringe patterns projected onto the object, it is also the step number of the PSP. The phase $\phi(x)$ can be calculated by:

$$\phi(x) = tan^{-1}\frac{\sum_{n=1}^{N_{step}} s_n(x)\,sin(2\pi n/N_{step})}{\sum_{n=1}^{N_{step}} s_n(x)\,cos(2\pi n/N_{step})}. \qquad (3.21)$$

Thus, the deformed fringe is:

$$s_{dn}(x) = A'(x)cos\left[\frac{2\pi n}{N_{step}} + \phi(x) +\Delta\,\phi(x)\right] + B'(x), n = 1,2,3, \ldots, N_{step} \quad . (3.22)$$

Combining Equation (3.20), (3.21) and (3.22) together, $\delta(x)$ can be determined from Equation (3.16) and (3.18). So than the surface height value is easy to calculate via Equation (3.19).

3. Calibration

The last procedure of the DFPP system is to do the 3D data calibration. From the previous section, the depth value of the object can be figured out by using the distance between the camera and reference L, the distance between camera and projector D_{pc} and the signal frequency f_0. However, all of these values are not the precise values, deformation exists in the real experiments. The response of the camera and the projector in the real experiment is non-linear. In addition, the assumption is given that the projector projects the parallel lights, while, in real experiment, the lights from the projector are not parallel. Due to these reasons, the calibration should be one of the significant steps to get the real coordinates of the 3D object surface.

The calibration is to convert the data from the height matrix to the real world coordinates. The height matrix is a 924×924 matrix from the data acquisition system, which means there are 924×924 pixels in the matrix. Each element in the matrix is the height value h for every pixel (i, j). The coordinates of this height matrix can be denoted as (U_c, V_c, h). The real world coordinates can be denoted as (x, y, h). This calibration is aimed to find out a relationship so that (U_c, V_c, h) can be mapped to (x, y, h).The conversion is related to the optical set-up parameters such as focal distance, reference plane location, fringe spacing, camera projection angle and geometric aberrations. These parameters all need to be considered. The calibration in this thesis is by using a calibration board marked with circles to establish a mathematical model from the fringe patterns projected to the reference plane. A calibration board used in the DFPP system is illustrated in Figure 3.5. The 9×11 circles are located on the calibration board. There are four larger circles which are tagged the direction of the board.

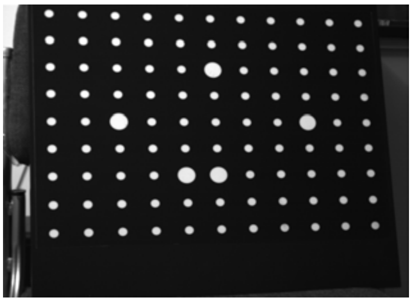


Figure 3.5: The calibration board.

3.2 Database

3.2.1 Setup of the 3D Data Acquisition System

The red/black fringe patterns are projected onto the reference and the probing faces to acquire 3D surfaces. The distance between the probing face and the camera is 1300mm. In Figure 3.6 is shown the acquisition system and the position of a probing individual. Both the position of the reference plane and the camera are fixed. The camera is on the top of the projector while the connecting line between the center of the camera and the center of the projector is vertical to the horizontal plane. The camera and the projector are both vertically pointed to the reference plane. The distance between the camera and the projector is 330mm. A frame is to fix the position between the reference plane and the camera in order to get the constants D_{pc} and L as shown in Figure 3.6. To get the best reconstruction results, the lights are always turned off in order to keep the acquisition environment in a relatively dark room. The position of the projector and the camera are shown in Figure 3.7.

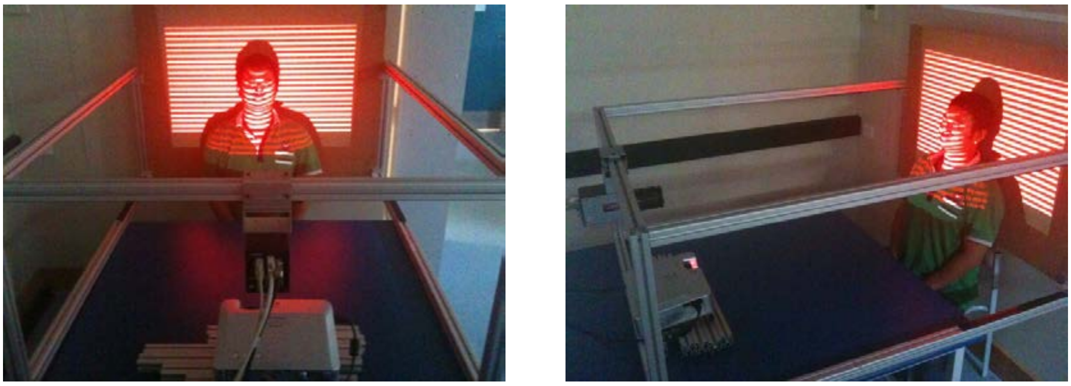


Figure 3.6: The acquisition system and position of one individual.

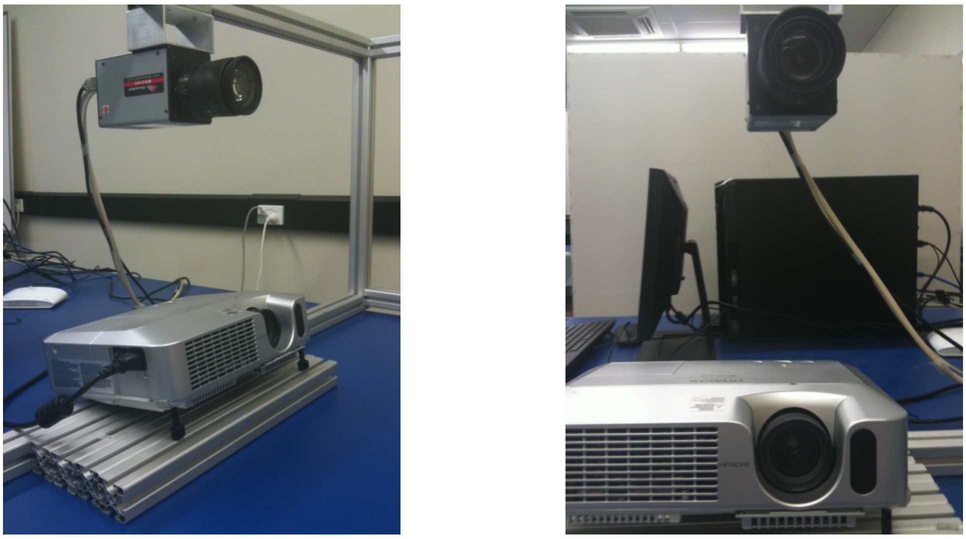


Figure 3.7: Position between the camera and the projector (camera on the top and projector on the bottom).

In this thesis, the projector HITACHI CP-X260 is selected to project fringe patterns. The specifications of the projector are listed as below:

- Type: 3 LCD.
- Resolution: 1024x768 Pixels.
- Throw Dist (m): 1.4-8.9.
- Image Size (cm): 102-76.
- Signal to noise ratio: 34.0 dB.

In this thesis, Nikon AF-S 16-35mm lens and a DuncanTech MS3100 3-CCD camera is used to capture the fringe patterns. The specifications of this camera are:

- Type: Area-scan, Colour RGB.
- Resolution: 1392x1040 Pixels.

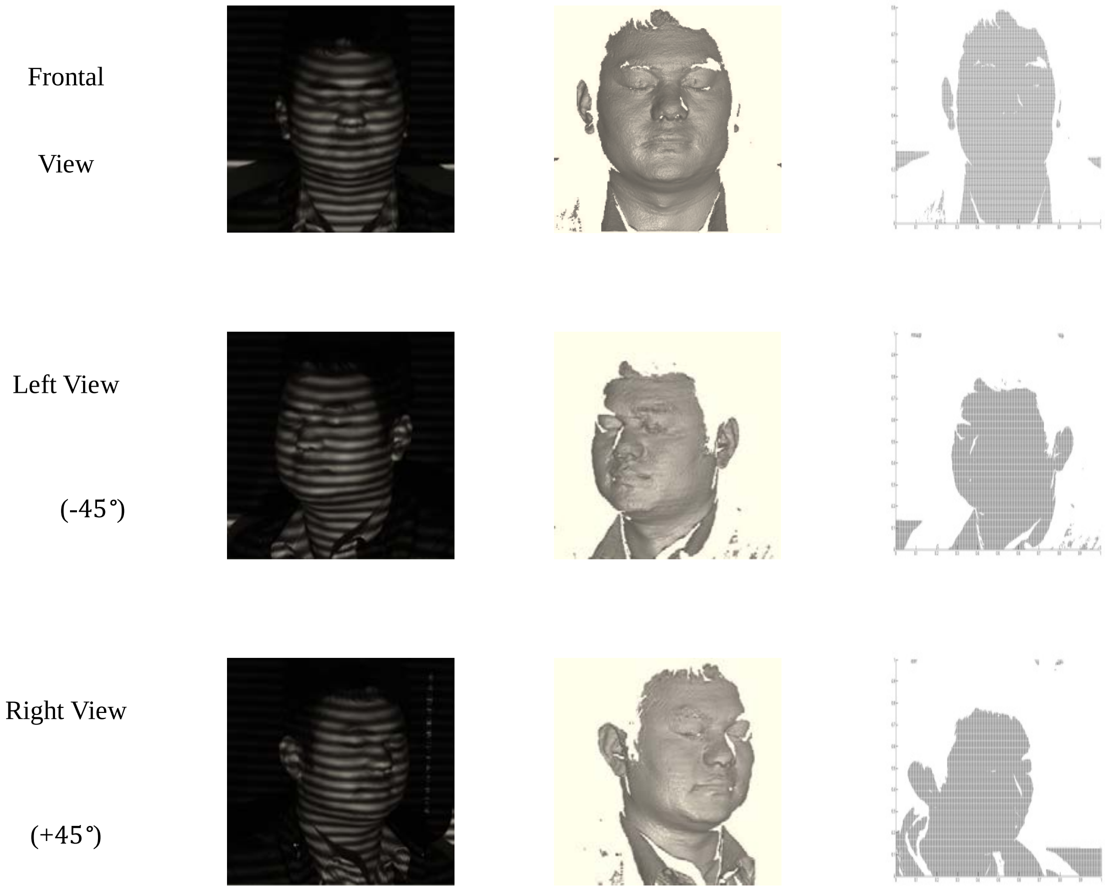
- ✧ Imager size (mm): 7.6x6.2.
- ✧ Lens focal length (mm): 16-85.
- ✧ Line rate: 8.032 kHz.
- ✧ Frame rate: Up to 7.6 fps.
- ✧ Pixel pitch: 4.65x4.65 m2.
- ✧ Shutter speed: promising sharp images hand-held at shutter speeds up to four stops slower than would usually be possible

3.2.2 Description of 3D Images in the Database

Based on the DFPP data acquisition system, a small-sized 3D face database is built. The database consists of 30 group images, with 90 images in total. In each group, there are three images with the same scale, which are from the same person with different views (frontal view, left view and right view). The right profile is about $+45°$ of rotation around the x axis and the left profile is about -45° of rotation around the x axis. The scale of the three images may be varied in different groups. No facial expressions are contained in the face images.

In Figure 3.8 are presented three face images in the database. The face images are 3D point clouds. In Figure 3.8 column (a) are shown the images captured from the camera. In Figure 3.8 column (b) and (c) are shown the reconstruction results. Column (b) is the reconstruction results shown by adding the light effect. Column (c) is the face images shown by point cloud which will be used in this thesis. There are over

100,000 points to describe one face image in the database.



(a) The images captured from the camera.

(b) The reconstruction results (shown by adding light effect).

(c) The reconstruction results (shown by points).

Figure 3.8: One example of 3D profiles of one individual.

More examples of the 3D face images are shown in Figure 3.9. All of the face images are obtained from the DFPP based 3D data acquisition system. Three views for one individual (left column: left view, middle column: frontal view, right column: right view in Figure 3.9) and one individual may have the varied scale factor with others.

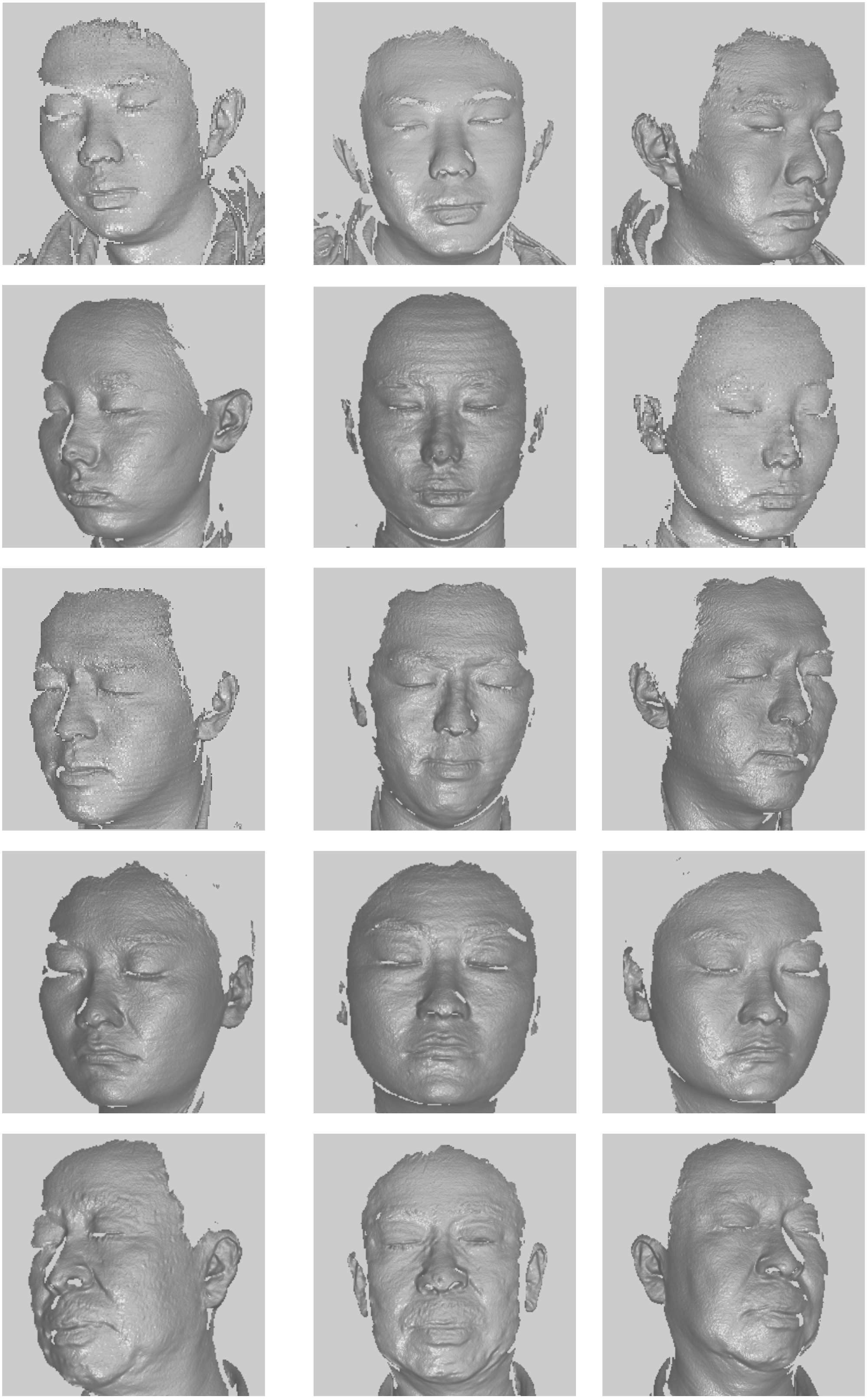


Figure 3.9: Examples of 3D face images in the database (continue).

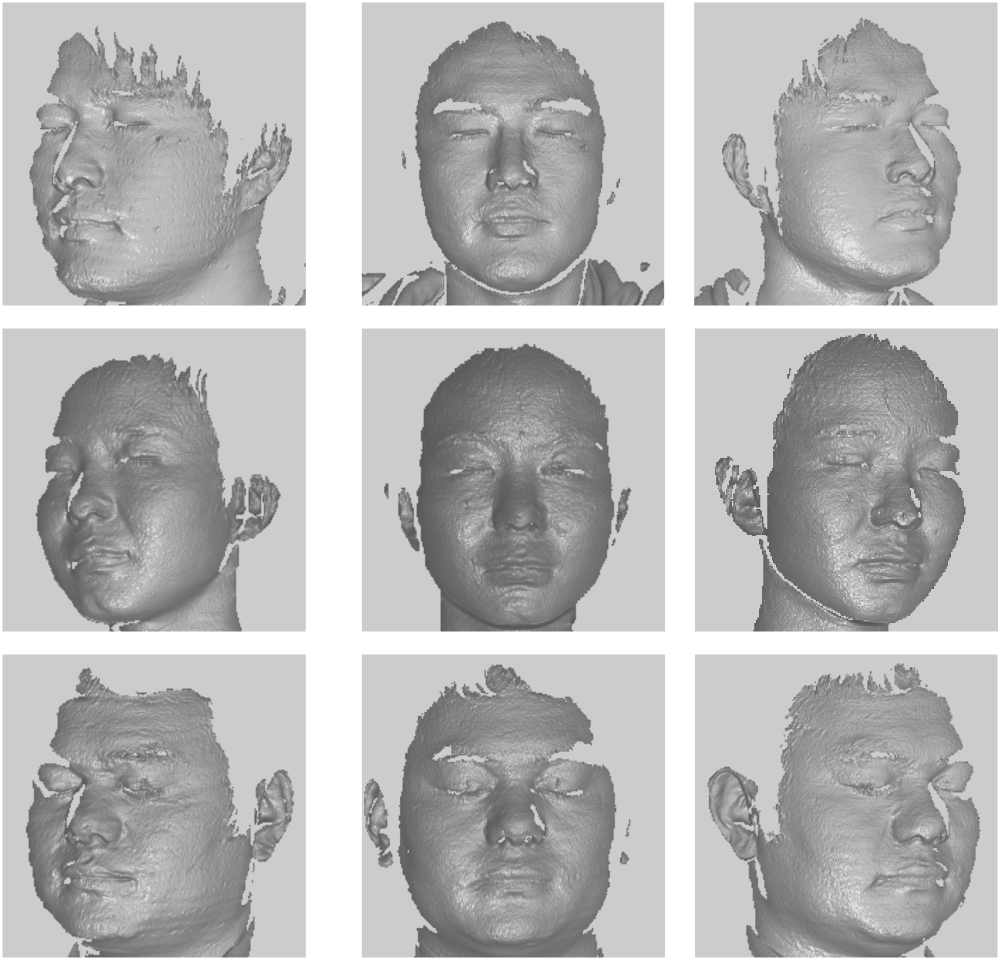


Figure 3.9: Examples of 3D face images in the database.

3.3 Summary

As we know, non-contact, accuracy and efficiency are the basic requirements for the 3D surface measurement in many industry applications. The techniques based on Structured Light Projection (SLP) can meet such requirements. DFPP based data acquisition system is a typical SLP technique which is employed to obtain 3D face data in this thesis.

The 3D face database consists of 30 groups of faces. Three different views, frontal view, left view and right view, are represented one individual. The 3D face images in this 3D face database are utilized to test the proposed 3D face recognition method in this thesis.

CHAPTER 4 FEATURE POINTS EXTRACTION AND ALIGNMENT

After the face database has been built up in Chapter 3, face recognition can be achieved by comparing the distances between the face images via the alignment process. A coarse-to-fine strategy is designed to complete the alignment which consists of two steps: coarse alignment which is described in this chapter and fine alignment which is discussed in the following chapter. The coarse alignment based on the feature points extraction is to give a rough motion value between two 3D face images (one probing face image and one reference image randomly selected from database) in order to coarsely align them.

There are two objectives of doing the coarse alignment: firstly, coarse alignment can be obtained an initial estimated value which is required in the fine alignment; secondly, the computational expenses for the fine alignment will be reduced with coarse alignment. Once the initial estimated value is applied, the distances between the two point sets will be decreased a lot, in other words, the two point sets will get much closer after the coarse alignment process. It will be less time-consuming for the fine alignment step.

4.1 Introduction

In most of the coarse alignment methods, finding the correspondences is a key point for rough motion computation. The correspondences are extracted from two point sets by feature analysis. The correspondences can be either points, lines, curves, vectors or surfaces. Extensive studies have been put on correspondences extraction, such as point signature method[24], spin image method[76], Ransac-based darces method[77]. However, such methods involve computation extensive pixel by pixel sorting operations to extract the features and thus time-consuming. In order to improve the speed, a line-based algorithm was proposed by Stamos and Leordeanu[78]. PCA method is a dimensional reduction method which can be categorized into the vector-based algorithm, the two face images can be treated as two vectors by using PCA and these two vectors are the correspondences of two point sets. Although PCA method is very fast, the results are not always accurate. Tarel, *et al*[79] used the polynomial model to align the two point sets, which is based on the surface correspondences, but the same drawback appeared with the point correspondences, the computational expense is too high to employ.

The proposed method is to estimate the motion between two point sets by using feature points. A rotation matrix $\boldsymbol{R}$, translation matrix $\boldsymbol{t}$ and scale matrix $\boldsymbol{S}$ are computed as the motion in the proposed coarse alignment method. The feature points selection is based on analysis of the shape types and curvatures. In the following section is discussed how to select the feature points and then the method of how to compute the initial estimation is given.

4.2 Feature Points Extraction

In this section is narrated the feature points extraction method. Firstly, three feature point regions are manually selected and tagged. Secondly the curvatures are calculated on the three regions in order to select the precise corresponding feature points between two face images.

4.2.1 Selecting and Tagging Feature Point Regions

In the data acquisition system, three different views of the face are acquired, the right-side view, the left-side view and the frontal view. No matter the view changes, the three regions (the right corner of the left eye, the left corner of the right eye and the nose tip) always exist in the database. So three 7×7 rectangular regions are manually selected to cover the three regions in every face image. Tagging the three different regions is to categorize the corresponding points for further selection. Four feature points are extracted from the three feature point regions in the next step.

4.2.2 Feature Points Selection

The three feature regions are manually selected in the previous section. The regions selected by hand, however, may not be in the precise corresponding regions between the two face images. A method of finding the precise corresponding feature points is presented below.

Once the feature point regions are decided manually, the four feature points will be determined. For the first feature point selection, the point is selected from the nose region which has the highest height value, which indicates the tip of nose. For other three feature points selection, the Shape Index (SI) value of every points in the three

7×7 regions are computed firstly. Then between the corresponding 7×7 region of reference image and model image, the two corresponding feature points can be selected if the two points on the reference and model have the closest SI value.

The SI value at point p is calculated by using the maximum (k_1) and minimum (k_2) curvatures. The maximum and minimum curvatures are called principal curvatures, the principal curvatures give the bent degree of the surface at point p on a differentiable surface I. The red curvatures in Figure 4.1 are the principal curvatures. At the point p, a normal plane can be gotten which contains one normal vector and therefore a unique tangent will also be in the normal plane. The intersection line of the normal plane and the surface is a curve. Different normal planes will have different curves and the curvatures will be varied.

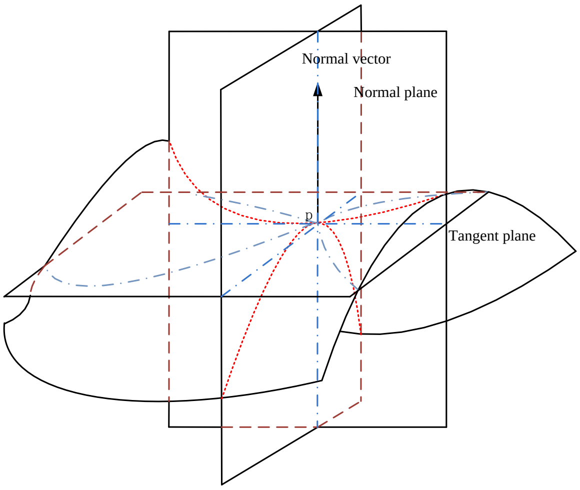


Figure 4.1: Principal curvatures.

The detail of how to calculate the Shape Index at a given point p is illustrated in Equation (4.1):

$$SI(p) = \frac{1}{2} - \frac{1}{\pi} tan^{-1} \frac{k_1(p)+k_2(p)}{k_1(p)-k_2(p)}. \tag{4.1}$$

The maximum curvature k_1 and minimum curvature k_2 in point p are given:

$$k_1(p) = H + \sqrt{H^2 - K}, \tag{4.2}$$

$$k_2(p) = H + \sqrt{H^2 + K}, \tag{4.3}$$

where H is the mean curvature, K is the Gaussian curvature. In Equation (4.4) is shown the calculation of the mean curvature at point p:

$$H(p) = H(i,j) = \frac{1}{2}(k_1(i,j) + k_2(i,j)), \tag{4.4}$$

where the number i and j are denoted as the location of point p which is in i-th row and j-th column of an image. Gaussian curvature equals to the product of the maximum curvature and minimum curvature at point p (Equation (4.5)).

$$K(i,j) = k_1(i,j)k_2(i,j). \tag{4.5}$$

The mean curvature and Gaussian curvature at point $p(i,j)$ can be calculated via both the first and second partial derivatives as follows [80]:

$$H(i,j) = \frac{\left(1 + f_y^2(i,j)\right)f_{xx}(i,j) + \left(1 + f_x^2(i,j)\right)f_{yy}(i,j) - 2f_x(i,j)f_y(i,j)f_{xy}(i,j)}{2(\sqrt{1 + f_x^2(i,j) + f_y^2(i,j)})^3}, \tag{4.6}$$

$$K(i,j) = \frac{f_{xx}(i,j)f_{yy}(i,j) - f_{xy}^2(i,j)}{(1 + f_x^2(i,j) + f_y^2(i,j))^2}, \tag{4.7}$$

where the first and the second partial derivatives at the given point $p(i,j)$ can be calculated by using the depth value of point p which is denoted as $I(i,j)$.

$$f_x = \frac{\partial I}{\partial x} = I(i+1,j) - I(i,j), \qquad (4.8)$$

$$f_y = \frac{\partial I}{\partial y} = I(i,j+1) - I(i,j), \qquad (4.9)$$

$$f_{xx} = \frac{\partial^2 I}{\partial x^2} = I(i+1,j) + I(i-1,j) - 2I(i,j), \qquad (4.10)$$

$$f_{yy} = \frac{\partial^2 I}{\partial y^2} = I(i,j+1) + I(i,j-1) - 2I(i,j), \qquad (4.11)$$

$$f_{xy} = \frac{\partial^2 I}{\partial x \partial y} = I(i+1,j+1) + I(i-1,j-1) - I(i+1,j-1) - I(i-1,j+1). \qquad (4.$$

Now from the SI equation, SI value can be computed within the interval $[0,1]$ at any arbitrary point. An example of the Shape Index values for nine shape types is shown in Figure 4.2. After computing the curvatures by using the first and second partial derivatives (Equation (4.8) to Equation (4.12)), the valid points are in the region of 5×5 instead of 7×7.

1.000 **0.000**

1.000	**0.875**	**0.750**	**0.625**	**0.500**	**0.375**	**0.250**	**0.125**	**0.000**
Spherical cap	**Dome**	**Ridge**	**Saddle ridge**	**saddle**	**Saddle rut**	**Rut**	**Trough**	**Spherical cup**

Figure 4.2: The Shape Index values of nine shape types.

The corresponding feature points can be extracted by comparing the SI value of every point from the corresponding regions. As we know, the SI values of the corresponding points should be equal. Here the corresponding points can be determined when the two points have the closest SI value after all of the points in the corresponding regions have been traversed. If the corresponding point pair is also the highest height value in the nose region which has been treated as the first feature point before, the corresponding point pair will be selected from the second closest SI value.

4.3 Motion Computation and Alignment

Once the feature points are selected, the motion between the four corresponding point pairs can be calculated to align the two face images in this section. The motion consists of the scale, the rotation and the translation.

4.3.1 Scale Matrix Computation

The scale matrix of two 3D point sets can be computed by using the coordinates of the points. The scale factor is the ratio of the two point sets coordinates. The constraint of using the coordinates of the points to compute the scale factor is the points which must be one-to-one correspondence.

For the feature points on the model face image $\boldsymbol{x}$ and feature points on the reference face image $\boldsymbol{y}$, the covariance matrices $\boldsymbol{C_x}$ and $\boldsymbol{C_y}$ of $\boldsymbol{x}$ and $\boldsymbol{y}$ can be calculated. Thus, the scale factor $\boldsymbol{S_{ini}}$ can be obtained between the two data sets:

$$S_{ini\_j} = \frac{1}{3}\sum_{i=0}^{3}\frac{\mu_j}{\lambda_j}\ (j = 1{,}2{,}3.), \tag{4.13}$$

where λ_j is the square root of the j-th eigenvalues of $\boldsymbol{C_x}$, μ_j is the square root of the j-th eigenvalues of $\boldsymbol{C_y}$.

The initial scale matrix can be written as:

$$\boldsymbol{S_{ini}} = \begin{bmatrix} S_{ini\_1} & 0 & 0 \\ 0 & S_{ini\_2} & 0 \\ 0 & 0 & S_{ini\_3} \end{bmatrix}. \tag{4.14}$$

In the experiment, it is better to set a threshold for further fine alignment so as to keep the scale factor more accurate. Hence a threshold δ is set, and the scale factor $\boldsymbol{S}$ will be in the interval:

$$\boldsymbol{S} \in [\boldsymbol{S_{ini}} - \delta, \boldsymbol{S_{ini}} + \delta], \tag{4.15}$$

where $\boldsymbol{S}$ is the scale factor computed from the fine alignment which will be discussed in the next chapter. If the threshold is set as a very small value, it means that the computed scale factor in fine alignment will be close to the initial scale factor $\boldsymbol{S_{ini}}$. Based on the experiment, the threshold δ equals to 0.12. To set a boundary will avoid the situation that the two face images are convergence to a small subset in the fine alignment. The new model face image will be applied by the scale matrix which can be denoted as:

$$\boldsymbol{x_{fp}} = \boldsymbol{S_{ini}}\boldsymbol{x}, \tag{4.16}$$

where $\boldsymbol{x}$ is the original model face image and $\boldsymbol{x_{fp}}$ is the model face image after the application of the scale effect, $\boldsymbol{S}$ is the scale matrix.

4.3.2 Rotation and Translation Computation

Dual quaternions method [42] is selected to compute the rotation and translation. The two feature point sets can be denoted as: $\boldsymbol{x_{fp}} = \{\boldsymbol{x_{fp_i}}\}$, i=1,2,3,4. Reference point set $\boldsymbol{y_{fp}} = \{\boldsymbol{y_{fp_i}}\}$, i=1,2,3,4. The objective function is:

$$\mathcal{F}(\boldsymbol{R}, \boldsymbol{t}) = d^2(\boldsymbol{x_{fp}}, \boldsymbol{y_{fp}}) = \frac{1}{4}\sum_{i=1}^{4}\left\|\boldsymbol{R}\boldsymbol{x_{fp_i}} + \boldsymbol{t} - \boldsymbol{y_{fp_i}}\right\|^2. \tag{4.17}$$

Regarding the objective function, we can see that it is an optimization problem about how to find out the best rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ so as to achieve the least squares. In 1843, Quaternions were firstly introduced by Hamilton. In 1873, Hamilton gave the preliminary sketch of biquaternions. In the late 20th Century, quaternions began to be used widely in computer graphics, computer vision, robotics, attitude control, control theory, signal processing and computer control, etc. Quaternions can be used to estimate the positions of objects in 3D space. Wahba Grace in 1965 [81] stated that the satellite attitude could be estimated by solving a least square problem. In 1977, Keat [82] found out a solution about how to compute the three-axis attitude of a spacecraft at a single time point by using quaternions. Till now, quaternions have been used extensively for parameterizing orientation. [42]

A brief conclusion of the steps to compute the rotation and translation is given in this section, the basic definitions and equations of dual quaternions are demonstrated in Appendix.

Quaternions are four-element vectors, the first three elements can be treated as a 3×1

vector, and the fourth element is a scalar component. A quaternion $\breve{\boldsymbol{q}}$ can be described like this:

$$\breve{\boldsymbol{q}} = \begin{bmatrix} \mathrm{q}_1 \\ \mathrm{q}_2 \\ \mathrm{q}_3 \\ \mathrm{q}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q} \\ \mathrm{q}_4 \end{bmatrix}, \tag{4.18}$$

where $\boldsymbol{q} = [q_1, q_2, q_3]^T$is a 3D vector which is equal to the values of the original coordinates, where T indicates the transpose of matrix, a scalar q_4 which is set to zero in this thesis based on the experiments.

Step 1: Compute matrices $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$.

$$\boldsymbol{C_1} = -2\sum_{i=1}^{N} \boldsymbol{Q}\left(\boldsymbol{y_{i_0}}\right)^T \boldsymbol{W}(\boldsymbol{x_i}), \tag{4.19}$$

$$\boldsymbol{C_2} = -2\sum_{i=1}^{N} \left[\boldsymbol{W}(\boldsymbol{x_i}) - \boldsymbol{Q}\left(\boldsymbol{y_{i_0}}\right)\right], \tag{4.20}$$

where

$$\mathbf{W}(\breve{\boldsymbol{q}}) = \begin{bmatrix} q_4\boldsymbol{I} - \boldsymbol{k}(\boldsymbol{q}) & \boldsymbol{q} \\ -\boldsymbol{q^T} & q_4 \end{bmatrix}, \tag{4.21}$$

$$\mathbf{Q}(\breve{\boldsymbol{q}}) = \begin{bmatrix} q_4\boldsymbol{I} + \boldsymbol{k}(\boldsymbol{q}) & \boldsymbol{q} \\ -\boldsymbol{q^T} & q_4 \end{bmatrix}, \tag{4.22}$$

$$\boldsymbol{K}(\boldsymbol{q}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}. \tag{4.23}$$

Step 2: Compute matrix A.

$$\mathrm{A} = \frac{1}{2}\left[\frac{1}{2l_m}\boldsymbol{C_2}^T\boldsymbol{C_2} - \boldsymbol{C_1} - \boldsymbol{C_1}^T\right], \tag{4.24}$$

where l_m is the number of corresponding point pairs. For coarse motion computation, $l_m = 4$ which means there are four corresponding feature point pairs.

Step 3: Compute the eigenvector $\breve{\boldsymbol{q}}$ corresponding to the largest positive eigenvalue of matrix A, compute $\breve{\boldsymbol{s}}$ from $\breve{\boldsymbol{q}}$.

$$\breve{\boldsymbol{s}} = -\frac{1}{2l_m}\boldsymbol{C_2}\breve{\boldsymbol{q}}. \tag{4.25}$$

Step 4: Compute rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$.

$$\boldsymbol{R} = (q_4^2 - \boldsymbol{q^T q})\boldsymbol{I} + 2\boldsymbol{q}\boldsymbol{q^T} + 2q_4\boldsymbol{K}(\boldsymbol{q}), \tag{4.26}$$

$$\breve{\boldsymbol{p}} = W(\breve{\boldsymbol{q}})^T\breve{\boldsymbol{s}}, \tag{4.27}$$

where the translation $\mathbf{t}$ is the vector part of quaternion $\breve{\boldsymbol{p}}$.

4.3.3 Face Images Alignment

The two face images can be aligned by using the motion computation method described in the previous sections. Once the scale matrix, rotation matrix and translation matrix have been computed, the motion which consists of the three matrices can be applied to the face images. An equation is given to align the two face images:

$$\boldsymbol{x_{ini}} = \boldsymbol{R_{ini}}\boldsymbol{S_{ini}}\boldsymbol{x} + \boldsymbol{t_{ini}}, \tag{4.28}$$

where $\boldsymbol{R_{ini}}$, $\boldsymbol{S_{ini}}$ and $\boldsymbol{t_{ini}}$ are the initial estimation of rotation, scale, and transformation matrices, $\boldsymbol{x}$ is the original probing face image, $\boldsymbol{x_{ini}}$ is the probing face image after applied the motion. The new probing face image $\boldsymbol{x_{ini}}$ can be aligned with the reference image by using Equation (4.28).

4.4 Procedure of Feature Point Extraction and Alignment

The procedure of feature points extraction and alignment can be concluded in this section (Figure 4.3). The *INPUT* is two face images, one is the probing face image and one is the reference which is selected randomly from the database. The *OUTPUT* is three matrices: scale matrix $\boldsymbol{S_{ini}}$, rotation matrix $\boldsymbol{R_{ini}}$, translation matrix $\boldsymbol{t_{ini}}$; a new probing face image applied the motion.

Step 1: Selecting and tagging three corresponding feature regions in the human face. The three regions are selected from the left corner of the right eye, the right corner of the left eye, and the nose tip.

Step 2: Extracting four feature points from the corresponding three feature regions by using the SI value.

Step 3: Compute and apply scale matrix $\boldsymbol{S_{ini}}$, rotation matrix $\boldsymbol{R_{ini}}$ and translation matrix $\boldsymbol{t_{ini}}$ between the two face images on the basis of the four feature points.

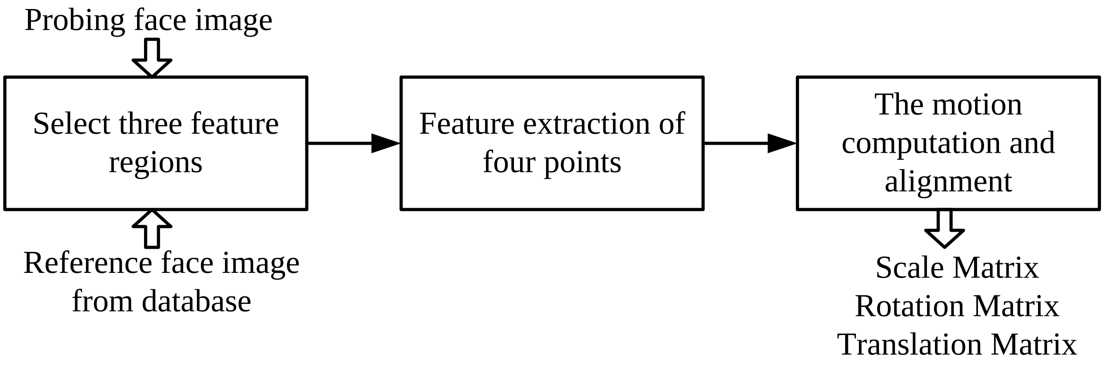

Probing face image
Select three feature regions
Feature extraction of four points
The motion computation and alignment
Reference face image from database
Scale Matrix
Rotation Matrix
Translation Matrix

Figure 4.3: Procedure of feature points selection.

4.5 Implementation and Results

In this section, two sets of experiments are given for feature points extraction and motion computation according to the methods demonstrated in the previous sections. The ***INPUT*** of the feature points extraction is two images, one is the model image and the other is the reference image selected randomly from the database. The SI values of the points in the feature point regions are needed to calculate by using the Equation (4.8) to (4.12). The ***OUTPUT*** of the feature points extraction is four feature points.

A set of experiments for motion computation and alignment are based on the analysis of four feature points. The ***INPUT*** of the motion computation and alignment is the four feature points, and the ***OUTPUT*** is: three matrices: scale matrix $\boldsymbol{S_{ini}}$, rotation matrix $\boldsymbol{R_{ini}}$, translation matrix $\boldsymbol{t_{ini}}$; a new probing face image.

4.5.1 Feature Point Extraction

In this section, the experiments for four feature points selection are given. Two experiments are designed to test the accuracy of the proposed method of feature points extraction. Experiment I is given the feature points extraction results when the input of the two face images are the genuine person, experiment II is given the feature points extracted from different persons.

1. Experiment I

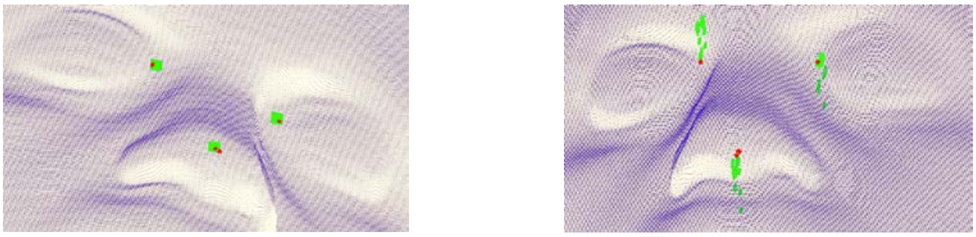
The input of this experiment is two face images from the identical person with different views. The SI values of the corresponding regions (the right corner of the left eye) of two images are listed in Table 4.1.

Table 4.1: Shape Index values of two corresponding regions (both on the right corner of the left eye, one reference image and one model image).

Model Image					Reference Image				
0.915	0.915	0.814	0.678	0.605	0.753	0.702	0.589	0.427	0.285
0.500	0.522	0.565	0.466	0.516	0.601	0.776	0.611	0.567	0.655
0.648	0.719	0.688	0.526	0.475	0.473	0.305	0.474	0.557	0.567
0.724	0.814	0.849	0.713	0.457	0.409	0.257	0.036	0.590	0.540
0.851	0.835	0.698	0.468	0.295	0.350	0.103	0.338	***0.500***	0.637

The left-side of Table 4.1 is the SI values in the right corner of the left eye region from the probing face image (model image), and the right-side is the SI values in the corresponding region from the reference image. The two points can be selected as the correspondences which belong to the corresponding regions if they have the closest SI value. In Table 4.1, the point of which SI value is 0.500 in model image has the closest SI value 0.500 in reference, so that these two points are treated as the corresponding point pair.

After analysis of all three correspondences between two point sets, the feature points can be obtained as is shown in Figure 4.4. Figure 4.4(a) and (b) are the two figures come from the model image and the reference image which are from the same person with different views. The green areas are the feature point regions, the four points in red are the feature points after the analysis of the curvatures. The proposed method can extract the corresponding feature points from the results shown in Figure 4.4.



(a) The feature points (reference). (b) The feature points (model).

Figure 4.4: Four feature points selected based on proposed method (genuine face).

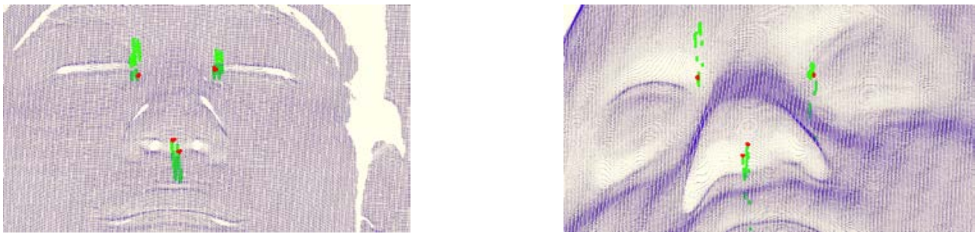
2. Experiment II

In this experiment, the feature points are extracted from different persons. The input of this experiment is two face images: one is the probing face image whereas the other is the reference face image from the database. The SI values of the region on the right corner of the left eye are listed in Table 4.2.

The feature points extraction results are shown in Figure 4.5. The two figures (a) and (b) are the images come from different people, (a) is the reference image selected from the database and (b) is the model face image. The red points are the extracted feature points which indicate that the feature points can be selected even if the two face images from the imposter face.

Table 4.2: Another group of Shape Index values of two corresponding regions (both on the right corner of the left eye, one reference image and one model image).

Model Image					Reference Image				
0.388	0.382	0.356	0.387	0.328	0.753	0.702	0.589	0.427	0.285
0.325	0.318	0.297	0.344	0.345	0.601	0.776	0.611	0.567	0.655
0.314	0.323	0.281	0.324	0.345	0.473	***0.305***	0.474	0.557	0.567
0.318	0.352	0.292	***0.316***	0.346	0.409	0.257	0.036	0.590	0.540
0.309	0.414	0.360	0.338	0.358	0.350	0.103	0.338	0.500	0.637



(a) The feature points (reference).

(b) The feature points (model).

Figure 4.5: Four feature points selected based on proposed method (different people).

4.5.2 Motion Computation and Alignment

This step is to calculate the motion of the two point sets based on the extracted feature points in the previous experiments and align the two point sets. Three experiments are designed to test the proposed method. Experiment I is to test the functionality and the accuracy of motion computation method by using the artificial data. Experiment II and III are using the real face images to achieve coarse alignment. The ***INPUT*** of this step is two point sets (reference and model) with four corresponding feature points; the ***OUTPUT*** is a rotation matrix $\boldsymbol{R}_{\boldsymbol{ini}}$, a translation matrix $\boldsymbol{t}_{\boldsymbol{ini}}$, a scale matrix $\boldsymbol{S}_{\boldsymbol{ini}}$ and a new aligned probing face image.

1. Experiment I

One group of simulated data is used to test the functionality and accuracy of the motion computation method. Firstly a face image is selected from the database, and then the face image is manually moved with a rotation matrix $\boldsymbol{R}_{sim}$ and a translation matrix $\boldsymbol{t}_{sim}$ and decreased the size with a scaling matrix $\boldsymbol{S}_{sim}$. The three simulated matrices are listed as follows:

$$\boldsymbol{R}_{sim\_0} = \begin{bmatrix} 0.9924 & -0.0868 & -0.0872 \\ 0.0793 & 0.9931 & -0.0868 \\ 0.0941 & 0.0793 & 0.9924 \end{bmatrix}, \tag{4.29}$$

$$\boldsymbol{t}_{sim\_0} = \begin{bmatrix} 0.1500 \\ 0.0500 \\ 0.0200 \end{bmatrix}, \tag{4.30}$$

$$\boldsymbol{S}_{sim\_0} = \begin{bmatrix} 0.8000 & 0.0000 & 0.0000 \\ 0.0000 & 0.8000 & 0.0000 \\ 0.0000 & 0.0000 & 0.8000 \end{bmatrix}. \tag{4.31}$$

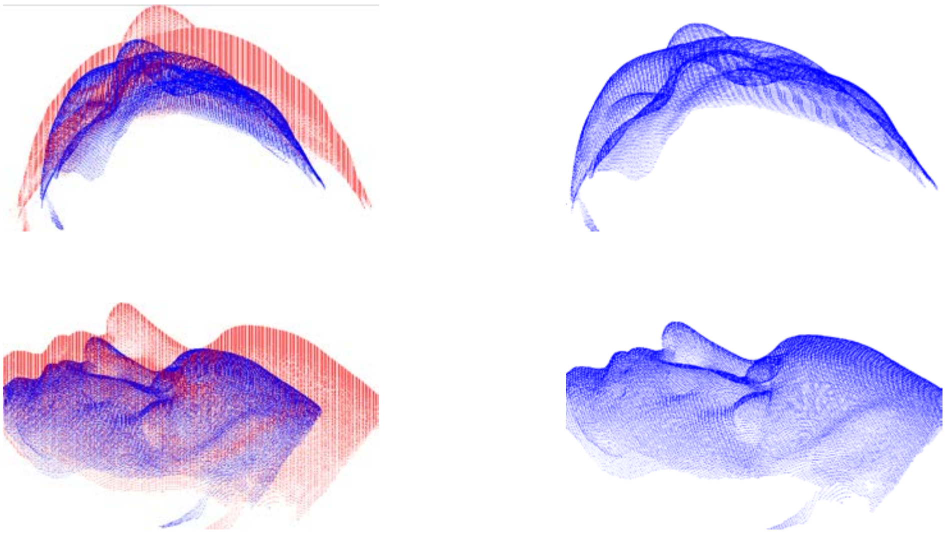
Because the second data set is obtained from the first one, the two point sets are totally overlapped and all of the points in point set one have the corresponding points in point set two. All of the points can be used via the proposed method to compute motion, the three matrices $\boldsymbol{R}_{sim\_ini}$, $\boldsymbol{t}_{sim\_ini}$ and $\boldsymbol{S}_{sim_ini}$are generated as follows:

$$\boldsymbol{R}_{sim\_ini} = \begin{bmatrix} 0.9924 & -0.0868 & -0.0872 \\ 0.0793 & 0.9931 & -0.0868 \\ 0.0941 & 0.0793 & 0.9924 \end{bmatrix}, \tag{4.32}$$

$$\boldsymbol{t}_{sim\_ini} = \begin{bmatrix} 0.0555 \\ 0.0396 \\ 0.1278 \end{bmatrix}, \tag{4.33}$$

$$\boldsymbol{S}_{sim\_ini} = \begin{bmatrix} 0.8000 & 0.0000 & 0.0000 \\ 0.0000 & 0.8000 & 0.0000 \\ 0.0000 & 0.0000 & 0.8000 \end{bmatrix}. \tag{4.34}$$

Figure 4.6(a) is given the original status of the reference and the simulated data, the results after applied the motion computed based on the proposed method are given in Figure 4.6(b). The red point set in Figure 4.6(a) is the reference point set, the blue point set in Figure 4.6(a) is the artificial data. The blue point set in Figure 4.6(b) is the experiment result after applying the motion computed from the proposed method. The two point sets are totally overlapped so that only one color is shown in Figure 4.6(b).



(a) The original status of reference and the artificial data.

(b) The result after estimate the motion between the two point sets.

Figure 4.6: Coarse alignment results of the artificial data.

To test the accuracy of the proposed method, the errors contained in the results are computed. The differences between the simulated values and the experiment results are used to show the error $\boldsymbol{err}$. It can be denoted as:

$$\boldsymbol{err_S} = \left|\boldsymbol{S_{sim\_ini}} - \boldsymbol{S_{sim\_0}}\right|, \tag{4.35}$$

$$\boldsymbol{err_R} = \left|\boldsymbol{R_{sim\_ini}} - \boldsymbol{R_{sim\_0}}\right|, \tag{4.36}$$

$$\boldsymbol{err_t} = \left|\boldsymbol{t_{sim\_ini}} - \boldsymbol{t_{sim\_0}}\right|, \tag{4.37}$$

where $\boldsymbol{err_S}$, $\boldsymbol{err_R}$ and $\boldsymbol{err_t}$ are the errors of the scale, rotation and translation respectively. Comparing the real results computed with the pre-set initial data, it is obtained that $\boldsymbol{err_S} = 0.0000$, $\boldsymbol{err_R} = 0.0000$ for every element in the matrices while the error of the translation is higher than others which is equal to

$$\boldsymbol{err}_t = \begin{vmatrix} 0.0555 - 0.1500 \\ 0.0396 - 0.0500 \\ 0.1278 - 0.0200 \end{vmatrix} = \begin{bmatrix} 0.0945 \\ 0.0104 \\ 0.1078 \end{bmatrix}. \tag{4.38}$$

From the results shown in Figure 4.6 and the given error, the coarse alignment can be achieved very well for artificial data via the proposed method. The output rotation matrix $\boldsymbol{R_{sim\_ini}}$, a translation matrix $\boldsymbol{t_{sim\_ini}}$ and a scale matrix $\boldsymbol{S_{sim\_ini}}$ will be the initial values for the fine alignment.

2. Experiment II

In this section, the real data which is required from the same person but with different views is used to test the proposed method. The two point sets are from the same person shown in Figure 4.7. The initial status of the two face images is given in Column (a), and the status of the two face images after coarse alignment is shown in Column (b). The coordinates of the four feature points are extracted by using the feature points extraction method which are:

$$fp_{ref} = \begin{bmatrix} 0.4143 & 0.3514 & 0.9767 \\ 0.3753 & 0.4620 & 0.4718 \\ 0.4685 & 0.4599 & 0.5044 \\ 0.3688 & 0.3037 & 0.9978 \end{bmatrix}, \tag{4.39}$$

$$fp_{mod} = \begin{bmatrix} 0.4189 & 0.3458 & 0.9497 \\ 0.4502 & 0.4323 & 0.4319 \\ 0.5323 & 0.4356 & 0.4532 \\ 0.4959 & 0.2575 & 0.8684 \end{bmatrix}, \tag{4.40}$$

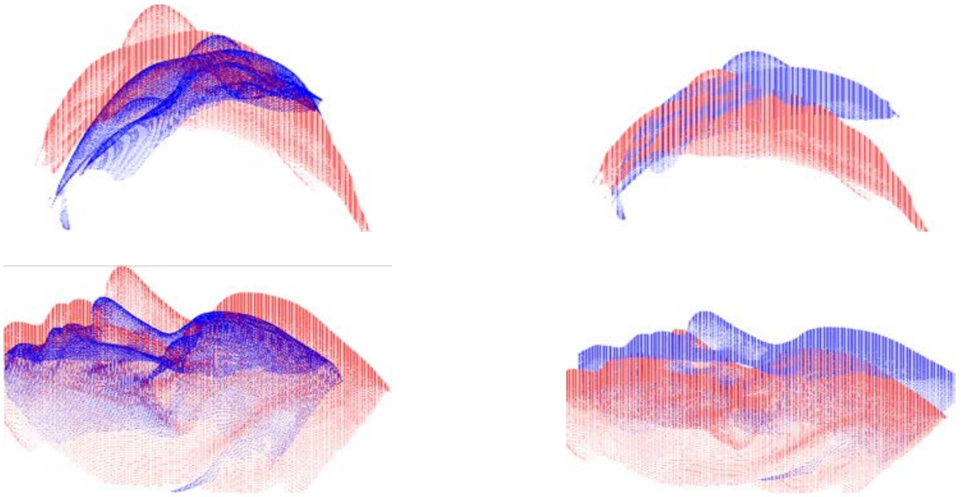
where fp_{ref} consists of the feature points coordinates from the reference and fp_{mod} consists of the feature points coordinates from model. Each row in the Equation (4.39) and (4.40) indicates the coordinates of one feature point. The four feature points in Equation (4.39) are corresponding to the four feature points in Equation (4.40).

Based on these coordinates of the feature points, the motion can be calculated via the method given in Section 4.3 which can be listed:

$$\boldsymbol{R}_{ini} = \begin{bmatrix} 0.9924 & 0.0944 & -0.0789 \\ -0.0868 & 0.9917 & 0.0944 \\ -0.0872 & 0.0868 & 0.9924 \end{bmatrix}, \tag{4.41}$$

$$\boldsymbol{t}_{ini} = \begin{bmatrix} -0.0738 \\ 0.0221 \\ 0.0787 \end{bmatrix}, \tag{4.42}$$

$$\boldsymbol{S}_{ini} = \begin{bmatrix} 0.8722 & 0 & 0 \\ 0 & 0.8722 & 0 \\ 0 & 0 & 0.8722 \end{bmatrix}. \qquad (4.43)$$



(a) The original status of reference and the artificial data.

(b) The result after estimate the motion between the two point sets.

Figure 4.7: Coarse alignment results of the real data I (the two point sets are from the same person).

The scale of the probing face image is getting closer to the reference image after the coarse alignment via the Equation (4.28), the two point sets are aligned via applying the motion computed from the feature points in the status of Column (b) compared with the status of Column (a).

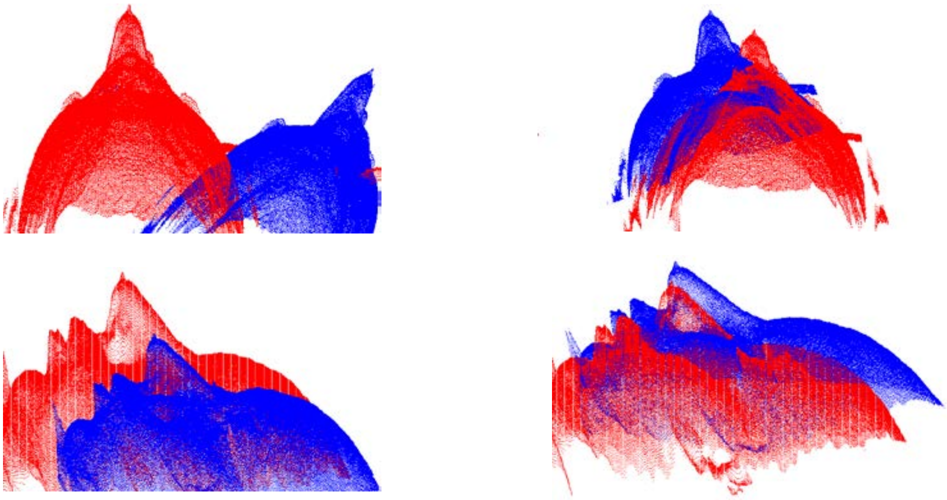
From the results of the scale matrix $\boldsymbol{S}_{ini}$, the threshold for the scale factor will be in the interval $[S_{ini\_j} - \delta, S_{ini\_j} + \delta]$(given in Section 4.3), where $S_{ini\_j} = 0.8722$, here we have $S_{ini\_1} = S_{ini\_2} = S_{ini\_3} = 0.8722$, $\delta = 0.1047$. The boundary of the scale factor is $[0.7675,0.9769]$.

The alignment result of another group of two point sets from the genuine person is shown in Figure 4.8. The initial status of the two face images is given in Column (a), and the status of the two face images after coarse alignment is shown in Column (b). The coordinates of the four feature points are extracted by using the feature points extraction method which are:

$$fp_{ref} = \begin{bmatrix} 0.2573 & 0.2886 & 0.7641 \\ 0.2509 & 0.2992 & 0.7333 \\ 0.1189 & 0.3951 & 0.7213 \\ 0.4002 & 0.3530 & 0.8999 \end{bmatrix}, \tag{4.44}$$

$$fp_{mod} = \begin{bmatrix} 0.3711 & 0.2932 & 0.8777 \\ 0.3614 & 0.3019 & 0.8508 \\ 0.4510 & 0.3964 & 0.7820 \\ 0.9374 & 0.4009 & 0.6876 \end{bmatrix}. \tag{4.45}$$

After computing the motion computation based on the feature points extraction, the two face images are tuned and getting closer in the status of Column (b) compared with the status of Column (a). The two point sets are aligned by using Equation (4.28).



(a) The original status of reference and model images.

(b) The result after estimate the motion between the two point sets.

Figure 4.8: Coarse alignment results of the real data II (the two point sets are from the same person).

The output is three matrices:

$$\boldsymbol{R}_{ini} = \begin{bmatrix} 0.8627 & 0.0755 & -0.500 \\ -0.0872 & 0.9962 & 0.0000 \\ 0.4981 & 0.0436 & 0.8660 \end{bmatrix}, \tag{4.46}$$

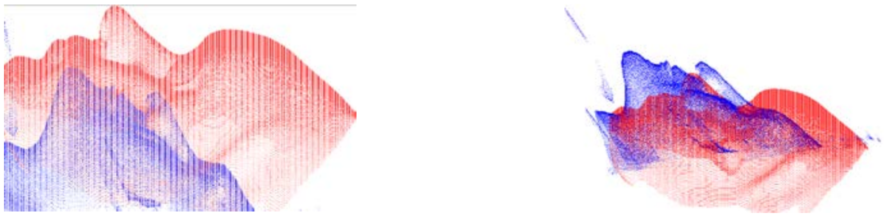
$$\boldsymbol{t}_{ini} = \begin{bmatrix} -0.1528 \\ -0.0013 \\ -0.3027 \end{bmatrix}, \tag{4.47}$$

$$\boldsymbol{S}_{ini} = \begin{bmatrix} 0.9032 & 0 & 0 \\ 0 & 0.9032 & 0 \\ 0 & 0 & 0.9032 \end{bmatrix}. \tag{4.48}$$

From the results of the scale matrix $\boldsymbol{S}_{ini}$, the threshold for the scale factor will be in the interval $\left[S_{ini\_j} - \delta, S_{ini\_j} + \delta\right]$, where $S_{ini\_1} = S_{ini\_2} = S_{ini\_3} = 0.9032$, $\delta = 0.1083$. The boundary of the scale factor is $[0.7948, 1.0115]$.

3. Experiment III

The two face images come from two imposter faces are compared in this section. The two point sets are from the different faces shown in Figure 4.9. The initial status of the two face images is given in Column (a), and the status of the two face images after coarse alignment is shown in Column (b). The two face images are getting closer in the status of Column (b) compared with the status of Column (a).



(a) The original status of reference and the artificial data.

(b) The result after estimate the motion between the two point sets.

Figure 4.9: Coarse alignment results of the real data III (the two point sets are from the different people).

The coordinates of the corresponding feature points are extracted via the proposed method:

$$fp_{ref} = \begin{bmatrix} 0.3037 & 0.4772 & 0.6127 \\ 0.4469 & 0.4816 & 0.6341 \\ 0.3579 & 0.3080 & 0.9967 \\ 0.3557 & 0.3102 & 0.9955 \end{bmatrix}, \tag{4.49}$$

$$fp_{mod} = \begin{bmatrix} 0.3869 & 0.3688 & 0.9202 \\ 0.4169 & 0.4946 & 0.2318 \\ 0.3774 & 0.4252 & 0.5713 \\ 0.3774 & 0.4282 & 0.5446 \end{bmatrix}. \tag{4.50}$$

The output is three matrices computed from the coordinates of the feature points via the proposed method:

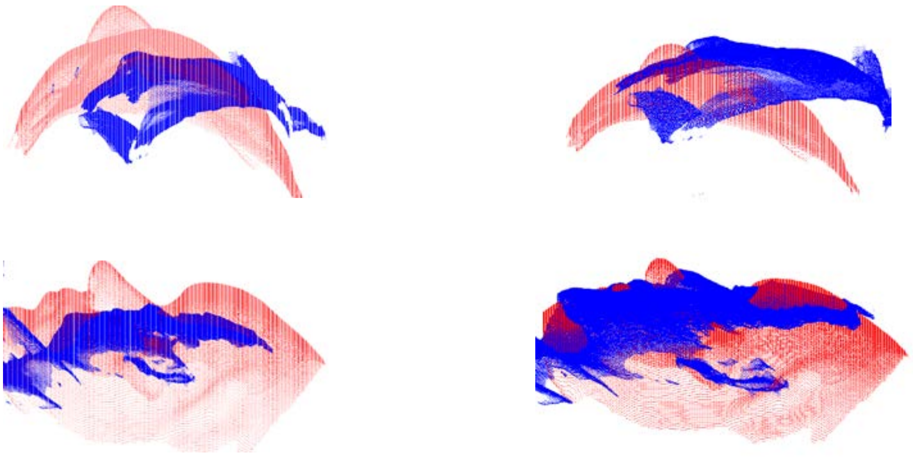
$$\boldsymbol{R}_{ini} = \begin{bmatrix} 0.9839 & -0.1418 & -0.1086 \\ 0.1157 & 0.9693 & -0.2168 \\ 0.1360 & 0.2007 & 0.9702 \end{bmatrix}, \tag{4.51}$$

$$\boldsymbol{t}_{ini} = \begin{bmatrix} 0.1072 \\ -0.0102 \\ 0.2797 \end{bmatrix}, \tag{4.52}$$

$$\boldsymbol{S}_{ini} = \begin{bmatrix} 0.8989 & 0 & 0 \\ 0 & 0.8989 & 0 \\ 0 & 0 & 0.8989 \end{bmatrix}. \tag{4.53}$$

All of the three matrices for the coarse alignment results are the input for the fine alignment. The scale factor will be in the interval $[0.7910, 1.0068]$. Although the two face images are from different people, the two face images can be still aligned for further refine alignment and distance comparison.

The two face images come from another groups of the imposter faces are compared as follows. The two point sets are from the different faces shown in Figure 4.10. The initial status of the two face images is given in Column (a), and the status of the two face images after coarse alignment is shown in Column (b). The distance between the two face images are closer in the status of Column (b) compared with the status of Column (a).



(a) The original status of reference and the artificial data.

(b) The result after estimate the motion between the two point sets.

Figure 4.10: Coarse alignment results of the real data IV (the two point sets are from the different people).

The coordinates of the corresponding feature points are extracted:

$$fp_{ref} = \begin{bmatrix} 0.3557 & 0.2972 & 0.9951 \\ 0.3254 & 0.2668 & 0.7384 \\ 0.4230 & 0.2777 & 0.8319 \\ 0.3080 & 0.4816 & 0.6422 \end{bmatrix}, \tag{4.54}$$

$$fp_{mod} = \begin{bmatrix} 0.3738 & 0.1358 & 0.9134 \\ 0.2642 & 0.1148 & 0.9448 \\ 0.4063 & 0.1382 & 0.9765 \\ 0.1681 & 0.4146 & 0.7994 \end{bmatrix}. \tag{4.55}$$

The output is three matrices calculated from the coordinates of the feature points via the proposed method:

$$\boldsymbol{R}_{ini} = \begin{bmatrix} 0.9623 & -0.0842 & 0.2588 \\ 0.1093 & 0.9904 & -0.0842 \\ -0.2493 & 0.1093 & 0.9632 \end{bmatrix}, \tag{4.56}$$

$$\boldsymbol{t}_{ini} = \begin{bmatrix} -0.4845 \\ -0.2036 \\ 0.1508 \end{bmatrix}, \tag{4.57}$$

$$\boldsymbol{S}_{ini} = \begin{bmatrix} 0.7428 & 0 & 0 \\ 0 & 0.7428 & 0 \\ 0 & 0 & 0.7428 \end{bmatrix}. \tag{4.58}$$

All of the three matrices for the coarse alignment results and the two point sets are the input for the fine alignment. The scale factor will be in the interval $[0.6537, 0.8319]$.

4.6 Summary

The feature point extraction and alignment method is introduced in this chapter. A curvature based feature points extraction method consisting of three steps is proposed to achieve coarse alignment. The three steps can be listed as following: three feature region selection, four feature point extraction, the motion computation and alignment.

As the experiment results show, no matter whether the two face images are from the genuine face or not, the corresponding feature points can be successfully extracted via the curvature based feature points extraction method. An efficient method to calculate the scale, rotation and translation matrices is given and evaluated by using different face images.

Although the face images can be aligned after this chapter, it is still need to achieve fine alignment. A modified Iterative Closest Point method will be discussed for fine alignment in order to precisely tune the location of the face images.

CHAPTER 5 3D FACE RECOGNITION

The coarse alignment has aligned the probing face image and the reference face image closer, then the fine alignment is required for further refine the alignment and calculate the different distance between the two face images. The two face images with a different distance smaller that a threshold will be recognized as from the same person. In this chapter, a fine alignment method on the basis of the partial ICP method is proposed to deal with partially-overlapped problem and scaling effect problem. The 3D face recognition can be achieved by make the comparison of the different distance between the two face images after fine alignment. The procedure of 3D face recognition is also given in this chapter, a set of experiments are designed to test the proposed 3D face recognition method at the end of this chapter.

5.1 Partial ICP Method

The three images of different views for each individual are collected in the database as is described in Chapter 3, meaning that the three images are partially overlapped, some parts of one face image are not contained in another face image. In fine alignment, the partial ICP method is selected to deal with the partially-overlapped problem.

5.1.1 Introduction

The partial ICP method was firstly proposed by Zhang in 1994[44], before the partial ICP method has been proposed, the ICP method was firstly introduced by Chen in

1991[83] and Besl and Mckay in 1992[39]. The ICP method is the one that iteratively aligns two 3D point sets together by using the motion matrix computed from the corresponding point pairs. The important difference between the ICP method and the partial ICP method is that the latter method can deal with the two point sets which are partially overlapped. Due to this advantage, the partial ICP method can deal well with gross errors and appearance, disappearance and occlusion of the objects[44].

The objective function of the partial ICP method is:

$$\mathcal{F}(\boldsymbol{R},\boldsymbol{t}) = \frac{1}{\sum_{i=1}^{l} p_i} \sum_{i=1}^{l} p_i d^2(\boldsymbol{R}\boldsymbol{x_i} + \boldsymbol{t}, D'), \tag{5.1}$$

where $\boldsymbol{x_i}$ is the i-th point in the model point set, l is the number of point in the model point set. $\boldsymbol{R}$ is the rotation matrix (3×3) and $\boldsymbol{t}$ is the translation matrix (3×1) between model point set and reference. D' is the reference image, $d^2(\boldsymbol{R}\boldsymbol{x_i} + \boldsymbol{t}, D')$ is the square of the Euclidean distance from point $\boldsymbol{x_i}$ to the surface D'. p_i is a weight factor to decide whether the point pair is a corresponding point pair or not, p_i could be equal to 0 or 1. If $\boldsymbol{x_i}$ can be matched to one point in D', p_i takes 1, otherwise p_i takes 0.

There being the case, $p_i = 1$ is needed to be considered. The objective function becomes:

$$\mathcal{F}(\boldsymbol{R},\boldsymbol{t}) = \frac{1}{l_m} \sum_{i=1}^{l_m} \|\boldsymbol{R}\boldsymbol{x_i} + \boldsymbol{t} - \boldsymbol{y_i}\|^2, \tag{5.2}$$

where $\boldsymbol{y_i}$ is the i-th corresponding point of $\boldsymbol{x_i}$ in reference. l_m is the number of the correspondences.

The ICP method always converges monotonically to a local minimum with respect to the mean square distance in the objective function[39]. This theorem indicates that the distance between the two point sets is decreasing through every iteration, in other words, the locations between the two point sets are getting closer recursively.

In Figure 5.1 is illustrated the process and the results during the iterations of partial ICP method. The original status of two point sets are given in the very left-side in Figure 5.1, after given the three status during the iteration, the well alignment result is shown in the very right-side of Figure 5.1. The procedures of the partial ICP method can be divided into three steps to achieve fine alignment: (1) finding corresponding point pairs, (2) rejection unreasonable point pairs and (3) the rotation and translation computation and alignment. Then the two point sets will be aligned with several iterations of repeating these three steps.

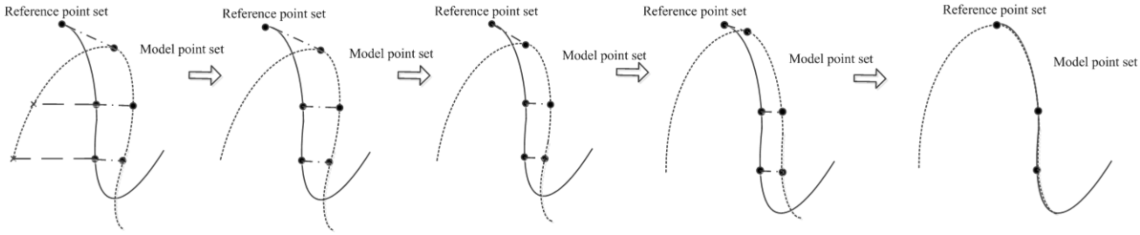


Figure 5.1: Alignment procedure in every iteration.

5.1.2 Procedure of Partial ICP

1. Finding of Corresponding Point Pairs

Since the pseudo corresponding point pairs are the closest point pairs in the two 3D point sets mentioned in the previous section, it is needed to figure out the points on the reference which have the closest distance for the corresponding points in model. The distance between one 3D point $\boldsymbol{x}$ in model data set to the reference data set $D^{'}$ is defined as:

$$d(\boldsymbol{x_i}, D') = \min_{\boldsymbol{y_i} \in D'} d(\boldsymbol{x_i}, \boldsymbol{y_i}), \qquad (5.3)$$

where $d(\boldsymbol{x_i}, D^{'})$ is the minimum Euclidean distance between one 3D point $\boldsymbol{x_i}$ from the model and the reference $D^{'}$. Say $\boldsymbol{y_i}$ is the closest point in the reference of $\boldsymbol{x_i}$ in the model, $d(\boldsymbol{x_i}, \boldsymbol{y_i})$ is the Euclidean distance between the 3D point $\boldsymbol{x_i}$ from model set and a 3D point $\boldsymbol{y_i}$ from reference set. In Cartesian coordinates, 3D point $\boldsymbol{x_i}$ = (x_1, x_2, x_3) and $\boldsymbol{y_i}$ = (y_1, y_2, y_3). Where (x_1, x_2, x_3) and (y_1, y_2, y_3) are the coordinates of point $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$. It can be described like this:

$$d(\boldsymbol{x_i}, \boldsymbol{y_i}) = d(\boldsymbol{y_i}, \boldsymbol{x_i}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} = \|\boldsymbol{x_i} - \boldsymbol{y_i}\|. \qquad (5.4$$

The cost for finding out one closest point in reference is O(N_l), where *l* is the number of points in reference. The total cost for every point in reference is O($\mathrm{N}_m\mathrm{N}_l$), where *m* is the number of points in model. It costs a lot of time and resources for computing the closest distance between the model point set and reference point set. To reduce the computational cost, K-D Tree method [84] is employed to find the closest points. K-D Tree is to build a new structure for points by segment the space into several parts. It can be used in the k-dimensional space. Here the 3D Tree is used to organize the points. The worst searching time by using K-D Tree will be reduced to O(log N_l).

2. Rejection of Unreasonable Point Pairs

The core point of the partial ICP is to find out the corresponding point pairs between the two 3D point sets and reject the non-correspondences in order to deal with the outliers and disappearances. The pseudo correspondences have been selected in the previous step. Then the corresponding point pairs can be determined by using a distance threshold D_{max}.

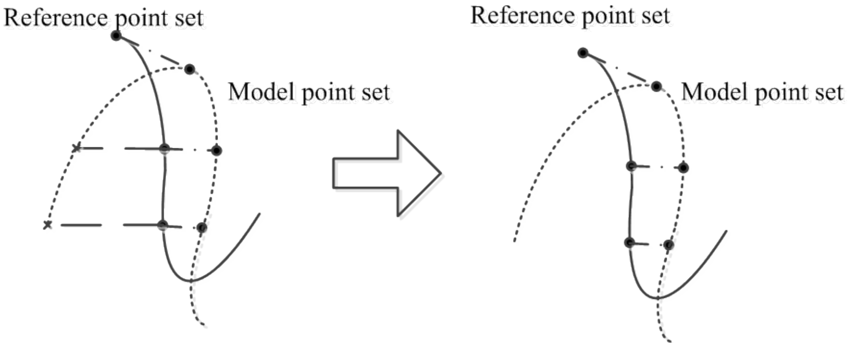


Figure 5.2: Discard unreasonable point pairs.

The process of discarding the incorrect corresponding point pairs can be shown in Figure 5.2. In the left-hand side in Figure 5.2, there are five points in the model point set corresponding to three points in the reference point set, which are selected via the closest points selection criteria. However, there are two unreasonable point pairs indicated by the dash lines which connect the cross points in reference and the dots in the model. These two unreasonable point pairs need to be discarded by using the threshold D_{max}. The result after discarding the unreasonable point pairs is shown in the right-hand side in Figure 5.2. As long as the distance of one point pair is larger than the threshold D_{max}, it will be discarded like the points denoted as cross in Figure 5.2, meanwhile, $p_i = 0$ in the objective function (5.1). The remaining point pairs are the corresponding point pairs, and $p_i = 1$. After several iterations, a two well-aligned point sets are obtained.

The distance threshold D_{max} is an adaptive threshold, its value will adjust the location to change the model. In iteration I, ${D_{max}}^I$ is denoted as the threshold. The ${D_{max}}^I$ is set as [44]:

$$\text{if } \mu < \mathcal{D}, D_{max}^I = \mu + 3\sigma,$$

$$\text{elseif } \mu < 3\mathcal{D}, D_{max}^I = \mu + 2\sigma,$$

$$\text{elseif } \mu < 6\mathcal{D}, D_{max}^I = \mu + \sigma,$$

$$\text{else } D_{max}^I = \xi.$$

A parameter $\mathcal{D}$ is set by user to judge whether the alignment is good or not. In this thesis, $\mathcal{D}$ is set to be equal to the resolution of the reference point set for tolerance. μ is the mean distance of the distances between the closest point pairs from the model to the reference, σ is the deviation of the distances which are given by

$$\mu = \frac{1}{l_m} \sum_{i=1}^{l_m} d_i, \tag{5.5}$$

$$\sigma = \sqrt{\frac{1}{l_m} \sum_{i=1}^{l_m} (d_i - \mu)^2}, \tag{5.6}$$

where d_i is the distance between the i-th corresponding point pair. l_m is the number of the closest point pairs in current iteration.

In the ${D_{max}}^{I}$ setting equation, there is a value ξ need to be set by the user as well. ξ is a threshold when the alignment is very poor to keep the distances convergence. To set ξ, a histogram is firstly built up when the case of $\mu > 6\mathcal{D}$ happens (Figure 5.3). The horizontal axis is the normalized distances of the closest point pairs, the vertical axis is the number of points. Secondly, it is easy to find out the highest number of points and the distance which is the first valley after the highest number of points. Meanwhile, the number of the points corresponding to the distance at the first valley must not go far beyond 50% of the highest number of point. The value of ξ is determined equal to the distance corresponding to the first valley. The highest number of points is 46 when the distance is 0.065, according to the ξ selection criteria, ξ can be determined which is equal to 0.075, because there are 14 numbers of points in this distance, the ratio of which between the number of points corresponding to ξ and the highest number of points is 14/46=30.43% less than 50%.

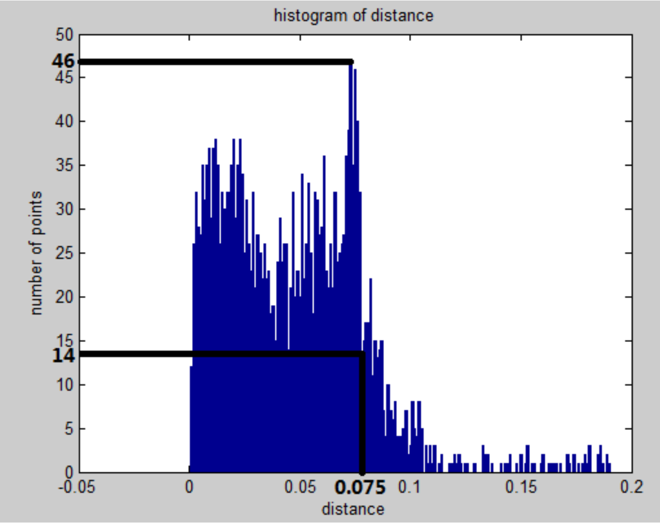


Figure 5.3: Histogram of distances.

3. Rotation and Translation Computation and Fine Alignment

The rotation matrix $\boldsymbol{R}$ and translation matrix $\boldsymbol{t}$ can be computed via the Dual Quaternion method that has been already previously discussed in Section 4.3. The alignment is made by using the equation:

$$\boldsymbol{x}^{I+1} = \boldsymbol{R}^{I}\boldsymbol{x}^{I} + \boldsymbol{t}^{I}, \tag{5.7}$$

where $\boldsymbol{x}^{I}$ is the model face image in the I-th iteration, $\boldsymbol{x}^{I+1}$ is the new probing face image after tuning with the I-th motion (rotation $\boldsymbol{R}^{\mathrm{I}}$ and translation $\boldsymbol{t}^{\mathrm{I}}$), $\boldsymbol{x}^{I+1}$ will be the input of the next iteration. Repeat these three steps during several iterations, the two point sets will be convergence to a global minimum.

5.2 Scaling Effect Computation

5.2.1 Introduction

The partial ICP is an accurate method for data registration and object recognition, however, it is a rigid transformation method without any consideration of its scaling effects. In the reality, the scaling effect exists in every face image in the database since the distances will be diverse between the different faces and the camera. In Figure 5.4 is illustrated the reason why scale difference exists. Point O is the position of one pinhole camera. Position A and B are the positions of the probing object. We can see, from this figure, that the position of object 2 is closer to the camera than that of the object 1. In position C, we can get the images of the objects in the camera. It is shown that if the distance between the probing object and the camera is different, the scale in the images captured from the camera will be different as well. The larger the distance exists, the smaller scale will be gotten from the camera.

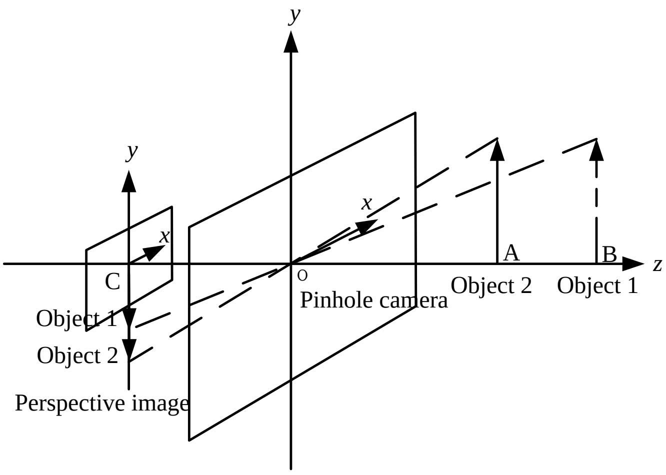
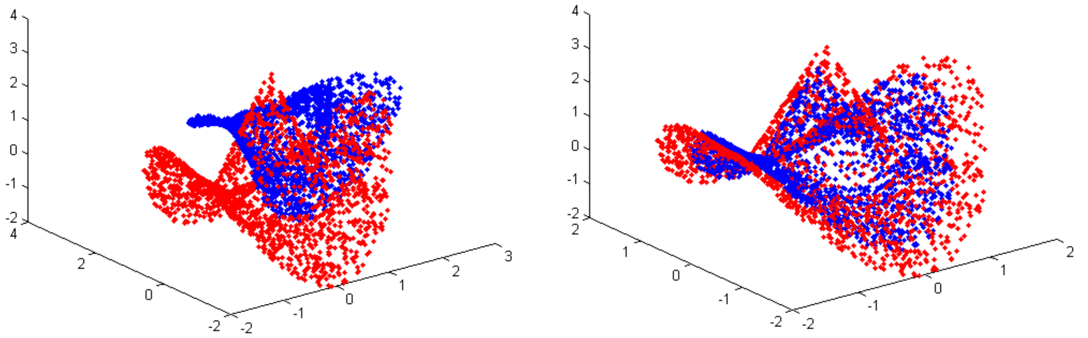


Figure 5.4: Description of scale difference.

If the partial ICP method is implemented without any modification, the scale difference will influence the recognition results. The results of the alignment of two scale different data sets via the partial ICP method are given in Figure 5.5. The point sets both in blue and red are two artificial data sets. The one in red is treated as the reference and the other in blue is the model. The model is got by manually rotated, translated and reduced the scale from the reference. In Figure 5.5(a) is shown the initial status of two point sets, in Figure 5.5(b) is shown the results after applying partial ICP method, we can see that the two point sets with scaling effect cannot align together even the two point sets are from the same object.



(a) Initial status of two artificial data sets.

(b) Final status of two artificial data sets.

Figure 5.5: Results of implementing partial ICP with scaling effect.

5.2.2 Scaling Effect Computation

Here a solution is presented to solve the scaling effects problem. Du, *et al*[54] gave an efficient method called Scaling ICP (SICP) to solve the scaling effect problem. In this thesis the proposed fine alignment method is using the scale matrix computation method from SICP combined with the partial ICP method. The scale matrix computation method can be described as listed below.

If a scale matrix **S** is inserted into the objective function (5.1), the new objective function will be:

$$\mathcal{F}(\boldsymbol{R},\boldsymbol{S},\boldsymbol{t}) = \frac{1}{\sum_{i=1}^{l} p_i}\sum_{i=1}^{l} p_i d^2((\boldsymbol{RSx_i}+\boldsymbol{t}),D'). \tag{5.8}$$

The objective function of the matched point pairs becomes:

$$\mathcal{F}(\boldsymbol{R},\boldsymbol{S},\boldsymbol{t}) = \frac{1}{l_m}\sum_{i=1}^{l_m}\|(\boldsymbol{RSx_i}+\boldsymbol{t})-\boldsymbol{y_i}\|^2, \tag{5.9}$$

where $\boldsymbol{y_i}$ is the corresponding points in reference, l_m is the number of corresponding point pairs.

We need to achieve the minimum of the new objective function (5.9).

If $\mathcal{F}(\boldsymbol{R},\boldsymbol{S},\boldsymbol{t}) = \frac{1}{l_m}\sum_{i=1}^{l_m}\|(\boldsymbol{RSx_i}+\boldsymbol{t})-\boldsymbol{y_i}\|^2$ has minimum, it will satisfy the following equation: $\frac{d\mathcal{F}(\boldsymbol{t})}{d\boldsymbol{t}} = 0$.

$$\frac{d\mathcal{F}(\boldsymbol{t})}{d\boldsymbol{t}} = \frac{2\boldsymbol{S}}{l_m}\sum_{i=1}^{l_m}((\boldsymbol{RSx_i}+\boldsymbol{t})-\boldsymbol{y_i}) = 0. \tag{5.10}$$

Hence, we have $\boldsymbol{t} = \frac{1}{l_m}\sum_{i=1}^{l_m}\boldsymbol{y_i} - \frac{1}{l_m}\sum_{i=1}^{l_m}\boldsymbol{RSx_i}$.

Therefore, the objective function is as follows in order to achieve the minimum.

$$\mathcal{F}(\boldsymbol{R},\boldsymbol{S}) = \frac{1}{l_m}\sum_{i=1}^{l_m}\left\|\boldsymbol{RS}\left(\boldsymbol{x_i}-\frac{1}{l_m}\sum_{i=1}^{l_m}\boldsymbol{x_i}\right)-\left(\boldsymbol{y_i}-\frac{1}{l_m}\sum_{i=1}^{l_m}\boldsymbol{y_i}\right)\right\|^2. \tag{5.11}$$

In Section 4.3, the rotation matrix is given: $\boldsymbol{R} = (q_4^2-\boldsymbol{q^Tq})\boldsymbol{I}+2\boldsymbol{qq^T}+2q_4\boldsymbol{K}(\boldsymbol{q})$. It is an orthogonal matrix. Thus, $\boldsymbol{R^TR}=\boldsymbol{I}$. Where $\boldsymbol{I}$ is an identity matrix.

Let:

$$\boldsymbol{u_i} = \boldsymbol{x_i} - \frac{1}{l_m}\sum_{i=1}^{l_m}\boldsymbol{x_i}, \tag{5.12}$$

$$\boldsymbol{v_i} = \boldsymbol{v_i} - \frac{1}{l_m}\sum_{i=1}^{l_m}\boldsymbol{y_i}. \qquad (5.13)$$

Then,

$$\mathcal{F}(\mathbf{R},\mathbf{S}) = \frac{1}{l_m}\sum_{i=1}^{l_m}\|\boldsymbol{RSu_i} - \boldsymbol{v_i}\|^2 = \frac{1}{l_m}\left(\sum_{i=1}^{l_m}\boldsymbol{u_i^T}\boldsymbol{S^2}\boldsymbol{u_i} - 2\sum_{i=1}^{l_m}\boldsymbol{v_i^T}\boldsymbol{RSu_i} + \sum_{i=1}^{l_m}\boldsymbol{v_i^T}\boldsymbol{v_i}\right) \quad . \qquad (5.14)$$

Equation (5.14) can be treated as a parabola with respect to $\boldsymbol{R}$ and $\boldsymbol{S}$. To find out the minimum of this parabola, we can derive the partial differential equation as:

$$\frac{\partial\mathcal{F}(\boldsymbol{R},\boldsymbol{S})}{\partial\boldsymbol{R}} = 0, \qquad (5.15)$$

$$\frac{\partial\mathcal{F}(\boldsymbol{R},\boldsymbol{S})}{\partial\boldsymbol{S}} = 0. \qquad (5.16)$$

In order to get the scale matrix from equation (5.14), we can get

$$\frac{\partial\mathcal{F}(\boldsymbol{R},\boldsymbol{S})}{\partial\boldsymbol{S}} = 2\sum_{i=1}^{l_m}\boldsymbol{u_i^T}\boldsymbol{SE_j}\boldsymbol{u_i} - 2\sum_{i=1}^{l_m}\boldsymbol{v_i^T}\boldsymbol{RE_j}\boldsymbol{u_i} = 0, \qquad (5.17)$$

where $\boldsymbol{E_j} = diag(0,\dots,0,1,0,\dots,0), (j = 1,2,3)$ is a diagonal matrix, j-th element is 1 while others are 0. j is a scalar which indicates the dimension. Here $j = 1,2,3$ as is indicated that the data set is a 3D data set.

Scale factor in j-th dimension S_j could be computed from equation (5.17)

$$S_j = \frac{\sum_{i=1}^{l_m}\boldsymbol{v_i^T}\boldsymbol{RE_j}\boldsymbol{u_i}}{\sum_{i=1}^{l_m}\boldsymbol{u_i^T}\boldsymbol{E_j}\boldsymbol{u_i}}. \qquad (5.18)$$

In coarse alignment section, it has been discussed that the scale matrix is set in the interval of $[\boldsymbol{S_{ini}} - \delta, \boldsymbol{S_{ini}} + \delta]$. If $S_j \in \left[S_{ini\_j} - \delta, S_{ini\_j} + \delta\right]$, the minimum is the point which is nearest to the vertex of the parabola (5.14), the scale can be computed:

$$S_j = \arg\min_{\boldsymbol{s} \in [\boldsymbol{S_{ini}} - \delta, \boldsymbol{S_{ini}} + \delta]} \left| \boldsymbol{s} - \frac{\sum_{i=1}^{l_m} \boldsymbol{v}_i^T \boldsymbol{R} \boldsymbol{E}_j \boldsymbol{u}_i}{\sum_{i=1}^{l_m} \boldsymbol{u}_i^T \boldsymbol{E}_j \boldsymbol{u}_i} \right|. \tag{5.19}$$

If $S_j < S_{ini} - \delta,\ S_j = S_{ini} - \delta$, if $S_j > S_{ini} + \delta, S_j = S_{ini} + \delta$.

The scale matrix can be written on the basis of scale factor S_j:

$$\boldsymbol{S} = \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix}, \tag{5.20}$$

where S_1, S_2, S_3 are the scale factors of $S_j,\ j = 1{,}2{,}3$.

The other two matrices $\boldsymbol{R}$ and $\boldsymbol{t}$ can be compute by using Dual Quaternion method described in Section 4.3.

5.3 Procedure of the Modified ICP Method

5.3.1 Procedure

The procedure of the proposed method is:

The ***INPUT*** of the proposed method: two 3D point sets: model point set $\boldsymbol{x_i}$, reference point set D', the initial value of rotation $\boldsymbol{R}^0$, translation $\boldsymbol{t}^0$, scale $\boldsymbol{S}^0$, distance threshold $D_{max}{}^0$.

The ***OUTPUT*** of the proposed method: an optimal motion consists of a rotation matrix $\boldsymbol{R}$, a translation matrix $\boldsymbol{t}$ and a scale matrix $\boldsymbol{S}$.

Iteration begins:(in I-th iteration)

Step 1: find the closest points $\boldsymbol{y_i}^I$ in reference D' corresponding to every point from model $\boldsymbol{x_i}^I$. The number of the corresponding point pairs is l_m. Discard the unreasonable closest point pairs with the distance threshold $D_{max}{}^I$.

Step 2: compute the motion between two point sets (compute $\boldsymbol{R}^I$, $\boldsymbol{t}^I$ and $\boldsymbol{S}^I$).

Step 3: apply the motion to the model ($\boldsymbol{x_i}^{I+1} = \boldsymbol{R}^I\boldsymbol{S}^I\boldsymbol{x_i}^I + \boldsymbol{t}^I$), the objective function $\mathcal{F}(\boldsymbol{R},\boldsymbol{t}) = \frac{1}{l_m}\sum_{i=1}^{l_m}||\boldsymbol{R}^I\boldsymbol{S}^I\boldsymbol{x_i}^I + \boldsymbol{t}^I - \boldsymbol{y_i}^I||^2 = \frac{1}{l_m}\sum_{i=1}^{l_m}||\boldsymbol{x_i}^{I+1} - \boldsymbol{y_i}^I||^2$ achieves the minimum.

Iteration ends if:

The condition of termination is to satisfy the termination constraint. For the terminate condition, if $||\boldsymbol{R}^I - \boldsymbol{R}^{I-1}|| < \varepsilon_R$ and $||\boldsymbol{t}^I - \boldsymbol{t}^{I-1}|| < \varepsilon_t$, the iteration ends, $\boldsymbol{x_i}^{\sim} = \boldsymbol{x_i}^I$. Where $\varepsilon_R, \varepsilon_t$ are the thresholds for rotation and translation.

The modified ICP method can be concluded in the flow chart below:

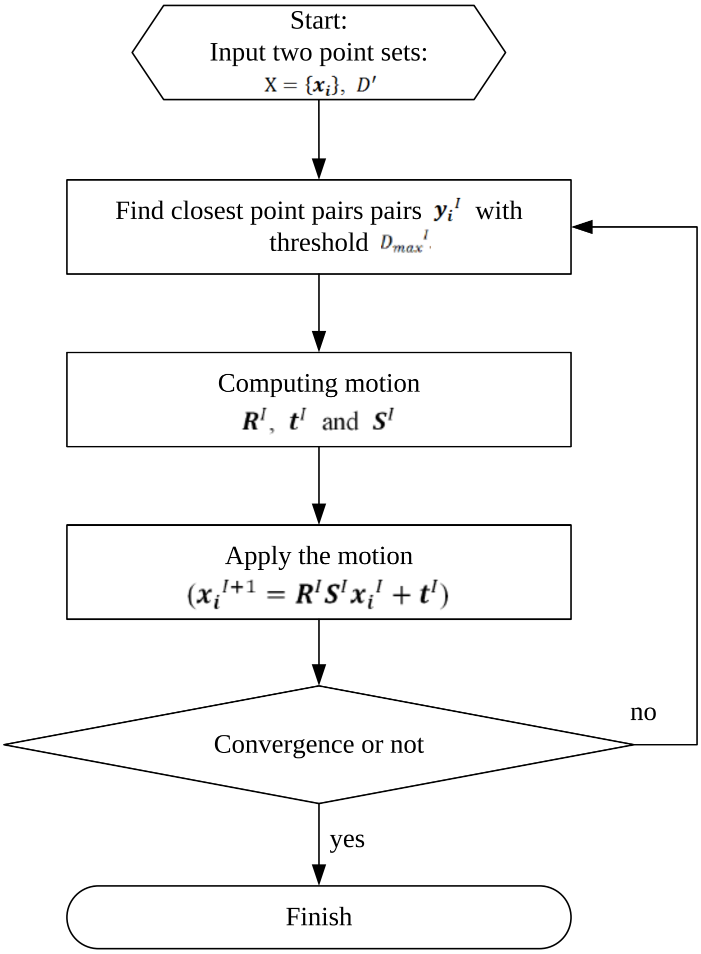

Start:
Input two point sets:
$X = \{\boldsymbol{x}_i\}, D'$
Find closest point pairs pairs $\boldsymbol{y}_i{}^I$ with threshold $D_{max}{}^I$.
Computing motion
$\boldsymbol{R}^I$, $\boldsymbol{t}^I$ and $\boldsymbol{S}^I$
Apply the motion
$(\boldsymbol{x}_i{}^{I+1} = \boldsymbol{R}^I \boldsymbol{S}^I \boldsymbol{x}_i{}^I + \boldsymbol{t}^I)$
Convergence or not
no
yes
Finish

Figure 5.6: Flow chart of modified ICP method.

5.3.2 Convergence Theorem

The modified ICP method will converge to a global minimum after several iterations. The mean square distances of the corresponding point pairs in the objective function will be reduced during every iteration, whereas the location between reference and model will also get closer during the iteration. The convergence theorem and proof are listed below.

Theorem: The modified ICP method is always convergence monotonically to a local minimum with respect to the mean square distance.

Proof: in I-th iteration, given two point sets: model point set $\boldsymbol{x_i}^{\boldsymbol{I}}$ and corresponding points $\boldsymbol{y_i}^{\boldsymbol{I}}$ in reference D', scale matrix $\boldsymbol{S}^I$, rotation matrix $\boldsymbol{R}^I$, translation matrix $\boldsymbol{t}^I$. Let $\boldsymbol{x_i}^{I+1} = \boldsymbol{R}^I\boldsymbol{S}^I\boldsymbol{x_i}^I + \boldsymbol{t}^I$ which allows the objective function to be the minimal.

According to Step 1 in the procedure of the modified ICP method and supposing that, in the I-th iteration, we have $e^I = \frac{1}{l_m}\sum_{i=1}^{l_m}\|\boldsymbol{x_i}^I - \boldsymbol{y_i}^I\|^2$. Once applied the best scale $\boldsymbol{S}^I$, rotation $\boldsymbol{R}^I$ and translation $\boldsymbol{t}^I$ is applied, the new square distance becomes: $\varepsilon^I = \frac{1}{l_m}\sum_{i=1}^{l_m}\|\boldsymbol{x_i}^{I+1} - \boldsymbol{y_i}^I\|^2$. Because ε^I is the minimum, there is a relationship: $\varepsilon^I \le e^I$. If $\varepsilon^I > e^I$, the least squares is larger than the identity transformation which is impossible. In the next iteration I+1, $e^{I+1} = \frac{1}{l_m}\sum_{i=1}^{l_m}\|\boldsymbol{x_i}^{I+1} - \boldsymbol{y_i}^{I+1}\|^2$ will be smaller than $\varepsilon^I = \frac{1}{l_m}\sum_{i=1}^{l_m}\|\boldsymbol{x_i}^{I+1} - \boldsymbol{y_i}^I\|^2$ in the previous iteration. Because in I+1-th iteration, the process will repeat Step 1 to find out the closest point pairs with the point set $\boldsymbol{x_i}^{I+1}$ which is applied motion with $\boldsymbol{S}^I$, $\boldsymbol{R}^I$ and $\boldsymbol{t}^I$. The square distance between $\boldsymbol{x_i}^{I+1}$ and $\boldsymbol{y_i}^I$ is closer than before. If the closest distance is larger than it in Step 3 the previous iteration, it is not the closest distance. So we have: $0 \le \varepsilon^{\sim} \le e^{\sim} \le \cdots \le e^{I+1} \le \varepsilon^I \le e^I \le \cdots \le \varepsilon^1 \le e^1$, for all iteration. From this relation, it can be concluded that the proposed fine alignment method converges monotonically to a minimum with respect to the mean square distance.

5.4 3D Face Recognition

The 3D face recognition can be achieved by make the comparison of the different distance between the two face images. The point-to-point distance minimized by the modified ICP method is used as the matching distance of the two face images (Equation (5.21)).

$$\mathcal{F}(\boldsymbol{R}, \boldsymbol{S}, \boldsymbol{t}) = \frac{1}{l_m} \sum_{i=1}^{l_m} \|\boldsymbol{x_i} - \boldsymbol{y_i}\|^2, \tag{5.21}$$

where $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ are the i-th corresponding point pairs in the probing image and reference respectively, l_m is the number of correspondences. $\|\boldsymbol{x_i} - \boldsymbol{y_i}\|^2$ is the square distance between the i-th corresponding point pair. $\mathcal{F}(\boldsymbol{R}, \boldsymbol{S}, \boldsymbol{t})$ is denoted as Mean Square Error (MSE) for further comparison. If the matching distance is smaller than a threshold, the two face images are treated as the identical face, otherwise the two face images are from the imposter faces.

The whole procedure of 3D face recognition is given in this section. In Figure 5.7 is shown the flowchart of the proposed method to achieve 3D face recognition. The procedure of 3D face recognition can be concluded as follows:

INPUT: one probing face image which is needed to recognize.

OUTPUT: the recognition result which is given after the database traversal.

Iteration begins:

Step 1: Select one face image from the database, this face image treated as the reference image.

Step 2: Select the feature points to compute the motion and coarsely align the two face images by the motion estimation.

Step 3: Fine align the reference image with the model image on the basis of the modified ICP method.

Step 4: Compute the mean square distance between the two point sets by using the equation $\mathrm{MSE} = \frac{1}{l_m}\sum_{i=1}^{l_m}\|\boldsymbol{x_i}^{I+1} - \boldsymbol{y_i}^{I}\|^2$, where MSE is the Mean Square Error, meaning the mean square distance between the two point sets.

Step 5: Compare the mean square distance.

Iteration ends if:

If MSE is smaller than a threshold, the two face images are treated as the same. The model image can be recognized after the images compared in the database.

If no image in the database can satisfy the constraint condition, the probing face image is an unknown face image.

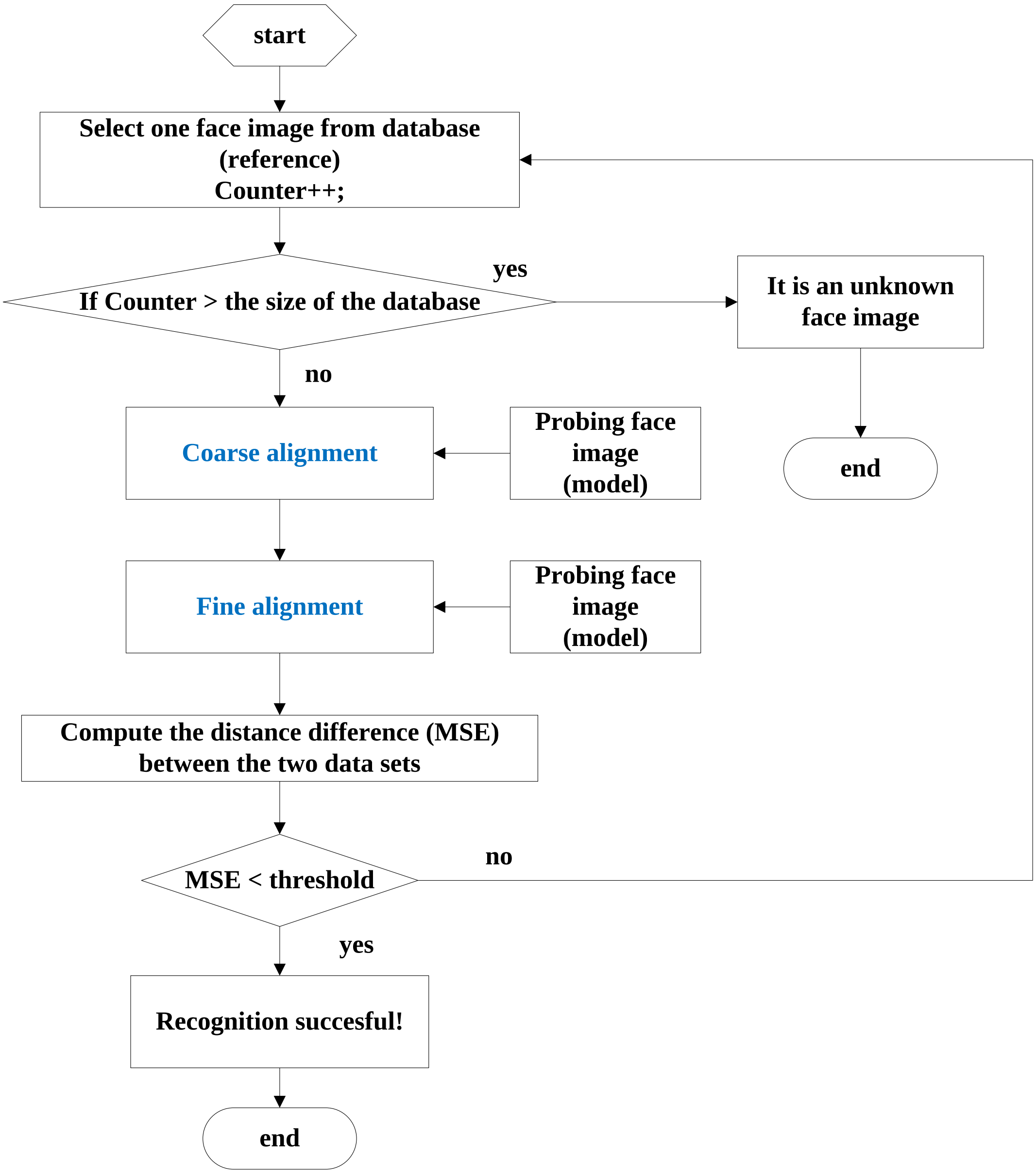

start
Select one face image from database
(reference)
Counter++;
If Counter > the size of the database
yes
It is an unknown face image
end
no
Coarse alignment
Probing face image (model)
Fine alignment
Probing face image (model)
Compute the distance difference (MSE) between the two data sets
MSE < threshold
no
yes
Recognition succesful!
end

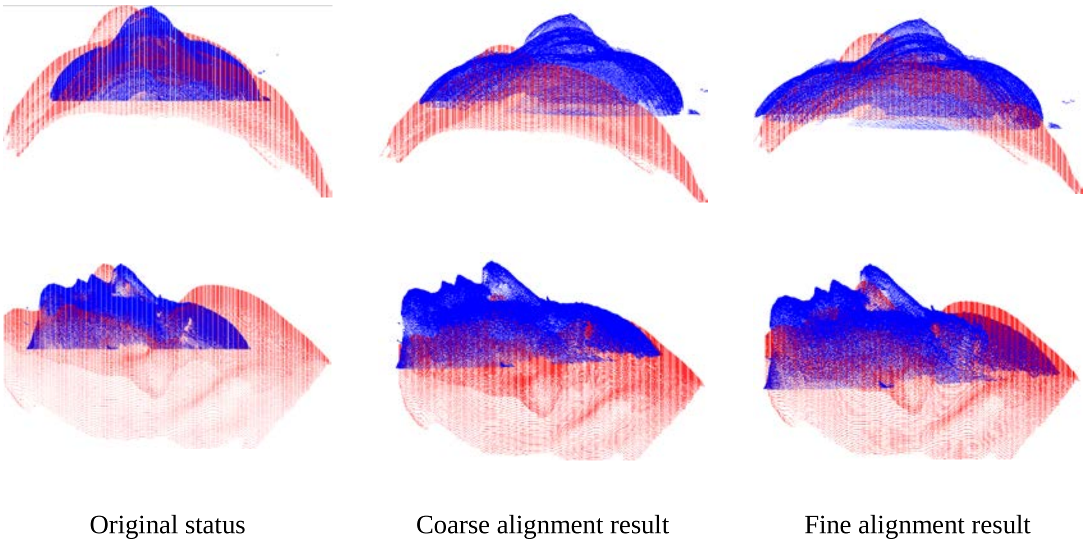
Figure 5.7: Flow chart of 3D face recognition method.

5.5 Experiments

Two sets of experiments are designed to test the proposed method of 3D face recognition. One is to test the functionality of the proposed method while the other is to compare its accuracy.

5.5.1 Experiment I

The objective of experiment I is to test whether the proposed method can achieve to recognize the different face images or not. The input of this experiment is one model image and several reference images selected randomly from the database. Figure 5.8 gives the initial status of two images from the imposter faces before alignment and results after using the proposed method. The left column in Figure 5.8 is shown the initial status of two face images, the middle column gives the coarse alignment result and the right column is the fine alignment result. The results are obtained by employing the proposed method. Although the two face images are aligned after applying the coarse alignment method, the two distinct face images cannot align together as the final alignment result shown in Figure 5.8.



Original status

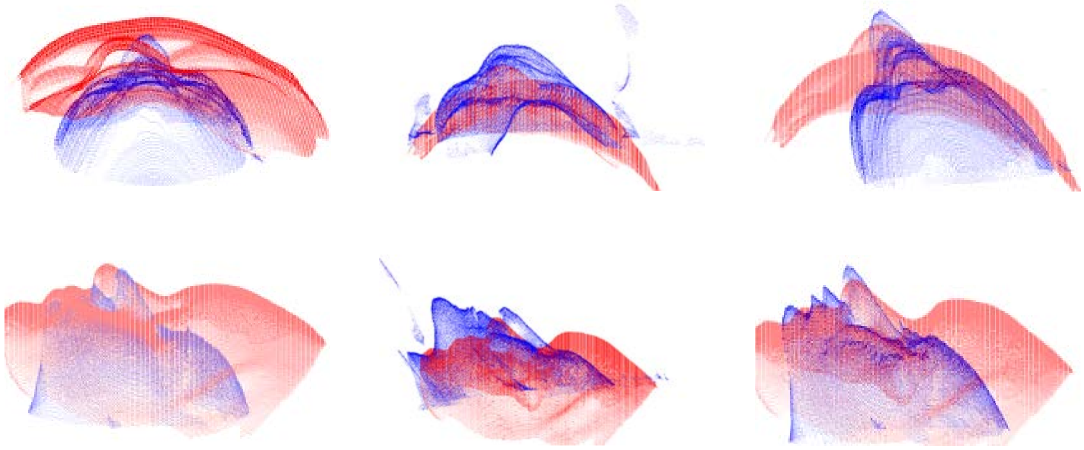
Coarse alignment result

Fine alignment result

Figure 5.8: Alignment results of two imposter face images (imposter face 1).

The corresponding Mean Square Error (MSE) value can be obtained by using the equation $\mathrm{MSE} = \frac{1}{l_m}\sum_{i=1}^{l_m}\|\boldsymbol{x_i}^{I+1} - \boldsymbol{y_i}^{I}\|^2$, which is shown by the red star line (imposter face 1) in Figure 5.12. The horizontal axis is the iteration time and the vertical axis is the MSE value. The MSE value is shown that the two face images are convergence to a minimum although the two face images are from different people.

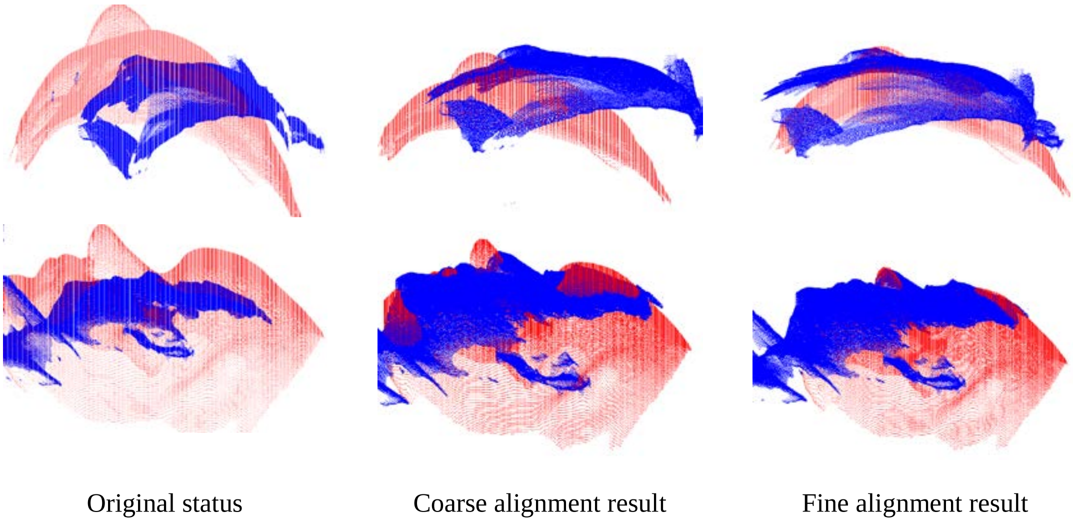
Another different face image is coming to compare with the probing face image in Figure 5.9. The left column in Figure 5.9 is given the initial status of two face images, the coarse alignment results are shown in the middle column, and the final results after fine alignment can be shown in the right column. The two face images are getting closer after the coarse alignment, but the two face images cannot align which is shown from the final results because these two face images are from different faces. The corresponding MSE is shown in the black square line (imposter face 2) in Figure 5.12.



Original status Coarse alignment result Fine alignment result

Figure 5.9: Alignment results of two different face images (imposter face 2).

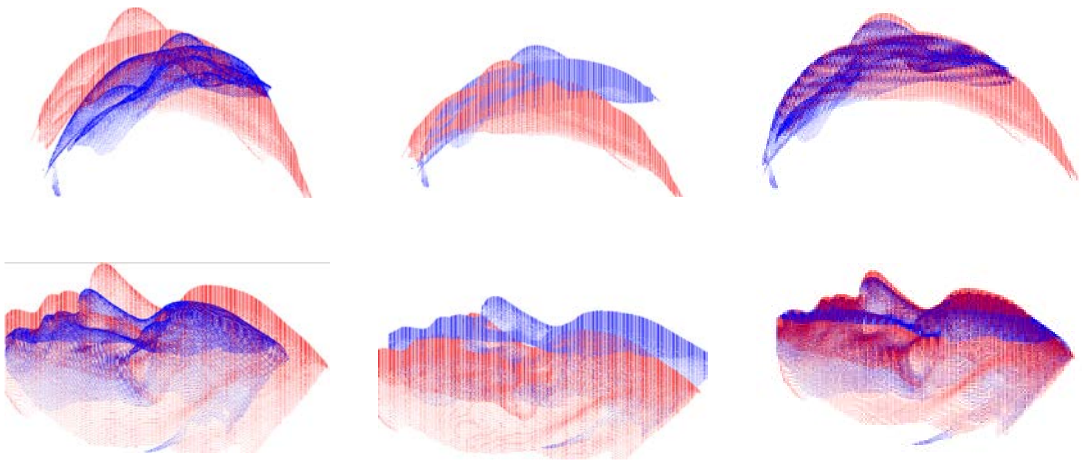
Figure 5.10 shows the comparison of the probing face image with the third imposter face image. The left column in Figure 5.10 is given the initial status of two face images, the coarse alignment results are shown in the middle column, and the final results after fine alignment can be shown in the right column. With the same situation of Figure 5.9, the two face images are getting closer after the coarse alignment, but the two face images cannot align shown from the final results. The corresponding MSE is shown in the green diamond line (imposter face 3) in Figure 5.12.



Original status　Coarse alignment result　Fine alignment result

Figure 5.10: Alignment results of two different face images (imposter face 3).

Figure 5.11 shows the results of the two face images which are from the genuine faces but with different scaling effect. The left column in Figure 5.11 is shown the initial status of two face images, the middle column gives the coarse alignment result and the right column is the fine alignment result. The scale of the probing face image (blue face image) is becoming large, and the location between the two face images are getting closer. After utilizing the proposed 3D face recognition method, we can clearly see that the two point sets align well from the fine alignment results in Figure 5.11. Meanwhile, MSE is quite close to 0 shown in the blue cross line in Figure 5.12.



Original status Coarse alignment result Fine alignment result

Figure 5.11: Alignment results of two genuine face images.

Figure 5.12 shows the MSE values of three groups of comparing different face images and one group of two images from the identical face. It is clear to show that if the two images come from the same person, the MSE will much less than the MSE value of imposter face images and close to the value of zero, in other words, after doing the coarse alignment and fine alignment, the proposed method will recognize the right face image from reference images by comparing MSE value. Meanwhile, the MSE value from Figure 5.12 is decreasing with the increasing iteration time which is shown the proposed method is a convergence method.

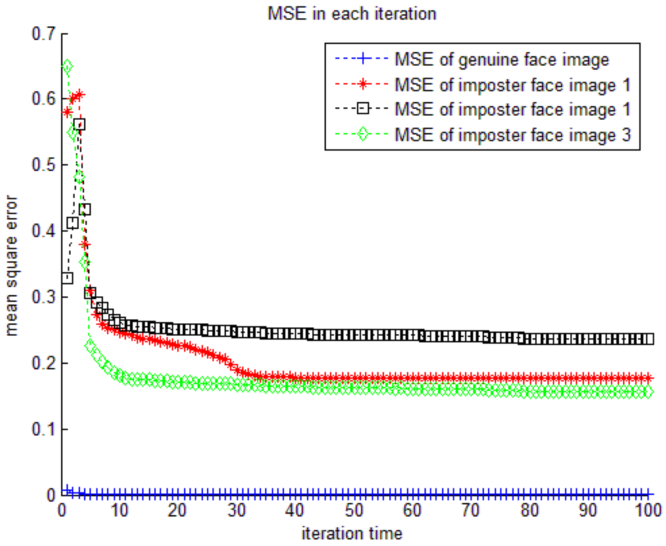


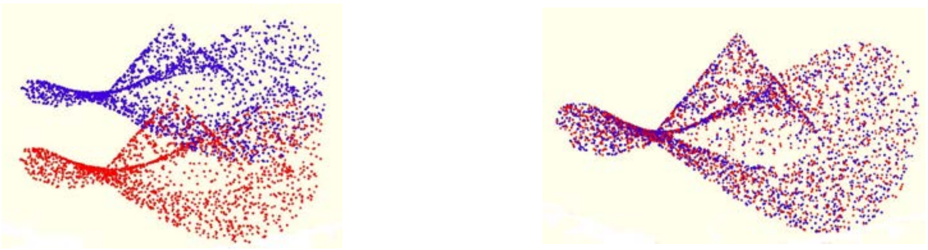
Figure 5.12: MSE of 3 different face images and one genuine face image.

5.5.2 Experiment II

This experiment is to compare the accuracy of the proposed method with the partial ICP method. Two experiments are contained in this set of experiment. The artificial data sets are firstly used to compare the accuracy of the proposed method and the partial ICP method followed by the real data sets.

1. Artificial Data

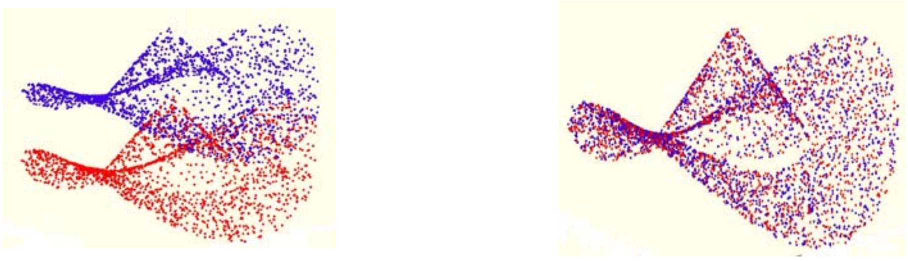
The input of this experiment is two artificial data sets are shown in the left-side of Figure 5.13. The one in red is treated as the reference and the other in blue is the model. The model is obtained by manually rotated, translated and reduced the scale from the reference. The results of the two genuine face images with the same scaling effects by using the partial ICP method are given in Figure 5.13. The results of using the proposed method to achieve fine alignment are shown in Figure 5.14. The left-side status is the initial status of the two point sets and the alignment result is given in the right-side both in Figure 5.13 and Figure 5.14 which are shown that the fine alignment results are quite good. The MSE value can be calculated via Equation (5.21). The comparison of MSE can be shown Figure 5.15, the horizontal axis is the iteration time, the vertical axis is the MSE value, the MSE value of every iteration of the proposed method is shown as the solid line, the MSE value of every iteration of the partial ICP method is shown as the dash line. Both the proposed method and the partial ICP method can reach a similar small MSE value that can successfully achieve 3D data sets fine alignment.



Original status

Fine alignment result

Figure 5.13: Alignment results of two artificial data sets with same scaling effects by using partial ICP method.



Original status

Fine alignment result

Figure 5.14: Alignment results of two artificial data sets with same scaling effects by using proposed method.

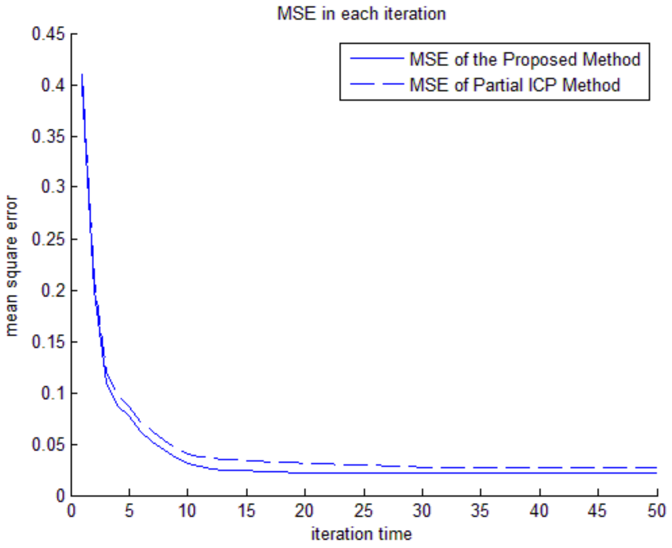
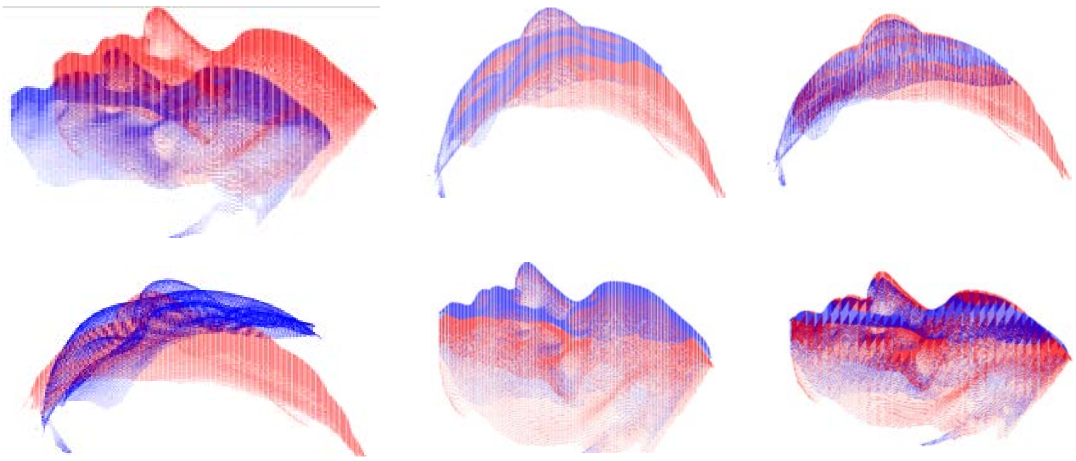


Figure 5.15: MSE of artificial data sets with partial ICP and proposed method.

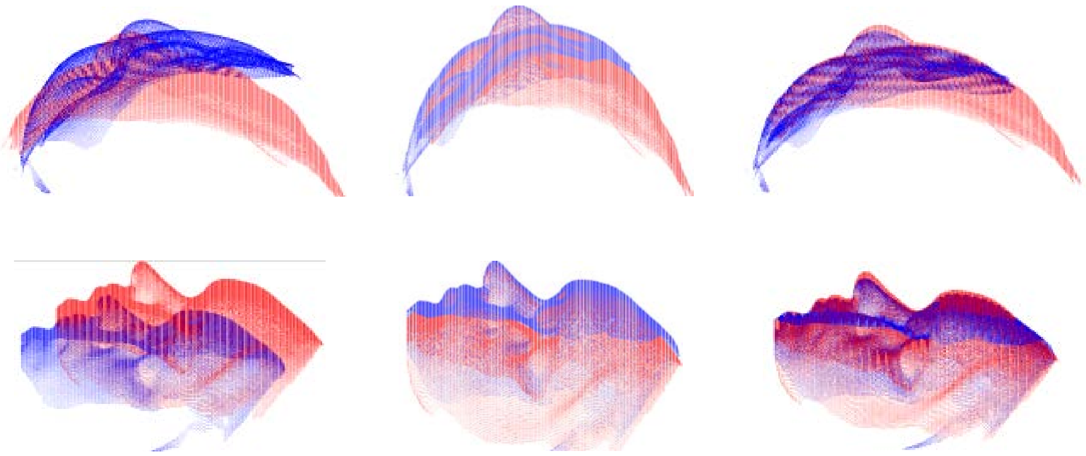
2. Real Data

This experiment is to compare the accuracy of the method with the partial ICP method with the real face images. The genuine face with the same scale is used by employing the partial ICP method and the proposed method. The two face images of the input are from the genuine person and without the scaling effect. The results of the two genuine face images with the same scaling effects by using the partial ICP method in the fine alignment method is shown in Figure 5.16. The left-side column is shown the initial status of the two face images, the middle column is shown the result after applying coarse alignment, the right-side column is shown the results after fine alignment which indicates that the two face images are aligned very well by using the partial ICP method.



Original status Coarse alignment result Fine alignment result

Figure 5.16: Alignment results of two genuine face images with same scaling effects by using partial ICP method.



Original status Coarse alignment result Fine alignment result

Figure 5.17: Alignment results of two genuine face images with same scaling effects by using proposed method.

The results of the two genuine face images with the same scaling effects by using the proposed method are given in Figure 5.17. Comparing with the result by using the partial ICP method, the fine alignment results by using the proposed method can reach the similar status with the partial ICP method. The MSE results of the two different methods are similar which is shown in Figure 5.18, which means the proposed method can achieve the similar accuracy of the the partial ICP.

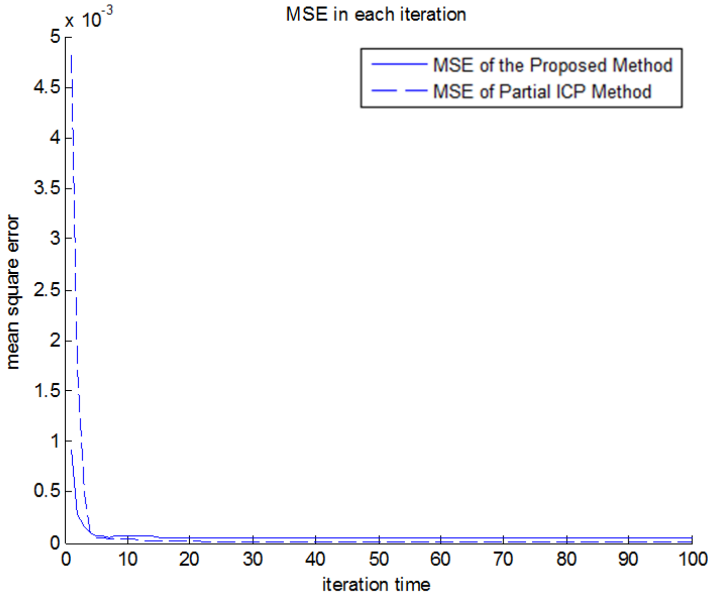
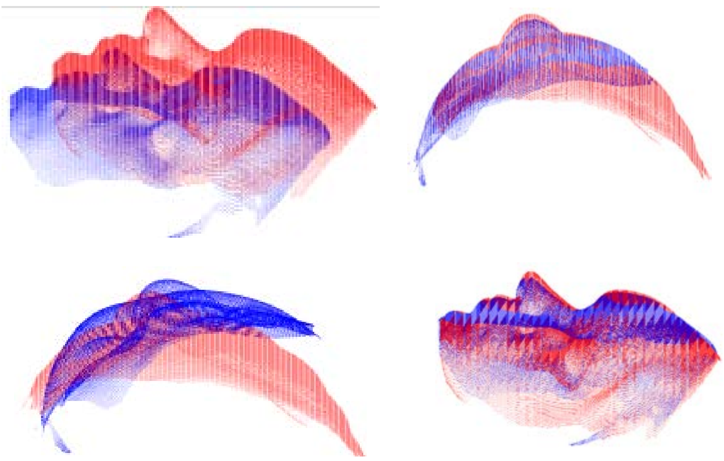


Figure 5.18: MSE of the genuine face images with partial ICP and proposed method.

5.5.3 Experiment III

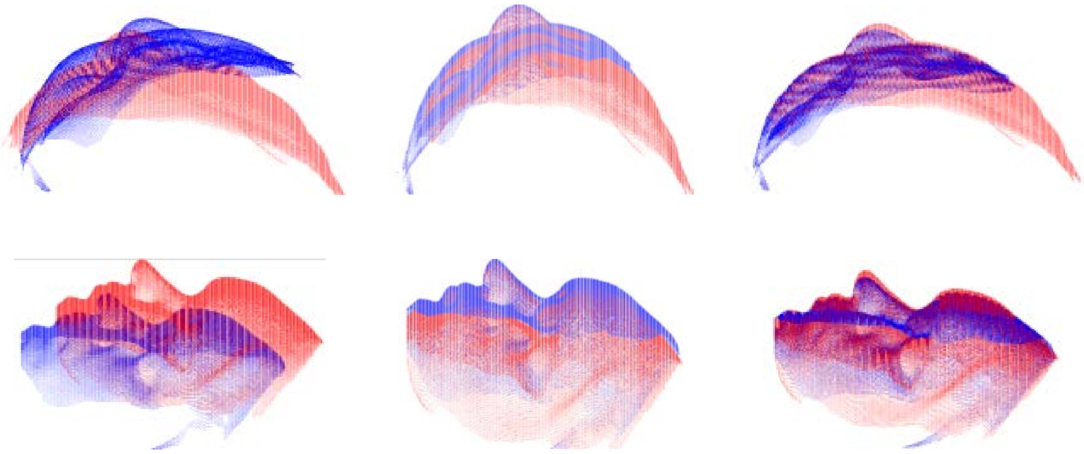
This experiment is to compare the computational time between the method coarse alignment method employed and just only fine alignment. The data used in this experiment is two face images from the same person and without the scaling effect.

The genuine face with the same scale is used by employing the fine alignment method and the proposed method (both coarse alignment method and fine alignment method). The results of the two genuine face images with the same scaling effects by only using the fine alignment method is shown in Figure 5.19. The left-side column is shown the initial status of the two face images, the right-side column is shown the results after fine alignment which indicates that the two face images are aligned by using the fine alignment method.



Original status Fine alignment result

Figure 5.19: Alignment results of two genuine face images with same scaling effects by using only fine alignment method.



Original status Coarse alignment result Fine alignment result

Figure 5.20: Alignment results of two genuine face images with same scaling effects by using proposed method.

The results of the two genuine face images with the same scaling effects by using the proposed method are given in Figure 5.20. The MSE results of the two different methods are similar which is shown in Figure 5.18, which indicates the proposed method only take no more than 10 iterations, the MSE value reaches 10^{-4},while it takes more than 40 iterations for the fine alignment method to reach the same level MSE of the proposed method.

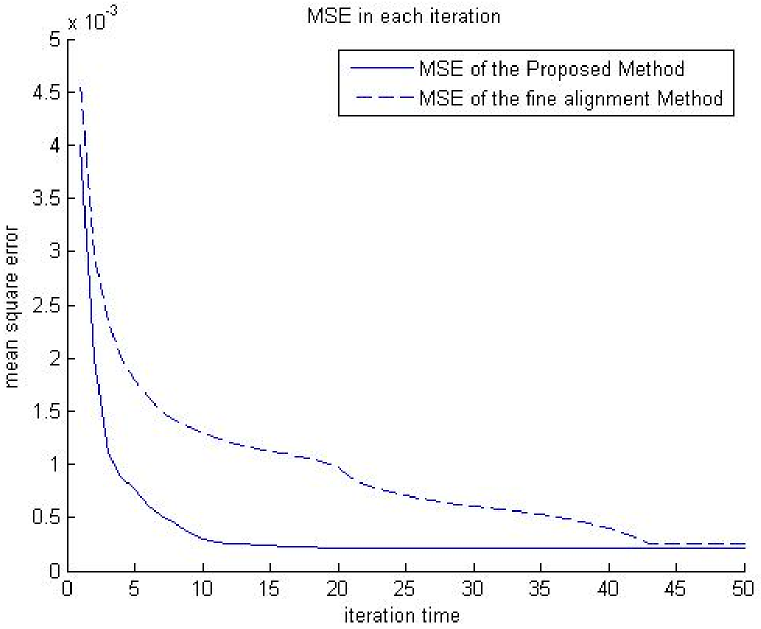


Figure 5.21: MSE of the genuine face images with partial ICP and proposed method.

5.6 Summary

The fine alignment method has been well presented in this chapter. Both the partially-overlapped problem and scaling effect problem have been solved via the fine alignment method. Combining with the scale matrix computation method, the proposed method can deal well with the partially-overlapped problem and scaling effects problem. The 3D face recognition method has also been thoroughly demonstrated in this chapter via the comparison of mean square distances between the face images.

Two sets of experiments are designed to test the functionality and accuracy of the proposed 3D face recognition method. The results of the experiments are shown that the proposed 3D face recognition method can be utilized so as to achieve the recognition of various face images. Meanwhile, the accuracy of the proposed method is keeping the same of the partial ICP method, which can be implemented to achieve 3D face recognition.

CHAPTER 6 CONCLUSION

A 3D face recognition method based on a modified ICP method is proposed for achieving 3D face recognition in this thesis. The proposed method consists of a coarse-to-fine alignment and the distance comparison to recognize the probing face image with the reference images from database. In the coarse alignment, a feature point extraction method is proposed to align the face images into a same coordinates system. A modified ICP method is proposed to overcome the partially-overlapped problem and scaling effect. 3D face recognition can be achieved by the different distance comparison between the two face images. The summary of this thesis is given in Section 6.1 and the future work is drawn and listed in Section 6.2.

6.1 Thesis Summary

This project aims to recognize the 3D face images. A 3D face database has been built up via the DFPP data acquisition system. There are 90 images for 30 people contained in this database. Three images corresponding to three different views with the same scale are represented one individual. The scales of different individuals may be different with one another.

A coarse-to-fine strategy method is developed to achieve 3D face recognition. The scale matrix, rotation matrix and translation matrix are computed by using four feature points in order to roughly align the two face images in coarse alignment. The two face images are one probing face image and one reference image selected randomly from the database. The four feature points are extracted based on the analysis of curvatures. The experiments of the coarse alignment method are carried out to evaluate the performance of the coarse alignment.

A modified ICP method is given in fine alignment. In the database, there are three views to describe one face, therefore the disappearances will exist among these three face images. Meanwhile, the scaling effects are existed in the face images. The partially-overlapped problem can be solved by using a distance threshold D_{max} when the closest points are selected. A scale matrix $\boldsymbol{S}$ is computed to deal with scaling effects problem. The whole procedure of the proposed 3D face recognition method is concluded in Chapter 5. A set of experiments are illustrated to verify the proposed method. The proposed method can achieve 3D face recognition and the accuracy is similar with the partial ICP from the results shown in Section 5.5.

6.2 Future Work

The 3D face recognition procedure by using the face database of DFPP based acquisition technique is stated in this thesis. However, some issues need to be further probed and developed in the future.

✧ Firstly, more and more face images will be added into the face database, it is required much more memory to store them. It is necessary to save computation cost while processing these data. Extracting feature region and only processing the feature region from the face data will reduce the computation cost. In the future, a method for feature region extraction can be discussed.

✧ Secondly, an automatic feature extraction procedure can be developed. It is needed to design a fully auto feature extraction procedure in the coarse alignment part.

REFERENCES

[1] W. Zhao, *et al.*, "Face Recognition: A Literature Survey," *ACM Computing Surveys*, vol. 35, pp. 399-458, 2003.

[2] J.-A. Beraldin, *et al.*, "Active 3D Sensing," in *The e-Way into the Four Dimensions of Cultural Heritage Congress*, Vienna, Austria, 2003, pp. 1-21.

[3] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology (2nd Edition)*: Institution of Engineering and Technology, 2004.

[4] Y. Lin, *et al.*, "3D ultrasound tracking of the left ventricle using one-step forward prediction and data fusion of collaborative trackers " in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.

[5] V. V. Kindratenko, "A survey of electromagnetic position tracker calibration techniques," *Virtual Reality: Research, Development, and Applications*, vol. 5, pp. 169-182, 2000.

[6] B. Gokberk and M. O. I. L. Akarun, "3D shape-based face representation and feature extraction for face recognition," *Image and Vision Computing*, vol. 24, pp. 857-869, 2006.

[7] K. W. Bowyer, *et al.*, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition," *Computer Vision and Image*

Understanding, vol. 101, pp. 1-15, 2006.

[8] D. Smeets, *et al.*, "Objective 3D face recognition: Evolution, approaches and challenges," *Forensic Science International*, vol. 201, pp. 125-132, 2010.

[9] C. Hesher, *et al.*, "A Novel Technique for Face Recognition Using Range Image," in *International Symposium on Signal Processing and Its Applications*, 2003, pp. 201-204.

[10] T.-W. R. Lo, "Feature Extraction for Range Image Interpretation using Local Topology Statistics," Doctor of Philosophy, Department of Computing Science, Faculty of Information and Mathematical Sciences, University of Glasgow, 2009.

[11] P. Besl and R. Jain, "Segmentation through symbolic surface descriptions," in *IEEE Conference on Computer Vision Pattern Recognition*, 1986, pp. 77-85.

[12] C. Dorai and A. K. Jain, "COSMOS-A representation scheme for 3D free-form objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1115-1130, 1997.

[13] J. Y. Cartoux, *et al.*, "Face authentication or recognition by profile extraction from range images," in *Proceedings of the Workshop on Interpretation of 3D Scenes*, 1989, pp. 194-199.

[14] J. C. Lee and E. Milios, "Matching Range Images of Human Faces," in *International Conference on Computer Vision*, 1990, pp. 722-726.

[15] G. G. Gordon, "Face Recognition Based on Depth and Curvature Features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992, pp. 808 - 810

[16] T. K. Kim, *et al.*, "Real-time normalization and feature extraction of 3D face data using curvature characteristics," in *Proceedings of 10th IEEE International Workshop on Robot and Human Interactive Communication,* 2001, pp. 74-79.

[17] R. J. Campbell and P. J. Flynn, "Recognition of free-form objects in dense range data using local features," in *Proceedings of the International Conference on Pattern Recognition*, 2002, pp. 607-610.

[18] A. B. Moreno, *et al.*, "Face recognition using 3D surface-extracted descriptors," in *Irish Machine Vision and Image Processing*, 2003, pp. 56-63.

[19] B. Bhanu and X. Zhou, "Face Recognition from Face Profile Using Dynamic Time Warping," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, pp. 499-502.

[20] Y. Sun and L. Yin, "Evaluation of 3D Facial Feature Selection for Individual Facial Model Identification," in *Proceedings of International Conference on*

Pattern Recognition, 2006, pp. 562-565.

[21] K. I. Chang*, et al.*, "Multiple nose region matching for 3D face recognition under varying facial expression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1695-1700, 2006.

[22] A. Colombo*, et al.*, "3D face detection using curvature analysis," *Pattern Recognition*, vol. 39, pp. 444-455, 2006.

[23] E. Akagündüz and I. Ulusoy, "Scale and Orientation Invariant 3D Interest Point Extraction Using HK Curvatures," in *IEEE 12th International conference on Computer Vision Workshops*, 2009, pp. 692-702.

[24] C. S. Chua and R. Jarvis, "Point Signatures: A New Representation for 3D Object Recognition," *International Journal of Computer Vision*, vol. 25, pp. 63-85, 1997.

[25] C.-S. Chua*, et al.*, "3D Human Face Recognition Using Point Signature," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 233-238.

[26] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.

[27] C. Xu*, et al.*, "Automatic 3D face recognition combining global geometric

features with local shape variation information," in *Proceedings of International Conference on Automated Face and Gesture Recognition*, 2004, pp. 308-313.

[28] Y. Shan, *et al.*, "Shapeme histogram projection and matching for partial object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, 2006.

[29] Y. Huang, *et al.*, "Discriminating 3D Faces by Statistics of Depth Differences," in *Proceedings of Asian Conference on Computer Vision*, 2007, pp. 690-699.

[30] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neurosicence,* vol. 3, pp. 71-86, 1991.

[31] V. Blanz and T. Vetter, "Face Recognition Based on Fitting a 3D Morphable Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, pp. 1063-1074, 2003.

[32] K. I. Chang, *et al.*, "Face Recognition Using 2D and 3D Facial Data," in *ACM Workshop on Multimodal User Authentication*, 2003, pp. 25-32.

[33] T. Russ, *et al.*, "3D Face Recognition Using 3D Alignment for PCA," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1391-1398.

[34] X. Li and F. Da, "Robust 3D Face Recognition Based on Rejection and Adaptive Region Selection," in *ACCV*, 2009, pp. 581-590.

[35] B. K. P. Horn, "Extended Gaussian Images," *Proceedings of the IEEE,* vol. 72, 1984.

[36] H. T. Tanaka and M. Ikeda, "Curvature-Based Face Surface Recognition Using Spherical Correlation Principal Directions for Curved Object Recognition," in *Third International Conference on Automated Face and Gesture Recognition*, 1998, pp. 372–377.

[37] N. I. Fisher, *et al.*, *Statistical analysis of spherical data*: The Cambridge University Press, 1987.

[38] K. W. Bowyer, *et al.*, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition," *Computer Vision and Image Understanding,* vol. 101, pp. 1-15, 2006.

[39] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, pp. 239-256, 1992.

[40] K. Arun, *et al.*, "Least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-9, pp. 698-700, 1987.

[41] B. K. P. Horn, *et al.*, "closed form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America A*, vol. 5, pp. 1127-1137, 1988.

[42] M. W. Walker and L. Shao, "Estimating 3-D Location Parameters Using Dual Number Quaternions," *CVGIP: Image Understanding*, vol. 54, pp. 358-367, 1991.

[43] D. W. Eggert, *et al.*, "Estimating 3-D rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications - Special issue on performance evaluation*, vol. 9, pp. 272-290, 1997.

[44] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," *International Journal of Computer Vision*, vol. 13, pp. 119-152, 1994.

[45] G. Medioni and R. Waupotitsch, "Face Modeling and Recognition in 3-D," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003, pp. 232-233.

[46] X. Lu, *et al.*, "Matching 2.5D Scans for Face Recognition," in *IEEE International Conference on Pattern Recognition*, 2004, pp. 30-36.

[47] K. I. Chang, *et al.*, "Adaptive Rigid Multi-region Selection for Handling Expression Variation in 3D," in *IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition, 2005.

[48] B. B. Amor, *et al.*, "New Experiments on ICP-Based 3D Face Recognition and Authentication," in *IEEE International Conference on Pattern Recognition*, 2006.

[49] C. Tong, *et al.*, "3D Face Recognition Based on Fast Feature Detection and Non-rigid Iterative Closest Point," in *IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2009.

[50] H. Zha, *et al.*, "Registration of Range Images with Different Scanning Resolutions," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2000, pp. 1495-1500.

[51] T. Zinßer, *et al.*, "Point Set Registration with Integrated Scale Estimation," in *International Conference on Pattern Recognition and Image Processing*, 2005, pp. 116-119.

[52] K. H. Ko, *et al.*, "Algorithms for Optimal Partial Matching of Free-Form Objects with Scaling Effects," *Graphical Models*, vol. 67, pp. 120-148, 2005.

[53] S. Du, *et al.*, "ICP with Bounded Scale for Registration of M-D Point Sets," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 1291-1294.

[54] S. Du, *et al.*, "Scaling iterative closest point algorithm for registration of m-D points," *Journal of Visual Communication and Image Representation,* vol. 21, pp. 442-452, 2010.

[55] B. Achermann, *et al.*, "Face Recognition Using Range Images," in *International Conference on Virtual Systems and MultiMedia*, 1997, pp. 129-136.

[56] Y.-h. Lee and J.-c. Shim, "Curvature Based Human Face Recognition Using Depth Weighted Hausdorff Distance," in *International Conference on Image Processing*, 2004, pp. 1429 - 1432.

[57] T. D. Russ, *et al.*, "A 2D Range Hausdorff Approach for 3D Face Recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, p. 169.

[58] C. Beumier and M. Acheroy, "Face Verification from 3D and Grey Level Clues," *Pattern Recognition Letters,* vol. 22, pp. 1321-1329, 2001.

[59] A. M. Bronstein, *et al.*, "Expression-Invariant 3D Face Recognition," in *Proceedings of International Conference on Audio- and Videobased Biometric Person Authentication*, 2003, pp. 62-70.

[60] F. Tsalakanidou, *et al.*, "Use of depth and colour eigenfaces for face recognition," *Pattern Recognition Letters*, vol. 24, pp. 1427-1435, 2003.

[61] A. Godil, *et al.*, "Face recognition using 3D facial shape and color map information: comparison and combination," in *Proceedings of the SPIE - The International Society for Optical Engineering*, 2004.

[62] T. Papatheodorou and D. Rueckert, "Evaluation of Automatic 4D Face Recognition Using Surface and Texture Registration," in *Proceedings of International Conference on Automated Face and Gesture Recognition,* 2004.

[63] A. S. Mian, *et al.*, "An Efficient Multimodal 2D-3D Hybrid Approach to Automatic Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, pp. 1927-1943, 2007.

[64] T.-W. R. Lo and J. P. Siebert, "Local feature extraction and matching on range images: 2.5D SIFT," *Computer Vision and Image Understanding,* vol. 113, pp. 1235–1250, 2009.

[65] S. Zhang, "Recent progress on real-time 3d shape measurement using digital fringe projection techniques," *Optics and Lasers in Engineering,* vol. 48, pp. 149-158, 2010.

[66] X. Su, *et al.*, "Automated phase measuring profilometry using defocused projection of a Ronchi grating," *Optics Communications,* vol. 94, pp. 561-573, 1994.

[67] P. S. Huang, *et al.*, "Trapezoidal phase-shifting method for the

three-dimensional shape measurement," *Optical Engineering (Bellingham),* vol. 44, 2005.

[68] P. Jia, *et al.*, "Two-step triangularpattern phase-shifting method for three-dimensional object-shape measurement," *Optical Engineering (Bellingham),* vol. 46, 2007.

[69] P. Jia, *et al.*, "Multiple-step triangular-pattern phase-shifting and the influence of number of steps and pitch on measurement accuracy," *Applied Optics,* vol. 46, pp. 3253-3262, 2007.

[70] Q. Fang, "Linearly coded profilometry with a coding light that has icosceles triangle teeth: Wave-number-sample coding method," *Applied Optics,* vol. 36, pp. 1615-1620, 1997.

[71] H.-N. Yen, *et al.*, "Full-Field 3-D Flip-Chip Solder Bumps Measurement Using DLP-Based Phase Shifting Technique," *IEEE Transactions on Advanced Packaging,* vol. 31, pp. 830-840, 2008.

[72] Y. Gong and S. Zhang, "Ultrafast 3-D shape measurement with an off-the-shelf DLP projector," *Optics Express,* vol. 18, pp. 19743-19754, 2010.

[73] P. S. Huang, *et al.*, "Double Three-Step Phase-Shifting Algorithm," *Applied Optics,* vol. 41, pp. 4503-4509, 2002.

[74] S. Zhang and S.-T. Yau, "Generic nonsinusoidal phase error correction for three-dimensional shape measurement using a digital video projector," *Applied Optics*, vol. 46, pp. 36-43, 2007.

[75] V. Srinivasan, *et al.*, "Automated phase-measuring profilometry of 3-d diuse objects," *Applied Optics*, vol. 23, pp. 3105-3108, 1984.

[76] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 433-449, 1999.

[77] J. Feldmar and N. Ayache, "Affine and Locally Affine Registration of Free-Form Surfaces," *Technical Report of INRIA*, 1994.

[78] I. Stamos and M. Leordeanu, "Automated Feature-Based Range Registration of Urban Scenes of Large Scale," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, pp. 555-561.

[79] J.-P. Tarel and H. Civi, "Pose Estimation of Free-Form3D Objects without Point Matching Using Algebraic Surface Models," in *IEEE Workshop on Model-Based 3D*, 1998, pp. 13-21.

[80] R. Jain, *et al.*, *Machine Vision*: MIT Press, 1995.

[81] G. Wahba, "A Least Squares Estimate of Satellite Attitude," *SIAM Review*, vol.

7, p. 409, 1965.

[82] J. E. Keat, "Analysis of Least-Squares Attitude Determination Routine DOAOP," *Computer Science Corperation Technical Report*, 1977.

[83] Y. Chen and G. Medioni, "Object Modeling by Reg strat ion of Multiple Range Images," in *ICRA*, 1991, pp. 2724 - 2729.

[84] F. P. Preparata and M. I. Shamos, *Compuatational Geometry: An Introduction*: Springer, 1985.

APPENDIX

Dual Quaternions Method

Quaternions are four-element vectors, the first three elements can be treated as a 3×1 vector, and the fourth element is a scalar component. A quaternion $\breve{\boldsymbol{q}}$ can be described like this:

$$\breve{\boldsymbol{q}} = \begin{bmatrix} \mathrm{q}_1 \\ \mathrm{q}_2 \\ \mathrm{q}_3 \\ \mathrm{q}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathrm{q}_4 \end{bmatrix}, \tag{0.1}$$

where $\mathbf{q} = [q_1, q_2, q_3]^T$is a 3D vector which is equal to the values of the original coordinates, where T indicates the transpose of matrix, a scalar q_4 which is set to zero in this thesis based on the experiment. To compute the rotation and translation, another interpretation of quaternion is given:

$$\breve{\boldsymbol{q}} = \begin{bmatrix} \sin(\theta/2)\mathbf{n} \\ \cos(\theta/2) \end{bmatrix}, \tag{0.2}$$

where θ is a scalar of rotation angle, $\breve{\boldsymbol{q}}$ is a quaternion, $\mathbf{n} = [n_1, n_2, n_3]^T$ is a unit direction vector which describes the direction by using quaternion representation. This interpretation is called Euler Symmetric Parameters[42].

The dual number was first considered by the German geometer E. study (1862-1930) in the beginning of 20th century.[42] A dual angle was presented by using the idea of dual number in his research. The dual angle was defined as:

$$\hat{\theta} = \theta + \varepsilon d, \tag{0.3}$$

Where $\hat{\theta}$ is a dual angle which represented the position between two lines in 3D space, d is the distance between two lines in 3D space, θ is the angle between the two lines. ε is a parameter which indicates that it is the dual part in the dual angle expression. A dual angle has the property:

$$\sin(\hat{\theta}) = \sin(\theta + \varepsilon d) = \sin(\theta) + \varepsilon d \cos(\theta), \tag{0.4}$$

$$\cos(\hat{\theta}) = \cos(\theta + \varepsilon d) = \cos(\theta) - \varepsilon d \sin(\theta). \tag{0.5}$$

Compared with the dual angle, a dual number can be described like:

$$\hat{a} = a + \varepsilon b, \tag{0.6}$$

where a and b are two real numbers, ε is a parameter which follows the multiplication rule $\varepsilon^2 = 0.$ a is the real part of the dual number $\hat{a}$ and b is the dual part of $\hat{a}$.

Dual numbers have their own properties of addition, subtraction and multiplication:

$$(a + \varepsilon b) \pm (c + \varepsilon d) = (a + c) \pm \varepsilon(b + d), \tag{0.7}$$

$$(a + \varepsilon b)(c + \varepsilon d) = ac + \varepsilon(ad + bc). \tag{0.8}$$

The product of a dual number $\hat{a}$ and its conjugate $\overline{\hat{a}}$ is:

$$\hat{a}\overline{\hat{a}} = a^2. \tag{0.9}$$

The modulus of a dual number is:

$$|\hat{\mathrm{a}}| = \mathrm{a}. \tag{0.10}$$

The idea of dual numbers can be extended to dual vectors, dual quaternions, etc.[42] Compared with a dual number, a dual number quaternion $\widehat{\boldsymbol{q}}$ is described as follows:

$$\widehat{\boldsymbol{q}} = \breve{\boldsymbol{q}} + \varepsilon\breve{\boldsymbol{s}}, \tag{0.11}$$

where $\breve{\boldsymbol{q}}$ and $\breve{\boldsymbol{s}}$ are two real quaternions. $\breve{\boldsymbol{q}}$ is the real part of the dual quaternion $\widehat{\boldsymbol{q}}$, $\breve{\boldsymbol{s}}$ is the dual part of $\widehat{\boldsymbol{q}}$. A dual number quaternion has the same interpretation with quaternions:

$$\widehat{\boldsymbol{q}} = \begin{bmatrix} \widehat{\mathrm{q}_1} \\ \widehat{\mathrm{q}_2} \\ \widehat{\mathrm{q}_3} \\ \widehat{\mathrm{q}_4} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{q}} \\ \widehat{\mathrm{q}_4} \end{bmatrix}, \tag{0.12}$$

where $\widehat{\mathbf{q}}$ is a dual 3D vector which equals to $[\widehat{\mathrm{q}_1}, \widehat{\mathrm{q}_2}, \widehat{\mathrm{q}_3}]^{\boldsymbol{T}}$. The only difference between a quaternion and a dual quaternion is that every element in the above equation is a dual number. A dual quaternion consists of eight elements to represent the 3D motion for one object. According to equation (0.1), the dual quaternion has the similar relationship with the dual angle $\hat{\theta}$ and dual vector $\widehat{\mathbf{n}}$:

$$\widehat{\boldsymbol{q}} = \begin{bmatrix} \sin(\hat{\theta}/2)\,\widehat{\mathbf{n}} \\ \cos(\hat{\theta}/2) \end{bmatrix}, \tag{0.13}$$

where $\hat{\mathbf{n}}$ is the dual vector contains the information about the rotation and translation for an object rotated and translated in a 3D space. $\hat{\theta}$ is a dual angle stores the rotation and translation angle.

A dual vector $\hat{\mathbf{n}}$ can be represented as:

$$\hat{\mathbf{n}} = \boldsymbol{n} + \varepsilon \boldsymbol{p} \times \boldsymbol{n}. \tag{0.14}$$

The dual angle $\hat{\theta}$ can be represented as:

$$\hat{\theta} = \theta + \varepsilon d, \tag{0.15}$$

where $\boldsymbol{n}$ is a unit vector which is the direction to rotate and translate, θ is the rotation angle and d is the distance of translation along the unit vector $\boldsymbol{n}$ passing by point $\boldsymbol{p}$. Figure 0.1 illustrates the rotation and translation of the dual number quaternion.

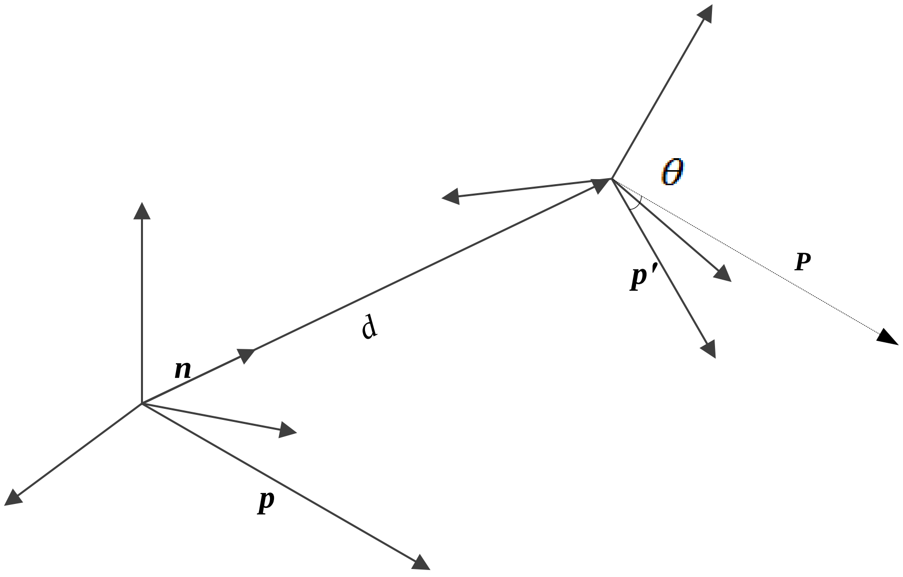


Figure 0.1: the rotation and translation for point $\boldsymbol{p}$.

We can place equation (0.14) and (0.15) into equation (0.11), by using the properties of the dual number, we can get:

$$\sin\left(\frac{\hat{\theta}}{2}\right)\hat{\mathbf{n}} = \sin\left(\frac{\theta}{2}+\varepsilon\frac{d}{2}\right)(\boldsymbol{n}+\varepsilon\boldsymbol{p}\times\boldsymbol{n})$$

$$= \mathbf{n}\sin\left(\frac{\theta}{2}\right)+\varepsilon\frac{d}{2}\sin\left(\frac{\theta}{2}\right)\boldsymbol{n}+\varepsilon\sin\left(\frac{\theta}{2}\right)(\boldsymbol{p}\times\boldsymbol{n}). \qquad (0.16)$$

$$\cos(\hat{\theta}/2) = \cos(\frac{\theta}{2}+\varepsilon\frac{d}{2}) = \cos\left(\frac{\theta}{2}\right)+\varepsilon\frac{d}{2}\cos\left(\frac{\theta}{2}\right). \qquad (0.17)$$

Quaternion $\breve{\boldsymbol{q}}$ in equation (0.11) becomes

$$\breve{\boldsymbol{q}} = \begin{bmatrix}\sin\left(\frac{\theta}{2}\right)\boldsymbol{n}\\ \cos\left(\frac{\theta}{2}\right)\end{bmatrix}. \qquad (0.18)$$

The dual part quaternion $\breve{\boldsymbol{s}}$ in equation (0.11) becomes

$$\breve{\boldsymbol{s}} = \begin{bmatrix}\frac{d}{2}\sin\left(\frac{\theta}{2}\right)\boldsymbol{n}+\sin\left(\frac{\theta}{2}\right)(\boldsymbol{p}\times\boldsymbol{n})\\ \frac{d}{2}\cos\left(\frac{\theta}{2}\right)\end{bmatrix}. \qquad (0.19)$$

From equation (0.18) and (0.19), we can get the constraints:

$$\breve{\boldsymbol{q}}^{\boldsymbol{T}}\breve{\boldsymbol{q}} = 1, \qquad (0.20)$$

$$\breve{\boldsymbol{s}}^{\boldsymbol{T}}\breve{\boldsymbol{q}} = 0. \qquad (0.21)$$

Two important 4×4 matrix in quaternions are given:

$$\mathbf{W}(\breve{\boldsymbol{q}}) = \begin{bmatrix} q_4\boldsymbol{I}-\boldsymbol{k}(\mathbf{q}) & \boldsymbol{q}\\ -\boldsymbol{q}^{\boldsymbol{T}} & q_4\end{bmatrix}, \qquad (0.22)$$

$$\mathbf{Q}(\breve{\boldsymbol{q}}) = \begin{bmatrix} q_4\boldsymbol{I} + \boldsymbol{k}(\mathbf{q}) & \boldsymbol{q} \\ -\boldsymbol{q}^{\boldsymbol{T}} & q_4 \end{bmatrix}. \tag{0.23}$$

A rotation matrix $\boldsymbol{R}$ which rotated by an angle θ and along the direction of unit vector $\boldsymbol{n} = \left[n_x, n_y, n_z\right]$ can be represented as a 3×3 matrix:

$$\boldsymbol{R} =$$

$$\begin{bmatrix} \cos\theta + {n_x}^2(1-\cos\theta) & n_x n_y(1-\cos\theta) - n_z\sin\theta & n_x n_z(1-\cos\theta) - n_y\sin\theta \\ n_y n_x(1-\cos\theta) - n_z\sin\theta & \cos\theta + {n_y}^2(1-\cos\theta) & n_y n_z(1-\cos\theta) - n_x\sin\theta \\ n_z n_x(1-\cos\theta) - n_y\sin\theta & n_z n_y(1-\cos\theta) - n_x\sin\theta & \cos\theta + {n_z}^2(1-\cos\theta) \end{bmatrix}.$$

$$\tag{0.24}$$

This matrix can be also written as:

$$\boldsymbol{R} = [\boldsymbol{I}\cos\theta + (1-\cos\theta)\boldsymbol{n}\otimes\boldsymbol{n} + (\sin\theta)\boldsymbol{n}\times\boldsymbol{n}], \tag{0.25}$$

where $\boldsymbol{I}$ is the 3×3 identity matrix, $\boldsymbol{n}\otimes\boldsymbol{n}$ is the tensor product:

$$\boldsymbol{n}\otimes\boldsymbol{n} = \begin{bmatrix} {n_x}^2 & n_x n_y & n_x n_z \\ n_x n_y & {n_y}^2 & n_y n_z \\ n_x n_z & n_y n_z & {n_z}^2 \end{bmatrix}. \tag{0.26}$$

The cross product $\boldsymbol{n}\times\boldsymbol{n}$ is:

$$\boldsymbol{n}\times\boldsymbol{n} = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}. \tag{0.27}$$

From equation (0.25) and combine equation (0.1) and (0.2), we could get:

$$\boldsymbol{I}\cos\theta = (\cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right))\boldsymbol{I} = ({q_4}^2 - \boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{q})\boldsymbol{I}, \tag{0.28}$$

$$(1-\cos\theta)\boldsymbol{n}\otimes\boldsymbol{n} = (2\sin^2(\tfrac{\theta}{2}))\boldsymbol{n}\otimes\boldsymbol{n} = 2\boldsymbol{q}\boldsymbol{q}^{\boldsymbol{T}}, \tag{0.29}$$

$$(\sin\theta)\boldsymbol{n}\times\boldsymbol{n} = \left(2\sin\frac{\theta}{2}\cos\frac{\theta}{2}\right)\boldsymbol{n}\times\boldsymbol{n} = 2q_4|\boldsymbol{q}|(\boldsymbol{n}\times\boldsymbol{n}), \tag{0.30}$$

where $|\boldsymbol{q}|$ is the modulus of vector $\boldsymbol{q}$. Let′s define a skew- symmetric matrix:

$$\boldsymbol{K}(\boldsymbol{q}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}. \tag{0.31}$$

The new rotation matrix can be written as:

$$\boldsymbol{R} = \ (q_4^2 - \boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{q})\boldsymbol{I} + 2\boldsymbol{q}\boldsymbol{q}^{\boldsymbol{T}} + 2q_4\boldsymbol{K}(\boldsymbol{q}), \tag{0.32}$$

or

$$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0}^{\boldsymbol{T}} & 1 \end{bmatrix} = \mathbf{W}(\breve{\boldsymbol{q}})^{\boldsymbol{T}}\mathbf{Q}(\breve{\boldsymbol{q}}). \tag{0.33}$$

As Figure 0.1 illustrates, an object rotated an angle of θ, translated the distance d in the direction of the unit vector $\boldsymbol{n}$ and passed by a point p. the translator vector $\boldsymbol{t}$ is:

$$\boldsymbol{t} = \boldsymbol{p} + d\boldsymbol{n} - \boldsymbol{p}' = \boldsymbol{p} + d\boldsymbol{n} - \boldsymbol{R}\boldsymbol{p} = (\boldsymbol{I} - \boldsymbol{R})\boldsymbol{p} + d\boldsymbol{n}. \tag{0.34}$$

In equation (0.25), we have the representation of rotation matrix $\boldsymbol{R}$, in the last part of equation (0.25), we have $\boldsymbol{n}\times\boldsymbol{n} = \boldsymbol{k}(\boldsymbol{n})$, $\boldsymbol{n}\otimes\boldsymbol{n}$ is equal to $\boldsymbol{I} + \boldsymbol{k}(\boldsymbol{n})\boldsymbol{k}(\boldsymbol{n})$, and equation (0.25) can be written like:

$$\begin{aligned} \mathbf{R} &= [\boldsymbol{I}\cos\theta + (1-\cos\theta)\boldsymbol{n}\otimes\boldsymbol{n} + (\sin\theta)\boldsymbol{n}\times\boldsymbol{n}] \\ &= \mathbf{I} + 2\sin^2(\tfrac{\theta}{2})\,\boldsymbol{k}(\boldsymbol{n})\boldsymbol{k}(\boldsymbol{n}) + \sin\theta\,\boldsymbol{k}(\boldsymbol{n}). \end{aligned} \tag{0.35}$$

Then we replace $\boldsymbol{R}$ in equation (0.34) by using (0.35), we have:

$$\boldsymbol{t} = (\boldsymbol{I} - \boldsymbol{R})\boldsymbol{p} + d\boldsymbol{n} = -2\sin^2\left(\frac{\theta}{2}\right)\boldsymbol{k}(\boldsymbol{n})\boldsymbol{k}(\boldsymbol{n})\boldsymbol{p} - \sin\theta\,\boldsymbol{k}(\boldsymbol{n})\boldsymbol{p} + d\boldsymbol{n}$$

$$= 2\sin^2\left(\frac{\theta}{2}\right)\boldsymbol{n} \times (\boldsymbol{p} \times \boldsymbol{n}) + \sin\theta\,(\boldsymbol{p} \times \boldsymbol{n}) + d\boldsymbol{n}. \tag{0.36}$$

By using equation (0.18) and (0.19), we have

$$\sin^2\left(\frac{\theta}{2}\right)\boldsymbol{n} \times (\boldsymbol{p} \times \boldsymbol{n}) = \check{\boldsymbol{q}} \times \check{\boldsymbol{s}}, \tag{0.37}$$

$$\sin\theta\,(\boldsymbol{p} \times \boldsymbol{n}) + d\boldsymbol{n} = 2\left(\frac{1}{2}\sin\theta\,(\boldsymbol{p} \times \boldsymbol{n})\right) + d\boldsymbol{n} = 2\left(\sin\frac{\theta}{2}\cos\frac{\theta}{2}\,(\boldsymbol{p} \times \boldsymbol{n})\right) + d\boldsymbol{n}$$

$$= 2(\mathrm{q}_4\mathbf{s} - \mathrm{s}_4\mathbf{q}). \tag{0.38}$$

Therefore, place equation (0.37) and (0.38) into the translation vector $\boldsymbol{t}$ equation (0.36), it can be written as:

$$\boldsymbol{t} = 2(\check{\boldsymbol{q}} \times \check{\boldsymbol{s}} + \mathrm{q}_4\mathbf{s} - \mathrm{s}_4\boldsymbol{q}). \tag{0.39}$$

We can find that

$$\mathbf{W}(\check{\boldsymbol{q}})^{\boldsymbol{T}}\check{\boldsymbol{s}} = \begin{bmatrix} q_4\boldsymbol{I} - \boldsymbol{k}(\boldsymbol{q}) & -\boldsymbol{q} \\ \boldsymbol{q}^{\boldsymbol{T}} & q_4 \end{bmatrix}\begin{bmatrix} \mathbf{s} \\ s_4 \end{bmatrix} = \begin{bmatrix} \left(q_4\boldsymbol{I} - \boldsymbol{k}(\boldsymbol{q})\right)\mathbf{s} - s_4\boldsymbol{q} \\ 0 \end{bmatrix}. \tag{0.40}$$

Place equation (0.39) into (0.40), we can get

$$\check{\boldsymbol{t}} = \begin{bmatrix} \frac{1}{2}\boldsymbol{t} \\ 0 \end{bmatrix} = \mathbf{W}(\check{\boldsymbol{q}})^{\boldsymbol{T}}\check{\boldsymbol{s}}. \tag{0.41}$$

After we know the representations of rotation and translation in dual quaternions, we can determine the rotation and translation of one object in 3D space. Let $\boldsymbol{x_i}$ is the coordinate of one object in the 3D space, $\boldsymbol{y_i}$ is the coordinate of the object after rotating by a rotation matrix $\boldsymbol{R}$ and translating by a translation matrix $\boldsymbol{t}$ in the same coordinate system. We have:

$$\boldsymbol{y_i} = \boldsymbol{t} + \boldsymbol{R}\boldsymbol{x_i}. \tag{0.42}$$

If we use quaternions to represent the above equation, we have:

$$\check{\boldsymbol{y}}_{\boldsymbol{\iota}} = \mathbf{W}(\check{\boldsymbol{q}})^{\boldsymbol{T}}\check{\boldsymbol{s}} + \mathbf{W}(\check{\boldsymbol{q}})^{\boldsymbol{T}}\mathbf{Q}(\check{\boldsymbol{q}})\check{\boldsymbol{x}}_{\boldsymbol{\iota}}. \tag{0.43}$$

Meanwhile, the direction quaternion of the object is:

$$\check{\boldsymbol{n}}_{\boldsymbol{\iota}} = \mathbf{W}(\check{\boldsymbol{q}})^{\boldsymbol{T}}\mathbf{Q}(\check{\boldsymbol{q}})\check{\boldsymbol{n}}_{\boldsymbol{\iota_0}}. \tag{0.44}$$

To compute the rotation and translation matrix, we need to minimize the error function extracted from equation (0.43) and (0.44**)**, which means we use $\check{\boldsymbol{q}}$ and $\check{\boldsymbol{s}}$ to determine the minimum error $\mathbf{E}$:

$$\mathbf{E} = \sum_{i=1}^{l}(\check{\boldsymbol{y}}_{\boldsymbol{\iota_0}} - \check{\boldsymbol{y}}_{\boldsymbol{\iota}})^2 + \sum_{i=1}^{k}(\check{\boldsymbol{n}}_{\boldsymbol{\iota_0}} - \check{\boldsymbol{n}}_{\boldsymbol{\iota}})^2, \tag{0.45}$$

where k and l are the number of points contains in the object and the direction quaternion, $\check{\boldsymbol{y}}_{\boldsymbol{\iota_0}}$ and $\check{\boldsymbol{n}}_{\boldsymbol{\iota_0}}$ are the computed quaternions, which mean these two quaternions are the theoretical values of the motions after the object applying the rotation matrix $\boldsymbol{R}$ and translation matrix $\boldsymbol{t}$. Here we have:

$$\left(\check{\boldsymbol{y}}_{\boldsymbol{\iota_0}} - \check{\boldsymbol{y}}_{\boldsymbol{\iota}}\right)^2 = \check{\boldsymbol{s}}^{\boldsymbol{T}}\check{\boldsymbol{s}} + 2\check{\boldsymbol{s}}^{\boldsymbol{T}}\left(\boldsymbol{W}(\check{\boldsymbol{x}}_{\boldsymbol{\iota}})) - \boldsymbol{Q}\left(\check{\boldsymbol{y}}_{\boldsymbol{\iota_0}}\right)\right)\check{q} - 2\check{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{Q}\left(\check{\boldsymbol{y}}_{\boldsymbol{\iota_0}}\right)^{\boldsymbol{T}}\boldsymbol{W}(\check{\boldsymbol{x}}_{\boldsymbol{\iota}})\check{\boldsymbol{q}} + \left(\check{\boldsymbol{x}}_{\boldsymbol{\iota}}{}^{\boldsymbol{T}}\check{\boldsymbol{x}}_{\boldsymbol{\iota}} + \check{\boldsymbol{y}}_{\boldsymbol{\iota_0}}{}^{\boldsymbol{T}}\check{\boldsymbol{y}}_{\boldsymbol{\iota_0}}\right),$$

$$(0.46)$$

$$(\breve{\boldsymbol{n}_{\iota_0}} - \breve{\boldsymbol{n}_{\iota}})^2 = 2(1 - \breve{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{Q}(\breve{\boldsymbol{y}_{\iota_0}})^{\boldsymbol{T}}\boldsymbol{W}(\breve{\boldsymbol{x}_{\iota}})\breve{\boldsymbol{q}}). \qquad (0.47)$$

Rewrite the error function by using the above two equations, we can get:

$$\mathbf{E} = \breve{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{C_1}\breve{\boldsymbol{q}} + \breve{\boldsymbol{s}}^{\boldsymbol{T}}\boldsymbol{C_2}\breve{\boldsymbol{s}} + \breve{\boldsymbol{s}}^{\boldsymbol{T}}\boldsymbol{C_3}\breve{\boldsymbol{q}} + \text{constant}, \qquad (0.48)$$

where

$$\boldsymbol{C_1} = -2\sum_{i=1}^{k}\boldsymbol{Q}(\breve{\boldsymbol{y}_{\iota_0}})^{\boldsymbol{T}}\boldsymbol{W}(\breve{\boldsymbol{x}_{\iota}}) - 2\sum_{i=1}^{l}\boldsymbol{Q}(\breve{\boldsymbol{n}_{\iota_0}})^{\boldsymbol{T}}\boldsymbol{W}(\breve{\boldsymbol{n}_{\iota}}), \qquad (0.49)$$

$$\boldsymbol{C_2} = l\boldsymbol{I}, \qquad (0.50)$$

$$\boldsymbol{C_3} = 2\sum_{i=1}^{l}(\boldsymbol{W}(\breve{\boldsymbol{x}_{\iota}}) - \boldsymbol{Q}(\breve{\boldsymbol{y}_{\iota_0}})), \qquad (0.51)$$

$$\text{constant} = 2k + \sum_{i=1}^{l}(\breve{\boldsymbol{x}_{\iota}}^{\boldsymbol{T}}\breve{\boldsymbol{x}_{\iota}} + \breve{\boldsymbol{y}_{\iota_0}}^{\boldsymbol{T}}\breve{\boldsymbol{y}_{\iota_0}}). \qquad (0.52)$$

Considered the constraints of $\breve{\boldsymbol{q}}$ and $\breve{\boldsymbol{s}}$. The error function can be rewritten:

$$\mathbf{E} = \breve{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{C_1}\breve{\boldsymbol{q}} + \breve{\boldsymbol{s}}^{\boldsymbol{T}}\boldsymbol{C_2}\breve{\boldsymbol{s}} + \breve{\boldsymbol{s}}^{\boldsymbol{T}}\boldsymbol{C_3}\breve{\boldsymbol{q}} + \text{constant} + \lambda_1(\breve{\boldsymbol{q}}^{\boldsymbol{T}}\breve{\boldsymbol{q}} - 1) + \lambda_2(\breve{\boldsymbol{s}}^{\boldsymbol{T}}\breve{\boldsymbol{q}}), \quad (0.53)$$

where λ_1 and λ_2 are Lagrange multipliers. In order to get the minimum value from the error function, we can take the partial derivatives:

$$\frac{\partial \mathbf{E}}{\partial \breve{q}} = (\boldsymbol{C_1} + \boldsymbol{C_1}^{\boldsymbol{T}})\breve{\boldsymbol{q}} + \boldsymbol{C_3}^{\boldsymbol{T}}\breve{\boldsymbol{s}} + 2\lambda_1\breve{\boldsymbol{q}} + \lambda_2\breve{\boldsymbol{s}} = 0, \qquad (0.54)$$

$$\frac{\partial \mathbf{E}}{\partial \breve{s}} = (\boldsymbol{C_2} + \boldsymbol{C_2}^{\boldsymbol{T}})\breve{\boldsymbol{s}} + \boldsymbol{C_3}\breve{\boldsymbol{q}} + \lambda_2\breve{\boldsymbol{q}} = 0. \qquad (0.55)$$

From equation (0.55), we can get:

$$\breve{\boldsymbol{s}} = -\left(\boldsymbol{C_2} + \boldsymbol{C_2}^{\boldsymbol{T}}\right)^{-1}(\boldsymbol{C_3}\breve{\boldsymbol{q}} + \lambda_2\breve{\boldsymbol{q}}). \tag{0.56}$$

To solve λ_2, we can multiply $\breve{\boldsymbol{q}}$ in equation (0.55) by considering the constraints of quaternions:

$$\lambda_2 = -\breve{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{C}_3\breve{\boldsymbol{q}}. \tag{0.57}$$

Because $\boldsymbol{C_3}$ is a skew matrix, $\lambda_2 = 0$,

$$\breve{\boldsymbol{s}} = -\left(\boldsymbol{C_2} + \boldsymbol{C_2}^{\boldsymbol{T}}\right)^{-1}\boldsymbol{C_3}\breve{\boldsymbol{q}}. \tag{0.58}$$

Equation (0.54) becomes:

$$\boldsymbol{A}\breve{\boldsymbol{q}} = \lambda_1\breve{\boldsymbol{q}}, \tag{0.59}$$

where $\boldsymbol{A} = \frac{1}{2}(\boldsymbol{C_3}^{\boldsymbol{T}}\left(\boldsymbol{C_2} + \boldsymbol{C_2}^{\boldsymbol{T}}\right)^{-1}\boldsymbol{C_3} - \boldsymbol{C_1} - \boldsymbol{C_1}^{\boldsymbol{T}})$.

Thus $\breve{\boldsymbol{q}}$ is an eigenvector of matrix $\boldsymbol{A}$ and λ_1 is the eigenvalue corresponding to the eigenvector $\breve{\boldsymbol{q}}$. But matrix $\boldsymbol{A}$ has four eigenvectors and we need to decide one of them in order to let the error function has the optimal result. We go back to equation (0.54) multiply by $\breve{\boldsymbol{q}}^{\boldsymbol{T}}$:

$$\frac{1}{2}\breve{\boldsymbol{q}}^{\boldsymbol{T}}\left(\left(\boldsymbol{C_1} + \boldsymbol{C_1}^{\boldsymbol{T}}\right)\right)\breve{\boldsymbol{q}} = -\frac{1}{2}\breve{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{C}_3\breve{\boldsymbol{r}} - \lambda_1 = \breve{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{C_1}\breve{\boldsymbol{q}}. \tag{0.60}$$

Multiply $\breve{\boldsymbol{s}}^{\boldsymbol{T}}$ to equation (0.55):

$$\tfrac{1}{2}\breve{\boldsymbol{s}}^{\boldsymbol{T}}\left(\left(\boldsymbol{C}_2+\boldsymbol{C}_2{}^{\boldsymbol{T}}\right)\right)\breve{\boldsymbol{s}} = -\tfrac{1}{2}\breve{\boldsymbol{s}}^{\boldsymbol{T}}\boldsymbol{C}_3\breve{\boldsymbol{q}} = \breve{\boldsymbol{s}}^{\boldsymbol{T}}\boldsymbol{C}_2\breve{\boldsymbol{s}}. \tag{0.61}$$

Replace $\breve{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{C}_{\boldsymbol{1}}\breve{\boldsymbol{q}}$ and $\breve{\boldsymbol{s}}^{\boldsymbol{T}}\boldsymbol{C}_{\boldsymbol{2}}\breve{\boldsymbol{s}}$ in the error function, we can get:

$$\mathbf{E} = \text{constant} - \lambda_1. \tag{0.62}$$

In order to minimize the error, we need to keep λ_1 has the largest value. Because λ_1 equals to one eigenvalue of matrix $\boldsymbol{A}$, and matrix $\boldsymbol{A}$ has four eigenvalues, here we select the largest eigenvalue equals to λ_1, and quaternion $\breve{\boldsymbol{q}}$ equals to the eigenvector which is corresponding to the largest eigenvalue.