A high performance automatic face recognition system using 3D shape information

Quan Ju

Submitted for the degree of Doctor of Philosophy

Department of Computer Science

THE UNIVERSITY of York

September 2010

Abstract

Face recognition is one of the most important applications to receive attention in the areas of Computer Vision and Pattern Recognition. However, face recognition has many challenges and difficulties, such as the requirement for high speed search in large datasets and the requirement for high match accuracy under various noise conditions. Currently, as numerous 3D face datasets become available, more and more researchers start to move their concentration to 3D face recognition. Compared with 2D face image, 3D face images contain more explicit information which is very useful for dealing with the head orientation and the facial expression problem.

In this thesis, a framework to implement automatic 3D face recognition is proposed and implemented. In the first stage, a key facial feature - the nose has to be extracted for the subsequent face recognition process. In order to exploit the local feature information, we present a face feature extraction methods based on a 3D shape descriptor. Two different 3D shape descriptor Multi Contour Surface Angle Moments Descriptor(MCSAMD) and Multi Shell Surface Angle Moments Descriptor(MSSAMD) are designed and implemented. The nose tip is identified using a binary neural network technique called k-Nearest Neighbour Correlation Matrix Memories(CMM) algorithm. The main face area is localized and cropped based on the nose tip localization with an identification rate of almost 100% on FRGC 3D face database. Secondly, a face aligned approach is implemented by applying a combination of methods including Principal Component Analysis(PCA) face correction, Iterative Closest Point algorithms(ICP) and the alignment using the symmetry of human

face. All faces are aligned to a unified coordinate system from the original pose position even under expression variations. The position of the nose tip is also further corrected. After the face alignment, the main face area is divided into several regions with different weights according to the face expression variability. Similarity measurement algorithms based on the pose-invariant 3D shape descriptor MSSAMD are used to match the corresponding regions for different faces. The expression variability weights are applied in the final consideration of face identification and verification. Experiments are performed on the FRGC database which is the largest 3D face database of 4950 faces with different expressions. In the experiments dealing with 4007 faces with different expressions, a 91.96% verification at a false acceptance rate(FAR) of 0.1% and a 97.63% rank-one identification rate are achieved.

Acknowledgments

Firstly, I would like to thank my parents for their endless love and unconditional support during my study in York.

I would like also to thank my supervisors, Dr Simon O'Keefe and Professor Jim Austin, for their helpful guidance, valuable advice and encouragement about my research. I also thank my assessor Dr Nick Pears for his support to my work.

Additionally, thanks to all the people in the Advanced Computer Architectures group for all their help, support and friendship.

Special thanks go to the people who helps me make the decision to pursue PhD study.

Declaration

I hereby declare that all the work in this thesis is solely my own, except where attributed and cited to another author. Some of the material in this thesis has been previously published by the author. A list of publications can be found on page iii.

List of Publications

1. Quan Ju, Simon O'Keefe and Jim Austin, Binary neural network based 3D facial feature localization. In *International Joint Conference on Neural Networks*, pp.1462-1469, 2009

2. Quan Ju, Simon O'Keefe. Automatic 3D facial feature localization and face alignment using binary neural network. In *Asian Conference on Computer Vision, Workshop: Community Based 3D Content And Its Applications In Mobile Internet Environments*, 2009.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In Biometrics, the science of differentiating the unique and intrinsic physical and behavioral attributes of human beings, the human face is thought to be an effective biometric indicator as well as finger/palm print, voice, iris/retina and handwriting signature. Due to their variations in properties, those biometric attributes are applied to satisfy different application requirements. Face recognition has a low requirement of intrusiveness, while others need the subjects to cooperate during the identification or verification process. Furthermore, it is relatively easier to acquire the necessary data because providing a photo seems acceptable to most people. On the contrary, for example, collecting fingerprints is always considered as an affront to a person's privacy. Since there always has been a great demand for the use of face recognition in law, security and business applications, face recognition has become more and more important in the research areas of computer vision and pattern recognition. Face recognition technology could make a great improvement to the applications that require distinguishing identities such as crowd surveillance and access control.

Among many various face recognition environments, in general, there are three face recognition scenarios: face verification, face identification and watch list [57].

Face verification:

Face verification is a face recognition operation to compare a query face image against a template face image in the database to determine whether or not the subject is someone who they claim to be. The query subject is matched only with a face image in the gallery database belonging to the identity that he/she claimed to be. The identity verification is approved only if the similarity score is above a certain threshold. The verification rate and the false acceptance rate(FAR) are two indicators to evaluate the verification performance. A good balance is required between these two rates. A Receiver Operating Characteristic(ROC) curve is plotted to show the performance by using the verification rates vs the false acceptance rates.

Face identification:

Face identification is to compare a query image with a number of images in a gallery database of known individuals to identify who this person is. The query subject is identified with the subject in the gallery dataset achieving the highest similarity matching score with the query image. The query subject is one subject in the gallery database. The percentage of queries for which the highest similarity score is a correct match is called the rank-one identification rate. The percentage of queries for which the top n similarity scores achieve a correct match is called rank n identification rate. A Cumulative Match Curve(CMC) is plotted to show the identification performance by using the different rank n vs corresponding percentages of correct identification.

Watch list:

An inquiry image is compared to all images in the gallery database(watch list) and each comparison generates a similarity score. If at any time a similarity score is greater than a threshold, an alarm is raised. The system will consider the query subject is in the database if there is an alarm. There are two indicators to show the performance of the watch list applications. One is the Detection and Identification Rate which is the percentage of queries for which an alarm is correctly raised. Another is the False Alarm Rate, which is the percentage of queries where an alarm is raised but the query subject is not in the gallery database.

1.1 3D face recognition

Over the past decades, face recognition technology has achieved many significant improvements. Many efficacious systems emerged within the recent ten years. Most of them are capable of obtaining a recognition rate of 90% or more under some controlled conditions [86]. Generally speaking, several problems are the key difficulties in face recognition. First one is: how to overcome the illumination variations? The light conditions and camera parameters both result in the variations of skin texture, which can significantly lower the performance of the face recognition. Secondly, head orientation variations also affect the results of the face recognition. Especially in 2D face images, severe head rotation will lose/occlude some parts of the face. Expression variations problem is an importance challenge in face recognition, because the appearance of the face changes when different expressions are produced. Aging factor is also

a problem, because the face varies over time, particularly after a long period. Another factor that will affect face recognition is the occlusion problem which can be caused by glasses, scarf, beard and hat.

A face recognition system is required to solve at least the problems of illumination, head orientation and expression variants in the key challenges mentioned above. There are many face recognition approaches which deal with 2D or 3D face data respectively. Some researchers also combine 2D and 3D information together to implement face recognition. As more and more 3D face collections become available, more 3D face recognition approaches appear because 3D face recognition has some advantages to deal with illumination and head orientation problems. For illumination variations, 3D shape is not affected by different lighting conditions. Thus 3D face recognition does not have illumination problems if texture/intensity information is not used. Unlike the occlusion occurred in severe 2D pose angles, there is no information loss in the different head orientations of 3D face. However, 3D face recognition (without using 2D texture data) still has some challenges left. The most challenging one is how to deal with the facial expression variations, which severely affect the face recognition process, because expressions such as laughter, anger and crying can generate very different 3D face shapes. That increases the difficulties to find the similarity between faces belonging to the same individual.

More and more 3D face data has become available in recent decades along with the rapid development of 3D data acquisition devices. Some researchers classify the data with more than two dimensional information into 2.5D and 3D representations [6]. A 2.5D face image only consists of a group of 3D points to represent the face surface, where the depth z values are stored in each pixel

in *xoy* plane. On the other hand, 3D face images cover the whole head by taking scans from different viewpoints. In this thesis, we will ignore this distinction. All 3D face images are considered as a cloud of 3D points (x,y,z). The 3D images can be considered as depth images, while the 2D images are referred to as intensity images. In 2004, Xu et al. [94] presented a comparison between intensity images and depth images in their discriminating ability of recognizing people. They concluded that the face recognition of depth images are less affected by illumination than intensity images. The results of Xu's work provide some evidence that the 3D face recognition has the advantage over 2D face recognition in dealing with illumination problems. Examples of 2D face image, depth face image and face point cloud are shown in figure 1.1.

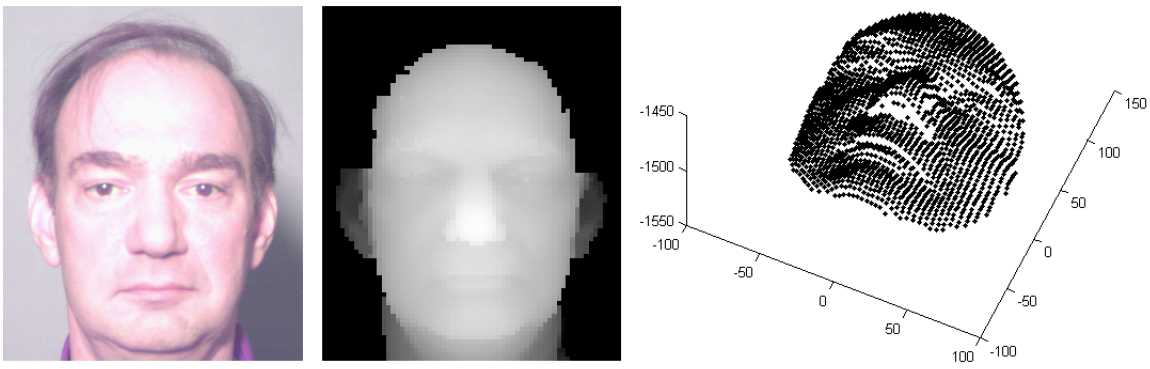


Figure 1.1: *Left to right side: examples of intensity/texture image, depth image and point cloud.*

1.2 Motivation and aims

To build a practical automatic 3D face recognition system, a concern is about the quality of face data. In most 3D face databases, the data are acquired by

some scan device, for example a laser scanner as used in FRGC database [72]. The face data are not perfectly restricted to the main face area only. Hair, clothes and other parts of the body like shoulder together with various noises lower the quality of face images. To reduce these non-face factors, we have to precisely detect where the main face area is. For the face detection, one previous approach is to use the texture information to find the face and remove the unnecessary elements [80]. However, this method has a precondition that the 2D and 3D channels must be perfectly aligned which is not 100% guaranteed in most 3D face databases. Another type of method to find the face is to localize the facial features such as nose tip, eyes and mouth by using pure 3D data [28] [16] [29] [93] [77] [49].

Another concern in 3D face recognition is how to handle the pose variations. There are many methods to solve this problem. Currently, the most feasible and widely used solution is to use face registration/alignment methods based on the variants of Iterative Closest Point algorithm(ICP) [14]. Unlike the texture information used in 2D face recognition,a pure 3D face image normally is a 3D point cloud which contains the x, y and z position information. Considering the differences in resolution, rotation and density of those points, it is inconvenient to compare two 3D point-clouds which represent two pieces of 3D surface. The information provided by those point clouds has to be converted into some other form that can be used to measure the similarity of two faces. Moreover, due to the expression variations in 3D faces, it is required that face recognition approaches have the ability to extract the common parts or factors between faces with different expressions. In summary, an integrated 3D face recognition system has three tasks:

1). Face feature extraction to localize the face.

2). Accurate face alignment.

3). Face recognition able to handle expressions.

An automatic 3D face recognition system has to achieve very high accuracy in all of these three parts. Any incorrect results in the face detection will affect the performance in face alignment and mistakes in the face alignment will also cause the inaccuracy in face recognition stage. The ultimate aim of this thesis is to implement a full automatic face recognition system including face detection, face alignment and a fast face recognition approach. And in the meantime, several issues are also concerned:

1). How reliable is the face detection based on the Face feature extraction (for example: nose detection) even under expression variations?

2). How to implement a face alignment under expression variations?

3). How to evaluate a face alignment approach?

4). How does a facial expression affect the face recognition?

5). What is the computational efficiency in face recognition.

In this thesis, firstly we review the classical face recognition algorithms and survey a number of state-of-the-art 3D face recognition techniques. Then we propose and implement an automatic 3D face recognition approach including three parts: nose tip detection/face detection, face alignment and face recognition. In the nose tip detection, we propose an accurate 3D facial feature localization approach based on 3D shape descriptors using k-Nearest Neighbour AURA (Advanced Uncertain Reasoning Architecture) algorithm to detect the nose tip with a recognition rate of 99.96. Then based on the results of the nose tip detection, the main face area is found and cropped. After that, an integrated ICP-based 3D face alignment is implemented to correct

the pose variations even under different expressions. Compared with state-of-the-art ICP-based face alignment techniques, our method achieves the best performance both in neutral faces and non-neutral faces evaluations. Using results of face detection and face alignment, we implement a high performance 3D face recognition approach which obtains a rank-one identification rate of 97.63% which the top 2 best performance achieved in the "first vs all" experiments in FRGC v2 database.

1.3 Thesis overview

The following sections are respectively describing the content of corresponding chapters in this thesis:

In chapter two, there will be a literature review of face recognition approaches. The review includes classical and state-of-the-art face recognition techniques about 2D and 3D face. The current 3D face databases are also introduced in this chapter as well as the performance evaluation methods and protocols.

In chapter three, a 3D facial feature extraction algorithm is proposed based on the 3D shape descriptor. Two 3D shape descriptors are implemented and compared. Nose tip and eye corners are localized by using a KNN-CMM algorithm.

After the localization of the nose tip, the main face area can be cropped for further tasks. In chapter four, an accurate face alignment using a combination of face alignment methods including PCA, ICP and symmetrical face aliment is implemented. All faces are aligned according to the pose of a standard face.

The position of the nose tip is further corrected along ox,oy and oz direction.

Chapter five performs a fast and efficient 3D face recognition approach based on the shape descriptor designed in the chapter three. Face regions are segmented according to the degree that they are affected by expression variations. Then a weight for each point relative to expression variability is applied during the matching between corresponding regions. Implementation of face identification and face verification are respectively proposed and performed.

In chapter six, conclusions will be drawn with further discussion of the progress achieved and the contributions of the whole 3D face recognition system and technology used in this thesis. Possible improvements and further investigation are also discussed in this chapter.

Chapter 2

Literature Review

2.1 Introduction

Face recognition is a complex system in Biometrics. Pattern recognition, machine learning, computer vision and graphics are all involved in face recognition. Bledsoe [19] began the first research of face recognition in 1964. The first automatic face recognition system was produced in 1977 by Kanade [52]. In the beginning, the majority of face recognition methods were based on a 2D face image. Face recognition in 2D utilizes the color or intensity information of 2D images. An identification rate of more than 90% was recently reported under controlled conditions [6]. However, the performance of 2D face recognition systems will decrease under changes to head orientation, illumination and expression variations. Due to having better abilities to deal with those two problems, 3D face recognition approaches have some advantages over 2D face recognition ones. In this chapter, we will present an overview of related works covered classical and state-of-the-art face recognition approaches.

2.2 2D Face recognition algorithms

The typical 2D face recognition approaches can be categorized into the appearance-based and the model-based algorithms. Appearance-based face recognition algorithms are based on representations of images such as vector space structures, and model-based approaches are based on the model constructed by facial features or internal facial elements [57].

2.2.1 Appearance-based face recognition

Computer graph/object recognition is called appearance-based or view-based recognition if it is based on the representation of the whole images using a vector space structure [57]. View-based approaches consider an image as a vector. An image can be understood as a point in a high-dimensional vector space. Pixel values of an image are used directly. A set of images comprise an image space, which is represented as $X = (x_1, x_2, ..., x_n)^T$, while x_1 represents a $p \times q$ image and n is the total number of images in training group. X is a matrix of image vectors which is also called the image space. X is a $p \times q \times n$ data matrix. Figure 2.1 is a simple example of image space. The image in this space is a two-pixels gray-level bitmap image. It is clear that images which have similar gray values of pixels locate closer together, otherwise, their positions are far away from each other.

Appearance-based face recognition can be classified into Linear Analysis and Non-linear analysis. Classical linear appearance-based analysis include PCA, ICA and LDA. Each has its own basis vectors of a high dimensional face image space [57]. What they have in common is: by utilizing those linear analysis methods, the face vectors can be projected to the basis vectors. Through pro-

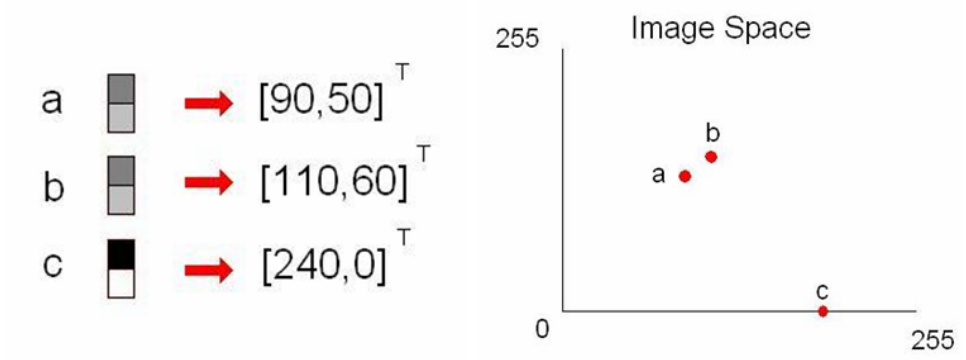
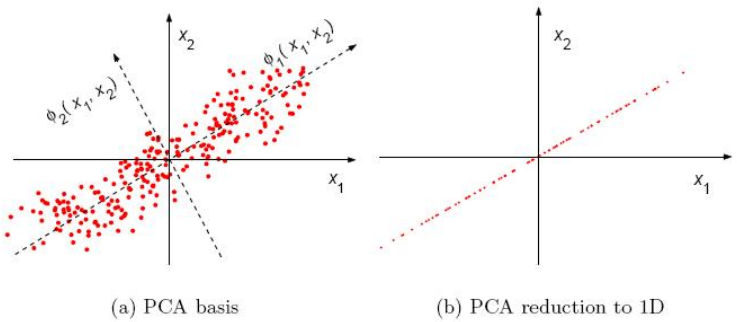


Figure 2.1: *Examples of three* 1×2 *pixel images and their positions in image space.*

jecting from a higher dimensional input image space to a lower dimensional space, dimensionality of the original input image space is reduced. The matching score between the test face image and training images can be achieved by calculation of the differences between their projection vectors. The higher the score corresponding to minimum distance, the more similar are these two face images.

The main idea of the Principal Component Analysis (PCA) [88] is to find the vectors which best describe the distribution of face images within the entire image space. PCA is an orthogonal transformation of the coordinate system in which the pixels are described. PCA aims to extract a subspace where the variance is maximized. PCA is performed by projecting a new image into the subspace called face space spanned by the eigenfaces and then classifying the face by comparing its position in face space with the positions of known individuals. Face space is comprised of eigenfaces, which are the eigenvectors of the set of the faces. The projection from the original image vectors to another vector space can be considered as a linear transformation. Figure 2.2 shows

principal components of a two-dimensional set of points. The principal components provide an optimal linear dimensionality reduction from 2D(a) to 1D(b). In face recognition, each point represents a face image in an image space. By applying PCA reduction, the distribution of the faces can be better described in a face space with lower dimensionality.



(a) PCA basis (b) PCA reduction to 1D

Figure 2.2: *Principal components of a set of points in 2D [81].*

PCA derives only the most expressive features which are unrelated to actual face recognition, and in order to increase performance additional discriminant analysis is needed. Independent Component Analysis (ICA) [46] provides a more powerful data representation than PCA. ICA is a generalization of PCA but the distribution of the components of ICA is designed to be non-Gaussian. The comparison between PCA and ICA is shown in figure 2.3. ICA seeks a linear transformation which can most reduce the statistical dependence between the components.

Similar image projections are close together, different image projections locate far away when using PCA, but the projections from different classes of images are mixed together. For example, female and male faces are not separated and

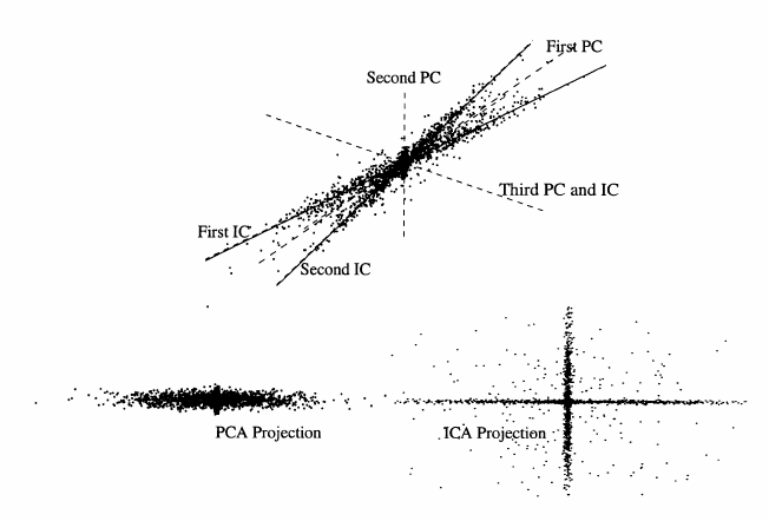


Figure 2.3: *A three dimension example of data distribution and the PCA and ICA axes. ICA uses a different face space than PCA. Left bottom shows the distribution according to the PCA coordinate of the data. Right bottom indicates that in this example ICA extracts better intrinsic distribution of the data [57].*

such class information is not used in PCA. Described by Belhumeur et al. [11], Linear Discriminant Analysis (LDA) exploits the face class information such as gender, age and nationality to help the recognition tasks, while such category information is not used in either PCA or ICA. LDA is able to maximize the ratio of between-class distribution to that of within-class distribution. This means that the training set for the LDA method can utilize multiple images of each individual to determine within-class variation, while eigenface uses only one image per person. Variations between images of the same person are minimized in the classification process. This is the main advantage of the LDA method over the eigenface method.

Linear discriminant methods concern the linear relationship between multiple

pixels in the images. Some non-linear relations may exist in a face image, especially under a complicated variation in viewpoint, illumination and facial expression variations which are highly non-linear. To extract non-linear features of images, the linear analysis method was extended to non-linear analysis such as Kernel PCA, Kernel ICA and Kernel LDA etc. By using non-linear analysis approaches the original input image space is projected non-linearly onto a high dimensional feature space. In this high dimensional space, the distribution of image vectors could be simplified to linear patterns. The face non-linear projection is more complex than linear projection. Figure 2.4 shows an example of PCA and KPCA. Unlike conventional PCA, Kernel PCA uses more eigenvector projections than the original input dimensionality but still uses the projection coefficients as features to classify. However, the suitable kernel and correspondent parameters will only be determined empirically [57]. In Yang's experiments [95], the conventional PCA, ICA and LDA approaches are compared to the non-linear analysis method kernel LDA. Experimental results based on two benchmark databases show that the kernel LDA methods are able to extract non-linear features and provide a more effective representation for face recognition and achieve lower error rates.

2.2.2 Model-based face recognition

The aim of model-based face recognition approaches is to produce a model to represent the facial variations. One significant advantage of model-based approaches is that it is convenient to make a good use of the biometrical knowledge of the human face. For example, model-based approaches may be based on the distance and relative position of features or internal facial elements (eyes, nose and mouth, etc.). The purpose of building a face model is to try

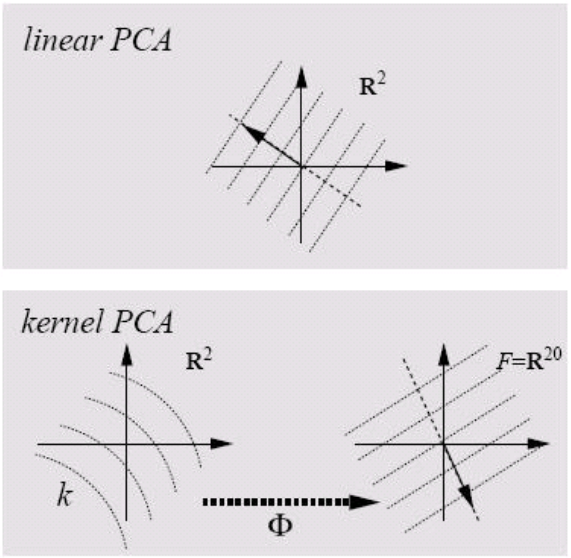


Figure 2.4: *Linear PCA and kernel PCA transformation. Kernel PCA uses a higher-dimensional projection. Linear PCA is performed in input space (top). Since the high dimensional feature space F (bottom right) is nonlinearly related to input space via ϕ the contour lines of constant projections onto the principal Eigenvector become nonlinear in input space. Kernel PCA does not actually perform the map into F. but instead performs all necessary computations by the use of a kernel function k in input space (R^2) [79].*

to eliminate the differences between the images of the same individual and emphasize the variance between different persons. Normally, the first step of the model-based methods is to construct the face model which contains the information of shape and texture of the face; then apply and fit the model to the face images within the training group; finally, compare the difference between the parameters of the fitted model of the test face and the training faces.

In 1973, Kanade [51] developed the earliest face recognition algorithms using automatic feature extraction. He detected the corners of eyes and nostrils in

frontal views by coarse scanning the gray-level pictures with a low-pass filter and then compared those features against the features of known faces. In 1992, Brunelli et al. [23] announced a system to recognize faces using 22 geometrical face features including eyebrow thickness and vertical position, nose vertical position and width, mouth vertical position, width and height, eleven radii describing the chin shape, bigonial breadth and Zygomatic breadth. Brunelli's experiment proved that geometrical feature recognition is effective. However, when the quantity of subjects increases to a large number, their system's capability to distinguish human faces is weakened because there is not enough information within these geometrical features to classify a great number of faces.

Wiskott et al. [91] developed a model-based matching system called elastic bunch graph matching. Since human faces have a similar topological structure, they classified the variance of a known class of individuals. A face can be structured as a graph called a bunch graph by nodes and edges. A Face Bunch graph (FBG) is generated from a set of training face images. The FBG serves as a general representation of a set of faces. In order to deal with the head orientation problem, different face bunch graphs of each possible orientation are generated. Among these face bunch graphs, a set of references are used to present the association of nodes at the same fiducial point in different bunch graphs. To perform the graph match between a query face image and other images in the training set, image graphs are produced by adaptation of the face bunch graph to fit the face of the query image. The face bunch graph is scaled and distorted to maximize a graph similarity between this graph and the FBG. Then the probe face is recognized by comparing the similarity between the graph of this face and graphs of every face stored in the FBG.

In 1998, Cootes et al. [33] [32] introduced a morphable face model - the Active Appearance Model (AAM) which is a 2D statistical model to capture the variation of shape and appearance of a human face from a full profile viewpoint to a frontal viewpoint. Any new image can be matched rapidly by finding the model parameters which minimize the difference between this image and the synthesized model. The AAM is potentially able to estimate the head pose of a probe image by finding the best fitting model to produce new views from the similar pose of the new image. Models are built based on a set of labeled images. Landmark points are marked on each example face image at key positions to describe the facial features. A set of models is used to describe the variation of the head orientation from different viewpoints. When matching a new face in which the head orientation is unknown, the head pose can be estimated by searching with each of these models to determine the best match. Given a probe image, the goal of recognition is to find the best match between the test parameter vector and the training data. They implemented their experiments on faces with different poses and claimed the highest recognition rate is over 97%.

2.3 3D face recognition approaches

2D face recognition approaches use grey scale or color 2D images to perform face recognition. Unfortunately, these 2D methods have some weaknesses handling the head orientation problem. For example, in Elastic Bunch Graph Matching system, the system requires the two images involved in the matching process to be at the approximated head pose. Otherwise, if two images at

different viewpoints are matched, reduction of the identification rate will be observed. Another weakness of the 2D face recognition system is the illumination problem. The variation of the lighting conditions also will change the texture information of a face and therefore may cause a poor performance on a 2D face recognition approach.

Since a face surface is naturally a 3D surface, using 3D images to describe faces is capable of capturing more details such as depth information than using 2D images. Moreover, the 3D shape (independently obtained without using 2D data) can not be affected by the illumination variations. Thus, if the 3D image is able to be captured reliably and precisely, exploiting the 3D depth or shape information is able to provide a pose-invariant information, which can lower the significance of the texture data. That means that the negative effect of the different lighting conditions could be diminished or even removed if texture information is not used. Unlike 2D images which could occlude some parts of a face due to a severe head rotation, the 3D face image contains a face shape and any pose variation does not result in a surface loss/occlusion. The head pose problems could be tackled through analyzing the 3D image, because a 3D image contains the information in any rotation direction. In the meantime, the extra information of the third dimension may enlarge the discrimination between different faces simply because it provides extra differences in the third dimension. Consequently, the 3D face recognition is expected to have more advantages to handle the problems of head pose and illumination than the face recognition in 2D images.

2.3.1 3D face recognition approaches based on 2D face recognition algorithms

PCA is a widely used algorithm in 2D face recognition to reduce the dimensionality and classify the faces. Hesher et al. [43] extended the PCA algorithm to 3D face recognition. Multiple images per person were used and stored in their gallery dataset. They treated the 3D image as a cloud of points and applied PCA directly to the point clouds. Their experimental performance reports an identification rate of 100% on a small dataset with expression variations. Another investigation of PCA in 3D face recognition has been presented by Chang et al. [25]. They applied the PCA on both intensity(2D) and depth(3D) images then fused the two results. The experiments were implemented on a relatively large database with 275 subjects. The identification rate for intensity images is 89.5% and the experiments for depth images achieved an identification rate of 92.8%. After combination of results is performed, the identification rate increased to 98.8%. Heseltine et al. [42] proposed a method using PCA on the facial surface representations created by convolution kernels and distance metrics. An identification rate of 87.3% is achieved in his experiments based on the University of York 3D face database. Another approach introduced by Heseltine et al. [41] used the fisherface algorithm to obtain an identification rate of 88.7%. Cook et a. [31] presented a 3D face recognition system using Log-Gabor filter. The face image is divided into many squared regions and subregions. A set of 147 features are extracted by applying PCA to each filter response of regions/subregions for each face. Then faces are matched by exploiting Mahalanobis-Cosine distance of two feature sets. The experiments were performed on FRGC v2 database, and a rank-one identification rate of 96.2% and a verification rate of 92.3% at 0.1% FAR are reported.

Cartoux et al. [24] segment face images based on principal curvature and find the face's bilateral symmetry plane. They used this plane to normalize for pose and used the methods to match the profile based on the symmetry plane. Their experiments implemented on a small database reported an identification rate of 100%. Nagamine et al. [67] localized five feature points and then utilized those points to normalize head orientation. Vertical profiles that pass through the central portion of the face are matched through face data. Beumier et al. [15] established a system using the central and lateral profiles both in 2D and 3D to classify faces. The final results are created based on a weighted sum rule to fuse the similarity score in 2D and 3D.

Similar recognition methods on 2D morphable models can be improved and applied on 3D models as well. 2D face models represent the shape and texture parameters of the model independently. However, only part of such information is distinguished from the imaging conditions, such as head pose and illumination. Thus V. Blanz et al. [18] [17] established a system for face recognition based on fitting a statistical, morphable model of 3D faces to images as an extension of a 2D morphable model. One aim of that system is to separate the intrinsic model parameters of the face from extrinsic imaging parameters. During the model fitting process, the shape and texture coefficients are optimized as well as other rendering parameters, such as pose angels, head position, size, color and intensity of the illumination etc. The similarity can be considered as the difference between the model coefficients of these two images, such as the sum of Mahalanobis distances of the segment shapes and textures. They claimed an identification rate of 95% on CMU-PIE [83] and 95.9% on FERET dataset [71]. Lu et al. [58] presented an approach using a 3D model to produce

several different 2D images. 2D images with different poses, illuminations and expressions are synthesized from a 3D model. They used a database of 10 subjects to synthesize 22 images per person with variations in pose, expression and illumination. They claimed a identification rate of 85% which outperforms the PCA-based algorithms using the same database. Unfortunately, the small number of images and subjects used in this experiment lower the reliability of this method. Both Blanz et al. and Lu et al. used various 2D face images synthesized from a 3D model and applied the classical approach in 2D face recognition to overcome the pose, illumination and expression problems. However, there are some concerns [6]: how much verisimilitude and accuracy can a synthesized face image provide?

As well as the depth image, texture or color information also can be utilized in 3D face recognition. Tsalakanidou et al. [87] utilized the color and depth information to establish a multi-modal face recognition system. They first localize the face by using depth and brightness information. The recognition is performed by applying the Embedded Hidden Markov Models(EHMM) to depth and color information. The results of color and depth image are combined to produce an identification rate of 91.67%. Xu et al. [92] proposed a novel system to describe the local features by using Gabor wavelets which are extracted from depth and intensity information. The most effective and robust feature are chosen based on a novel hierarchical selecting scheme embedded in LDA and AdaBoost learning to build an effective classifier. Their experiments are performed on FRGC v2 database and CASIA 3D face database. A verification rate of 97.5% in "neutral vs all" experiment is claimed.

2.3.2 3D face recognition using shape analysis

Some of previously introduced 3D face recognition approaches used 2D texture information which will bring illumination problems. Some approaches only used parts of 3D information which may lose some useful information. These are the main problems of above approaches. Another kind of approach is to convert 3D information to other representations or describe the 3D surface by using a shape descriptor.

Wang et al. [89] performed a multi-modal 3D face recognition using point signatures in 3D images. They also use a 3D feature together with the 2D feature produced by using a Gabor filter. Support Vector Machine(SVM) is used in classification. An identification rate of 90% is reported. Bronstein et al. [22] analyzed 3D face by using an isometric transformation approach. They used a bending invariant canonical representation to overcome the expression problems. The facial expressions can be modeled by applying isometric transformation. 2D texture is also flattened and mapped to the canonical image. Their experimental results only show examples without reporting any recognition rate.

A. Mian et al. [65] proposed tensors matching for pose invariant 3D face recognition system. They defined a $15 \times 15 \times 15$ 3D bin grid to crop 3D faces. The surface area of the face crossing each bin of the grid is recorded in a third order tensor. Each element of the tensor is the face surface area that intersects the bin which corresponds to this tensor element. Then the linear correlation coefficient of two tensors are computed to measure the similarity between two faces. They reported a recognition rate of 86.4% on a dababase of 277 subjects. The main problem of Mian's approach is that it is to sensitive to misalignment

of faces. They also did not consider the problems of expression variations.

In 1999, Johnson and Hebert [48] first introduced the Spin Image to describe 3D shape and then used it to recognize 3D objects. They defined an oriented point at a surface vertex using the 3D position of the vertex and surface normal at the vertex. The surface normal at a vertex is then calculated by fitting a plane to the points connected to the vertex. Two cylindrical coordinates are defined according to this oriented point: the radial coordinate α, defined as the perpendicular distance to the line through the surface normal, and the elevation coordinate β defined as the signed perpendicular distance to the tangent plane defined by vertex normal and position. α and β are computed for all vertices. The bin indexed by α and β is then incremented in the accumulator. The resulting accumulator can be considered as an image. Wang et al. [90] used a Sphere-Spin-Image(SSI) technique to describe the local 3D shape. The main idea of SSI is to map the 3D points within a sphere to a 2D histogram. They used a series of points to produce a set of SSI histograms to represent a face. The similarity between different sets of SSI is measured by using a simple correlation coefficient. The experiment performed on 31 models achieved an identification rate of 91.68%. Conde, Rodriguez-Aagon and Cabello [29] also make use of the Spin Image to implement a feature points selection to find the nose tip and eye corners. Then they used these feature points to normalize faces to create depth maps. Face verification experiments were implemented by analyze the linear relation of the depth maps. They reported a Equal Error rate of 2.59% on FRAV3D database. The main problem of the Spin Image approaches is that it requires an accurate feature point localization to fix the position of the selected oriented point as the origin point to create the spin image. The Spin image can be considered as a projection from 3D to 2D which

may lose some information.

In 1986, Besl and Jain [13] introduced an invariant surface characteristics method to recognize 3D objects. They used two fundamental second-order surface characteristics which represent extrinsic and intrinsic surface geometry respectively to describe 3D shape and capture domain-independent surface information. Tanaka et al. [85] used a descriptor based on maximum and minimum principal curvature and directions to represent a face shape. The descriptor is then mapped onto two spheres called Extended Gaussian Image(EGI). Then they measure the similarity between EGIs by using Fisher's spherical approximation. A 100% identification rate on a small database is reported. However, Stein and Medioni [84] pointed out that the computation of curvature requires a higher order derivative than the tangent. That implies that the signal to nose ratio is lower for a curvature based representation than for a tangent based scheme.

The Iterative Closed Point (ICP) algorithm is first introduced by Besl et al. [14]. ICP is a method widely used to fit points in a target image to points in a standard model. The target group of points is aligned to the model by minimizing the sum of square errors of pairs of corresponding points. Firstly, the position and overlay of two images are estimated. Then, based on the initial estimate, a translation and rotation matrix is computed and applied to minimize distances between each pair of corresponding points. The transformation procedure is iteratively performed until the sum of distances between corresponding points falls below a particular preset threshold. ICP is an effective method to reduce the misalignment in face registration. Meanwhile, ICP also can be used to match the difference between two faces. Details of Besl's

ICP algorithm is introduced in Appendix B.

Recently, many ICP-based face recognition approaches were published. Lu et al. [58] implemented a method to extract feature points in 3D face images by classifying the local minimum and maximum of curvature. Then ICP is applied on those points to align face images. Faces are matched by using the local features correlated by ICP. They used a database with 18 subjects and in total 113 3D face images. An identification rate of 96.5% is reported. In [60] Lu et al. exploited ICP and LDA to match 3D models synthesize by multiple 2.5D face images. In their experiment on a set of faces with various poses and expressions, they found that almost all mistakes in recognition are caused by expression variations. In their further research [59], a deformable model is proposed to match 2.5D faces with different expression and pose. Each expression has its synthesized deformation template. A neutral face can use those templates to generate a 3D deformation model. ICP is then applied in model matching. They reported that using the deformation models, the identification rate exceeds that obtained without using deformation models. Papatheodorou et al. [69] presented a face recognition approach adding texture information into the ICP algorithm. The similarity between faces is produced by measuring the 4D Euclidean distance of three spatial dimension value and the texture information. They reported an identification rate from 66.5% to 100% according to different head orientations and expressions. Chang et al. [26] segmented the whole face into several regions by using a method called Adaptive Rigid Multi-Region Selection(ARMS). They considered the regions around the nose to be the expression invariant regions. Those regions are matched with their corresponding ones in another face by using ICP algorithm. The results of matching are evaluated by Root Mean Square Error(RMSE). The product rule is applied

to fuse the similarity scores of different regions. Experiments on neutral images in FRGC database report a rank-one identification rate of 97.1%. A rank-one identification rate of 87.1% is achieved using faces with expression variations. Mian et al. [64] introduced an approach to fuse 2D and 3D face recognition. They use Scale-Invariant-Feature Transform(SIFT) to extract local features in 2D images. Matches are measured by the Euclidean distance between features. In 3D face recognition, they first use a PCA pose correct method to align 3D faces. Then the 3D face is segmented into a nose region and an upper face region including eyes and forehead. The ICP algorithm is applied to match different regions. The overall similarity score is produced by combining each 2D or 3D matching methods. Kakadiaris et al. [50] designed an automated 3D face recognition framework. An annotated face model (AFM) is used to deal with the expression variations. The face image is aligned to the AFM model by a combination of three matching/alignment algorithms: Spin image, ICP and Simulated Annealing(SA) on Z-buffers. A deformation image is generated by the fitted model. Two wavelet transformations, Pyramid and Haar, are used respectively. The match is implemented by measuring the distance metric for each wavelet type. A 97% rank-one identification rate is reported in the "first vs other" experiment of FRGC v2 database. A verification rate of 97.0% is obtained in the ROC III experiment of the FRGC v2. Faltermier et al. [35] established a framework to combine matches based on 28 facial regions. ICP is applied during the matching of corresponding regions. Consensus Voting and Borda Counting are used as fusion methods to combine different matching scores. In experiments on the FRGC v2 database, they reported that a rank-one identification rate of 97.2% and a verification rate of 93.2% are obtained.

Using similar idea as ICP, Chaua et al. [74] presented a framework to perform 3D face recognition by using Simulated Annealing(SA) and Surface Interpenetration Measure(SIM). They use SA to implement face registration and exploit SIM rather than RMSE to measure the difference of two surfaces. A set of regions including the full face, the upper face, the nose region and the expression-invariant region are segmented and matched respectively. Then the final similarity is obtained by combining all results of regions. They claimed a verification rate of 96.6% in "all vs all" experiment and a rank-one identification rate of 98.4% in "first vs other" experiment by using FRGC v2 database, which are so far the best results based on FRGC v2 database.

In the pure 3D face recognition approaches, several algorithms [26] [64] [50] [35] [74] based on the 3D shape matching both achieved an outstanding performance especially ICP-based approaches. However, implementing a surface matching algorithm such as ICP or SA/SIM in the face recognition is a time-consuming task. There are usually more than thousands of face matches occurring in the face recognition experiment or in a practical face recognition system. Furthermore, ICP and its variant are also frequently used in the face registration stage as well as the SA/SIM algorithm [74]. Points of a face are repeatedly used in the computation of the surface matching algorithm in the face alignment and face recognition stages [35] [74]. Therefore, a more efficient face recognition algorithm using surface/shape matching method is required. Moreover, the feasibility of a 3D face recognition method depends on its ability to deal with at least two key problems:(1)head orientations;(2)expression variations. The evaluation also takes into account how many people and images in the experimental database. A small dataset is not convincing enough to justify and evaluate an approach.

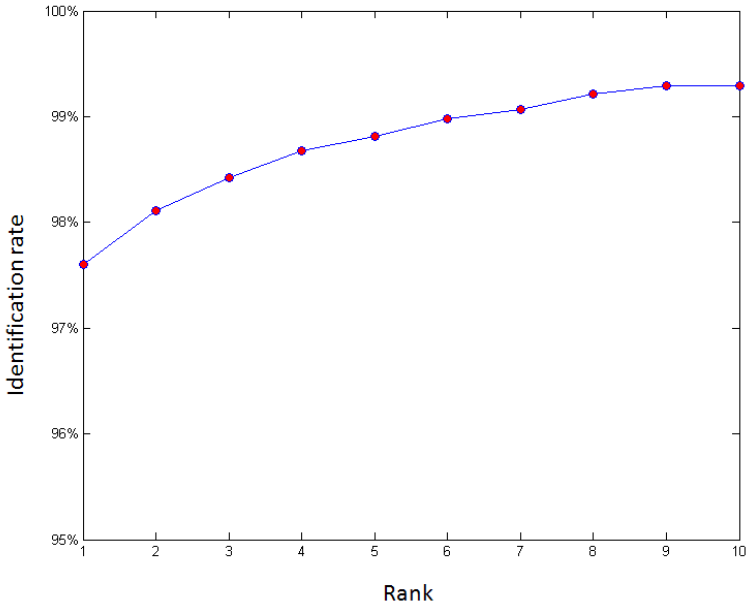


Figure 2.5: *An example of the Cumulative Match Characteristic(CMC) curve.*

2.4 Face databases and performance evaluation

In order to evaluate the performance of a face recognition system, some general principles should be established. In most published papers, two face recognition scenarios are evaluated: identification and verification. For the identification scenario, the most widely used way to show how good a face recognition system will be is to give the rank of the matches between the test face and gallery faces. Then a rank-one identification rate is produced by calculating the number that correctly identifies (at first rank position) the same subject from a group of gallery faces. Identification rates at different ranks also are

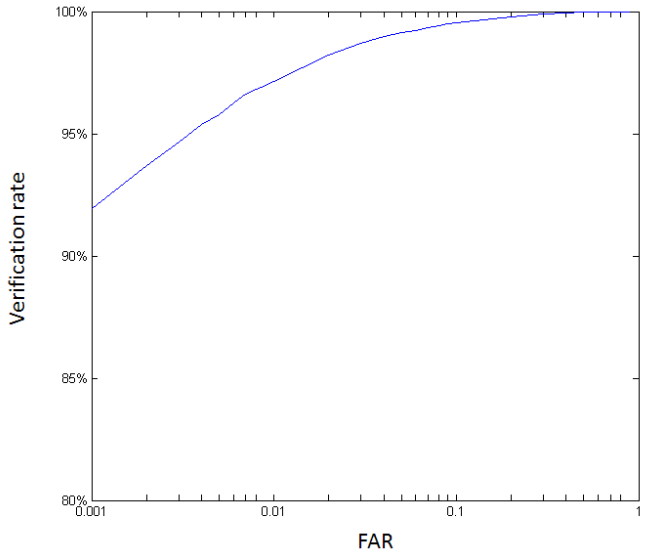


Figure 2.6: *An example of the Receiving Operating Characteristic(ROC) curve.*

computed to plot a Cumulative Match Characteristic(CMC) curve as shown in figure 5.15. For verification, there are several protocols for subject identification and verification: False Acceptance Rate (FAR), False Rejection Rate (FRR). FAR is the percent of cases that incorrectly accept a correct match. FRR is the probability that the system incorrectly declares failure of the match between the input pattern and the matching template in the database. Equation 2.1 and 2.2 shows how to calculate FAR and FRR respectively. Generally, Verification Rate(VR) (shown in equation 2.3) at different FAR are produced, then a Receiving Operating Characteristic(ROC) curve(an example is shown in figure 2.6) is created to show the verification performance of a face recognition system.

$$FAR = \frac{n}{N} \tag{2.1}$$

Where n is the number of matches between different subjects being incorrectly considered as a correct match, and N is the total number of matches between different subjects.

$$FRR = \frac{m}{M} \tag{2.2}$$

Where m is the number of matches between same subjects being considered as an incorrect match, and M is the total number of matches between same subjects.

$$VR = 1 - FRR \tag{2.3}$$

There are more than 20 face databases available currently. These face databases are constructed and designed for different face recognition tasks. Researchers choose the appropriate database normally based on the task given (aging, expressions, lighting etc). As fast 3D data acquisition devices become cheaper and more reliable, more and more 3D face databases begin to be available to face recognition researchers. 3D face images are normally captured by laser scanning devices or 3D cameras. As well as the depth or 3D information, texture information for some databases also can be obtained. Table 2.1 lists details of the 3D face databases available to academic researchers.

It is not easy to benchmark of all the algorithms because the researchers have their own choices of database. For the same algorithms, the recognition rate may vary due to different evaluation protocols and different image resources.

Therefore, for the testing or comparison of different face recognition systems, the standard database and evaluation method have to be decided. Face Recognition Vendor Tests (FRVT) 2006 [73] follows five previous face recognition technology evaluations - three FERET evaluations (1994, 1995 and 1996) and FRVT 2000 and 2002 [71] [75] [76]. In FRVT 2006, a standard dataset and test methodology is employed so that all participants are evenly evaluated. Both the test data and the test environment will be provided to participants. The test environment is called the Biometric Experimentation Environment (BEE). It allows the experimenter to focus on the experiment by simplifying test data management, experiment configuration, and the processing of results. The Face Recognition Grand Challenge (FRGC) [72] is then being conducted to fulfill the comparison of new techniques as one of the goals of the FRVT 2006. The FRGC is open to all face recognition researchers and developers from companies, academic or research institutions.

Among those 3D face database listed in table 2.1, Face Recognition Grand Challenge 3D face database(FRGC) has the largest number of individuals and face images including pose and expression expression variations. A great number of researchers implemented their approaches and experiments based on FRGC database [31] [92] [26] [50] [35] [74]. In this thesis, all experiments are performed on FRGC 3D face database. The details of the FRGC 3D face database are introduced in the Appendix A.

2.5 Summary

This chapter presented a review of the classical 2D/3D face recognition algorithms and a number of state-of-the-art 3D face recognition approaches. Compared with 2D face recognition approaches, several significant challenges: 3D face detection, pose variations and expression variations are the key problems of the 3D face recognition. From the review of the covered face recognition techniques, 3D face recognition algorithms based on shape/surface analysis/matching achieved a good performance on large face databases such as FRGC v2 database, which gives us a direction of research. In the following chapters, we plan to solve those challenges step by step and finally implement a high performance automatic 3D face recognition system.

Database	Subjects	Images	Texture	Conditions	Availability
Xm2vtsdb [62]	295	2/subject	yes	pose	charge
3D RMA [1]	120	3/subject	no	orientations	free
GavabDB [66]	61	549	no	pose, expression	free
FRAV3D [5]	106	16/subject	yes	poses, lighting	free
BJUT-3D [2]	500	500	yes	n/a	free
Univ. of York 1 [3]	97	10/p	no	pose, expressions, occlusion	free
Univ. of York 2 [3]	350	15/p	no	pose, expressions	free
Bosphorus [4]	105	31-54/p	yes	pose, expressions, occlusions	free
FRGC v1 [73] [72] [71]	275	943	yes	illumination, pose, expressions	free
FRGC v2 [73] [72] [71]	466	4007	yes	illumination, pose, expressions	free

Table 2.1: *Most available 3D face database.*

Chapter 3

Feature localization

3.1 Introduction

A 3D face is a group of high dimensional vectors of the x, y and z positions of the vertices of a face surface. The R, G and B color information can be added into this vector if the texture values of those vertices is required. A 3D face is usually represented by a 3D shape file and 2D texture image. Face recognition based on 3D has the potential to overcome the challenging problems caused by expression and illumination variations [20]. However, many 3D face recognition approaches, especially the feature-based ones, require a robust and accurate facial feature localization.

This chapter focuses on the task of identifying and localizing 3D facial features. As the nose tip is the most prominent feature of the face, many works [74] [35] [92] [60] [64] perform nose tip detection and use the nose tip as the foundation to detect other features. Some facial feature identification algorithms use an assumption that the nose is the closest point to the camera or device which acquires the 3D data [43] [55]. Although this supposition is

Figure 3.1: *An example of bad 2D-3D correspondence.*

true in most cases, there is no guarantee because the noise, pose rotations and the complex situation of hair and clothes could make some places closer than the nose.

Making use of the corresponding 2D texture information is a possible way to detect the face area first then localize the nose tip within the selected 3D face crop. That requires 2D texture and 3D shape to correspond correctly. However, in some face datasets such as Spring2003 subset of FRGC, the 2D texture channel is not always perfectly matched with the 3D shape channel (as shown in figure 3.1). Using the 2D face crop method in a face with a poor 2D-3D corresponding will often obtain a poor 3D shape crop.

Colombo et al. [28] presented a method to identify the shape of facial features based on 3D geometrical information only by using HK Gaussian Curvature classification. They achieved a 96.85% identification rate on a small dataset, although only the rough nose/eye shapes are identified and no accurate loca-

tions of nose tip or other features are detected. Of other algorithms, Bevilacqua et al. [16] implemented an experiment to detect the nose tip based on extending the Hough Transform to 3D point cloud. However, only 18 3D faces are involved in the experiment. Spin image and support vector Machine (SVM) are used to represent and classify 3D shape [29] [93]. In [93], a 99.3% successful localization rate of the nose tip is claimed, but it was tested on a limited dataset without benchmark evaluation. The main problem of those approaches is that they only used a small face database which is not enough to evaluate the performance of the facial feature localization. A small database does not provide enough cases about different noise and variations which are crucial in performance evaluation.

Segundo et al. [80] proposed a 3D facial landmark detection based on the analysis of y-projections and x-projections of the topographic depth information. They used a combination of region/edge detection algorithms and a Hough transform based shape detection method to localize the main face area first and then detect facial features. They reported a nose detection rate of 99.95% on FRGC v2 database. However, using methods to detect face area first may result in extra chance of mistakes and they did not report the accuracy of their face detection.

To the best of our knowledge, most of the methods do not use benchmark datasets to evaluate their results. Romero et al. [77] presented the first work on benchmark datasets based on FRGC database. They manually marked landmarks of eleven facial features. With those marked feature locations, the results of automatic feature identification can be measured and evaluated.

Some approaches mentioned above use the HK Gaussian Curvature or the mean value and derivative of other attributes to represent 3D shapes. Within a sphere of radius r at a point P, some statistical attributes such as mean and derivative values are computed for point P and its neighbouring points P_i. However, over a very large number of faces, the effectiveness of representation could be impaired because of the noise caused by clothes, hair and other unwanted facial features. To solve this problem, we use more attributes to describe a 3D surface. The number of attributes can be increased according to the requirements of different feature identification tasks. More attributes mean describing a piece of 3D shape will create a relative complex pattern which requires a powerful classification method. In this thesis, we use a binary neural network technique based on Advanced Uncertain Reasoning Architecture (AURA) to implement the facial feature matching and searching. Each point P will have a similarity score to tell how much it looks like the trained features.

This chapter is organized as follows. In section 2, two 3D local shape/surface descriptors called Multi Contour Surface Angle Moments Descriptor(MCSAMD) and Multi Shell Surface Angle Moments Descriptor(MSSAMD) are introduced. Section 3 describes the feature matching and searching algorithm based on a binary neural network which is called AURA k-Nearest Neighbour technique. The methodology for nose tip identification using both MCSAMD/MSSAMD and AURA k-NN is presented in section 4. Section 5 proposes the medial canthi (eye corners) localization using the same method after the nose tip detection is implemented. Section 6 shows the experimental results and proved that using MCSAMD and MSSAMD, the feature especially the nose tip can be located more precisely than with other methods. Section 7 makes the con-

clusion of this chapter.

3.2 3D Local Shape/Surface Descriptor

3D facial features can be considered as small groups of points and pieces of 3D surface. There are many methods to describe a 3D shape or surface. In 1984, Grimson and Lozano-Perez [38] first discussed how local measurements of 3D position and surface normals recorded by a set of tactile sensors may by used to identify and locate objects. They mentioned that angles relative to the surface normal is an efficient local constraint. Compared with curvature-based shape descriptors, Stein and Medioni [84] proposed a method using a splash structure to describe a surface. At a given location P they compute the surface normal n. Then a circular slice around n with the geodesic radius r is computed. A surface normal $n^{`}$ can be determined at every point on this circle. $\theta$ angles between the $n$ and all $n^{`}$ are obtained. By using splashes, a 3D surface can be described. They also stated that the computation of curvature requires a higher order derivative than the tangent. For a curvature based scheme, the signal to noise ratio is lower than for a tangent(or surface normal) based scheme. In 1997, Chua and Jarvis [27] introduced the Point Signature method to describe a 3D shape. They used a sphere to crop a 3D shape at a point P. Then a number of contour points are produced. The surface normal and normal plane also can be calculated at the point P. Distances d from the contour of points to the normal plane are computed starting from a certain position along a clockwise direction. d and the angle θ of the clockwise rotation together can be used to describe a 3D surface within a sphere. Rather than only use the contour points cropped by a sphere, Xu et al. [93] computed

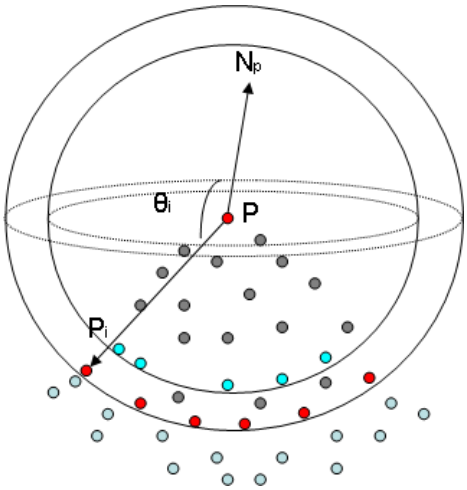


Figure 3.2: *P and its neighbouring point within two spheres.*

the distances d of all points to the normal plane at the center point P within a sphere. Then the central and second statistical moments - mean and the deviations of these d are computed. A 3D surface patch cropped by a sphere is described using these two moments. Inspired by the above approaches, in this thesis, the moments of the local shape characteristics - angles related to the surface normal are used to describe a 3D surface. We provide a novel method to describe the convex or concave degree of 3D local shape within a given sphere but related to a number of shells.

3.2.1 Multi Contour Surface Angle Moments Descriptor

For a point P in a 3D point cloud, itself and its neighbouring points P_i together forms a 3D surface as shown in figure 3.2. By finding all the points P_i with the length of edge $P_i - P$ approximately equal to the radius r, point P and those P_i create a $1-$ring mesh. Then the angle between $P_i - P$ and the

vertex normal N_p can be calculated by using the following equation:

$$\theta = \arccos\left(\frac{(P_i - P) \cdot N_p}{|P_i - P||N_p|}\right) \tag{3.1}$$

where N_p is the vertex normal of point P, θ is the angle between the vertex normal N_p and the edge $P_i - P$, r is the radius of a sphere. θ is between $0^o \sim 180^o$

After the θ of all farthest neighbouring points are calculated, each point P has one of the farthest neighbouring point set $PF(P) = \{P_1, P_2, ..., P_n\}$ and one angle set $\theta(P) = \{\theta_1, \theta_2, ..., \theta_n\}$($n$ is the number of farthest neighbouring point). By calculating the mean θ using equation 3.2, we can find out how convex or concave the mesh surface is. For instance, if the mean θ_i of all those farthest neighbouring points is greater than 90^o, this surface within a sphere of radius r can be considered as a convex surface. When the mean of θ_i is less than 90^o,the surface will be a concave one.

$$mean(\theta) = \frac{1}{n}\sum_{i=1}^{n}\theta_i \tag{3.2}$$

The calculation of θ requires the direction of vertex normal at a point P. As mentioned by Xu et al. [93], this method has a very large computational load on localizing the neighbouring points. The computational cost of this method is $O(n^2)$ (n is the total number of points) distance calculations of all points. As most 3D face databases such as the FRGC 3D face dataset are captured by a structured light sensor, all the points of a face have an ordered index. Thus it is easy to find all the neighbouring points P_i of a particular point P in terms of the vertical and horizontal relationship between those points. Therefore, we can use an approximate algorithm to simplify the computation. For example,

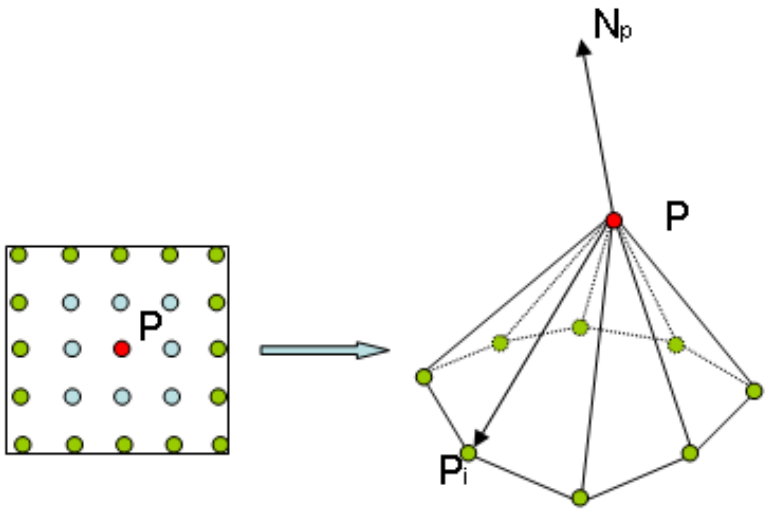


Figure 3.3: *P and neighbouring points P_i consist of a 1-ring mesh.*

as shown in figure 3.3, P has 24 neighbouring points within a 5×5 grid. We can use point P and its farthest neighbouring points (the outermost circle of grid) to create a 1-ring mesh. The cost of computation is reduced to $O(nm)$ where m is the number of neighbouring points of the point - P_i. This approximate algorithm may cause a scale problem because different faces contain different numbers of points, but it is possible to solve this problem by training faces with different numbers of points.

According to the comparison of algorithms for vertex normal computation made by Jin et al. [47], the mean weighted equally algorithm (MWE) is the fastest one and it works well in most cases. Therefore, MWE is used for the calculation of the vertex normal. Equation 3.3 is used by Jin et al. to calculate the vertex normal by using MWE algorithm.

$$N_{MWE} \| \sum_{i=1}^{n} N_i \tag{3.3}$$

where the summation is over all n triangle faces adjacent to the point P. The '||' makes implicit the normalization steps.

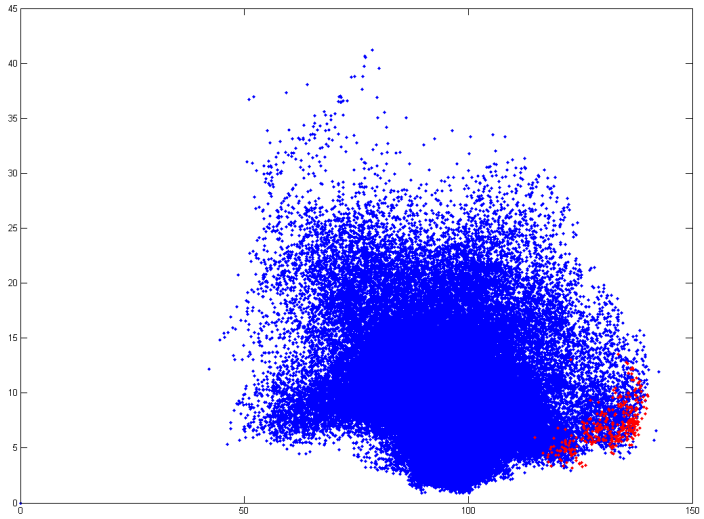


Figure 3.4: *The distribution of points according to mean(θ) and STD(θ), red points are nose tip and their neighbouring points (within a sphere), blue points are the other points.*

However, mean θ above is not enough to describe the subtlety of 3D shape. Therefore, we use the two statistical attributes: mean and standard deviation (calculated by equation 3.4) of θ to simply represent the shape within a sphere. By using these two features as a 2D space coordinates, the 3D local surface is projected into this 2D space. Moreover, only two attributes probably will lose a lot of information. The various situation of the clothes and hair in the FRGC dataset sometimes may cause unexpected points to have similar mean

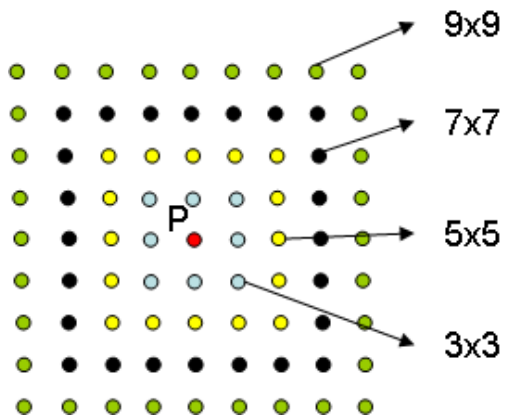


Figure 3.5: *An example of different grid sizes.*

value and STD of θ to an expected local facial feature. As shown in figure 3.4, we can see the distribution of the nose tip points and its neighbouring points. Although the nose tip points are clustered together, there are still other non-nose points mixed among them.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(\theta_i - \overline{\theta})^2 \tag{3.4}$$

Ankerst et al. [9] introduced 3D shape histograms as an intuitive and powerful similarity model for 3D object. Among three technique for decomposing the 3D information, they suggested a multi-shell model. The 3D surface/shape is decomposed into concentric shells around the center point which is particularly independent from a rotation of the objects. Any rotation of an object around the center point results in the same histogram. Inspired by M. Ankerst's work, we introduce more circles to calculate mean and deviation of angles. Those two kinds of attributes are used with more than one different grid size as shown in figure 3.5 to create a Multi Contour Surface Angle Moments Descriptor(MCSAMD).

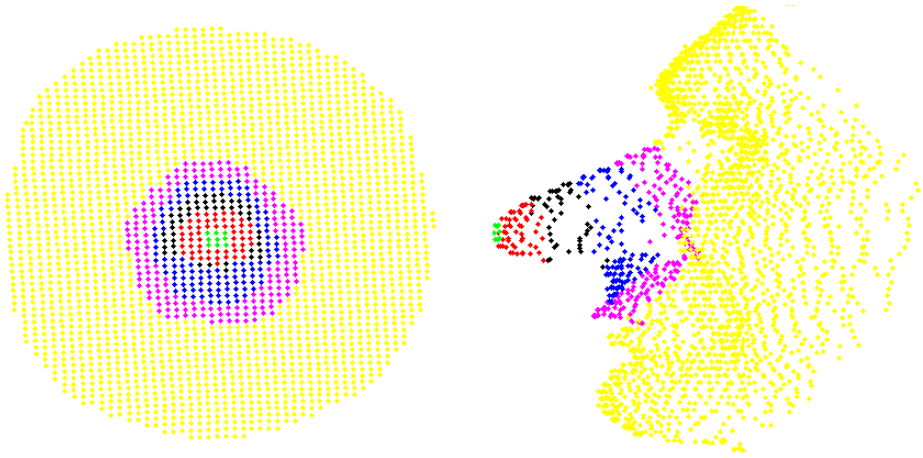


Figure 3.6: *The 3D surface is separated by several shells around a point.*

This MCSAMD descriptor depends on the order of points. When the orientation of the head varies, the order of points will change. Therefore, MCSAMD descriptor is not a complete orientation invariant method to describe a 3D surface.

3.2.2 Multi Shell Surface Angle Moments Descriptor

If we use spheres to replace the grid circles in MCSAMD, another similar descriptor is created. Every point between two spheres is used to compute the mean and standard deviation of θ shown in figure 3.2. According to the reasons mentioned above, only one pair of standard deviation and mean value is not enough to describe the shape of a 3D surface and further to precisely classify them. Thus, increasing the number of spheres to produce more shells between spheres is a simple solution. An example of this new descriptor called Multi Shell Descriptor(MSSAMD) is shown in figure 3.6.

Given the surface normal N_p at point p shown in figure 3.2, the θ_i represents the angle between N_p and PP_i, where P_i is one neighbouring point of point P. Each 'shell' has its standard deviation and mean value of the angles of the points located in its range. Therefore, a 3D surface is described by this MSSAMD including two vectors:

$$[std_1, std_2, ..., std_n] \tag{3.5a}$$

$$[mean_1, mean_2, ..., mean_n] \tag{3.5b}$$

Since it is difficult to give the neighbouring points a particular order by using MSSAMD, the MSE algorithm can not be used. Xu et al. [93] and Romero et al. [77] use similar methods to compute the third eigenvector of the covariance matrix as the direction of the normal on point P. Given point $p(x, y, z)$ as the center of a sphere and its neighbouring points $p_i(x_i, y_i, z_i)$ inside the sphere, the covariance matrix of point p is:

$$C = \frac{1}{n}\sum_{i=1}^{n}(p_i - m)(p_i - m)^T \tag{3.6}$$

$$CV = DV \tag{3.7}$$

where m is the mean vector of all points, V is the matrix of eigenvectors and D is the matrix of eigenvalues.

Since the $p(x, y, z)$ is a three dimensional vector, by means of PCA three eigenvectors can be obtained and each of them represents three directions which are orthogonal to each other. According to the definition of PCA, the corresponding eigenvalues of these three eigenvector show the degree of data distribution. Since the shape of face is a barrel like shape, when we use a sphere to cut a piece of 3D surface, the corresponding eigenvalue of the surface normal direction will

be the least of three eigenvalues. This has been confirmed by reviewing 943 faces in the training set of FRGC database. Figure 3.7 shows the histogram of the ratios of height/width, height/depth, width/depth at the nose tip where is the most prominent place of the face($r = 25mm$). We can find the values of the height and the width of a certain face are both greater than its depth. Thus, the eigenvector corresponding with the smallest eigenvalue is the surface normal at point p.

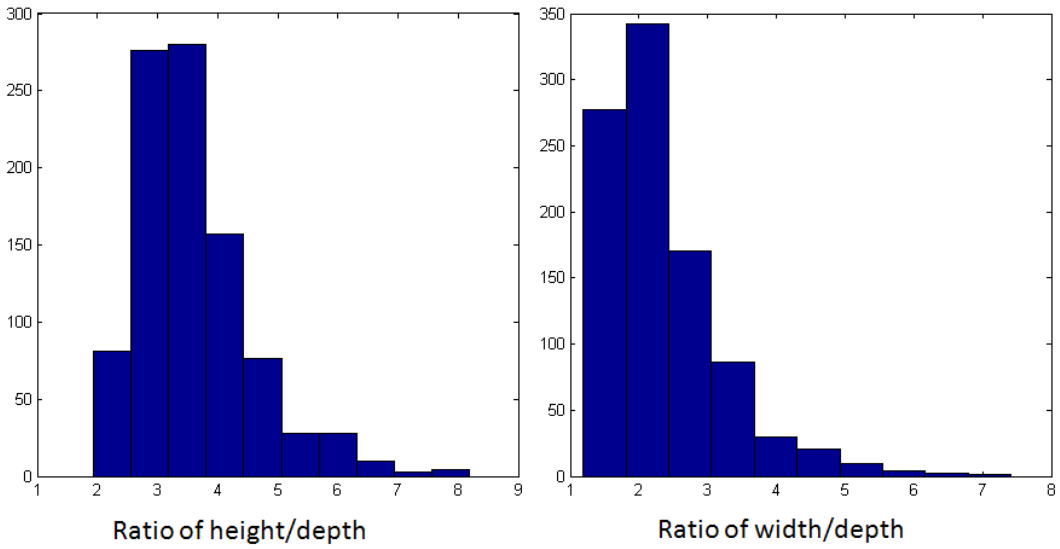


Figure 3.7: *Histogram of the ratio between height, width and depth at nose tip cropped by a sphere* $r = 25mm$*(943 faces of* 275 *individuals).*

3.2.3 Summary

So far, we have proposed two 3D surface descriptors. Theoretically, MCSAMD has a lower cost of computation - $O(nm)$ than MSSAMD - $O(n^2)$. Moreover, when using most 3D data acquisition devices (for example: the structured light sensor Minolta Vivid 900/910 series used in FRGC database), the depth data

are captured with a structured grid order. Some particular shapes, for instance, the shape at a high slope point may cause the distance between neighbouring points to be too large to be included in one of the shells of MSSAMD. Thus, the MSSAMD will lose the ability to describe the information of shape at this place, while MCSAMD makes sure that all neighbouring points are included. This may cause differences in accuracy of the feature localization. However, since MCSAMD depends on the structure order, the same shape with different orientations may result in slight differences in MCSAMD. On the contrary, the MSSAMD is an orientation/pose invariant descriptor. It is difficult to judge which one is better at this stage. Therefore, two descriptors are both used and evaluated in the feature localization experiments.

3.3 k-Nearest Neighbour AURA Algorithm

Facial features can be considered as small pieces of surface. Those small pieces of surface can be described by 3D shape/surface descriptors introduced in previous section. To localize a facial feature, the shape descriptor of a feature has to be selected as a standard model. The most similar shape within a face to the standard model is the most likely position of this facial feature. A face point-cloud may contain thousands of points and a face database usually consists of thousands of faces. Thus, a high effective pattern storage and pattern retrieval method is required. In this chapter, we use a binary neutral network technique (k-Nearest Neighbour AURA algorithm) to measure the similarity between the query shape and the standard feature model.

3.3.1 AURA

Advanced Uncertain Reasoning Architecture (AURA) is a set of methods based on binary neural networks in the form of correlation matrix memories (CMMs) for high performance pattern matching [10]. Correlation Matrix Memories (CMMs) are a form of static associative memories. Kohonen [54]first introduced the idea of correlation matrix memories in 1972 and made the pioneering contribution together with Anderson [8]. AURA has two ways to implement a neutral network: software and hardware. Implementation of AURA on hardware can significantly increase the speed of pattern recognition. In this thesis, we only use the AURA in software.

CMMs learn and store the associations between input patterns P and outputs O, which have to be transformed to a binary vector. The input and output patterns are involved in the training of an initially empty binary matrix M. During training, the values within M are only changed to '1' where both input and output vectors are set according to the Hebbian learning introduced in 1949 [40]. The training of M is presented as the following equation.

$$M = \bigvee P^T O \tag{3.8}$$

P: input pattern (a row vector of binary elements); O: output pattern; M: Correlation Matrix memory; $\bigvee$ is logical OR. Figure 3.8 shows an example of CMM training process.

After training, the recall operation returns a summed integer output vector V, then can be thresholded to be a binary vector. If I is the input vector for recall operation, then (following equation):

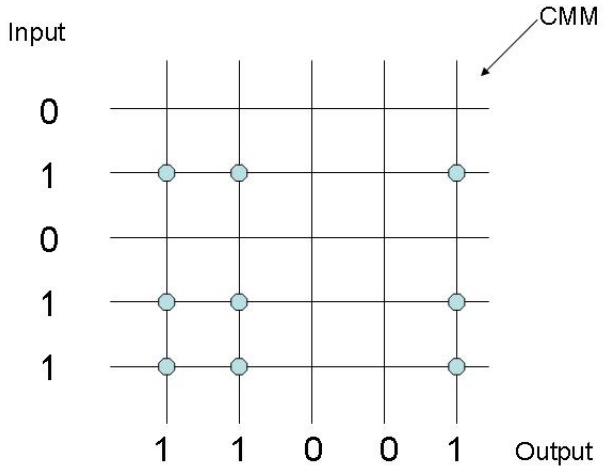


Figure 3.8: *Example of training a CMM. When both of the bit of the input and output vectors are '1', a connection of corresponding position in matrix will be set.*

$$V = MI^T \tag{3.9}$$

The most important characteristic of CMM based systems is that each training/learning operation is quite simple only requiring the binary encoding and bits setting of a binary matrix. Training time for very large dataset is dramatically reduced in comparison to other networks such as MLPs which need to train all other patterns at the same time when the new associations are trained.

In order to apply AURA technique, input patterns have to be quantized and converted into binary values. The simplest way to transform decimal values into binary values is to divide the possible range of the decimal value of an attribute into several parts called bins, then a binary bit is set to '1' on the basis of which bin the actual value belongs to.

In this thesis, we chose 40 3D faces in the Spring2003 subset of the FRGC database as the training group. Nose tips of section one have been manually marked by Romero et al. [77]. Thus, by using the MCSAMD method introduced in section 2, the input pattern of the nose tip can be divided into $2 \times n$ attributes (n: the number of grid circles). The ranges of each attribute are divided into ten bins. The decimal values of each attribute is converted into binary value depending on which bin a decimal value belongs to.

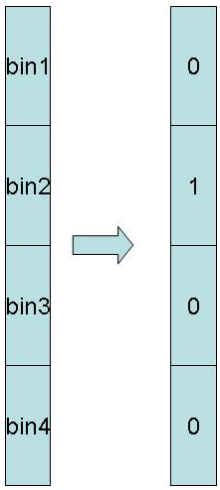


Figure 3.9: *Convert a decimal value into a binary value. When the decimal value is located in bin2, then responding bit of the binary vector will be set to '1'.*

For example, if the range of an attribute decimal value is from '1' to '11' and there are ten bins with same width of '1', a value of '4.5' is located in the bin of '4' to '5' which is the second left bin. Thus, the binary value of a decimal value '4.5' will be '0001000000'. Figure 3.9 shows the decimal - binary conversion. The width of a bin can be decided according to the distribution of

data. In this thesis, we simply choose the bin width by using the range of data (equation 3.10).

$$WIDTH_{bins} = \frac{max(value) - min(value)}{n} \tag{3.10}$$

After all the values of the four attributes have been converted into four 10-bit binary values, an input vector can be generated by concatenating all binary attributes together as shown in following Figure 3.10.

Figure 3.10: *Several attributes combine to be an input vector.*

CMM necessitates both input and output vectors. In this system, the training process stores the binary attributes value into a column of the matrix. Therefore, the output vector is designed as the sequence number of the faces in the training group (as shown in figure 3.11).

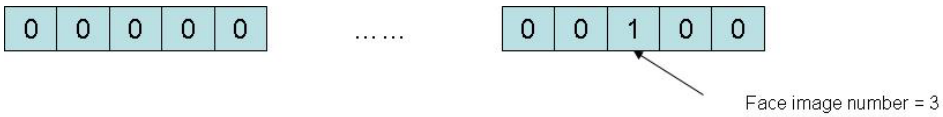


Figure 3.11: *Output vector represents the sequence of training faces.*

As shown in Figure 3.12, the training process is to store the nose tips one by one until all the training faces have been saved in the matrix.

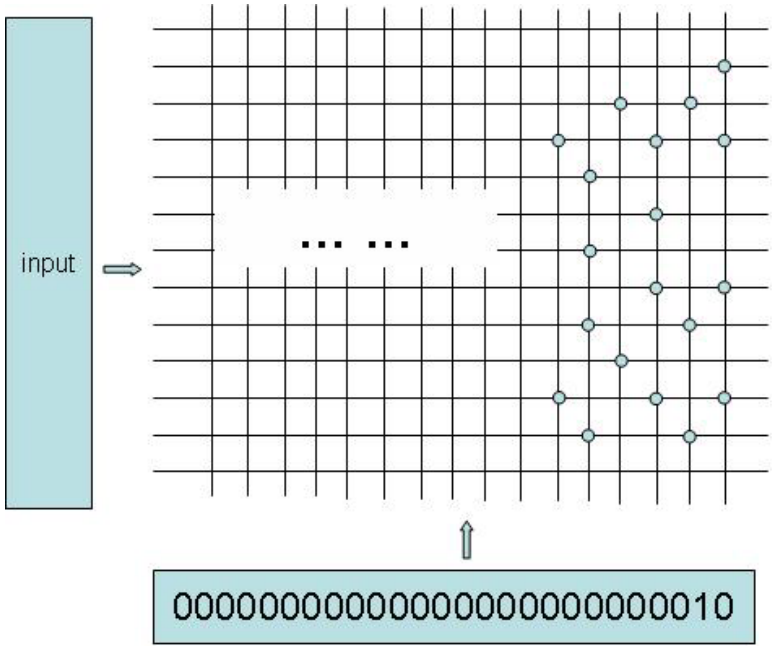


Figure 3.12: *Store the each image into a column one by one.*

3.3.2 AURA matching by using k-Nearest Neighbour algorithm

In the recall or query phase, the query pattern is measured and then feature attributes are generated. In the way same as the encoding procedure of the training images, a binary query input vector is produced. However, a difficulty of the quantization method is the boundary effect. Since there are clear boundaries between bins, a decimal value will only belong to one bin. Thus, the distance between two values within the same bin may be greater than the distance of two values in two neighbouring bins. For example, two boundaries are set at '2.00' and '4.00'. '2.01' belongs to the same bin of '3.99'. However, it is clear that '2.01' is much closer to '1.99' which is in the prior bin than the gap from '2.01' to '3.99'. In order to compensate for that situation, Hodge et al. [44] developed a binary Neural k-Nearest Neighbour technique called Inte-

ger Pyramid in 2005.

The input attributes are concatenated to form the input vector, with one bit set per attribute. This is used during the training phase to store data in the CMM. However, during recall the Integer Pyramid technique replaces the single bits set in the query vector, each with a 'triangular kernel' of integer values arranged so that the maximum value of a kernel is located where the set bit was, and adjacent zero bits are replaced with smaller integers, decreasing uniformly. This vector of integers then forms the input to the CMM, with the response V calculated in the same way as before.

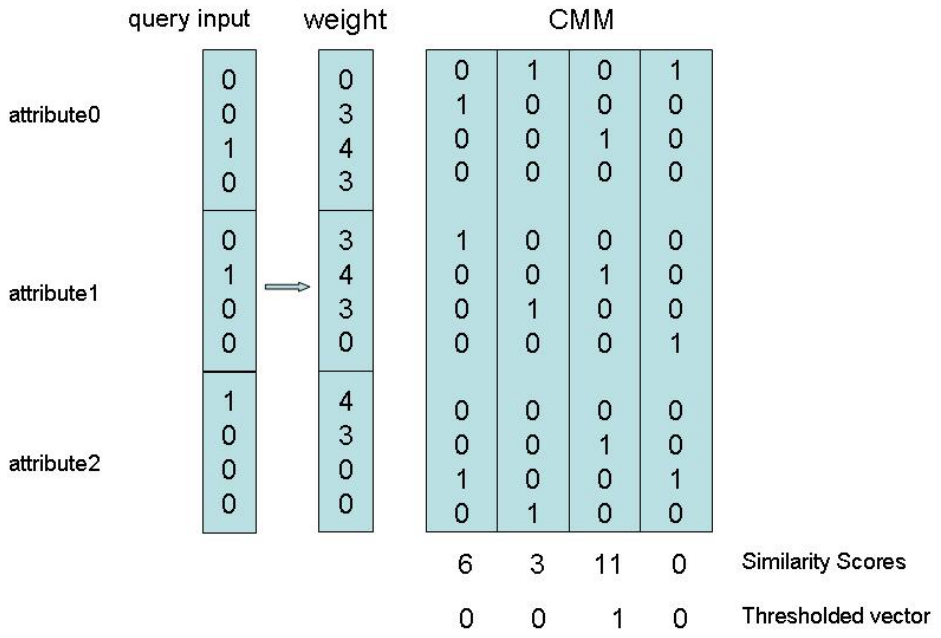


Figure 3.13: *An example of CMM recall with kernel weighted inputs.*

This use of kernels gives a maximum value in V for the stored vector that has been most closely corresponding to the query vector. Vectors that do not match exactly will have a reduced but non-zero response to each query bit.

This gives a more gradual decrease in response for non-matching vector than in the original CMM application. Knowing what the maximum response should be, we convert the reduction in response to a vector of 'distances' of the query from the stored vectors. With the triangular kernel described, the distance approximates the quantized City Block Distance. An example of this use of kernels is shown in figure 3.13.

The Integer Pyramid technique was later improved using a parabolic kernel [44] to approximate the quantized squared Euclidean distance. For one stored vector, the distance is:

$$d_E^2 = \sum_{\forall f} (x_f - x'_f)^2 \tag{3.11}$$

where d_E^2 is the squared Euclidean distance, x_f is the query attribute value and x'_f is the stored value for attribute f.

To calculate this distance using a CMM, the parabolic kernel weight values are calculated as in the equation below. For the attribute f and bin k, with the original set bin in bin t:

$$W_{f,k} = (\frac{n^*}{2})^2 - (t-k)^2 \alpha_f \tag{3.12}$$

$$\alpha_f = \frac{n^{*2}}{n_f^2}$$

where n^* is the maximum number of bins for any attribute and n_f is the number of bins for the attribute f. a_f is to ensure the spread of the kernel for all attributes within the CMM input vector. Figure 3.14 shows the parabolic shape weight values.

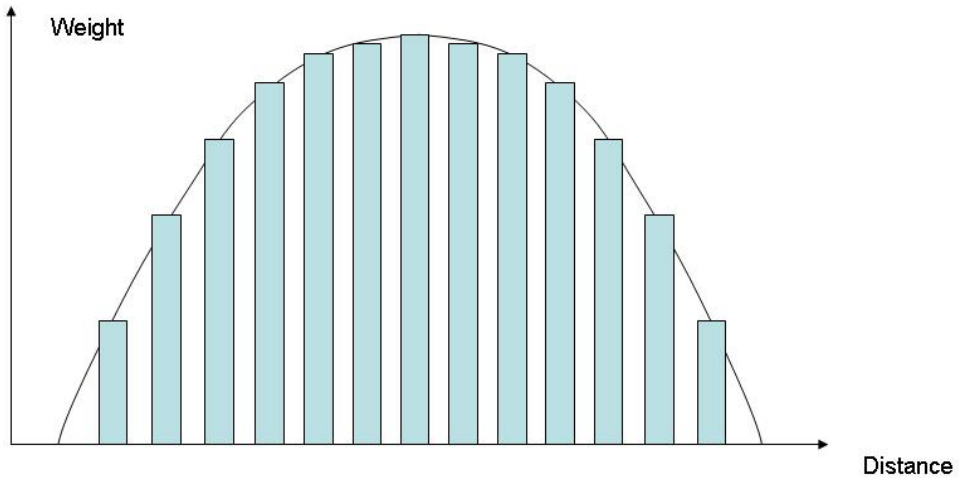


Figure 3.14: *The weight values of the CMMs are set to be analogous to parabolic shape which describe the distance from the central bin.*

By using the parabolic kernel Integer Pyramid technique, the output V contains scores ranked by Euclidean distance. These can be used as a similarity score vector for each query, so $V = \{v_1, v_2, ..., v_p\}$ is the similarity with each of the training nose tips. $max(V)$ tells the level of similarity that a query pattern has to at least one nose tip of the training group.

3.4 Nose tip localization hierarchical methodology

In [77], Romero et al. manually marked eleven facial features including nose tip and eye corners. We place those nose tip landmarks as the center of a grid with two different sizes (empirically choose the fifth circle(9×9) and the ninth circle(17×17)) to generate the MCSAMD attributes. Each point P has four

attributes - the mean and standard derivation of θ within 17×17 and 9×9 grid. The reason why choose two circles is to verify the benefit that extra information from the second circle provides.

In order to create a MSSAMD descriptor suitable for nose tip detection, the maximum radius of the farthest sphere is defined as $25mm$ simply because it is the approximately range from a nose tip to its edges. Using different width of a shell and the number of shells can change the ability of a MSSAMD to describe a piece of 3D shape. In this thesis, we simply used $5mm$ to be the width of a shell because we have to make sure there are enough points existing in every shell area. As a result, there are totally five shells.

By implementing the binary encoding method introduced in section 3, those attributes in MSSAMD or MCSAMD are converted into binary vectors then stored into the CMM. After the training process, the attributes of the points of the target faces are also calculated and encoded with AURA k-NN weights.

We define the three following steps to reduce the number of candidate points for the nose tip in a particular image:

Step one: For a point P_i, the attributes of MCSAMD or MSSAMD are matched with the features stored in the AURA. By using a k-NN AURA matching algorithm, a similarity score vector V is generated. V contains the similarity scores to all features from different subjects stored in AURA. The highest similarity score $S = max(V)$ is chosen as the final similarity score for this point P_i. Then by simply defining a threshold T_{nose}, any candidate with a similarity score below T_{nose} is deleted from the candidates list. This step can significantly

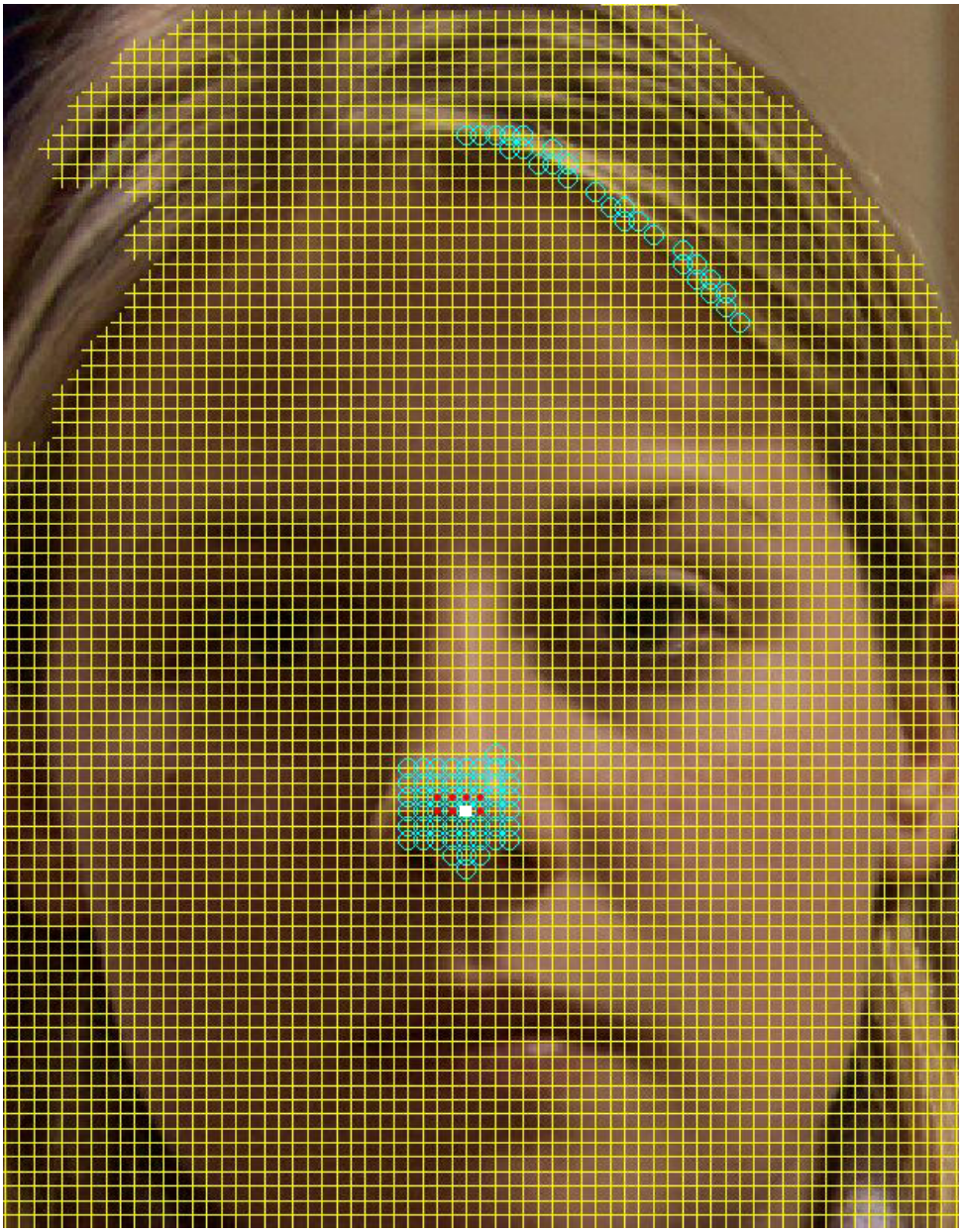


Figure 3.15: *Yellow grids represents the projection of 3D point to 2D space; blue circles means the candidates using similarity score filter; red points are results of applying density filter, the white square is the final selection of nose tip.*

narrow down the range of candidate points.

Step two: There are usually some other points left in the candidate list such as those in the hair, clothes or chin areas that cannot be eliminated in step one. However, most of those exceptional points are scattered and the points around the actual nose tip always get a relatively high similarity score. Therefore, we can locate the correct nose tip cluster by calculating the number of the candidates within a certain range. The cluster with the highest density of candidate points is chosen as the nose tip candidate cluster.

Step three: After the nose tip cluster is selected, the candidate with the highest similarity score inside this cluster is considered as the final choice.

If we implement this methodology in nose tip detection, figure 3.15 shows an example of how a final nose tip selection is made. In figure 3.16 , the work flow of the nose tip localization steps is illustrated.

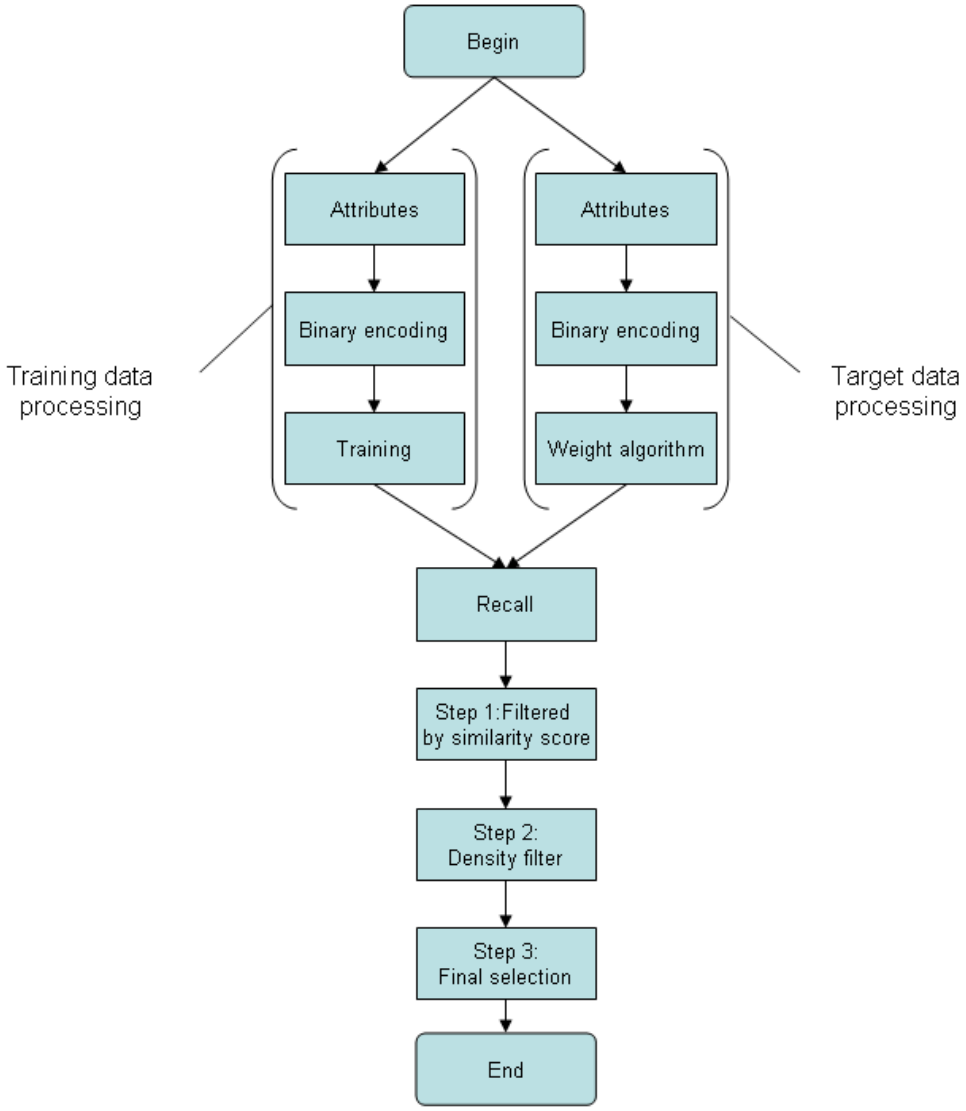


Figure 3.16: *The work flow of nose tip localization.*

3.5 Medial Canthi Detection

In the training set of MCSAMD, not only are the four MCSAMD attributes of each nose tip stored in the CMM of AURA system, but the corresponding

MCSAMD attributes of the two medial Canthi(inside eye corners) are also encoded into binary vectors and stored in the CMM. We also know the distances between the nose tip and each of the eye corners are limited within some ranges. D_{ne1} is the distance between the nose tip and the left eye inside corner. D_{ne2} is the distance between the nose tip and another eye corner. D_{ee} represents the distance between two eye corners. Each of those three distances within the training set can be used as a limitation. D_{ne1} and D_{ne2} should approximately equal each other and the ratio of $(D_{ne1} + D_{ne2})/D_{ee}$ should be in some limited range. Those relationships can be converted into binary input attributes and stored in CMM. In the eye corner identification, we choose two neighbouring grid sizes $N \times N(N = 9, 7)$.

As with the nose tip localization, we evaluate the similarity score to the stored eye corners of each neighbouring grid centered at a point P by using MCSAMD and AURA k-NN techniques. After deleting the points with lower similarity scores than a threshold T_{eye}, a number of candidate points are considered as potential eye corner points. However, only this filter is not enough. The potential eye corner points are still mixed with some noise points.

The nose tip, left inside eye corner and right inside eye corner forms a triangle. The number of triangles formed by potential eye corners and nose tips is very large. Thus, we continue to reduce the number of candidates by using the relationship between nose and eyes stored in CMM. Another score S_{rel} is designed to represent the degree that a combination of the nose tip and two eye corners is similar to the relationships of trained combinations. The candidate with the highest S_{rel} is our final selection(red points shown in figure 3.17). However, in some 3D faces, there is a crevice near the eyebrow which is not easy to fix

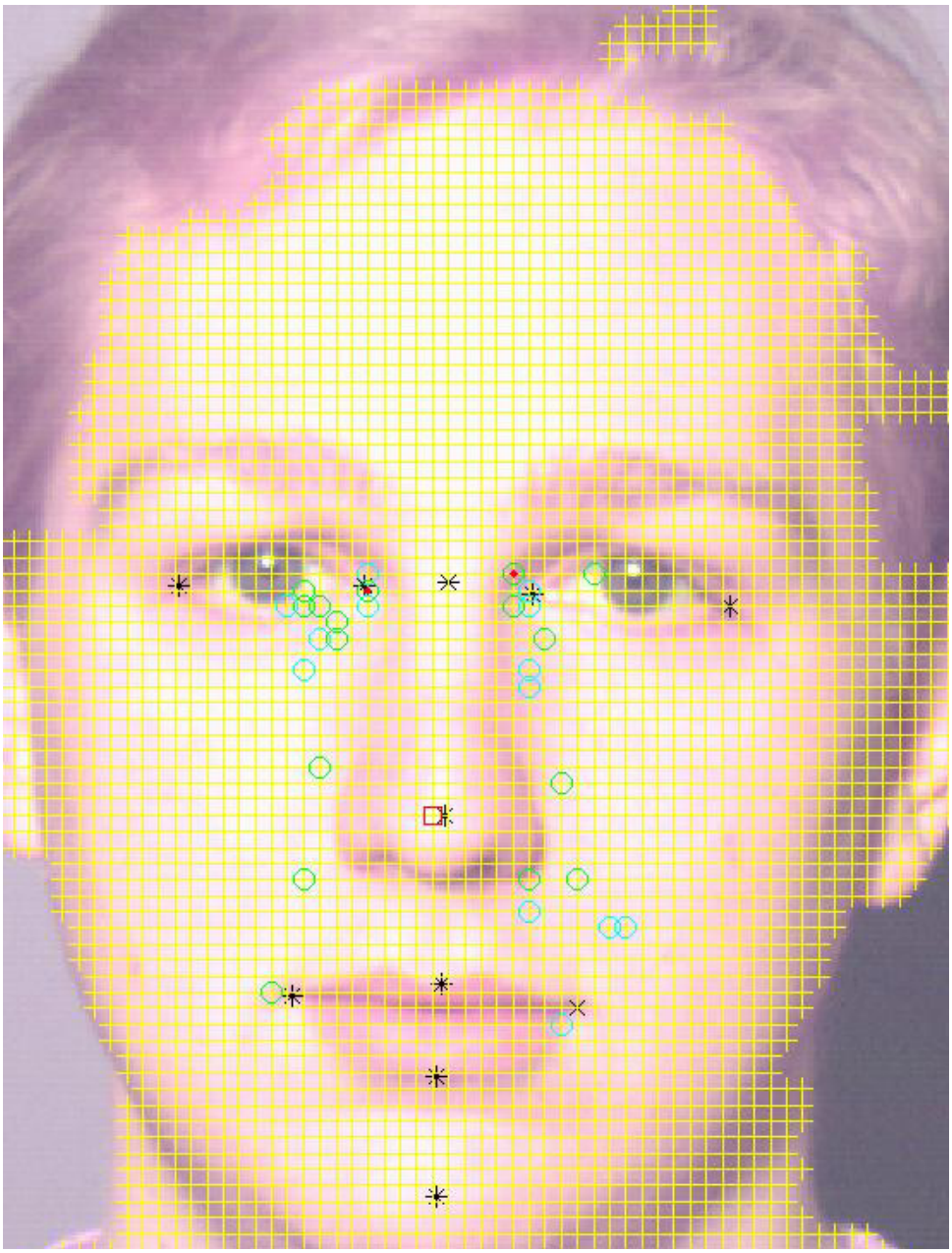


Figure 3.17: *Yellow grid represents the projection of 3D points to 2D space; blue/green circles shows the eye corner candidates, using similarity score filter; red points are final choices for eye corners, the red square is the final selection of nose tip; '*' symbols represent the manually selected landmarks.*

in the preprocessing steps. That could cause the wrong selection of the eye corner(an example is shown in figure 3.18).

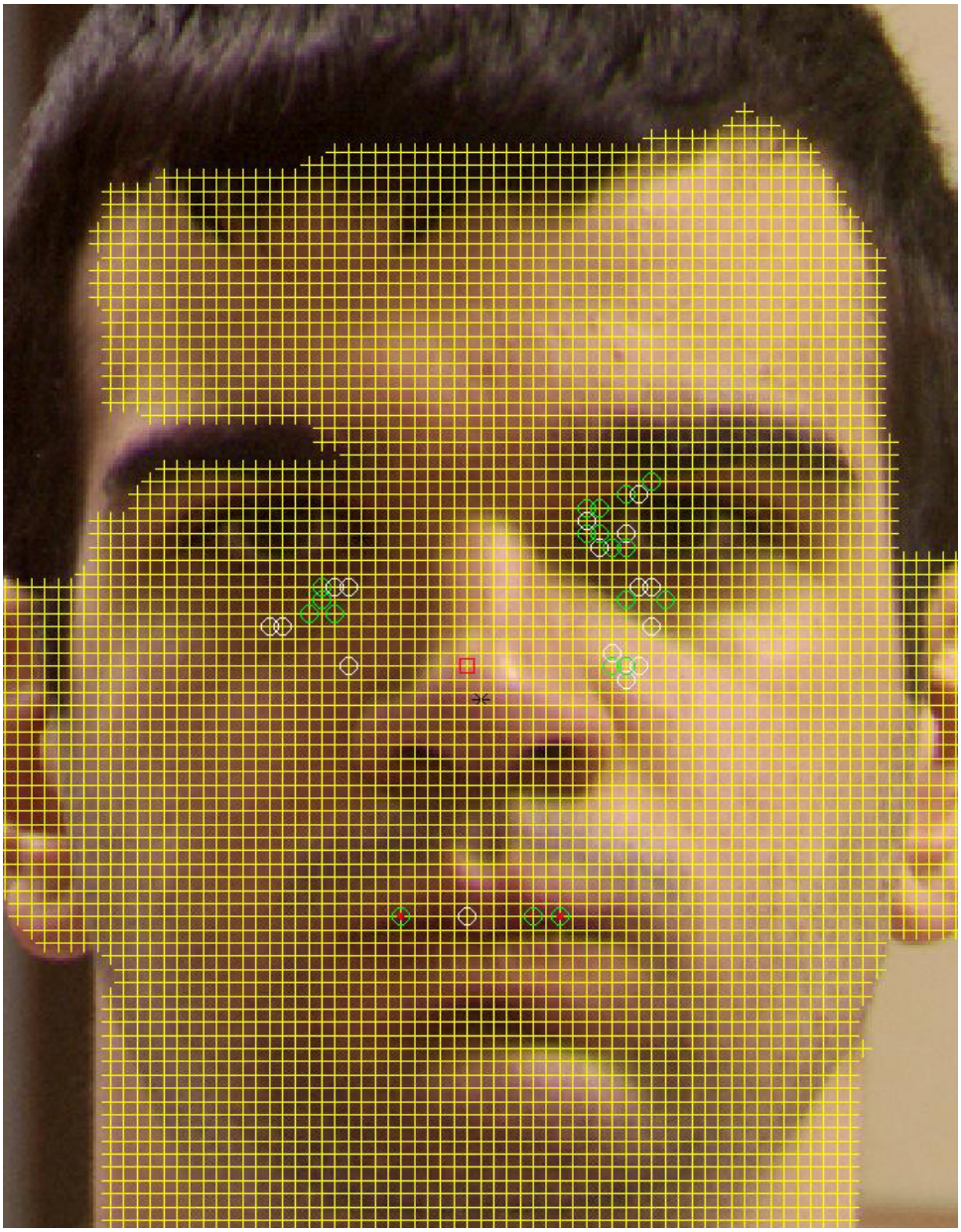


Figure 3.18: *The crevice near the eyebrow is so close to the real eye corner that the selection of the potential eye corner is seriously affected.*

3.6 Experimental results

3.6.1 Database

In this chapter, the FRGC dataset is chosen as the experimental database. The FRGC 3D dataset has three subsets. The Spring2003 subset is the 3D training set that contains 3D scans, and controlled and uncontrolled still images from 943 subject sessions. In Fall2003 and Spring2004 subsets which are designed as target subsets, there are 4,007 subject sessions of 466 subjects. Each subject session has a 3D scan file containing 3D points and a 2D still image file representing texture information.

The original size file is the high resolution face image. The resolution of faces in the FRGC dataset is 640×480. In order to reduce the cost of computation in data processing, we resize the 3D channel file to a smaller size. We choose 160×120 as the downsized resolution because it is able to keep the balance between details and cost of computation. It has enough details to evaluate the localization of facial features. The resized 3D files are smoothed to delete the spikes and to fill in the unexpected holes by using a similar technique to that proposed by Mian et al. [63]. Firstly, we remove spikes from the face surface by locating outlier points. For a particular point p in the FRGC database, it has eight connected neighbouring points as shown in figure 3.19. Any point whose distance(red line in figure 3.19) is greater than a certain threshold d from any of its neighbouring points will be considered as a spike point. d is defined using $d = \mu + 0.6\sigma$, where μ is the mean distance between neighbouring points(green lines in figure 3.19) and σ is the standard deviation. The holes caused by the removal of spike points can be filled by using cubic interpolation.

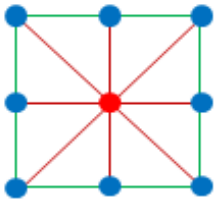


Figure 3.19: *A point p with its eight connected neighbouring points. Green lines are distances between neighbouring points. Red lines are distances from p to its neighbouring points.*

Most of the faces in this subset are captured under controlled illumination with neutral expressions. 40 3D faces are selected from the Spring2003 subset as the training set. Those 40 faces are from 40 individuals including different races, genders and numbers of points. 4007 faces from the Fall2003 subset and the Spring2004 subset are used as test groups. Since there are 139 faces with very poor 2D-3D corresponding, 3868 faces having good 2D-3D correspondence are selected to more precisely evaluate the performance.

Since there are all neutral expression faces in the training group and the target group includes faces with expressions, expression variations may affect the feature localization . In order to evaluate the effect of expression to feature localization, we separated the FRGC v2 dataset(Fall2003 subset and the Spring2004 subset) into two groups: neutral faces and faces with expressions according to the selections used in [74]. The first group contains 2128 neutral faces and the second group has 1740 faces with expressions. The details of those subsets are listed in table 3.1.

Groups	Descriptor	Number of face images
I	All faces	4007
II	Good corresponding between 2D and 3D channels	3868
III	Neutral faces	2128
IV	Non-neutral faces	1740

Table 3.1: *Different selections of face subsets.*

3.6.2 Nose tip and eye-corners localization results using MCSAMD

Thanks to the work of benchmark datasets made by Romero et al [77], we can use those landmarks to evaluate our experimental results. We used the methodology introduced in the previous section based on MCSAMD to localize nose tip and two eye-corners. Figure 3.20 shows how the detection rates of nose tip and two eye-corners changes as the allowable error distance is increased. The localization results of those three features are also shown in a histogram in Figure 3.21 using the following standards:

$Good :\leq 12mm$

$Poor :\geq 12mm \& \leq 24mm$

$Failure :\geq 24mm$

$20mm$ is the approximate width of the nose and the error distance that we are using is in 3D, so we choose $24mm$ as a threshold to determine the success or failure in feature localization. Any error distance larger than this value will be considered as a failure. An error distance below the half of this value is considered as a successful detection.

Since the landmarks are marked on 2D faces and there are some poor 2D-3D correspondences in the FRGC database, the error distance does not completely represent the accuracy of the localization. Therefore, we verified the FRGC database manually to remove the faces with bad correspondence in 2D and 3D. Figures 3.22 and 3.23 show the results on the good 2D-3D correspondence dataset by applying MCSAMD on nose tip and eye corners localization. Although faces in Fall2003 and Spring2004 subsets present facial expression variations, over 99.69% of nose tips are successfully located. Identification rates of left and right eye corners are 96.41% and 96.80% respectively.

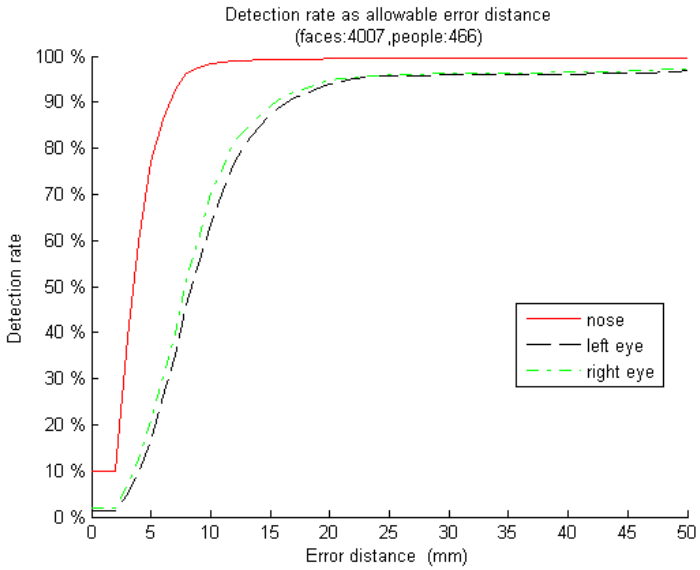


Figure 3.20: *Cumulative Curves of error distance for the feature identification on Fall2003 and Spring2004 subsets before 2D-3D correspondence verification.*

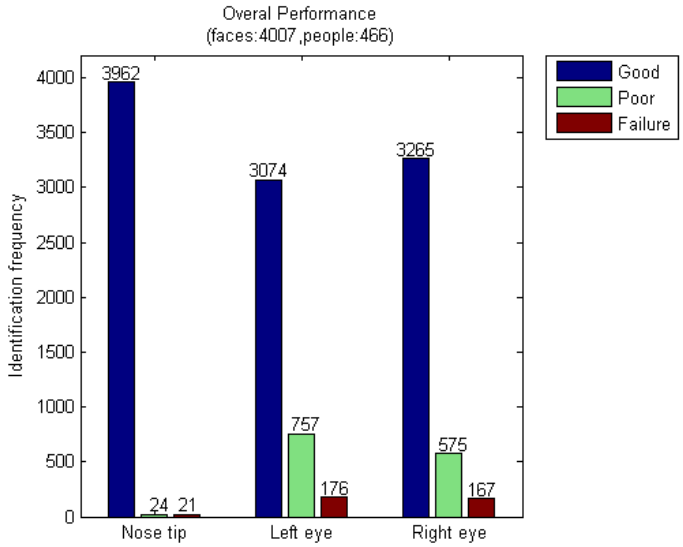


Figure 3.21: *Histogram of the identification frequency on Fall2003 and Spring2004 subsets before 2D-3D correspondence verification.*

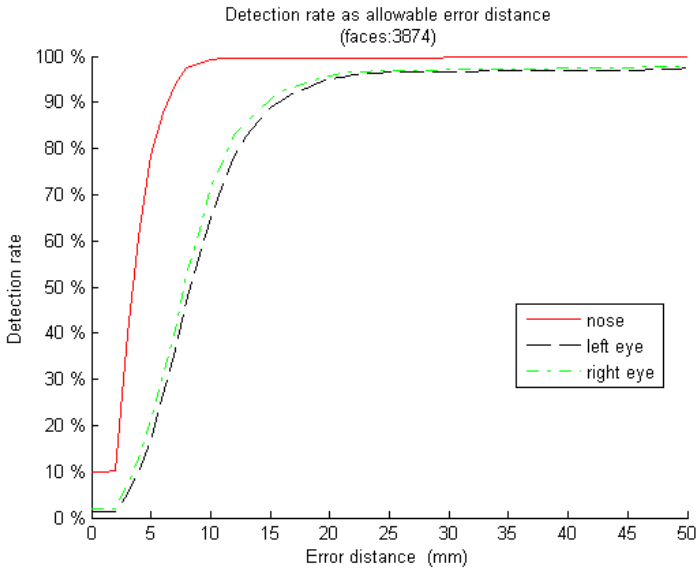


Figure 3.22: *Cumulative Curves of error distance curve for the feature identification on Fall2003 and Spring2004 subsets after 2D-3D correspondence verification.*

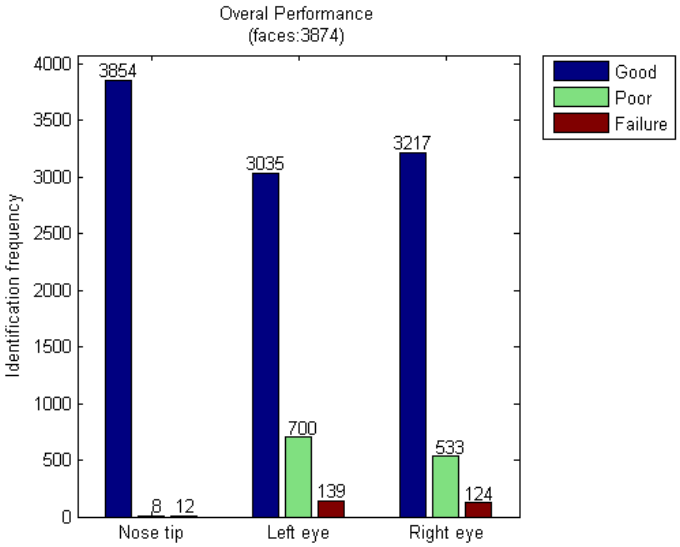


Figure 3.23: *Histogram of the identification frequency on Fall2003 and Spring2004 subsets after 2D-3D correspondence verification.*

3.6.3 Nose tip localization results comparison between MCSAMD and MSSAMD

The figure 3.24 shows the comparison of MCSAMD and MSSAMD in the nose tip localization. Although the system using MCSAMD has a little bit higher accuracy in good and poor(acceptable) detections than the system using MSSAMD, the MSSAMD system has fewer failure detections shown in the histogram of figure 3.24. When compared with ground truth data, the mean error distance of MCSAMD and MSSAMD are 3.8574mm and 4.7174mm respectively. When the faces with bad 2D-3D correspondence are removed from the experimental list, the number of detection failure using MSSAMD becomes zero while the MCSAMD system still has eleven detection failure shown in figure 3.25. Table 3.2 summaries the differences between two descriptors.

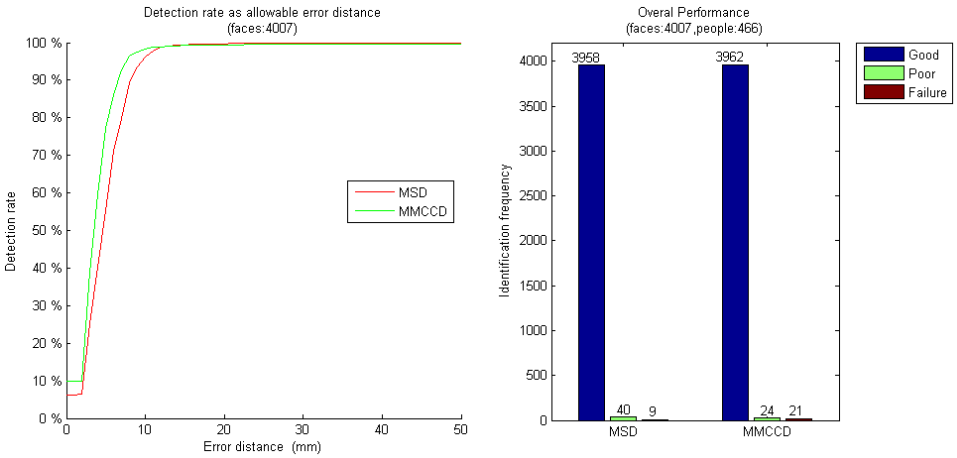


Figure 3.24: *Results comparison between MSSAMD and MCSAMD with all faces in FRGC v2 dataset.*

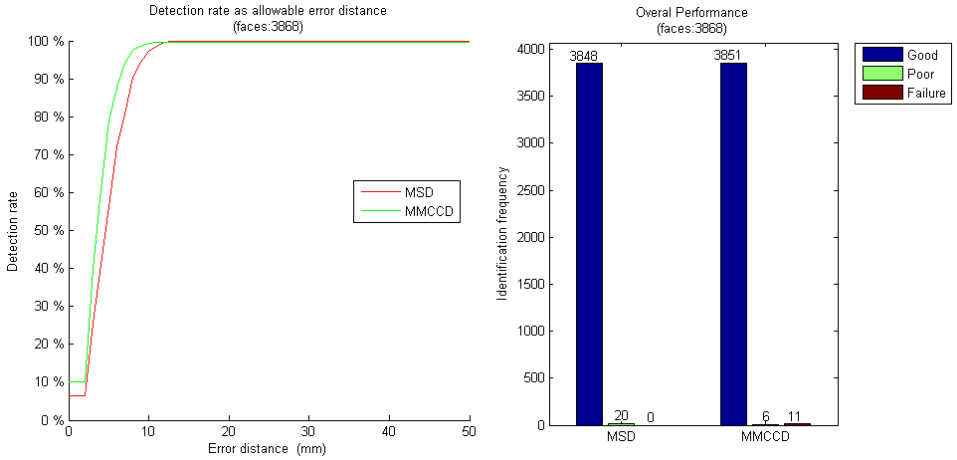


Figure 3.25: *Results comparison between MSSAMD and MCSAMD with good 2D-3D correspondence faces.*

	MSSAMD	MCSAMD
Detection rate on FRGC v2	99.78%	99.48%
Detection rate on FRGC v2 with good 2D-3D corresponding	100%	99.72%
Average error distance	$4.7174mm$	$3.8574mm$
Orientation invariant	Yes	Partial
Cost of computation	$O(n^2)$	$O(nm)(m \ll n)$

Table 3.2: *Comparison between MSSAMD and MCSAMD.*

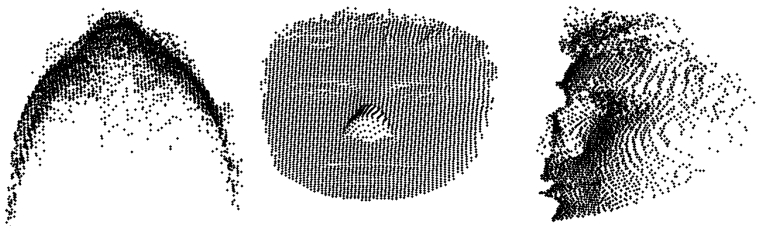


Figure 3.26: *A face without nose.*

In the FRGC v2 database, for some unknown reasons, actually two faces have no noses at all. Figure 3.26 shows an example of one of them. By manual reviewing the results of MSSAMD nose tip detection, in 4007 faces, only those two faces have incorrect nose tip detections. The detection rate of the nose tip localization on FRGC v2 database is actually 99.95%. Furthermore, even in those two noseless faces, the detected position of the nose tip by applying MSSAMD is close to the nose and the center of the face, shown in figure 3.27.

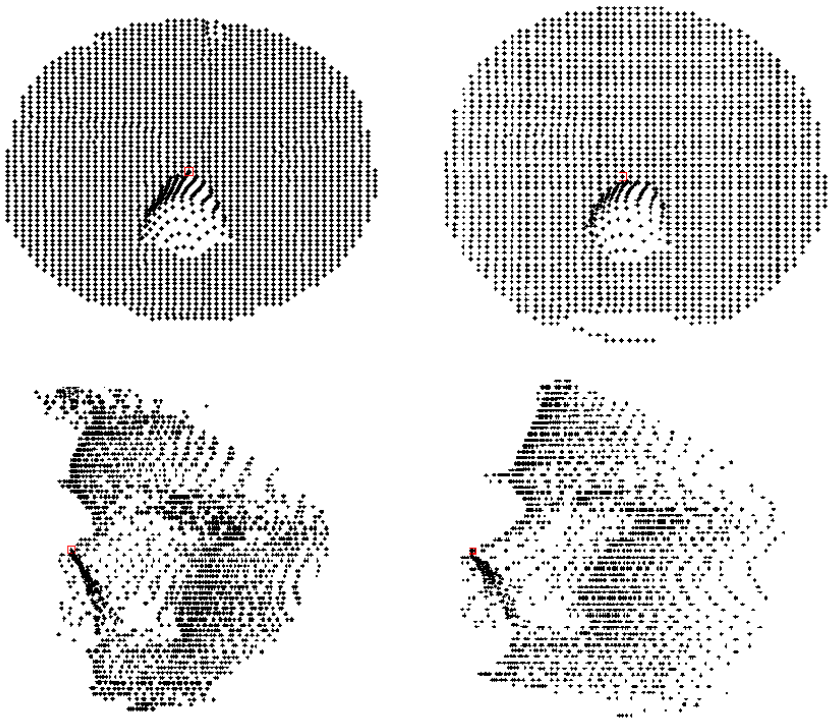


Figure 3.27: *Nose localization on two noseless faces; red squares are the positions of nose tip detected.*

3.6.4 The effect of expression variations on nose tip localization results

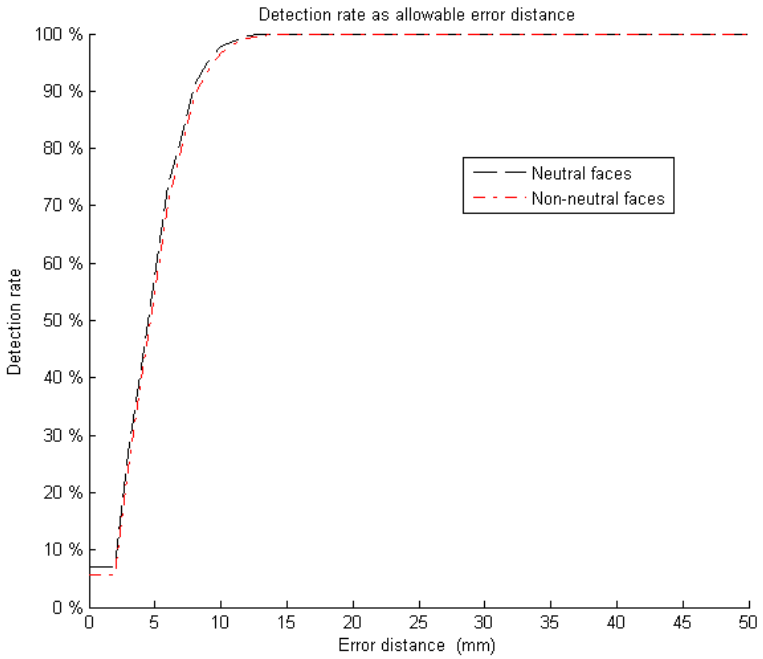


Figure 3.28: *Error distance curves for the feature identification on neutral and non-neutral faces.*

Expression variations could lower the performance of the feature localization because only neutral faces are used in training process. Figure 3.28 and 3.29 show the performance of the nose tip localization on neutral faces and non-neutral faces. We can see from these figures that the effect of expression variations is very slight. The performance of neutral faces and non-neutral faces are very close to each other. One reason why expression variations do not cause a drop of performance is because the nose is a facial feature which does not vary when the expression changes.

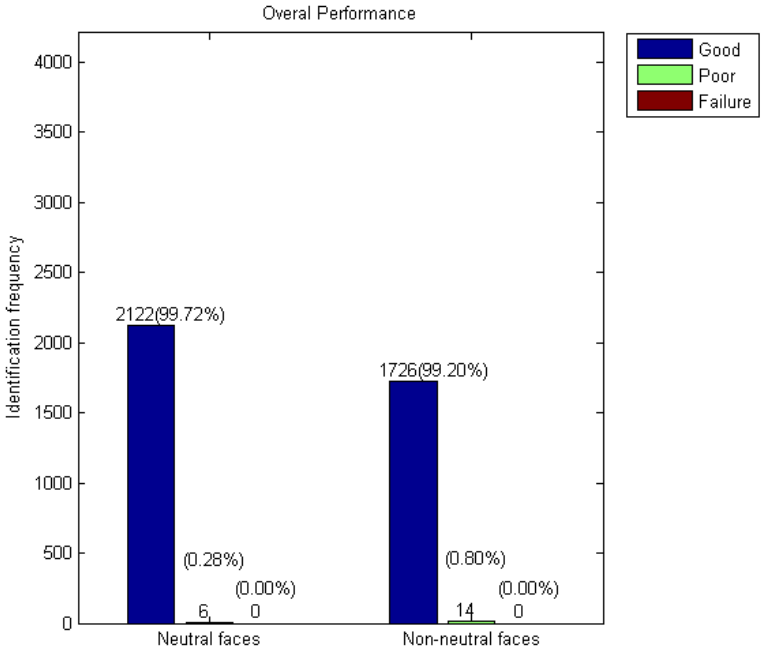


Figure 3.29: *Histogram of the identification frequency on neutral and non-neutral faces.*

3.6.5 Comparison with state-of-the-art techniques

Unlike some techniques making use of the texture information in the 2D face detection, this approach is a pure 3D shape analysis which is naturally invariant to illumination variations. It is also an orientation-invariant method. In order to compare with approaches using all faces in FRGC database including v1 and v2 datasets, the nose tip localization is also implemented on FRGC v1 database, the nose tip detection rate is 100% on 943 faces. Thus the nose tip detection of whole FRGC database is 99.96%(2 failures out of 4950). Compared with results using other state-of-the-art techniques, the MSSAMD achieved the highest detection rate of the nose tip localization, shown in table 3.3.

	MSSAMD	Segundo [80]	Faltemier [35]	Pears [70]	Mian [63]
Detection rate on FRGC v2 (4007 faces)	99.95%	99.95%	98.20%	n/a	n/a
Detection rate on FRGC v1&v2 (4950 faces)	99.96%	n/a	n/a	n/a	98.3%
Detection rate on FRGC in good 2D-3D corresponding	100% (totally 3868 faces)	n/a	n/a	99.92% (totally 3680 faces)	n/a
Compare with ground truth data	Yes	No	No	Yes	No
Orientation invariant	Yes	Partial	Yes	Yes	Partial

Table 3.3: *Details in comparison with state-of-the-art techniques.*

3.7 Conclusions

This chapter presented a method based on two 3D surface descriptors and AURA k-NN algorithm to identify and localize facial features, especially the nose tip. The MCSAMD has slight higher accuracy in nose tip detection, but the MSSAMD got more correct or acceptable detection. For a database with orientation variations and other noise, such as FRGC v2, the MSSAMD is more suited to a face detection system because it is a complete pose-invariant approach and it obtains a zero failure rate in the nose tip localization.

Eye-corner identification is not as good as for nose tips, probably because the eye corner shape is relatively more complex than the nose tip. Moreover, unlike nose, the eye corners of faces from different individuals have less similar shapes. That increases variations of eye corner. When collecting the training data, the unstable manual selection could also make the situation more difficult.

A 99.95% identification rate of the nose tip localization in a large dataset(FRGC v2) with expression variations demonstrated the robustness and effectiveness of this method. If we use the results of the nose tip localization using MSSAMD to detect and crop the main face area, all faces can be used in the following task. Even the noseless faces still can be used because the nose tip position which is automatically detected is very close to the actual position of the nose tip. That means there is no loss in the nose detection/face detection stage. It builds a good foundation for face detection, segmentation and further recognition.

Chapter 4

Face Localization and Alignment

4.1 Introduction

In order to implement 3D face recognition, firstly we need to know where the main face area is, especially when the 3D face surface includes face, hair, clothing and other noise caused by objects surrounding the face. If the main face area can be found, the face area then can be cropped from the original 3D surface to reduce the effect of noise and other non-face factors. A sphere around the nose tip can be defined to crop the face area. Thus we can make use of the results in the nose tip detection in the previous chapter to implement the face detection task. However, even when the main face area is localized, the head orientations of different faces in a large face database vary. The head orientation variations could lower the performance of the face recognition. Thus, an effective face alignment is required to correct the poses of all faces. A face alignment method implemented for the face recognition task is required to handle expression variations and noise situations. Furthermore, faces belong-

ing to the same individual should be in a consistent pose.

Mian et al. [64] used a Principle Component Analysis (PCA) based algorithm to correct the pose variations. Three principle components are used as the x, y and z-coordinates of the point cloud of a face. However the noise (for example hair), surface loss and distortion of a face will affect the performance of this method. Another solution is ICP-based face alignment. Faltemier et al. [35] proposed a method for curvature and shape index based nose tip detection to localize the position of the nose tip and then align the whole input image to a template using the ICP algorithm. Kakadiaris et al. [50] implemented a multistage alignment method including three algorithmic steps: Spin-images based alignment, ICP-based alignment and Simulated Annealing on Z-Buffers alignment. However, both of these approaches used the whole face area during their alignments. The expression variations could affect the results of alignment by using the whole area of the input images. Other ICP-based approaches [60] [92] attempted to solve the expression problems by only using the less malleable face area such as areas around nose and eyes. Although using the least affected areas is theoretically robust to expression variations, it is based on an assumption that the localization of the nose tip is extremely accurate and 100% correct, which is normally difficult to obtain.

In order to provide an accurate 3D face alignment method, especially one that is able to align the faces of the same subject into a consistent form, in this chapter we propose an integrated improved ICP-based face alignment approach to correct 3D face images. The whole face alignment procedure has four phases as following:

1. Crop the main face area by using a sphere with its center at the nose tip which is detected as in the previous chapter.

2. Align all cropped faces according to its PCA coordinates.

3. Make use of the symmetric character of the face to implement the alignment especially along y and z-axis.

4. Align faces to a standard face template by using ICP algorithm to optimize the alignment along x-axis.

4.2 Face localization

A 3D human face is not a rigid body. Emotional variations generate different expressions. People's appearances are different under different facial expressions. That means that the 3D face surface will change. Therefore, we have to find which part of the face will remain rigid under different expressions. That may require localizing other facial features such as mouth, eyes and forehead etc. However, to the best of our knowledge, the best techniques to localize facial features except the nose can not guarantee 100% accuracy. On the other hand, the face region around the nose is the most constant area because there is only one facial action unit related to the nose region [39]. A facial action unit is the basic measurement unit defined in the Facial Action Coding System (FACS), which is a system to categorize human facial expressions, originally developed by Paul Ekman in 1976 [34]. In FACS, all anatomical facial expressions are decomposed into some facial action units. The following table 4.1 shows the number of facial action units related to the major parts of the human face.

	Nose	Forehead	Eyes	Checks	Mouth/Lips	Chin
Facial action units	1	> 2	> 5	> 5	> 15	> 5

Table 4.1: *The number of facial action units related to major parts of human face.*

A possible solution for face localization is to roughly correct the position of the face by applying Principle Component Analysis (PCA). Since we have already acquired the position of the nose tip, we can extract the face region above the nose tip to avoid the expression variations. The next step is to use ICP (Iterative Closest Point) to align the face to a standard position and then to separate the expression-invariant area of the face. However, even after the face localization step, there are still some parts of the hair being cropped into the main face area due to some hair styles. Consequently, a further alignment tuning process is necessary to improve the accuracy of the alignment.

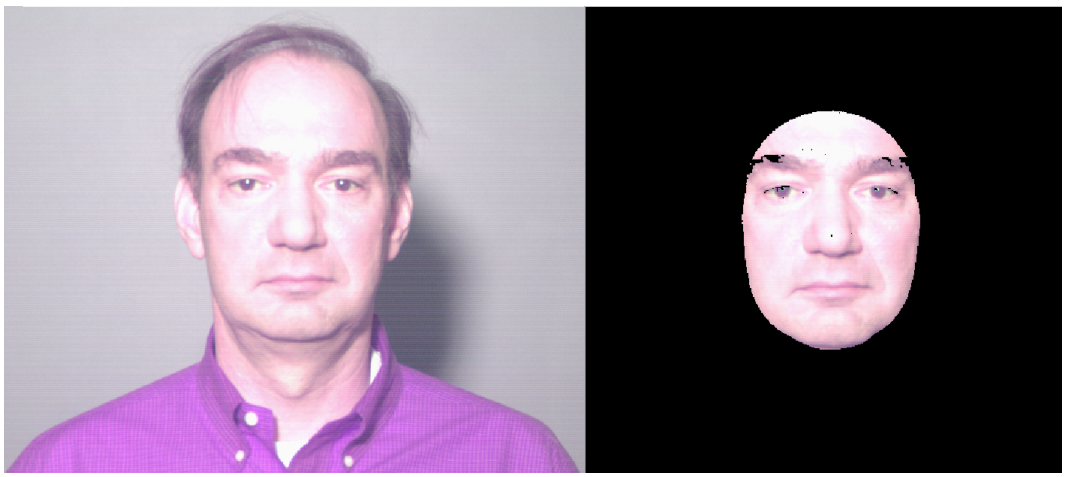


Figure 4.1: *Left figure is the original face; right side is the cropped face using a sphere* $r = 100mm$*; the center of sphere is at the nose tip.*

In the previous chapter, the nose tip has been successfully identified and lo-

calized. The nose tip is in the center of the face. Thus, using the nose tip as the center of a sphere, the main face area can be extracted from the original image. According to many face processing works [64] [74] [35], spheres with radius of $80-100mm$ are used. In this thesis, $100mm$ is selected as the radius of this sphere to crop face in order to keep as much detail as possible. An example is shown in figure 4.1.

4.3 Face pose correction based on Principle Component Analysis

On the basis of results acquired in the section 4.2, the face shape appears as a 3D shape that has the most convex point at its center - the nose tip. The other parts of the face are very close to a cropped piece of barrel surface as shown in figure 4.2. The length of c is shorter than the length of a and b and the length of b is longer than the length of a. That fact has been illuminated by A. Mian et al [64] and L. Zhang et al. [96].

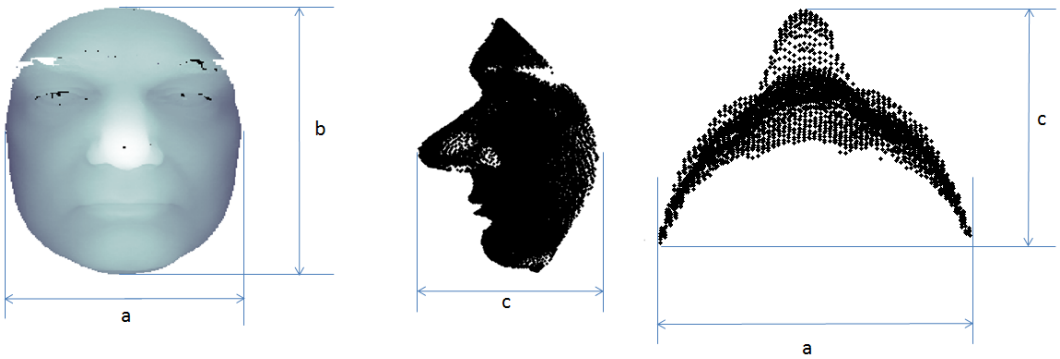


Figure 4.2: *a,b and c are the width, height and depth of the 3D face surface.*

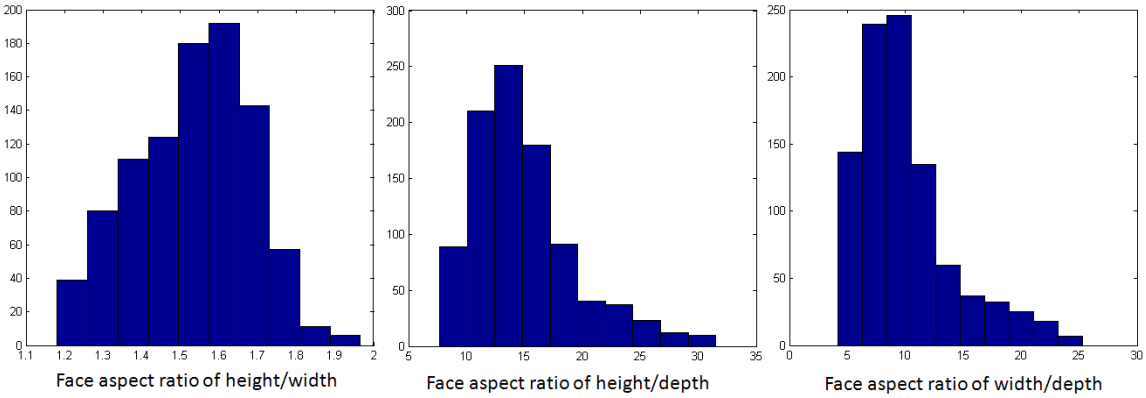


Figure 4.3: *The distribution along the depth direction is the smallest one. The distribution along the height direction is the largest among three directions.*

Thereupon, according to the distribution information of points such as a,b and c, the top three largest principle components can be used as x, y and z coordinates axes. Then the pose of all faces theoretically can be aligned into a consistent coordinate system. Firstly, let $p_i(x_i, y_i, z_i)$ $1 \leq i \leq n$ represent a point within a face surface S, which has n points. Taking m as the mean vector of all p_i:

$$m = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{4.1}$$

Then the covariance matrix C can be given by:

$$C = \frac{1}{n}\sum_{i=1}^{n}(p_i - m)(p_i - m)^T \tag{4.2}$$

By performing PCA on the covariance matrix C, a matrix V of eigenvectors and a diagonal matrix D of eigenvalues are given by:

$$CV = DV \tag{4.3}$$

Then three eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and three corresponding eigenvectors ν_1, ν_2 and ν_3 can be computed. Due to the particular shape of the cropped

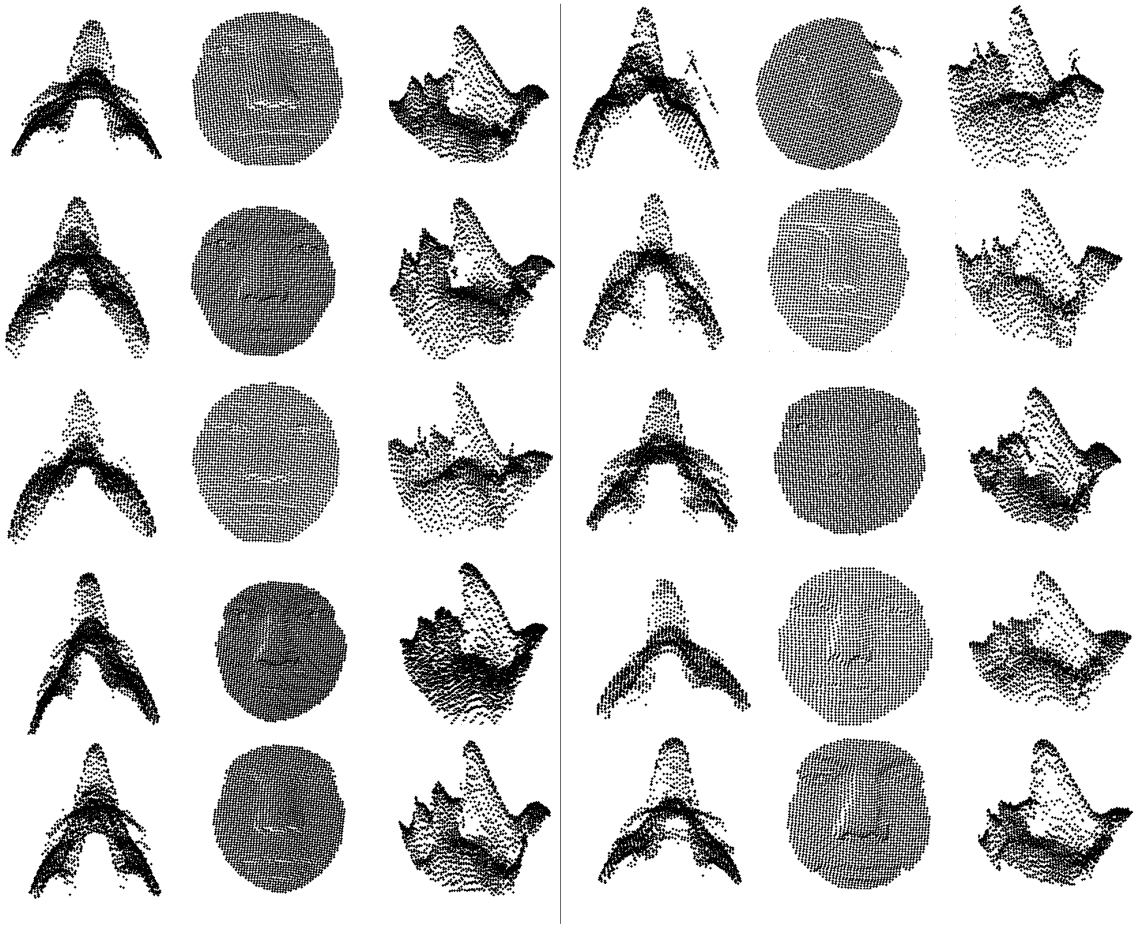


Figure 4.4: *Examples of faces after the PCA alignment.*

face, the smallest distribution of the point cloud of a face is along the normal direction of the face surface. Figure 4.3 shows the histogram of the ratio between height, width and depth. Consequently, the eigenvector ν_3 represents the normal direction and the ν_1 and ν_2 are the vertical and horizontal dimension directions. By means of PCA, the matrix V is also a rotation matrix to convert the coordinates of S to be its principal axes:

$$S_{new} = V(S - m) \tag{4.4}$$

Figure 4.4 shows some faces after the PCA alignment. Most faces are at a

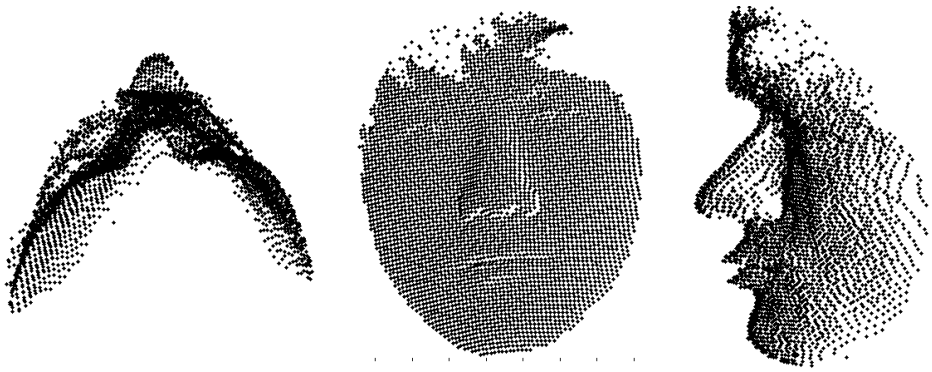


Figure 4.5: *A misalignment example due to hair style.*

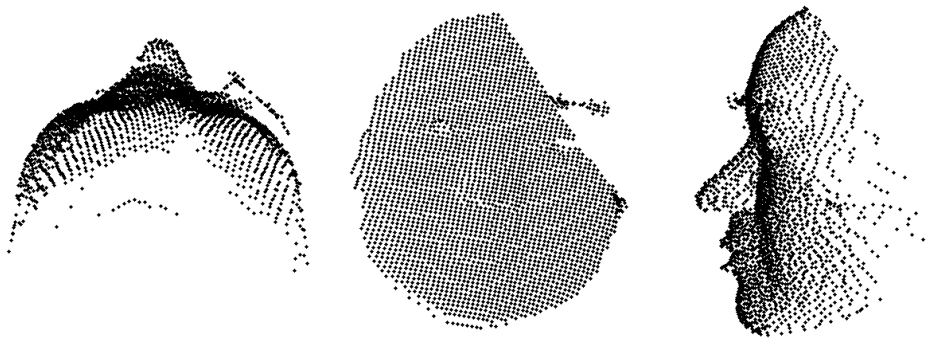


Figure 4.6: *A misalignment example due to surface loss.*

good front view position. However, some faces are not correctly aligned. From those misaligned faces, we can see that the asymmetric shape produced by different hair styles (an example is shown in figure 4.5) is one reason for the misalignment. In some cases (an example is shown in figure 4.6), surface loss at some positions will cause misalignment. Additionally, distortion of the face (an example in figure 4.7) also will affect the accuracy of the face alignment. Due to those failures in PCA alignment, a further alignment method is required to improve the performance.

Figure 4.7: *A misalignment example due to surface distortion.*

4.4 Face alignment based on the symmetry of human face using ICP algorithm

4.4.1 The Iterative Closest Point(ICP) Algorithm

Recently, the ICP algorithm has been used to align the faces by many state-of-the-art face recognition approaches [60] [92] [35]. The iterative closest point algorithm algorithm (ICP) is widely used for geometric alignment of 3D models. ICP is a method to fit a target cloud of points to another cloud of points which constitute a model image. The whole idea of ICP is to minimize the sum of square error between target points and the model points, then estimate an appropriate transformation to align the target points to the model points. Besel et al. [14] proposed the first ICP algorithm and proved that the ICP algorithm always converges monotonically to the nearest local minimum of a mean-square distance metric. The smallest distances between each point in the target image and the points of model image are calculated to form a rotation matrix. This procedure is repeated until the squared error distance of the points of the target image to their closest points in the model image falls below a preset threshold. The complete procedure of ICP algorithm is shown

in figure 4.8.

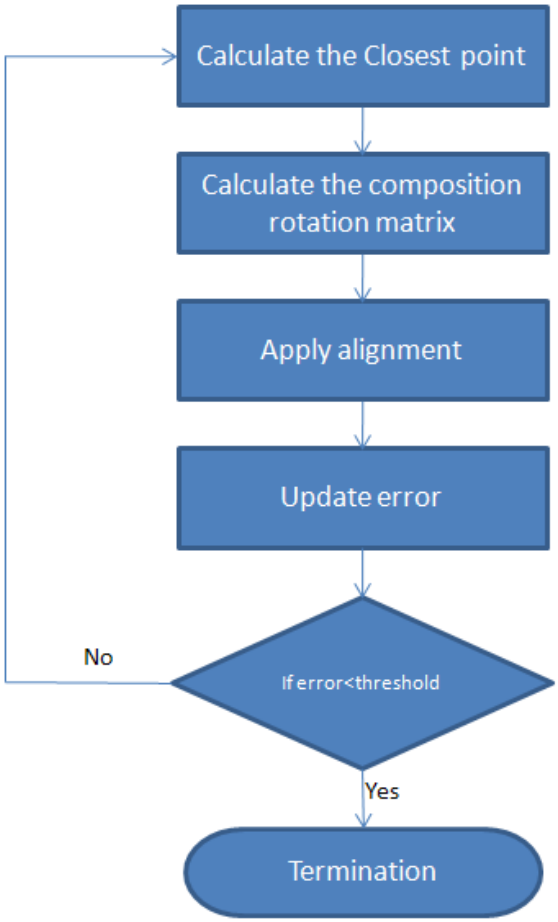


Figure 4.8: *The procedure of ICP alogrithm.*

Since the introduction of ICP by Besl et al. [14], there have been many variants of the ICP algorithm based on different selection and matching of points to the minimization strategy. These variants of ICP result in different accuracy and performance of convergence [78]. In the following parts of this section, we will propose an accurate face alignment algorithm based on the symmetry of the human face using ICP algorithm. In this thesis, we ignore

the differences of these variants of ICP algorithms. We only use the basic concept and algorithms of ICP first proposed by Besl et al [14] to produce a baseline performance. Appendix B shows the details of Besl's ICP algorithm. If the face alignment based on this algorithm can satisfy the requirement for face alignment and the following face recognition tasks, using another more efficient variant of ICP algorithms or other range image registration methods such as GA/SIM [82] will also be practicable and could further increase the accuracy and speed of the ICP alignment process.

4.4.2 Face alignment based on the symmetry of the human face

As mentioned in section 4.3, PCA-based face alignment is not capable of handling surface loss, hair styles and distortion problems. Moreover, if the automatic localized position of the nose tip is not at the exact position of the nose tip, the symmetry of the cropped face area could be affected. And thus the PCA-based face alignment could produce slightly inaccurate results. An example is shown in figure 4.9. We can see from this figure that the inaccuracy in nose tip localization makes the cropped face area slight asymmetric. And the asymmetry results in misalignment after PCA-based face alignment.

On the other hand, the human face can be considered as a symmetric surface along the OYZ plane as shown in figure 4.10. There are several methods making use of the symmetry of human face to implement face authentication or registration. Inspired by [12], [96] and [68], face alignment based on the Iterative Closest Point(ICP) algorithm can be optimized by utilizing the symmetry

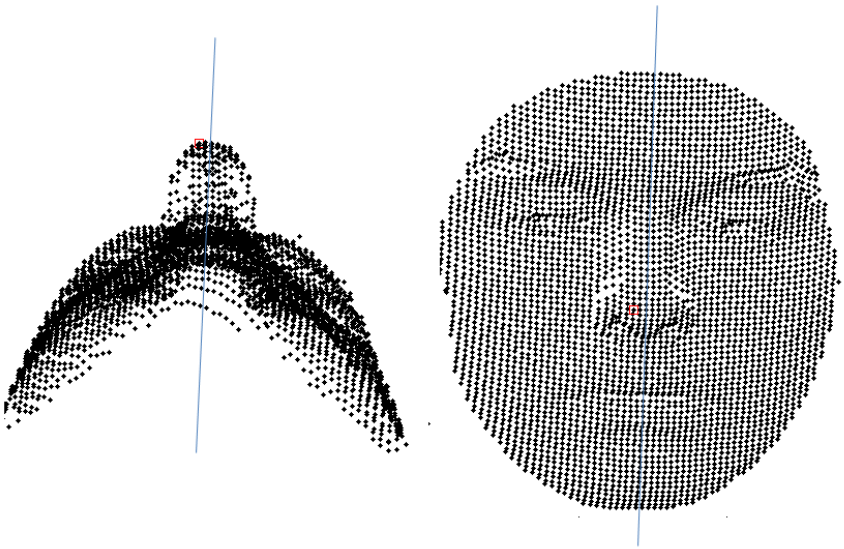


Figure 4.9: *An example of misalignment caused by inaccurate nose tip localization. The small red square is the position of automatic localized nose tip. The left side of the face has slightly more number of points. The PCA-based face alignment method is thus affected by the asymmetry.*

of the face. However, in their implementation they only located a symmetry plane of the face and did not consider the effect of expression variations.

If there is a target face: $F = (X_t, Y_t, Z_t)$, we can define a mirror face as the model face M:

$$M = F_{mirror} = (-1 \cdot X_t, Y_t, Z_t) \tag{4.5}$$

By applying the ICP algorithm, the target face can rotate to fit the model face if the mirror face is used as the model face. The rotation matrix and the transformation matrix can be calculated and obtained. According to the fundamentals of computer graphics [37], every 3D rotation is a composition of three rotations about the x-axis, y-axis and z-axis:

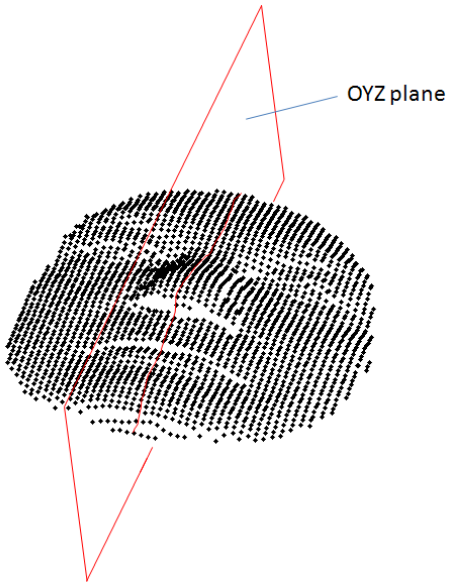


Figure 4.10: *Human face is a symmetric surface about OYZ plane.*

$$R = R_y(\theta) \cdot R_x(\alpha) \cdot R_z(\beta) \tag{4.6}$$

Where:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\beta) = \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 & 0 \\ -\sin(\beta) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since the model face is the mirror face of the target face along the oyz plane, the rotation angle α along the x-axis is equal to zero and there are two rotations left as shown in figure 4.11. If the target face is rotated by angle $\frac{\theta}{2}$ along the y-axis and angel $\frac{\beta}{2}$ along the z-axis, the aligned face is at the desired front view pose as shown in figure 4.12.

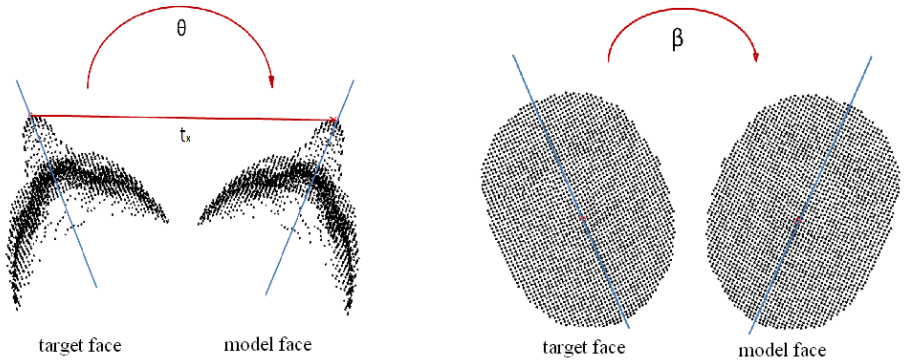


Figure 4.11: *Rotations along y-axis(left figure) and z-axis(right figure) from the target face to model face(mirror face).*

Then, we can use this part of the face as the target model to fit the mirror face. After applying the ICP algorithm between the target model and the mirror model, a rotation matrix R and a transformation matrix T can be calculated. Given the rotation matrix:

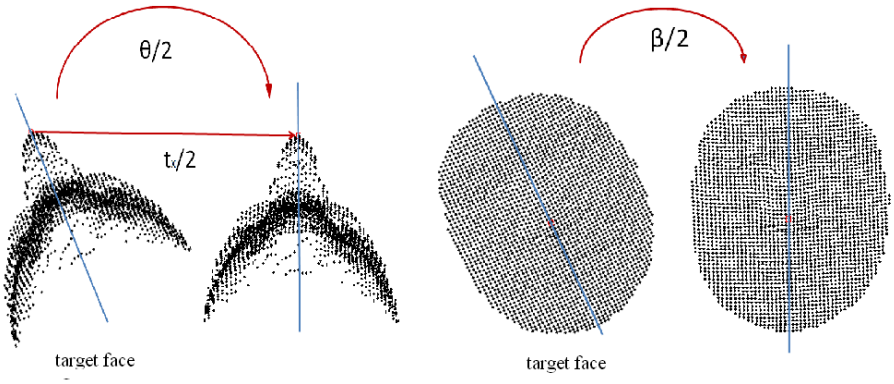


Figure 4.12: *The target face is aligned to a perfect front view position according to the θ and β generated by applying ICP to rotate target face to model face (mirror face).*

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix}$$

According to the above equations, we can calculate:

$$r_{23} = \sin(\alpha) \tag{4.7a}$$

$$r_{13} = -\sin(\theta)\cos(\alpha) \tag{4.7b}$$

$$r_{21} = -\sin(\beta)\cos(\alpha) \tag{4.7c}$$

Thus, we can calculate the three angles α, θ and β respectively:

$$\alpha = \arcsin(r_{23}) \tag{4.8a}$$

$$\theta = \arcsin(-r_{13}/\cos(\alpha)) \tag{4.8b}$$

$$\beta = \arcsin(-r_{21}/\cos(\alpha)) \tag{4.8c}$$

As we already know that the model face is the x mirror of the target face, the rotation along x-axis is almost equal to zero. The composite rotation is mainly formed by rotations about the y-axis and z-axis. If there is a rotation defined as follows:

$$\alpha_{new} = 0 \tag{4.9a}$$

$$\theta_{new} = \frac{\theta}{2} \tag{4.9b}$$

$$\beta_{new} = \frac{\beta}{2} \tag{4.9c}$$

The translation matrix $T = [t_x, t_y, t_z]$ can be calculated by applying ICP algorithm. Then the new transformation matrix can be created as:

$$T_{new} = [\frac{t_x}{2}, 0, 0] \tag{4.10}$$

Then we can apply the rotation according to the new rotation matrix R_{new} and the transformation matrix T_{new}. The target face is aligned to a new position by applying the rotation:

$$F_{new} = R_{new} \cdot F + T_{new} \tag{4.11}$$

Even when the automatic localized position of the nose tip has a certain distance to the real nose tip that is exactly on the symmetry plane, the error distance along x-axis of the nose tip to the real position is neutralized because of the calculation of $\frac{t_x}{2}$ as shown in figure 4.13. Thus, another effect of this rotation is that the error distance of the automatically localized nose tip position along x-axis is further reduced towards zero.

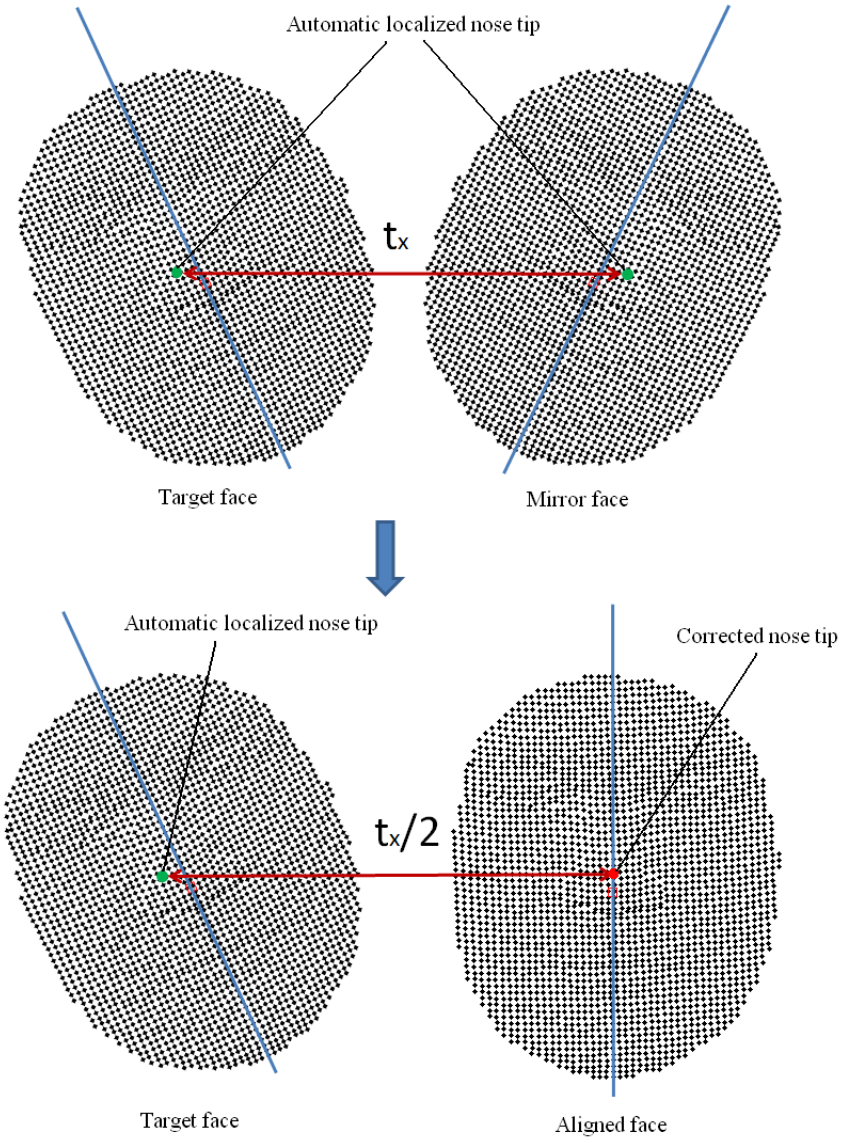


Figure 4.13: *The position of the nose tip is further corrected by implementing* $[\frac{t_x}{2}, 0, 0]$ *as the transformation matrix.*

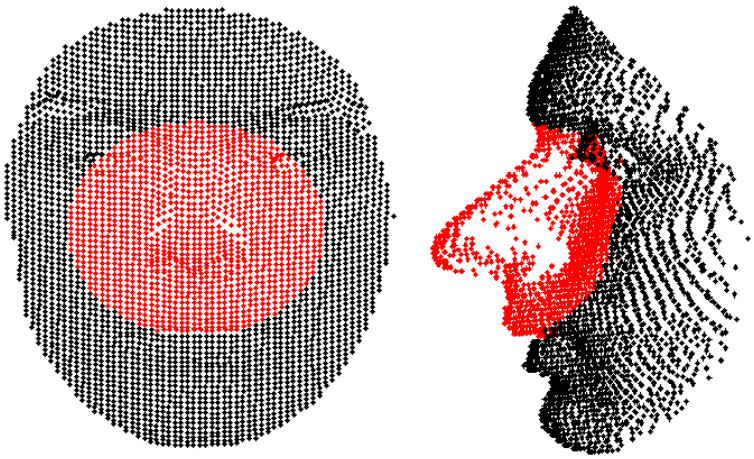


Figure 4.14: *Red region is involved in the symmetric alignment because This region is the most expression-invariant area.*

Facial expression variations could generate some asymmetric shapes, which will affect the mirror face alignment. However, according to the table 4.1, most facial expressions occur in the area near the mouth and the facial region around the nose tip is the area least affected by expression variations. Consequently we can use a sphere around the nose tip to crop a piece of the face surface as a relatively expression-invariant and symmetric area. Additionally hair also may affect the symmetry of this area. Thus we choose $45mm$ as the radius of this sphere to avoid the effect of hair and keep the symmetry of this area as shown in figure 4.14. The whole procedure is shown in figure 4.15. The target face is only rotated to half of the rotation angles to the mirror model. Since the face is a symmetrical surface, the position of the target face after this rotation is exactly the front view position.

Finally, implementing the face alignment using the symmetry of human face has two outcomes:

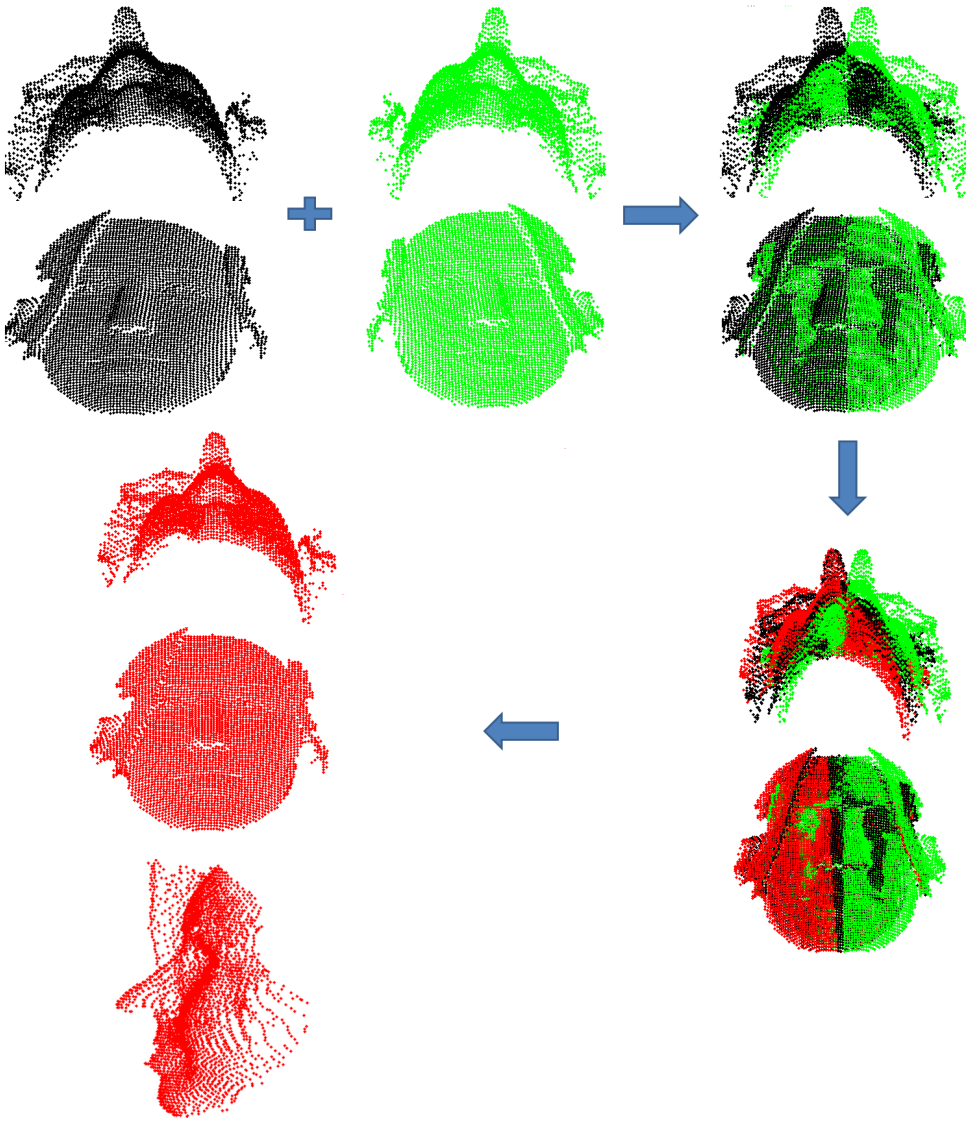


Figure 4.15: *Black face is the target face; green face is the mirror face about OYZ plane; red face is the face after applying alignment of symmetric algorithm.*

1. Error distance of the localized nose tip position along the x-axis is reduced towards zero.

2. Face misalignments along the y-axis and z-axis are minimized.

4.4.3 ICP face alignment using expression-invariant regions

After face alignment based on the symmetry of the human face, the misalignment along the x-axis is still not aligned and there is still an error in the automatic localized position of the nose tip along the y-axis and z-axis. On the other hand, human faces share relatively similar facial features and structure. So it is possible to align a face to another face by adjusting its rotation to a standard position. Figure 4.16 shows as an example that two faces are fitted together by using ICP algorithm. If the slight imprecision of the alignment caused by the variations of facial expression is temporarily ignored, the faces from the same individual share a common shape. Thus, when those faces are fitted to a standard face template which is from another individual, their alignments will appear very close to being the same. Every facial feature is aligned to almost the same position. That result also can be used to further improve the accuracy of the nose tip detection. Since faces belonging to the same person share more elements in common than faces from different individuals, the facial features, especially the nose tip, if they are from the same people, will be corrected to similar positions. Shown in figure 4.17, three faces are aligned to a standard face which belongs to a different person. Each of them has a very closely aligned position.

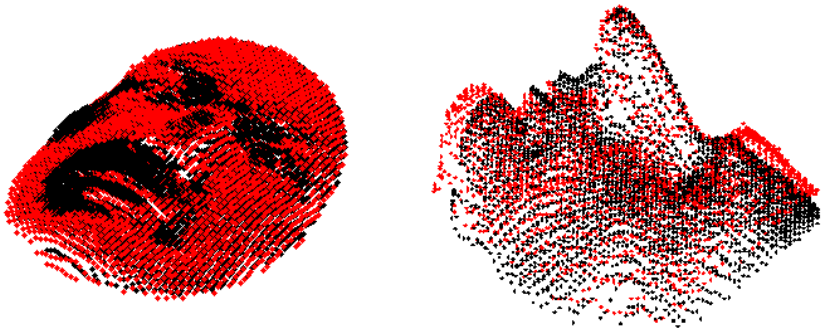


Figure 4.16: *Face one(red) can be fitted to a face templet(black) by applying ICP alignment.*

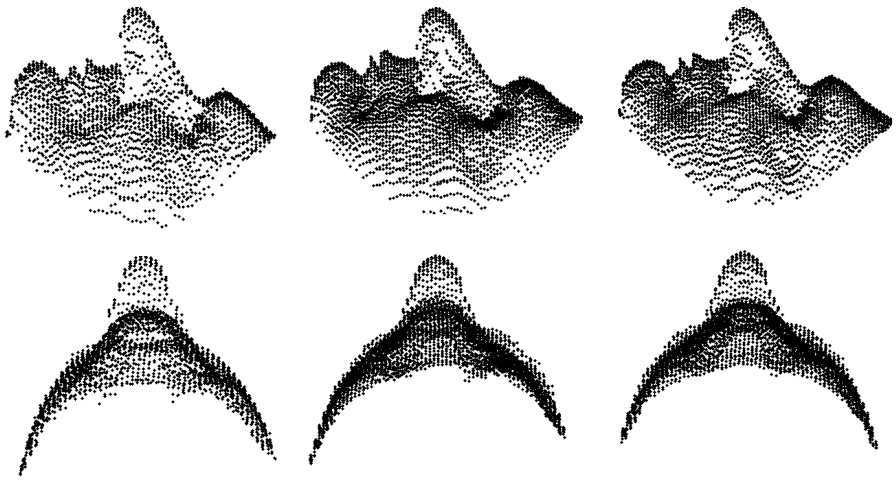


Figure 4.17: *Three faces from the same individual show close positions after applying ICP alignment to fit each of them to a standard face template respectively.*

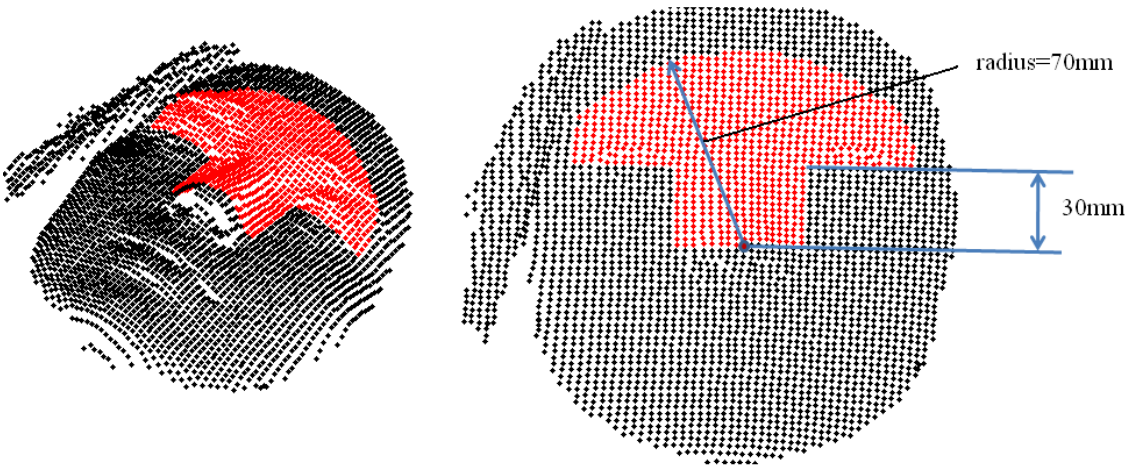


Figure 4.18: *When apply the ICP algorithm, only the points within the red region of the target model are used.*

In order to reduce the number of misalignments caused by expressions, it is required that the parts of the face insensitive to expressions are used in the alignment. In face alignment based on the symmetry of the face, the misalignments along y and z-axis have been minimized. As a result, we can define a region shown in figure 4.18 Only points near the nose tip and above the eyes (within a sphere $r = 70mm$ to reduce the effect of hair) are used in the ICP alignment just because the nose, eyes and the forehead regions are the least affected by expressions in 3D shapes. In some cases, because of the error of the nose tip position, the target face area may exceed the range of template face if we use the same size to crop the expression-invariant area to apply ICP. Therefore, the expression-invariant region cropped in the standard face template is slightly (radius=75mm) larger than the corresponding region of the target face to avoid unexpected incorrect results.

However, such a region which is on the upper face could be affected by hair

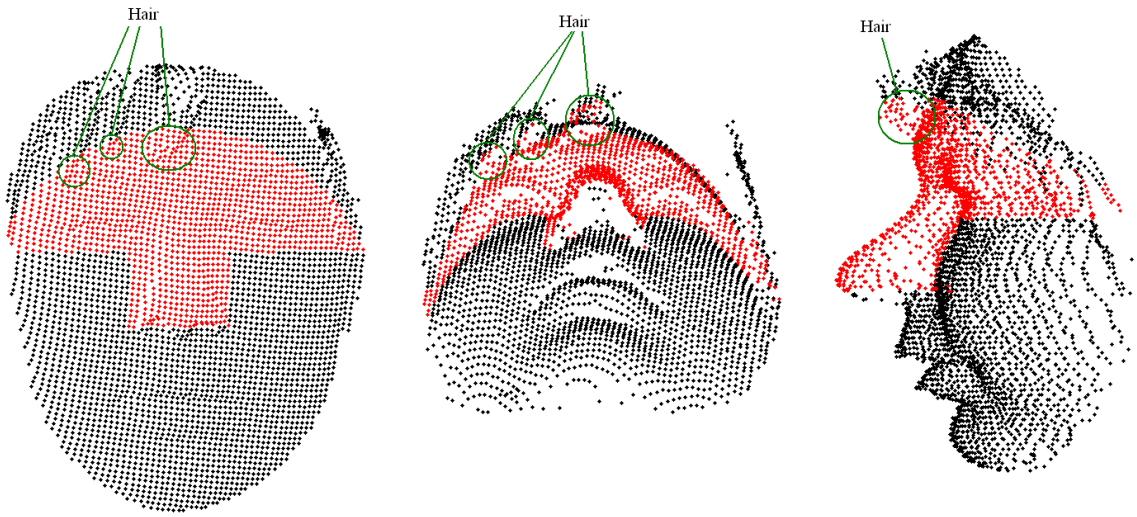


Figure 4.19: *The hair could damage the symmetry of the shape in expression-invariant region. The hair noise also could affect the results of ICP-based alignment.*

noise as shown as in figure 4.19. Hair style variations may cause asymmetric shapes. Fortunately, we have aligned the face according to the symmetry of the face. The shape of a face especially in the expression-invariant region should be a symmetric shape. So, the z value of a certain point should equal its corresponding point(with the same y value and $-x$ value) on the mirror side. Consequently, the hair can be detected by finding the much larger z values (by defining a threshold) compared to the corresponding points of the mirror side of the face. Then those points are removed before applying the ICP algorithm in case those points affect the alignment.

Unlike other face alignment approaches [60] [92] [35] based on the ICP algorithm, which used the whole composite rotation matrix to rotate the target face, we only use the information about rotation along the x-axis to align the target face. Given a composite rotation matrix generated by the ICP algo-

rithm and equation 4.8a, we can obtain the rotation angles α, θ and β along the x, y and z-axis. Since we have minimized the misalignments on the y-axis and z-axis in the face alignment based on the symmetry of the face, here we only need the α along the x-axis to align the target face. Then the rotation matrix R can be calculated by using the following equations:

$$R = R_y(\theta) \cdot R_x(\alpha) \cdot R_z(\beta) \tag{4.12}$$

Where $\theta = 0$ and $\beta = 0$, so:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\beta) = \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 & 0 \\ -\sin(\beta) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And the transformation matrix T can be computed as:

$$T = [0, y_{template}, 0] \tag{4.13}$$

Where $y_{template}$ is the y value of the nose tip of the standard face template.

Then we can implement the composite rotation by using equation 4.11. Fur-

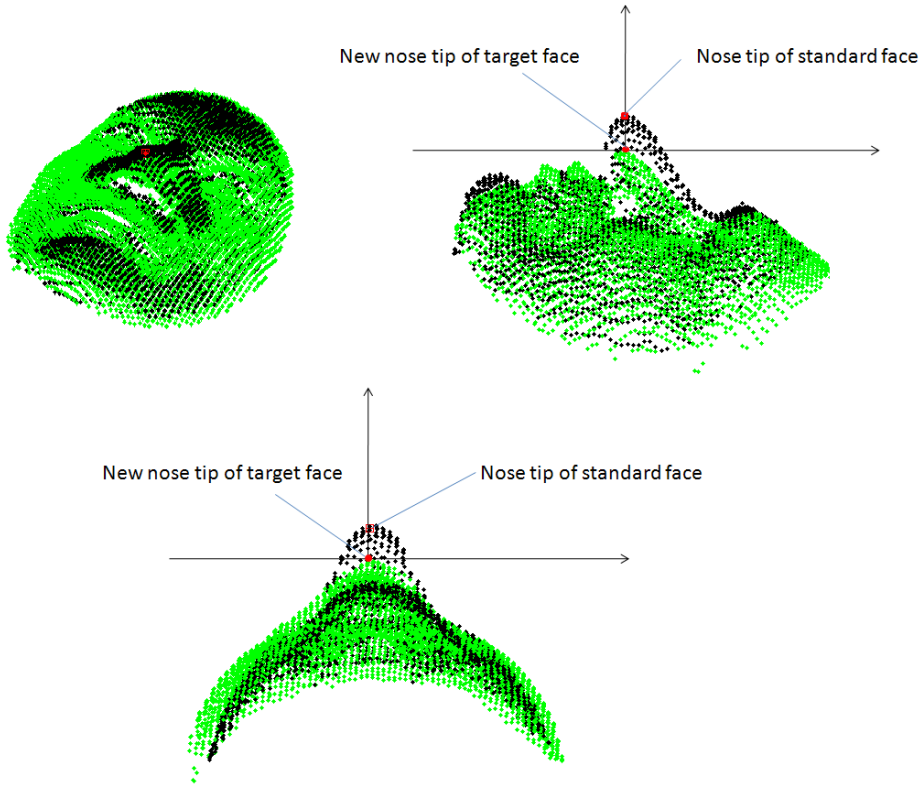


Figure 4.20: *Nose tip re-localization. Green face is the target face and the black face is the standard face template. Using the y value of the nose tip position of standard face template and the original x value to locate the new nose tip position.*

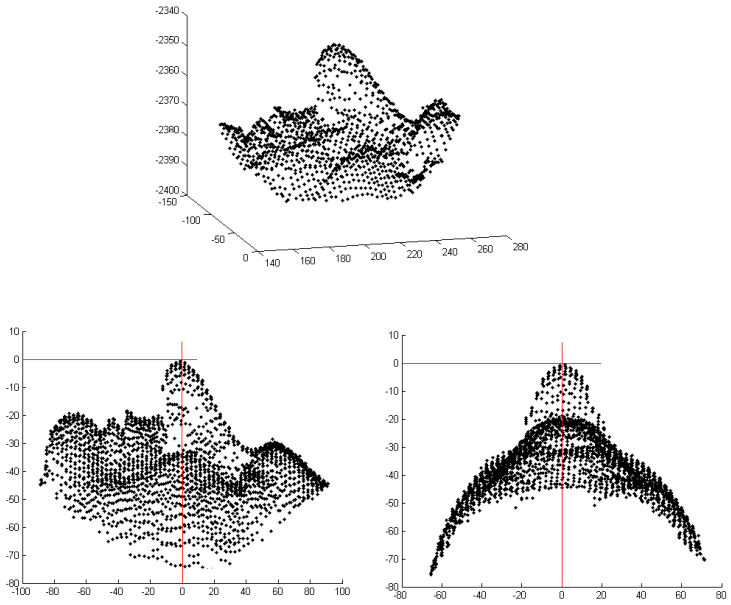


Figure 4.21: *Target face is shifted to a new coordinate system. The nose tip is shifted to the center of the coordinate system.*

thermore, after applying ICP alignment, the nose tip of the target face is re-localized by using the nose tip of the standard face template. The new position of the nose tip uses the y value of the nose tip position of the standard face template plus its own x value of the nose tip to find the closest z value within the target face. Figure 4.20 demonstrates an example of how to implement nose tip re-localization. This process can further improve the accuracy of the nose localization, especially the nose position accuracy between faces belonging to the same individual simply because those face share a similar shape. After this ICP-based alignment using the expression-invariant region, all faces are precisely aligned into a desired front view position even along all of the x, y and z-axis. Defining the re-localized nose tip as the zero point of the coordinate system, all faces are shifted into the same coordinate system as shown in figure 4.21.

4.5 Evaluations

In this thesis, we use FRGC v2 as the experimental database to evaluate the performance of our face alignment approach. The face images have been downsized from 640×480 to 160×120. Although the subjects are asked to look at the camera during the data acquisition procedure and most of the faces show a front view pose, there are still some faces appearing pose variations. Some examples are shown in figure 4.22. Also expression variations exist in FRGC v2 database. Figure 4.23 shows some example of one subject in the FRGC v2 database.

Figure 4.22: *Examples of pose variations in FRGC v2 database.*

Figure 4.23: *Examples of expression variations in FRGC v2 database.*

After the PCA alignment, about 10% of the 4007 faces appear to have a certain misalignment. By applying the integrated face alignment approach, no observable misalignment is found during the manual check. Even the face that does not have a complete nose achieves a relatively correct alignment as shown in figure 4.24.

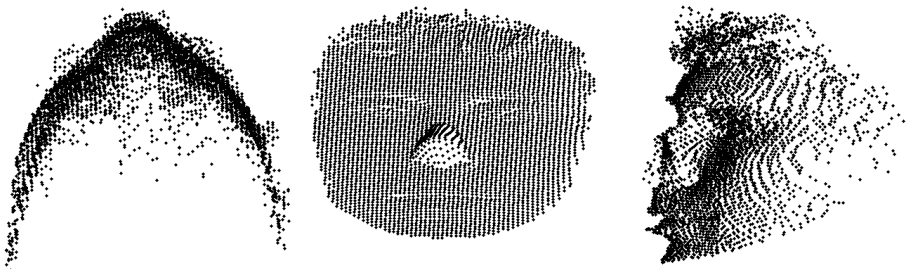


Figure 4.24: *Three views of a noseless face. We can found that even a face without a complete nose can also be aligned by applying our face alignment approach.*

However, it is not easy to compare the performance of our face alignment method with other state-of-the-art techniques. In this chapter, we try to evaluate the in-class and between-class differences of all faces in the FRGC v2 database by comparing different face alignment approaches. We separate the FRGC v2 face database into two categories: neutral faces (2182 faces) and

non-neutral faces (1825 faces) to test the performance of correcting face pose and the ability to handle the expression variations. We classify the current state-of-the-art techniques into four types and then use the following methods to simulate those four face alignment techniques.

1. PCA-based face alignment using the whole face area which is introduced in section 4.3 (a similar method is used in [64]).
2. Face alignment using the ICP algorithm to fit the whole target face to a standard face template (similar methods are used in [35] [50]).
3. Face alignment using the ICP algorithm to fit a sphere (r=45mm) area around the nose tip of the target face to a standard face template (a similar method is used in [92]).
4. Face alignment using the ICP algorithm to fit the expression-invariant area of the target face to a standard face template (a similar method is used in [60]).

Since the expression-invariant regions of faces belonging to the same people share similar shapes, we can use the differences of the expression-invariant region between faces of the same individual to represent how good the face alignment is. It is also an indicator of the in-class difference. We can calculate the mean squared error distance (MSE) between the corresponding points within the expression-invariant region of faces belonging to the same individual. If a subject has n face images, we will calculate the MSE of every possible face-face combination. The total number of these combinations is $(n-1)+(n-2)+...+2+1$. Then we compute the mean values of the error distances between the corresponding points (the closest points) of these face-face combinations by using equation B.18. Table 4.2 shows the in-class MSE values of different face alignment methods. Our method achieves the smallest

Methods	MSE(Neutral faces)	MSE(Non-neutral faces)
Method 1(PCA)	$0.5033mm$	$0.5793mm$
Method 2(Whole face)	$0.2594mm$	$0.3327mm$
Method 3(Nose)	$0.3186mm$	$0.4358mm$
Method 4(Expression-invariant)	$0.2729mm$	$0.3084mm$
Our method	$0.1940mm$	$0.2550mm$

Table 4.2: *Comparison the MSE between faces belonging to the same individual by using different face alignment approaches in.*

in-class MSE values both in neutral faces and non-neutral faces. The cumulative percentages of the in-class MSE of neutral faces and non-neutral faces using different face alignment methods are shown in figure 4.25 and figure 4.26. In these figures, we can see that our method outperforms other methods under neutral expression and even under expression variations.

The MSE evaluation given above tests the in-class differences of these approaches. On the other hand, we can use the results of the identification experiment based on the results of different alignment approaches to compare the between-class distinguishing ability. In the FRGC v2 database there are 465 subjects. We select the first face images of each subject as the gallery dataset. The remaining face images are separated into two datasets: neutral faces and non-neutral faces. We define two rank-one identification experiments: "first face vs neutral face" and "first face vs non-neutral face". In the "first face vs neutral face" experiment, 1761 neutral faces consist of the test dataset and the gallery dataset includes all of the first face image (465 faces) of each individual in FRGC v2 dataset. Each face in the test dataset is matched to every face in

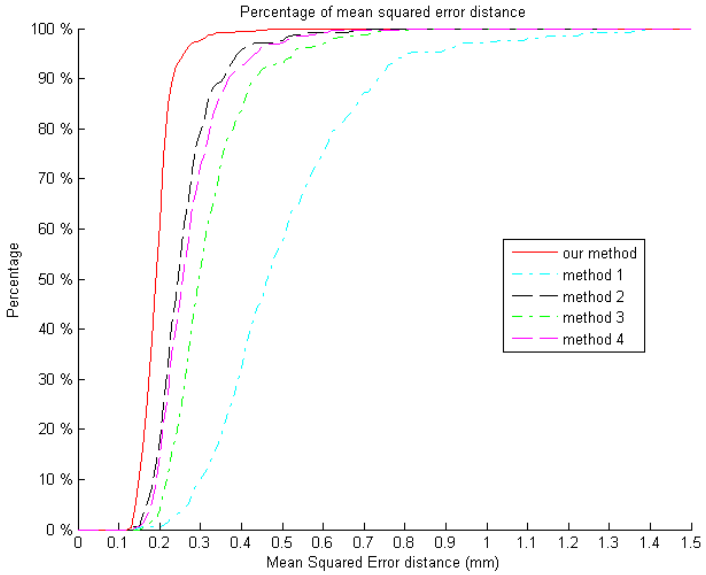


Figure 4.25: *Cumulative percentages of the in-class Mean Squared Error Distance of neutral faces.*

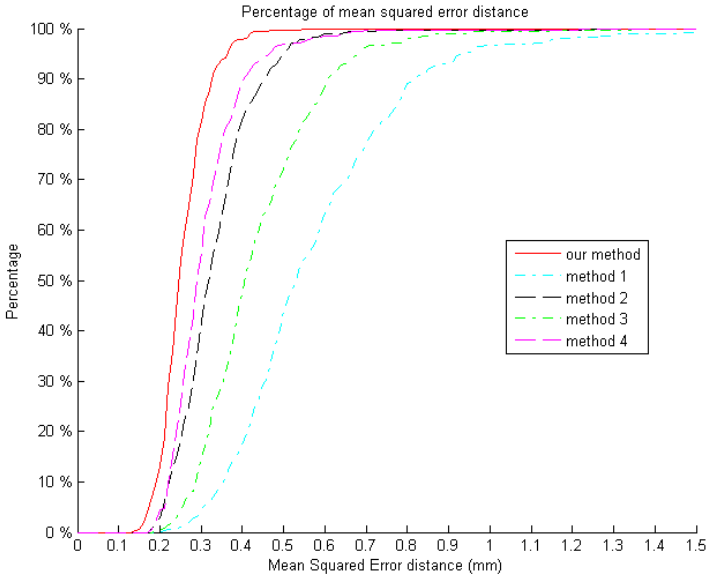


Figure 4.26: *Cumulative percentages of the in-class Mean Squared Error Distance of non-neutral faces.*

Methods	firs vs neutral	first vs non-neutral
Method 1	27.71%	19.99%
Method 2	63.60%	44.97%
Method 3	47.42%	30.43%
Method 4	53.83%	46.60%
Our approach	96.31%	85.29%

Table 4.3: *Comparison of rank-one identification rates.*

the gallery dataset. If the match with rank-one similarity is a match between two faces belonging to the same person, this match is considered as a correct match, otherwise it is an incorrect one. So there are 1761×465 matches. In the "first face vs non-neutral faces" experiment there are 1781×465 matches. To generate the similarity score of a match, we use the mean squared error distance method to measure the similarity between expression-invariant regions of two faces. The mean squared error distance method is also used in the ICP-based face recognition approach [35] [64]. Table 4.3 shows the results of these two experiments. We find that our approach outperforms the other methods both in "neutral faces vs neutral faces" and "non-neutral faces vs non-neutral faces" experiments.

4.6 Conclusions

In this chapter, we proposed an integrated ICP-based approach to align faces even with expression variations. The first PCA alignment makes it possible to roughly correct the severe misalignments of faces. Then a face alignment based on the symmetry of the face minimizes the possibility of misalignments along

the y and z-axis and reduces the error distance of the automatically localized nose tip position along x-axis to zero. That makes it possible to precisely extract an expression-invariant region. After that, a face alignment based on ICP algorithm using the expression-invariant region produces the rotation angle α along the x-axis. By rotating that angle α along the x-axis, the face can be further aligned to a front view position. The position of the nose tip is also further corrected by using the y value information of the standard face template's nose tip. In the comparison with four state-of-the-art face alignment techniques, our approach achieves the best performance both in the in-class and between-class evaluation experiments.

Chapter 5

Face Recognition

5.1 Introduction

Face recognition is a very difficult task due to many challenges in face localization, alignment and matching. Among the FRGC face database related state-of-the-art techniques, a surface matching (or range image matching/registration) algorithm [64] [35] [74] is frequently used, such as ICP or SA/SIM. Those approaches both use surface matching to match the nose, eyes and forehead regions respectively, which are considered as expression-invariant regions. However, implementing such algorithms is a very time-consuming task. In particular, in [35] and [74], the surface matching algorithm has been used in the face alignment stage. In other words, points of the face surface are reused many times in such algorithms, which causes a low efficiency in the face recognition system.

In previous chapters, 3D face detection based on nose tip localization and face alignment using an integrated method have been accomplished. On the basis of these achievements, in this chapter we will propose a face recognition system

using a weighted surface matching algorithm based on the shape descriptor - MSSAMD. Since pose variations of faces have been corrected and aligned in the previous chapter, there is one challenge left: expression variations. We attempt to segment the face area into two regions: the expression-invariant region and the expression-variant region. Inspired by state-of-the-art face recognition approaches [35] [74], which both actually assigned a high weight to the expression-invariant region in the face recognition stage, we propose an accumulating weighted face surface matching method. Unlike those approaches matching different face region by using a surface matching algorithm such as ICP or SA/SIM, the proposed method compares two face surfaces/shapes using a simpler method which depends on the pose-invariant ability of the shape descriptor. Our method does not have a very high computational cost unlike those methods applying ICP or similar surface registration/matching algorithm.

The remainder of this chapter consists of the face matching system based on the shape descriptor in section 5.2, the face segmentation in section 5.3, an improved accumulating weighted face matching method in section 5.4 and the hierarchical face verification in section 5.5. Face identification and the face verification experiments are performed based on the FRGC v2 face database in section 5.6.

5.2 Face matching based on the shape descriptor - MSSAMD

In chapter 3, the 3D shape descriptor MSSAMD actually represents the relationship of a point with its neighboring points. A MSSAMD of a certain point symbolizes a piece of 3D surface around this point. A 3D face image is a point cloud with n points, and each of these points can produce a MSSAMD. The values of MSSAMD of all points constitute a $n \times m$ matrix (m is the number of shells). Since a MSSAMD represents the relationship of neighboring points around a particular point, the surfaces around those points are overlapped together. To compare two faces, corresponding points of each face have been matched in pairs. The number of pairs of corresponding points which achieve a correct match is an indicator of similarity. In some face recognition work [64] [74] [35], researchers made an assumption that 3D face images belonging to the same person share similar shapes especially in the region least affected by expressions. Thus, the more pairs of corresponding points sharing similar MSSAMD that there are, the higher similarity of those two faces is.

Since faces belonging to the same individual share similar shapes, to compare the difference of two faces, we can simply compare their shapes. In [64] [35], they used a surface match algorithm such as ICP to match different faces and then use mean squared error(MSE) to measure the different of two surfaces. If we also use MSE to compare two faces based on the results of chapter 4 and implement the identification experiments "first vs neutral" and "first vs non-neutral", the identification rates are 96.31% and 85.29% respectively which is mentioned in section 4.5. The overall rank-one identification rate of all neutral and non-neutral faces is about 90.77%. Compared with state-of-the-art tech-

niques [64] [50] [35] [74] which achieved over 95% rank-one identification rates in the same experiment on the FRGC v2 database, apparently, this performance is not acceptable. The accuracy of the face alignment is not precise enough for face matching using the MSE measurement. The MSE measurement is too sensitive to the slight misalignment which is difficult to solve in the face alignment stage. That is also the reason why further surface matching algorithms are applied to precisely match and measure the difference of two surfaces in [64] [35] [74].

However, applying these surface matching algorithms are high computational cost tasks. In this chapter, we attempt to use a face matching method based on the pose-invariant surface/shape descriptor - MSSAMD to quickly evaluate the similarity of two 3D face surfaces. If a 3D face is separated into numerous overlapped small surface patches, the difference of two faces can be represented as the number of these patches that have similar shapes to their corresponding patches in another face. The greater the number of patches that have similar shapes, the more similar these two faces are. A MSSAMD shape descriptor contains the information of the relationship of a certain point with its neighboring points, thus a MSSAMD can be considered to represent a small piece of 3D surface. Since MSSAMD is a pose-invariant surface/shape descriptor, the MSSAMD is able to tolerate a small misalignment between two faces. Although there is a possibility that different shapes generate similar MSSAMD values, considering that the MSSAMD descriptor is a multi-shell descriptor(two 1×5 vectors) and a face point cloud has a great number of points, the chance that two faces with different shapes achieve a great number of points having similar MSSAMDs is very low. When two corresponding points Pa and Pb from two faces are matched, if they have similar MSSAMD,

their shapes represented by the MSSAMDs can be considered identical. The similarity of these two points at this position i can be defined as the distance between two MSSAMD:

$$dist = \mid distance(MSSAMD_{Pa} - MSSAMD_{Pb}) \mid \tag{5.1}$$

If $dist$ is below a certain threshold ε, the shape at this position can be considered as identical shapes. If we use the same parameters of MSSAMD as in previous chapters, the radii of the spheres are $5mm$, $10mm$, $15mm$, $20mm$ and $25mm$. A MSSAMD has five shells, so the mean angle value vector and the STD angle value vector both have five values: mean angle value vector $m = \{m_1, m_2, m_3, m_4, m_5\}$ and STD angle value vector $s = \{s_1, s_2, s_3, s_4, s_5\}$. The distance of two MSSAMDs will also have the difference vector of mean value and the difference vector of STD value. We define two distance vectors to describe the difference between two MSSAMD: d_{mean} and d_{std} as shown in figure 5.1. In order to reduce the effect of noise, two thresholds ε_m and ε_s are defined to filter d_{mean} and d_{std}. Any match generating a MSSAMD difference below ε_m and ε_s can be considered as a match between two identical 3D surfaces. Then the similarity of two faces can be represented by the number of correct matches between corresponding points. In order to choose the proper values of these two thresholds, we run a face recognition experiment on the FRGC v2 database. The gallery dataset is the first face image of each subject, and the query dataset is the remaining face images in the database. In this chapter, we temporarily choose $\varepsilon_m = 3$ and $\varepsilon_s = 1$.

nd_{mean} is defined to represent the number of values in d_{mean} below threshold ε_m and nd_{std} is the number of d_{std} below threshold ε_s. Thus, the range of nd_{mean}

Algorithm 1 Face Matching Algorithm

Require: Two 3D face surfaces: query face $\mathcal{Q}$ and gallery face $\mathcal{G}$;

$S = 0$
for each $p \in \mathcal{Q}$ **do**
  compute two vectors m and s of the MSSAMD of p;
  search the corresponding point p' in $\mathcal{G}$;
  compute two vectors m' and s' of the MSSAMD of p';
  compute the distance vector $d_{mean} = |m - m'|$;
  compute the distance vector $d_{std} = |s - s'|$;
  $nd_{mean} = 0$
  for $dm_i \in d_{mean}$ **do**
    if $dm_i < \varepsilon_m$ **then**
      $nd_{mean}++$;
    end if
  end for
  $nd_{std} = 0$;
  for $ds_i \in d_{std}$ **do**
    if $ds_i < \varepsilon_s$ **then**
      $nd_{std}++$;
    end if
  end for
  if $nd_{mean} < t_m$ and $nd_{std} < t_s$ **then**
    $S++$;
  end if
end for

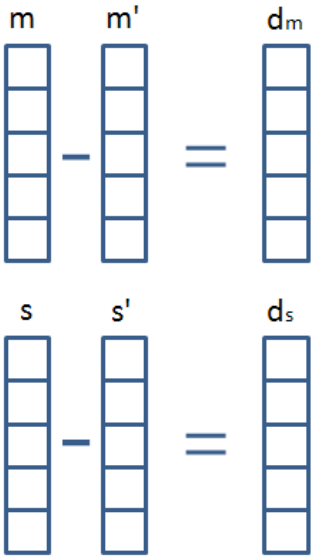


Figure 5.1: *s and m are the two vectors of the MSSAMD of a point p, and s' and m' are the two vectors of the MSSAMD of the corresponding point p'.*

and nd_{std} is $[1 \sim 5]$. Since the region affected by expression variations may affect the MSSAMD of points at the edge of the expression invariant region, we define two distinguish/tolerate thresholds t_m and t_s to adjust the ability to tolerate noises and the degree to distinguish shapes from different persons. In particular the hair and expressions could also affect the values of MSSAMD at some positions. A face matching algorithm is required to provide enough information to distinguish faces from different individuals. And in the meantime, the face matching algorithm also should have the ability to tolerate the slight differences or noise between the faces belonging to the same person. Therefore, it is necessary to adjust these two thresholds properly to keep the balance of noise-tolerance and ability to distinguish. Two shapes will be considered to match when d_{mean} and d_{std} both are below their corresponding thresholds t_m and t_s. Experimentally, we found the combination of $t_m = 3$ and $t_s = 3$ is a suitable choice for FRGC v2 database. We define $\mathcal{M}$ as the result of a match

	Range
Max(x)	$58.88mm \sim 95.42mm$
Min(x)	$-59.12mm \sim -95.19mm$
Max(y)	$85.69mm \sim 131.511mm$
Min(y)	$-48.19mm \sim -91.32mm$

Table 5.1: *The range of x and y in FRGC v2 database.*

between two corresponding points. If $\mathcal{M}$ of a correct match is set to '1' and $\mathcal{M}$ of an incorrect match is set to '0', then the overall similarity score S between two faces can be defined as the sum of $\mathcal{M}$:

$$S = \sum_{i=1}^{n} (\mathcal{M}_i) \tag{5.2}$$

Where n is number of points of the query face.

The computational complexity to match two faces by using this algorithm is $O(mn)$, where n is the number of points of gallery face and m is the number of points of query face. Since more than thousands of faces may be involved in face recognition experiments, there is a very high requirement of face matching efficiency. For example, a 4007 faces vs 4007 faces experiment will generate $16,056,049$ calculations of similarity score. If the computational time of a single match is about one second, the overall time to complete the whole experiment will be more than 185 days which is infeasible both for experiments and the real world system. In order to reduce the complexity of computation, the positions for matching need to be pre-processed. Thus, a group of sampling positions are used to preset the corresponding position.

Each face image may be a different resolution, so the density and number of

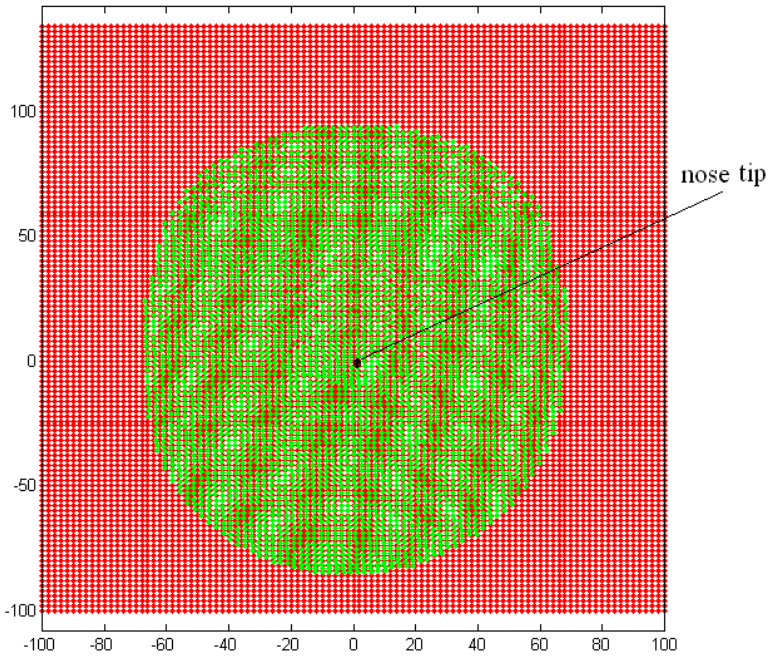


Figure 5.2: *Green region is the area of a face; Red points are the sampling positions.*

the points representing a certain size of 3D face are different. Table 5.1 shows the range of maximum/minimum values along ox and oy directions of all faces in FRGC v2 database. The sampling positions used in this thesis are shown in figure 5.2. By measuring the downsized FRGC v2 database, the range of number of points for each face is from 2157 to 6162. The range of values along ox direction is from $121.51mm$ to $186.23mm$ and the range of values along oy direction is from $153.89mm$ to $198.03mm$. The density of points on the oxy plane can be calculated: from $4.39mm^2$ to $11.22mm^2$ per point, so the range of the interval between points is from $2.09mm$ to $3.35mm$. In order to keep as much information as possible, $2mm$ is therefore chosen as the interval of the sampling position on both oy and oy directions to cover the highest density. Since there is not always a point existing exactly at the sampling position,

the closest point to the sampling position is selected to provide its MSSAMD values. However, if the distance from the closest point to a sampling position is too large, this sampling position will be marked as a invalid point, because this position is out of the range of the face. If we define N as the number of valid points, the similarity score S therefore is modified to:

$$S = \frac{\sum_{i=1}^{N}(\mathcal{M}_i)}{N} \tag{5.3}$$

In order to evaluate the improvement of our algorithm from MSE method used in ICP approaches, we can compare the distance between the within-class and between-class similarity scores in the 'all vs all' experiment by using MSE and our algorithm. Every face in the FRGC v2 database matches with every other faces. Matches between faces belonging to the same subject are within-class matches. The within-class similarity score represent the similarity of faces belonging to the same subject. Matches between faces belonging to different subjects will generate between-class scores. The between-class similarity score is an indicator to show the difference between two different subject. We can see in figure 5.3 that a part of the distribution of within-class and between-class MSE scores overlaps together. On the other hand, as shown in figure 5.4, the histograms of within-class and between-class similarity scores using our algorithm show that the overlapped part is smaller. Compared with the MSE method used in ICP approaches, our method has better ability to enlarge the difference between subjects. We also can use the Fisher's [36] method to compute the separation between two distributions which is the ratio of the between-class variance to the within-class variance by using equation 5.4.

$$J = \frac{|\overline{m_1} - \overline{m_1}|^2}{s_1^2 + s_2^2} \tag{5.4}$$

Where m represents a mean, s^2 represents a variance, and the subscripts de-

note the two classes.

The J values generated by using MSE and our algorithm are 1.543 and 2.744 respectively. These results also show the separation of the between-class and within-class score by using our algorithm is larger than the separation of the MSE method.

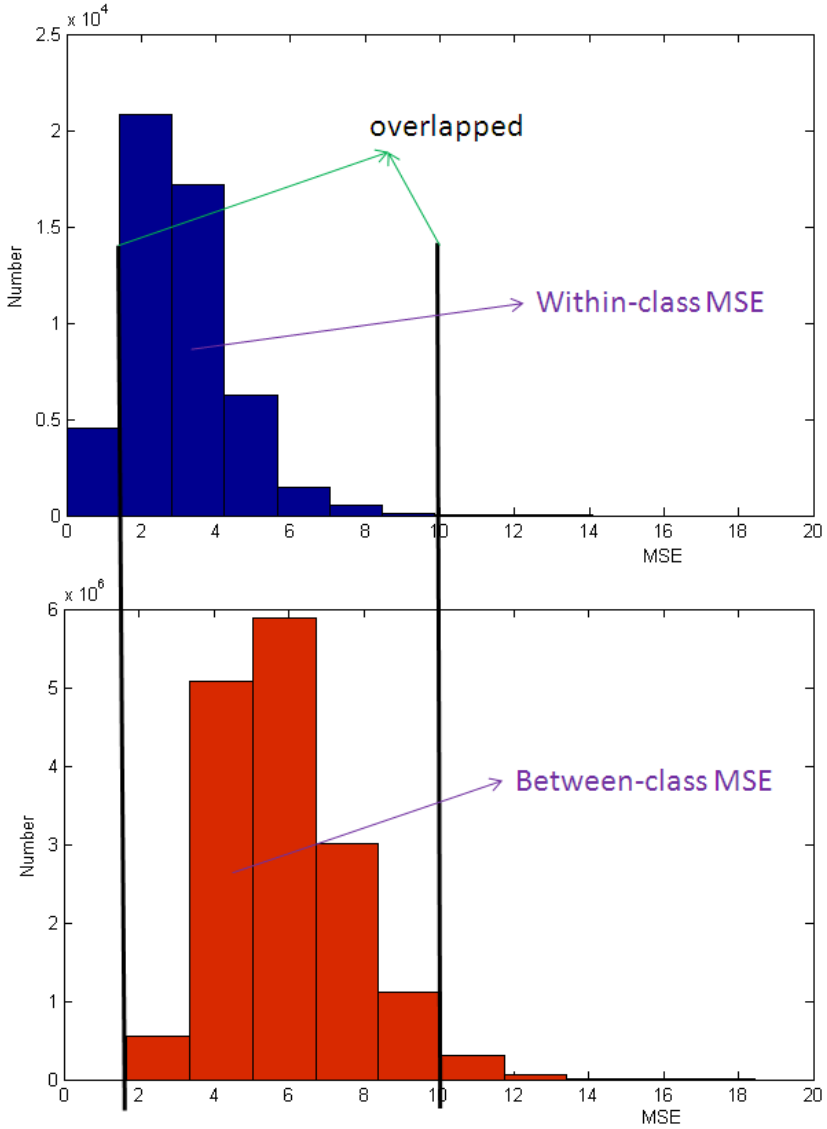


Figure 5.3: *The histograms of the within-class and between-class MSE scores in 'all vs all' experiment.*

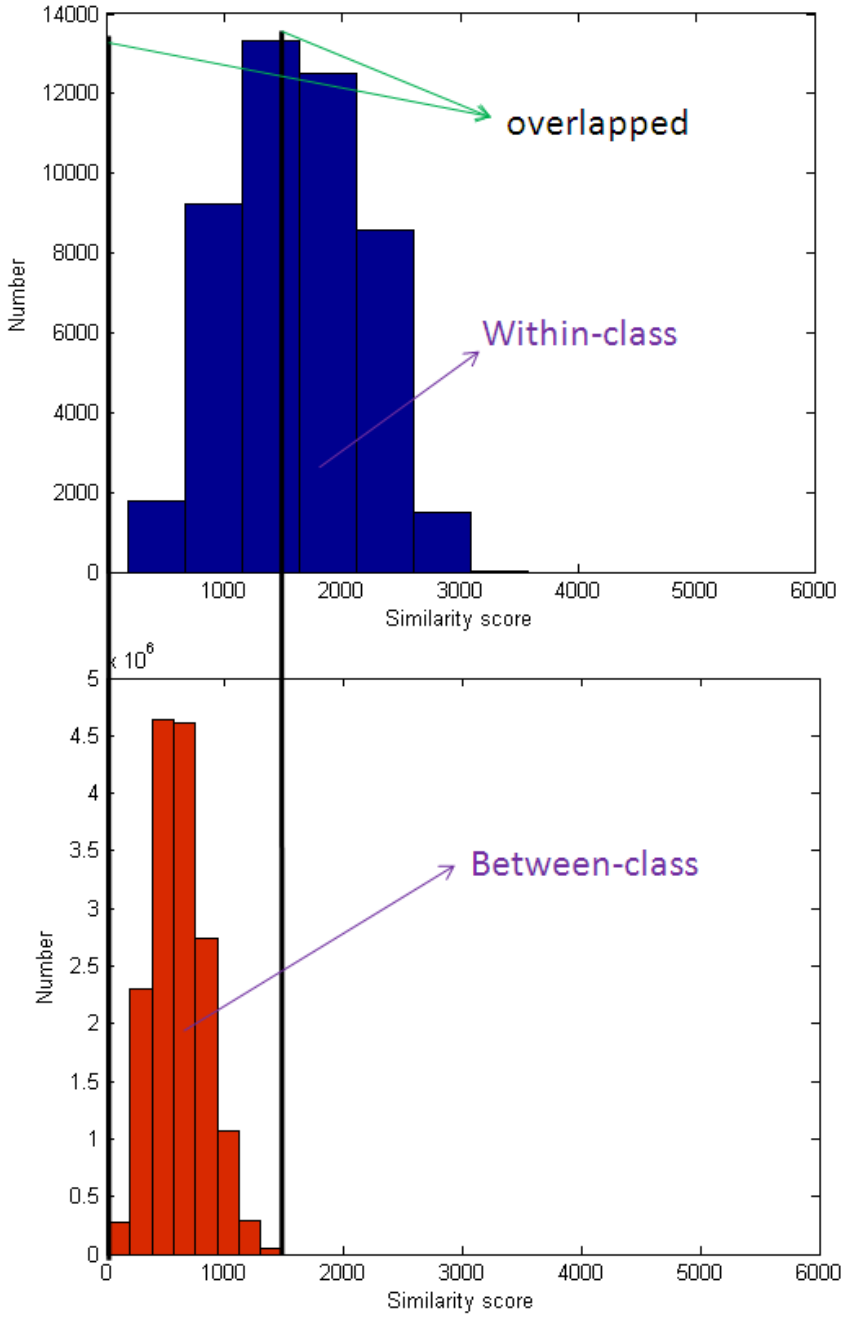


Figure 5.4: *The histograms of the within-class and between-class similarity scores by using our algorithm in 'all vs all' experiment.*

5.3 Face Segmentation

Without the texture information, pure 3D face does not have the illumination problem which is a difficult problem in 2D face recognition. This is an advantage of 3D face recognition. However, the most difficult challenge in 3D face recognition is how to deal with the variations of facial expression. Considering what has been achieved in previous chapters, one possible solution is to precisely segment the face to find and match regions which are not affected by expression variations. According to what has been discussed in the chapter of face alignment, the upper face including nose and forehead are the regions least affected by different expressions and can be called expression invariant regions. The rest of the face appears surface changes to various degrees when expressions are produced. So, an accurate face segmentation is necessary before face matching is performed.

After the successful face detection and alignment in previous chapters, the nose tip has been localized and main face region has been cropped. All faces have been aligned to a certain position according to its shape. Since we use a sphere $r = 100mm$ to crop the main face area, the projection of the 3D face onto xoy plane is a circle $r = 100mm$. Every 3D point can be projected onto the xoy plane. The projection of the nose tip on the xoy plane is set to be the origin of the coordinate system. Currently a very precise eye corner detection has not been achieved, an alternative way has to be used to find the expression invariant region which is related to the positions of eyes. If we use a neutral face as the template face and calculate the z error of the corresponding position between template face and other faces belonging to the same person, different z error values are related to different positions or regions. Since the

pose variations have already been corrected in previous chapters, there are only expression variations existing in the FRGC v2 data. Thus, these z error values can be considered as an indicator to represent the expression-invariant levels of different positions. Then we calculate within-class z error values of all subjects in FRGC v2 database. A figure can be created to show different expression-variations levels at different positions by using the Root Mean Square Error(RMSE) of these values. As shown in figure 5.5, different colors represent different ranges of RMSE values (red $<$ blue $<$ green $<$ magenta), so we can see that the red region around nose, eye and forehead is the most expression-invariant region.

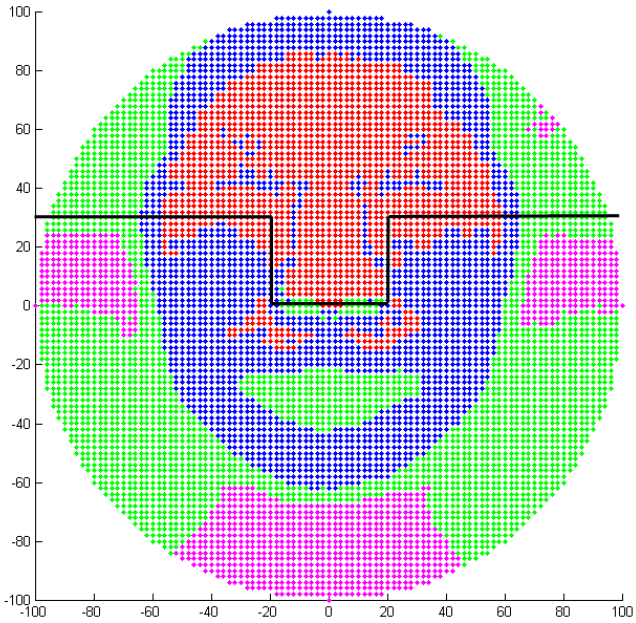


Figure 5.5: *Different colors represent different ranges of RMSE values. Red region has less RMSE values than blue region, (Red:* $0 \sim 1.5mm$*; Blue:* $1.5 \sim 3mm$*; Green:* $3 \sim 5mm$*; Magenta:* $5mm \sim \infty$*). The black lines show the borders of expression-invariant region.*

According to figure 5.5 and by measuring the ground truth data of the FRGC v1 [77], $30mm$ can be selected as the distance from the nose tip to the bottom of the eyes along the oy direction and $20mm$ is chosen as the width of the nose. As a result, the expression invariant region can be defined as the rectangular area around the nose plus the area above the bottom of eyes as shown in figure 5.6. Mian et al. [64] used a similar way to segment the expression-invariant area. This region which is marked red in figure 5.6 keeps relatively constant no matter what expression is produced. Thus this expression invariant region can be granted more weight in face recognition than other regions.

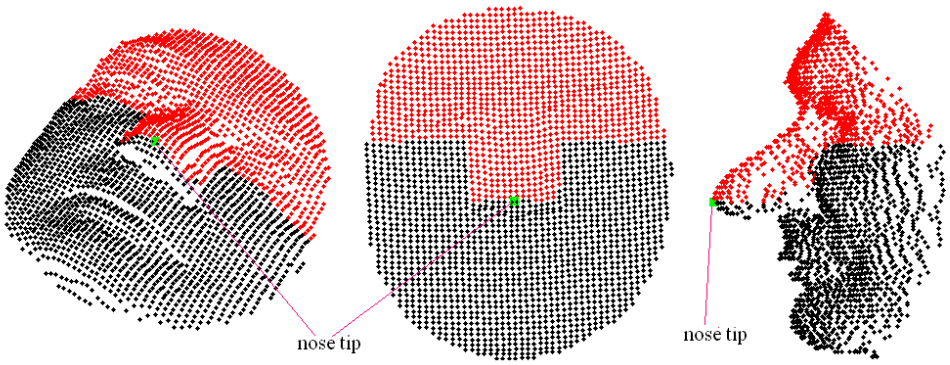


Figure 5.6: *Red region is the region lest affected by expression variations; green square is the position of nose tip.*

5.4 Accumulating weighted face matching

According to table 4.1 in chapter 4, areas near the nose, eye and forehead have different numbers of facial action units(FAU). The region near the nose has the highest tolerance to expression variations because there is only one FAU within this region. Therefore, even in the expression-invariant region segmented in the previous section, different positions should have different

expression-invariant abilities. In some state-of-the-art face recognition approaches [35] [74], these differences have been emphasized in different ways according to their expression-invariant abilities.

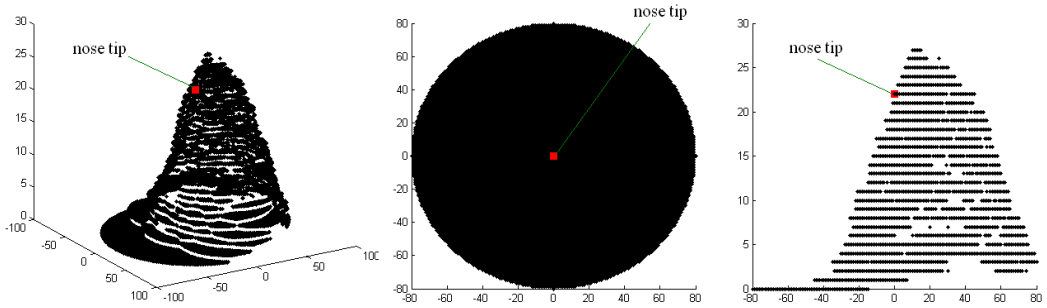


Figure 5.7: *Red square is the weight value at the nose tip; different positions have different values according to [35].*

In [35], Faltermier et al. defined 38 regions on the face and eventually chose 28 regions as the best committee of local regions for maximum results. Table 5.2 shows the parameters of the locations of those regions. Based on the sampling positions created in the previous section, every time that a sampling position is used in a committee region, '1' is added to the weight value of this position. After then, according to table 5.2, we can count how many times a sampling position is used. Finally, we can accumulate and create a vector to represent the weight of sampling positions on the whole face. If we use a circle to represent a face's projection on the xoy plane, z indicates the weight value. This weight vector is illustrated in figure 5.7. From figure 5.7, we can see that the region around the nose has been used the most times can be considered as a weight. Weight values at different position vary according to their distances to the nose region.

Region	x(mm)	y(mm)	Radius(mm)	Region	x(mm)	y(mm)	Radius(mm)
1	0	10	25	15	40	10	45
2	0	10	35	16	0	30	40
3	0	10	45	17	0	30	35
4	0	0	25	18	0	30	45
5	0	0	45	19	0	40	40
6	0	−10	25	20	0	40	35
7	0	40	45	21	0	20	45
8	0	20	35	22	−15	30	35
9	15	30	35	23	−30	20	45
10	−40	30	45	24	40	30	45
11	−20	0	25	25	30	40	45
12	−15	15	45	26	−30	40	45
13	−40	10	45	27	0	60	35
14	15	15	45	28	30	20	45

Table 5.2: *The definition of regions used in [35].*

Queirolo et al. [74] segmented the whole face into several regions: nose circle, nose ellipse, upper head and a region including nose square and forehead. After accumulating all regions together, the weight vector can be generated and shown in figure 5.8. We find that the region around the nose has been used five times, the forehead has been used three times and the cheek regions have been used two times, the mouth region only has been used once. Once again the nose region is the most important region which obtains the highest weight value. Both of [74] and [35] actually emphasize the region least effected by expressions while the face matching is being performed.

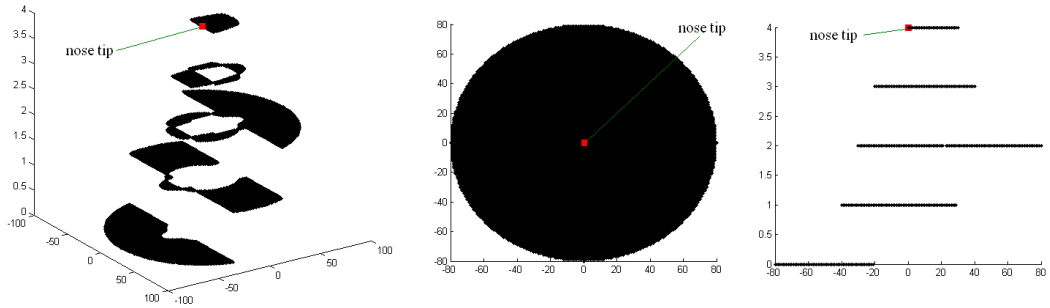


Figure 5.8: *Weight values of different positions according to the segmentation in [74].*

Segmentation method	Ours	Faltermier et al.	Queirolo et al.
Identification rate	97.63%	95.71%	96.78%

Table 5.3: *Identification results by using different segmentation methods.*

In this thesis, on the basis of face segmentation in [35] and [74], we separate the whole face into two main regions shown in figure 5.6. In the upper region which is considered as expression invariant region, the weight of each point depends on its distance to the nose tip. To reduce the complexity and create a simple model, we use several steps to represent the differences of distance. we define the radii as $10mm$, 20, 30, 40, 50, 60, 70 to compute the weight values for each point respectively as shown in figure 5.9. By summing all regions together, the closer a certain position is to the nose tip, the higher weight value it will receive. A weight vector w of all positions can be created. The relationship of weight values and the distance to nose tip is shown in figure 5.10. By using our method to segment the face when performing face recognition experiments, a slightly better performance can be obtained when compared to the results using methods to segment the face in [35] and [74]. Table 5.3 shows the "first vs other" identification experiment by using three segmentation methods.

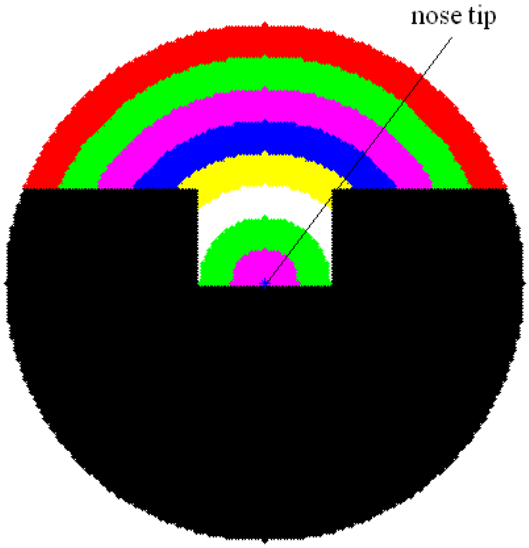


Figure 5.9: *Several circles with different colors segment the expression invariant region. Weight values of each circle region decrease when the distance to the nose tip increases.*

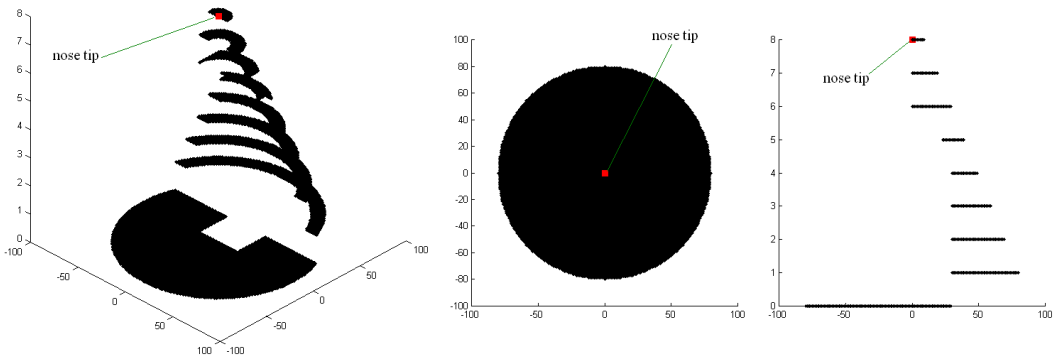


Figure 5.10: *Points have different weight values according to their distances to nose tip in the region lest suffered from expressions.*

To combine the similarity score of different regions, Faltemier et al. [35] and Lu et al. [60] used a sum rule to fuse difference measures. Chang et al. [26]

used both sum rule and product rule in combination. Kittler et al. [53] proved that the sum rule fusion is more resilient to errors than the product rule in combination classifiers. Our accumulating weight method actually is a sum of the number of times that each region is involved in face matching. Every sampling position belonging to one of those regions receives a weight '1'. In a match between two faces, the similarity score at a certain position on the xoy plane will be set to '1' if these two shapes at this position are considered identical, otherwise the score will be set '0'. Then we obtain a similarity vector s containing positions of all points in a particular region. To implement 3D face identification, we only need to calculate the number of positions being considered to have identical shapes. Then the similarity score S can be calculated by equation 5.3. The weight values are applied when the number of identical sampling positions is counted. Equation 5.3 will be modified as:

$$S = \frac{\sum_{i=1}^{N}(w_i \cdot \mathcal{M}_i)}{N} \tag{5.5}$$

Where N is the number of valid points of the query face.

Table 5.4 shows the results when different regions are independently used in face matching and the results by applying accumulating weight in face matching. After the weight vector is applied in the face matching, the identification rate is improved by about 3.4%.

Other works [74] and [35] employed surface alignment algorithms such as the Iterative closest point algorithm (ICP) or Simulated annealing (SA) to match regions they segmented. However most of these regions are overlapped together. To implement a face match, a point may be used many times in

Region:	Identification rate
Full face	93.68%
Expression invariant region	94.24%
Accumulating weight applied	97.63%

Table 5.4: *Rank-one identification rates of "first vs other" experiment by using full face, expression-invariant region only and full face applying accumulating weight respectively*

computation. On the other hand, points are only used once in our method. When we need a region to match, we just apply the weight vector to emphasize the necessary part of the match score vector. Thus, the cost of computation of our method may be much lower than the methods used in [74] and [35].

5.5 Hierarchical face verification

In verification, a match is considered as a correct one when the matching score ms is greater than the threshold th. Otherwise, this match will be reported as incorrect. If the number of matches between different people is N and n is the incorrect match within N matches, which means its similarity score is greater than the threshold th, the FAR of this match can be computed as:

$$FAR = \frac{n}{N} \tag{5.6}$$

The threshold th can be calculated according to a certain FAR. If we define the total number of matches between faces belonging to the same individual as M, the number of matches with a matching score above threshold th can be computed. If we define m as the number of matches above th, then the

verification rate V is calculated as:

$$V = \frac{m}{M} \tag{5.7}$$

By employing the accumulating weighted face matching in verification experiments, a verification rate of 96.35% at 0.1% FAR is obtained in the "neutral vs neutral" experiment. Inspired by the method used in [74], we also use a hierarchical evaluation model to pursue higher verification rates. Firstly, the whole face region is segmented into several parts including nose circle, nose rectangle, upper face, expression invariant region and the accumulating weighted face shown in figure 5.11. Each step of evaluation is to access the match of one region. As shown in figure 5.12, two faces are reported as identical if any step of evaluation generates a positive result.

When a particular FAR is required, we only need to tune the threshold th_i of each region. If we tuned the matches of all regions under a certain FAR, the overall FAR is:

$$FAR = \frac{\sum_{i=1}^{k} n_i}{\sum_{i=1}^{k} N_i} \tag{5.8}$$

Where k is the number of regions.

The overall verification rate V at this FAR will be:

$$V = \frac{\sum_{i=1}^{k} m_i}{\sum_{i=1}^{k} M_i} \tag{5.9}$$

Where k is the number of regions.

Since N_i equal to each other and the FARs of matches of each region have been tuned to a certain value, the overall combined FAR will be the same

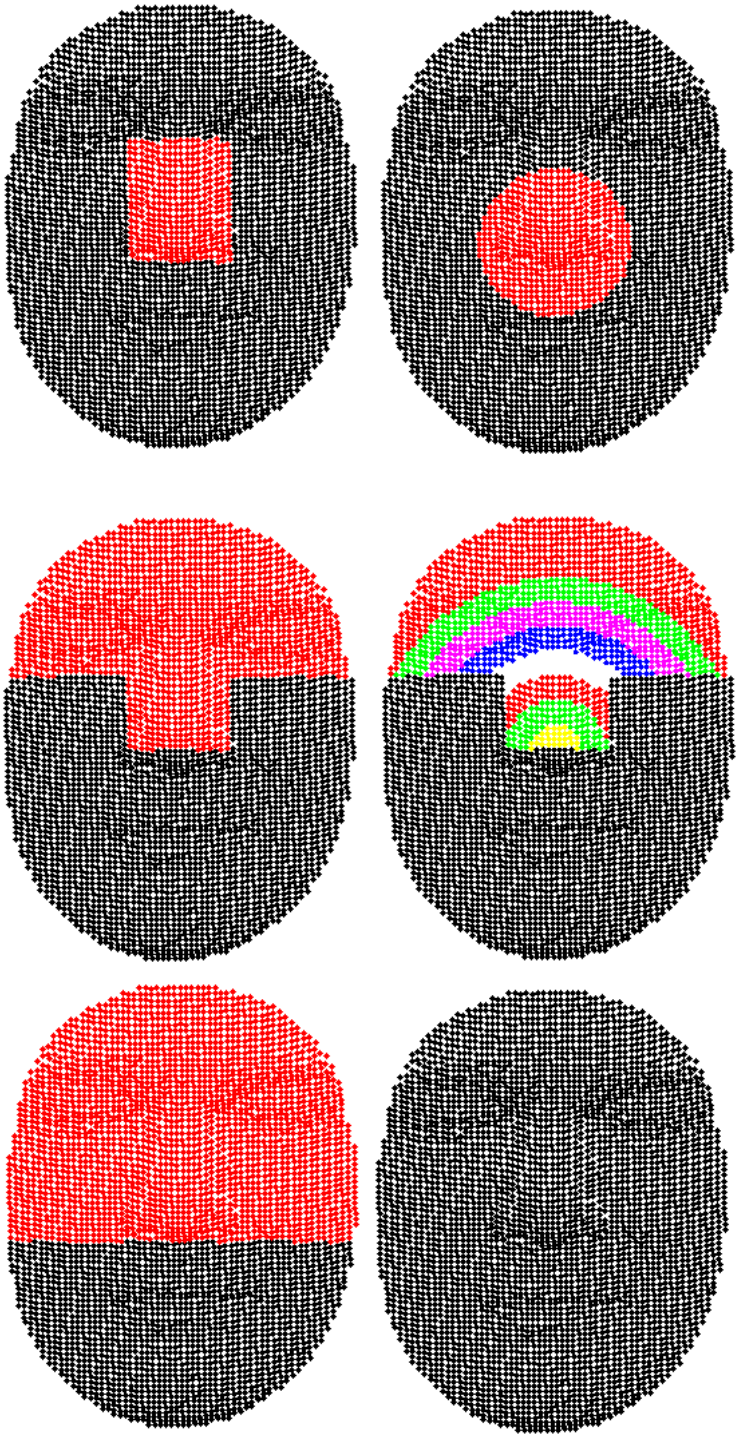


Figure 5.11: *From left to right of first row: two nose regions; from left to right of second row: expression invariant region and accumulating weighted face; from left to right of third row:upper face and full face.*

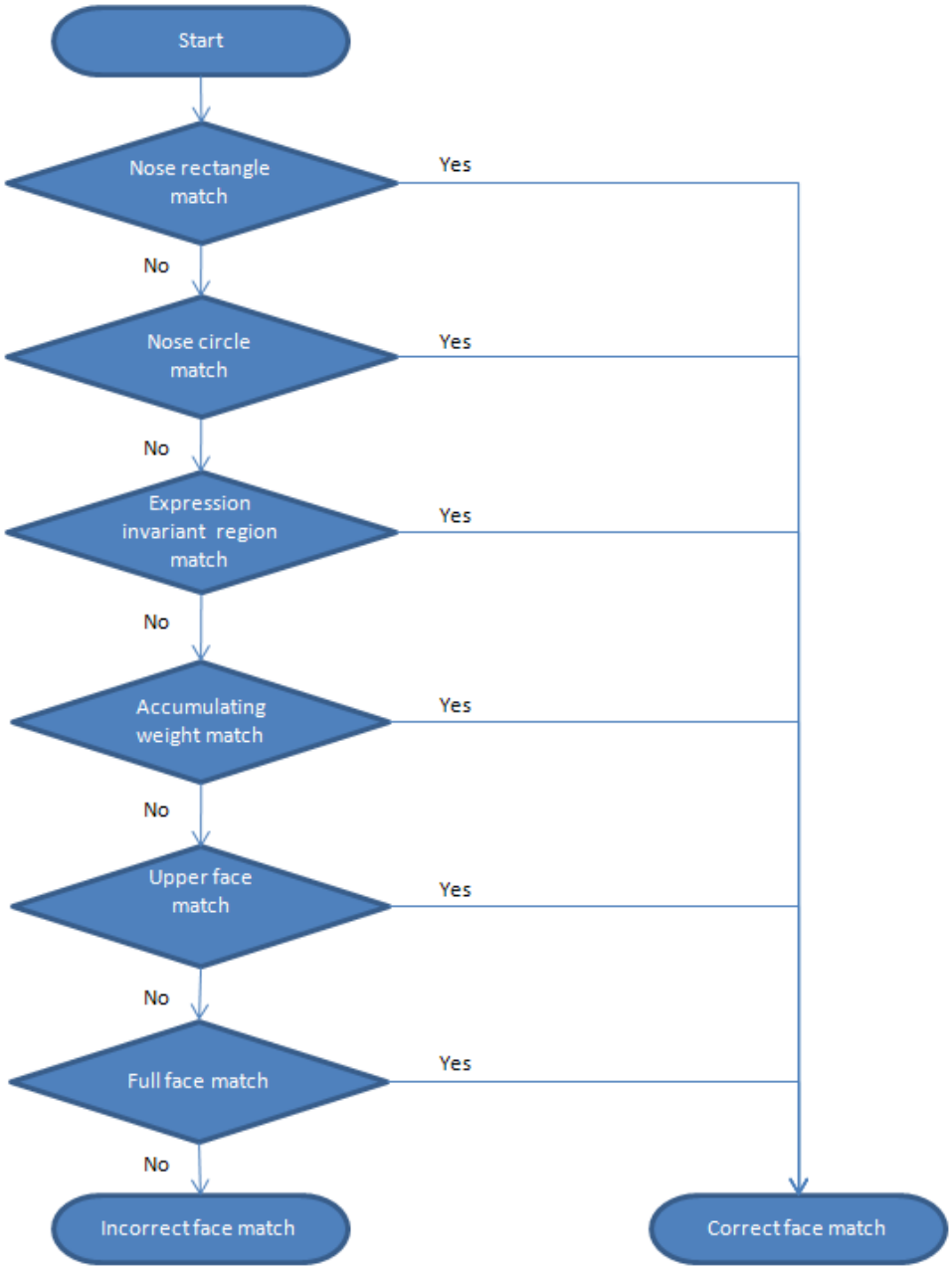


Figure 5.12: *Steps to evaluate and combine matches of different regions.*

Region	Verification
Full face (F)	96.35%
Upper face (U)	95.06%
Weighted region (W)	96.18%
Expression invariant region (E)	92.32%
Nose circle (N2)	95.26%
Nose rectangle (N1)	89.38%

Table 5.5: *Verification rates at* 0.1% *FAR for different regions.*

Region combined	Verification
N1	89.38%
N1 + N2	96.04%
N1 + N2 + E	97.97%
N1 + N2 + E + W	98.44%
N1 + N2 + E + W + U	98.95%
N1 + N2 + E + W + U + F	99.36%

Table 5.6: *Verification rates at* 0.1% *FAR for different region combinations.*

value. For example, if the FAR of every match of different regions is 0.1%, the overall FAR is also equal to 0.1%. A similar method is also used in [35] to fuse the match scores of all regions/sub-regions. In this thesis, if regions are matched separately, the verification rates at 0.1% FAR are listed in table 5.5. By implementing the hierarchical face verification model in the "neutral vs neutral experiment", when we start to combine the result of each region together one by one, the performance can be increased step by step as shown in table 5.6. The best performance 99.36% at FAR 0.1% is obtained when all regions are combined together.

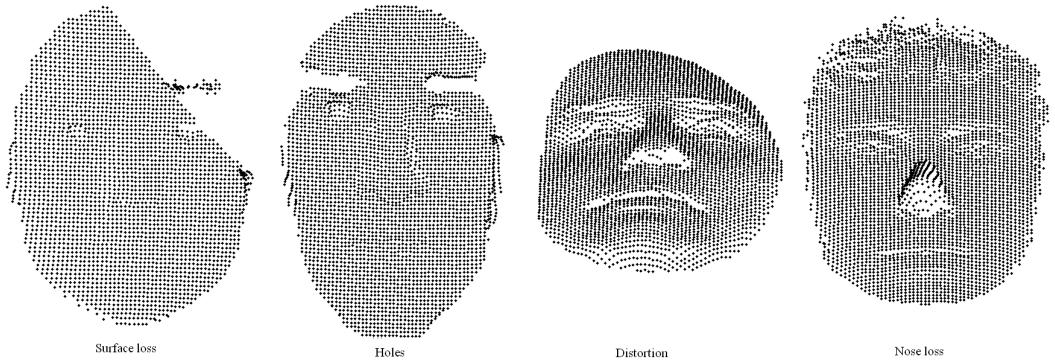


Figure 5.13: *Examples of faces with poor image quality.*

5.6 Experiments results

In this thesis, FRGC v2 database is used as the experiment database. This database has 4007 3D face images from 466 individuals [72]. Each individual has several images with different expressions including neutral, sad, happy, angry, surprised and puffy cheek. However, there are 56 subjects which have only one image per person. In order to conveniently use the results of previous chapters for face detection and face alignment, the resolution of 3D image is 160×120, downsized from 640×480 to be the same size as used in previous chapters. In the FRGC v2 database, the quality of some face images is very poor. Some faces appear to have distortion in the mouth or forehead region due to the sudden head movement during the data-acquisition process; some face images do not have noses. Holes near mouth, eyes and eyebrows could also affect the face recognition performance. Examples of these faces with quality problems are shown in figure 5.13. Thus, based on the selections of faces in [74], the entire FRGC v2 database can be divided into several datasets. Each dataset has different level of difficulty according to noise and expression conditions. Table 5.7 shows the description of these datasets.

Dataset	Description	Number
1	All neutral face images in excellent quality	933
2	All face images other than dataset1	3074
3	All neutral face images in various qualities	2182
4	All non-neutral face images in various qualities	1825
5	All face images in the database	4007

Table 5.7: *Datasets for different levels of difficulties.*

In this thesis, two kinds of experiments are performed to evaluate our technique. The first type of experiment is defined as a number of face identification experiments which concentrate on the rank-one identification rate. The second type of experiment is the face verification experiment in which the experiments' results are quoted as a verification rate at a certain FAR.

5.6.1 Experiment 1:Identification

We define three comparison groups of face identification experiments and each contains several face identification experiments. The first group is to compare the performance under a perfect controlled environment(using neutral and noiseless faces) and the performance under uncontrolled environments (using faces with expressions and various image qualities). The second comparison is to evaluate the effect of expression variations in face identification experiments. The third group is designed to simulate the performance of a real face identification system which contains multiple face images for each subject in the gallery dataset.

Set	Gallery	Query
1	First neutral and noiseless face image (248 faces)	Neutral and noiseless faces (685 faces)
2	First neutral face image (248 faces)	All remaining faces (3759 faces)

Table 5.8: *Gallery and query datasets for each identification experiment sets of the first comparison group.*

5.6.1.1 Face image quality effect

In the first comparison group, the gallery dataset includes 248 faces corresponding to the same number individuals. The remaining neutral and noiseless face images described in table 5.7 and all remaining face images with various qualities are used as the query datasets in the first comparison group. Table 5.8 shows the details of gallery and query datasets. The first set of gallery and query faces is designed to test the ability to find a face with a similar shape to the query face in the gallery dataset under a perfect environment. The query dataset has 685 remaining faces with neutral expression in dataset 1 in table 5.7. The second set is to verify whether the uncontrolled environment such as expressions and noise could affect the identification results. The gallery group of this set is the same as the first set, but the query group includes all remaining faces with various expressions and noise levels(3759 faces) to simulate a real system. The Cumulative Match Curve(CMC) is shown in figure 5.14. We can see from figure 5.14, that image quality and expressions could affect the performance of the face identification system. Using neutral faces with a good face image quality, a rank-one identification of 100% is achieved. Affected by various qualities and expression variations, the identification rate is lowered to 98.21%.

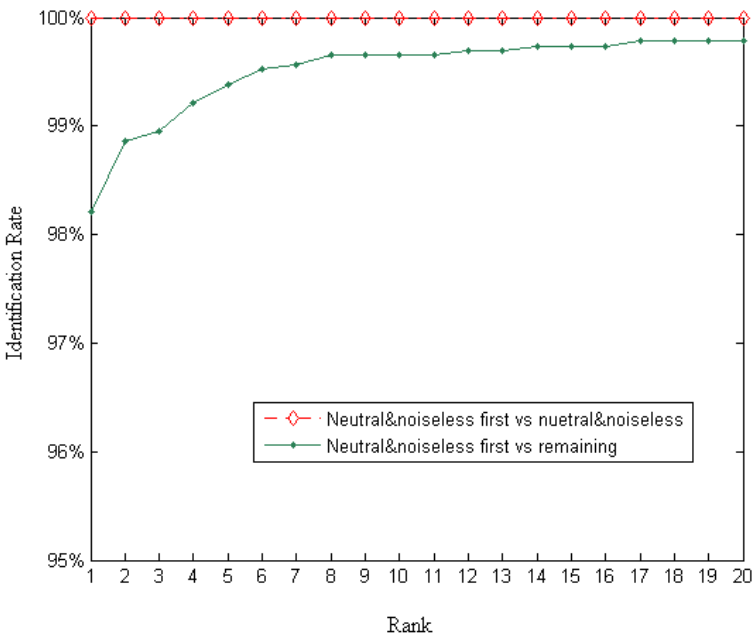


Figure 5.14: *Rank-one identification rate for "Neutral&noiseless first vs Neutral&noiseless" and "Neutral&noiseless first vs remaining".*

5.6.1.2 Effect of expression variations

In the second comparison group of face identification experiments, three sets of gallery and query datasets are classified to express the different ability to handle expression variations and the noise. These gallery and query classifications are listed in table 5.9. In this comparison group, all of the first face images of 465 subjects are used as the gallery dataset. The remaining faces in neutral faces and non-neutral faces are used as the query datasets respectively. The third combination of gallery and query datasets is to simulate the real world identification system. The query dataset contains the remaining faces (3542 faces) in the whole FRGC v2 database. Rank-one identification rates of these experiments are presented in table 5.10. CMC curves of these experiments are

Set	Gallery	Query
1	First faces of each individual(465faces)	Neutral faces(1761 faces)
2	First faces of each individual(465faces)	Non-neutral faces(1781 faces)
3	First faces of each individual(465faces)	All remaining faces(3542 faces)

Table 5.9: *Gallery and query datasets for each identification experiment sets of the second comparison group.*

shown figure 5.15. From the figure 5.15, we can discover that the experiment of neutral expression faces achieves the highest result and the facial expressions could affect the performance of face identification.

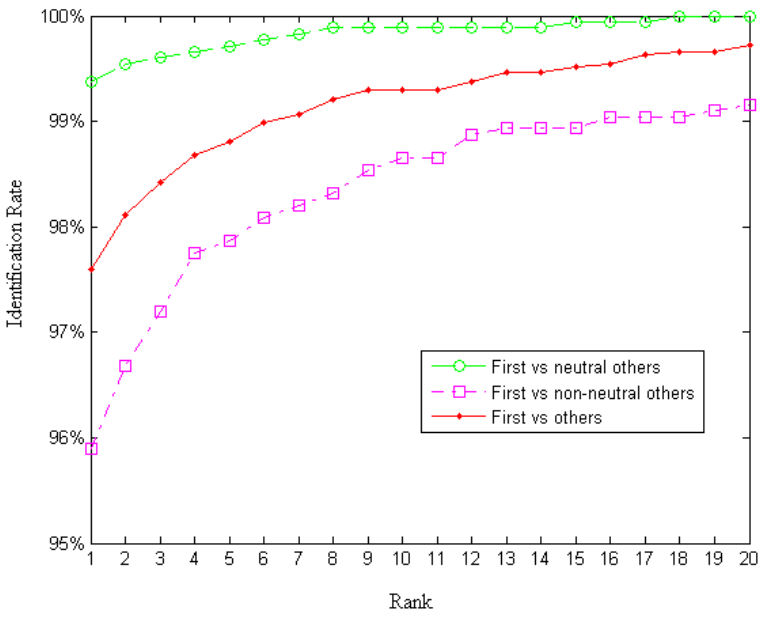


Figure 5.15: *Rank-one identification rates for datasets with different levels of expression variations in the second comparison group.*

Experiment	Identification rate
First vs neutral others	99.38%
First vs non-neutral others	95.90%
First vs others	97.63%

Table 5.10: *Rank-one identification rate of different datasets of the second comparison group.*

Set	Gallery	Query
1	Neutral&noiseless faces (933 faces)	Neutral&noiseless faces (933 faces)
2	Neutral faces (2182 faces)	Neutral faces (2182 faces)
3	Non-neutral faces (1825 faces)	Non-neutral faces (1825 faces)
4	Fall2003 (1893 faces)	Spring2004 (2114 faces)
5	All faces (4007 faces)	All faces (4007 faces)

Table 5.11: *Gallery and query datasets for each identification experiment sets in the third comparison group.*

5.6.1.3 Simulations of real systems

The third comparison group is to simulate the different situations of the real face identification system. There are five face identification experiments in this group. Neutral&noiseless faces, neutral faces, non-neutral faces and all faces are matched with each face in their own datasets to simulate different conditions and the different level of difficulties. The fourth experiment set is recommended by the FRGC [72]. The gallery group was defined as all faces in $fall2003$ dataset in FRGC v2 and the query group includes all faces from $spring2004$ dataset. The faces from $fall2003$ datasets are collected earlier than the $spring2004$ datasets. The time interval between two datasets makes

Experiment	Identification rate
Neutral&noiseless vs neutral&noiseless	99.70%
Neutral vs Neutral	99.95%
Non-neutral vs non-neutral	95.45%
Fall2003 vs Spring2004	96.02%
All vs all	99.32%

Table 5.12: *Rank-one identification rates of the third comparison group.*

the experiments more difficult. This is a common situation in a real face recognition system. The table 5.12 shows the rank-one identification rates of these experiments.

5.6.2 Experiment 2:Verification

This experiment concentrates on the face verification test. It is designed to assess the possibility that a match between faces belonging to the same individual above a threshold under a certain FAR. In this experiment, we designed five sets of gallery and query datasets shown in table 5.13. In the first set, we selected a group of faces which have a neutral expression and an excellent image quality to test the ability to match two faces with identical shape under ideal conditions. The number of faces in this group is 933. This group of faces is defined as the gallery dataset and the query dataset as well. Thus the total number of match is 933×933. The second set is designed to test the face verification ability to deal with neutral expression but with various image qualities. The third set is the verification experiment of non-neutral faces with various image qualities. The gallery and query datasets of the last experiments are all faces in the FRGC v2 database. This experiment is to simulate the situation

Set	Gallery	Query
1	933 neutral expression faces with excellent quality	933 neutral expression faces with excellent quality
2	2182 neutral expression faces	2182 neutral expression faces
3	1825 non-neutral expression faces	1825 non-neutral expression faces
4	1893 faces in Fall2003 dataset	2114 faces in Spring2003 dataset
5	All 4007 faces	All 4007 faces

Table 5.13: *Gallery and query combinations for each verification experiments.*

Set	1	2	3	4	5
Faces	933×933	2182×2182	1825×1825	1893×2114	4007×4007
Matches	$870,489$	$4,761,124$	$3,330,625$	$4,001,802$	$16,056,049$
In-class	$5,911$	$16,754$	$11,177$	$10,824$	$50,927$
Between-classes	$864,578$	$4,744,370$	3319448	$3,990,978$	$16,005,122$

Table 5.14: *The number of matches performed in each experiments.*

and performance in a real face verification system which includes faces with various expression and image qualities.

In a verification experiment, every face in query dataset is matched with the faces in gallery dataset respectively. Thus, the size of the final similarity score matrix is $N \times M$, while N is the number of faces of gallery and M is the number of faces of the query group. Table 5.14 shows the number of matches performed in every set of gallery-query combination.

In those verification experiments, the most computationally complex one is the "all vs all" experiment, because it has a total $16,056,049$ matches per-

Set	Description	Verification rate
1	Neutral faces vs neutral faces both in excellent quality	99.36%
2	Neutral faces vs neutral faces	98.38%
3	Non-neutral faces vs non-neutral faces	89.41%
4	Fall2003 vs spring2004	90.90%
5	All vs all	91.96%

Table 5.15: *The verification rates at* 0.1% *FAR of each experiment.*

formed in this experiment. Within those matches, there are $50,927$ matches performed between the faces belonging to the same person. For the "all vs all" experiment, a verification rate of 91.96% at 0.1% FAR is achieved. Results at 0.1% FAR of all verification experiments are listed in table 5.15. Figure 5.16 shows the Receiver Operating Characteristic (ROC) curves of experiments of "neutral faces vs neutral faces", "non-neutral faces vs non-neutral faces" and "all vs all". We can see clearly the effect of expression variations by comparing these curves. Figure 5.17 presents the ROC curves of "neutral&noiseless faces vs neutral&noiseless faces", "fall2003 vs spring2004" and "all vs all".

5.6.3 Comparison with state of the art face recognition approaches

Other face recognition researchers have published their results on FRGC v2 database. We thus can compare our results with theirs. However, they used different selections of the faces in FRGC database. We only can compare the results using the same or similar selections of faces. There are three experiments in this thesis sharing almost the same gallery-query selections as some

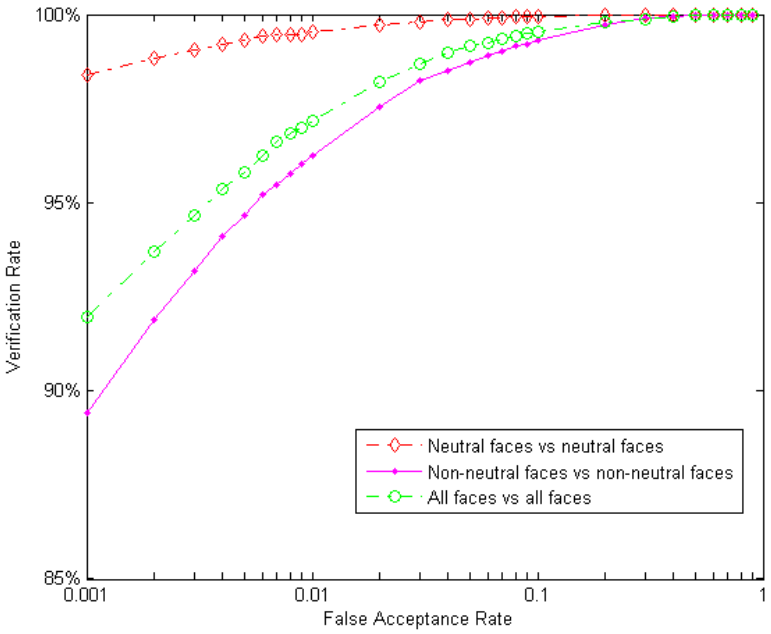


Figure 5.16: *Performance of verification experiment "neutral vs neutral".*

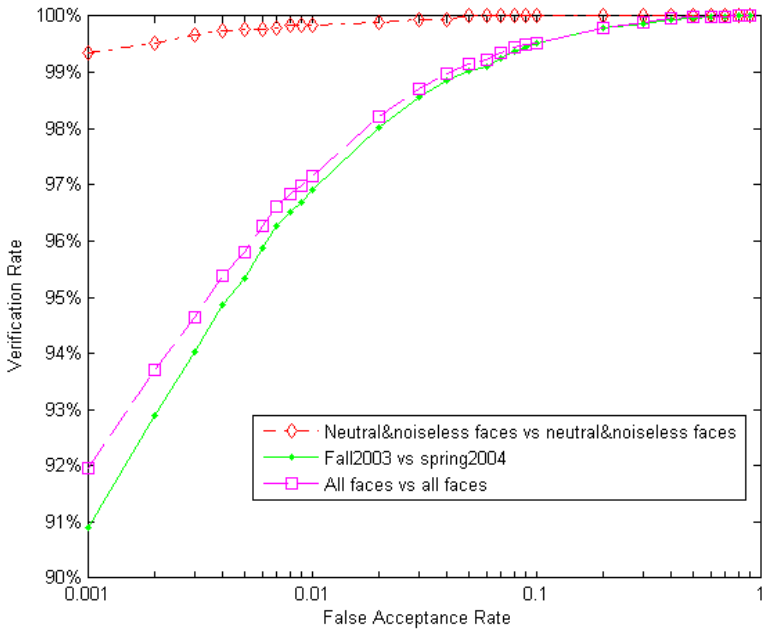


Figure 5.17: *Performance of verification experiment fall2003 vs spring2004.*

Method	Rank-one identification rate
Chang et al. [21]	91.9%
Cook et al. [30]	92.9%
Mian et al. [64]	96.2%
Kakadiaris et al. [50]	97.0%
Faltemier et al. [35]	97.2%
Queirolo et al. [74]	98.4%
Our performance	97.63%

Table 5.16: *The results in "first vs other" identification experiment.*

Method	Verification rate
Mian et al. [64]	86.6%
Maurer et al. [61]	87.0%
Cook et al. [31]	92.31%
Faltemier et al. [35]	93.2%
Queirolo et al. [74]	96.5%
Our performance	91.96%

Table 5.17: *The results in "all vs all" verification experiment.*

works. Table 5.16 presents the rank-one of "First vs others" comparison. Table 5.17 shows the results of "all vs all" together with some state of the art methods and table 5.18 provides the comparison of "fall2003 vs spring2004".

From those tables, we can see in the identification experiments, our result is 97.63%. It is currently the top second result so far. The verification results are on the average level. However, one major concern of the face recognition is the computation time of the algorithm. In the face identification experiment, a query face will compare with a great number of faces in the gallery dataset.

Method	Verification rate
Husken et al. [45]	86.9%
Lin et al. [56]	90.0%
Al-Osaimi et al. [7]	94.1%
Faltemier et al. [35]	94.8%
Queirolo et al. [74]	96.6%
Kakadiaris et al. [50]	97.0%
Our performance	90.90%

Table 5.18: *The results in "fall2003 vs spring2004" verification experiment.*

The computational cost of such a comparison is very sensitive and important to a real face identification system. In [74], [64] and [35], they used a ICP or SA surface alignment/matching algorithms in the face matching procedure. Using such algorithms makes the recognition a very time-consuming task. In [74], the average time for two faces is claimed to be about 11 seconds. By using our methods, the average time of a match between a query face and a gallery face is about 0.0045 second on the configuration: Matlab R2007a, AMD Athlon(tm) 64×2 Dual core Processor 4200+ $2.2GHz$, 3.0GB of RAM. In the "First vs others" identification experiment, the time of a query face matching with 465 gallery faces is about 2 second. Including the time for pre-processing such as nose detection and face alignment, the total identification time for a real world system is still feasible and tolerable.

5.7 Conclusions

This chapter presented a face recognition algorithm based on the pose-invariant surface/shape descriptor. Our algorithm matches two faces by applying an accumulating weight to the match between each pair of corresponding points to emphasize the expression-invariant regions according to the distance to the nose tip. Rank-one identification rates of over 99.38% are achieved in various identification experiments on neutral faces, which presents the ability to correctly identify constant shapes. In the "first vs others" experiment, compared with state-of-the-art face recognition techniques, our approach achieved a rank-one identification rate of 97.63%, which is the second best performance based on FRGC v2 so far. In verification experiments, a verification rate of 98.38% at 0.1% FAR is obtained on neutral faces which is comparable to the best state-of-the-art techniques. The verification rate on non-neutral faces outperforms some state-of-the-art techniques but is not as good as some of the best performance of those approaches. That indicates that although our algorithm has an excellent ability to correctly identify faces even under expression-variations, our algorithm based on shape/surface similarity measurement is to some extent more sensitive to expression variations. The similar shapes produced by the same expression(especially the puffy mouth) could generate a high similarity score which will lower the performance of verification experiments on non-neutral faces. Unlike other approaches based on the surface matching algorithm such as ICP and SA/SIM, our method has much lower computational cost which is very important in the face recognition system.

Chapter 6

Conclusion and future work

6.1 Progress achieved and contribution of this thesis

In the beginning of this thesis, we gave an overview of current face recognition approaches. We reviewed the classical 2D face recognition algorithms and surveyed a number of state-of-the-art 3D face recognition techniques. The existing challenges in face recognition especially in 3D face recognition have been discussed. In the following chapters of this thesis, an automatic 3D face recognition approach has been proposed and implemented including three parts: face detection, face alignment and face recognition.

6.1.1 Pose-invariant and expression-invariant face detection based on the localization of nose tip

In chapter 3, two 3D face shape/surface descriptors have been introduced. Through representing 3D shape by statistical attributes with a number of cir-

cles or shells, the piece of 3D surface at a facial feature position can be stored and trained in a binary neutral network, the CMM. The localization of the position of the same feature in another face therefore can be performed by using a binary k-NN CMM algorithm. Nose tips can be detected by using this algorithm and an identification rate of about 99.95% has been obtained. If two noseless faces in FRGC v2 database are not included, the identification rate is 100%. Even in the noseless faces, the position produced by this automatic feature localization algorithm is very close to the actual position of the nose tip. Additionally, this 3D nose tip localization approach is pose and expression invariant. Compared with other techniques, it gives the best performance achieved within face recognition work based on the FRGC database. Accurate face detection can be implemented based on the results of the nose tip localization. Even noseless faces may also be correctly detected and cropped because the position of the nose tip detected is very close to the center of the face. However, this method only has the ability to localize one nose tip within a 3D image. If there are two people's faces appearing in an face image, this method only can localize one of them, which will result in the neglect of another face.

6.1.2 Integrated expression-invariant face alignment framework

Unlike other work which only uses one face registration or alignment algorithm, in chapter 4 we proposed and performed a combination of three face alignment methods. Firstly, we use a PCA alignment algorithms to roughly correct the pose of faces. Then we correct the position of the nose tip along the ox axis by analyzing the symmetrical characteristics. The head orientation is also fur-

ther aligned particularly in *oy* and *ox* directions by using the symmetry of the human face. Finally, we exploited ICP to match the expression-invariant part of the face to a standard face to correct the head orientation in *oz* direction. By implementing this integrated 3D face alignment, all faces are rotated and aligned to the same coordinate system based on the expression-invariant region to provide a substantial foundation for face recognition. In the in-class and between-class evaluations, our method outperforms the simulations of state-of-the-art ICP-based methods even under expression variations. Our method emphasizes the importance of the expression-invariant region without depending on the eye/forehead localization. Using the face symmetry plane extraction method to segment and localize the expression-invariant region is more reliable and precise. Other range image registration methods which are able to generate a composite rotation matrix also can be used in this framework as well as ICP.

6.1.3 Fast and accurate Face recognition

In chapter 3, we used the 3D shape descriptor to represent a piece of 3D surface and implement the facial feature localization. So, a 3D face consisting of a cloud of points can be represented by a shape descriptor vector. To match two faces, we only need to measure the difference between two shape descriptor vectors. Since a 3D face with only shape information does not have illumination problems and the head orientation problems have been solved in the framework of face alignment, there is only one challenge left in 3D face recognition - expression variations. As the expression-invariant part of a face is segmented out and used in the face alignment stage, in face recognition stage the whole face also can be segmented into different regions and the expression-invariant

region is given a high weight in face matching. Even though we performed the face recognition experiments on faces in the FRGC v2 with a downsized resolution (from 640×480 to 160×120), we obtained a 100% rank-one identification rate in the "neutral first vs neutral" identification experiment and a verification rate of 99.36% at 0.1% FAR in the "neutral vs neutral" verification experiment. For the experiments on faces with expression variations, a rank-one identification rate of 97.63% in the "first vs other" experiment and a verification rate of 91.96% at 0.1% FAR in the "all vs all" experiment have been obtained. This identification rate of "first vs other" is the second higher performance achieved in the FRGC v2 database so far. Moreover, our approach of face matching has a very high computational efficiency. Implemented on a normal desktop computer and in a matlab environment, the computational time to match two faces is about 0.0045 second.

6.1.4 Summary

By implementing localization of the nose tip, the face alignment and the face recognition, we build a high performance 3D face recognition system. Each task of these three stages can be completed automatically. The most important aspect of our system is the highly reliable nose tip localization which can detect the nose tip with an identification rate of almost 100% even on a large face database - the FRGC database. The robustness of this method produces no loss for the following tasks. In the face alignment stage, our integrated method uses symmetry face alignment to precisely segment the express-invariant region avoiding extra errors by implementing another feature localization of related facial features. Compared with other techniques, our face recognition method matches faces more effectively. Our method ignores computationally expensive

algorithms to accurately match face regions, while still producing a relatively high performance especially in the face identification. One limitation of our work is that the FRGC face database used in evaluation only has a limited range of head orientation variations which may not be enough to prove the pose-invariant ability of our approach.

6.2 Future work

In chapter 2, we performed the localization of nose tip and eye corners. Although a high identification performance has been achieved in nose tip detection, the recognition rate of eye corners is not as accurate as the nose tip localization. This is because the shape of the eyes is more subtle than the nose and also the facial expressions could severely affect the shape of eyes. If a further localization to find more facial features such as the eyes, cheek or mouth is required, a better shape descriptor able to precisely represent the shape near the positions of facial features is required. Also how to better deal with the effect of expression has to be considered. As we have shown that our face recognition system has a good ability in face identification even with expression variations, the performance of face verification experiments still has space to improve. One possible solution is to use the original size of images from the FRGC database which can provide more information and detail than the downsized faces. Another interesting piece of work would be to use an extra database like the University of York 3D face database to test and evaluate the three parts of our implementations. This database has relatively more background noise such as wall, desk and even another person. Thus using this database can test the noise-tolerance ability of our system.

References

[1] 3D RMA Face Dabase http://www.sic.rma.ac.be/ beumier/DB3d_rma.html.

[2] BJUT-3D Face Database http://www.bjut.edu.cn/sci/multimedia/mullab/3dface/face_database.htm.

[3] The 3D Face Database, the University of York. http://www-users.cs.york.ac.uk/ nep/research/3Dface/tomh/3DFaceDatabase.html.

[4] Bosphorus 3D Face Database http://bosphorus.ee.boun.edu.tr/.

[5] Frav3d http://www.frav.es/research/facerecognition/frav3d/.

[6] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2d and 3d face recognition: A survey. *Pattern Recognition Letters*, vol. 28, no.14:1885–1906, 2007.

[7] F. Al-Osaimi, M. Bennamoun, and A. Mian. An expression deformation approach to non-rigid 3d face recognition. *Int'l Journal of Computer Vision*, 2009.

[8] J. A. Anderson. A simple neural network generating an interative memory. *Mathematical Biosciences*, vol 14:197–220, 1972.

[9] M. Ankerst, G. Kastenmuller, H. Kiegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. *SSD'99*, pages 207–226, 1999.

[10] J. Austin. Distributed associative memories for high speed symbolic reasoning. *IJCAI'95 Working Notes of Workshop on Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, pages 87–93.

[11] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.

[12] M. Benz, X. Laboureux, T. Maier, E. Nkenke, S. Seeger, F. Neukam, and G. Hausler. The symmetry of faces. *VMV'02*, pages 43–50, 2002.

[13] P. Besl and R. Jain. Invariant surface characteristics for 3d object recognition in range images. *Computer Vision, Graphics, And Image Processing - Lectures notes in computer science, Vol. 201*, 33:33–80, 1986.

[14] P. J. Besl and N. D. Mckay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no.2:239–256, 1992.

[15] C. Beumier and M. Acheroy. Automatic 3d face authentication. *Image and Vision computing*, 18 (4):315–321, 2000.

[16] V. Bevilacqua, P. Casorio, and G. Mastronardi. Extending hough transform to a points' cloud for 3d-face nose-tip detection. *Lecture Notes in Computer Science*, vol 5227/2008:1200–1209, 2008.

[17] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. *Proc, IEEE In-*

ternational conference on Automatic Face and Gesture recognition, pages pp.202–207, 2002.

[18] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. *Proc. ACM SIGGRAPH*, pages pp.187–194., 1999.

[19] W. W. Bledsoe. The model method in facial recognition,. *Panoramic Research Inc., Technical Report PRI15, Palo Alto, CA,*, 1964.

[20] K. Bowyer, K. Chang, and P. Flynn. A survey of approaches and chanllenges in 3d and multi-model 3d+2d face recognition. *In: CVIU*, 101:1–15, 2006.

[21] K. W. Bowyer, K. Chang, and P.Flynn. Adaptive rigid multi-region selection for handling expression variation in 3d face recognition. *IEEE workshop on Face Recognition Grand Challenge Experiments*, pages 157–157, 2005.

[22] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, 16 (1):188–197, 2007.

[23] R. Brunelli, T. Poggio, and I. P. Trento. Face recognition through geometrical features. In *in European Conference on Computer Vision (ECCV*, pages 792–800, 1992.

[24] J. Y. Cartoux, J. T. Lapreste, and M. Richetin. Face authentication or recognition by profile extraction from range images. *Prceedings of the Workshop on Interpretation of 3D Scenes*, pages pp. 194–199, 1989.

[25] K. Chang, K. Bowyer, and P. Flynn. Face recognition using 2d and 3d facial data. *Multimodal User Authentication Workshop*, pages pp. 25–32, 2003.

[26] K. I. Chang, K. W. Bowyer, and P. J. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.28, no. 10:1695–1700, 2006.

[27] C. S. Chua and R. Jarvis. Point signature: A new representation for 3d object recognition. *Internat. J. Computer Vision*, 25 (1):63–85, 1997.

[28] A. Colombo, C. Cusano, and R. Schettini. 3d face detection using curvature analysis. *Pattern Recognition*, vol. 39, number 3:444–455, 2006.

[29] C. Conde, A. Serrano, L. Rodriguez-Aragon, and E. Cabello. 3d facial normalization with spin images and influence of range data calculation over face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,*, vol 3, issue 20-26, 2005.

[30] J. Cook, V. Chandran, and C. Fooks. 3d face recognition using log-gabor templates. *Proc. British Machine Vision Conference*, pages 83–83, 2006.

[31] J. Cook, C. McCool, V. Chandran, and S. Sridharan. Combined 2d/3d face recognition using log-gabor templates. *IEEE Int'l Conf. Video and Signal Based Surveillance*, page 83, 2006.

[32] T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. *Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages pp. 227–232, 2000.

[33] G. J. Edwards, T. F. Cootes, and C. J. Taylor. Face recognition using active appearance models. *Proc. European Conference on Computer Vision*, vol.2:pp. 581–695, 1998.

[34] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press, Palo Alto*, 1978.

[35] T. Faltemier, K. W. Bowyer, and P. J. Flynn. A region ensemble for 3d face recognition. *IEEE Trans. Inf. Forensics Security,*, vol. 3, no. 1:62–73, 2008.

[36] R. Fisher. The use of multiple measurements in taxonomic problems. *In: Annals of Eugenics*, 7:p. 179–188, 1936.

[37] J. D. Foley and A. V. Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Systems Programming Series, 1983.

[38] W. Grimson and T. Lozano-Perez. Model-based recognition and localization from tactile data. *IEEE International Conf. on Robotics, Atlanta, GA*, 1984.

[39] J. C. Hager, P. Ekman, and W. V. Friesen. Facial action coding system. *UT: A Human Face, Salt Lake City*, 2002.

[40] D. O. Hebb. The organization of behavior. 1949.

[41] T. Heseltine, N. Pears, and J. Austin. Three-dimensional face recognition: A fishersurface approach. *Proc. Image Analysis and Processing, ICIAP*, 2004.

[42] T. Heseltine, N. Pears, and J. Austin. Three-dimensional face recognition: An eigensurface approach. *Proc. Internat. Conf. on Image Processiong*, 2004.

[43] C. Hesher, A. Srivastava, and G. Erlebacher. A novel technique for face recognition using range imaging. *Proc. IEEE Int. Symposium on Signal Processing and Its Applications*, 2003.

[44] V. J. Hodge and J. Austin. A binary neural k-nearest neighbour technique. *Knowledge and Information Systems*, vol 8, number 3:276–291, 2005.

[45] M. Husken, M. Brauckmann, S. Gehlen, and C. V. der Malsburg. Strategies and benefits of fusion of 2d and 3d face recogntion. *Proc. IEEE Conf. Computer Vision and Pattern Recognition. IEEE Computer Society*, pages 174–174, 2005.

[46] A. Hyvarinen. Survey on independent component analysis. In *Neural Computing Surveys.*, 1999.

[47] S. Jin, L. R. Robert, and W. David. A comarision of algorithms for vertex normal computation. *The Visual Computer*, vol 21:71–82, 2005.

[48] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:433–449, 1999.

[49] Q. Ju, S. O'keefe, and J. Austin. Binary neural network based 3d facial feature localization. In *IJCNN'09: Proceedings of the 2009 international joint conference on Neural Networks*, pages 843–850, Piscataway, NJ, USA, 2009. IEEE Press.

[50] I. Kakadiaris, G. Passalis, G. Toderici, N. Murtuza, and T. Theoharis. Three-dimensional face recognition in the presence of facial expression: An annotated deformable model approach. *IEEE Trans. Pattern Anal. Mach. Intel.*, vol 29, no.4:671–680, 2007.

[51] T. Kanade. Picture processing system by computer complex and recognition of human faces. *PhD thesis, Kyoto University*, 1973.

[52] T. Kanade. Computer recognition of human faces. *Interdisciplinary Systems Research*, 47, 1977.

[53] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell*, vol.20, no.3:226–239, 1998.

[54] T. Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, vol 21:353–359, 1972.

[55] Y. Lee, K. Park, J. Shim, and T. Yi. 3d face recognition using statistical multiple features for the local depth information. *In:Proc. 16th internat. Conf. Vision Interf.*

[56] W. Y. Lin, K. C. Wong, N. Boston, and Y. H. Hu. 3d face recognition under expression variations using similarity metrics fusion. *Proc. IEEE Int'l Conf. Multimedia and Expo*, pages 727–730, 2007.

[57] X. Lu. Image analysis for face recognition ?a brief survey. *personal notes*, page 36 pages, 2003.

[58] X. Lu, R. Hsu, A. Jain, and B. Kamgar-Parsi. Face recognition with 3d model-based synthesis. *Proc. Internat. Conf. on Biometric Authentication (ICBA)*, pages pp. 139–146, 2004.

[59] X. Lu and A. Jain. Deformation modeling for robust 3d face matching. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages pp.1377–1383, 2006.

[60] X. Lu, A. K. Jain, and D. Colbry. Matching 2.5d face scans to 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no.1:31–43, 2006.

[61] T. Maurer, D. Guigonis, I. Maslov, B. Pesenti, A. Tsaregorodtsev, D. West, and G. Medioni. Performance of geometrix activeid 3d face recognition engine on the frgc data. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 154–154, 2005.

[62] K. Messer, J. Matas, J. Kittler, J. Lttin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *In Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77, 1999.

[63] A. Mian, M. Bennamoun, and R. Owens. Automatic 3d face detection, normalization and recognition. *Interantional Symposium on: 3D Data Processing Visualization and Transmission*, vol 0:735–742, 2006.

[64] A. Mian, M. Bennanmoun, and R. Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11:1927–1943, 2007.

[65] A. Mian, M. Bennaoun, and R. Owens. Matching tensors for pose invariant automatic 3d face recognition. *Proceedings of Computer Vision and Pattern Recognition*, 2005.

[66] A. B. Moreno and A. Sanchez. Gavabdb: A 3d face database. *n: Proc. 2nd COST275 Work- shop on Biometrics on the Internet, Vigo (Spain)*, 2004.

[67] T. Nagamine, T. Uemura, and I. Masuda. 3d facial image analysis for human identification. *International Conference on Pattern Recogntion*, pages 324–327, 1992.

[68] G. Pan, Y. Wang, Y. Qi, and Z. Wu. Finding symmetry plane of 3d face shape. *Proc. of 18th International Conference on Pattern Recognition (ICPR'06)*, vol 3:1143–1146, 2006.

[69] T. Paratheodorou and D. Ruechert. Evaluation of automatic 4d face recognition using surface and texture registration. *Proc. Sixth IEEE Internat. Conf. on Automatic Face and Gesture Recogntion*, pages 321–326, 2004.

[70] N. Pears, T. Heseltine, and M. Romero. From 3d point clouds to pose-normalised depth maps. *International Journal of Computer Vision*, Volume 89, Numbers 2-3,:152–176, 2010.

[71] J. P. Phillips, H. Moon, A. S. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Machine Intell.*, pages pp/ 1090–1104, 2000.

[72] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. *Proc. IEEE Conf. Computer Vision and Pattern Recognition,*, pages 947–954, 2005.

[73] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, and M. S. C. L. Schott. Frvt 2006 and ice 2006 large-scale experimental results. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32 no. 5:831–846, 2010.

[74] C. C. Queirolo, L. Silva, O. Bellon, and M. P. Segundo. 3d face recognition using simulated annealing and the surface interpenetration measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, February 2010 (vol. 32 no. 2):206–219, 2010.

[75] S. Rizvi, P. Phillips, and H. Moon. The feret verification testing protocol for face recognition algorithms. *Technical Report NISTIR 6218 Nat'l Inst. Standards and Technology*, 1998.

[76] S. Rizvi, P. Phillips, and H. Moon. The feret verification testing protocol for face recognition algorithms. *Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages pp. 48–53, 1998.

[77] M. Romero and N. Pears. 3d facial landmark localisation by matching simple desciptors. *2nd IEEE Int. Conf. Biometrics: Theory, Applications and Systems*, 2008.

[78] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. *Proc. of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[79] B. Scholkopf, A. Smola, and K. Muller. Nonolinear component analysis as a kernel eigenvalue problem. *Neural Computation*, vol. 10, no. 5:pp. 1299–1319, 1998.

[80] M. P. Segundo, C. Queirolo, O. R. Bellon, and L. Silva. Automatic 3d facial segmentation and landmark detection. *Image Analysis and Processing, ICIAP*, pages 431–436, 2007.

[81] G. Shakhnarovich and B. Moghaddam. Face recognition in subspaces. *Handbook of Face Recognition, Eds. Stan Z. Li and Anil K. Jain, Springer-Verlag*, page 35, December 2004.

[82] L. Silva, O. R. Bellon, and K. L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:762–776, 2005.

[83] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Internat. Conf. on Automatic Face and Gesture Recognition*, 25 (12):1615–1618, 2003.

[84] F. Stein and G. Medioni. Structural hashing: Efficient three dimensional object recognition. *Proceedings of Computer Vision and Pattern Recognition*, pages 244 – 250, 1991.

[85] H. T. Tanaka, M. Ikeda, and H. Chiaki. Curvature-based face surface recognition using spherical correlation-principal directions for curved object recognition. *Proc. 3rd Internat. Conf. on Face & Gesture Recognition*, pages pp. 327–377, 1998.

[86] L. Torres. Is there any hope for face recognition? *Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 21–23, 2004.

[87] F. Tsalakanidou, S. Malassiotis, and M. G. Strintzis. Face localization and authentication using color and depth images. *IEEE Transactions on Image Processing*, 14 (2):152–168, 2005.

[88] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, Vol. 3, pp. 72-86, 1991.

[89] Y. Wang, C. Chua, and Y. Ho. Facial feature dectection and face recogntion from 2d and 3d images. *Pattern Recognition letters*, 23:1191–1202, 2002.

[90] Y. Wang, G. Pan, and Z. Wu. Sphere-spin-image: A viewpoint-invariant surface representation for 3d face recognition. *Proc. Internat. Conf. on Computational Science, Lecture Notes In Computer Science*, Vol. 3037:pp. 427–434, 2004.

[91] L. Wiskott, J. Fellous, N. Kruger, and C. von der Malsburg. Face recogntion by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no.7:pp. 775–779, 1997.

[92] C. Xu, S. Z. Li, T. Tan, and L. Quan. Automatic 3d face recognition from depth and intensity gabor features. *Pattern Recognition*, 42(9):1895–1905, 2009.

[93] C. Xu, T. Tan, Y. Wang, and L. Quan. Combining local features for robust nose location in 3d facial data. *Pattern Recognition Letters*, vol 27, issue 13:1487–1494, 2006.

[94] C. Xu, Y. Wang, T. Tan, and L. Quan. Depth vs. intensity: Which is more important for face recognition? In *ICPR (1)*, pages 342–345, 2004.

[95] M. H. Yang. Face recognition using kernel methods. *Advances in Neural Information Processing Systems, T. Diederich, S. Becker, Z. Ghahramani, Eds.*, vol. 14,:8 pages, 2002.

[96] L. Zhang, A. Razdan, G. Farin, J. Femiani, M. Bae, and C. Lockwood. 3d face authentication and recognition based on bilateral symmetry analysis. *The Visual Computer*, 22:43–55, 2006.

Appendix A

Face Recognition Grand Challenge 3D face database

Face Recognition Grand Challenge 3D face database(FRGC) has a large number of individuals and face images including pose and expression expression variations. The face images of the FRGC database was segmented into training and validation partitions. The data captured in the $2002-2003$ academic year is the training partition. In the training partition, there are two datasets: still image dataset and 3D dataset. The 3D image training dataset consists of 943 faces. One face has one 3D channel file describing 3D information and one 2D channel image containing texture information. Face images in the validation partition were collected during the fall of 2003 and the spring of 2004. The validation partition is also called as FRGC v2 database. The total number of face images is 4007 from 466 subjects. C. Chaua [74] reported that the label of subject 04643 is actually 04783. Thus, the number of subjects in the validation partition is 465 rather than 466. Table A.1 shows the details of the FRGC 3D face database. In the FRGC v2 database, each subject contains $1-22$ face images. Each subject has several face images with different expressions includ-

Figure A.1: *Examples of different expressions in FRGC database.*

ing neutral, sad, happy, angry, surprise and puffy cheek. Figure A.1 shows the examples of different expressions of a subject. Percentage of different races are 22% asian, 68% white and 10% others. There are 57% male and 43% subjects in this dataset. The range of age of subjects are: $18-22(65\%)$, $23-27(18\%)$ and $28+(17\%)$.

Partition	Faces	Subject	Dataset
Training	943	275	FRGC v1
Validation	4007	466(465*)	FRGC v2

Table A.1: *Details of FRGC 3D face database.*

The 3D images were taken under controlled illumination conditions appropriate for the Vivid 900/910 sensor [72]. The Minolta Vivid 900/910 series is a structured light sensor which takes 640×480 3D sampling and a registered color image. Subjects were asked to stand or sit approximately 1.5 meters from the sensor. In the FRGC, 3D images include both range (3D) and texture (2D) channels. Each 3D face has two files which store 2D and 3D information respectively. Figure A.2 and figure A.3 show two examples of the 2D and 3D

Figure A.2: *An example of the 2D face image [72].*

channel files of a face in the FRGC 3D database. The Vivid sensor captured the texture channel after the acquisition of the 3D channel, which may cause poor registration between the 2D and 3D channels. The 2D channel file is a color image file containing sRGB values in Portable Pixel Map format ('.ppm'). The resolution of the 2D image is 640×480. The 3D channel file is a '.abs' file which contains the x,y and z values in 3D space of each pixel in the 2D image file. The format of the 3D channel file is shown in figure A.4. The first two rows are the resolution in x and y directions. Then there is a row containing the flag value which represents which pixel is a valid face pixel. When the flag value is '1', then the corresponding pixel is a valid pixel. After the flag row, there are three rows containing values of x, y and z. Values of an invalid pixel are set to '-9999999'.

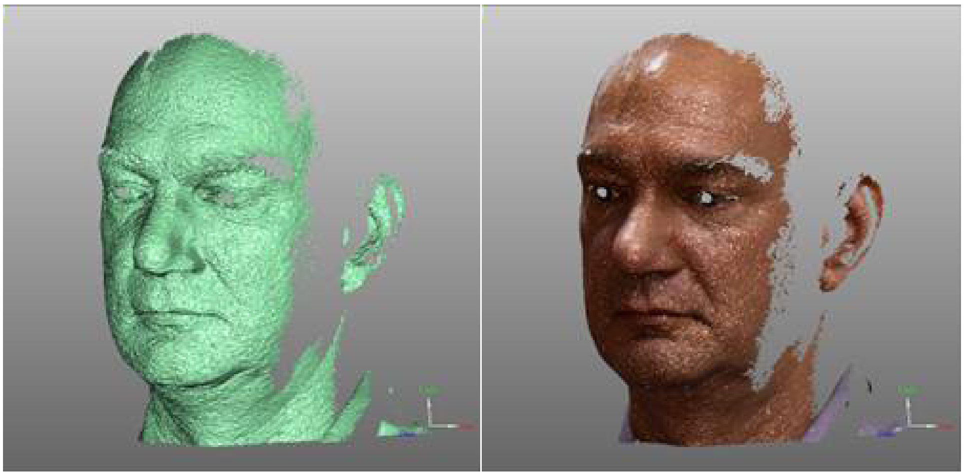


Figure A.3: *An example of the 3D channel image [72].*

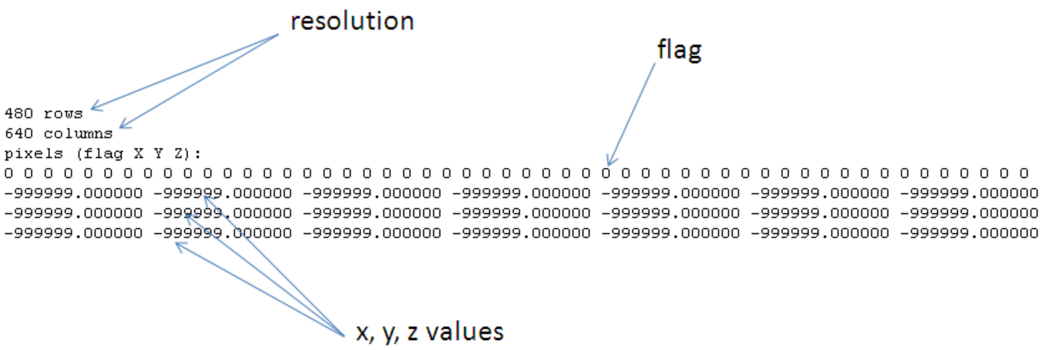


Figure A.4: *The format of 3D channel file.*

Appendix B

Iterative Closed Point (ICP) algorithm

The first and also the most important step in ICP is to compute the nearest distance between every point in the target to a point in the model. For example, the distance between two points is denoted by the following equation:

$$D_{istance}(p_1, p_2) = \| p_1 - p_2 \| \tag{B.1}$$

x_{p1}, y_{p1}, z_{p1} are the three-dimensional values of point $p1$ and x_{p2}, y_{p2}, z_{p2} are the three-dimensional values of point $p2$.

Given a point t_j in the target set of points T, the Euclidean distance of t_j to the model set of points M is:

$$D_{istance}(t_j, M) = \min_{i \in 1..n} D_{istance}(t_j, m_i) \tag{B.2}$$

Where m_i is a point in $M (m_i \in M)$.

Thus, if we define $y \in M$, C as the closest point operator and Y is the set of closest points to M, using equation B.2 we can find the corresponding closest point in model M:

$$Y = C(T, M) \tag{B.3a}$$

$$Y \subseteq M \tag{B.3b}$$

After each point's corresponding closest point in the model is computed, given Y we can calculate the alignment:

$$(Ro, Tr, d) = \Phi(T, Y) \tag{B.4}$$

where Ro is the rotation matrix and Tr is the translation matrix. d is the error distance between T and M.

When the alignment is repeated, T will be updated to be:

$$T_{new} = Ro(T) + Tr \tag{B.5}$$

In [14], Besl et al. used a quaternion-based algorithm to yield the least squares rotation and translation for the data in two and three dimensions and used. They recommended use of the singular value decomposition(SVD) method in any $n > 3$ dimensional application. In this thesis, we are considering face data in three dimensions. Therefore, we are able to use the quaternion-based algorithm stated as follows:

We consider point clouds of the target and the model as two matrices: $T(x, y, z)$, $M(x, y, z)$, then the cross covariance matrix $Cov(T, M)$ between these two ma-

trices can be calculated from following equation.

$$Cov(T, M) = \frac{1}{N_t}\sum_{i=1}^{N_t}[(T - \mu_t)(M - \mu_m)^T] = \frac{1}{N_t}\sum_{i=1}^{N_t}(TM^T) - \mu_t\mu_m^T \quad \text{(B.6)}$$

Where μ_t and μ_m are the mean values of T and M respectively.

$$\mu_t = \frac{1}{N_t}\sum_{i=1}^{N_t} T \;\; and \;\; \mu_m = \frac{1}{N_t}\sum_{i=1}^{N_t} M \quad \text{(B.7)}$$

After we have $Cov(T, M)$, let it be:

$$C = \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{bmatrix}$$

Define a matrix A:

$$A = Cov(T, M) - Cov(T, M)^T \quad \text{(B.8)}$$

Let A be represented as follows:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

Then define a vector D as follows:

$$D = [a6, a7, a2] \quad \text{(B.9)}$$

A scalar S is defined as follows:

$$S = c1 + c5 + c9 \quad \text{(B.10)}$$

If a matrix T is defined as follows:

$$T = (C + C^T) - S \cdot I \tag{B.11}$$

where I is a 3×3 identity matrix.

then:

$$T = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & t_9 \end{bmatrix} \tag{B.12}$$

We define the quaternion matrix Q as follows:

$$Q = \begin{bmatrix} S & a_6 & a_7 & a_2 \\ a_6 & t_1 & t_2 & t_3 \\ a_7 & t_4 & t_5 & t_6 \\ a_2 & t_7 & t_8 & t_9 \end{bmatrix} \tag{B.13}$$

We can use the quaternion matrix to calculate the composite rotation matrix. The first step is to find the maximum eigenvalue and its corresponding eigenvector for Q. The corresponding eigenvector of Q is defined as a row vector $[q_1, q_2, q_3, q_4]$. In the previous steps, we know the mean vectors of T and M are $\mu_t = [\overline{x_t}, \overline{y_t}, \overline{z_t}]$ and $\mu_m = [\overline{x_m}, \overline{y_m}, \overline{z_m}]$. Then we define two new vectors as $U_1 = [\overline{x_t}, \overline{y_t}, \overline{z_t}, 1]$ and $U_2 = [\overline{x_m}, \overline{y_m}, \overline{z_m}, 1]$. The transformation matrix R can be defined as follows:

$$R = \begin{bmatrix} R_1 & R_2 & R_3 & R_4 \\ R_5 & R_6 & R_7 & R_8 \\ R_9 & R_{10} & R_{11} & R_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.14}$$

where:

$$R_1 = q_1^2 + q_2^2 - q_3^2 - q_4^2 \tag{B.15a}$$

$$R_2 = 2 \cdot (q_2 \cdot q_3 - q_1 \cdot q_4) \tag{B.15b}$$

$$R_3 = 2 \cdot (q_2 \cdot q_4 + q_1 \cdot q_3) \tag{B.15c}$$

$$R_4 = 0 \tag{B.15d}$$

$$R_5 = 2 \cdot (q_2 \cdot q_3 + q_1 \cdot q_4) \tag{B.15e}$$

$$R_6 = q_1^2 + q_3^2 - q_2^2 - q_4^2 \tag{B.15f}$$

$$R_7 = 2 \cdot (q_3 \cdot q_4 - q_1 \cdot q_2) \tag{B.15g}$$

$$R_8 = 0 \tag{B.15h}$$

$$R_9 = 2 \cdot (q_2 \cdot q_4 - q_1 \cdot q_3) \tag{B.15i}$$

$$R_{10} = 2 \cdot (q_3 \cdot q_4 + q_1 \cdot q_2) \tag{B.15j}$$

$$R_{11} = q_1^2 + q_4^2 - q_2^2 - q_3^2 \tag{B.15k}$$

$$R_{12} = 0 \tag{B.15l}$$

After the composite rotation matrix is generated, we can use this matrix to implement the fitting process by rotating T to model M. A matrix L is defined to update the composition rotation matrix to repeat the fitting process:

$$U_2 = R \cdot U_1 + L \tag{B.16}$$

$$L = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{bmatrix}$$

L is used to update the composite rotation matrix:

$$R_4 = l_1 \tag{B.17a}$$

$$R_8 = l_2 \tag{B.17b}$$

$$R_{12} = l_3 \tag{B.17c}$$

The mean squared error distance of F to M can be calculated:

$$e = \frac{1}{N} \sum_{i=1}^{N} \parallel R \cdot t_i - m_i \parallel \tag{B.18}$$

The iteration continues until $e_{k+1} - e_k < \tau$, where τ is a preset threshold.