3D FACE RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BÜLEND ÜSTÜN

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2007

Approval of the thesis

"3D FACE RECOGNITION"

Submitted by **BÜLEND ÜSTÜN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen

Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmen

Head of Department, **Electrical and Electronics Eng.** _____

Prof. Dr. . Uğur Halıcı

Supervisor, **Electrical and Electronics Eng.** _____

Examining Committee Members

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Eng, METU _____

Prof. Dr. Uğur Halıcı
Electrical and Electronics Eng, METU _____

Associate Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Eng, METU _____

Assist. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Eng, METU _____

Mehmet Dikmen (M.S.)
Computer Eng, Başkent University _____

Date: 05.12.2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Bülend ÜSTÜN
Signature :

ABSTRACT

3D FACE RECOGNITION

Üstün, Bülend
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Uğur Halıcı

December 2007, 87 pages

In this thesis, the effect of registration process is evaluated as well as several methods proposed for 3D face recognition. Input faces are in point cloud form and have noises due to the nature of scanner technologies. These inputs are noise filtered and smoothed before registration step. In order to register the faces an average face model is obtained from all the images in the database. All the faces are registered to the average model and stored to the database. Registration is performed by using a rigid registration technique called ICP (Iterative Closest Point), probably the most popular technique for registering two 3D shapes. Furthermore some variants of ICP are implemented and they are evaluated in terms of accuracy, time and number of iterations needed for convergence. At the recognition step, several recognition methods, namely Eigenface, Fisherface, NMF (Nonnegative Matrix Factorization) and ICA (Independent Component Analysis) are tested on

registered and non-registered faces and the performances are evaluated.

Keywords: ICP variants, Eigenface, Fisherface, NMF, ICA

ÖZ

3 BOYUTLU YÜZ TANIMA

Üstün, Bülend
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Uğur Halıcı

Aralık 2007, 87 sayfa

Bu tezde, çakıştırma işleminin üç boyutlu yüz tanıma algoritmalarından birkaçı üzerine etkisi incelenmiştir. Girdiler üç boyutlu nokta kümesi şeklinde olup tarayıcıdan kaynaklanan gürültüler mevcuttur. Gürültü filtreleme işlemi sayesinde veriler daha pürüzsüz bir hale getirilmiştir. Yüz yüzeylerinin çakıştırılması amacıyla katı bir çakıştırma metodu olan ve muhtemelen üç boyutlu iki şeklin çakıştırılması amacıyla en sık kullanılan metot olan, Döngülü Yakın Nokta (DYN) metodu kullanılmıştır. Ayrıca DYN metodu için öne sürülmüş birkaç değişken doğruluk ve yakınsama için gerekli olan zaman ve döngü sayısı temel alınarak incelenmiştir. Tanıma aşamasında özyüzler, Fisherface, Bağımsız Bileşenler Analizi (BBA), Negatif Olmayan Matris Ayrıştırma (NOMA) metotları hayata geçirilmiştir. Çakıştırılmış ve çakıştırılmamış veriler üzerinde, tanıma algoritmalarının performansları incelenmiştir.

Anahtar Kelimeler: DYN değişkenleri, Özyüzler, Fisherface, Negatif Olmayan Matris Ayrıştırma, Bağımsız Bileşenler Analizi

To my family

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Uğur Halıcı who encouraged and supported me at every level of this work.

I would like to also express my thanks to Assistant Prof. İlkay Ulusoy for her precious support.

Finally, I would like to thank my family for their infinite love and support. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

2D: Two dimensional

3D: Three dimensional

CMC: Cumulative Match Characteristic

FLD: Fisher Linear Discriminant

FRGC: Face Recognition Grand Challenge

ICA: Independent Component Analysis

ICP: Iterative Closest Point

LDA: Linear Discriminant Analysis

MSE: Mean Square Error

NMF: Nonnegative Matrix Factorization

PCA: Principal Component Analysis

ROC: Receiver Operating Characteristic

SVD: Singular Value Decomposition

TPS: Thin Plate Spline

CHAPTER 1

INTRODUCTION

Authentication systems of the future are believed to operate on biometrics which means the systems will be able to identify the people from their biological characteristics. These types of systems are being used more and more in security systems and consumer electronics.

Fingerprints have been the most common biometric over 100 years [54]. As the technology develops other types of biometrics have emerged such as hand geometry, iris, retina, signature, vein patterns, face, and DNA. They have several advantages and disadvantages when compared to others in terms of accuracy, speed, intrusiveness, environmental tolerance, and cost. For instance retina recognition is the most accurate biometrics however it is also the most expensive one.

When compared to other biometrics one of the advantages of face recognition is its non-intrusive nature. To illustrate, fingerprint and iris based recognition systems perform better than face recognition systems however these methods necessitate the collaboration of the subjects. For a typical face recognition system, subjects may not even be aware of the scanner. Moreover in case a face recognition system fails to identify a person, security personnel may authorize that person for access by examining the individual's image in the database.

Big majority of the face recognition systems works with 2D images. These systems use intensity values of the pixels of the images for

extracting features and making decisions. Although these techniques can achieve good performances under controlled conditions, their performance can degrade dramatically when pose, illumination, and scale parameters change.

3D techniques are thought to be advantageous in terms of these factors since the 3D shape of the face does not change due to these factors. Advances in computer vision make it possible to design face recognition systems with lower error rates. Also by the recent developments better sensor systems provide cheaper and accurate 3D data which increases the availability of 3D data.

Finally evaluations like FRGC (Face Recognition Grand Challenge) and FRVT (Face Recognition Vendor Test) [55] aims to improve face recognition systems by encouraging and channeling more researchers into this area.

FRVT 2006 results [51] show that FRGC has reached its aim since the results are much better than FRVT 2002 results. Three biometrics were compared in FRVT 2006: recognition from very-high resolution still face images, 3D face images, and single-iris images. Recognition performances were reported to be comparable for all three biometrics. Moreover some of the algorithms performed better than humans.

In this thesis, FRGCv1 database in which 3D face data is given as point clouds is used as inputs to the system. After cropping the region of interest, the remaining face data is filtered in order to remove the spikes and holes and to obtain a smoother surface. Prior to storing a face data, it is registered to the average face model which is built by, as the name implies, taking the average of all the faces in the database. Several face recognition algorithms are implemented and their performances are evaluated on registered and non-registered face data.

The rest of this thesis is organized as follows: Chapter 2 gives a background about common terms in face recognition area, algorithms

used in this thesis, and literature survey. Chapter 3 presents the results of the experiments performed. Finally in Chapter 4 conclusions and possible future works are summarized.

CHAPTER 2

BACKGROUND

In order to evaluate performances of face recognition systems, some common terms are employed [048]. *Identification* task is the case, when the person to be identified is known to be in the database and by comparing this person's new image with all the images in the database, similarity scores are calculated. After sorting them, the most similar person in the database is given as output. If the system correctly identifies the person, when only the first similar match is considered, this is called a "top match". If the system correctly identifies the person within the most similar n number of images from the database, this is called "Rank n score". A curve which shows the relation between the rank and the number of correct identifications is referred as Cumulative Match Characteristic (CMC) curve. This curve gives an idea about how close a system is from the correct match. The ideal case would be having 100% top match score. However if a good n can be determined by examining the CMC curve, top n matches can be given to security personnel to make the final decision. An example for a CMC curve is given in Figure 2.1.

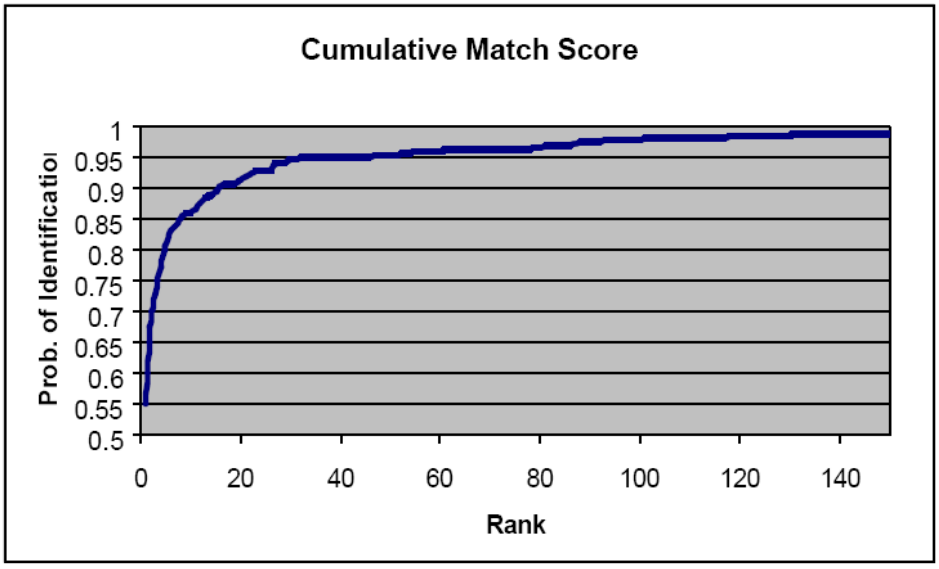

Figure 2.1 A CMC curve

In a *verification* task, the person makes a claim to be an identity in the database. In fact, he may even not have an image in the database. By comparing this person's image with the claimed identity's image, a similarity score is obtained. If this score is above a threshold, the system accepts that the person is who he claimed to be. Otherwise, if the similarity score is below than the threshold, the system will reject the person. There are two errors that a system can make for a verification task. Firstly, the person may make an errant claim to be an identity, and after calculating the similarity score the system may think that the person says the truth although he does not. This is called a "false accept." Secondly, the person make appropriate claim regarding his identity, but the calculated similarity score is lower than the threshold. The system will reject the person although he is saying the truth. This is called a "false reject." These two errors reversely related. If threshold is increased to lower FAR (False Acceptance Rate), this will in turn increase FRR (False Rejection Rate) and decrease the probability of verification. A plot

that related these parameters is called ROC (Receiver Operating Characteristic). It is also possible to plot this curve, putting the probability of correct verification onto y axis. The probability of correct verification can be computed by subtracting FRR from 1. An example ROC curve is given in Figure 2.2.

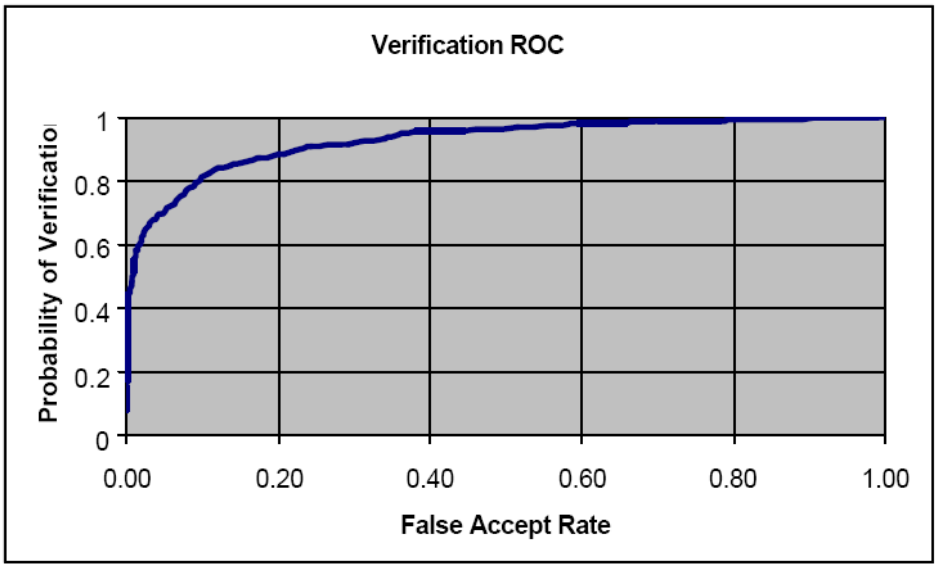

Figure2.2 A ROC curve

Dimensionality Reduction

Consider a set of images all having size N by M dimensions. The pixels in each image can be represented by an NxM size vector, in other words by a point in an NxM dimensional *image space*. Conditioned that pixel values are ordered in the same manner, similar images will be nearer to each other in the image space. However, making an analysis in an NxM-dimensional space is very difficult.

On the other hand face images have some common properties which make it possible to represent them in fewer dimensional spaces.

For example left part of the face is almost symmetric to the right part and every image has two eyes, a nose and a mouth in the middle. Furthermore value of a pixel is strongly related with the values of its neighbors which makes pixel-wise sampling unnecessary. Faces will not be distributed randomly among the image space. Instead great majority of the image space will not be used due to reasons explained above. As a result, it would be possible to represent them using fewer dimensions. This fewer dimensional space is called *face space*.

2.1 PREPROCESSING

2.1.1 FRGC DATABASE

All references made to the FRGC (Face Recognition Grand Challenge) database in this thesis are always related to part v.1 of the database. Faces are scanned with a Minolta Vivid 3D laser scanner. This device can produce 3D shape and color information registered to each other (Figure 2.3). The FRGC database contains 943 2D images and 943 3D data belonging to 275 different persons. People have different number of images in the database. The numbers are given in Table 2.1.

Although great majority of 2D-3D data pairs are registered to each other, there are two completely unrelated pairs. These pairs are removed from database. Although there are some badly registered pairs in the data base, that is to say, there are small translation differences about 10 pixels between the color image and depth image. These pairs are not removed from the database. All the experiments are carried out by the remaining 941 pairs.

Table 2.1 Images per person in the database

# of images per person	# of persons	2D image	3D data
1	78	78	78
2	32	64	64
3	46	138	138
4	33	132	132
5	28	140	140
6	30	180	180
7	15	105	105
8	13	104	104

The resolution of 2D images is 480x640 pixels. Color information is provided by these images which have '.ppm' file extensions. Each pixel has 24 bits per pixel to code color in RGB (Red-Green-Blue) space (Figure2.4).

3D data has 480x640x4 number of elements. For each pixel position in the registered 2D image there are 4 values in corresponding position (i,j) of 3D data: x, y, z, and validity. x y, and z values shows the position of the point in the space. Validity is a flag to determine whether the point corresponds to subject whose image is taken or not. If the point does not correspond to an object in the view, then there is no information relating to this point and x, y, and z values are assigned as -999999.

There are some problems with the 3D data files. Firstly, the points where the laser scanner cannot get any reflections are regarded invalid. An example is given in Figure 2.5. Invalid points are black where as white points are valid. Some parts in the faces especially eyes, eyebrows are problematic. Although these points are visible, scanner cannot provide depth information.

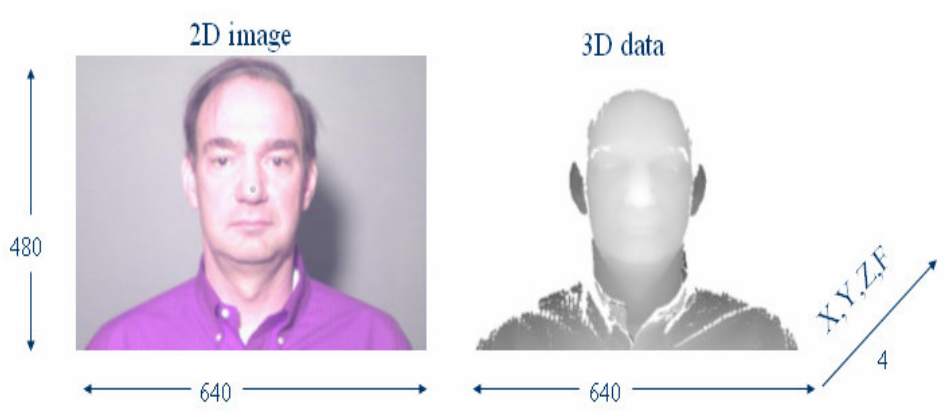

Figure2.3. 2D image and 3D data

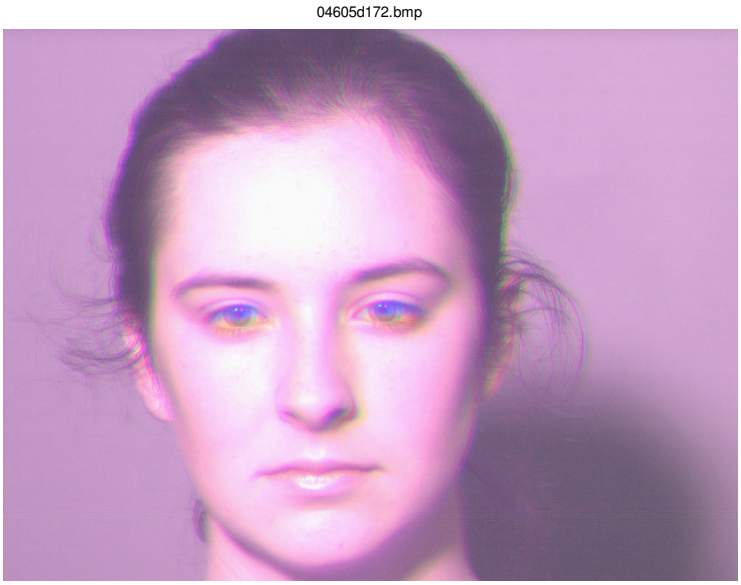

Figure2.4. bmp converted 2D image file

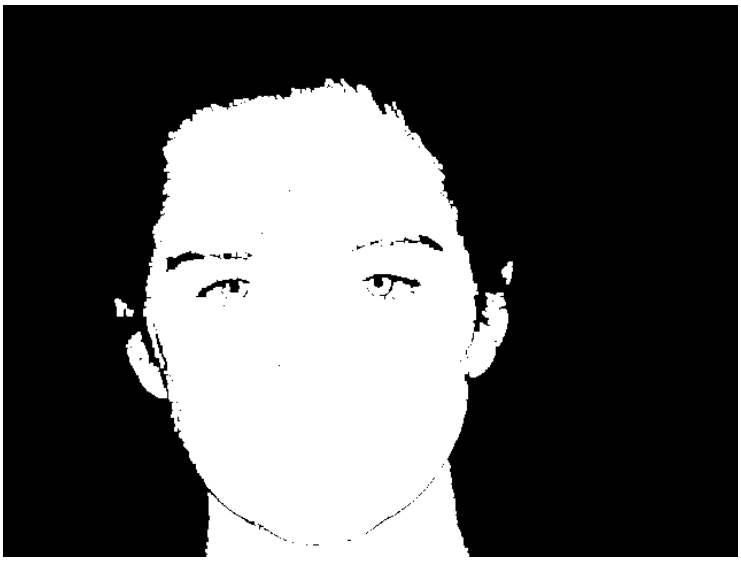

Figure2. 5 Validity flag points in the 3D data file

Figure2.6 Depth (or Range) image constructed from 3D data

Unfortunately not every valid point has a proper x, y, and z values. Points within a few pixels of neighborhood of the invalid points tend to have improper x, y, and z values, such as eye corners.

All images in the FRGC database are taken from front with minor in-depth pose variations. However, significant translation and scale variations are present in the images. In some of the images, rotation and expression variations are also present.

2.1.2 PREPROCESSING OF IMAGES

8 landmarks namely outer eye corners, inner eye corners, nose tip, middle points of upper and lower lips, and chin are marked manually on 2D images as shown in the Figure 2.7.

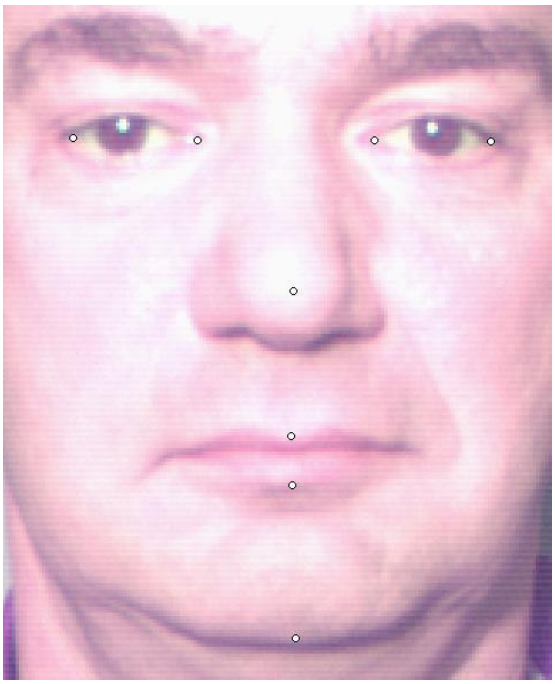

Figure 2.7 Landmarks marked on a face image

Since some of the landmark locations are especially problematic such as eye corners many of them tend to have invalid pixel correspondences. As a result, only the nose tip is used in this thesis. Images having invalid pixel for the nose tip, are still marked because during the preprocessing of images, nose tip are carried to the origin by subtracting nose tips' x, y, and z values from all the pixels in the image. If the nose tip is invalid, x, y, and z values will have -999999 and subtracting this value from every pixel will raze the image and perturb the system.

After a 3D data is positioned so that the nose tip is at origin, filtering procedures for noise starts. Almost all the 3D face representations suffer from erroneous data points due to current 3D sensor technology which makes a noise removal step necessary. The distances of all the pixels with respect to nose tip are calculated and those having bigger distances than a threshold value are deleted. Then, impulse-like noises in the image are eliminated by noise filtering (Figure 2.8)

After the holes in the images are filled with interpolation, faces are placed in a grid centered at origin. By considering all the images in the database, a proper choice is made to determine the starting and ending points of the grid.

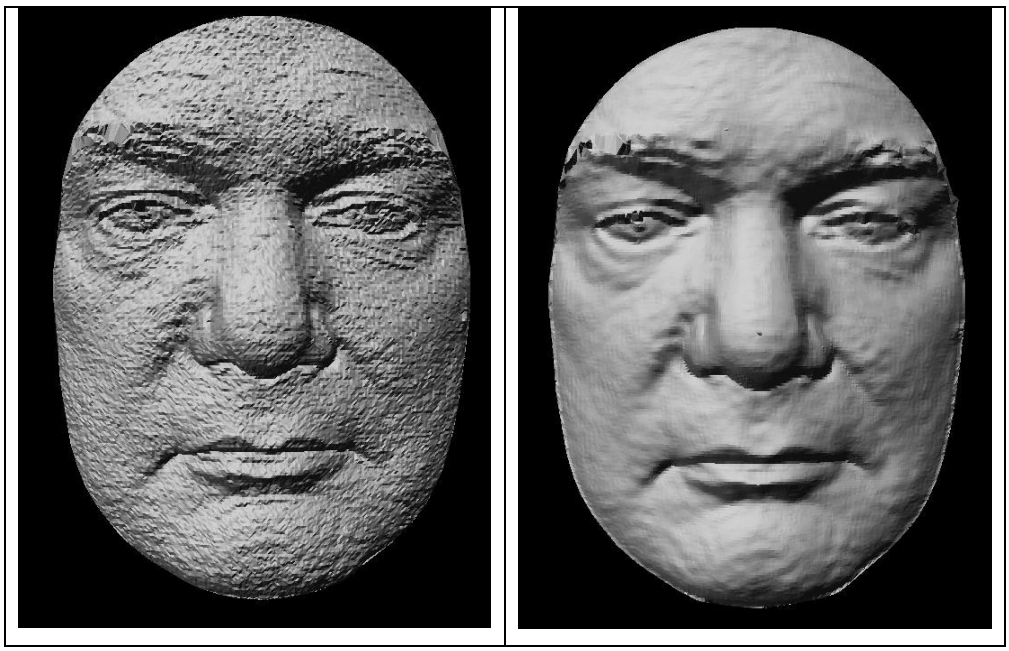

Figure 2.8 Effect of noise filter

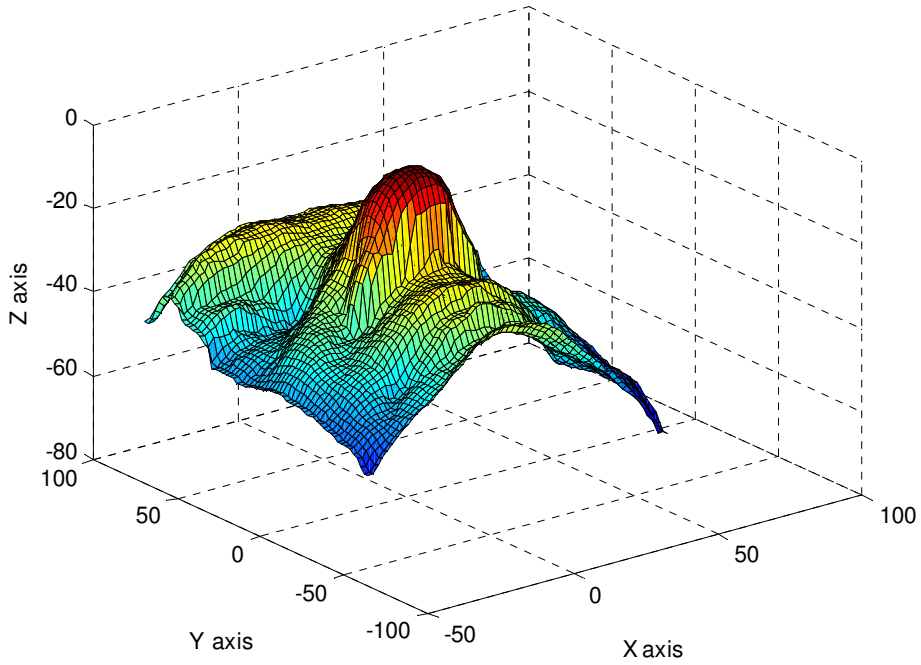

Figure2. 9 Face image placed on an x-y grid

Since the faces are all placed on the same grid, and all nose tips are at the origin (Figure 2.9), z values can be used as features. By applying this method to all the faces in the database, depth images of size nxn are generated.

In this thesis n is chosen as 21, 41, 61, 81, 101, 121, and 161 to see the effect of the grid size on the algorithms.

2.2 ITERATIVE CLOSEST POINT

Iterative Closest Point (ICP) [2] is, as the name implies, an iterative algorithm which registers rigidly a 3D set to a 3D model while minimizing the mean square error (MSE) between them. After the registration is complete, the given data set is in the best alignment with the model set with respect to the chosen error metric.

ICP is a rigid transformation applying the same rotation and translation to all points of the data set.

If the correct correspondences between the data set and the model shape are known, the registration parameters can be found easily. (Figure 2.10a)

$$Data_{registered} = R * Data_{nonregistered} + T \tag{2.1}$$

where R is the rotation matrix having size of 3x3 and T is the translation matrix having size of 1x3.

However, finding the correct correspondences is not always trivial. Thus, ICP considers the closest points (Figure 2.10b) and calculates the Rotation and Translation matrices iteratively. The algorithm repeats the steps below, until the error falls below a preset threshold or the difference between consecutive error values fall below a threshold.

- Compute the closest points on the model set for every point in the data set
- Compute the registration parameters (R, rotation matrix and T, Translation matrix)
- Apply the registration to the data set
- Calculate the error between the model and the registered set

The algorithm will result in a good position if two set, data and model set, are close enough. (Figure 2.10c)

ICP always converges monotonically to a local minimum [2] but a global minimum is not guaranteed. In order to end up in the global minimum, either a good initial estimate should be given to the algorithm or after trying several initial conditions the one resulting in the lower error rate would be chosen.

There are also some papers that perform ICP on local regions. This technique has two main advantages. Firstly, non-rigid deformations of a face such as expressions cannot be expressed by standard ICP. Secondly, matching local regions is much faster than matching all the face data [42, 37].

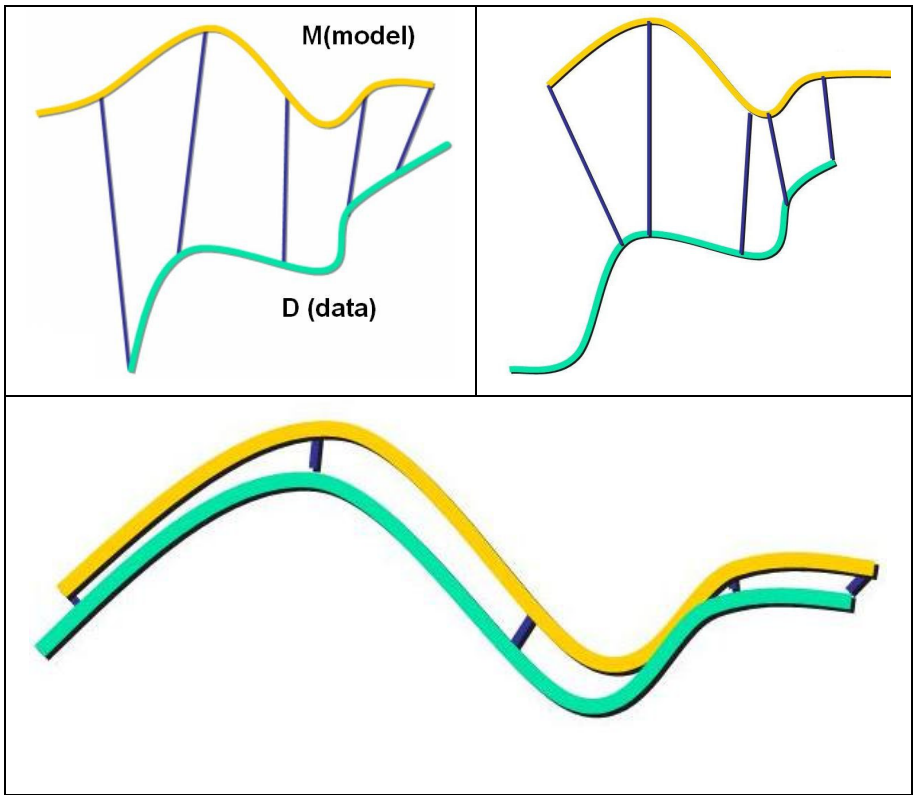

Figure 2.10a (top left) If the correct correspondences are known, which is not possible in general, registration can be handled without any problems.2.10b (top right) ICP considers the nearest points in the model. 2.10c (bottom) If starting point is good, the registration will give a good result.

In every loop, the closest point in the model is calculated for each point in the data set. After that, registration can be computed by several means. In [52] 4 registration algorithms are evaluated and Singular Value Decomposition (SVD) has found to be better in general. In this thesis SVD approach given in [53], is used to compute registration algorithms as explained in the following.

2.2.1 SINGULAR VALUE DECOMPOSITION

The error that is to be minimized is the mean square error of the Euclidian distance between each point in the data set and its registered correspondence.

$$\mathrm{E} = \frac{1}{n}\sum_{i=1}^{n} \| \, p_i^{\,model} - (Rp_i^{\,data} + T) \, \|^2 \tag{2.2}$$

In order to find R and T firstly, the centroid of each set is calculated and subtracted from the analogous set to have two sets having the same centroid.

$$\mathrm{centroid}_{\mathrm{data}} = \sum_{i=1}^{n} \; p_i^{\,data} \tag{2.3}$$

$$\mathrm{centroid}_{\mathrm{model}} = \sum_{i=1}^{n} \; p_i^{\,model} \tag{2.4}$$

$$\mathrm{d_i}^{\mathrm{model}} = \mathrm{p_i}^{\mathrm{model}} - \mathrm{centroid}_{\mathrm{model}} \tag{2.5}$$

$$\mathrm{d_i}^{\mathrm{data}} = \mathrm{p_i}^{\mathrm{data}} - \mathrm{centroid}_{\mathrm{data}} \tag{2.6}$$

Data points are centered at their own centroid. These sets are sorts of normalized sets. Rotation parameter will be calculated so that one centroid will rotate and match the other.

$$\mathrm{E} = \sum_{i=1}^{n} \| \, d_i^{\,model} - (Rd_i^{\,data}) \, \|^2 \tag{2.7}$$

Once R is found, T can be found as

$$\mathrm{T} = \mathrm{centroid}_{\mathrm{model}} - R * centroid_{\mathrm{data}} \tag{2.8}$$

If the equation (2.2) is expanded, it can be seen that the eigenvectors of the covariance matrix formed by the data set and the model set, are

the rotation axis which maps one set into the other. This covariance matrix is decomposed by SVD to get

$$H = UWV^T \tag{2.9}$$

and

$$R = V\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(VU^T) \end{bmatrix}U^T \tag{2.10}$$

In a general SVD case similar to equation (2.10), U is the matrix of eigenvectors of HH^T; V is the matrix of eigenvectors of H^TH. W is a diagonal matrix containing the square root of the eigenvalues of U and V. Note that eigenvalues of HH^T and H^TH are same.

To sum up

- The centroid of the data and the model is found. The centroid of each set is subtracted to normalize them.
- Covariance matrix of the normalized sets is formed.
- Using SVD on the covariance matrix, R is found and then T is found.

At each iteration R is found and updated according to equations:

$$R_i = R_{update} * R_{i-1} \tag{2.11}$$

$$T_i = R_{update} * T_{i-1} + T_{update} \tag{2.12}$$

where R_i is the resulting rotation matrix of i^{th} iteration calculated using the rotation matrix of the previous iteration (R_{i-1}) and the rotation matrix computed at that iteration (R_{update}). T_i is, again, the resulting translation matrix calculated by the translation matrix of the previous iteration (T_{i-1}) and the translation matrix computed at that iteration (T_{update}).

2.2.2. VARIANTS OF THE ICP ALGORITHM

Many different variants are proposed since the introduction of ICP. In [3] ICP variants are classified in six stages:

- Selection of sample points
- Matching of the points to the other mesh
- Weighting the corresponding pairs
- Rejecting certain pairs
- Assigning an error metric
- Minimizing the error metric

In this thesis, the same classification will be followed and the effect of some variants will be examined. Different than [3] the scenes in this thesis are 3D faces which are either represented by depth images or point clouds.

Selection of points: The original method [2] proposes using all the available points always. However in order to speed-up the process either uniform subsampling of the available points [5] or random sampling of the points at each iteration can be chosen as proposed in [58]. These are the variants that will be examined in this thesis. Also in [3] maximizing the distribution of normals is proposed. Moreover, there are some other methods such as selecting points with high intensity grey level gradient [57]. This method is preferable for improving the alignment if per-sample color or intensity information is available.

Matching points: The original paper suggests finding the closest point in the other mesh [2]. One of the other methods is normal shooting proposed in [7]. In other words starting from the source point, normal direction is followed until the destination surface is reached. Another

method is to project the source point onto the destination mesh [49], which is called reverse calibration. In order to increase the probability of correct matching, a neighborhood of the matched point is searched in terms of a given compatibility metric based on color or angle between normals also proposed in the literature.

Rejecting Pairs: A threshold may be defined by a user and the corresponding points having point-to-point distances greater than the threshold can be rejected. Another method is to reject worst n% of pairs, n=10 is used in [3]. Also it is possible to reject pairs whose point-to-point distance is larger than 2.5 times the standard deviation. These variants will be examined in this thesis. In addition, rejecting pairs that are not consistent with neighboring pairs was proposed by [59]. Finally the pairs containing points on mesh boundaries [5] can be rejected.

In this thesis, our baseline ICP algorithm uses all the points in both meshes (no subsampling or rejection is performed), computes point-to-point distances between pairs and weights are given uniformly to every pair. As model, 61-by-61 depth image obtained by averaging all the 941 faces is used.

Firstly, a face model is constructed by averaging all the faces in the database (Figure 2.11). Then, every 3D face image read from FRGC database is registered to the model face and the resulting depth images form new database on which all the experiments are performed. In other words, the faces in the database are already registered with the average face model. Each test image given to the system is registered to average face by ICP before recognition is performed.

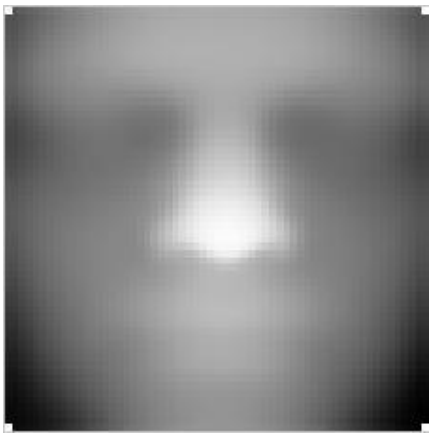

Figure 2.11 Mean image

2.3 BACKGROUND ON RECOGNITION ALGORITHMS

Firstly, by ICP the 3D data is registered such that the nose tip is at origin and the new 3D data is converted to depth images. All the algorithms considered in this chapter are applied to depth images. Each algorithm is implemented in Matlab. Except ICA algorithm, which is obtained from internet [56], codes are written in Matlab. The downloaded code was also written in Matlab and some small modifications were made to the code in order to permit it operate with the database properly.

2.3.1 EUCLIDIAN DISTANCE

After the registration step is complete, all the images lies on the same x-y grid. Thus differences between z values can be used to recognize the test image.

$$Error(i_{database}, j) \sum_{y=-Ny}^{Ny} \sum_{x=-Nx}^{Nx} (I_{i\,database}(x,y) - I_{j_{test}}(x,y))^2 \quad (2.13)$$

where Nx and Ny are chosen to be 30 since the size of the depth images are 61x61, and I(x,y) is the value of the depth image at x,y.

2.3.2 EIGENFACES

Principal Component Analysis (PCA), also called Karhunen-Loève transform, is one of the most common techniques used in different areas including the field of pattern recognition. It was first introduced in the early 1990's and became very popular since then. It was first used in face recognition in [5]. PCA was also used many times for representing, detecting, recognizing, and compressing face images for decades.

In order to find the vectors which best represent the distribution of the data set, a covariance analysis is performed. The eigenvectors of the covariance matrix are calculated and by eliminating the vectors having the smallest eigenvalues, the dimension of the data is reduced.
The reason why this method is called Eigenface method is that when PCA is performed over a set of faces, the eigenvectors look like faces.

Eigenface approach can be summarized as follows

- Get the training images ($\Gamma_1, \Gamma_2, ...\Gamma_M$). M is the number of faces in the training set.
- Calculate the mean ($\Psi = \sum_{n=1}^{M} \Gamma_n$). n is the number of pixels in an image.
- Subtract it from every image in the training set ($\Phi_n = \Gamma_n - \Psi$)
- Calculate the covariance matrix C=AA^T where $A = [\Phi_1 \Phi_2 .....\Phi_M]$

- Find its eigenvectors, choose first k of them having the largest eigenvalues and project every image in the training set to face space and calculate the coefficients

$$weight(projected,i) = \sum_{i=1}^{k} (\Gamma_{test} - \Psi) \cdot eigenvector_i \quad (2.14)$$

$$\Gamma_{projected} = \sum_{i=1}^{k} weight(projected,i) \cdot eigenvector_i \quad (2.15)$$

- If recognition is to be performed, project the test depth image to face space after subtracting mean and by considering the coefficients, find the image in the database having the closest coefficients minimizing the equation

$$Error = \sum_{i=1}^{k} (weight(projected,i) - weight(training,i))^2 \quad (2.16)$$

In general C is so big that finding its eigenvectors becomes very difficult. The number of faces in the training set (M=941) is smaller than the number of pixels in a depth image (n=3721). Solving C, having size of 3721x3721, for eigen-analysis is computationally expensive. In [5] Turk and Pentland propose a way to determine the eigenvectors of the covariance matrix C.

Consider a matrix L= A^TA. The eigenvectors of this matrix will satisfy the equation

$$A^T A v_i = \mu_i v_i \quad (2.17)$$

Multiplying both sides by A from left, we obtain $AA^T A v_i = \mu_i A v_i$ which means Av_i are the eigenvectors of C. Therefore firstly the eigenvectors of L is found to reduce the calculations. Finding the eigenvectors of L is simpler since L has a size of 941x941.

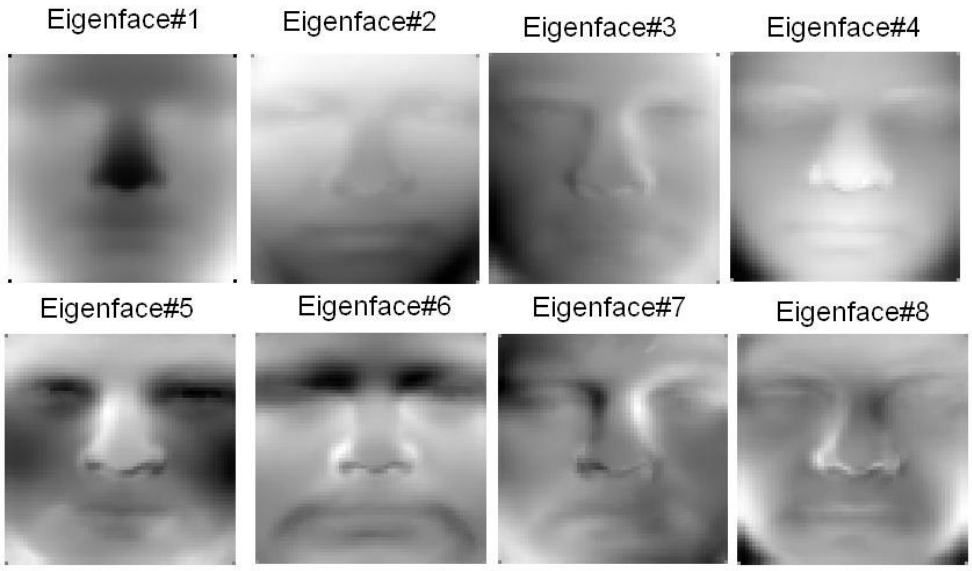

Figure 2.12 Sample eigenfaces, corresponding to 10 biggest eigenvalues

2.3.3 NONNEGATIVE MATRIX FACTORIZATION (NMF)

The aim of NMF is to find W and H matrices for a given V matrix such that

$$V \approx W * H \tag{2.18}$$

In order to use NMF for face recognition [9], images in the training set are placed to columns of V. V has a size of nxm where n is the number of pixels, and m is the number of images in the training set. $W \in R^{nxr}$, $H \in R^{rxm}$ and r is chosen smaller than n in order to reduce the dimension and compress the data.

Firstly, PCA is applied to reduce the dimensionality of the data. Before the dimension-reduced training set is given as V to NMF algorithm, all the samples are added by a constant so that minimum value of V becomes zero since NMF requires V, W, and H to be non-

negative. Therefore, during the experiments in this thesis, minimum value of V is added to all of the samples and test images.

Non-negativity constraint enables parts-based representation since only additive combinations are allowed.

W and H are initiated with random values. Reconstruction error function is defined as

$$E(W,H) = \| V - WH \|^2 \tag{2.14}$$

Convergence is assured if W and H are updated according to the equations below [8]:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T WH)_{a\mu}} \qquad \text{a=1,…r and μ=1,…m} \tag{2.15}$$

$$W_{ia} \leftarrow W_{ia} \frac{(VH^T)_{ia}}{(WHH^T)_{ia}} \qquad \text{i=1,…n and a=1,…r} \tag{2.16}$$

After factorization is finished, the transpose of H matrix is directly used for similarity matching.

V=WH and H=pinv(W)V where pinv symbolizes pseudo-inverse. Columns of H can be thought of representation of training images in fewer dimensional space and representation of each image is a column of H (Figure 2.13). This column is used to determine the similarity between a test image and a training image.

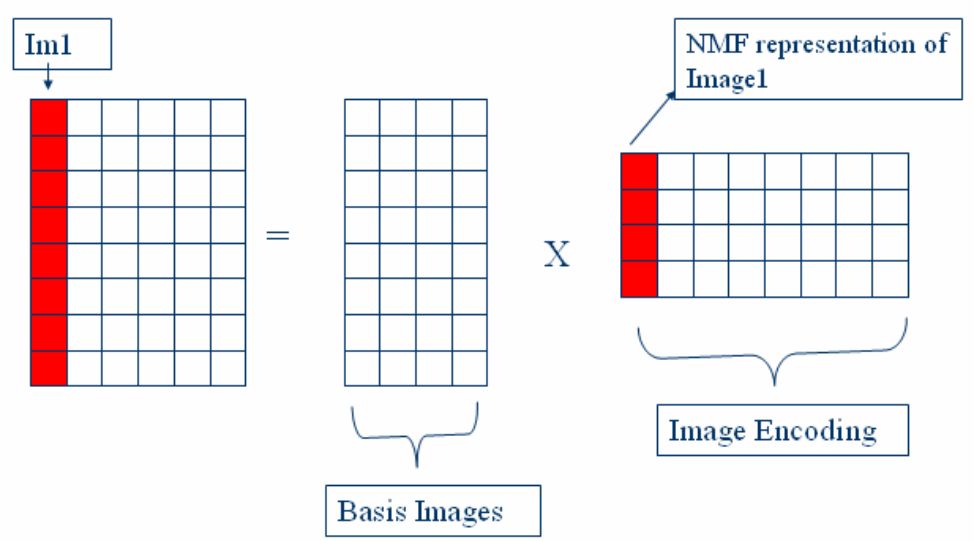

Figure 2.13 NMF factorization

Test images are multiplied by eigenvectors of PCA analysis after mean correction. The resulting vector is pre-multiplied with pseudo inverse of W matrix to get the weights of the test image. Finally cosine similarity is used and the test image is compared with all the images in the training set. The one having smallest angle is chosen.

$$Weight_{Training} = H(:, Training) \text{ (corresponding column of H)} \quad (2.17)$$

$$Weight_{Test} = pinv(W) * PCA(Testimage) \quad (2.18)$$

$$\alpha_{(Test,Training)} = \frac{dot(Weight_{Test}, Weight_{Training})}{\| Weight_{Test} \| * \| Weight_{Training} \|} \quad (2.19)$$

2.3.4 FISCHERFACE

Fisherface method was first proposed in [29]. The idea is to use class specific information to find the projection direction which results in better recognition rates. In other words, after the projection the samples

belonging to different class are separated while samples belonging to same class are clustered.

Mathematically it is achieved by maximizing the ratio of the between class scatter matrix to within class scatter matrix. Between class and within class matrices are calculated as follows:

$$S_{\mathrm{B}} = \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T \tag{2.20}$$

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \tag{2.21}$$

where μ is the mean of all the samples, μ_i is the mean of class X_i and N_i is the number of samples in class X_i. Therefore optimal projection W_{opt} is chosen as

$$W_{opt} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 w_2 ... w_m] \tag{2.22}$$

where $\{w_i \mid i=1, 2, \ldots, m\}$ is the set of generalized eigenvectors of S_B and S_W corresponding to the m largest generalized eigenvalues $\{\lambda i \mid i=1, 2, \ldots, m\}$, i.e.

$$S_B w_i = \lambda_i S_w w_i \tag{2.23}$$

However S_W has to be nonsingular for the above equations to be valid. If there are N number of images and c number of classes in the training set, the rank of S_W is at most N-c. S_W has a size of nxn where n is the number of pixels in an image. In general, n > N which means S_W will be singular. In order to avoid this problem, PCA analysis performed on the training set, vectors belonging to the (N-c) biggest eigenvalues are kept and the dimension is reduced to N-c, as proposed in [8]. Then Fisher Linear Discriminant (FLD) is performed to reduce the dimension further, to c-1.

For the recognition purposes, dimensionality of training images is reduced by PCA. Then these reduced vectors are multiplied by eigenvectors of Linear Discriminant Analysis (LDA) to obtain weights. In order to find the most similar training image, the weights are evaluated by the same method used for Eigenface.

2.3.5 INDEPENDENT COMPONENT ANALYSIS (ICA)

ICA is an iterative method which is meant to solve the blind source separation problem. A sample data, or an observation, is regarded as a linear combination of some unknown sources.

$$x=As \quad (2.24)$$

where s is the set of sources, A is the mixing matrix and x is the set of observations. The aim of ICA is to make sources as independent as possible. In order to find the inverse of the mixing matrix, the number of observations should be at least equal to the number of sources. The inverse of the mixing matrix is found iteratively and then used to obtain weights.

It was introduced for 2D face recognition in [46]. Also some papers have used ICA for 3D face recognition purposes [13], [19], [47]. In this thesis source code available in [56] is used with small modifications. PCA was performed to reduce the dimensionality of the training image set prior to ICA.

There are two architectures in ICA. In the first one, pixel values are treated as observations and face images are treated as variables. The aim is to find statistically independent basis images. In this thesis, Architecture II (Figure 2.14) is used, in which the aim is to find statistically independent coefficients.

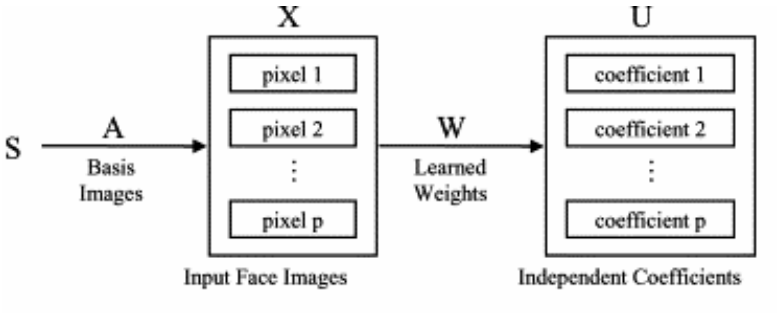

Figure 2.14 Architecture II of ICA

For recognition, as in the previous cases, training images are subjected to PCA to reduce dimensionality. ICA is performed to obtain weights and weights are considered by cosine similarity metric as in the NMF algorithm.

2.4 LITERATURE REVIEW

In the last decade, face recognition has become one of the most popular research areas. The majority of these researches are using 2D images. In [1] Zhao et al. carried out a survey of 2D-based and video-based techniques. There are also several surveys performed on 3D face recognition [26], [14].

One of the common representations used for 3D Face Recognition is depth image, also called depth image. Intensity value of each pixel corresponds to the depth value of the corresponding point. 3D point clouds can be converted to a depth image after being placed on an x-y grid. Irregularly sampled point clouds are interpolated to fit the grid at the cost of loss of some information. 2D Face Recognition techniques can be used for recognition of faces with 3D data once depth images are generated.

In [13] scanner outputs are used to generate depth images. Using the nose tip and bridge of the noses, depth images are registered and

finally component analysis, PCA and ICA, are performed on the depth images. Database consists of 37 subjects each having 6 images with different expressions total 222 images. When 185 images are used during training and 37 are given as test images, ICA performed better. In other cases, PCA performed better. The performance of ICA decreased faster than PCA with the decrease in the number of training images which leads ICA to underperform PCA.

In [19] depth images are projected to a linear subspace. A stochastic optimization algorithm is used to find the subspace which maximizes the classifier performance on the training set. The performance of the optimal projection is reported to be higher than that of standard algorithms such as PCA, ICA.

In [41], Discriminant Common Vectors method was proposed and compared with PCA and ICA algorithms. Discriminant Common Vectors is reported to be superior.

In [31] Hausdorff distance metric is used to depth images and reported to be better than PCA-based matching algorithm. Hausdorff distance of two sets is computed in two steps. Firstly, for all the points in the first set, the closest point on the second set is calculated. Secondly, the maximum value of the minimum values is equal to the Hausdorff distance. This measure is beneficial for cases where some of the data is missing, noisy, or occluded. In [15] Hausdorff distance is used to measure the similarity between two point sets or two voxel representations. In order to represent a 3D point cloud by a voxel function, a 3D grid is constructed whose center cell overlaps with the center point of the point cloud data. If there is at least one point within a voxel, its function takes value of 1, else it is 0. Voxel discretization is faster when the number of points increases, however it leads to some loss of information. Hausdorff metric is useful if the facial data is missing

or incomplete. However, it is computationally expensive. Voxel arrays are suggested for performing Hausdorff based comparisons faster.

In [30] Hausdorff distance was used, however “The number of subjects in the identification study was limited because of the computational cost of the current algorithm.”

In [37], several depth images are used to construct a 3D model for each subject. Matching a depth image to the model is handled in two steps. Firstly, in the coarse alignment step, three feature points are used to rotate the face. Secondly, five feature points are used to align grids of control points with ICP. Critical area of the face is determined and cropped to create a synthesized image. A virtual light source is used for simulation of lighting. LDA is performed on synthesized images and finally scores from shape and texture are fused to make a decision.

In [12] a system using depth images as input is described. Eigenface and Hidden Markov Models operating for grey level images are extended to deal with depth images. A small database is used and eigenface outperformed HMM. Smoothing and rotating increased the performance of eigenface whereas the performance of HMM decreased. Their classification performances are better than standard PCA based matching algorithms.

Depth image representation is only one of the representation techniques used for 3D face recognition. Other representation techniques are surface based methods, point-cloud based methods, curve based methods, and free form shape descriptor based methods. These are explained in the following very briefly.

Extended Gaussian Image (EGI) representation which uses surface normals is one of the 3D face representation methods. Surface normals are mapped to a unit sphere, which is called Gaussian sphere, so that their tails lie at the origin of the sphere and their head lies on the surface. Then the area of the surface having the given normal is given as

weight to each point in the sphere. EGI based methods can not differentiate two faces of similar shape but different size since they use only curvature information of the face.

In [6] EGI (Extended Gaussian Images) are used to estimate orientations between two point cloud sets. The space of rotations is traversed exhaustively to find the estimation which maximizes the correlation between EGI's. In order to find this estimate efficiently, spherical histogram of surface orientations are used to approximate EGI's. Fine registration is carried out by ICP.

In [21] the sign of the mean and the Gaussian curvature are used to segment depth images and an EGI is created for each convex region. A graph matching algorithm is applied to establish correspondence.

In [16] both input and the model faces are represented as an Extended Gaussian Image constructed by mapping principal curvatures and their directions at each surface point.

In [20] facial surfaces are acquired with a structured light. A central and several lateral profiles are obtained from 3D shape. These profiles are used to match 3D surfaces.

In [44] normal maps are generated for meshes in order to compare two 3D surfaces fast and accurately. Meshes are represented on 2D by applying a spherical projection. Keeping the correspondences between 3D meshes and 2D images, normals are encoded in RGB channels of images. A normal map is a 24 bit color image. Each pixel corresponds to a normal of a polygon in the given mesh. Three scalar components of normal are encoded by 8 bits each which correspond to 360/256=1.4 degrees quantization.

In [27] surface and profile matching are fused. For each individual a statistical model is built to perform surface matching. After finding the symmetry plane of the face, profile is extracted and k-th Hausdorff

distance is used to measure similarity. At the end, two scores are combined to make a recognition decision. In [38]

In [10], symmetry plane of the face is found iteratively using two points. Profiles obtained from the plane of symmetry are matched or the surfaces of two faces are compared.

In [11], plane of symmetry is determined using five feature points. Faces are rotated to fit a standard position and then some vertical, horizontal and circular curves extracted from faces are matched.

In [40], five manually marked landmark points are used to determine the region of interest. 70 facial curves (35 horizontal curves and 35 vertical curves) are extracted from each face. Affine integral invariants are used to represent the curves since they are not affected by translation, scaling, rotation and shearing distortion of 3D faces. Discriminant analysis and Jensen-Shannon divergence analysis are carried out on these curves and it is found that 12 of the affine invariant curves are more characteristic. It is found that 10 of them were vertical and areas near the central face profile, nose, and eye corners are more distinguishing.

In [17] faces are segmented based on signs of the mean and the Gaussian curvatures. Descriptors are obtained from segmented regions.

Alternatively, after the profiles are obtained, area between two profiles [20] and also the distances between curves can be evaluated. In [18] L1, L2, and Hausdorff distances are compared and L1 norm is found to be better. Also some approaches for 3D face recognition, such as methods based on extended Gaussian images, ICP matching, depth profile, PCA, and linear discriminant analysis (LDA) are compared. It was reported that ICP and LDA overperformed other approaches although all the scores are similar except PCA alone. During the tests 571 images from 106 people were used.

In [33] curvatures obtained from 8 different points are processed by a SVM for classification.

In [22] a 3D facial model consisting of a sparse depth map is constructed from stereo images. Isoluminance lines are used for stereo matching. Edges and isoluminance contours are used for finding irises. Using the position of irises, the center of the mouth is found and pose of the images are standardized. Finally for each 3D data, in-depth differences are calculated for finding the best match.

In [23] both 2D and 3D data are used for face recognition. Gabor filter responses of 2D images and point signatures from 3D data together form a feature vector after applying PCA. A similarity function or SVM is used for classification.

In [24] both 2D and 3D data are used. Flattened texture and canonical images are formed from 2D and 3D data respectively. One set is constructed for each and eigen-decomposition is used separately to perform recognition.

In [32] point cloud is represented by a regular mesh grid. Nose is utilized to initialize the mesh grid and other four areas are approximated on the mesh. Both global geometric shape of the mesh and local shape information form a feature vector. Matching is performed by examining both. Due to limited number of samples, PCA is applied to reduce the dimensionality.

In [34] after registering the faces by hybrid-ICP, matching of facial scans is performed by considering three attributes: surface matching, texture matching, and shape index matching.

In [35] in order to avoid facial expression changes, both rigid and non-rigid registrations are performed. Rigid registration of the probe image to the gallery is performed by ICP whereas Thin Plate Spline (TPS) is used for non-rigid deformation.

In [36] 3D face matching is performed by ICP on subsets of the face in the motivation of some subsets of face are considerably rigid between different expressions.

In [39], a triangular mesh is given as input to the system. The symmetry plane of the face is found by registering the face with its mirror transformation. Three points are found on this curve and two horizontal curves passing around cheek and forehead are obtained. These three profiles are compared similarly. Although the most discriminative profile was found to be the symmetry profile, it is shown that combining three scores obtained separately improves performance.

In [43], in order to overcome facial expression changes, for each subject a deformable model was built. Using a small control set, the deformations between neutral and non-neutral facial expressions are learned. This template is applied to all the models in the database to obtain the deformable models for individuals. A set of fiducial landmarks are determined and using the landmarks, deformation information is obtained by TPS. Finally this information is transferred to every model in the database.

In [45] point signature, a way to represent free form surfaces, is used as a 3D descriptor.

Comparing the strengths of different face recognition algorithms is difficult since used database differs to a great extent and the experimental conditions are not same. FRVT evaluation protocols respond to this need and light the way for researchers.

CHAPTER 3

EXPERIMENTS

In order to compare the efficiency of the recognition algorithms, seven Tn experiments (T1, T2, … T7) are held where n denotes the number of images used in the training set (Table 3.1). All available images are used in these experiments. The reason why different numbers of images are used in the training and test sets is that, individuals in the database have different number of images. Since the number of images per person in the database changes, the number of images in the training set also has to change. For example, in T7 experiments, all 13 people having 8 images in the database are used. For T4 experiments, 4 images from each person having more than 4 images are taken in the training step. In the test set, the rest of the faces belonging to the subjects participated in the training step are used.

Table 3.1 Distribution of the number of images for Tn experiments

Tn	# of images per person	# of persons (or classes)	# of training images	# of test images
T1	1	197	197	666
T2	2	165	330	469
T3	3	119	357	304
T4	4	86	344	185
T5	5	58	290	99
T6	6	28	168	41
T7	7	13	91	13

T5, T6 and T7 experiments are not performed for every algorithm since these cases do not seem to be challenging enough. In general, as the number of training images increases, the recognition rate of algorithms increases. In this thesis, or for this database, there is another important factor: the number of classes and the images in the test set decrease which reduces the probability that the algorithms produce false responses. To illustrate, even for nonregistered database, the performances of T7 and T6 experiments are 13 out of 13 and 40 out of 41 respectively.

3.1. DETERMINING PARAMETERS

Before carrying out tests, two parameters were determined: the size of the depth image and the number of eigenfaces. For both tests, all 941 images of the non-registered database are given during training step. After that every face is given to the system one by one and the algorithm found the most similar face in the database except the test image itself. Only the top match counts are given in the tables referenced in the following sections. The maximum probable value is 863 since 78 person out of 941 have only one image in the database.

3.1.1 SIZE OF THE DEPTH IMAGE

There is a trade-off between the time complexity and the performance of the algorithms. As the size of the depth image increases, number of sampling points increases and thus, better recognition performances are expected. On the other hand, working with fewer samples greatly reduces the time consumption.

Size of the depth image is changed as in Table3.2. Non-registered faces are given during the training. In conclusion, 61-by-61 grid is chosen.

Table 3.2 The effect of depth image size on performance

# of eigenvectors	**21-21**	**41-41**	**61-61**	**81-81**	**101-101**	**121-121**	**161-161**
20	475	496	500	498	505	502	501
40	510	538	541	542	545	542	544
60	520	554	557	556	559	560	560
100	526	558	562	568	571	567	569

Table 3.3 The effect of depth image size on performance as percentage

# of eigenvectors	**21-21**	**41-41**	**61-61**	**81-81**	**101-101**	**121-121**	**161-161**
20	55,0%	57,5%	57,9%	57,7%	58,5%	58,2%	58,1%
40	59,1%	62,3%	62,7%	62,8%	63,2%	62,8%	63,0%
60	60,3%	64,2%	64,5%	64,4%	64,8%	64,9%	64,9%
100	61,0%	64,7%	65,1%	65,8%	66,2%	65,7%	65,9%

3.1.2 NUMBER OF EIGENFACES

As in the previous case, greater number of eigenvalues will represent the faces better. However the number of samples will again be high and the computations will be heavier.

Table 3.4 The effect of number of eigenfaces

# of eigen-vectors	Top Match (Score)	Top Match (%)	Ratio of eigen-values (*)	Top Match (Score)	Top Match (%)	Ratio of eigen-values (*)
10	421	48,8	94	766	88,8	71,7
20	500	57,9	96,9	803	93,0	80,3
30	523	60,6	97,8	809	93,7	84,5
40	541	62,7	98,4	815	94,4	86,9
50	552	64,0	98,7	815	94,4	88,6
60	557	64,5	98,9	815	94,4	89,9
70	560	64,9	99	815	94,4	90,9
80	562	65,1	99,2	815	94,4	91,7
100	563	65,2	99,4	813	94,2	93
500	566	65,6	~100	805	93,3	99,4

(*) Ratio of the eigenvalues corresponding to selected vectors to all of the eigenvalues

3.2 REGISTRATION ALGORITHMS

In order to examine the effect of registration variants one of the faces having a big alignment error with respect to other faces in the database is chosen. All the tests concerning registration variants are performed using this image.

Two different models are used to register the faces. In fact, the models are constructed in a similar manner, by averaging all the images

in the database. But the difference is that one of them has a size of 61-by-61 whereas the second has 121x121. The first model is used throughout the registration tests. The second is used only to demonstrate the effect of establishing dense correspondence between the model and images. The experiments concerning the second model will be explained in detail in section 3.3. All the registration variants are evaluated on the first model.

Inıtial condition is a crucial parameter for performance of ICP. Although the chosen image is relatively far away from the model when compared to other images, the experiments on variants show that the initial condition is too close to global minimum to examine the effect of variants so that every ICP variant performed similar. Therefore the test image is rotated to move far away from the model. During first two tests, the test image is in point cloud form, which is its original form. Thirdly, original image is converted to grid form.

Experiments are performed in Matlab on a computer having 512 MB RAM and AMD Athlon 3000 2 GHz processor.

Some points should be noted before the graphs are examined. Firstly, alignment error is calculated using the actual correspondences between the sets themselves instead the errors found by algorithms had been used. The reason is that the result would be subjective if the errors found by the algorithms had been used since different algorithms will chose different pairs as corresponding pairs and evaluate the alignment error using these correspondences. Thus, to determine the actual correspondences, the image is registered as it is. Since it is close to the model, the resulting position is considered as global minimum and the last pairs as actual correspondences. And then, effects of variants on rotated image are evaluated using these correspondences.

Secondly, the first alignment error values in the figures showing alignment errors are the initial values. The second values are the results

of the first iterations of each method. All the variants are at the same point at the beginning and thus “iteration 1” values are all the same. Once the algorithm goes through the loop, this value changes with respect to efficiency of the algorithm and “iteration 2” values change between different variants.

Thirdly, the algorithms stop in two case: either the update in the error- measured by algorithm- is below a preset value for 10 times, or the maximum number of iterations is reached. The maximum number of iterations was kept 125 except the tests in Section 3.2.1. In Section 3.2.1, the maximum number of iterations is smaller and the update error threshold is higher. Otherwise, understanding the graphs becomes even harder as the differences between variants are not easily sensible already.

Finally, there are some abbreviations used in the graphics. The meanings are given below:

Std-Dev: Rejecting pairs having distances greater than 2.5 times the standard deviation.

Grid-lookup: In order to find correspondences, a simple lookup table has been constructed.

3.2.1 TESTS ON UNCHANGED FACE IN POINT CLOUD REPRESENTATION

Rejecting pairs according to the threshold defined by the user: At the initial state, some values regarding to the distances between corresponding point pairs are

max=53.85, min=0.06, mean=8.67, and std= 8.93

Giving an “inf” threshold means that the threshold is so high that no pair will be rejected.

It can be seen that as the threshold decreases, the speed of convergence also decreases. Keeping the threshold too small prevents the system from converging to global minimum. The reason is that, if for 10 times, the alignment error measured by algorithm is not updated more than a predetermined value, the algorithm understand that the speed has decreased considerably and thinks that it is about to reach the minimum and thus stops iterations. Figure 3.1 and Figure 3.2 show alignment errors with different rejection thresholds.

Rejecting a percentage of pairs: Unless the percentage is extremely high, the system settles down at the same position almost at the same time. However if the percentage is kept increasing, the speed of convergence decreases. Figure 3.3 and Figure 3.4 show alignment errors for rejecting different percentages of worst pairs.

Rejection method: Not rejecting any pairs, rejecting 10% of pairs, and rejecting the pairs having distances bigger than 2.5*standard deviation methods all seem to have very similar performances. The only difference is that 10% worst algorithm is slower when compared to others. Figure 3.5 and Figure 3.6 show alignment errors for different rejection methods.

Subsampling: Subsampling methods follow the same path when the number of iterations is considered. However, in terms of time consumption, subsampling methods are much better than using all available points. As the number of points in the set decreases, the algorithm speeds up because the most time consuming step of ICP is finding the correspondences between data set and the model set. Figure 3.7 and Figure 3.8 show alignment errors for different subsampling methods.

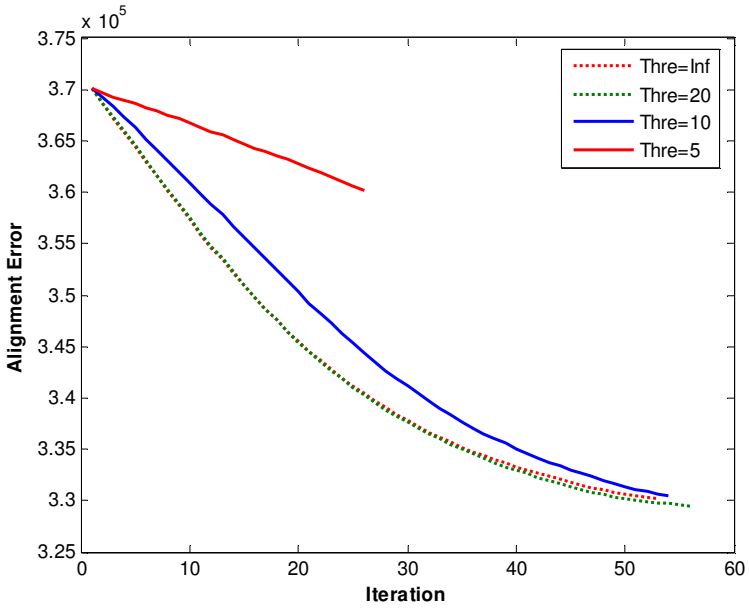

Figure 3.1 Alignment error for different threshold values for rejecting pairs with respect to the number of iterations

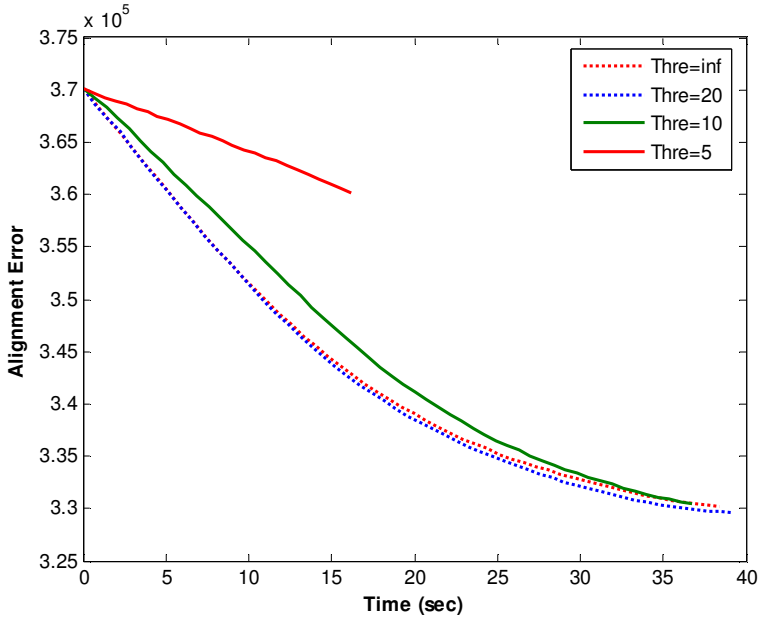

Figure 3.2 Alignment error for different threshold values for rejecting pairs with respect to time

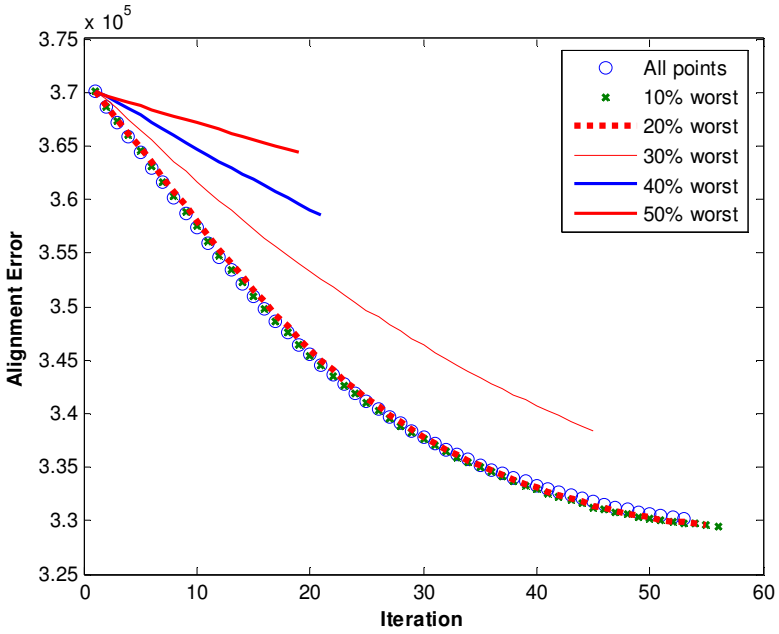

Figure 3.3 Alignment error for rejecting different percentage of worst pairs with respect to the number of iterations

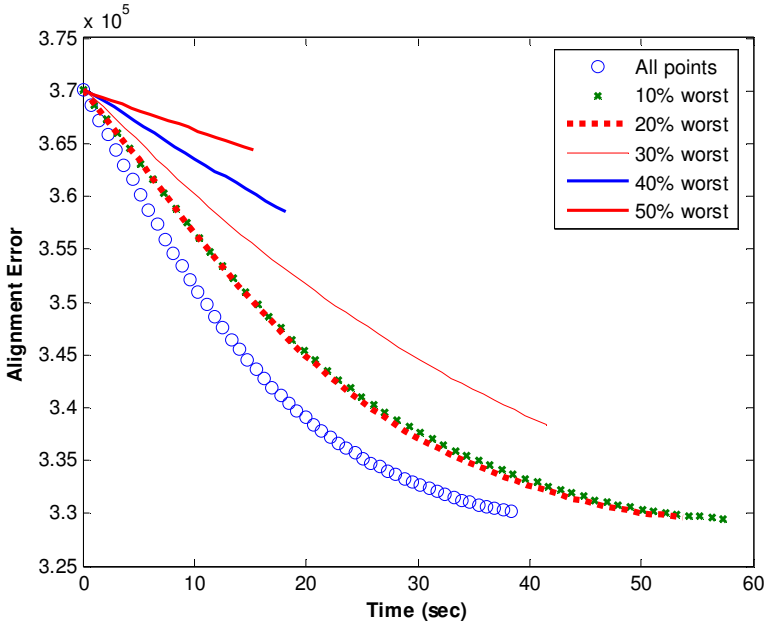

Figure 3.4 Alignment error for rejecting different percentage of worst pairs with respect to time

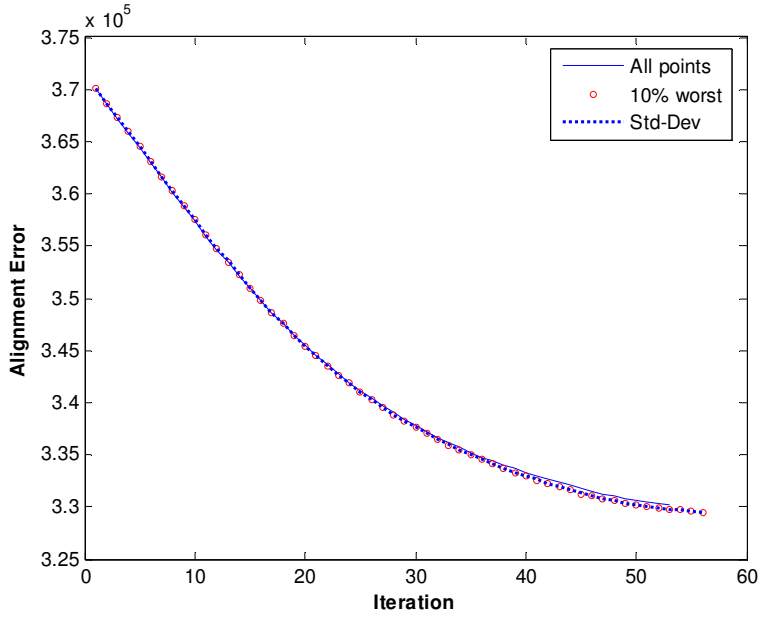

Figure 3.5 Alignment error for different rejection methods with respect to the number of iterations

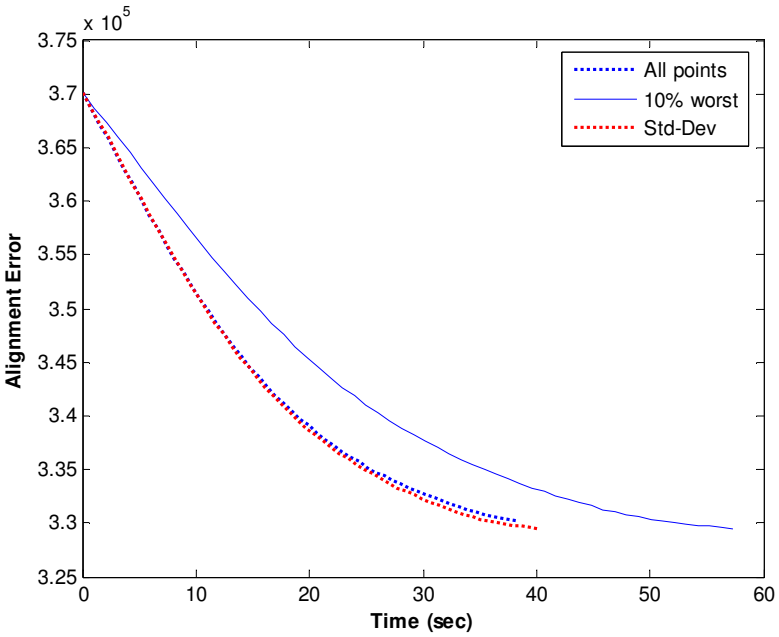

Figure 3.6 Alignment error for different rejection methods with respect to time

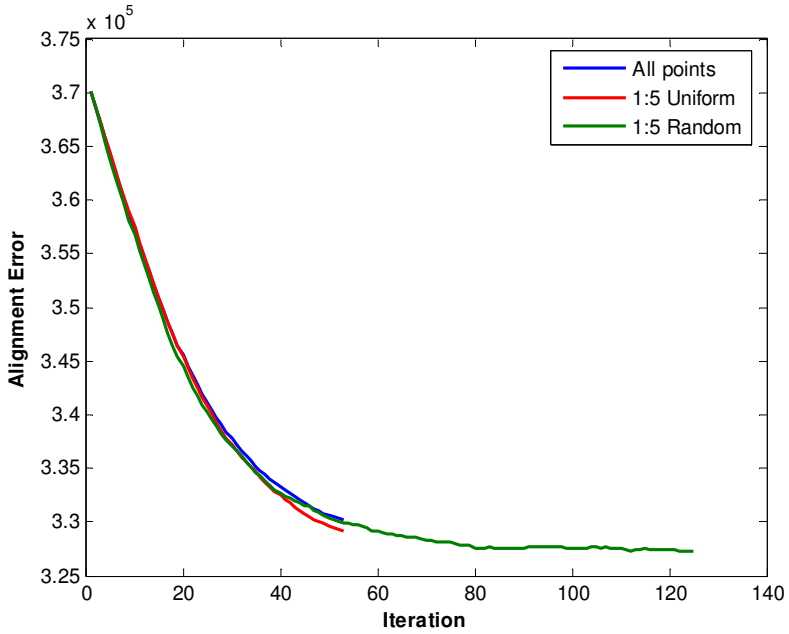

Figure 3.7 Alignment error for different subsampling methods with respect to the number of iterations

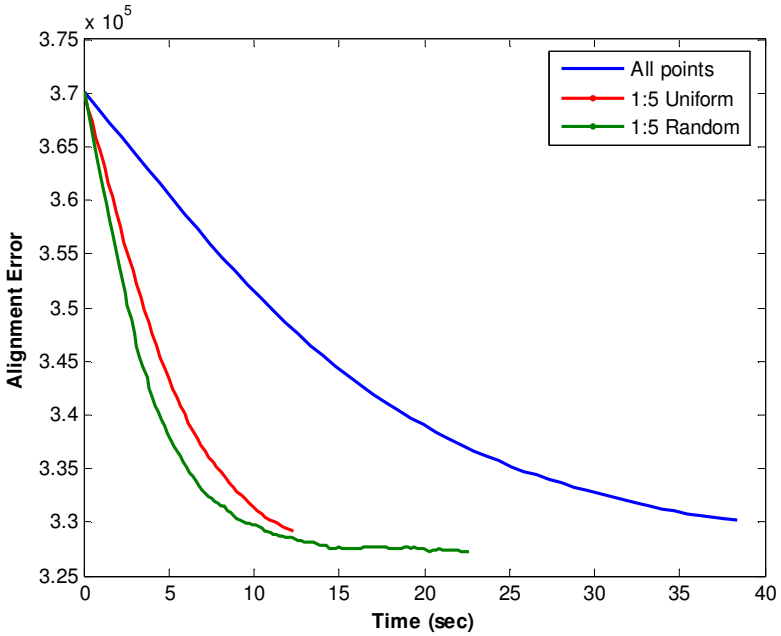

Figure 3.8 Alignment error for different subsampling methods with respect to time

In conclusion, the initial state of the test face and the model is very close so that it does not matter which pairs are used for reaching the final point. When the final alignment error of the variants are considered, all the variants reach to same minimum, which is most probably the global minimum. The only difference between variants is the speed of convergence. Cases listed below perform worst in their scope, when compared to others.

(i) Thre =5 has the worst performance

(ii) Rejecting 50% has the worst performance

Using the pairs having larger differences neither disturb the system nor prevent it from converging. On the contrary, these pairs carry valuable information in terms of registration and thus speed up the process. However, not rejecting these pairs would prevent the system from converging if the initial condition is not good enough.

3.2.2 TESTS ON ROTATED FACE AND POINT CLOUD REPRESENTATION

The same face data, used in Section 3.2.1, is rotated 30 degrees around x, y, and z axis each in order to disturb the system and increase the alignment error between the model. These values are determined experimentally. Many of the variants considered can still converge at that rotation angles. If the image is rotated a little more, then variants start to fail to converge to global minimum.

max= 70.64, min= 0.15, mean= 25.55, std= 14.12

Rejecting a percentage of pairs: Once again, rejecting worst half of the corresponding pairs decrease the speed of convergence. In general, increasing the rejection percentage decreases the speed and the probability of convergence. Figure 3.9 and Figure 3.10 show alignment errors for rejecting different percentages of worst pairs.

Rejection method: Figure 3.11 and Figure 3.12 show effects of different rejection methods. When time is considered, 10% worst acts slower but the number of iterations are almost same.

Subsampling: In terms of the number of iterations, except 1:5 uniform subsampling, subsampling methods converge faster. When the time needed for convergence is considered, all subsampling algorithms perform better than using all the available points. The advantage of random subsampling becomes obvious. Not all points carry the same information in terms of registration. Random subsampling gives every point a chance to get involved to registration step and thus speeds up the process. Figure 3.13 and Figure 3.14 show alignment errors for different subsampling methods.

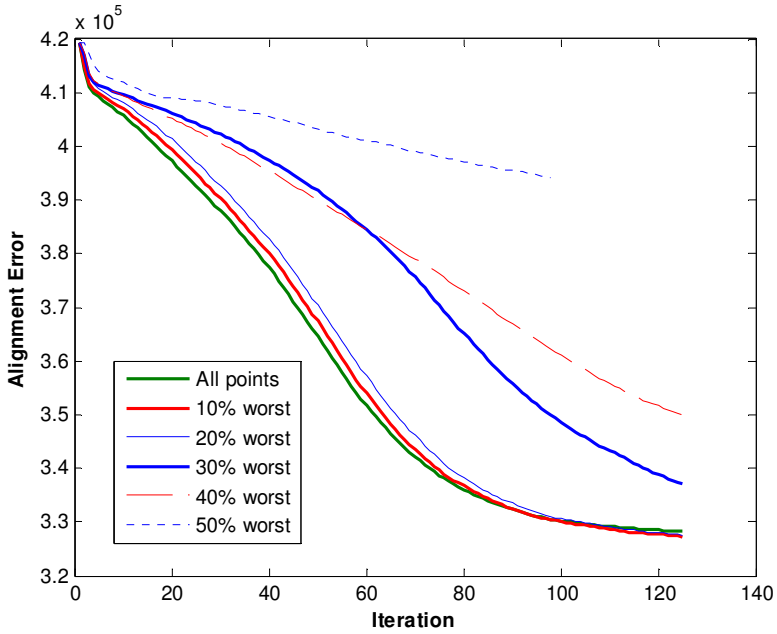

Figure 3.9 Alignment error for rejecting different percentage of worst pairs with respect to the number of iterations

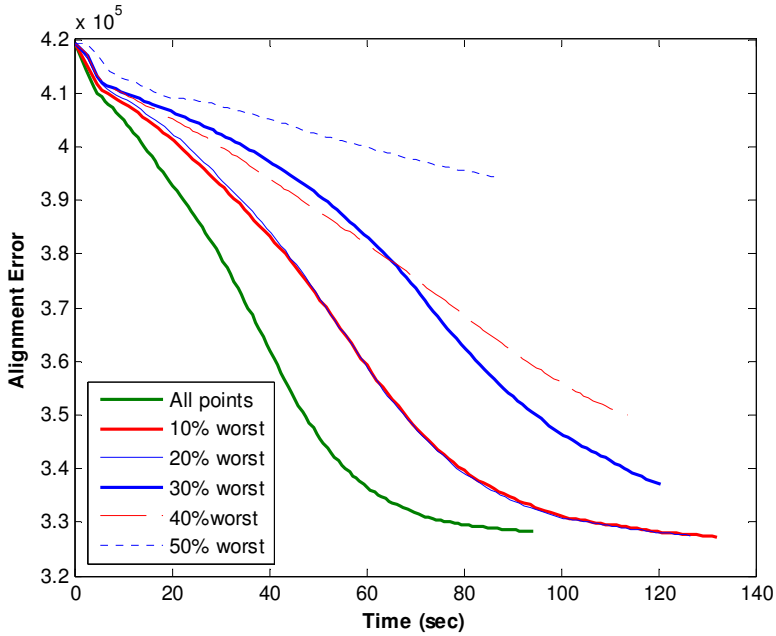

Figure 3.10 Alignment error for rejecting different percentage of worst pairs with respect to time

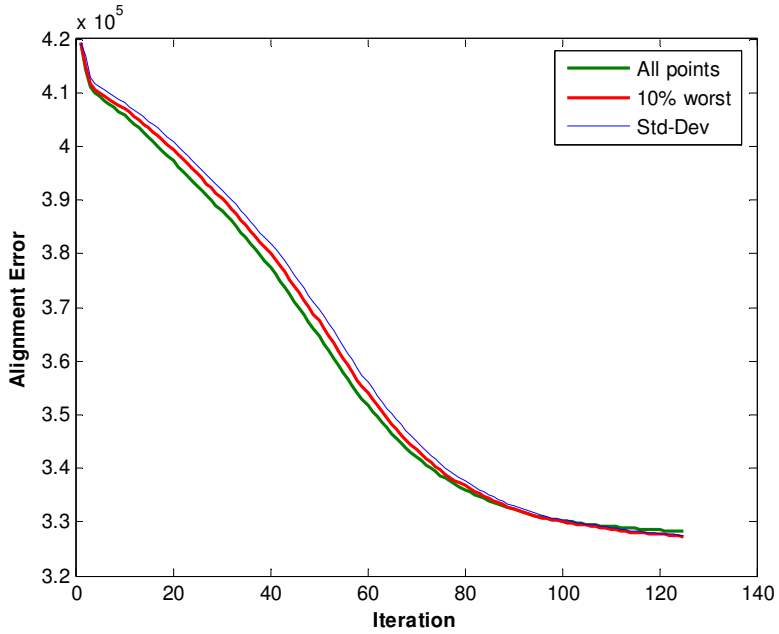

Figure 3.11 Alignment error for different rejection methods with respect to the number of iterations

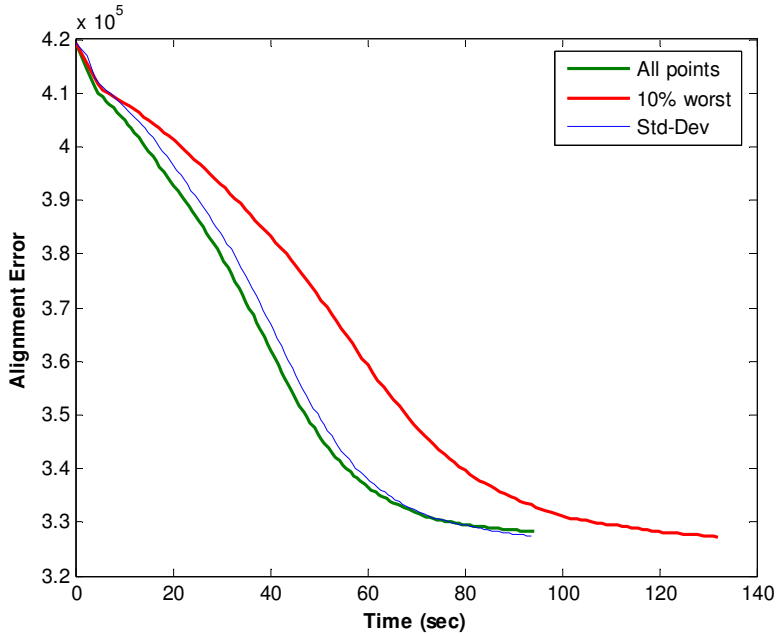

Figure 3.12 Alignment error for different rejection methods with respect to time

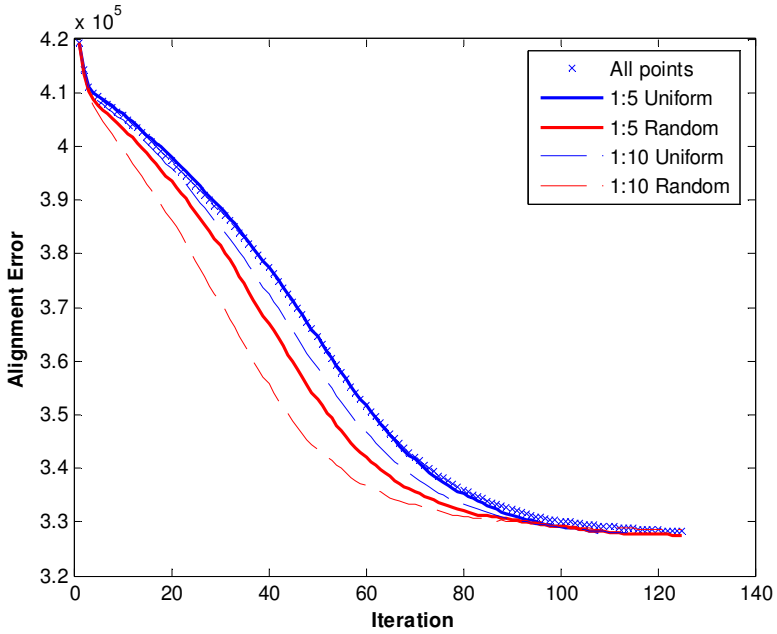

Figure 3.13 Alignment error for different subsampling methods with respect to the number of iterations

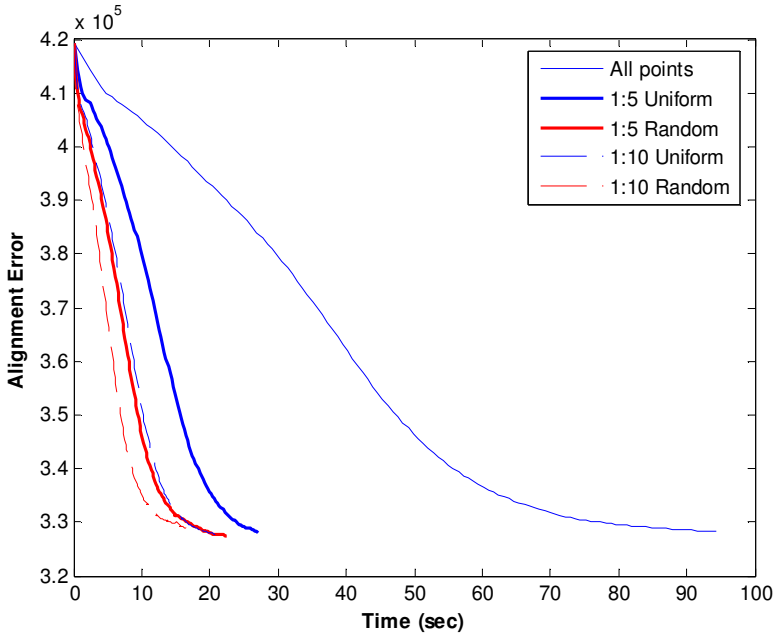

Figure 3.14 Alignment error for different subsampling methods with respect to time

3.2.3 TESTS ON GRID DATA REPRESENTATION

In Section 3.2.1 and 3.2.2, data to be registered is in point cloud form. It is also possible to register the test image using grid data representations. The model is already in grid form. Up to this point, registration parameters are found in by registering raw point cloud to average model. In this part raw point cloud data is represented by a grid and grid is used to determine the registration parameters.

Matching pairs: One of the variants proposed for finding correspondences is to construct a lookup table for regularly sampled data. In general, the most time consuming step of ICP is to finding correspondences in the other mesh. Using a lookup table will surely reduce time consumption.

In order to find correspondences between data image and the model, simply the pixel positions are used (Figure 3.15).

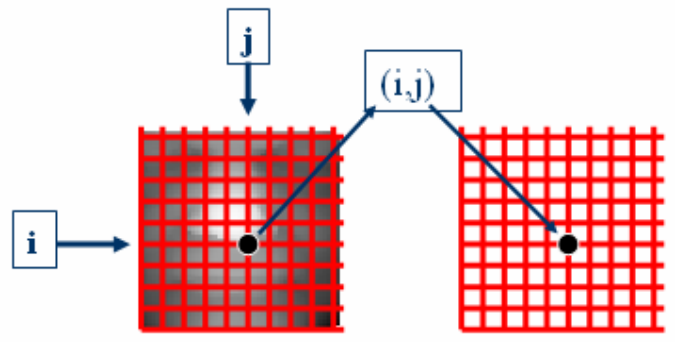

Figure 3.15 Matching by a look-up table

In the examples below, the effect of lookup table was examined. In order to evaluate the accuracy of this algorithm, all the images in the database are registered with this method without rejecting any pairs and

without any subsampling. It was seen that the performance results are exactly the same with ICP baseline algorithm which is working on 61x61 grid images. The results can be examined in Section 3.3 for 'M61Grid' case.

This variant is known to be faster than but not as accurate as other variants. In cases, where speed is crucial but ending up in a little wrong result is acceptable, this variant is preferred if the sets to be registered are regularly sampled, such as grid represented images. In this case, two sets are already close to each other, which also enhances the probability of ending in global position. Furthermore, this variant perform same with ICP baseline which means when initial condition is good, this variant also converges to global minimum.

In Figure 3.16 and Figure 3.17, several rejection methods are compared when matching is performed with the help of a lookup table. The most interesting point is that, it is possible to obtain a good registration in a single iteration by using all the points.

In terms of speed, grid is very fast. However accuracy depends on "sampling frequency" of data while obtaining grid form. When the input data is converted to grid form, if it is not sampled dense enough, performance may decrease. In Figures 3.18-3.25 input data have size of 61-by-61. If the input data is sampled twice denser, then the algorithm will reach to almost same point as in the point cloud case. In Figures 3.26-3.29 input data have size of 121-by-121. When (Figure 3.26 and Figure 3.29 show the results of comparison when the input faces are in point cloud form and in grid form).

In other words, in Figures 3.18-3.29, all the faces are registered to same model which has a size of 61-by-61. The difference is that, the input images are either in point cloud form, or in grid form having size either 61-by-61 or 121-by-121.

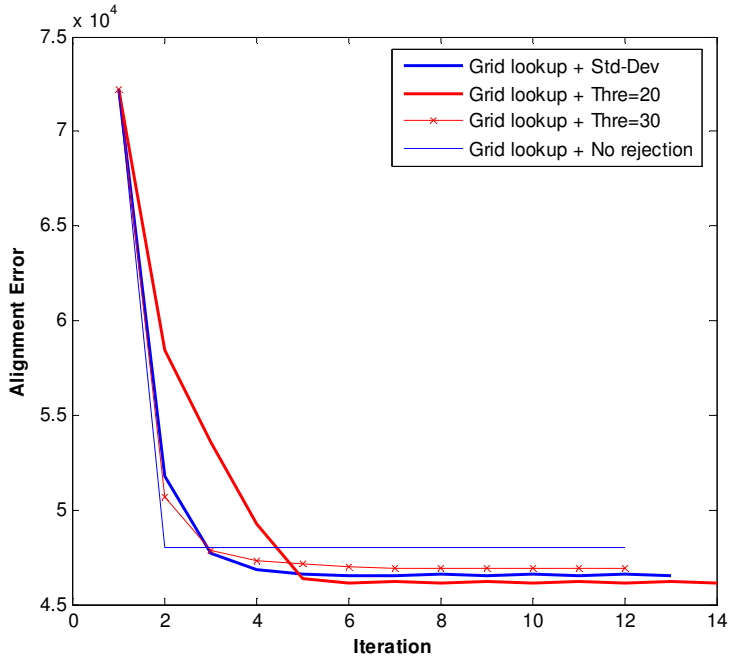

Figure 3.16 Alignment error for grid look-up matching method with respect to the number of iterations

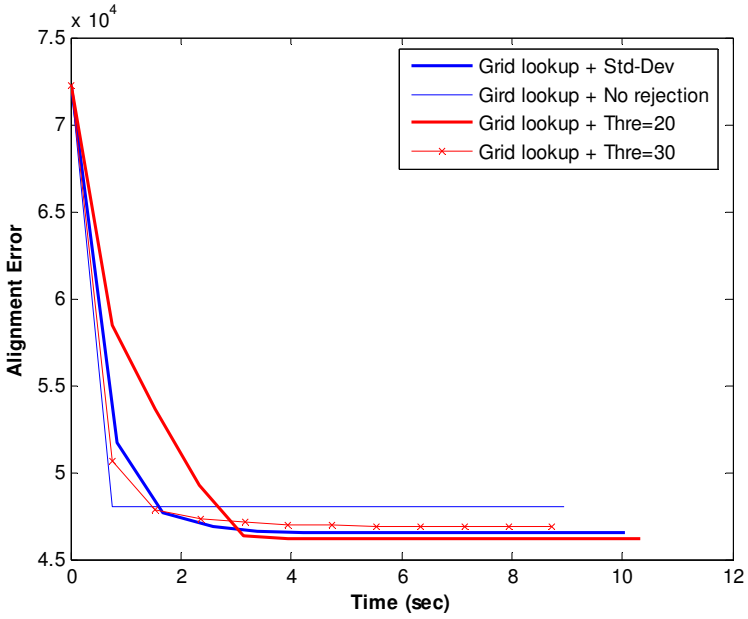

Figure 3.17 Alignment error for grid look-up matching method with respect to time

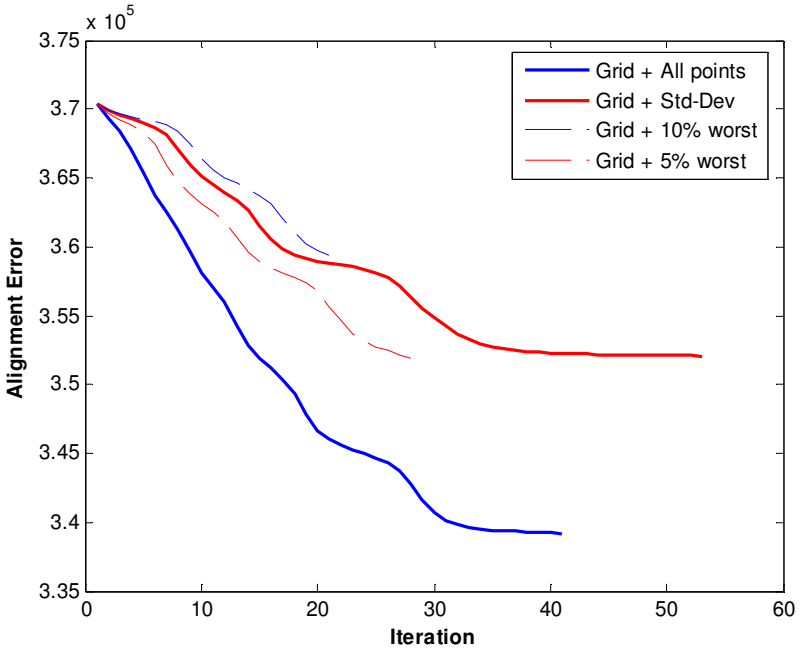

Figure 3.18 Alignment error for different rejection methods with respect to the number of iterations

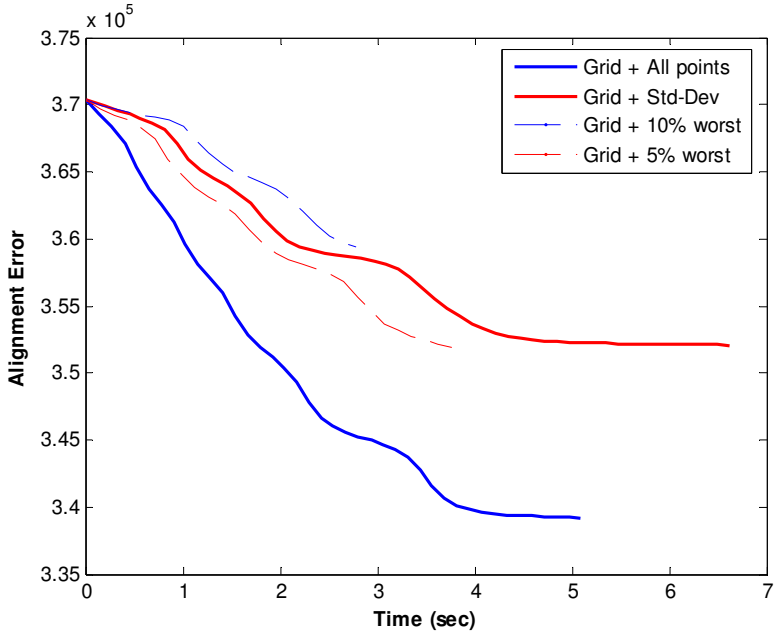

Figure 3.19 Alignment error for different rejection methods with respect to time

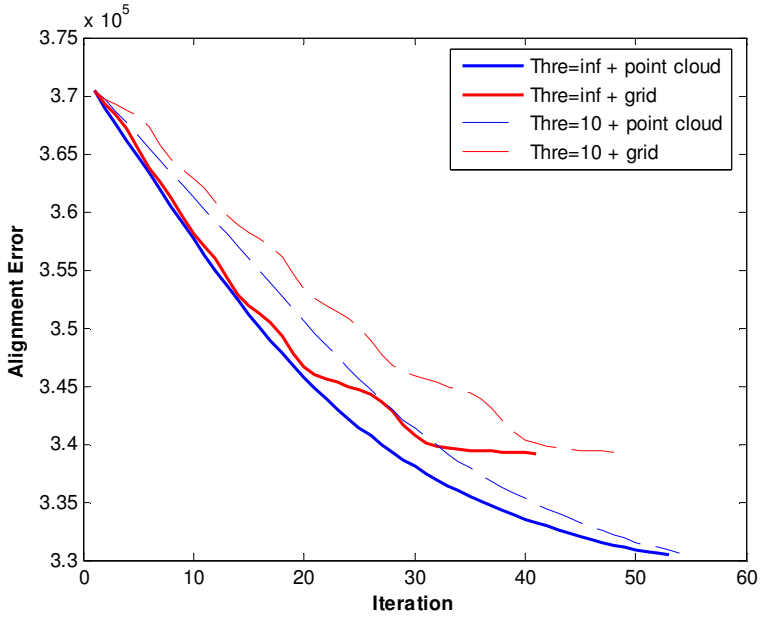

Figure 3.20 Threshold rejection for grid 61x61 and point cloud cases

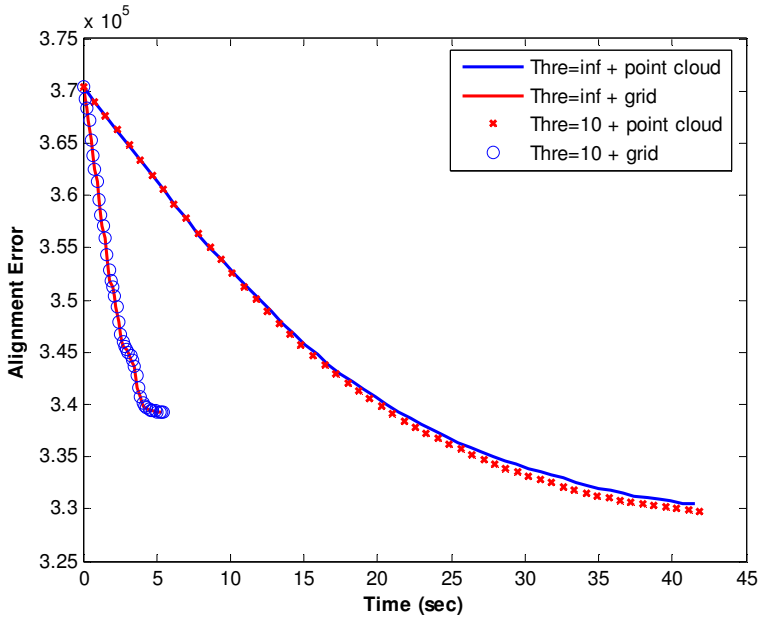

Figure 3.21 Threshold rejection with respect to time

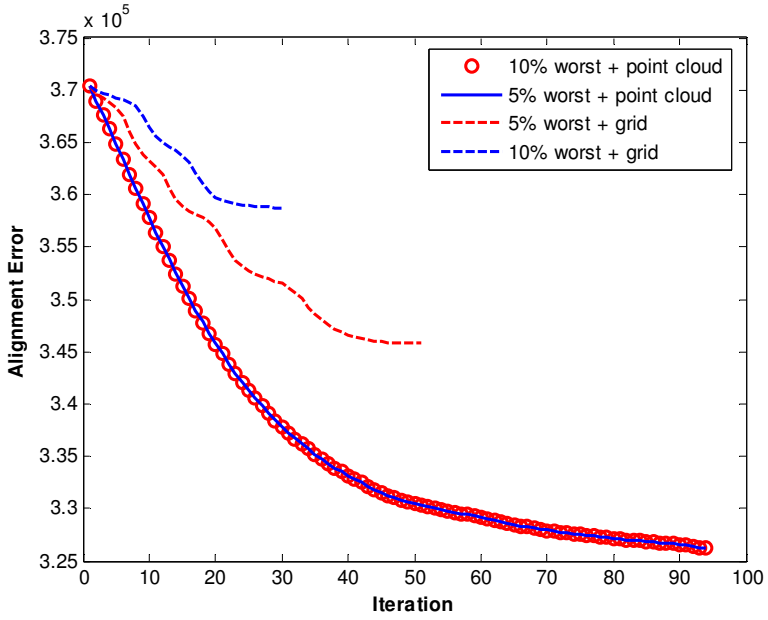

Figure 3.22 Alignment error for rejecting different percentage of worst pairs with respect to number of iterations

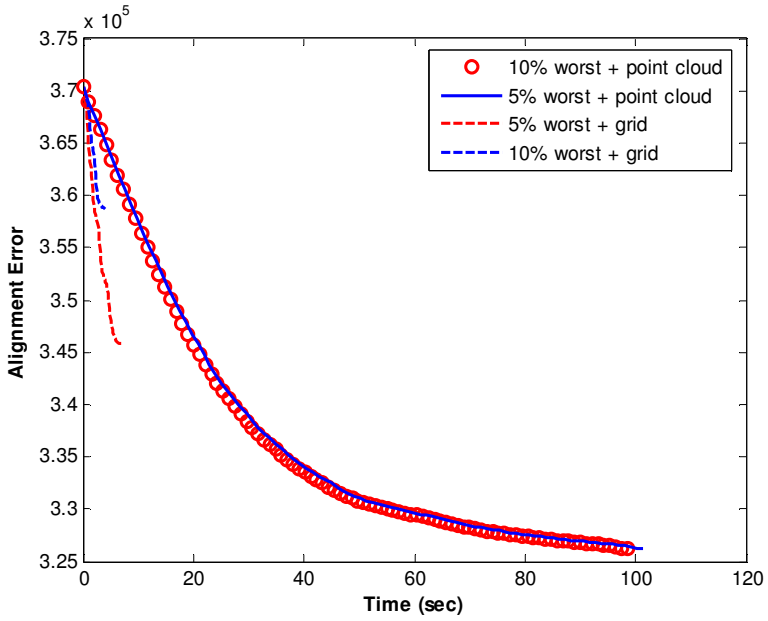

Figure 3.23 Alignment error for rejecting different percentage of worst pairs with respect to time

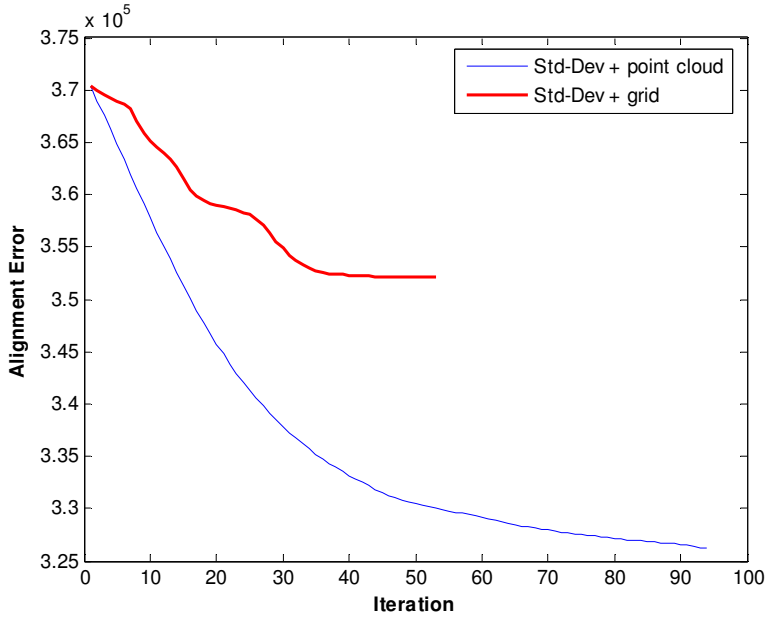

Figure 3.24 Alignment error for standard deviation rejection method with respect to the number of iterations

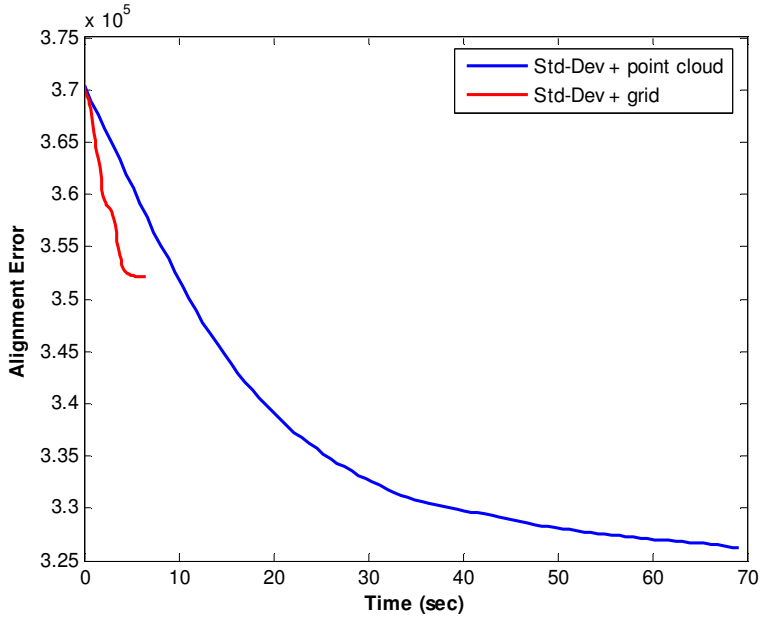

Figure 3.25 Alignment error for standard deviation rejection method with respect to time

Figure 3.26 and 3.27 show the difference when both the model and the test are twice sampled in each axis (the number of samples becomes four times greater) the time consumption does not increase by four. The area of face used does not change but the number of samples is doubled. Elapsed time for 61x61 grid case are 6.61 sec, 7.02 sec, and 7.3 sec for Std-dev, 5% worst, and all points respectively. For 121x121 grid case, 21.34 sec, 20.39 sec, and 13.81 sec in the same order of the cases. Point cloud registration requires 69.83 sec to reach convergence. These figures indicate the difference between 61x61 grid and 121x121 grid. 121x121 grid is much more accurate than 61x61 grid case and yet much more faster than point cloud case. Figure 3.28 and 3.29 demonstrate the difference between 121x121 grid and point cloud case.

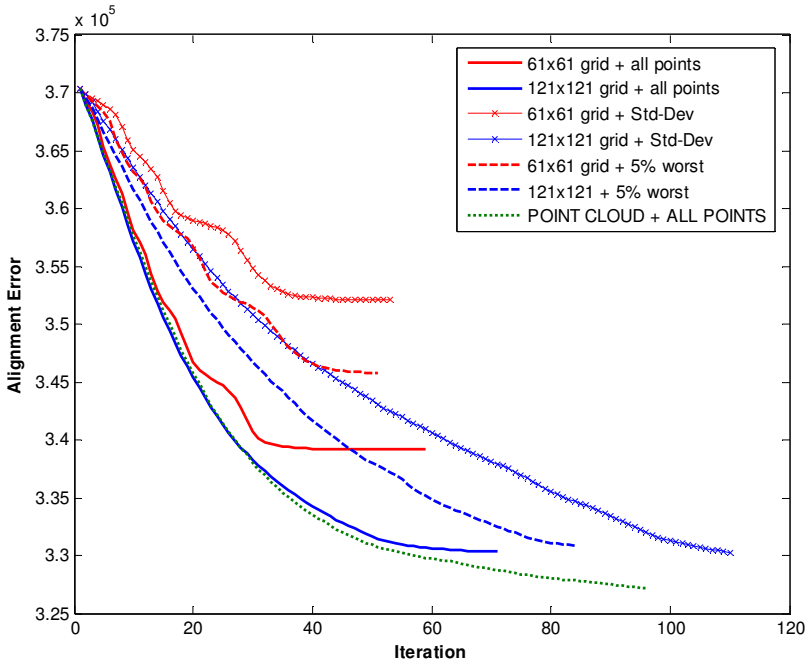

Figure 3.26 Effect of grid size with respect to the number of iterations

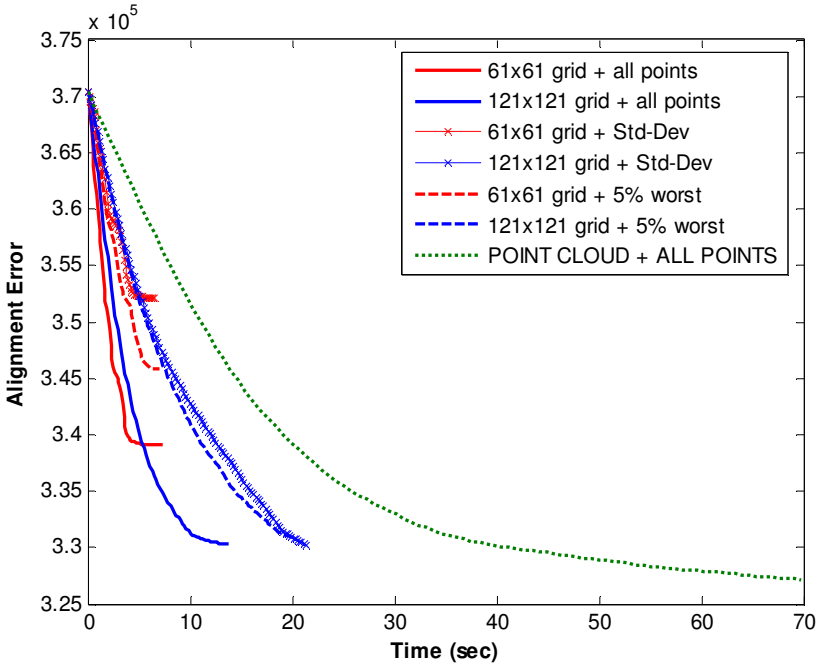

Figure 3.27 Effect of grid size with respect to time

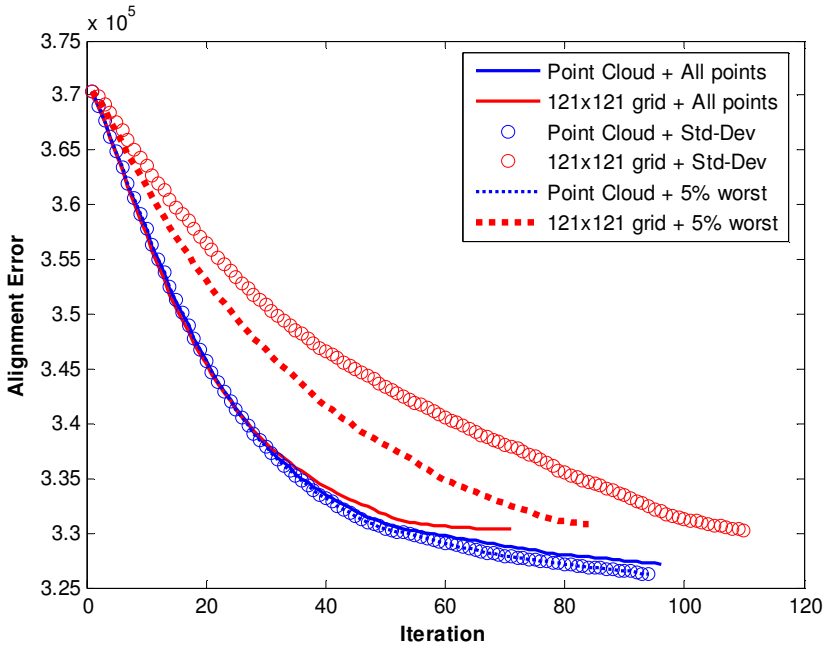

Figure 3.28 Grid size 121x121 vs Point Cloud

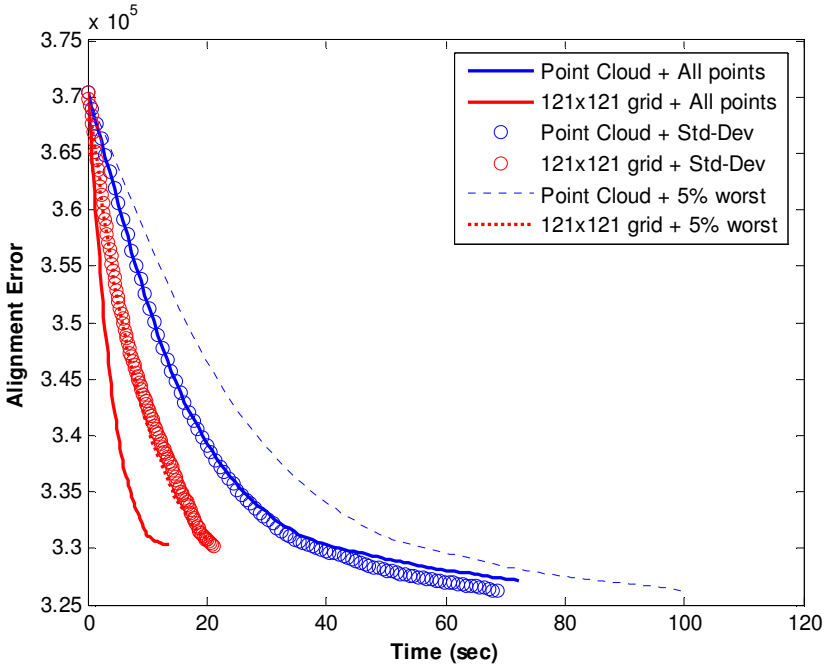

Figure 3.29 Grid size 121x121 vs Point Cloud

121x121 grid is comparable in terms accuracy and speed. In order to evaluate the effect of denser sampling, all the images in the database are registered with 121x121 model. Again, both point cloud form and grid form are evaluated in terms of accuracy by using the recognition algorithms of Section 3.3

3.3. EFFECT OF REGISTRATION ON DIFFERENT RECOGNITION ALGORITHMS

During the registration step, the choice of region of face and the choice of model are two important parameters.

To determine the region of face all the images are examined and boundaries on x and y axis, having valid points are considered. For each image, maximum value having a valid point is determined in + and -, x and y axis. Using minimum of these values, a region is obtained, in which every face will have valid data points. If during this step used images are not registered, region of face is smaller (Region 1). If registered faces are used, the region becomes greater (Region 2). These two regions for a sample image are shown in Figure 3.30.

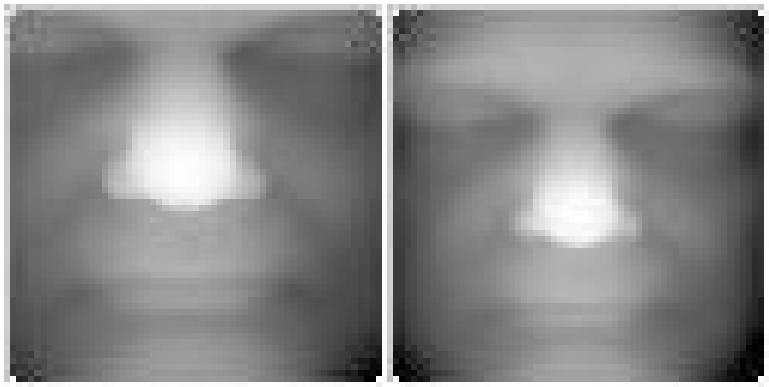

Figure 3.30 Region 1 and Region 2

Also three different cases were considered for the choice of the model:

Js1: A sample image whose pose is favorable + Region 2

Js2: Taking average of all the images + Region 2

Js3: Summing all the faces with their x-y symmetrics + Region 2

Js4: Summing all the faces with their symmetrics + Region 1

Main: Taking average of all the images + Region 1

Using the eigenface method and ICA method, the above methods are compared. The results are given in Table 3.5 and 3.6 respectively.

Table 3.5 Eigenface method on differently chosen models

	T1	T1(%)	T2	T2(%)	T3	T3(%)
Main	571	85,7%	447	95,3%	297	97,7%
Js1	559	83,9%	429	91,5%	283	93,1%
Js2	552	82,9%	435	92,8%	286	94,1%
Js3	576	86,5%	438	93,4%	287	94,4%
Js4	369	55,4%	350	74,6%	244	80,3%

Table 3.6 ICA method on differently chosen models

	T1	T1(%)	T2	T2(%)	T3	T3(%)
Main	602	90,4%	451	96,2%	298	98,0%
Js1	590	88,6%	439	93,6%	290	95,4%
Js2	580	87,1%	435	92,8%	286	94,1%
Js3	594	89,2%	443	94,5%	287	94,4%
Js4	405	60,8%	381	81,2%	255	83,9%

The choice of a smaller region decreased the performance of method which symmetrically sums all the faces (js3 vs js4). However it improves the performance of averaging all the image case (js2 vs main).

It should be pointed that although the face region of interest changes between Region1 and Region2, depth image sizes are same. That is, the starting and end lines on the face changes but the sampling of 3D face data is also changed accordingly so that all the resulting depth images have 61-by-61 size.

Finally, the grey level images are processed to compare their performace with respect to 3D images. Using the boundaries of 3D data for each image boundary pixels are determined. Then the image is cut, and the resulting image is resized to 61-by-61. Finally, histogram equalization is performed on all the resulting images to reduce the effect of lighting. Table 3.7 shows the performances of algorithms if only the grey level images are used.

Table 3.7 Grey Level Images

	T1	T1(%)	T2	T2(%)	T3	T3(%)
Eigenface	465	69,8%	371	79,1%	250	82,2%
NMF	452	67,9%	362	77,2%	247	81,3%
ICA	513	77,0%	398	84,9%	267	87,8%
Fisherface	-	-	396	84,4%	268	88,2%

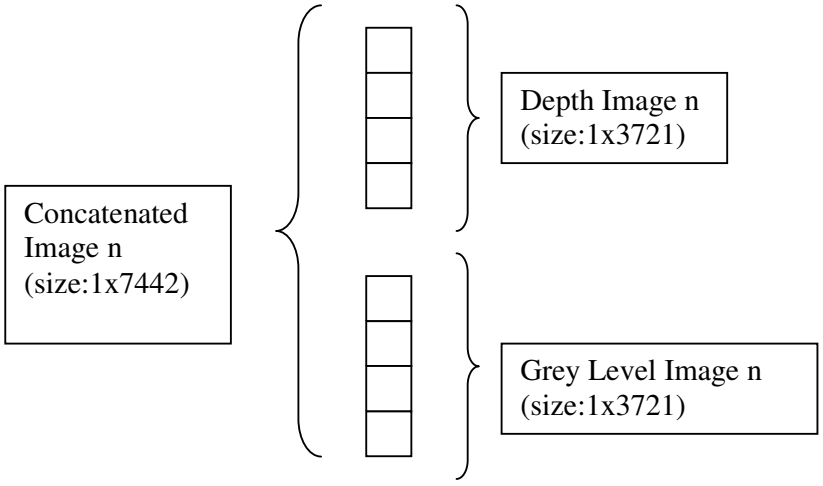

Figure 3.31 Concatenating Grey Level Images and Depth Images

Images are concatenated as explained in Figure 3.31. before concatenation, normalization of data is performed. The performance does not improve when concatenated images are used.

Table 3.8 Normalized Image Concatenation

	T1	T1(%)	T2	T2(%)	T3	T3(%)
Eigenface	475	71,3%	378	80,6%	255	83,9%
NMF	463	69,5%	371	79,1%	252	82,9%
ICA	522	78,4%	403	85,9%	270	88,8%
Fisherface	-	-	403	85,9%	271	89,1%

To evaluate the effect of ICP on the performance, Tn experiments are performed as explained in Section 3.1. Same number of images from all classes is given to the algorithms as input and other images, not used in the training step and belonging to one of the training classes, is given as test image.

Some abbreviations used in this section for representing the databases are given below. Performances of these algorithms are shown in Table 3.9:

Non-Reg: Database consisting of nonregistered images. Only the nose tips are carried to origin. Neither fine tuning nor rotation performed.

M61Grid: During the registration process, the model image has a size of 61-by-61 and the faces registered to the model are in grid form. The resulting images are stored in 61-by-61 size.

M61PC: During the registration process, the model image has a size of 61-by-61 and the faces registered to the model are in point cloud form. The resulting images are stored in 61-by-61 size.

M121Grid: During the registration process, the model image has a size of 121-by-121 and the faces registered to the model are in grid form. The resulting images are stored in 121-by-121 size.

M121PC: During the registration process, the model image has a size of 121-by-121 and the faces registered to the model are in point cloud form. The resulting images are stored in 61-by-61 size.

3.3.1 EFFECT OF NUMBER OF IMAGE PER CLASS IN THE TRAINING SET

The number of classes is kept constant at 119 and for each class 1, 2, and 3 images are used during training step. An image not used during training and belonging to one of the 119 classes is given as input. The resulting performances are given in Figure 3.32 for eigenface method.

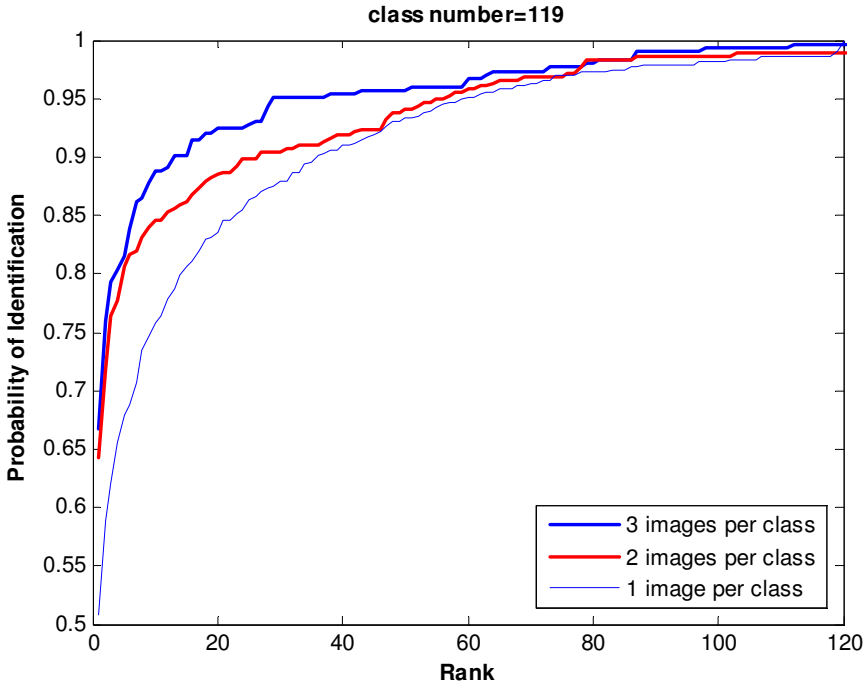

Figure 3.32 Effect of number of images per class

3.3.2 Tn EXPERIMENTS

For each recognition method, two graphs are drawn (Figure 3.33-3.39) for each recognition method: one for nonregistered database and the other for M61PC database. In this section “point cloud registered” term in the graph corresponds to M61PC database.

EIGENFACE:

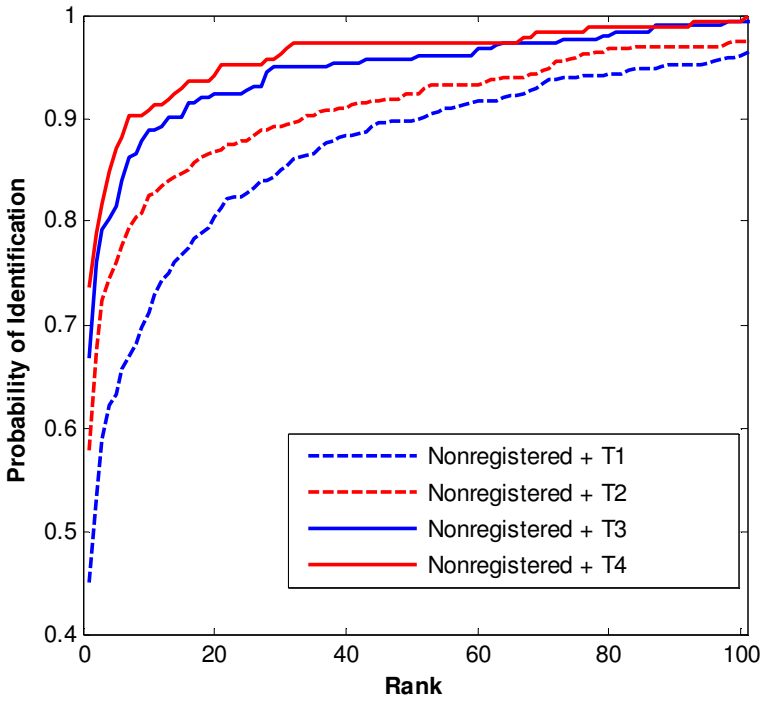

Figure 3.33 Tn experiments using eigenface on nonregistered database

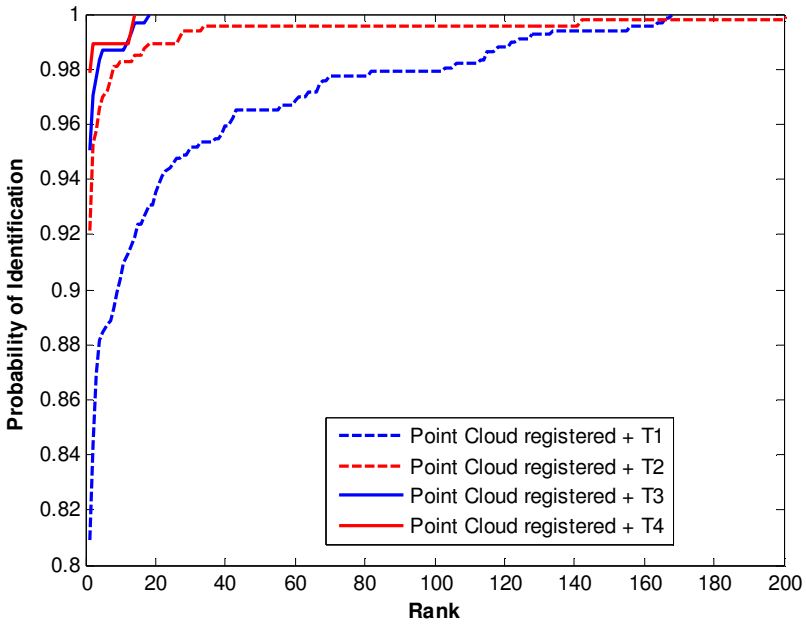

Figure 3.34 Tn experiments using eigenface on M61PC database

NMF:

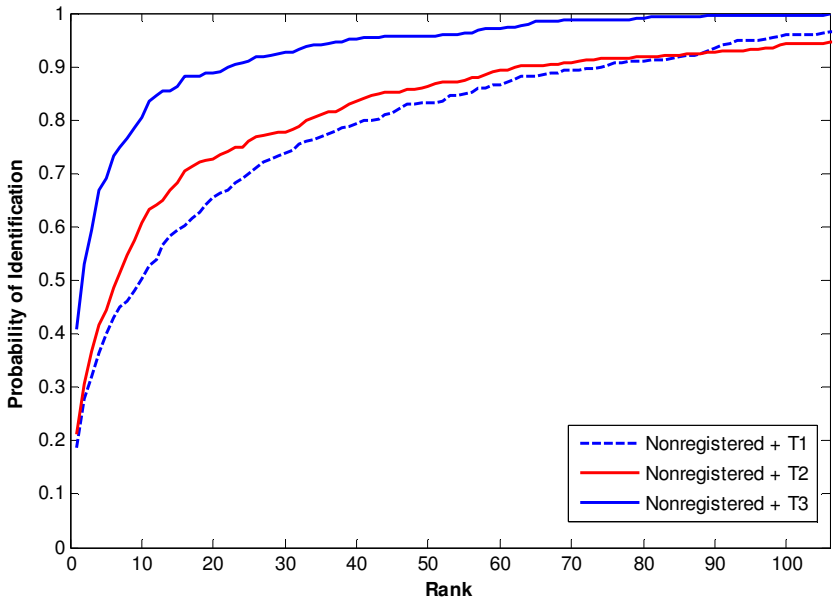

Figure 3.35 Tn experiments using NMF on nonregistered database

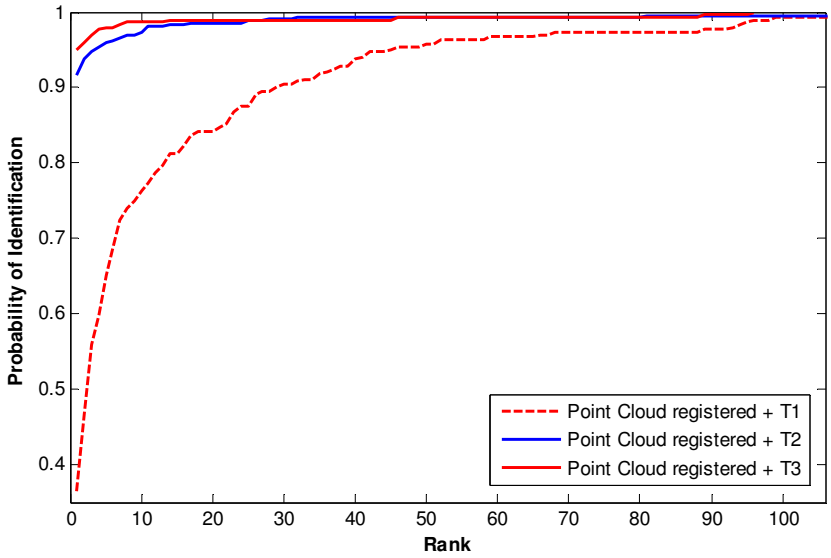

Figure 3.36 Tn experiments using NMF on M61PC database

ICA:

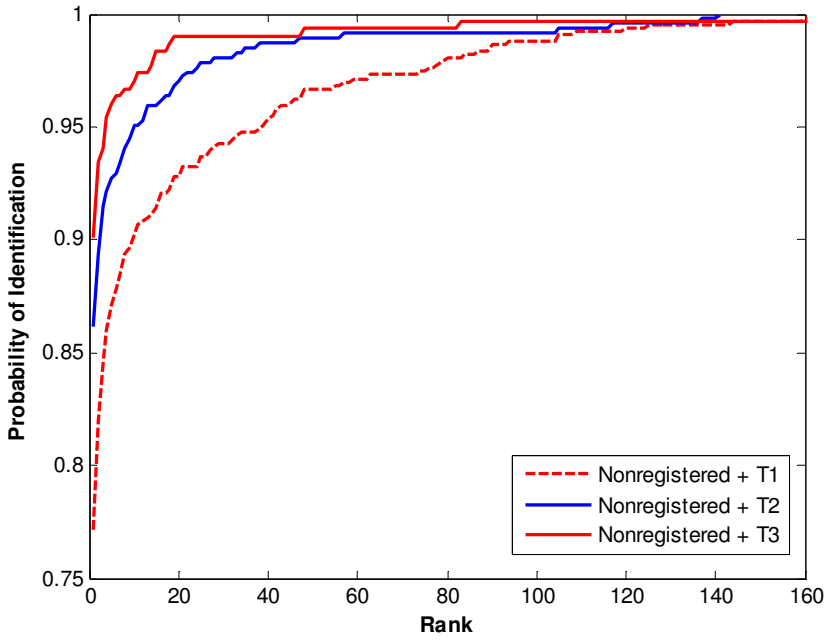

Figure 3.37 Tn experiments using ICA on nonregistered database

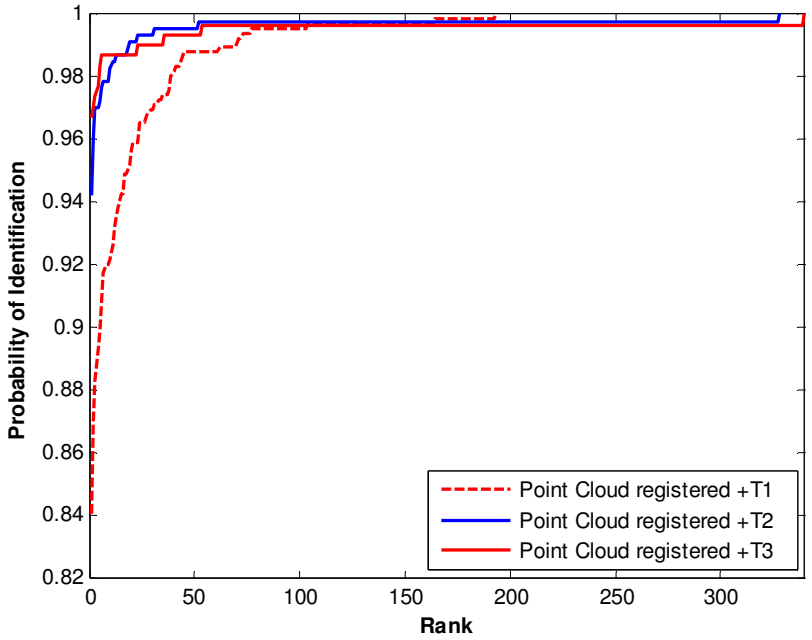

Figure 3.38 Tn experiments using ICA on M61PC database

FISHERFACE:

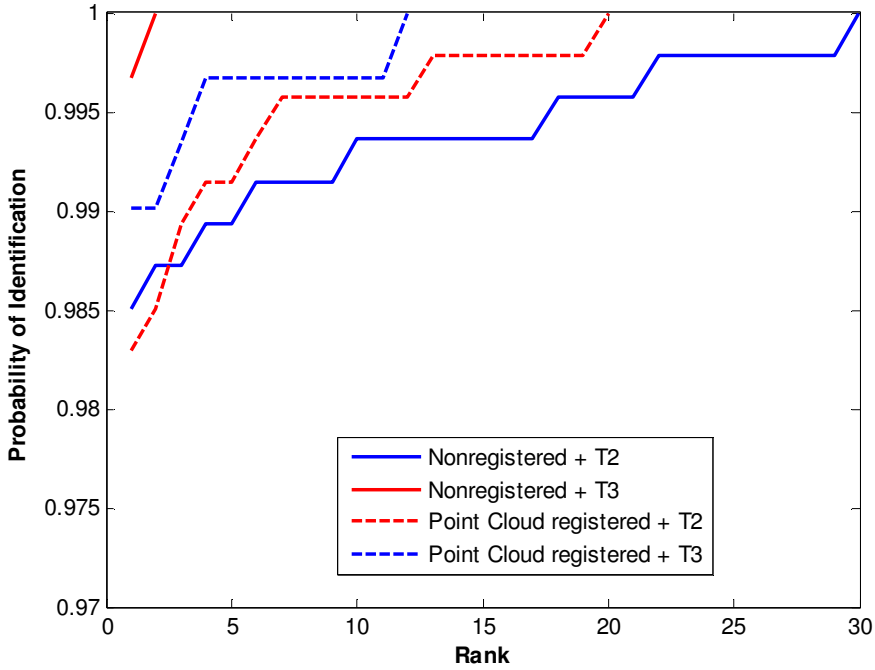

Figure 3.39 Tn experiments using Fisherface

T1 experiments cannot be performed for Fisherface algorithm since it requires both between class scatter matrix and within class scatter matrix. In order to compute the within class scatter matrix, at least two samples are needed for each class.

Besides when images have 121x121 points, between class scatter and within class scatter matrices have to be 14641 square matrices. The computer running the algorithms is not able to define that big matrix in Matlab 7. Thus Fisherface experiments could not be performed for these data sets.

Figure 3.40-3.44 show Tn experiments performed on same registered database with different recognition algorithms. Fisherface perform better. For T1, where Fisherface cannot be performed, ICA perform best.

Table 3.9 Performances of face recognition algorithms as top match scores and percentages

	T1					T2				
	Non-Reg	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**	**Non-Reg**	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**
Euclidian	304	453	544	595	574	273	372	434	451	448
Eigenfac	300	453	539	586	571	271	372	432	451	447
Fisherfac	-	-	-	-	-	462	400	461	-	-
NMF	125	486	529	590	567	73	398	442	454	453
ICA	514	512	555	611	602	405	416	443	455	451
	T1					**T2**				
	Non-Reg	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**	**Non-Reg**	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**
Euclidian	45,6	68,0	81,7	89,3	86,2	58,2	79,3	92,5	96,2	95,5
Eigenfac	45,0	68,0	80,9	88,0	85,7	57,8	79,3	92,1	96,2	95,3
Fisherfac	-	-	-	-	-	98,5	85,3	98,3	-	-
NMF	18,8	73,0	79,4	86,6	85,1	15,6	84,9	94,2	96,8	96,6
ICA	77,2	76,9	83,3	91,7	90,4	86,4	88,7	94,5	97,0	96,2

Table 3.9 (continued) Performances of face recognition algorithms as top match scores and percentages

	T3					T4				
	Non-Reg	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**	**Non-Reg**	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**
Euclidian	203	261	290	297	297	136	167	181	183	184
Eigenface	203	261	289	297	297	136	167	181	183	184
Fisherface	303	276	301	-	-	184	171	183	-	-
NMF	118	273	290	299	297	89	169	179	183	182
ICA	273	284	294	300	298	166	172	178	182	182
	T3					**T4**				
	Non-Reg	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**	**Non-Reg**	**M61 Grid**	**M61 PC**	**M121 Grid**	**M121 PC**
Euclidian	66,8	85,9	95,4	97,7	97,7	73,5	90,3	97,8	98,9	99,5
Eigenface	66,8	85,9	95,1	97,7	97,7	73,5	90,3	97,8	98,9	99,5
Fisherface	99,7	90,8	99,0	-	-	99,5	92,4	98,9	-	-
NMF	38,8	89,8	95,4	97,7	97,7	48,1	91,4	96,8	98,9	98,4
ICA	89,8	93,4	96,7	98,0	97,7	89,7	93,0	96,2	98,4	98,4

T1 EXPERIMENT

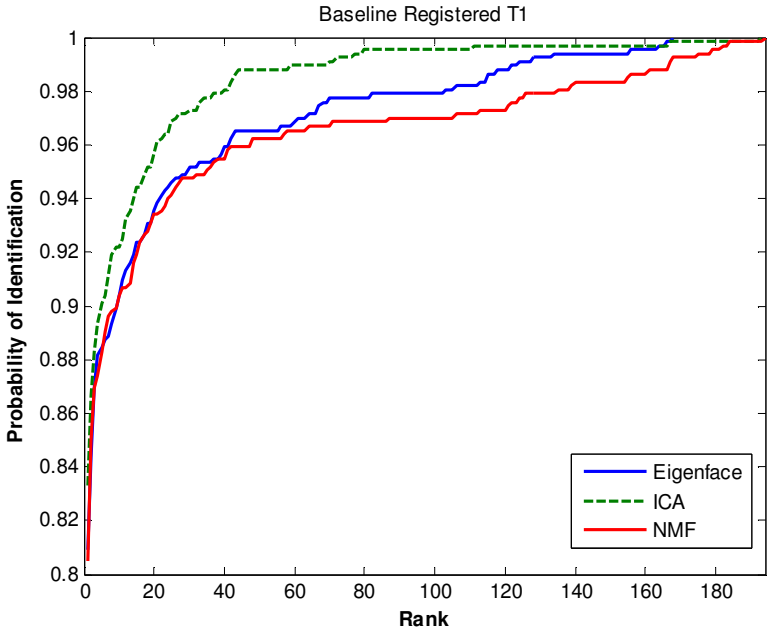

Figure 3.40 T1 experiments with different algorithms on the same database

T2 EXPERIMENT

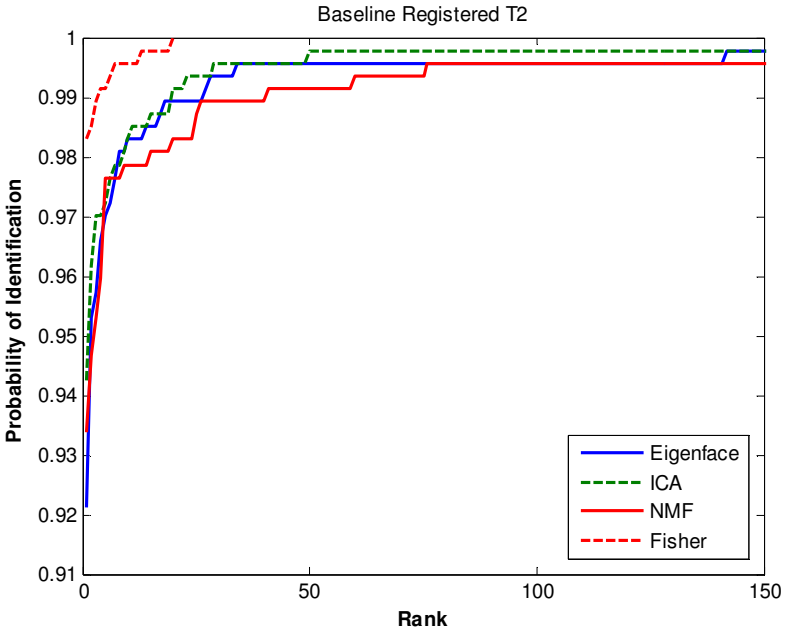

Figure 3.41 T2 experiments with different algorithms on the same database

T3 EXPERIMENT

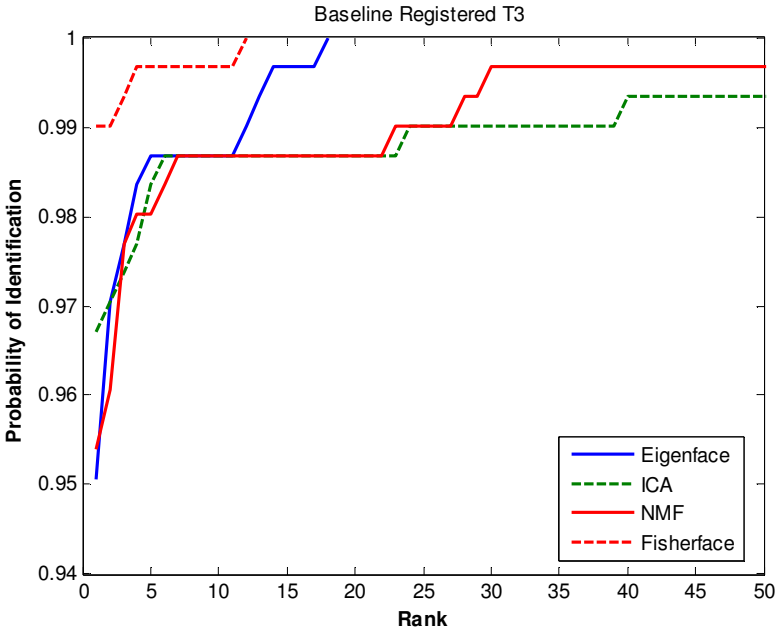

Figure 3.42 T3 experiments with different algorithms on the same database

T4 EXPERIMENT

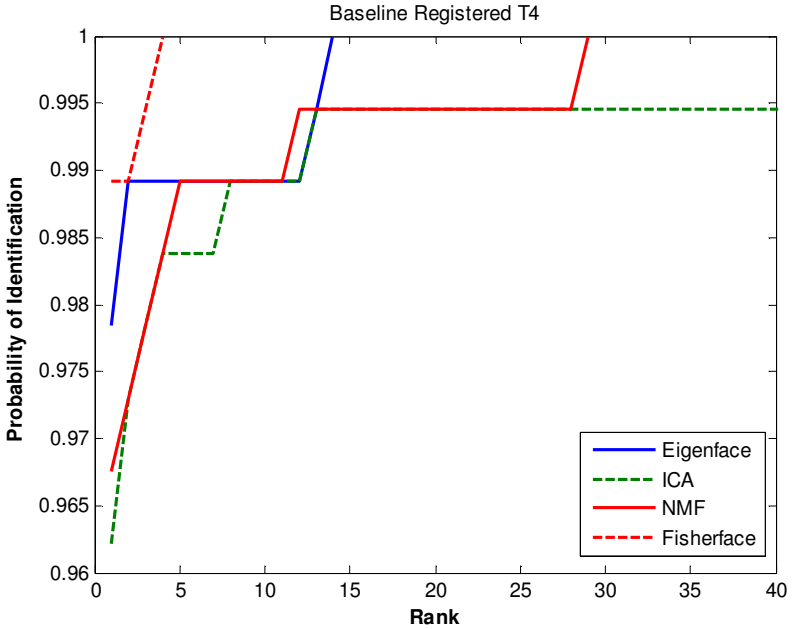

Figure 3.43 T4 experiments with different algorithms on the same database

T5 EXPERIMENT

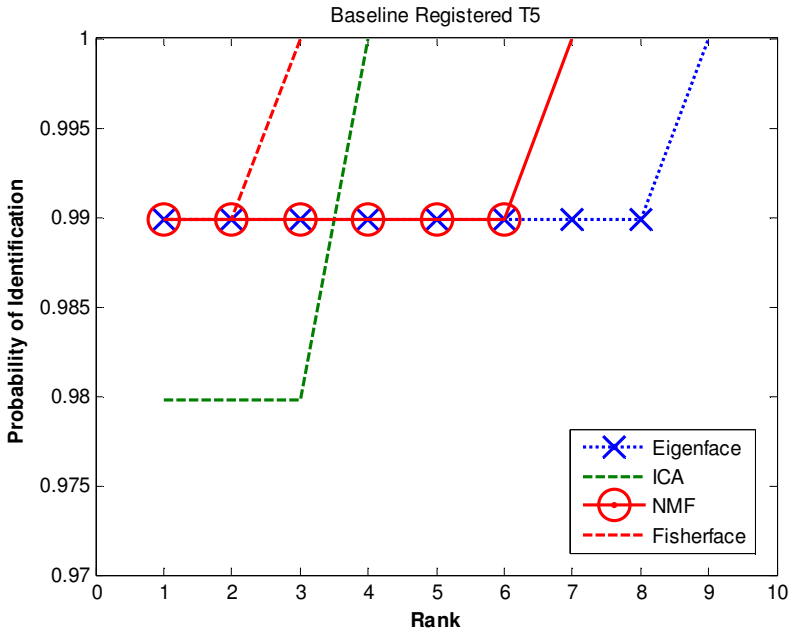

Figure 3.44 T5 experiments with different algorithms on the same database

CHAPTER 4

CONCLUSION

In this thesis different variants of ICP are considered and their performances are evaluated by different recognition methods. During these experiments, effects of some parameters such as sampling rate are considered.

4.1 ICP VARIANTS

If the model and the sample are close to each other, rejecting pairs with greater distances slows down the process. These pairs may carry valuable information in regards to registration parameters. Although it is possible to reach the global minimum by using only the smallest correspondences, this process is very slow when compared to the former.

While defining a threshold by a user, the distribution of the distances between pairs should be considered. Choosing the threshold too small may prevent convergence. Also the speed of convergence reduces greatly if threshold becomes smaller. However for some cases, not rejecting some pairs would surely prevent convergence to global minimum unless they are rejected.

The choice of rejection method (worst n% or std-dev) does not make big difference if two sets are close enough initially.

Subsampling seems to increase the speed. The main reason is that number of correspondences, in other words number of pairs to be processed is reduced.

Random subsampling seems better than uniform subsampling since it gives a chance to each pair to contribute to solution. Firstly, choosing pairs randomly gives a chance to all pairs thus increasing the chance of harvesting information from useful pairs. If an informative point is not included in the iteration, there is a chance of processing it in the next iteration. On the contrary, if an informative pair is not selected in a uniformly sampled subset, it will never be used during registration.

If a point which can mislead or slow down the registration process is included in the subset of uniform subsampling case, this pair will try to disturb the system until the process finishes. However, random subsampling determines the points at the beginning of each iteration which means a sample, which is 'bothering' the system, will be left at the end of the iteration and will not appear in the next iterations unless it is again selected by chance. That is to say, random subsampling does not insist on any points, including inefficient ones.

Grid based registration is faster than point cloud based registration in any case unless the number of samples reach to number of points in the cloud.

If grid form is not sampled dense enough, registration may not lead to global minimum and will be less accurate than using raw point cloud data. However, sampling the data denser would result in a much better situation both in terms of accuracy and speed. This is also true for point cloud based registration. Although the registered images are in point cloud form in M121PC case and M61PC case, former performed better. The only difference is the size of the model or in other words density of correspondences.

If two data sets are close, it is possible to obtain a relatively acceptable registration in a single iteration by constructing a lookup table.

4.2 RECOGNITION ALGORITHMS

As expected the performances of the algorithms get better as the number of training images for each class increases.

When overall performance of recognition algorithms are considered, Fisherface outperforms other methods since it uses class specific information. Eigenface method approaches the problem in a similar manner to Fisherface without regarding class specific information and thus tries to maximize the discrimination of all the samples. As a result it performs worse. Fisherface finds the projection which maximizes the distribution of samples belonging to different classes and which minimizes the distribution of samples of same class.

PCA alone (eigenface method) is not sufficient for making a good face recognition system. However it is a very good method for reducing dimensionality of the data. NMF, Fisherface, ICA algorithms in this thesis all use PCA-reduced data and overperform eigenface method when images are registered.

ICA overperforms eigenface and NMF. PCA considers the correlation between data elements and does not consider dependencies between three or more pixels. ICA can analyze higher order dependencies which is an important advantage to PCA.

4.3 FUTURE STUDIES

ICP is a very efficient method for registering two 3D shapes. However convergence to global minimum is still can not be ensured. In terms of face registration, variants of ICP can be examined in more difficult situations. For instance, faces can be rotated around a single axis or multiple axes and for different cases, by examining the behavior of the variants, a more robust algorithm for registering face image data can be determined. Immunity to noise would be another parameter to work on.

Also more difficult cases such as partially overlapping, occluded scenes can be studied.

3D face recognition is considered to be one of the future technologies. There are many different types of face recognition systems that use 3D information of the face. In this thesis, only a part of approaches were implemented. There are many other methods that can be implemented.

A solid face recognition system should be able to handle with non-rigid deformations of a face such as expressions. In this thesis faces are considered as rigid objects. Almost all the faces in the FRGC v1 database have neutral expressions. Although the effect of change in the facial expression is not tested in this thesis, it is obvious that PCA based approaches will not be able to handle facial expressions. More robust algorithms such as using a deformable model would be more appropriate.

REFERENCES

1. W. Zhao, R.C., P. J. Phillips and A. Rosenfeld, Face Recognition: A Literature Survey. ACM Computing Surveys, 2003. Vol.35(4): p. 399-458.
2. Besl, P. and N. McKay, A Method for Registration of 3D Shapes", IEEE Trans.on PAMI, Vol. 14, pp. 239-256, 1992
3. Szymon Rusinkiewicz and Marc Levoy, “Efficient Variants of the ICP Algorithm:’ in Proceedings of the 3rd International Conference on 3-0 Digital Imaging and Modeling, Quebec City, Canada, pp 145-152, May 2001.
4. Turk, M., Pentland, A. Face recognition using eigenfaces, Proc. CVPR, pp.586-591 (1991).
5. Turk, M. and A. Pentland, Eigenfaces for recognition", Journal of Cognitive Neuroscience, 1991. Volume 3(Number 1): p. 71-86.
6. Ameesh Makadia, A.P.I., and Kostas Daniilidis, Fully Automatic Registration of 3D Point Clouds, Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Vol 1, pp 1297 – 1304, 2006
7. Medioni Gerard, Yang Chen, Object Modeling by Registration of Multiple Range Images. in IEEE International Conference on Robotics and Automation. 1991. Sacramento, California.
8. Lee, D. and H. Seung, Algorithms for Nonnegative Matrix Factorization", Advances in Neural Information Processing Systems, Vol. 13, 2001.
9. Menaka Rajapakse and Lonce Wyse Face Recognition with Non-negative Matrix Factorization. in Visual Communications and Image Processing 2003. 2003.

10. Cartoux, J.Y., J.T. LaPreste, M. Richetin, “Face authentication or recognition by profile extraction from range images,” in Proc. of the Workshop on Interpretation of 3D Scenes, pp.194-199, 1989.
11. Nagamine, T., T. Uemura, I. Masuda, “3D facial image analysis for human identification,” in Proc. ICPR, pp.324-327, 1992.
12. Achermann, B., X. Jiang, H. Bunke, “Face recognition using range images,” in Proc. Int. Conf. on Virtual Systems and MultiMedia, pp.129-136, 1997.
13. Hesher, C., Srivastava, A., Erlebacher, G.: A novel technique for face recognition using range imaging. In: International Symposium on Signal Processing and Its Applications. (2003) 201–204
14. Akarun, L., B. Gokberk, and A. A. Salah, 3D Face Recognition for Biometric Applications", Proc. of the 13th European Signal Processing Conference (EUSIPCO), 2005.
15. Achermann, B. and H. Bunke, Classifying range images of human faces with Hausdorff distance", 15-th International Conference on Pattern Recognition, p.809-813, 2000.
16. Tanaka, H., M. Ikeda, and H. Chiaki, \Curvature-based face surface recognition using spherical correlation principal directions for curved object recognition", Third International Conference on Automated Face and Gesture Recognition, pp. 372-377, 1998.
17. Moreno, A. B., A. Sanchez, J. F. Velez, and F. J. Diaz, \Face recognition using 3D surface-extracted descriptors", Proc. of the Irish Machine Vision and Image Processing Conf., 2003.
18. Gokberk, B., A. A. Salah, and L. Akarun, Rank-based Decision Fusion for 3D Shape-based Face Recognition", Kanade, T., A. Jain, and N. K. Ratha (editors), Proceedings of Audio- and Video-based Biometric Person Authentication, Lecture Notes in Computer Science, Vol. 3456, pp. 1019-1029, 2005.

19. Srivastava, A., X. Liu, and C. Hesher, \Face Recognition Using Optimal Linear Components of Range Images", Image and Vision Computing, Vol. 24, No. 3, pp. 291-299, 2006.
20. Beumier, C. and M. Acheroy, \Automatic 3D Face Authentication", Image and Vision Computing, Vol. 18, No. 4, pp. 315-321, 2000
21. Lee, J.C., Milios, E.: Matching range images of human faces. In: International Conference on Computer Vision. (1990) 722–726
22. Shihong Lao, Masato Kawade, Yasushi Sumi, Fumiaki Tomita: 3D Template Matching for Pose Invariant Face Recognition Using 3D Facial Model Built with Isoluminance Line Based Stereo Vision. ICPR 2000: 2911-2916
23. Yingjie Wang, C.-S.C., Yeong-Khing Ho, Facial feature detection and face recognition from 2D and 3D images. Pattern Recognition Letters, 2002. 23: p. 1191-1202.
24. Bronstein, A. M., M. Bronstein, and R. Kimmel, Expression-invariant 3D face recognition", International Conference on Audio- and Video-Based Person Authentication (AVBPA), Vol. 2688, pp. 62-70, 2003.
25. Chang, K., K. W. Bowyer, and P. J. Flynn, Face recognition using 2D and 3D facial data", ACM Workshop on Multimodal User Authentication, pp. 25-32, 2003.
26. Kevin W. Bowyer, K.C., and Patrick Flynn. A Survey Of Approaches To Three-Dimensional Face Recognition. in 17th International Conference on Pattern Recognition. 2004.
27. Gang Pan, Yijun Wu, Zhaohui Wu, and Wenyao Liu. 3D face recognition by profile and surface matching. In Proc. International Joint Conference on Neural Networks, pages 2168-2174, Portland, Oregon, 2003.
28. Bowyer, K. W., K. Chang, and P. Flynn, A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition",

Computer Vision and Image Understanding, Vol. 101, No. 1, pp. 1-15, 2006.

29. Peter N. Belhumeur, J.P.H., and David J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Transactions On Pattern Analysis And Machine Intelligence, 1997, Vol.19, No.7.
30. T. D. Russ, M. W. Koch and C. Q. Little, “3D face recognition: a quantitative analysis”. IEEE Carnahan Conference on Security Technology, October 2004.
31. Russ, T., M. Koch, and C. Little, A 2D range Hausdorff approach for 3D face recognition", IEEE Workshop on Face Recognition Grand Challenge Experiments, 2005.
32. Xu, C., Y. Wang, T. Tan, and L. Quan, Automatic 3D face recognition combining global geometric features with local shape variation information", Proc. of the Sixth Int. Conf. on Automated Face and Gesture Recognition, pp. 308-313, 2004..
33. Yonguk Lee, H.S., Ukil Yang, Hyungchul Shin, and Kwanghoon Sohn, Local Feature Based 3D Face Recognition. AVBPA 2005, 2005: p. 909-918.
34. Xiaoguang Lu, Dirk Colbry, and Anil K. Jain, Matching 2.5D Face Scans to 3D Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 1, pp. 31-43, 2006.
35. Xiaoguang Lu and Anil K. Jain Deformation Analysis for 3D Face Matching. in Seventh IEEE Workshop on Applications of Computer Vision. 2005.
36. Chang, K., K. Bowyer, and P. Flynn, Adaptive rigid multi-region selection for handling expression variation in 3D face recognition", IEEE Workshop on Face Recognition Grand Challenge Experiments, 2005.

37. Lu, X., A. Jain, and D. Colbry, Matching 2.5D Face Scans to 3D Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 1, pp. 31-43, 2006
38. Gang Pan, Z.W., 3D Face Recognition From Range Data. International Journal of Image and Graphics, 2004. Vol. 5, No.3 pp. 573-593.
39. Zhang, L., A. Razdan, G. Farin, J. Femiani, M. Bae, and C. Lockwood, 3D face authentication and recognition based on bilateral symmetry analysis", Visual Comput (22), p. 43-55, 2006.
40. Feng, S., H. Krim, I. Gu, and M. Viberg, 3D Face Recognition using Affine Integral Invariants", Proc. of. ICASSP, pp. 189-192, 2006.
41. Cheng Zhong, T.T., Chenghua Xu, and Jiangwei Li, Automatic 3D Face Recognition Using Discriminant Common Vectors. ICB 2006, 2005. LNCS 3832: pp. 85-91.
42. Mian, A., M. Bennamoun, and R. Owens, 2D and 3D Multimodal Hybrid Face Recognition", Proc. of. ECCV, LNCS 3953 , p. 344-355, 2006.
43. Xiaoguang Lu and Anil. K. Jain, “Deformation modeling for robust 3D face matching,” in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, 2006, pp. 1377–1383.
44. Andrea F. Abate, Michele Nappi, Stefano Ricciardi, Gabriele Sabatino: Fast 3D face recognition based on normal map. ICIP (2) 2005: 946-949
45. G. Turk and M. Levoy, “Zippered Polygon Meshes From Range Images,” SIGGRAPH 94, pp. 311-318, 1994.
46. Bartlett, M. S., J. R. Movellan, and T. J. Sejnowski, Face Recognition by Independent Component Analysis", IEEE

Transactions on Neural Networks, Vol. 13, No. 6, pp. 1450-1464, 2002.

47. Helin Dutagacı, Bülent Sankur, Yücel Yemez, 3D Face Recognition by Projection Based Methods, SPIE Electronic Imaging Conference: Image and Video Communications and Processing, 16-18 January 2006, San Jose, USA.
48. Duane M. Blackburn, Face Recognition 101: A Brief Primer, 07 April 2003
49. Gerard Blais, a.M.D.L., Registering Multiview Range Data to Create 3D Computer Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995. Vol. 17, No. 8.
50. Rusinkiewicz, S., 3D Scan Matching and Registration, ICCV2005 Short Course. 2005.
51. P. Jonathon Phillips, W.T.S., Alice J. O'Toole, Patrick J. Flynn, Kevin W. Bowyer, Cathy L. Schott, Matthew Sharpe, FRVT 2006 and ICE 2006 Large-Scale Results. 2007.
52. A. Lorusso, D.W.Eggert.and.R.B.Fisher. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. in British Machine Vision Conference.
53. K. S. Arun, T. S. Huang, and S. D. Blostein. Least-square fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Machine Intell., 9(5):698-700, Sept. 1987.
54. Advanced Biometrics Inc, http://www.advanced-biometrics.com/faq/faq.html, last accessed: 27.11.2007
55. Face Recognition Vendor Test, http://www.frvt.org, last accessed: 27.11.2007
56. Bartlett, M.S., Movellan, J.R., & Sejnowski, T.J, http://mplab.ucsd.edu/~marni/code.html, last accessed: 27.11.2007

57. Weik, S. “Registration of 3-D Partial Surface Models Using Luminance and Depth Information,” Proc. 3DIM. 1997.
58. Takeshi Masuda, K.S., Naokazu Yokoya. Registration and Integration of Multiple Range Images for 3-D Model Construction. in ICPR'96. 1996.
59. Chitra Dorai, G.W., Anil K. Jain, and Carolyn Mercer, Registration and Integration of Multiple Object Views for 3D Model Construction. IEEE Transactions On Pattern Analysis And Machine Intelligence, 1998, Vol. 20, No.1.