

3D FACE REPRESENTATION AND RECOGNITION USING  
SPHERICAL HARMONICS

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

BY

FAHRİ TUNÇER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2008

Approval of the Thesis

**3D FACE REPRESENTATION AND RECOGNITION USING  
SPHERICAL HARMONICS**

submitted by **FAHRİ TUNÇER** in partial fulfillment of the requirements for the degree  
of **Doctor of Philosophy 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 Erkmén

Head of Department, **Electrical and Electronics Eng.** \_\_\_\_\_

Prof. Dr. . Uğur Halıcı

Supervisor, **Electrical and Electronics Eng. Dept., METU** \_\_\_\_\_

**Examining Committee Members**

Prof. Dr. Kemal Leblebiciođlu

Electrical and Electronics Eng. Dept., METU \_\_\_\_\_

Prof. Dr. Uğur Halıcı

Electrical and Electronics Eng. Dept., METU \_\_\_\_\_

Prof. Dr. Volkan Atalay

Computer Engineering Dept., METU \_\_\_\_\_

Assist. Prof. Dr. İlkey Ulusoy

Electrical and Electronics Eng. Dept., METU \_\_\_\_\_

Assist. Prof. Dr. Pınar Duygulu Şahin

Computer Engineering Dept., BİLKENT University \_\_\_\_\_

**Date:** 28.08.2008

**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 : Fahri TUNÇER

Signature: \_\_\_\_\_ :

## **ABSTRACT**

### **3D FACE REPRESENTATION AND RECOGNITION USING SPHERICAL HARMONICS**

Tunçer, Fahri

Ph.D., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Uğur Halıcı

August 2008, 134 pages

In this study, a 3D face representation and recognition method based on spherical harmonics expansion is proposed. The input data to the method is range image of the face. This data is called 2.5 dimensional. Input faces are manually marked on the two eyes, nose and chin points. In two dimensions, using the marker points, the human face is modeled as two concentric half ellipses for the selection of region of interest. These marker points are also used in three dimensions to register the faces so that the nose point tip is at the origin and the line across the two eyes lies parallel to the horizontal plane. A PCA based component analysis is done to further align the faces vertically. The aligned face is stitched and mapped to an ellipsoid and transformed using real spherical harmonics expansion. The real harmonics expansion coefficients are labeled and stored into a gallery. Using these coefficients as input, several classification algorithms are applied and the results are reported.

Keywords: Face Recognition, 3D Face Recognition, and Spherical Harmonics.

## ÖZ

### KÜRESEL HARMONİKLERLE 3 BOYUTLU YÜZ MODELLEME VE TANIMA

Tunçer, Fahri

Doktora, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Uğur Halıcı

Ağustos 2008, 134 sayfa

Bu tez çalışmasında küresel harmonik dönüşümüne dayalı bir 3 boyutlu yüz modelleme ve tanıma yöntemi geliştirilmiştir. Bu yönteme girdi olarak yüz derinlik bilgisi kullanılmıştır. Bu derinlik verisi 2.5 boyutlu olarak isimlendirilir. Girdi yüzlerde manuel olarak, iki göz, burun ve çene noktaları işaretlenmektedir. İki boyutta, istenilen yüz bölgesinin seçilmesi için, işaretlenen noktalar kullanılarak, yüz eş merkezli iki yarım elips olarak modellenmiştir. Ayrıca yine bu noktalar kullanılarak yüz 3 boyutta burun merkez ve gözleri birleştiren çizgi yatay eksene paralel olacak şekilde hizalanmıştır. PCA bazlı bir yöntemle yüz dikey eksende de hizalanmıştır. Hizalanan yüzler bir elipsoid üzerine dikilerek yerleştirilmiş, ve sonrasında gerçek küresel harmonikler kullanılarak dönüştürülmüşlerdir. Dönüşüm parametreleri etiketlenerek bir yüz galerisi oluşturulmuştur. Bu dönüşüm parametreleri kullanılarak farklı sınıflandırma yöntemleri kullanılmış ve sonuçları karşılaştırmalı olarak raporlanmıştır.

Anahtar kelimeler: Yüz Tanıma, 3 Boyutlu Yüz Tanıma, Küresel Harmonikler.

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

ABSTRACT .....	iv
ÖZ .....	v
ACKNOWLEDGEMENTS .....	vii
TABLE OF CONTENTS .....	viii
CHAPTER	
1. INTRODUCTION.....	1
1.1 Problem Definition and Motivation.....	1
1.2 The Data Used .....	4
1.3 The Proposed Face Recognition System.....	5
1.4 Contribution.....	6
1.5 Organization of the Thesis .....	10
2. LITERATURE REVIEW.....	11
2.1 Biometrics .....	11
2.2 Face Recognition Methods.....	12
2.3 Face Recognition Algorithm Performance Evaluation Metrics .....	14
2.4 2D Methods for Face Recognition.....	17
2.4.1 Eigenfaces .....	17
2.4.2 Nonnegative Matrix Factorization (NMF).....	19
2.4.3 Fischerface .....	20
2.4.4 Independent Component Analysis (ICA).....	22
2.4.5 Elastic Bunch Graph Matching (EBGM).....	23
2.4.6 Kernel Methods .....	23
2.4.7 Active Appearance Model .....	24
2.4.8 Trace Transform .....	24
2.4.9 Bayesian Framework .....	25
2.4.10 Support Vector Machines.....	25
2.5 3D Methods for Face Recognition.....	29
2.6 Distance Metrics .....	32

2.6.1	Euclidean Distance .....	33
2.6.2	Mahalanobis Distance.....	34
3.	SPHERICAL HARMONICS EXPANSION .....	35
3.1	Spherical Harmonic Functions .....	35
3.2	Spherical Harmonics Expansion.....	37
3.3	Applications of Spherical Harmonics .....	48
3.3.1	Spherical Harmonic Lighting .....	48
3.3.2	Modeling Face Illumination Variation with Spherical Harmonics .....	48
3.3.3	Spherical Harmonics as 3D Shape Descriptor .....	50
3.3.4	Face Recognition via Spherical Embedding .....	52
3.4	Fast Spherical Harmonics Transform using FFT .....	54
3.4.1	Pixelization Assumptions.....	54
3.4.2	Forward SHT.....	55
3.4.3	The Backward SHT .....	56
4.	3D FACE REPRESENTATION AND RECOGNITION USING SHT. ....	57
4.1	Input face data .....	58
4.2	Preprocessing of Images .....	61
4.2.1	Gap Filling .....	62
4.2.2	Noise filtering.....	63
4.2.3	Marking the Points on the Face.....	65
4.2.4	Cropping .....	66
4.2.5	Registration .....	68
4.3	Application of Spherical Harmonic Projection .....	72
4.3.1	Partial Mapping Trial.....	73
4.3.2	Hemi-spherical Mapping Trial.....	78
4.3.3	Full-Spherical Mapping .....	80
4.3.4	Full-Spherical Mapping with Symmetry .....	84
4.4	SH Transformation .....	85
4.4.1	Power Spectral Density.....	86
4.4.2	Rotation Invariancy .....	87

4.4.3	Signal to Noise Ratio .....	89
4.4.4	Effect of Symmetry .....	90
4.4.5	Effect of a Single Coefficient.....	91
4.5	Face Classification using SHT Coefficients.....	92
4.5.1	k-Nearest Neighbor with k=1 for SHT Coefficients .....	93
4.5.2	k-Nearest Neighbor with k=1 for PSD Coefficients.....	95
4.5.3	Linear Disriminant Analysis .....	98
4.5.4	Support Vector Machines .....	99
4.6	Confusion Matrix.....	100
5.	EXPERIMENTS. ....	104
5.1	Face Database .....	104
5.2	Results .....	106
5.2.1	T1 Experiments .....	106
5.2.2	T2 Experiments .....	107
5.2.3	T3 Experiments .....	108
5.2.4	T4 Experiments .....	109
5.3	ROC Analysis.....	110
5.4	Results from Literature for Comparison .....	111
6.	CONCLUSION. ....	112
6.1	Future Study .....	113
	REFERENCES .....	116
	APPENDICES	
A.	ITERATIVE CLOSEST POINT .....	124
B.	NONNEGATIVE MATRIX FACTORIZATION (NMF).....	127
C.	KERNEL PCA .....	129
D.	SPHERICAL HARMONICS LIGHTING.....	132
	CURRICULUM VITAE .....	134

## LIST OF TABLES

### TABLES

Table 4.1 Distances between faces in terms of SHT coefficients.....	93
Table 4.2 Coefficients for the first 4 bands .....	94
Table 4.3 Face Distances after whitening transform.....	95
Table 4.4 Mean and Variance of the PSD .....	96
Table 4.5 Face distances using PSD .....	98
Table 4.6 A sample confusion matrix .....	101
Table 5.1 FRGC Database .....	104
Table 5.2 Images used in experiments .....	105
Table 5.3 Top-10 Rank scores for experiment T1 .....	107
Table 5.4 Top-10 Rank scores for experiment T2.....	108
Table 5.5 Top-10 Rank scores for experiment T3.....	109
Table 5.6 Top-10 Rank scores for experiment T4.....	110

## LIST OF FIGURES

### FIGURES

Figure 1.1 Face recognition problem .....	3
Figure 1.2 Range image and corresponding texture image .....	4
Figure 1.3 3D rendered range data using texture color .....	4
Figure 1.4 Recognition system .....	5
Figure 1.5 Cropped and aligned face image .....	9
Figure 1.6 Cropped and stitched face image .....	9
Figure 2.1 Spreading of biometrics (from [1]) .....	11
Figure 2.2 Face Recognition Methods .....	13
Figure 2.3 A CMC curve (from [15]) .....	15
Figure 2.4 A ROC curve (from [15]) .....	16
Figure 2.5 Sample eigenfaces, corresponding to 8 biggest eigenvalues (from [19]).....	19
Figure 2.6 Architecture of ICA.....	23
Figure 2.7 Separating hyperplanes.....	26
Figure 2.8 Separating hyperplane with maximum margin .....	26
Figure 2.9 Extended Gaussian Image of a cube .....	30
Figure 3.1 Orthogonal basis expansion .....	37
Figure 3.2 Orthogonal basis expansion .....	38
Figure 3.3 Orthogonal basis expansion .....	38
Figure 3.4 The first 10 associated Legendre polynomials.....	39
Figure 3.5 $Y_{00}(\theta, \varphi)$ .....	42
Figure 3.6 $Y_{10}(\theta, \varphi)$ .....	42
Figure 3.7 $Y_{11}(\theta, \varphi)$ .....	43
Figure 3.8 $Y_{20}(\theta, \varphi)$ .....	43
Figure 3.9 $Y_{21}(\theta, \varphi)$ .....	43
Figure 3.10 $Y_{22}(\theta, \varphi)$ .....	44

Figure 3.11 $Y_{30\theta}, \varphi$ .....	44
Figure 3.12 $Y_{31\theta}, \varphi$ .....	44
Figure 3.13 $Y_{32\theta}, \varphi$ .....	45
Figure 3.14 $Y_{33\theta}, \varphi$ .....	45
Figure 3.15 The first 5 SH bands.....	46
Figure 3.16 SH expansion with increasing orders of approximation ([8]).....	47
Figure 3.17 Rotation Invariancy of SH transform.....	47
Figure 3.18 Images of a face, lit from a number of different directions. ....	49
Figure 3.19 Spherical harmonic shape descriptor (from [8]).....	51
Figure 3.20 Precision vs recall performance for 3d SH shape descriptor [8].....	51
Figure 4.1 FRGC registered color image.....	59
Figure 4.2 FRGC range image.....	59
Figure 4.3 FRGC validity flag map.....	60
Figure 4.4 Range image shown as a mesh.....	60
Figure 4.5 Face reconstructed using range and color image.....	61
Figure 4.6 Gaps on the face.....	62
Figure 4.7 Face after filling.....	63
Figure 4.8 Noisy artifacts on the face surface.....	64
Figure 4.9 Image after spike removal.....	64
Figure 4.10 Effect of smoothing filter.....	65
Figure 4.11 Points marked on the face.....	66
Figure 4.12 Elliptical crop region.....	67
Figure 4.13 Cropped 2D image.....	67
Figure 4.14 Cropped 3D image and registration triangle.....	68
Figure 4.15 Registered face image centered at nose tip.....	69
Figure 4.16 Face after offsetting on nose point.....	70
Figure 4.17 Rotations around z and y-axes.....	71
Figure 4.18 PCA Alignment.....	72
Figure 4.19 Partial mapping extent.....	74
Figure 4.20 Partial mapping.....	74
Figure 4.21 Bandwidth $L_{\max}=1$ .....	76

Figure 4.22 Bandwidth $L_{max}=10$ .....	76
Figure 4.23 Bandwidth $L_{max}=50$ .....	77
Figure 4.24 Bandwidth $L_{max}=100$ .....	77
Figure 4.25 Bandwidth $L_{max}=300$ .....	78
Figure 4.26 Hemi-spherical mapped surface from left-viewpoint.....	79
Figure 4.27 Reconstructed hemi-spherical surface .....	80
Figure 4.28 Full spherical mapping .....	81
Figure 4.29 Stitching band of the previous face image.....	82
Figure 4.30 Stitched face surface.....	82
Figure 4.31 Full-spherical face surface .....	83
Figure 4.32 The reconstructed surface for $L_{max}=200$ .....	83
Figure 4.33 Full-spherical with symmetry face.....	84
Figure 4.34 Reconstructed symmetrical face for $L_{max}=200$ .....	85
Figure 4.35 Spherical Harmonics Coefficients of a sample face.....	86
Figure 4.36 PSD for a sample face .....	87
Figure 4.37 Hemi-spherical face surface (left), the same surface rotated around the nose tip (right).....	87
Figure 4.38 Rotation invariancy. PSD of the surfaces given in Figure 4.37.....	88
Figure 4.39 SNR vs bandwidth.....	89
Figure 4.40 PSD of symmetrically mapped face .....	90
Figure 4.41 SNR for symmetrically mapped surface.....	91
Figure 4.42 Effect of a Single Coefficient .....	91
Figure 4.43 Test faces .....	94
Figure 4.44 Mean and Variance of PSD .....	96
Figure 4.45 Confused subjects.....	102
Figure 4.46 PSD plot of images of confused subjects .....	102
Figure 4.47 1-D LDA projection of the faces of confused subjects .....	103
Figure 5.1 CMC curve for experiment T1.....	106
Figure 5.2 CMC curve for T1 from [19] .....	107
Figure 5.3 CMC Curve for Experiment T2 .....	107
Figure 5.4 CMC Curve from [19].....	108

Figure 5.5 CMC curve for experiment T3.....	108
Figure 5.6 CMC curve from [19].....	109
Figure 5.7 CMC curve for experiment T4.....	109
Figure 5.8 CMC Curve from [19].....	110
Figure 5.9 Verification ROC .....	111
Figure A.1 ICP.....	125
Figure B.1 NMF factorization .....	128
Figure D.1 Light transfer function (left), estimated spherical plot(right); .....	133
Figure D.2 Scene lit using shadowed spherical harmonics lighting method in real-time.....	133

## LIST OF ABBREVIATIONS

2D	2 Dimensional
2.5D	2.5 Dimensional
3D	3 Dimensional
AAM	Active Appearance Model
CMC	Cumulative Match Characteristics
EBGM	Elastic Bunch Graph Matching
EER	Equal Error Rate
EGI	Extended Gaussian Image
EVD	Eigenvalue Decomposition
FAR	False Acceptance Rate
FFT	Fast Fourier Transform
FLD	Fisher Linear Discriminant
FRGC	Face Recognition Grand Challenge
FRR	False Rejection Rate
FRVT	Face Recognition Vendor Test
ICA	Independent Component Analysis
ICP	Iterative Closest Point
k-NN	k Nearest Neighbor
LDA	Linear Discriminant Analysis
MSE	Mean Square Error
MTT	Masked Trace Transform
NMF	Nonnegative Matrix Factorization
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristics
PSD	Power Spectral Density
SFS	Shape From Shading
SHT	Spherical Harmonics Transform
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
SVM	Support Vector Machines
VRML	Virtual Reality Markup Language
WTT	Weighted Trace Transform

# CHAPTER 1

## INTRODUCTION

### 1.1 Problem Definition and Motivation

Face recognition is a challenging problem because of the diversity of faces and variations caused by expressions, gender, pose, illumination, age and makeup. Automatic face recognition has been traditionally associated with the fields of computer vision and pattern recognition. Face recognition is considered a widely accepted biometric identification method [1], [2]. Even it has the potential of becoming the leading biometric technology in near future. Unfortunately, it is also one of the most difficult pattern recognition problems. So far, all existing solutions provide only partial, and usually unsatisfactory, answers to the recognition problem.

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. The performance of face recognition systems that use two dimensional (2D) images is dependent on consistent conditions such as lighting, pose and facial expression. While most efforts have been devoted to face

recognition from 2D images , a few approaches have utilized depth information provided by 2.5D range images. Current 2D face recognition systems can achieve good performance in constrained (or controlled) environments, however, they still encounter difficulties in handling large amounts of facial variations due to head poses, lighting conditions and face expressions. Because the human face is a 3D object whose 2D projection (image) is sensitive to the above changes, utilizing 3D face information can improve the face recognition performance. Range images captured explicitly by a 3D sensor present face surface shape information.

3D face recognition is a relatively recent trend that in some sense breaks the long-term tradition of mimicking the human visual recognition system, like the 2D methods attempt to do. As evaluations such as the Face Recognition Vendor Test (FRVT) demonstrate in an unarguable manner that current state of the art in 2D face recognition is insufficient for high-demanding biometric applications [1], trying to use 3D information has become an emerging research direction in hope to make face recognition more accurate and robust.

Before going further, it is valuable to define the term *recognition* and to distinguish between the problem of *authentication* (verification) and that of *recognition*.

- In the *authentication* problem, the “probe” individual requests a privilege by claiming the identity of a person whose template is stored in the “gallery” database. The data used for a specific recognition task is defined as a template. The authentication algorithm needs to retrieve the template from the gallery and to compare it with the given face and verify their equivalence. Shortly, the authentication problem is a one-to-one matching problem. Such a biometric technology can be used to secure financial transactions, for example, in an automatic teller machine (ATM)

or to secure the entrance into security zones. In the authentication case, it is assumed that the user of the system is collaborative.

- The *recognition* problem is much more difficult. Recognition means that the given probe should be compared with all the templates stored in the gallery database. The face recognition algorithm should match a given face with one of the templates in the database. Finding a criminal in a crowd is one such application. Face recognition problem is one-to-many matching problem. It should be mentioned that no collaboration is assumed in this case. At current technological level, one-to-many face recognition with *non-collaborative* users is practically unsolvable [3].

The main goal of this thesis study is to develop a method or algorithm to distinguish and recognize human faces using the 2.5 dimensional (2.5D) scans of human faces which are acquired by special hardware. Our study will focus on 2.5D dimensional, or 3D in general, face recognition problem. In our study, we will assume that the users are collaborative, i.e. template and probe data are acquired in a controlled environment.

The general face recognition problem can be depicted as follows:

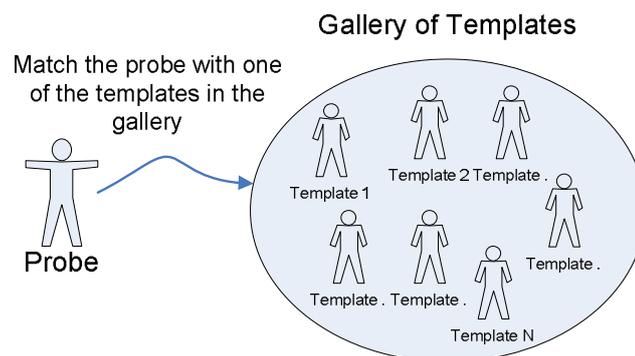


Figure 1.1 Face recognition problem

The template and probe data consists of 2.5D range images of the human face. 2.5D face range image is a simplified 3D  $(x,y,z)$  surface representation that contains at most one depth value ( $z$ -direction or away from the camera projection plane) for every point in the  $(x,y)$  plane. 2.5D range images are acquired using a special hardware.

## 1.2 The Data Used

In this study, we used the sample data set provided by Face Recognition Grand Challenge (FRGC). FRGC human face range image is a 2D image which contains a distance value for every  $x,y$  pair on the image plane. Fortunately, the FRGC dataset also contains a registered texture image for every range scan. A sample 2.5D range scan, its corresponding texture and its 3D rendering is shown below. The FRGC dataset is acquired with Minolta Vivid 900/910 series sensor [4].



Figure 1.2 Range image and corresponding texture image

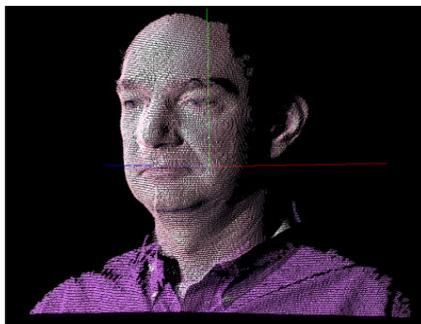


Figure 1.3 3D rendered range data using texture color

As seen on the Figure 1.2, the range image and the face texture image can be easily distinguished from its background. The first image is the range scan. Darker regions are closer to the camera. The second image is the texture image associated with the range image. Figure 1.3 shows the 3D rendered image of the range scan colored with the texture image.

The FRGC dataset is obtained in a controlled environment, i.e. location of the head with respect to camera, and the lighting is setup appropriately [4].

Notice that there are gaps in the range image, where the range sensor did not work well. For example, there are gaps above the eyes. These gap points are points where there is no depth information on the range image. These gap points must be handled by the recognition system. And also, there are unwanted regions on the range image such as shoulders, or neck. These unwanted regions will be cropped by the recognition system.

### 1.3 The Proposed Face Recognition System

The overall process workflow of the proposed recognition system can be represented with the diagram given in Figure 1.4;

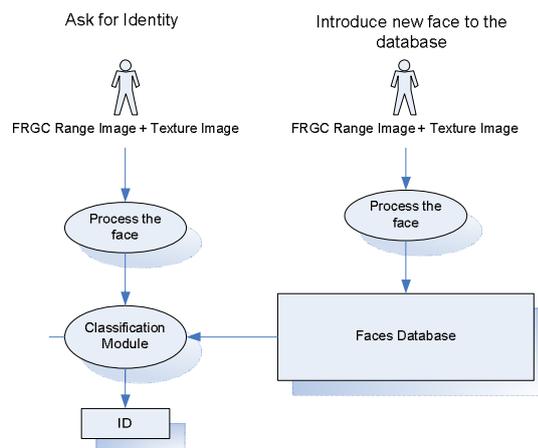


Figure 1.4 Recognition system

There are two main tasks of the proposed system. First, populate the face database by introducing new face data to the system. Assuming that the new face data is not already in the database, the face data is processed and recorded into the database (shown on the right flow in Figure 1.4). Second task is to request for the identity for the face data, first by processing it and afterwards by matching it one of the faces in the database (shown on the left flow).

The processing module of the system works as follows;

- Preprocess the 3D face data
- Crop the face and place it on ellipsoid
- Transform the spherical face by real Spherical Harmonics Transform (SHT) and obtain the coefficients

The input to the classification module is the real SHT coefficients obtained at the processing module. Several algorithms are used in the classification module and their performances are compared. These algorithms are;

- k-Nearest Neighbor (k-NN) with Euclidean distance metric
- k-Nearest Neighbor (k-NN) with Mahalanobis distance
- Linear Discriminant Analysis (or Fisher Discriminant Analysis)
- Support Vector Machines

#### **1.4 Contribution**

In this study we propose a 3D face recognition algorithm based on spherical harmonics decomposition of the face range image. The novel part of the study is to use real spherical harmonics transform for an *intra-class* classification problem.

In literature, there are some studies which use spherical harmonics decomposition for 3D shape modeling, matching and retrieval problems [5], [6], [7], [8]. However, all these studies concentrated on inter-class classification problems, i.e. they proposed systems to distinguish objects from different

classes; for example they tried to distinguish an airplane from a vase. And the way they model the shapes are completely different than our method.

It is well known that any non-reentrant 3D surface can be accurately modeled using simultaneous evaluation of spherical harmonics coefficients [6]. It means that the surface or function which will be transformed by spherical harmonics must be a spherical function, or in other words it must be a one-to-one function in spherical radial direction. Because of this requirement, most of the above mentioned studies converted the 3D models into volumetric datasets, and using this voxel grid they computed shape descriptors by intersecting the model with concentric spheres. Assuming that these intersections are spherical functions, they computed the frequency decomposition of each spherical function. Fortunately, the human face or more specifically the frontal range scan of the human face can be modeled as a spherical function by mapping it onto a sphere.

Spherical harmonics decomposition is not a new topic to the face recognition community. [9], [10] used spherical harmonics for face recognition.

Zhanfeng et al. ([10]) used spherical harmonics to overcome the illumination problem by encoding the pose information by spherical harmonics. Under the assumption of Lambertian reflectance, they modeled the illumination variations for a fixed pose. Using only a face image which is semi-frontal and illuminated by multiple light sources, they were able to synthesize a different view of the face under arbitrary lighting. They utilized the fact that 2D harmonic basis images at different poses are related to close-form linear transformations.

Bronstein et al. ([9]) used spherical harmonics transform to compute the dissimilarity measure between the *spherical canonical images*. They compute expression-invariant signatures based on isometry-invariant representation of the facial surface. They propose to embed the face geometric structure into a spherical space, and they call these new invariants as spherical canonical images.

Our study is mainly interested in efficient representation of the face range image by spherical harmonics. Most computer vision and pattern recognition algorithms aim to represent an object with the minimum amount of data and maximum possible detail so as to increase the efficiency. Fourier, Fourier related or wavelet based transforms can be used for approximations in 2D situations, like an image, and can be extended to 3 or more dimensions. However, for our case where the 3D face points are represented by two angular spatial coordinates and radius from some centre point, a more appropriate set of orthogonal basis functions turn out to be the spherical harmonics.

Our approach to make the face range image “spherical” assumes an ellipsoidal face model. In fact, what we call ‘spherical’ is an “ellipsoid” defined with two different radiuses. Figure 1.5 and 1.6 show a cropped face and a spherically mapped face. Notice that the spherical face is stitched on the ellipsoid in the spherically mapped face.

The main reason of stitching the face on an ellipsoid is to model the face shape as a complete spherical function, i.e. define it over the whole angular extends. If we don't complete the face shape spherically using an ellipsoid, we'll end up with very high values at higher spatial frequencies because of the discontinuity at face boundaries. Since we're trying to model the face with fewer coefficients and store these coefficients into template gallery for further comparison with the given probe image, high frequency SHT coefficients are not suitable for our classification purposes.

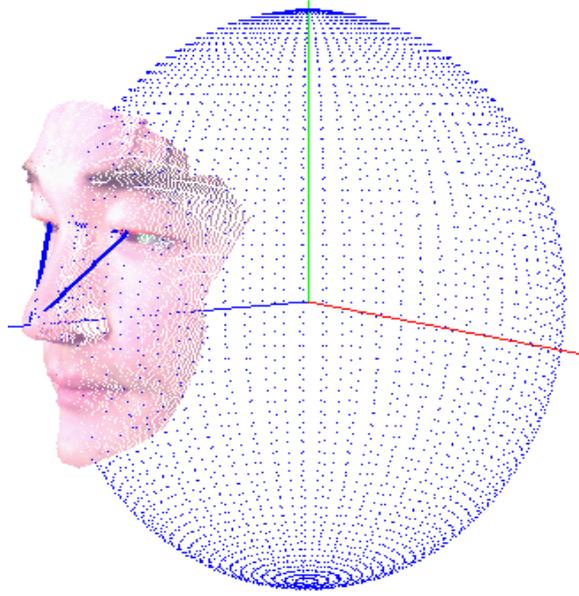


Figure 1.5 Cropped and aligned face image

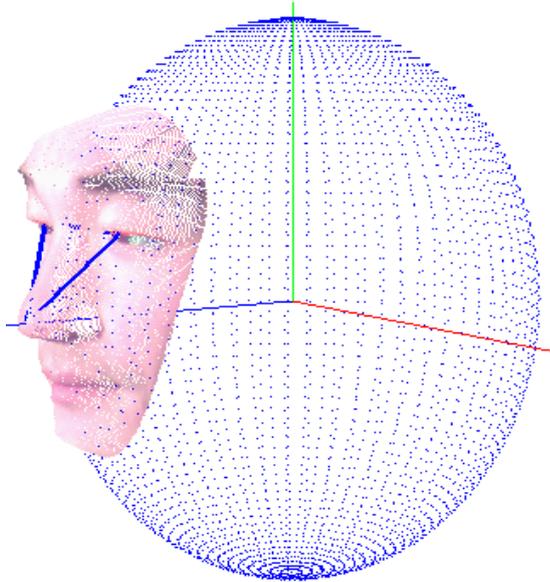


Figure 1.6 Cropped and stitched face image

## **1.5 Organization of the Thesis**

In Chapter 2, literature review of face recognition, both of 2D and 2.5D/3D approaches are summarized.

Chapter 3 will be about spherical harmonic functions and its application in computer graphics and face recognition.

In Chapter 4, we will give a detailed description of our face recognition algorithm.

In Chapter 5 results and comparison of our algorithm with other methods are given.

Chapter 6 is the conclusion chapter.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Biometrics

Many recent events, such as terrorist attacks, showed that there are serious weaknesses in most sophisticated security systems. Various government agencies are now more motivated to improve security data systems based on body or behavioral characteristics, often called biometrics [11].

Perhaps the most common biometrics modalities are fingerprints and iris, but many other human characteristics have been studied in last years: finger/palm geometry, voice, signature, face. Figure 2.1 shows the spreading of the most popular biometrics in the last years from a commercial point of view [1].

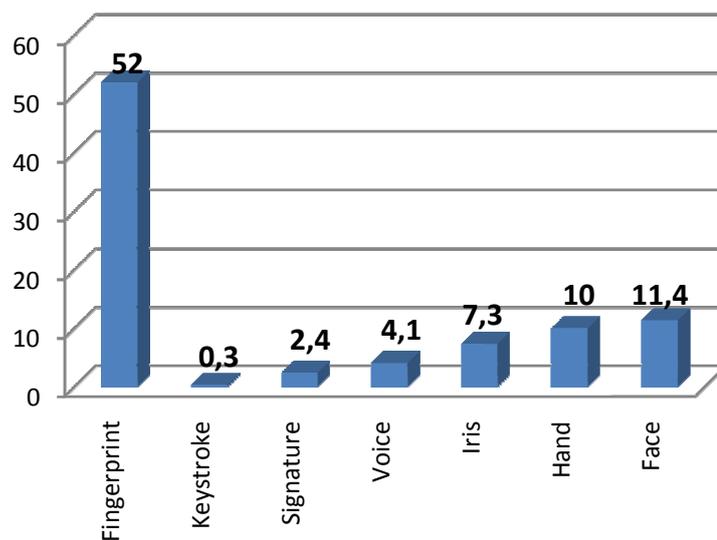


Figure 2.1 Spreading of biometrics (from [1])

Biometrics have drawbacks. Iris recognition is extremely accurate, but expensive to implement and not very accepted by people. Fingerprints are reliable and non-intrusive, but not suitable for non-collaborative individuals. On the contrary, face recognition seems to be a good compromise between reliability and social acceptance and balances security and privacy well.

## **2.2 Face Recognition Methods**

Face recognition and verification have been at the top of the research agenda of the computer vision community for more than two decade. To date, most of the research efforts, as well as commercial developments, have focused on 2D approaches. This focus on monocular imaging has partly been motivated by costs but to a certain extent also by the need to retrieve faces from existing 2D image and video database. Last but not least, it has been inspired by the ability of human vision to recognize a face from single photographs where the 3D information about the subject is not available and therefore the 3D sensing capability of the human perception system cannot be brought to bear on the interpretation task.

Since the automatic face recognition problem is still an unsolved problem of the computer vision community, there are so many research groups and institutions around the world trying to find a first-class solution to the problem. In accordance, there are so many face recognition methods developed up to date. Several techniques are developed and applied to the problem. All these methods can be classified as shown in Figure 2.2.

In the last decade, major advances occurred in face recognition, with many systems capable of achieving recognition rates greater than 90% [12]. However real-world scenarios remain a challenge, because face acquisition process can undergo to a wide range of variations. There are five key factors that can significantly affect system face recognition performances:

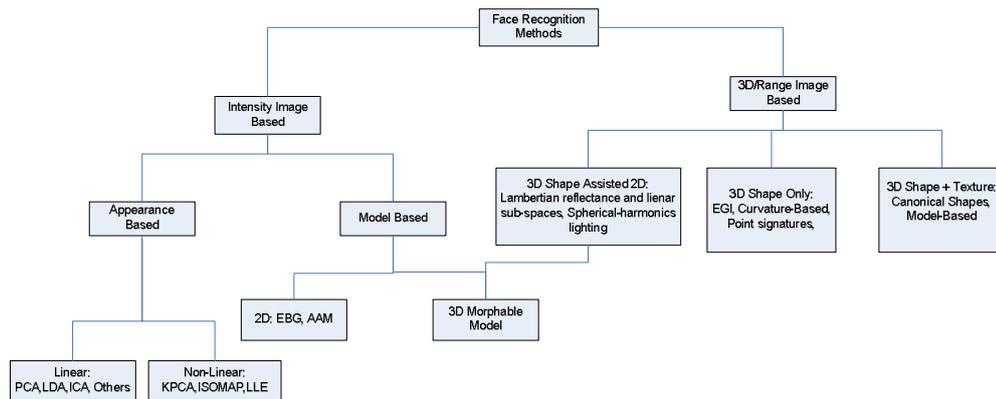


Figure 2.2 Face Recognition Methods

- ***Illumination variations*** due to skin reflectance properties and due to the internal camera control. Several 2D methods do well in recognition tasks only under moderate illumination variation, while performances noticeably drop when both illumination and pose changes occur.
- ***Pose changes*** affect the authentication process, because they introduce projective deformations and self-occlusion. Even if methods dealing with up to 32 head rotation exist, they do not solve the problem considering that security cameras can create viewing angles that are outside of this range when positioned. On the contrary, with exception of extreme expressions such as scream, the algorithms are relatively robust to facial expression.
- ***Time delay*** is another important factor, because the face changes over time, in a nonlinear way over long periods. In general this problem is harder to solve with respect to the others and not much has been done especially for age variations.
- ***Occlusions*** can dramatically affect face recognition performances, in particular if they located on the upper-side of the face, such as hair occluding the eyes.

As can be seen on Figure 2.2, most work on face recognition is carried out using 2D intensity images. Although these techniques can achieve good performances under controlled conditions, their performance can degrade drastically 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 [13]. 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) [4] aim to improve face recognition systems by encouraging and challenging more researchers into this area.

FRVT 2006 results [14] 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.

### **2.3 Face Recognition Algorithm Performance Evaluation Metrics**

In order to evaluate performances of face recognition systems, some common terms are employed [15]. *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 2.3.

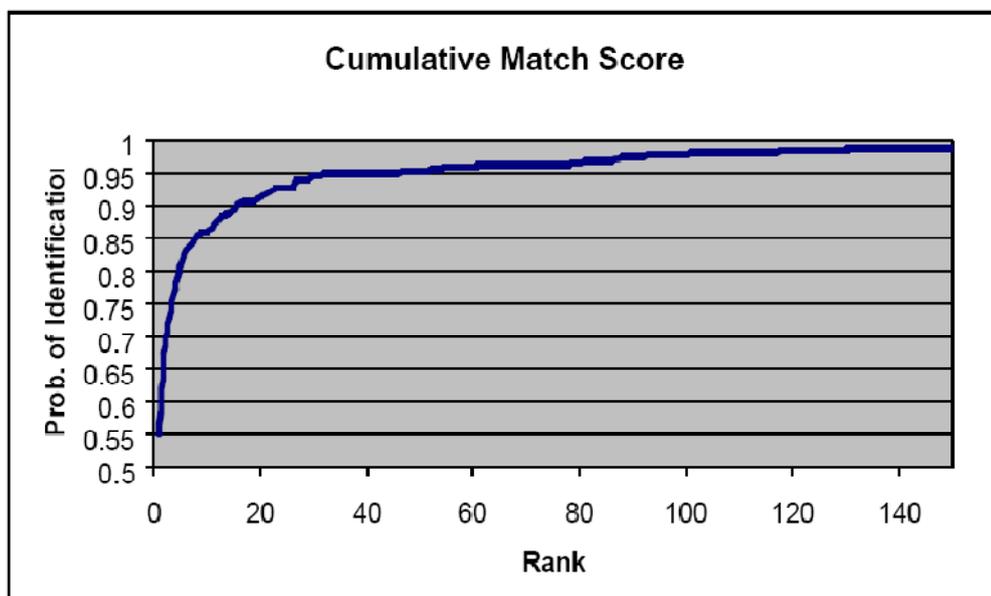


Figure 2.3 A CMC curve (from [15])

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 2.4.

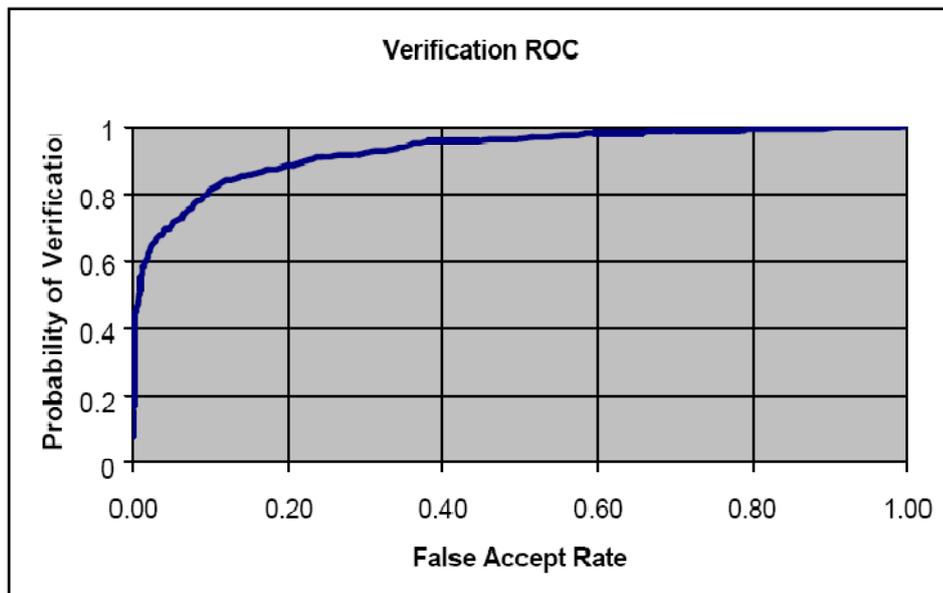


Figure 2.4 A ROC curve (from [15])

## 2.4 2D Methods for Face Recognition

Zhao et al. [16] have an excellent literature survey about the face recognition methods. The website [www.face-rec.org](http://www.face-rec.org) [17] also contains valuable information about face recognition methods.

### 2.4.1 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 [18]. 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 the dimension of the data is reduced by eliminating the vectors having the smallest eigenvalues.

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  $(\Gamma_1, \Gamma_2, \dots, \Gamma_M)$ .  $M$  is the number of faces in the training set.
- Calculate the mean  $(\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n)$ .  $n$  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  $A = [\Phi_1 \Phi_2 \dots \Phi_M]$

- 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

$$weight(projected, i) = \sum_{i=1}^k (\Gamma_{test} - \Psi) \cdot eigenvector_i \quad (2.1)$$

$$\Gamma_{projected} = \sum_{i=1}^k weight(projected, i) \cdot eigenvector_i \quad (2.2)$$

- 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

$$Error = \sum_{i=1}^k (weight(projected, i) - weight(training, i))^2 \quad (2.3)$$

In general  $C$  matrix is so large that finding its eigenvectors becomes very difficult. In [18] Turk and Pentland propose a way to determine the eigenvectors of the covariance matrix  $C$ .

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

$$A^T A v_i = \mu_i v_i \quad (2.4)$$

Multiplying both sides by  $A$  from left, we obtain  $AA^T A v_i = \mu_i A v_i$  which means  $A v_i$  are the eigenvectors of  $C$ . Therefore firstly the eigenvectors of  $L$  is found to reduce the calculations. Finding the eigenvectors of  $C$  is now simpler by multiplying with the inverse of  $A$ .



Figure 2.5 Sample eigenfaces, corresponding to 8 biggest eigenvalues (from [19])

### 2.4.2 Nonnegative Matrix Factorization (NMF)

Another 2D method for face recognition is the NMF algorithm. The aim of NMF is to find  $W$  and  $H$  matrices for a given  $V$  matrix such that

$$V \approx W * H \tag{2.5}$$

In order to use NMF for face recognition [20], images in the training set are placed to columns of  $V$ .  $V$  has a size of  $n \times m$  where  $n$  is the number of pixels, and  $m$  is the number of images in the training set.  $W \in R^{n \times r}$ ,  $H \in R^{r \times m}$  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. Non-negativity constraint enables parts-based representation since only additive combinations are allowed.

In this study, NMF is not implemented; however the work in [19] implemented this method. Since the results of the work in [19] are used for performance comparison, a detailed NMF description can be found in Appendix B.

### 2.4.3 Fischerface

Fisherface is based on Linear Discriminant Analysis (LDA).

#### 2.4.3.1 Linear Discriminant Analysis

LDA is a method used in statistics and machine learning to find the linear combination of features which best separate two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification [36].

LDA is also closely related to principal component analysis (PCA) and factor analysis in that both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and dependent variables (also called criterion variables) must be made.

For multiclass LDA, suppose that each of  $C$  classes has a mean  $\mu_i$  and the same covariance  $\Sigma$ . Then the between class variability may be defined by the sample covariance of the class means:

$$\Sigma_b = \frac{1}{C} \sum_{i=1}^C (\mu_i - \mu)(\mu_i - \mu)^T \quad (2.6)$$

where  $\mu$  is the mean of the class means. The class separation in a direction  $\vec{w}$  in this case will be given by

$$S = \frac{\vec{w}^T \Sigma_b \vec{w}}{\vec{w}^T \Sigma \vec{w}} \quad (2.7)$$

This means that when  $\vec{w}$  is an eigenvector of  $\Sigma_b \Sigma^{-1}$ , the separation will be equal to the corresponding eigenvalue. Since  $\Sigma_b$  is of most rank  $C-1$ , then these non-zero eigenvectors identify a vector subspace containing the variability between features. These vectors are primarily used in feature reduction, as in PCA.

### 2.4.3.2 Fisherface Method

Fisherface method was first proposed in [21]. 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:

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (2.8)$$

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \quad (2.9)$$

where  $\mu$  is the mean of all the samples,  $\mu_i$  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

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 w_2 \dots w_m] \quad (2.10)$$

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

$$S_B w_i = \lambda_i S_W w_i \quad (2.11)$$

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  $n \times n$  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

$(N-c)$  largest eigenvalues are kept and the dimension is reduced to  $N-c$ , as proposed in [22]. 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 method.

#### **2.4.4 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.

$$x=As \tag{2.12}$$

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 [23]. Also some papers have used ICA for 3D face recognition purposes [24], [25], [26].

There are two architectures in ICA. In the first one, pixel values are treated as observations and face images are treated as variables. The second architecture which aims to find statistically independent coefficients is depicted in Figure 2.6.

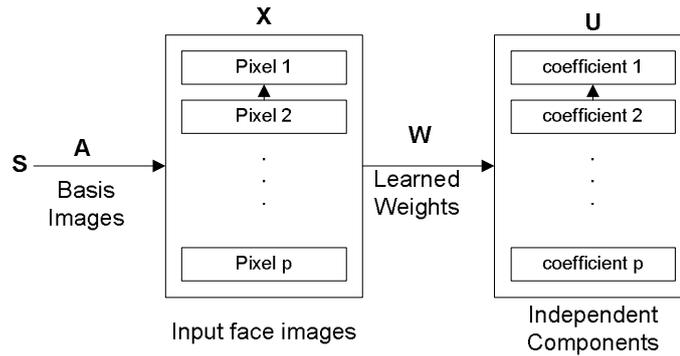


Figure 2.6 Architecture of ICA

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.

#### 2.4.5 Elastic Bunch Graph Matching (EBGM)

EBGM is first introduced in [27]. The main idea behind EBGM is that all human faces share a similar topological structure. In EBGM faces are represented as graphs, with nodes positioned at fiducial points. (eyes, nose...) and edges labeled with 2-D distance vectors. Each node is represented by a set of 40 complex Gabor wavelet coefficients at different scales and orientations (phase, amplitude). They are called "jets". Recognition is based on labeled graphs. A labeled graph is a set of nodes connected by edges, nodes are labeled with jets, edges are labeled with distances.

#### 2.4.6 Kernel Methods

The purpose of kernel methods is to extend PCA and Fisher Linear Discriminant so that higher order correlations between images are taken into account. PCA and FLD represent faces on second order statistics of the image set, and does not address higher order statistical dependencies such as the relationships among three or more pixels. Kernel Eigenface and Kernel Fisherface methods [28],

[29], [30] methods use higher order correlations between images, and hence are able to extract nonlinear features and thus provide better recognition results. A detailed discussion of these kernel methods are given in Appendix C.

#### **2.4.7 Active Appearance Model**

An Active Appearance Model (AAM) is an integrated statistical model which combines a model of shape variation with a model of the appearance variations in a shape-normalized frame. An AAM contains a statistical model of the shape and gray-level appearance of the object of interest which can generalize to almost any valid example. Matching to an image involves finding model parameters which minimize the difference between the image and a synthesized model example projected into the image. It is introduced by Cootes et al. in [31].

#### **2.4.8 Trace Transform**

The Trace transform, a generalization of the Radon transform, is a tool for image processing which can be used for recognizing objects under transformations, e.g. rotation, translation and scaling. To produce the Trace transform one computes a functional along tracing lines of an image. Different Trace transforms can be produced from an image using different trace functional [32], [33]. Srisuk et al. in [34] propose a new texture representation of face image using a robust feature from the trace transform. The so called masked trace transform (MTT) offers “texture” information for face representation which is used to reduce the within-class variance. They first transform the image space to the trace transform space to produce the MTT. Weighted trace transform (WTT) identifies the tracing lines of the MTT which produce similar values irrespective of intra-class variations. They proposed a new distance measure by incorporating the WTT for measuring the dissimilarity between reference and test images.

### **2.4.9 Bayesian Framework**

In [35], Moghaddam et al. proposed a similarity measure for direct image matching based on a Bayesian analysis of image deformations. They modeled two classes of variation in object appearance: intra-object and extra-object. The probability density functions for each class are then estimated from training data and used to compute a similarity measure based on the posteriori probabilities. They further present a novel representation for characterizing image differences using a deformable technique for obtaining pixel-wise correspondences. This representation, which is based on a deformable 3D mesh in XYI-space, is then experimentally compared with two simpler representation: intensity differences and optical flow.

### **2.4.10 Support Vector Machines**

Support Vector Machines (SVMs) are a set of related supervised learning methods used for classification and regression. Viewing input data as two sets of vectors in an  $n$ -dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the margin between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring datapoints of both classes, since in general the larger the margin the better the generalization error of the classifier [36].

In Figure 2.7,  $H_3$  doesn't separate the 2 classes.  $H_1$  does, with a small margin and  $H_2$  with the maximum margin. The aim of SVM is to find the optimum plane  $H_2$ .

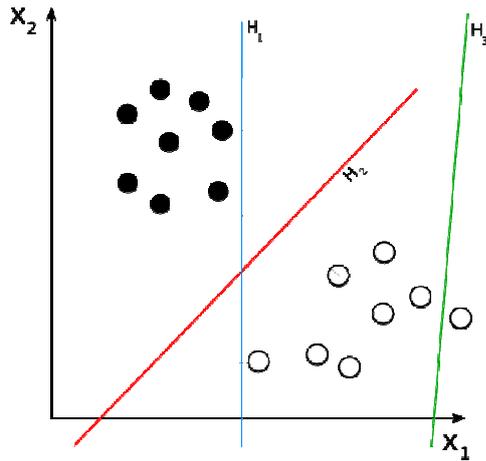


Figure 2.7 Separating hyperplanes

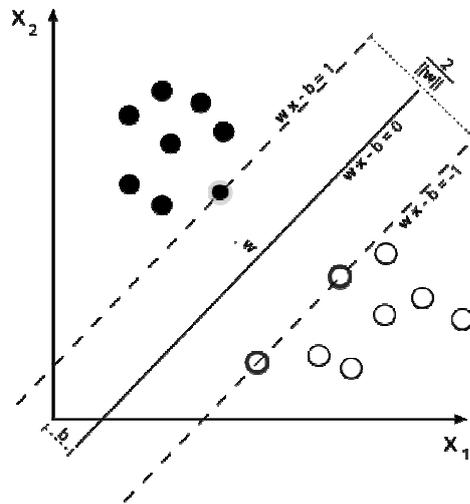


Figure 2.8 Separating hyperplane with maximum margin

SVMs belong to a family of generalized linear classifiers. A special property is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers. Given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a  $p$ -dimensional vector (a list of  $p$  numbers), and we want to know whether we can separate such points with a  $p-1$ -dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. However, we are additionally interested

in finding out if we can achieve maximum separation (margin) between the two classes. By this we mean that we pick the hyperplane so that the distance from the hyperplane to the nearest data point is maximized. That is to say that the nearest distance between a point in one separated hyperplane and a point in the other separated hyperplane is maximized. Now, if such a hyperplane exists, it is clearly of interest and is known as the maximum-margin hyperplane and such a linear classifier is known as a maximum margin classifier.

Given some training data, a set of points of the form

$$\mathcal{D} = \{(x_i, c_i) \mid x_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^n \quad (2.13)$$

where the  $c_i$  is either 1 or  $-1$ , indicating the class to which the point  $x_i$  belongs. Each  $x_i$  is a  $p$ -dimensional real vector. We want to give the maximum-margin hyperplane which divides the points having  $c_i=1$  from those having  $c_i=-1$ . Any hyperplane can be written as the set of points  $\mathbf{x}$  satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0 \quad (2.14)$$

The vector  $\mathbf{w}$  is a normal vector: it is perpendicular to the hyperplane. The parameter  $\frac{b}{\|\mathbf{w}\|}$  determines the offset of the hyperplane from the origin along the normal vector  $\mathbf{w}$ . We want to choose the  $\mathbf{w}$  and  $b$  to maximize the margin, or distance between the parallel hyperplanes that are as far apart as possible while still separating the data. These hyperplanes can be described by the equations

$$\mathbf{w} \cdot \mathbf{x} - b = 1 \text{ and } \mathbf{w} \cdot \mathbf{x} - b = -1. \quad (2.15)$$

Note that if the training data are linearly separable, we can select the two hyperplanes of the margin in a way that there are no points between them and then try to maximize their distance. By using geometry, we find the distance between these two hyperplanes is  $\frac{2}{\|\mathbf{w}\|}$ , so we want to minimize  $\|\mathbf{w}\|$ . As we also have to prevent data points falling into the margin, we add the following constraint: for each  $i$  either

$$\mathbf{w} \cdot \mathbf{x}_i - b \geq 1 \text{ for } x_i \text{ for the first class or} \quad (2.16)$$

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1 \text{ for } x_i \text{ of the second} \quad (2.17)$$

This can be rewritten as:

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 \text{ for all } 1 \leq i \leq n. \quad (2.18)$$

We can put this together to get the optimization problem:

Choose  $\mathbf{w}$ ,  $b$  to minimize  $\|\mathbf{w}\|$ ,

Subject to  $c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$  for all  $1 \leq i \leq n$ .

The original optimal hyperplane algorithm proposed by Vladimir Vapnik in 1963 was a linear classifier. However, in 1992, Bernhard Boser, Isabelle Guyon and Vapnik suggested a way to create non-linear classifiers by applying the kernel trick to maximum-margin hyperplanes [37, 38, 39, 40, 41]. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space. The transformation may be non-linear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space it may be non-linear in the original input space. Some common kernels are:

Polynomial (homogeneous):  $k(x, x') = (x \cdot x')^d$

Polynomial (inhomogeneous):  $k(x, x') = (x \cdot x' + 1)^d$

Radial basis function:  $k(x, x') = \exp(-\gamma\|x - x'\|^2)$  for  $\gamma > 0$

Gaussian radial basis function:  $k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$

Sigmoid:  $k(x, x') = \tanh(\kappa x \cdot x' + c)$  for some  $\kappa > 0$  and  $c < 0$ .

SVM models are closely related to neural networks. In fact, a SVM model using a sigmoid kernel function is equivalent to a two-layer perceptron neural network. Using a kernel function, SVM's are an alternative training method for polynomial, radial basis function and multi-layer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training [42].

In face recognition, PCA is first used to extract features of face images and then discrimination functions between each pair of images are learned by SVMs [43, 44, 45].

## 2.5 3D Methods for Face Recognition

The activity to exploit 3D data to improve the accuracy and robustness of face recognition system is still weakly addressed. Only a few works on the use of 3D data have been reported. These methods can be categorized into four groups:

- Methods based on curvature analysis,
- Methods by shape representation,
- Methods by model fitting and image synthesis, and
- Other methods.

Many of the early studies concentrate on curvature analysis [46]. [47], [48], [49], [50]. The work by Gordon et al. [47, 48] presents a template-based recognition system using descriptors derived from range image. The sensed surface regions are classified as convex, concave and saddle by calculating the minimum and maximum principal curvature, and then the locations of nose, eyes, mouth and other features are determined, which are used for depth template comparison.

Lee *et al.*[46] propose a method to detect corresponding regions in two range images by graph matching based on Extended Gaussian Image (EGI).

An approach to label the components of human faces is proposed by Yacoob *et al.*[49]. Its preprocessing stage employs a multistage diffusion process to identify convexity and concavity points. These points are grouped into components. Qualitative reasoning about possible interpretations of the components is performed, followed by consistency of hypothesized interpretations.

Tanaka *et al.* [50] also use the Extended Gaussian Image. For each face, two EGIs are constructed from maximum principal curvature and minimum principal curvature. The EGI similarity is measured by Fisher's spherical correlation. However, because they are involved in computing curvature, all these techniques

require high resolution of the range data; otherwise the computation of curvature will be inaccurate and unreliable.

Extended Gaussian Image (EGI) is used to model 3D face shape and curvature [50, 51, 52]. EGI computes the distribution of surface normal vectors of the face on a unit sphere. EGI normal distribution is usually called an EGI image. A point on the EGI image is presented by its position on the unit sphere which represents surface orientation and the height of the point represents the surface area of that specific orientation.

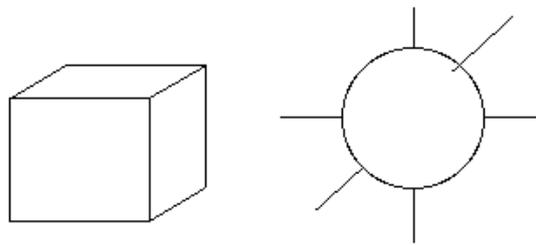


Figure 2.9 Extended Gaussian Image of a cube

The main advantage of EGI is that EGI transformation eliminates position information, i.e. the transformation is translation invariant. However, EGI is variant under scaling and rotation transformations. And also, EGI transformation is not localized, i.e. a point on the EGI image does not involve information about the exact point on the face surface.

References [46-50] attempt to use a shape representation to analyze the 3D facial data. Chua *et al.* in [53] describes a technique for 3D face recognition based on Point Signature, a representation for free-form surfaces, which is also highly dependent on the quality of facial range data. In the method, the rigid parts of the face of one person are simply extracted to deal with different facial expressions. Their subsequent work in [54] combines Point Signature on 3D range data and Gabor filter response on 2D grayscale image for facial feature detection and

recognition. Pan et al. [55] presents a novel signature *Curgram* for pose-invariant detection of facial feature from range data.

The third kind of approach is to use model fitting or image synthesis to cope with the influence of illumination and pose. For example, Blanz *et al.* [56, 57] utilize a 3D morphable model to fit the input facial image to tackle variation of pose and illumination. For this approach, the shape and texture fitting procedure is hugely time-consuming.

In Ref. [58], Lee *et al.* employ an edge model and a color region model to analyze face image, and a wireframe model to synthesize the face image in virtual view for recognition. And Zhao *et al.* [59] present a method to synthesize the virtual image with shape from shading (SFS)-based 3D shape recovery. For this kind of approach, the input is a 2D face image but not 3D data. Because an image is essentially the projection from 3D space to 2D space, due to the nature of 2D image, there difficulties in accurate recognition across pose and illumination.

Other 3D face recognition approaches include those mentioned in Refs. [60, 61]. Beumier *et al.* in [60] propose two 3D comparison methods based on surface and profiles matching respectively. In Reference [61], authors do the task of face recognition via feature vector that is generated from depth information of the area in some contour line. Recently Bronstein *et al.* propose a novel 3D face recognition method in [9]. It converts facial shape and texture to the special images by a bending-invariant mapping scheme, and then perform eigenface decomposition on the special images to do the recognition task.

A fairly new work published by Iordanis et al. in [62] proposes a geodesic polar parameterization of the face surface. With this parameterization, the intrinsic surface attributes of the face shape do not change under isometric deformations, and therefore, the proposed representation is appropriate for expression-invariant

3D face recognition. They even considered the special case of an open mouth that violates the isometry assumption and further propose a modified geodesic representation that also leads to invariant representation. Based on this representation they reduce the problem of 3D face recognition to recognition of expression compensated 3D images. While preprocessing the surface face, they used Iterative Closest Point (ICP) algorithm to align the surfaces. An overview of ICP is given in Appendix A.

In reference [63], Xiaoguang et al. propose a multi-view face recognition method based on 3D face mesh which are generated from 2.5D range scans. They automatically detects feature points on the 2.5D face scans, and transform 2.5D range image coarsely with the full 3D model. Afterwards, they finely register the image using ICP. They used commercial software to generate a Virtual Reality Markup Language (VRML) 3D model of the 2.5D scan. They used root-mean-square distance minimized by the ICP algorithm for primary matching score of face scans.

In an M.Sc. thesis study completed in METU Computer Vision and Intelligent Systems Research Laboratory, the effect of registration on the 3D face recognition algorithms is studied in which ICP (see Appendix A) is used for registration. PCA, ICA, NMF and Fisherface are used for dimension reduction and classification. The experimental results obtained by applying the mentioned methods on the 3D face images in FRGC database. The experiments conveyed in this thesis are applied in the same manner as in [19] so that direct comparison of the results becomes possible.

## **2.6 Distance Metrics**

In pattern classification problems, the first step is always the feature extraction step. This follows dimension reduction step, where the number of features are reduced for further classification. Most of the classification methods require a

similarity or distance measure between reduced feature vectors. Several distance metrics are defined in the literature. Some of them can be listed as follows;

- Euclidean distance
- Manhattan distance (city-block)
- Minkowski metric
- Tanimoto metric
- Tangent distance
- Mahalanobis distance

A metric  $D$  must have four properties [64]: For all vectors,  $\mathbf{a}$ ,  $\mathbf{b}$  and  $\mathbf{c}$ , these properties are as follows:

- Nonnegativity:  $D(\mathbf{a}, \mathbf{b}) \geq 0$
- Reflexivity:  $D(\mathbf{a}, \mathbf{b}) = 0$  if and only  $\mathbf{a}=\mathbf{b}$
- Symmetry:  $D(\mathbf{a}, \mathbf{b}) = D(\mathbf{b}, \mathbf{a})$
- Triangle inequality:  $D(\mathbf{a}, \mathbf{b}) + D(\mathbf{b}, \mathbf{c}) \geq D(\mathbf{a}, \mathbf{c})$

In this study, Euclidean distance and Mahalanobis distances will be used.

### 2.6.1 Euclidean Distance

Euclidean distance in  $d$  dimensions can be defined as follows:

$$D(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2} \quad (2.19)$$

Although it is simple to compute the Euclidean distance between two vectors, the results may or may not be meaningful. For example, if the input vector space is transformed by multiplying each coordinate by an arbitrary constant, the Euclidean distance relationships in the transformed space can be very different from the original distance relationships. Such scale changes can have a major impact on nearest neighbor classifiers [64].

## 2.6.2 Mahalanobis Distance

Formally, the Mahalanobis distance from a group of values with mean  $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_p)^T$  and covariance matrix  $\Sigma$  for a multivariate vector  $\boldsymbol{x} = (x_1, x_2, \dots, x_p)^T$  is defined as:  $D(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})}$ . Mahalanobis distance can also be defined as dissimilarity measure between two vectors  $\boldsymbol{x}$  and  $\boldsymbol{y}$ :

$$D(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{y})} \quad (2.20)$$

If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. If the covariance matrix is diagonal, then the resulting distance measure is called the normalized Euclidean distance:

$$D(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^p \frac{(x_i - y_i)^2}{\sigma_i^2}} \quad (2.21)$$

where  $\sigma_i^2$  is the standard deviation of the  $x_i$  over the sample set.

Related to the Mahalanobis distance, the *whitening* transform ensures that the feature vector is transformed such that its components are uncorrelated and their variances equal to unity. In other words, the covariance matrix of the feature vectors equal to identity matrix, i.e.  $\Sigma = I$ . The whitening transform can be performed using eigenvalue decomposition (EVD) of the covariance matrix  $\Sigma = EDE^T$ , where E is the orthogonal matrix of eigenvectors of  $\Sigma$  and D is the diagonal matrix of its eigenvalues,  $D = \text{diag}(d_1, \dots, d_p)$ . Whitening can now be done by

$$\tilde{\boldsymbol{x}} = ED^{-1/2}E^T\boldsymbol{x} \quad (2.22)$$

where the matrix  $D^{-1/2}$  is computed by a simple component-wise operation as  $D^{-1/2} = \text{diag}(d_1^{-\frac{1}{2}}, \dots, d_p^{-\frac{1}{2}})$ . It is easy to check that now  $\Sigma = I$ .

## CHAPTER 3

### SPHERICAL HARMONICS EXPANSION

In this chapter, the theory of spherical harmonics and spherical harmonic transform is given. Application of spherical harmonics in computer graphics, shape classification and face recognition are also discussed.

#### 3.1 Spherical Harmonic Functions

In mathematics, the spherical harmonics are the angular portion of an orthogonal set of solutions to Laplace's equation represented in a system of spherical coordinates. Spherical harmonics are important in many theoretical and practical applications, particularly in the computation of atomic electron configurations, the representation of the gravitational field, geoid and magnetic field of planetary bodies, as well as characterization of the cosmic microwave background radiation. In 3D computer graphics, spherical harmonics plays a special role in a wide variety of topics including indirect lighting (ambient occlusion, global illumination, precomputed radiance transfer etc) and in recognition of 3D shapes [36].

Laplace's equation in spherical coordinates is:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \varphi^2} = 0 \quad (3.1)$$

For  $f(r, \theta, \varphi) = R(r)\Theta(\theta)\Phi(\varphi)$ , the angular portion of Laplace's equation satisfies

$$\frac{\Phi(\varphi)}{\sin \theta} \frac{d}{d\theta} \left( \sin \theta \frac{d\Theta}{d\theta} \right) + \frac{\Theta(\theta)}{\sin^2 \theta} \frac{d^2 \Phi}{d\varphi^2} + l(l+1)\Theta(\theta)\Phi(\varphi) = 0 \quad (3.2)$$

Using the technique of separation of variables, two differential equations result:

$$\frac{1}{\Phi(\varphi)} \frac{d^2\Phi(\varphi)}{d\varphi^2} = -m^2 \quad (3.3)$$

$$l(l+1)\sin^2(\theta) + \frac{\sin(\theta)}{\theta(\theta)} \frac{d}{d\theta} \left[ \sin(\theta) \frac{d\theta}{d\theta} \right] = m^2 \quad (3.4)$$

for some  $m$  and  $l$ . Hence, the angular solutions can be shown to be products of trigonometric functions and associated Legendre functions:

$$Y_l^m(\theta, \varphi) = N e^{im\varphi} P_l^m(\cos\theta), \quad (3.5)$$

where  $Y_l^m(\theta, \varphi)$  is called a spherical harmonic function of degree  $l$  and order  $m$ ,  $P_l^m$  is associated Legendre function,  $N$  is a normalization constant, and  $\theta$  and  $\varphi$  represent colatitude and longitude, respectively. In particular, the colatitude  $\theta$ , or polar angle, ranges from  $0 \leq \theta \leq \pi$  and the longitude  $\varphi$ , or azimuth, ranges from  $0 \leq \varphi \leq 2\pi$ . Thus,  $\theta$  is 0 at the north pole,  $\pi/2$  at the Equator, and  $\pi$  at the south pole.

When Laplace's equation is solved on the surface of the sphere, the periodic boundary conditions in  $\varphi$ , as well as regularity conditions at both the north and south poles, ensure that the degree  $l$  and order  $m$  are integers that satisfy  $l \geq 0$  and  $|m| \leq l$ . In contrast, if the function  $f$  were only to have been defined for  $\theta \leq \theta_0$ , then the resulting spherical cap harmonics would have been defined for integer order, but non-integer degree.

The general solution to Laplace's equation is a linear combination of the spherical harmonic functions multiplied by the solutions of  $R(r)$ :

$$f(r, \theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l r^{-1-l} f_l^m Y_l^m(\theta, \varphi) + \sum_{l=0}^{\infty} \sum_{m=-l}^l r^l f_l^{m'} Y_l^m(\theta, \varphi) \quad (3.6)$$

where  $f_l^m$  and  $f_l^{m'}$  are constants. The terms in the first summation approach zero as  $r$  goes to infinity, whereas the terms in the second summation approach zero at the origin.

Several different normalizations are in common use for the spherical harmonic functions. In physics and seismology, these functions are generally defined as

$$Y_l^m(\theta, \varphi) = \sqrt{\frac{(2l+1)}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) e^{im\varphi} \quad (3.7)$$

which are orthonormal

$$\int_{\theta=0}^{\pi} \int_{\varphi=0}^{2\pi} Y_l^m Y_{l'}^{m'*} d\Omega = \delta_{ll'} \delta_{mm'} \quad (3.8)$$

where  $\delta_{ll'}$  and  $\delta_{mm'}$  are Kronecker delta and  $d\Omega = \sin\theta d\varphi d\theta$ .

### 3.2 Spherical Harmonics Expansion

Basis functions are small pieces of signal that can be scaled and combined to produce an approximation to an original function, and the process of working out how much of each basis function to sum is called *projection*. To approximate a function using basis functions we must work out a scalar value that represents how much the original function  $f(x)$  is like the each basis function  $B_i(x)$ . We do this by integrating the product  $f(x)B_i(x)$  over the full domain of  $f$ .

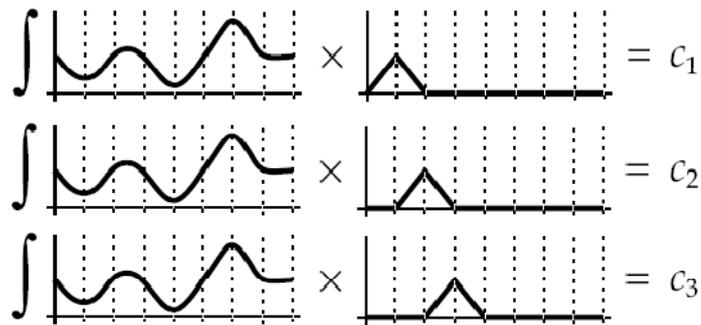


Figure 3.1 Orthogonal basis expansion

Using this projection process over all our basis functions returns a vector of approximation coefficients. If we scale the corresponding basis function by the coefficients

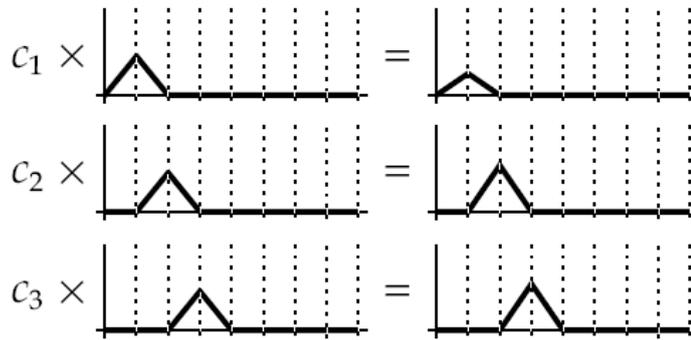


Figure 3.2 Orthogonal basis expansion

and sum the results we obtain our approximated function.

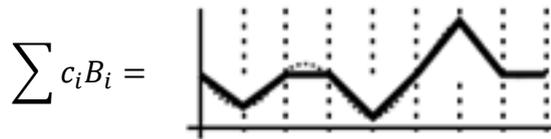


Figure 3.3 Orthogonal basis expansion

Polynomials are mostly used as basis functions in expansions. Legendre polynomials, specifically the *Associated Legendre Polynomials* are used as the basis functions in spherical harmonics expansion. Traditionally represented by the symbol  $P$ , the associated Legendre polynomials have two arguments  $l$  and  $m$ , are defined over the range  $[-1,1]$  and return real numbers (as opposed to the ordinary Legendre Polynomials which return complex values – be careful not to confuse the two).

The two arguments  $l$  and  $m$  break the family of polynomials into *bands* of functions where the argument  $l$  is the *band index* and takes any positive integer value starting from 0, and the argument  $m$  takes any integer value in the range  $[0,l]$ . Inside a band the polynomials are orthogonal w.r.t. a constant term and between bands they are orthogonal with a different constant. We can diagram

this as a triangular grid of functions per band, giving us a total of  $n(n+1)$  coefficients for an  $n$  band approximation:

$$\begin{aligned}
 &P_0^0(x) \\
 &P_1^0(x), P_1^1(x) \\
 &P_2^0(x), P_2^1(x), P_2^2(x) \\
 &P_3^0(x), P_3^1(x), P_3^2(x), P_3^3(x) \\
 &\dots
 \end{aligned}$$

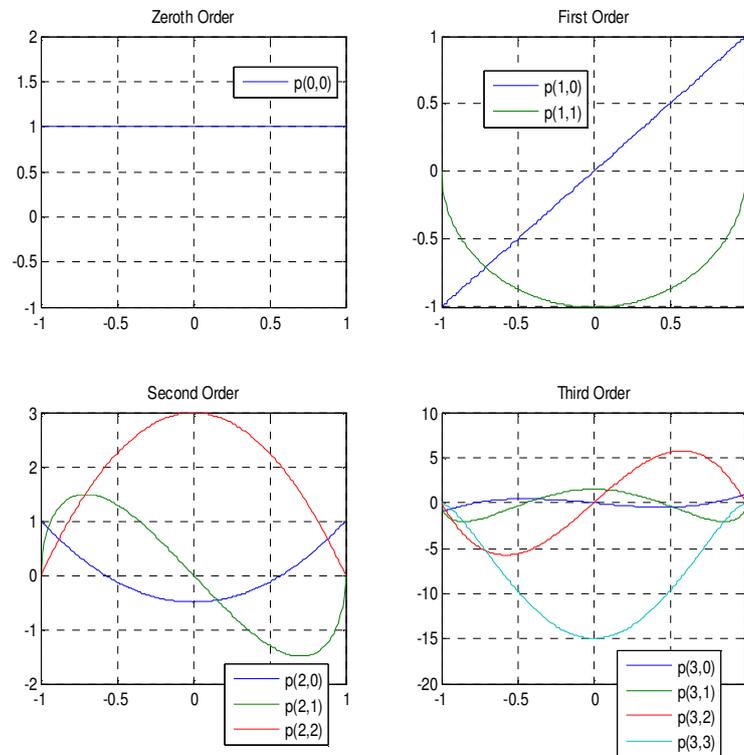


Figure 3.4 The first 10 associated Legendre polynomials.

The associated Legendre polynomials are at the heart of the *Spherical Harmonics*, a mathematical system analogous to the Fourier transform but defined across the surface of a sphere. The SH functions in general are defined on imaginary numbers but in this study only real functions are approximated

over the sphere, so only the *Real Spherical Harmonics* will be used. In this study, when a SH function is referred, only the Real Spherical Harmonic functions are considered.

The spherical harmonics form a complete set of orthonormal functions and thus form a vector space analogous to unit basis vectors. On the unit sphere, any square-integrable function can thus be expanded as a linear combination of these functions. The process for projecting a spherical function into SH coefficients is very simple. To calculate a single coefficient for a specific band you just integrate the product of your function  $f$  and the SH function  $y$ , in effect working out how much your function is like the basis function:

$$f(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l f_l^m Y_l^m(\theta, \varphi) \quad (3.9)$$

This expansion is exact as long as  $l$  goes to infinity. Truncation errors will arise when limiting the sum over  $l$  to a finite bandwidth  $L$ . The expansion coefficients can be obtained by multiplying the Equation 3.9 by the complex conjugate of a spherical harmonic, integrating over the solid angle  $\Omega$ , and utilizing the above orthogonality relationship. For the case of orthonormal harmonics, this gives:

$$f_l^m = \int_{\Omega} f(\theta, \varphi) Y_l^{m*}(\theta, \varphi) d\Omega = \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \sin\theta f(\theta, \varphi) Y_l^{m*}(\theta, \varphi) \quad (3.10)$$

An alternative set of spherical harmonics for real functions may be obtained by taking the set:

$$Y_l^m = \begin{cases} Y_l^0 & \text{if } m = 0 \\ \frac{1}{\sqrt{2}}(Y_l^m + (-1)^m Y_l^{-m}) = \sqrt{2}N_{(l,m)}P_l^m(\cos\theta)\cos m\varphi & \text{if } m > 0 \\ \frac{1}{i\sqrt{2}}(Y_l^{-m} - (-1)^m Y_l^m) = \sqrt{2}N_{(l,m)}P_l^{-m}(\cos\theta)\sin m\varphi & \text{if } m < 0 \end{cases} \quad (3.11)$$

where  $N(l,m)$  denotes the normalization constant as a function of  $l$  and  $m$ . These functions have the same normalization properties as the complex ones above. In this notation, a real square-integrable function can be expressed as an infinite sum of real spherical harmonics as:

$$f(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l f_l^m Y_l^m(\theta, \varphi) \quad (3.12)$$

Using the above spherical harmonics notation, analytic expressions for the first few orthonormal spherical harmonics can be written as follows;

$$Y_0^0(\theta, \varphi) = \frac{1}{2} \sqrt{\frac{1}{\pi}}$$

$$Y_1^{-1}(\theta, \varphi) = \frac{1}{2} \sqrt{\frac{3}{2\pi}} \sin\theta e^{-i\varphi}$$

$$Y_1^0(\theta, \varphi) = \frac{1}{2} \sqrt{\frac{3}{\pi}} \cos\theta$$

$$Y_1^1(\theta, \varphi) = -\frac{1}{2} \sqrt{\frac{3}{2\pi}} \sin\theta e^{i\varphi}$$

$$Y_2^{-2}(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{15}{2\pi}} \sin^2\theta e^{-2i\varphi}$$

$$Y_2^{-1}(\theta, \varphi) = \frac{1}{2} \sqrt{\frac{15}{2\pi}} \sin\theta \cos\theta e^{-i\varphi}$$

$$Y_2^0(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{5}{\pi}} (3\cos^2\theta - 1)$$

$$Y_2^1(\theta, \varphi) = -\frac{1}{2} \sqrt{\frac{15}{2\pi}} \sin\theta \cos\theta e^{i\varphi}$$

$$Y_2^2(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{15}{2\pi}} \sin^2\theta e^{2i\varphi}$$

The above spherical harmonics and some other one will be depicted on the following figures.

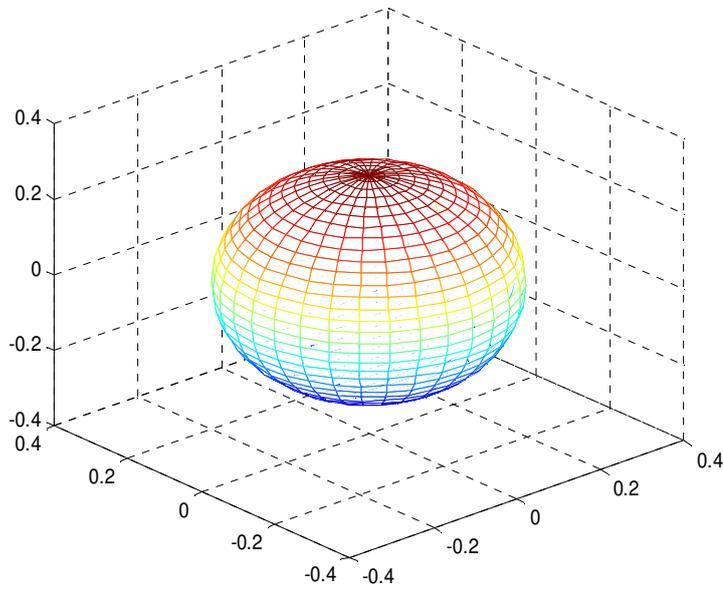


Figure 3.5  $|Y_0^0(\theta, \varphi)|$

Notice that,  $Y_0^0$  is just a sphere.

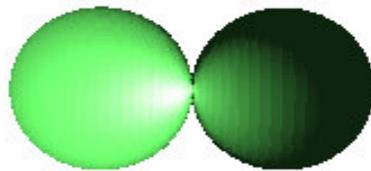


Figure 3.6  $|Y_1^0(\theta, \varphi)|$

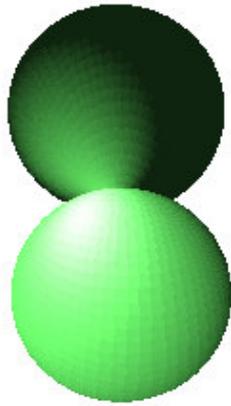


Figure 3.7  $|Y_1^1(\theta, \varphi)|$

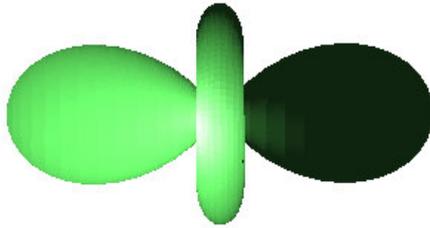


Figure 3.8  $|Y_2^0(\theta, \varphi)|$

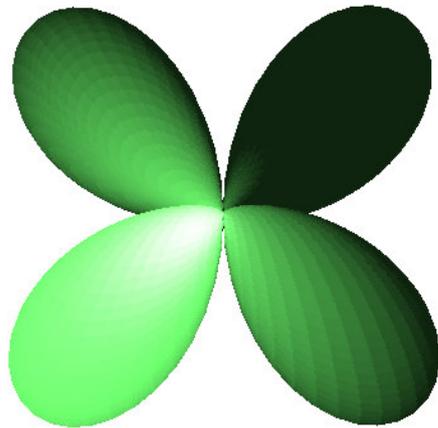


Figure 3.9  $|Y_2^1(\theta, \varphi)|$

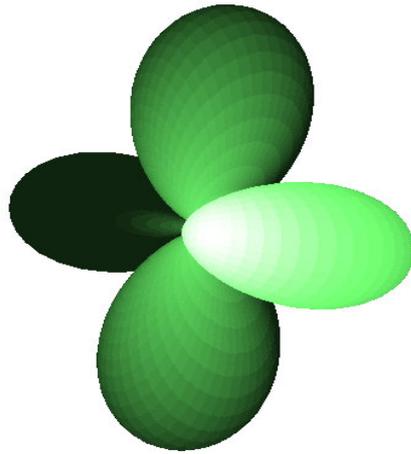


Figure 3.10  $|Y_2^2(\theta, \varphi)|$

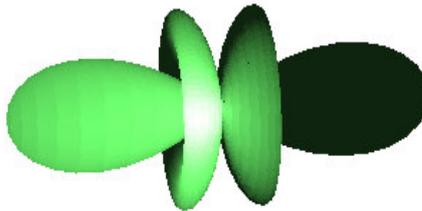


Figure 3.11  $|Y_3^0(\theta, \varphi)|$

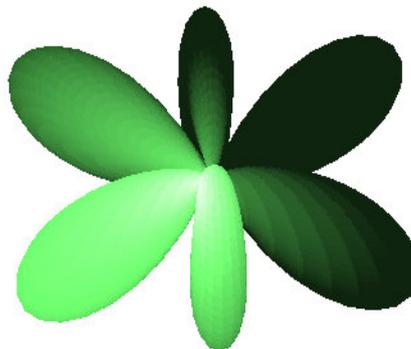


Figure 3.12  $|Y_3^1(\theta, \varphi)|$

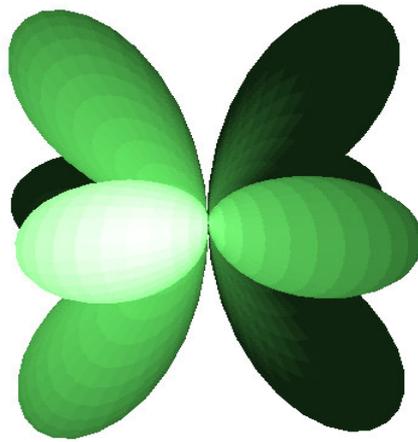


Figure 3.13  $|Y_3^2(\theta, \varphi)|$

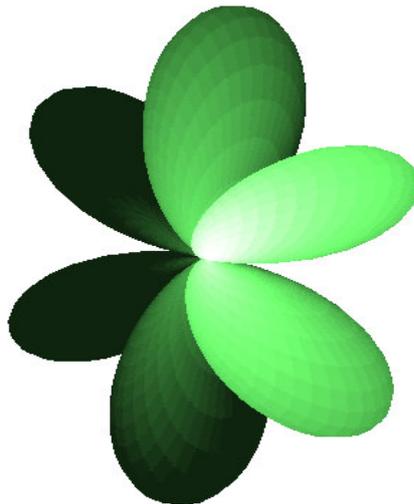


Figure 3.14  $|Y_3^3(\theta, \varphi)|$

A more explanatory image of the first 5 spherical harmonics band is depicted in Figure 3.15. Notice the phase difference between spherical harmonics at the same degree with different order signs.

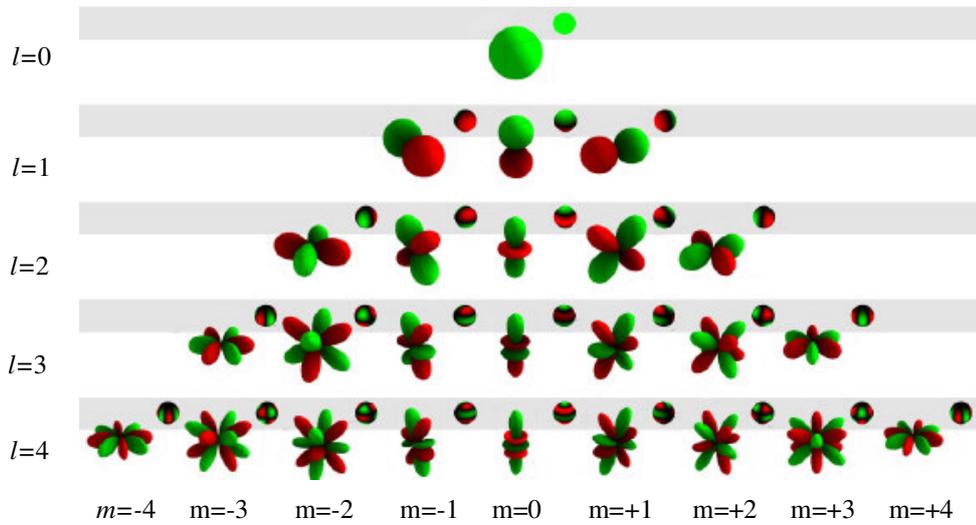


Figure 3.15 The first 5 SH bands

To reconstruct the approximated function ( $\tilde{f}$ ), we just take the reverse process and sum scaled copies of the corresponding SH functions. It should be noted that the true function  $f$  could be reconstructed if the infinite series of all SH coefficients are used in the summation, therefore every reconstruction in practical use will be an approximation to the true function, technically known as a *band-limited* approximation where band-limiting is just the process of breaking a signal into its component frequencies and removing frequencies higher than some threshold.

$$f(\theta, \varphi) \approx \sum_{l=0}^N \sum_{m=-l}^l f_l^m Y_l^m(\theta, \varphi) \quad (3.13)$$

In Figure 3.16., projection of some spherical functions with different approximation levels are shown. Notice that, a band-limited approximation is similar to low-pass filtering of signals using Fourier transform.

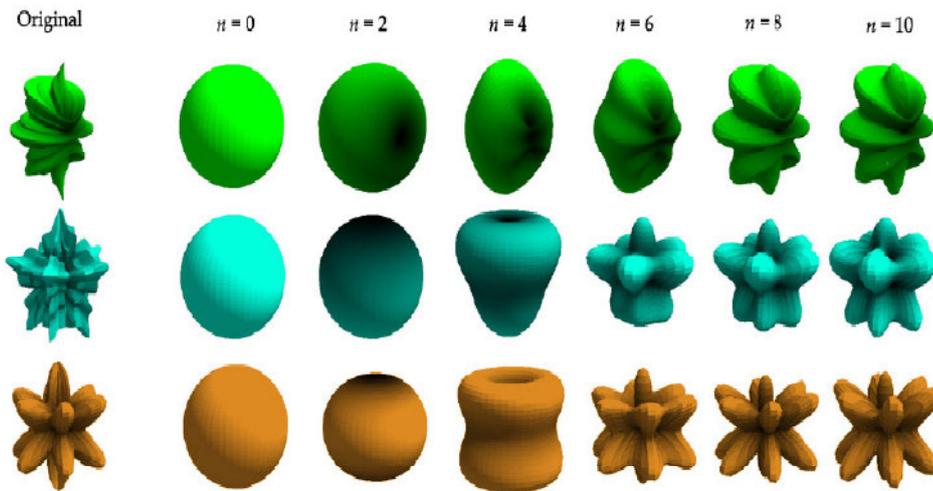


Figure 3.16 SH expansion with increasing orders of approximation ([8]).

It should be noted that SH projection of a spherical function is rotation invariant. In order to make SH projection rotation-invariant, the norm of the coefficient at the  $m$ th order considered. The Figure 3.17 illustrates this property of SH projection.

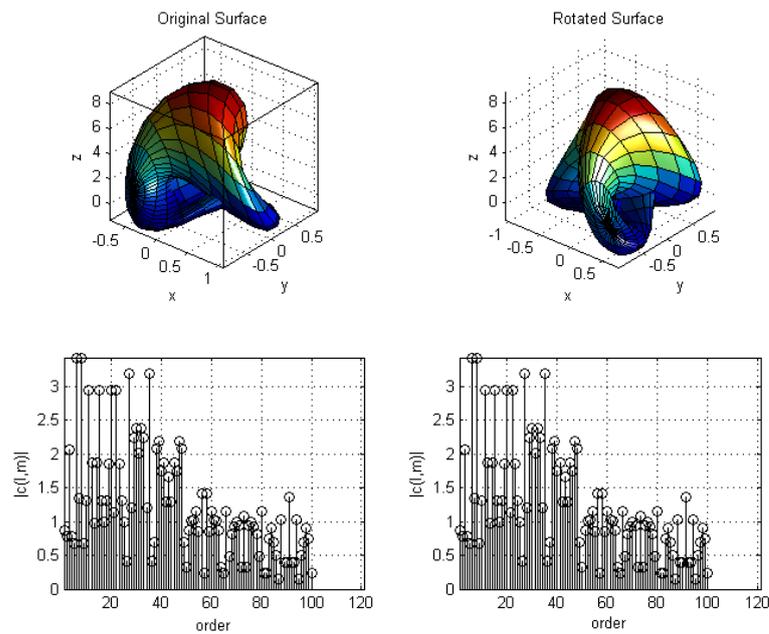


Figure 3.17 Rotation Invariance of SH transform

Notice that, the norm of the coefficient are the same for the two shapes above.

### **3.3 Applications of Spherical Harmonics**

Spherical harmonics are important in many theoretical and practical applications, particularly in the computation of atomic electron configurations, the representation of the gravitational field, geoid, and magnetic field of planetary bodies, characterization of the cosmic microwave background radiation. In 3D computer graphics, spherical harmonics plays a special role in a wide variety of topics including indirect lighting (ambient occlusion, global illumination, precomputed radiance transfer etc) and in recognition of 3D shapes.

In this study, we will briefly give some detailed about usage of spherical harmonics in computer graphics, and 3D shape recognition which are closely related to the face recognition problem.

#### **3.3.1 Spherical Harmonic Lighting**

Spherical Harmonic lighting (SH lighting) is a technique for calculating the lighting on 3D models from area light sources that allows to capture, relight and display global illumination style images in real time. It was introduced in a paper at ACM-Siggraph 2002 by Sloan, Kautz and Snyder as a technique for ultra realistic lighting of models [65]. The details of the SH lighting is given in Appendix D.

#### **3.3.2 Modeling Face Illumination Variation with Spherical Harmonics**

Illumination can have a significant impact on the appearance of surfaces, as the patterns of shading, specularities and shadows change. For instance, some images of a face under different lighting conditions are shown in Figure 3.18.



Figure 3.18 Images of a face, lit from a number of different directions.

There are many research papers published by the computer graphics community about illumination modeling of an object under different lighting conditions. Ravi et al. and Basri et al. ([66], [67]) adapted this research to the face recognition problem, in order to overcome the illumination problem in face recognition. Basri et al. [67] propose a method to identify a subject from a test image that is acquired under different pose and illumination condition from only one training sample of this subject in the gallery database. This sounds a challenging scenario in face recognition. For example, the test image could be semifrontal and illuminated by multiple lighting sources while the corresponding training image is frontal under a single lighting source. Under the assumption of Lambertian reflectance, they used the spherical harmonics representation in modeling illumination variations for a fixed pose. In their work, we extend the spherical harmonics representation to encode pose information. More specifically, they utilized the fact that 2D harmonic basis images at different poses are related by close-form linear transformations, and give a more convenient transformation matrix to be directly used for basis images. So they managed to synthesize a different view of a face under arbitrary lighting conditions by changing the coefficients of the spherical harmonics representation. They call their method “pose-encoded spherical harmonics”.

The first paper in literature about this subject is published by Zhang et al. ([68]) under the title “face recognition under variable lighting using harmonic image exemplars”. They demonstrated that the set of images of a convex Lambertian

object obtained under a wide variety of lighting conditions can be approximated accurately by a low-dimensional linear subspace.

### 3.3.3 Spherical Harmonics as 3D Shape Descriptor

Kazhdan et al. in [8] propose a model matching algorithm to find the computational representation of a shape for which an index can be built and a geometric matching can be performed efficiently. They define the following properties for a shape description. It should be **(1)** computationally efficient, **(2)** efficient for storage, **(3)** easy to query and retrieve, **(4)** invariant under similarity transforms, and **(5)** independent of 3D object representation, tessellation, genus or topology.

Their harmonic shape representation can be outlined as follows;

(1) Given a model, rasterize its polygons into a 64x64x64 voxel grid, (assigning a voxel a value of 1 if it was within one voxel of a point on the boundary, and a value of 0 otherwise). The model is aligned so that its center of mass is at the center of the grid, and so that its bounding sphere has radius 32.

(2) Treating it as a function defined in three dimensional-space, decompose the voxel grid into 32 spherical functions by restricting the voxel grid to spheres with radii 1 through 32.

(3) Decompose each of these functions as a sum of its first 16 harmonic components, analogous to a Fourier decomposition into different frequencies.

(4) Using the fact that rotations do not change the norm of the harmonic components, define the signature of each spherical function as a list of these 16 norms.

(5) Combine these different signatures to obtain a 32x16 signature for the 3D model. The resultant rotation invariant signature is a two-dimensional grid where the value of the  $(i; j)$ -th index is equal to the norm of the  $j$ -th order component of the spherical function on the sphere of radius  $i$ .

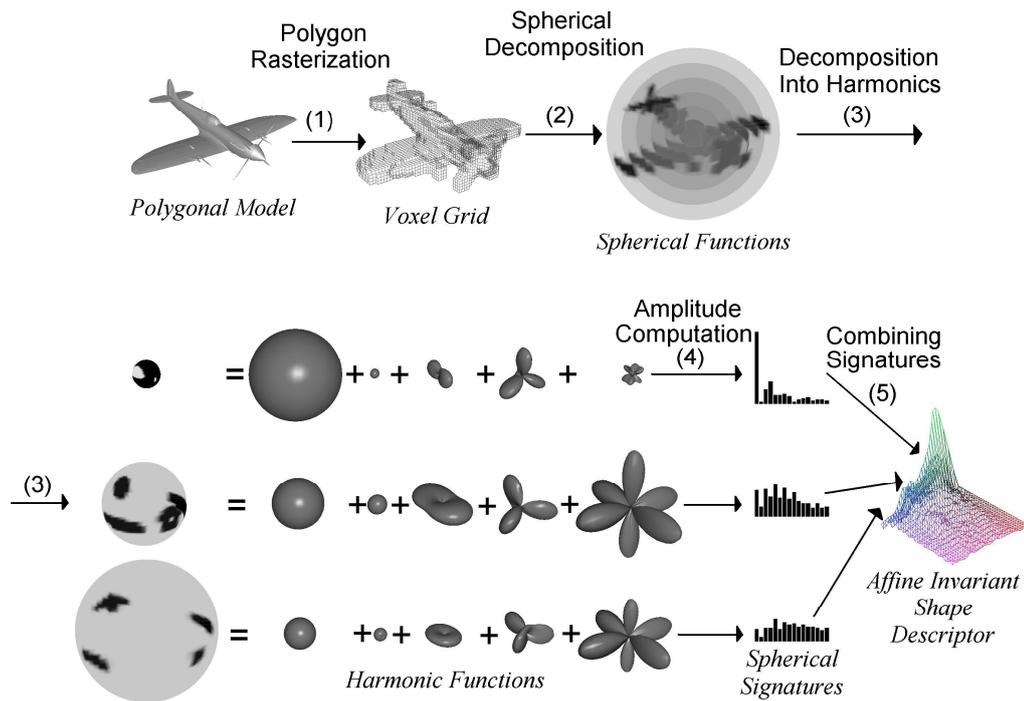


Figure 3.19 Spherical harmonic shape descriptor (from [8])

To compare two harmonic representations, simply compute the Euclidean distance between them. Thus, finding the  $K$  closest models to a query is equivalent to solving the nearest-neighbor problem in a  $32 \times 16$  dimensional space. They compared their matching method's performance to other five existing methods, and they report very promising results (Figure 3.20).

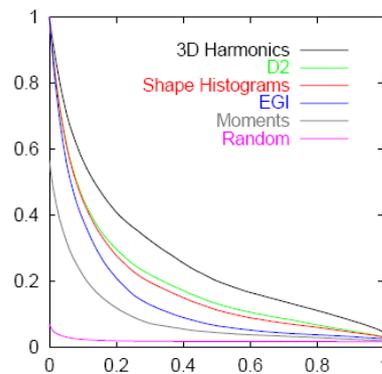


Figure 3.20 Precision vs recall performance for 3d SH shape descriptor [8]

### 3.3.4 Face Recognition via Spherical Embedding

Bronstein et al. [9] reported that their novel 3D face recognition method can overcome expression variability and even can distinguish between images of twins. They modelled the facial expressions as isometries, that is, geodesic distances on the facial surface using elliptic geometry. They used the empirical fact that geodesic distances on the facial surface are less sensitive to facial expressions compared to Euclidean ones. Their system is able to compute expression invariant signatures based on isometry-invariant representation of the facial surface. They embed the geometric structure of the facial surface into a spherical space in order to construct isometry invariant representations of the facial image.

They modeled a face as a complete compact smooth two-dimensional Riemannian manifold (surface)  $(S, g)$  with a Riemannian metric  $g$ , endowed with some property field, the scalar field  $\rho: S \rightarrow [0,1]$  representing the gray-scale albedo of the face. They are using 2.5D range image and the intensity image for sampling  $S$  and the corresponding reflectance value  $r$  at the points.

They define two factors which affect the performance of face recognition algorithms, which are *external* factors influencing the reflectance image such as illumination conditions, head orientation, and *internal* factors such as facial expressions.

Under the assumption of the isometric model, they obtained an expression-invariant (isometry-invariant) representation of the face, which is isolated of its *extrinsic* geometry, that is, the way the surface  $S$  is immersed into the ambient three-dimensional Euclidean space. Their isometry representation is, keeping only the intrinsic geometry, that is, the geometry on the surface itself.

An obvious isometric invariant of the surface is the set of all the geodesic distances between its points. In order to construct the invariant of the face surface, they propose a procedure called *isometric embedding*, which allows to isolate the surface data from its extrinsic geometry. Isometric embedding is a mapping between two finite metric spaces,

$$\varphi: (\{\xi_1, \xi_2, \dots, \xi_N\} \subset S, D) \rightarrow (\{\xi'_1, \xi'_2, \dots, \xi'_N\} \subset S^m, D') \quad (3.14)$$

such that for all  $i, j=1, \dots, N$ ,  $d'_{ij}=d_{ij}$ . The matrices  $D=(d_{ij})$  and  $D'=(d'_{ij})$  denote the mutual distances between the points in the original and the embedding spaces, respectively. The image of  $\{\xi_1, \xi_2, \dots, \xi_N\}$  under  $\varphi$  is called the canonical form of  $(S,g)$ . Notice that the canonical form is an approximate representation of the discrete face surface. The canonical form is uniquely defined up to any transformation in the embedding space that does not alter the distances (like translations, rotations and reflections in an Euclidean embedding space).

They also propose to embed the facial image into a two-dimensional sphere  $S^2$  rather than a plane. And also they use the spherical harmonic transform in order to obtain a truly invariant signature of the face.

They used A data-set of 104 faces was used for the experiments. The set consisted of four subjects (two of which are identical twins) with extreme facial expressions. Each subject was acquired with five instances of neutral expression and three instances of smile, anger, surprise, inflated cheeks, deflated cheeks, and neutral expression with eyeglasses. The faces were preprocessed, and 500x500 matrices  $D$  of geodesic distances between points on the facial surfaces were computed.

They also tested the influence of the embedding sphere radius on the embedding error was tested. They report that embedding sphere radius yielding the

minimum embedding error ranges from 90 to 100mm. Unfortunately , no information is given in [9] on how to the embedding is done.

### 3.4 Fast Spherical Harmonics Transform using FFT

Spherical harmonics transform of a face can take long time on a fast desktop computer. In order to speed up the execution, Fast Fourier Transform (FFT) can be applied. The relation between SHT and FFT can be explained as follows;

The spherical harmonics are the eigen functions of the Laplace operator ( $\nabla^2$  or  $\nabla$ ) on the surface of unit sphere. The spherical harmonics satisfy the following equation:

$$\nabla^2\psi = -\psi \quad (3.15)$$

Basis functions of the Fourier transform are a set of orthonormal solutions to the above equation on  $R^n$ , Euclidean space.

Basis of the spherical harmonic transform are a set of orthonormal solutions to the Equation 3.15 on  $S^2$ , the surface of the unit sphere. Hence, the spherical harmonic transform is just the Fourier transform for the unit sphere.

#### 3.4.1 Pixelization Assumptions

A spherical coordinate system for the face geometry representation will be used throughout this study. Here first define a coordinate system as  $(\theta, \varphi)$ .  $\theta$  is the angle down from the north pole (latitude). This angle is in radians and 0 at the north pole and  $\pi$  at the south pole.  $\varphi$  is the azimuthal angle (longitude). This angle is also in radians and goes from 0 to  $2\pi$ .

It is assumed that the pixels (or surface samples) lay on rows of constant  $\theta$ . Pixels on the same row cover equal area on the unit sphere surface, and the longitudinal spacing between pixels is fixed for each row.

Given the above assumptions, FFT can be used in SHT calculation. As defined in Section 3.1,  $Y_{l,m}$  can be defined as follows;

$$Y_l^m(\theta, \varphi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{im\varphi} \quad (3.16)$$

Here define

$$Q_{l,m}(x) = \sqrt{\frac{(l-m)!}{(l+m)!}} P_l^m(x) \quad (3.17)$$

as the scaled associated Legendre polynomials. The following recursion equation can be used to compute the scaled associated Legendre polynomials.

$$Q_{m,m}(x) = \sqrt{\prod_{i=1}^m \left(1 - \frac{1}{2i}\right)} (-1)^m (1-x^2)^{m/2} \quad (3.18)$$

$$Q_{m+1,m}(x) = x\sqrt{(2m+1)} Q_{m,m}(x) \quad (3.19)$$

$$Q_{l,m}(x) = (l^2 - m^2)^{-1/2} \left[ x(2l-1)Q_{l-1,m}(x) - \sqrt{(l-1)^2 - m^2} Q_{l-2,m}(x) \right] \quad (3.20)$$

### 3.4.2 Forward SHT

Calculate,

$$f_l^m = \int_{\Omega} f(\theta, \varphi) Y_l^{m*}(\theta, \varphi) d\Omega = \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \sin\theta f(\theta, \varphi) Y_l^{m*}(\theta, \varphi) \quad (3.21)$$

Approximate Equation 3.21 with Riemann sum. Assume pixels lie on azimuthal rings and that pixels on the same ring have equal area.

$$f_l^m = \sum_{i=0}^{r-1} \sum_{j=0}^{s_i-1} f(\theta_i, \varphi_{i,j}) Y_{l,m}^*(\theta_i, \varphi_{i,j}) A_i \quad (3.22)$$

In order to use the FFT, assume that rows are evenly spaced across  $\varphi$  such that

$$\varphi_{i,j} = \varphi_{i,0} + \frac{2\pi}{s_i} j \quad (3.23)$$

Lets define a phase shift constant:

$$\kappa_{i,m} = e^{\sqrt{-1}m\varphi_{i,0}} \quad (3.24)$$

Now write the following expression for  $f_l^m$

$$f_l^m = \sqrt{\frac{2l+1}{4\pi}} \sum_{i=0}^{r-1} A_i \kappa_{i,m}^* Q_{l,m}(\cos\theta_i) \sum_{j=0}^{s_i-1} f_{i,j} e^{-\sqrt{-1}mj\frac{2\pi}{s_i}} \quad (3.25)$$

Lets define the function  $G$  which can be calculated using FFT.

$$G_{i,m} = \sum_{j=0}^{s_i-1} f_{i,j} e^{-\sqrt{-1}mj\frac{2\pi}{s_i}} \quad (3.26)$$

finally,

$$f_l^m = \sqrt{\frac{2l+1}{4\pi}} \sum_{i=0}^{r-1} A_i \kappa_{i,m}^* Q_{l,m}(\cos\theta_i) G_{i,m} \quad (3.27)$$

### 3.4.3 The Backward SHT

In order to calculate;

$$f(\theta, \varphi) = \sum_{l=0}^L \sum_{m=-l}^l f_l^m Y_l^m(\theta, \varphi) \quad (3.28)$$

Substitute and switch sum order;

$$f(\theta, \varphi) = \sum_{l=0}^L \sum_{m=-l}^l \sqrt{\frac{2l+1}{4\pi}} Q_{l,m}(\cos\theta_i) \kappa_{i,m}^* f_l^m e^{-\sqrt{-1}mj\frac{2\pi}{s_i}} \quad (3.29)$$

Define  $\alpha$ :

$$\alpha_{m,i} = \sum_{l=0}^L \sqrt{\frac{2l+1}{4\pi}} Q_{l,m}(\cos\theta_i) \kappa_{i,m}^* f_l^m \quad (3.30)$$

Finally, the following expression can be computed using FFT.

$$f(\theta, \varphi) = \sum_{l=0}^L \alpha_{m,i} e^{-\sqrt{-1}mj\frac{2\pi}{s_i}} \quad (3.31)$$

## CHAPTER 4

### 3D FACE REPRESENTATION AND RECOGNITION USING SHT

Our proposed face recognition algorithm consists of two stages. The first stage is feature extraction. The second stage is classification. The feature extraction stage is the stage where we compute SHT coefficients for further classification. The feature extraction stage is composed of the following steps:

1. Preprocess the 3D face data
  - 1.1. Fill the gaps in 3D
  - 1.2. Remove the spikes and filter
2. Crop the face and place it on the ellipsoid
  - 2.1. Using texture image mark the two eye, nose and chin points
  - 2.2. Crop the image using two co-centric ellipses
  - 2.3. Offset the face so that the nose is the origin
  - 2.4. Using the marked points align the face horizontally
  - 2.5. Align the face vertically using PCA
  - 2.6. Offset the nose to the predetermined ellipsoid
  - 2.7. Stitch the face to the ellipsoid (with or without symmetry)
3. Transform the spherical face and obtain SHT coefficients

In classification stage, different algorithms are used to distinguish the faces using SHT coefficients. These algorithms are as follow:

- K-Nearest Neighbor with Euclidean distance
- Linear Discriminant Analysis
- Support Vector Machines

## 4.1 Input face data

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 900/910 series 3D laser scanner. This device can produce 3D shape and color information registered to each other.

The resolution of 2D images is 480 rows by 640 columns pixels. Color information provided by these images has 24 bits per pixel to code color in RGB (Red-Green-Blue) color space.

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 flag.  $x$ ,  $y$  and  $z$  values show the position of the point in the space. Validity flag is a marker which indicates whether there is a valid position data at the specified point. If there is no valid position data, 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 as invalid. An example is given in Figures 4.1, 4.2 and 4.3. Invalid points are black where as white points are valid. Some parts on the face especially eyes, eyebrows have gap problems. Although these points are not occluded to the range scanner, the scanner sensor cannot provide depth information at these points.



Figure 4.1 FRGC registered color image

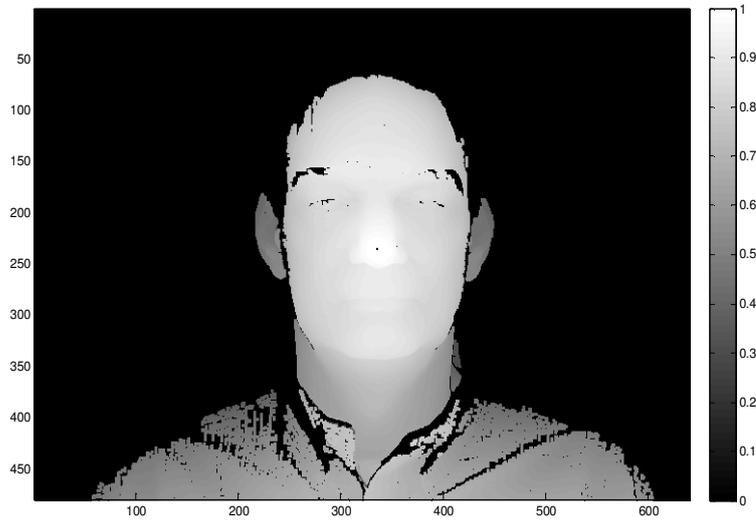


Figure 4.2 FRGC range image

Figure 4.1 shows a sample image from the FRGC database. The image is the registered visible-light image of the face. Figure 4.2 is the range image of the face on Figure 4.1. The black points on this image are marked as invalid, hence have a depth value of -999999. A grayscale color map is used to indicate the depth information. Figure 4.3 shows the validity flag map of the image.

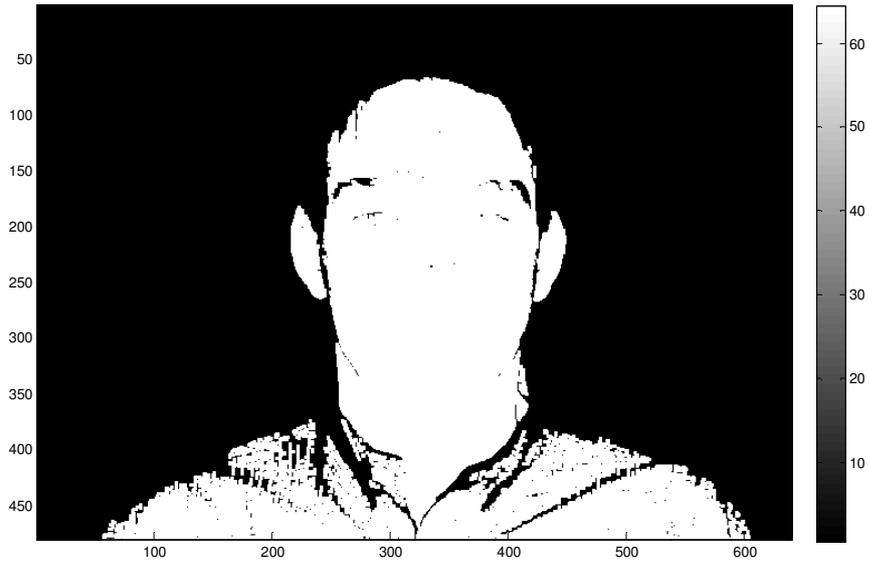


Figure 4.3 FRGC validity flag map

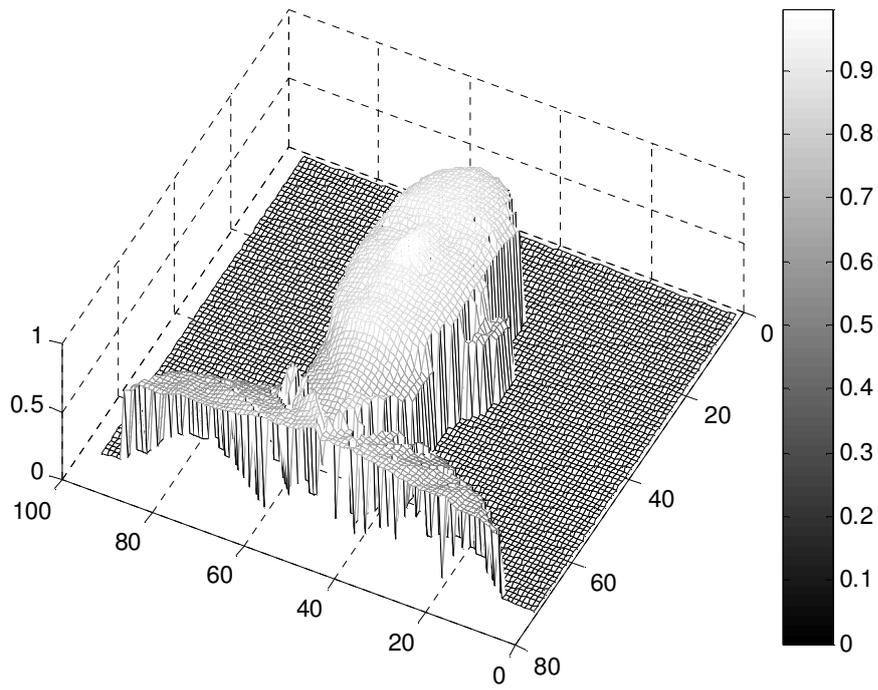


Figure 4.4 Range image shown as a mesh



Figure 4.5 Face reconstructed using range and color image

As can be seen on Figure 4.3, unfortunately not every point on the face image has valid  $x$ ,  $y$ , and  $z$  values. For example, there are gaps on the eye corners in the Figure 4.3. This missing data regions should be filled in the preprocessing step.

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

## 4.2 Preprocessing of Images

As seen on Figure 4.3, there are missing information on the face surface, and the face image is large, i.e. we need only the face surface, not the shoulders or background, and also the range images in FRGC database are not aligned. There is a need for preprocessing the range images in order to make them ready for the 3D face matching algorithm. The preprocessing steps can be listed as follows;

- Gap filling
- Noise filtering

- Point marking
- Cropping
- Registration

#### 4.2.1 Gap Filling

Since the range images in the FRGC database are acquired by Minolta Vivid series of scanners which uses laser range scanning technique, certain areas of the scene are not accurately sampled or even not sampled due to surface reflectance properties or lighting conditions. So there are holes or undesirable artifacts on the face shape. In order to process the face surface further we need to fill the gaps on the face. Wang et al [69] proposes an algorithm based on moving least squares estimation which interpolates the missing parts of the surface locally smooth. Instead of using this complex algorithm, we developed a two pass algorithm for gap filling. First a row-wise linear interpolation is done followed by a column-wise linear interpolation. More accurate results can be obtained using bilinear interpolation by looking at the pixel values at the top and bottom edges of the gap as in texture mapping algorithms.

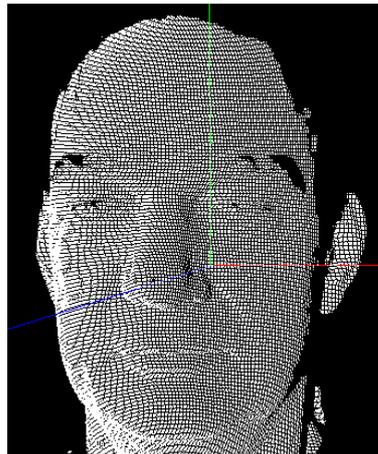


Figure 4.6 Gaps on the face

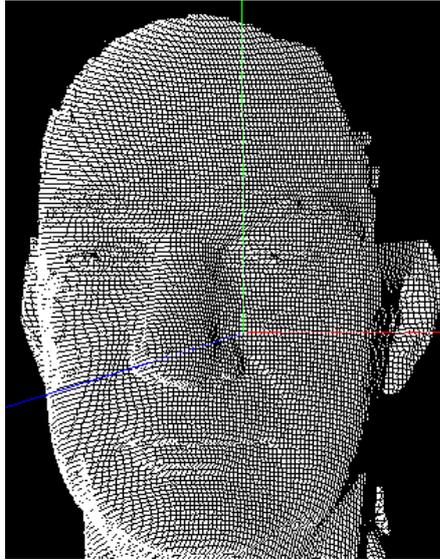


Figure 4.7 Face after filling

#### 4.2.2 Noise filtering

Although, after the gap filling step the range image looks cleaner, there are still some noise on the data as can be noticed in Figure 4.8. The noise is mainly due to acquisition characteristics of the device. 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 noise filtering step has almost two sub steps. First the spikes are removed by thresholding. The distance differences of all the pixels with respect to their neighbor points are calculated and those having larger distance than a threshold value are deleted. So that, impulse-like noises in the image are eliminated by noise filtering as in Figure 4.9.

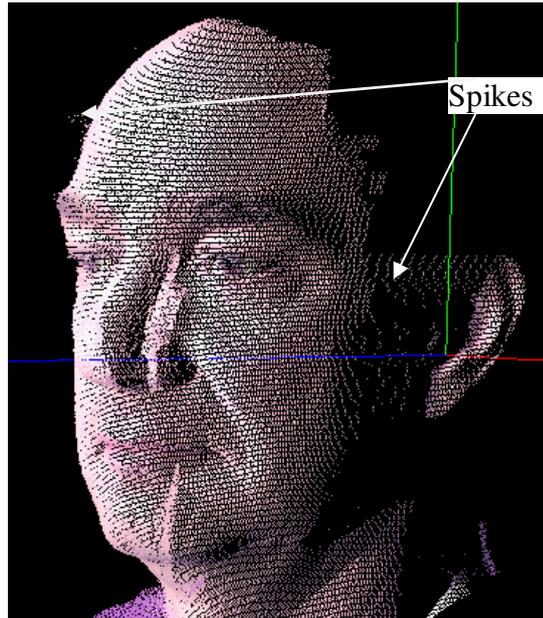


Figure 4.8 Noisy artifacts on the face surface

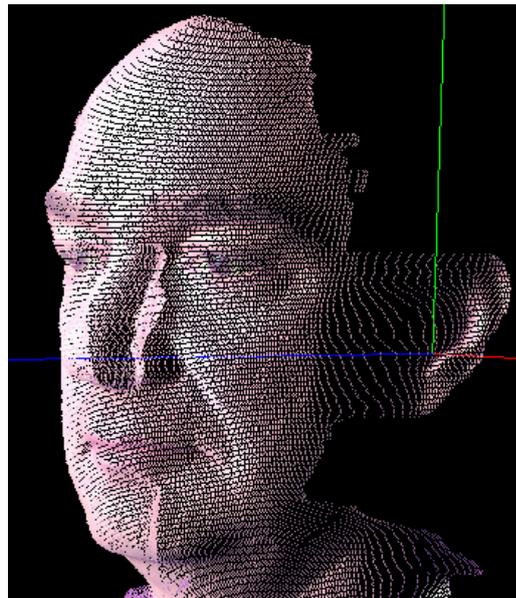


Figure 4.9 Image after spike removal

After spike removal, a 2x2 distance weighted smoothing filter is applied to the range image. The result can be observed in Figure 4.10.

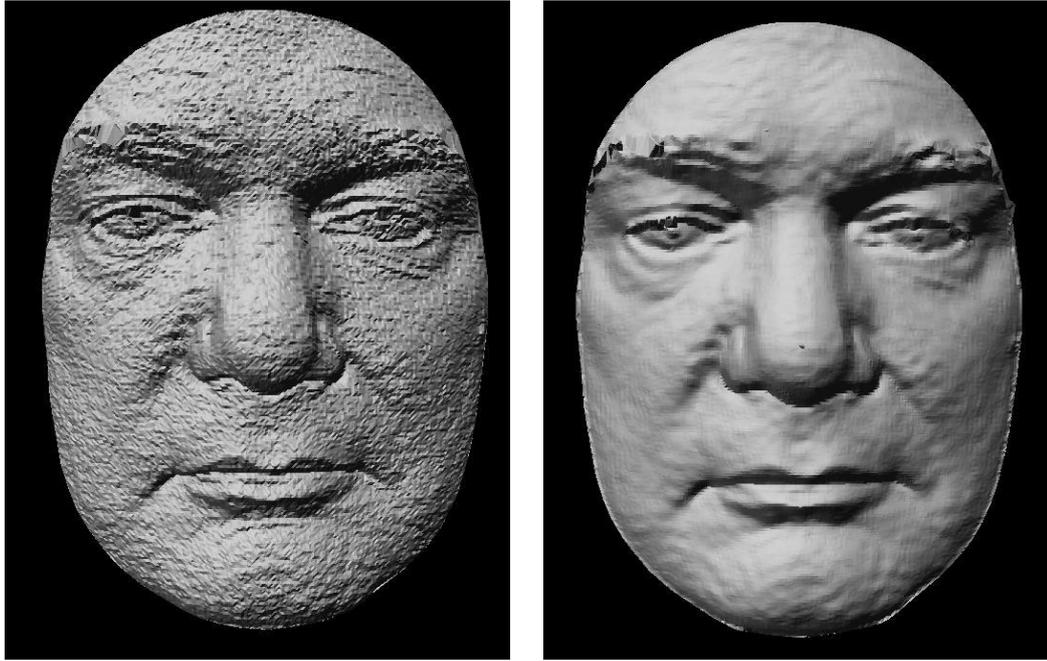


Figure 4.10 Effect of smoothing filter

### 4.2.3 Marking the Points on the Face

The FRGC face database provides a relational database for the faces which includes coordinates of the eye, nose and chin points. The outer eye corners rather than eye centers are used as landmark points because the eyeball is an artifact-prone region for the range sensor, whereas the eye corners marked on the skin are more reliable [84]. These marked points are used for both cropping and 3D registration.

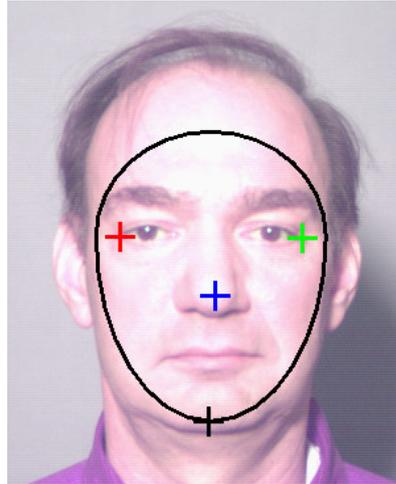


Figure 4.11 Points marked on the face

#### 4.2.4 Cropping

As can be seen on Figure 4.1, besides the face, the FRGC face image also contains the shoulders and a large background area. The proposed 3D face matching method requires only the face surface to be defined. So, there is a need to detect the position of the face within the image, and outline the face surface for further processing. There are many methods available to detect faces in color or monochrome images, even in complex backgrounds [70]. Automatic face detection and outlining is out of the scope of this study. So we manually marked the eyes, chin and nose tip as shown in Figure 4.11.

We used 3 of the points marked at the previous stage for cropping. Using the two eye points we determined the center point. A bottom half ellipse and a top half ellipse with the calculated center is used. The two ellipses have a common horizontal radius which length is the distance between two eyes. The vertical radius for the top-half is an adhoc distance, hence 80 pixels above the center. The vertical radius for bottom ellipse is the vertical distance between the center and chin point. In Figure 4.12 notice that, the nose point is not used for cropping.

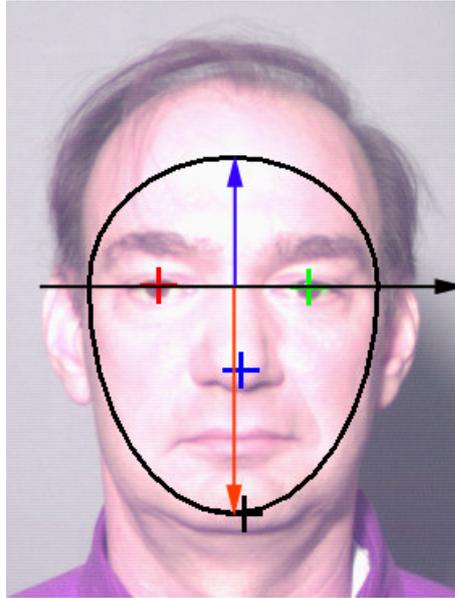


Figure 4.12 Elliptical crop region

2D and 3D face images after cropping are show on the Figures 4.13 and 4.14.

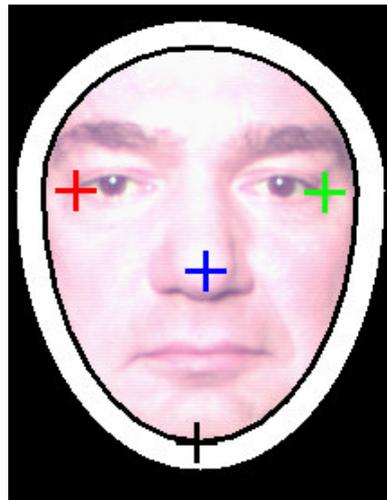


Figure 4.13 Cropped 2D image

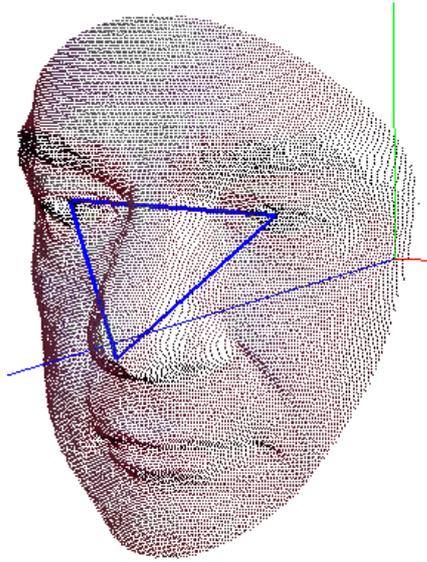


Figure 4.14 Cropped 3D image and registration triangle

## 4.2.5 Registration

The input to the registration method is the cropped face surface marked at the two eye points and the nose. The first step is to offset the surface so that the nose point becomes the origin of the surface. We used two different methods for further alignment of the face.

### 4.2.5.1 ICP Registration

As mentioned in the literature review, most of the 3D face recognition methods used Iterative Closest Point (ICP) technique to align face surfaces. In this study, we also used the baseline ICP algorithm. This algorithm uses all the points in both model and test meshes (no subsampling or rejection is performed), computes point-to-point distances between pairs and weights are given uniformly to every pair. As the model, a depth image is obtained by averaging all the 941 faces in the database.

Firstly, a face model is constructed by averaging all the faces in the database. 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 matching experiments will be 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 the further steps are performed.

By considering all the images in the database, a proper choice is made to determine the starting and ending points of the grid (Figure 4.15).

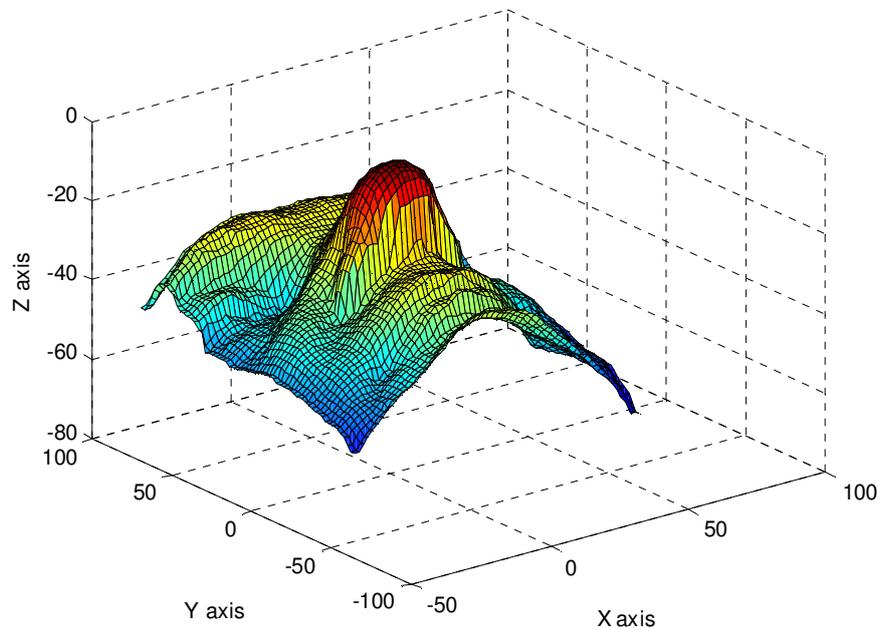


Figure 4.15 Registered face image centered at nose tip

Since the faces are all placed on the same grid, and all nose tips are at the origin z values can be used as features.

The drawback of the ICP algorithm is that it is computationally very expensive. And also while building up the model face by averaging all faces in the database, some more cropping should be performed in order to keep the face within the

boundaries of the predetermined grid, and this causes lots of surface information be lost. Because of these drawbacks, even though the ICP method is investigated and implemented in this study, the next method “geometrical” performed very well, and preferred over ICP.

#### 4.2.5.2 Geometrical Registration

Since we have landmark points on the eye and nose, we can make use of this information to align the face surfaces geometrically. We call the triangle defined by these points as registration triangle. As seen on Figure 4.16, after offsetting the surface to make the nose point the origin, the head of the *registration triangle* is on the z-axis (outward).

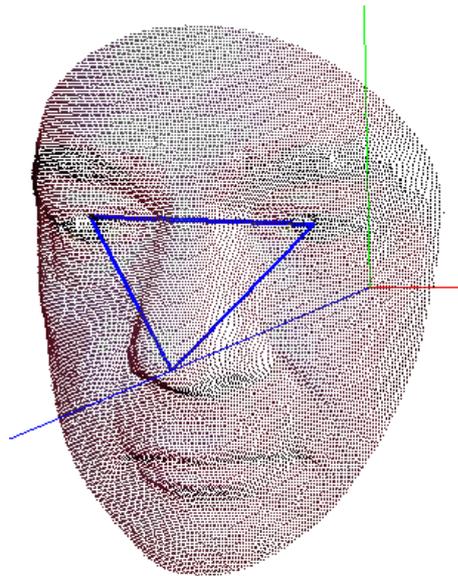


Figure 4.16 Face after offsetting on nose point

In the first step of alignment, the line through the two eyes is aligned with x-axis by rigidly rotating the face surface first around z-axis and afterwards around y-axis (see Figure 4.17) assuming the nose-tip is the origin of the face surface. After these alignments, the only alignment we need to perform is the alignment

around the  $x$ -axis. This can be done geometrically, however it won't give accurate results, because person's nose sizes differ. As can be seen on Figure 4.18, for the alignment we could set a predefined angle for the angle  $\alpha$ , however this would give inaccurate results, because this angle depends on the size of the nose. However, rotating the face around  $x$ -axis by  $\beta$  angles so that the principal component of the face surface becomes aligned with  $y$ -axis give more accurate results. The principal axis (see Figure 4.18) of the surface can be computed using PCA algorithm in 3D Euclidean space. The principal axis will be the axis with the largest eigenvalue.

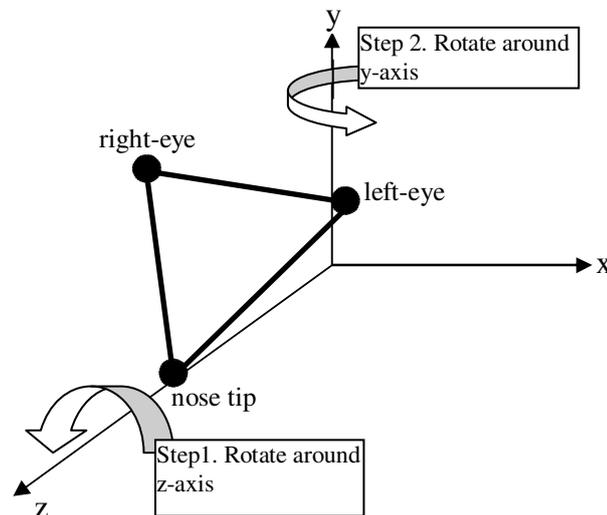


Figure 4.17 Rotations around  $z$  and  $y$ -axes

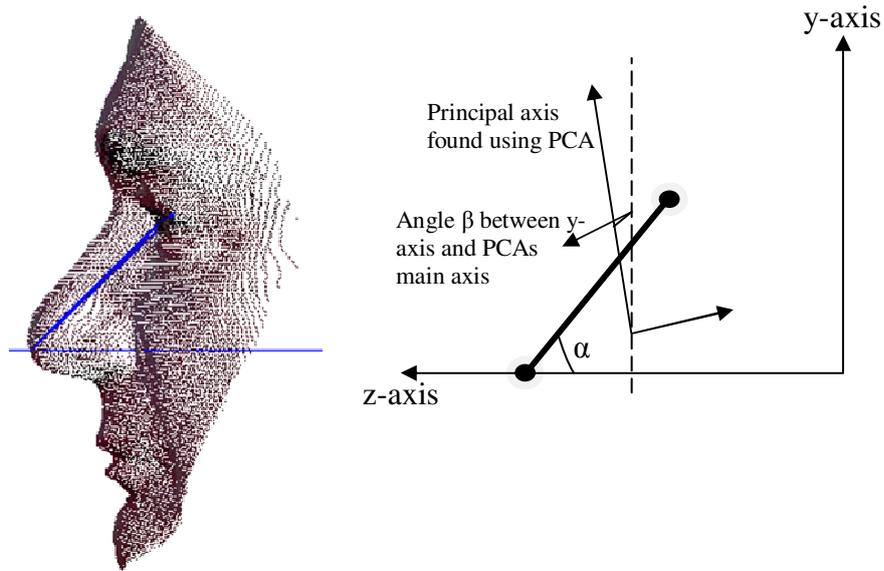


Figure 4.18 PCA Alignment

### 4.3 Application of Spherical Harmonic Projection

Let  $M$  denote a mesh of an object embedded in  $R^3$ .  $M$  is said to be star-shaped if there exists a point  $c \in R^3$  such that every line segment drawn from  $c$  in any direction intersects the surface of  $M$  at exactly one point. Considering that  $c$  is the center of the spherical coordinate system, the radial function induced by  $M$  and  $c$  is a well defined spherical function  $f: S^2 \rightarrow R^+$ , where  $S^2$  is the unit sphere.

The paragraph above states that the face surface should be star-shaped, or basically a spherical function. It is a fact that the human face is not star shaped, and hence not a spherical function. For example, think off the nose, where there are holes inside, and makes it obviously a non-spherical surface. However, in this study we're using the face range image which is acquired from front-view with a flat sensor, i.e. the range image itself is two-dimensional. A two-dimensional image can be assumed spherical when mapped on the sphere.

So the question how can we make the face surface spherical arises? The obvious solution to this problem is to map or cover the face surface on a sphere or ellipsoid with a predefined radius.

In this study we applied and compared the performances of 4 different spherical mapping methods. These are;

- Partial mapping
- Hemi-spherical mapping
- Full-spherical mapping
- Full-spherical mapping with symmetry

#### **4.3.1 Partial Mapping Trial**

In partial mapping, we map the face surface to the spherical surface patch which has the extends where the surface itself is defined. The Figures 4.19 and 4.20 show how partial mapping is done. The main idea behind partial mapping is that the face surface is not defined outside the cropped face surface, so we can assume that the surface outside is zero.

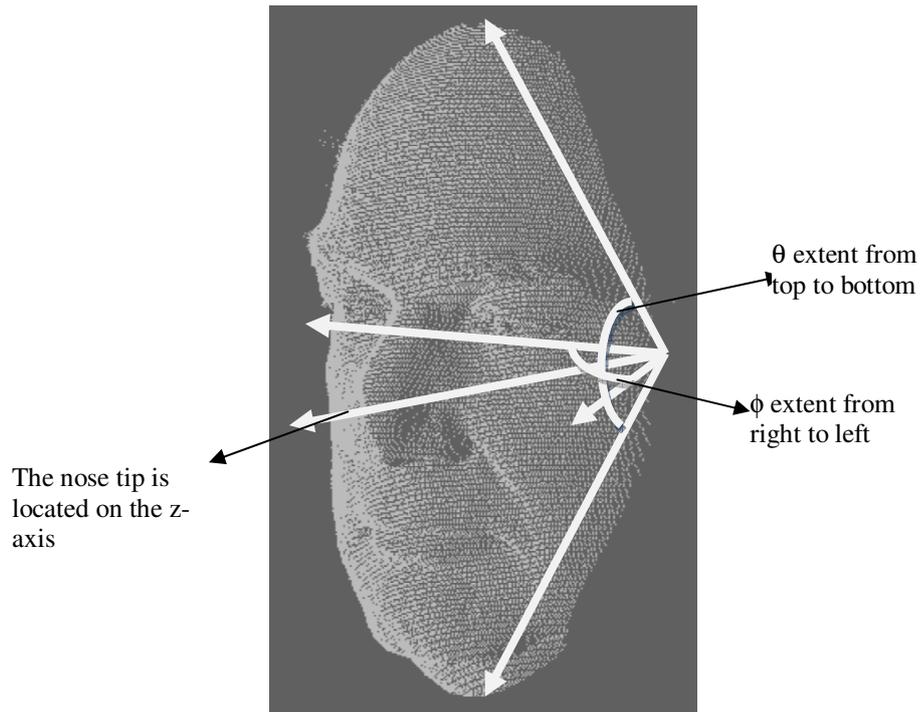


Figure 4.19 Partial mapping extent

The result of partial mapping can be observed on Figure 4.20. Notice that the spherical face surface is not defined (or simply zero) outside the extends of the cropped face surface.

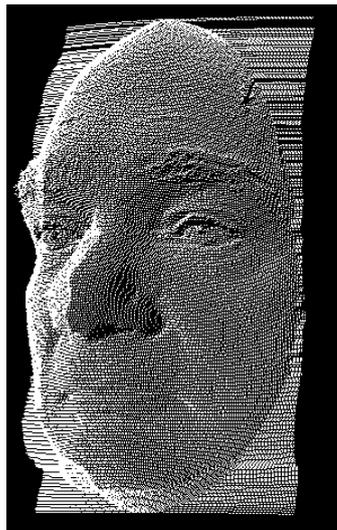


Figure 4.20 Partial mapping

Notice that in partial mapping, the missing parts outside the cropping region are filled with the nearest neighbor's depth value on the same row (constant longitude). This makes the spherical presentation of face surface continuous on the edges of the elliptical clipping region.

While forward spherical harmonic transform of the partially mapped surface, we integrate only within the extents of the face surface as follows;

$$f_l^m = \int_{\Omega} f(\theta, \varphi) Y_l^{m*}(\theta, \varphi) d\Omega \quad (4.1)$$

where  $\Omega$  is defined for  $\theta_{start} \leq \theta \leq \theta_{end}$  and  $\varphi_{start} \leq \varphi \leq \varphi_{end}$ . And while backward SH transform; we use the same extent for integration.

Since in this study we'll use the SHT co-efficients just for face classification, there is no need to backward transform the SHT coefficients. However, we'll use the backward transformed face surface as a quality metric for face representation. We define the face representation error as follows;

$$E_p = \frac{1}{N} \sum_{\Omega} |z_r - z_o| \quad (4.2)$$

Where  $E_p$  is representation error,  $N$  is the number of samples defined in the face surface extent  $\Omega$ .  $z_r$  is the depth value on the reconstructed surface,  $z_o$  is the depth value on the original face surface.

Figures 4.21, 4.22, 4.23, 4.24 and 4.25 show the reconstructed surface of a sample face surface for different spherical harmonics bandwidth  $L_{max}$ .

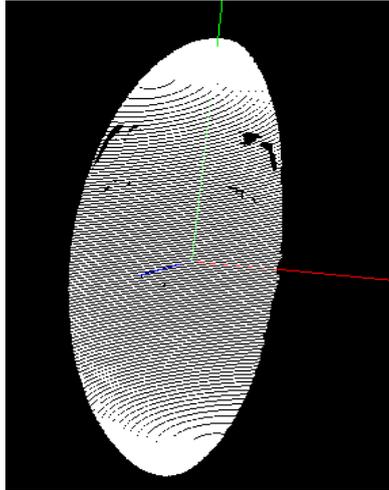


Figure 4.21 Bandwidth  $L_{\max}=1$

Notice that on Figure 4.21, only one spherical harmonic is used to represent the surface. In this case, only a partial sphere clipped by the elliptical clipping region is displayed. The radius of this one and only sphere is the average pixel distances to the origin of the original face surface. We can assume this radius as the average or DC (direct current) value of the original face surface.

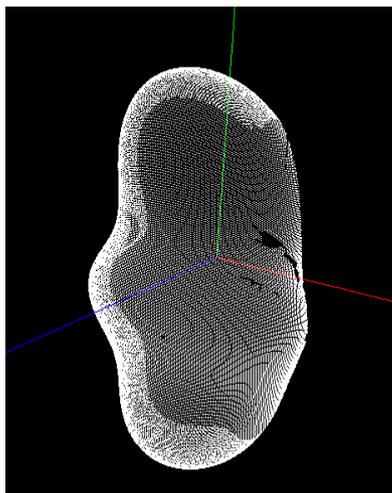


Figure 4.22 Bandwidth  $L_{\max}=10$

In Figure 4.22, the reconstructed surface with  $L_{max}=10$  is shown. Notice that the reconstructed shape looks like a face. However, it is still far from an exact representation. So, we need higher order frequencies for an exact representation of the facial surface.

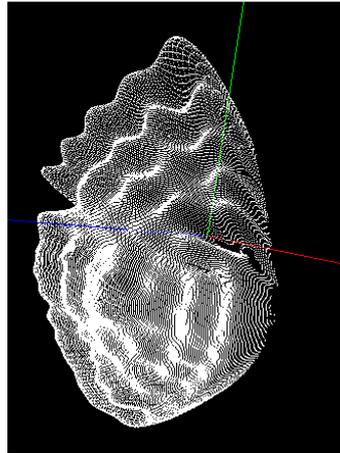


Figure 4.23 Bandwidth  $L_{max}=50$

In the figures 4.23, 4.24 and 4.25, the reconstructed surfaces with  $L_{max}=50$ ,  $L_{max}=100$  and  $L_{max}=300$  are shown respectively.

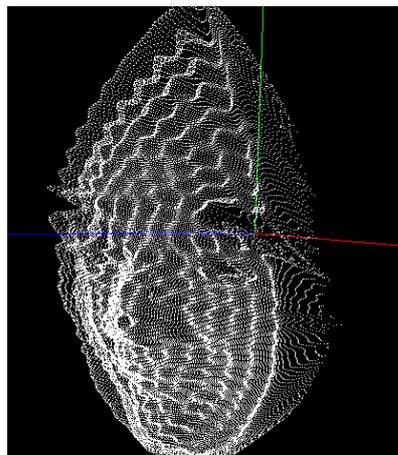


Figure 4.24 Bandwidth  $L_{max}=100$

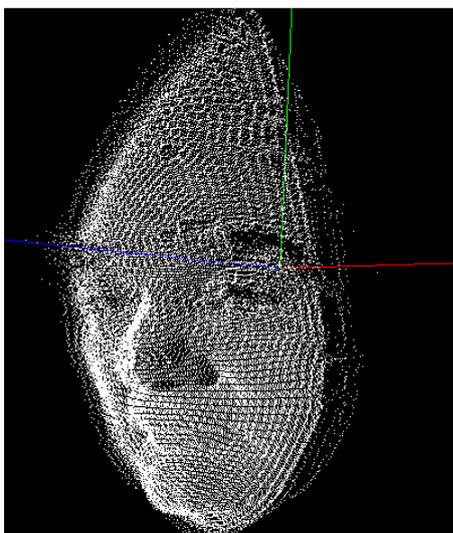


Figure 4.25 Bandwidth  $L_{max}=300$

In Figure 4.23, notice that even for a bandwidth of 300 there is still a huge amount of information loss in the partial face representation. And also notice that for a bandwidth of 300, there are  $L_{max} * (L_{max} + 1) = 300 * 301 = 90300$  coefficients. This large number of coefficients is not suitable for representation and also for classification. The main reason for a need for higher order frequencies is that the SHT basis functions are not localized, and the SHT transform assumes that the spherical face surface is zero outside its defined domain. As can be seen clearly in Figure 4.25., the backward SHT tries to pull the face surface to the origin (zero) just on the edge of the partial face. These high frequency components on the edge occur mainly due to the discontinuity on the partial spherical face surface.

### 4.3.2 Hemi-spherical Mapping Trial

In hemi-spherical mapping, we first find the extends of the cropped face surface as in partial mapping, where  $\theta_{start} \leq \theta \leq \theta_{end}$  and  $\varphi_{start} \leq \varphi \leq \varphi_{end}$  denote the extends of the original surface. This time we do the forward transform integration within the hemi-sphere defined as follows;

$$f_l^m = \int_{\Omega} f(\theta, \varphi) Y_l^{m*}(\theta, \varphi) d\Omega \quad (4.3)$$

where  $\Omega$  is defined for  $0 \leq \theta \leq \pi$  and  $-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}$ . The Figure 4.26 shows a hemi-spherical mapped face surface.

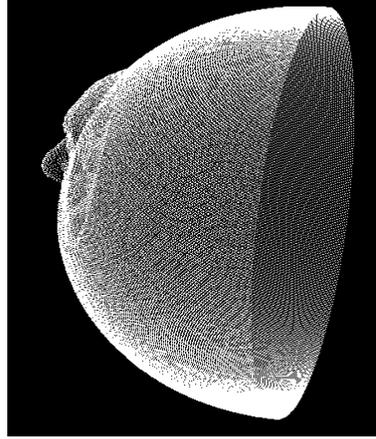


Figure 4.26 Hemi-spherical mapped surface from left-viewpoint

The radius of the hemi-sphere is taken as the average radial distance of all the surface pixels. The points inside  $\Omega$ , but outside the cropping region are interpolated to this radius with a weight which depends on the distance of the pixel to the center of the clipping region. The reconstructed surface for  $L_{max}=50$  is depicted in the Figure 4.27. Notice that the reconstructed surface is much more accurate than the partial mapping one.

There are two main problems with hemi-spherical mapping. First, using a different radius with every face is not suitable. It is a fact that every person's face has a different size, however using different radius for every face causes some information lost while SH transform which is not suitable for our classification purposes.

The second problem is that the SH transform again assumes that the spherical face surface is zero outside the hemi-sphere. Since the SH basis functions are not localized, and defined over the whole sphere, the SH transform tries to pull the

surface to the origin outside the defined hemi-sphere. This causes a need for high frequency components for a better representation of the face surface.

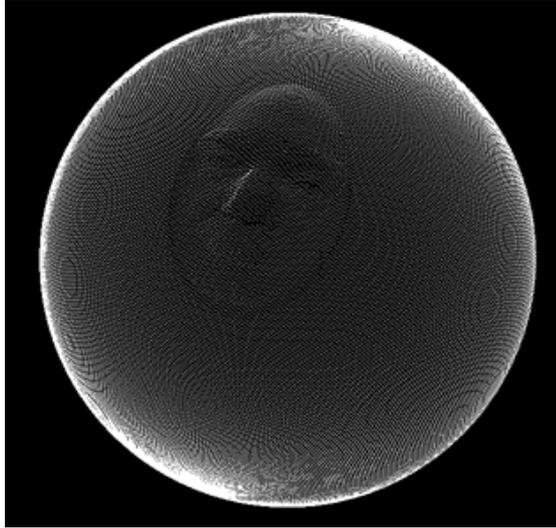


Figure 4.27 Reconstructed hemi-spherical surface

### 4.3.3 Full-Spherical Mapping

After observing high frequency SHT components on the edges of the spherical face surface with partial and hemi-spherical mapping, we decided to use a face surface which is defined over the whole sphere. And also, to omit distortion on the elliptically cropped face surface while mapping on a spherical patch, we decided to use ellipsoid instead of a sphere. As seen on the Figure 4.28, an ellipsoid with two different predefined radiuses is used for mapping.

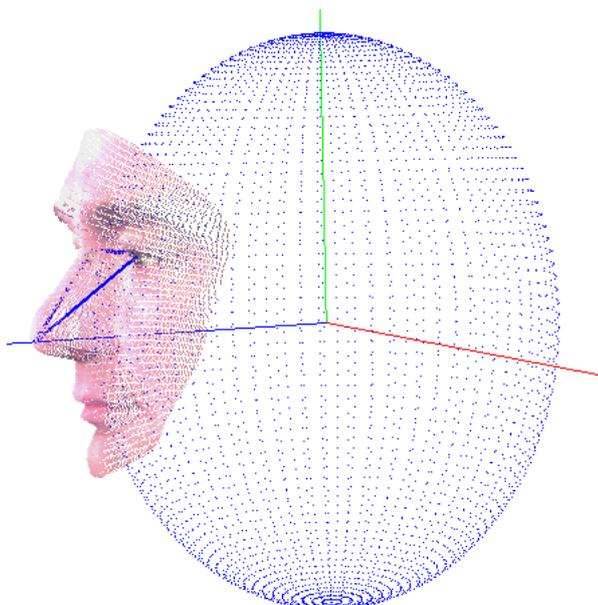


Figure 4.28 Full spherical mapping

The radiuses of the ellipsoid are set as 15 cm vertically and 12 cm horizontally. These values are adhoc values, or decided after observing face images in the database.

While mapping the face to the ellipsoid, first the face surface is translated along the z-axis, so that the point at the center of the eyes and nose is exactly located on the ellipsoid's surface. Next step is to stitch the face surface to the ellipsoid so that there is no geometric discontinuity (first order) between face and ellipsoid surface. In order to stitch the face to the ellipsoid smoothly, a stitching band is defined over which the spherical face surface reconstructed by linear interpolation of the face surface and ellipsoid's surface. This stitching method can be observed on Figures 4.29 and 4.30.

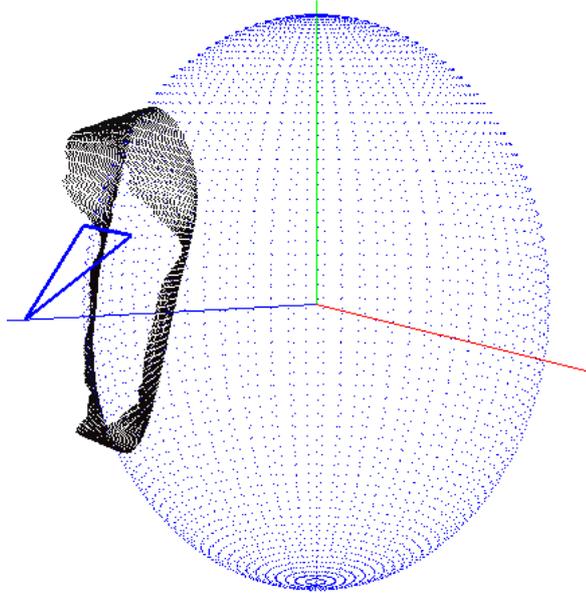


Figure 4.29 Stitching band of the previous face image

The stitched face surface can be seen on Figure 4.30.

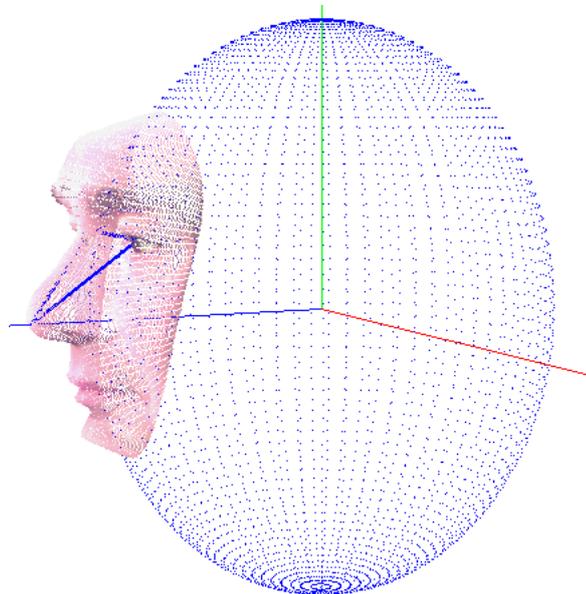


Figure 4.30 Stitched face surface

Notice that there are some distortions, and hence information lost on the cropped face. However these lost is mostly below the chin and above the eyebrows and this lost does not affect the classification results.

Finally, the stitched and full spherically mapped surface and the reconstructed surface can be seen on the Figures 4.31 and 4.32.

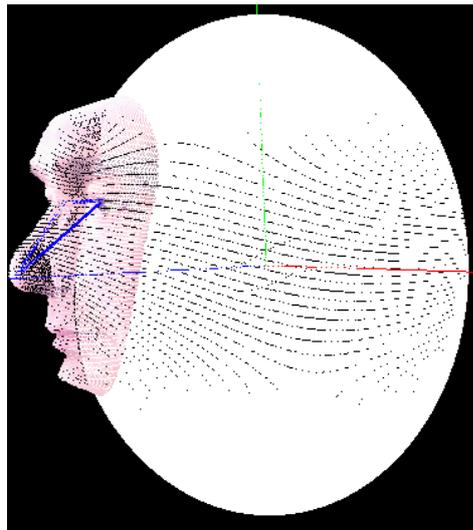


Figure 4.31 Full-spherical face surface

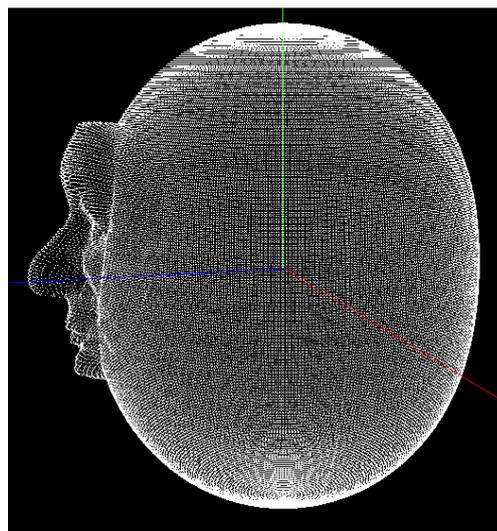


Figure 4.32 The reconstructed surface for  $L_{max}=200$

#### 4.3.4 Full-Spherical Mapping with Symmetry

In order to make use of spherical harmonics being symmetric functions, it would be a clever method to use symmetry while mapping the face surface. In this case, the symmetry of the surface constructed in hemi-spherical one is used.  $\Omega$  is defined for  $0 \leq \theta \leq \pi$  and  $0 \leq \varphi \leq 2\pi$ , and surfaces defined in  $0 \leq \varphi \leq \pi$  and  $\pi \leq \varphi \leq 2\pi$  are symmetries of each other (Figure 4.33).

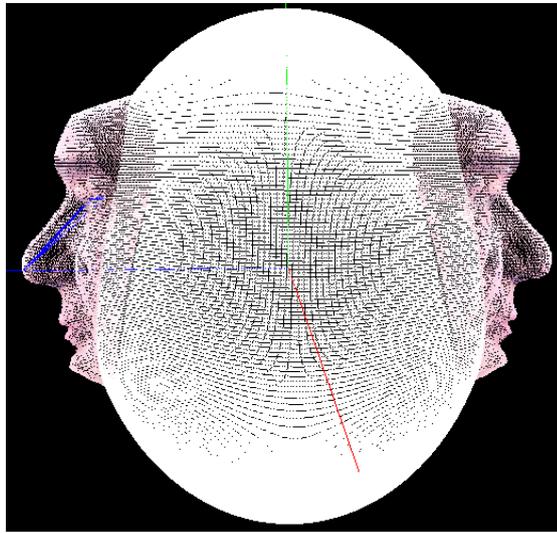


Figure 4.33 Full-spherical with symmetry face

In Figure 4.34, a reconstructed surface with bandwidth  $L_{max}=200$  is shown. The spherical-symmetry representation of the face surface gives worse results than the one without symmetry. So, there is no need for the symmetrical representation. Indeed, a quantitative error analysis will be performed to compare the representations of with symmetry and without symmetry.

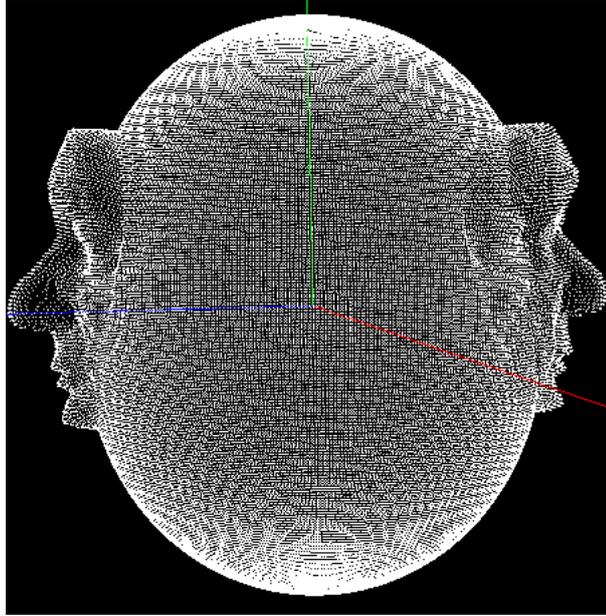


Figure 4.34 Reconstructed symmetrical face for  $L_{max}=200$

#### 4.4 SH Transformation

After the face surface is mapped spherically, the Spherical Harmonics transform is applied to the surface in order to obtain the SHT coefficients  $f_l^m$ . For SH transform we used FFT based Real Spherical Harmonics transform. In Section 3.2, the real spherical harmonics transform was defined by the set of  $Y_l^m$  as given in Eq. 3.11, and the transformation is defined in 3.10.

The equations 3.10 and 3.11 say that for a degree  $l$  there are  $l$  positive  $m$  coefficients and  $l$  negative  $m$  coefficients and a coefficient for  $m=0$ . It means that there are  $2l+1$  coefficients for degree  $l$ . The Figure 4.35 shows a 3 dimensional mesh display of SHT coefficients for a sample face surface. Notice that  $0 \leq l \leq 200$  and  $-l \leq m \leq l$ , and y-axis is log of the coefficient, and also as  $l$  increases SHT coefficients becomes smaller.

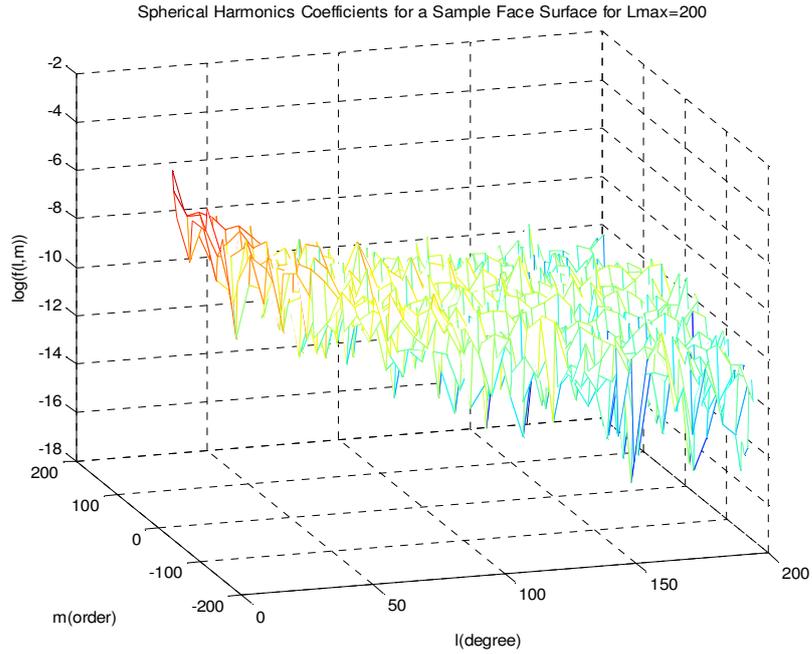


Figure 4.35 Spherical Harmonics Coefficients of a sample face

#### 4.4.1 Power Spectral Density

The power contained at a specific degree  $l$  can be defined as:

$$power_l = \frac{1}{2l + 1} \sum_{m=-l}^{m=l} (f_l^m)^2 \quad (4.4)$$

So the power spectral density (PSD) of a surface can be defined as the power contained in all degrees of the spherical harmonics spectrum. The PSD of a surface gives us information about its shape. In this study, we will use PSD of SHT coefficients for classification. The PSD plot for a sample face with full-spherically mapping without symmetry is depicted on Figure 4.36. Notice that the power contained at higher frequencies decreases, which means that the surface is bandlimited to an  $L_{max}$ . And also notice that the y-axis is in log scale, hence the power at 0<sup>th</sup> degree is very large compared to the power at higher frequencies. This must be compensated in some manner for later classification.

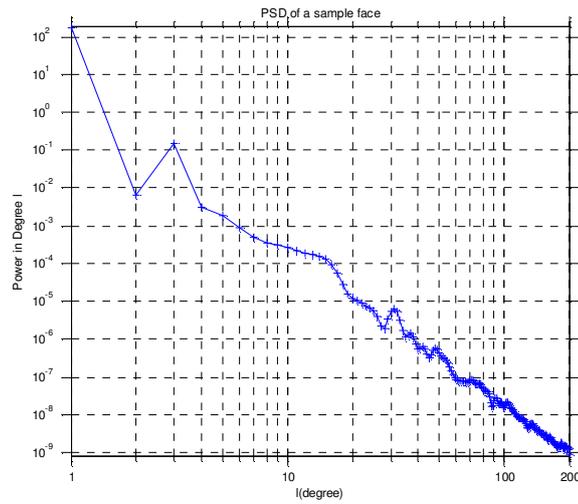


Figure 4.36 PSD for a sample face

#### 4.4.2 Rotation Invariancy

An important property of the PSD is that it is rotation invariant, i.e. the power contained in a frequency band does not change with rotation. This can be seen on Figures 3.37 and 4.38.

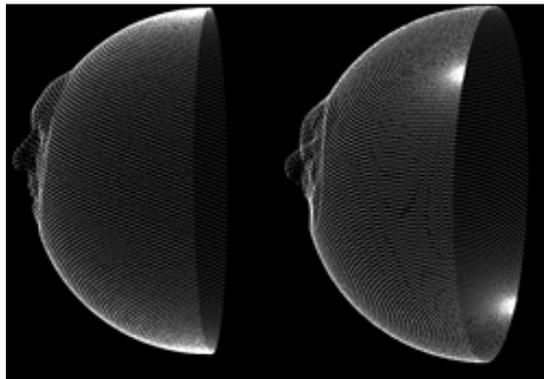


Figure 4.37 Hemi-spherical face surface (left), the same surface rotated around the nose tip (right)

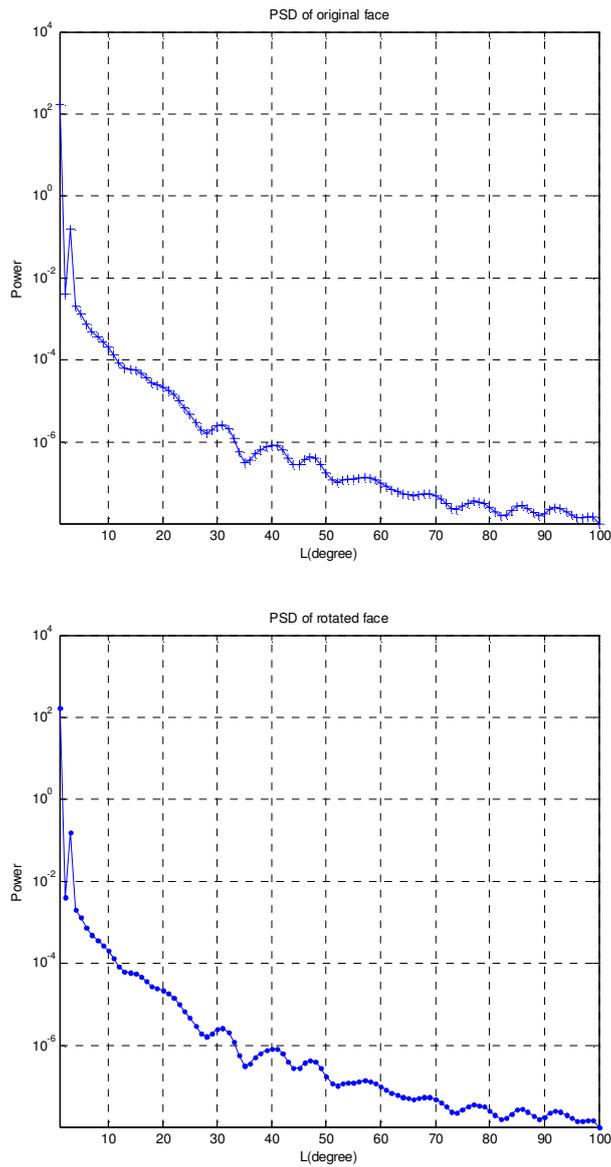


Figure 4.38 Rotation invariancy. PSD of the surfaces given in Figure 4.37

In our study, the nose tip of the face is located along the z-axis of the spherically mapped surface, and it is aligned geometrically. In case of an alignment error due to rotations of the face, this rotation will be discarded by the rotation invariancy of the PSD.

### 4.4.3 Signal to Noise Ratio

In Section 4.3.1, we proposed representation error as follows;

$$E_p = \frac{1}{N} \sum_{\Omega} |z_r - z_o| \quad (4.5)$$

where  $N$  is the number of samples defined in the face surface extent  $\Omega$ .  $z_r$  is the depth value on the reconstructed surface,  $z_o$  is the depth value on the original face surface. Using the same variables, we defined reconstruction signal to noise ratio (SNR) as follow;

$$SNR = 10 \log \frac{\text{signal power}}{\text{error power}} \text{ dB} \quad (4.6)$$

Where

$$\text{signal power} = \sum_{i=1}^N |z_o|^2 \quad (4.7)$$

And

$$\text{error power} = \sum_{i=1}^N |z_r - z_o|^2 \quad (4.8)$$

We used SNR to find a limit  $L_{max}$  which we will use for SHT transformation. As can be seen on Figure 4.39, SNR increases with increasing  $L$ , however at some point it begins to converge to some limit., hence we have chosen  $L_{max}=200$  throughout this study.

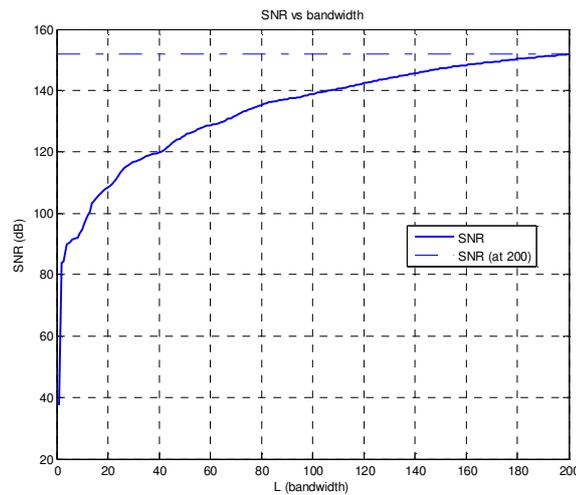


Figure 4.39 SNR vs bandwidth

#### 4.4.4 Effect of Symmetry

As we discussed in Section 4.3, we tried several mapping methods to get a better spherical representation of the face surface, and concluded that full-spherical mapping without symmetry is more convenient for our purposes. Our conclusion was due to fact that SNR plot of symmetrical representation was nearly the same as or a bit worse than the one of the non-symmetrical representation. That is why we abandoned to use symmetrical representation. The effect of the symmetry on the PSD and SNR plot can be seen on the Figures 4.40 and 4.41. Notice that, PSD oscillates by two consecutive frequency bands. The SNR is worse than the one without symmetry. The reason why symmetrical face's SNR is worse is that the symmetrical face contains more details than the one without symmetry. And also with symmetrical mapping, we just use symmetry about the x-axis, however spherical harmonics basis are not only symmetric around the x-axis. Their symmetrical behavior changes with respect to their degree and order.

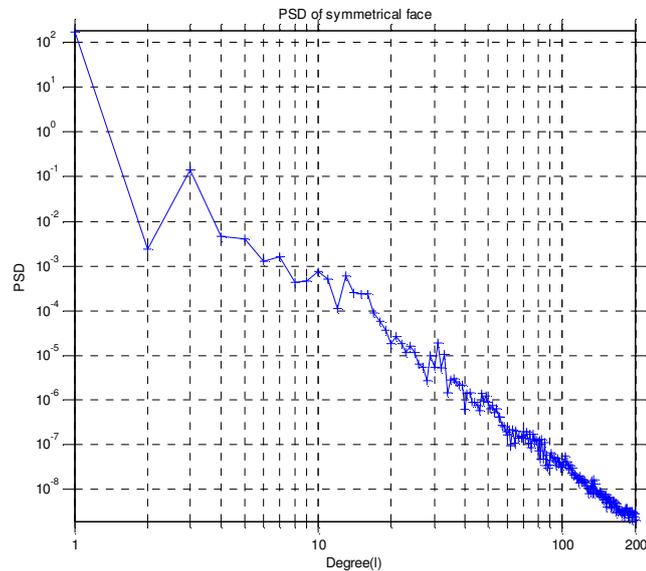


Figure 4.40 PSD of symmetrically mapped face

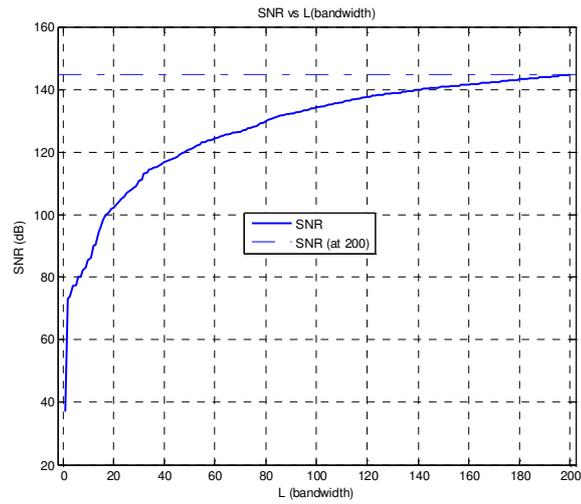


Figure 4.41 SNR for symmetrically mapped surface

#### 4.4.5 Effect of a Single Coefficient

It is known that the spherical harmonics basis functions do not have local support, i.e. they are not localized. So it is expected that a change in a single

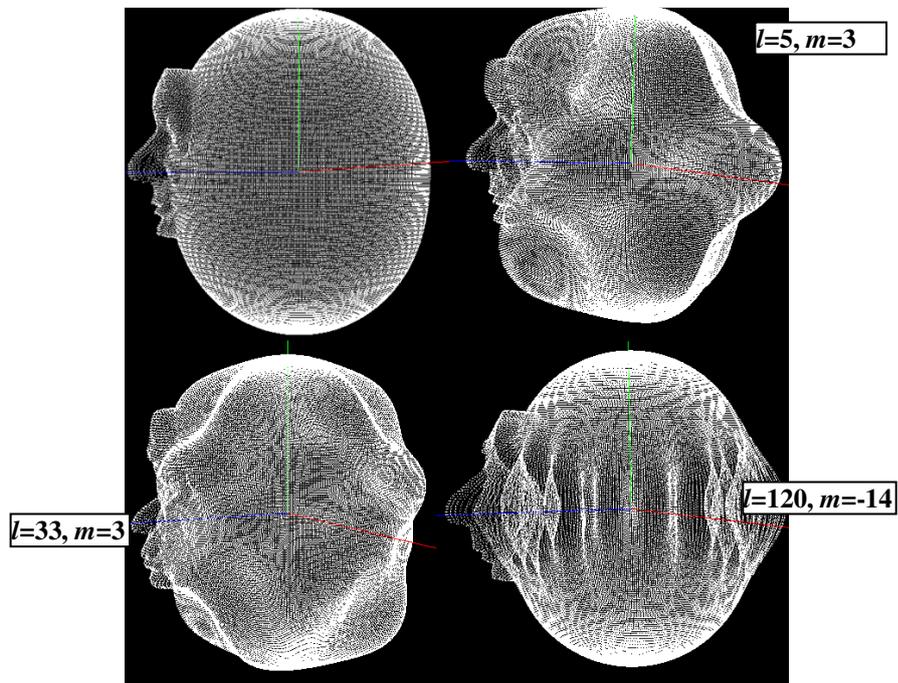


Figure 4.42 Effect of a Single Coefficient

SHT coefficient will affect the whole surface globally. To prove this expectation we performed a test where we reconstructed a face surface by changing a single coefficient by multiplying it by 100. The effect can be seen on Figure 4.42. In Figure 4.42, the top-left image is the original reconstructed surface. The changed coefficient is overlaid on the top of each image. Notice that the effect is global, i.e. affecting the whole face surface.

#### **4.5 Face Classification using SHT Coefficients**

The recognition parameter  $L_{max}$  which specifies the bandwidth for the SHT is the key performance factor in the recognition process. Since the SHT is computationally very expensive, even when FFT used, we limit  $L_{max}$  to 200 with a compromise between computation time and quality of the representation.

In SHT, there are  $L_{max}*(L_{max}+1)$  coefficients. For  $L_{max}=200$  there are 40200 coefficients.

For classification of the face SHT coefficients, we've used the following algorithms:

- k-Nearest Neighbor with k=1 using Euclidean distances of all SHT coefficients
- k-Nearest Neighbor with k=1 using Euclidean distance of PSD of SHT coefficients
- Mahalanobis Distance of SHT coefficients
- Mahalanobis Distance of PSD of SHT coefficients
- Linear Discriminant Analysis (LDA) using SHT coefficients
- Linear Discriminant Analysis (LDA) using PSD of SHT coefficients
- Support Vector Machines using SHT coefficients as feature vector
- Support Vector Machines using PSD of SHT coefficients as feature vector

#### 4.5.1 k-Nearest Neighbor with k=1 for SHT Coefficients

In this method, we will use the Euclidean distance between the SHT coefficients. The distance can be defined as follows;

$$D_{i,j} = \sqrt{\sum_{l=0}^{L_{max}} \sum_{m=-l}^l (C_i(l, m) - C_j(l, m))^2} \quad (4.9)$$

where  $C_i(l, m)$  and  $C_j(l, m)$  are the SHT coefficients for the  $i^{\text{th}}$  and  $j^{\text{th}}$  face surface.

In order to test this method, we used 4 sample faces as shown in Figure 4.43. In this figure, while the Face 1 and Face 2 images belong to the same person, Face 3 and Face 4 belong to different persons.

The distance matrix for these face are also given on Table 4.1.

Table 4.1 Distances between faces in terms of SHT coefficients

	Face1	Face2	Face3	Face4
Face1	0	6.21	8.47	11.62
Face2	6.21	0	7.74	12.05
Face3	8.47	7.74	0	10.77
Face4	11.62	12.05	10.77	0

As seen on the table, the intra-personal coefficient distances are smaller compared to inter-personal distance, therefore SHT coefficients are promising face classification. However, better results can be obtained by analyzing the coefficients statistically.



Figure 4.43 Test faces

The first 4 band coefficients for a sample face (Face 1) are given in Table 4.2.

Table 4.2 Coefficients for the first 4 bands

Degree (l)	Order (m)	Coeff. for $m > 0$	Coeff. for $m < 0$
0	0	13,1275073554070	0
1	0	0,0429682906445230	0
1	1	0,0126484263596761	0,132164932176493
2	0	0,851525474768107	0
2	1	0,00615333523172825	0,0667870146795359
2	2	-0,125784876013191	0,0242323231725326
3	0	-0,0514422309933368	0
3	1	-0,00488640391345551	-0,0396765245979460
3	2	-0,0664134114498508	0,0127169275744366
3	3	-0,0317670863699876	-0,110301928849857
4	0	-0,0295548914359871	0
4	1	-0,00564492539884076	-0,0525142271621880
4	2	0,0139228782861514	-0,00494385060447562
4	3	-0,0165711701011474	-0,0542845458666222
4	4	0,0911563319112260	-0,0342214925809437

Notice in Table 4.3 that the coefficient for  $(l,m)=(0,0)$  is very large compared to the other coefficients. This can be also observed on the PSD plot on Figure 4.36 where a major part of the power is contained at the first bands. This property of the SHT coefficients must be eliminated for classification by transforming the coefficients. In [64] and [71], it is proposed to use *whitening* so that the feature vector is transformed such that its components are uncorrelated and their variances equal to unity

After whitening transform the distance table in Table 4.1. becomes now;

Table 4.3 Face Distances after whitening transform

	Face1	Face2	Face3	Face4
Face1	0	544	614	623
Face2	544	0	602	637
Face3	614	602	0	682
Face4	623	637	682	0

As can be observed on Table 4.3, we have similar results as in Table 4.2. However, this time the differences between distances come more obvious which will further improve the classification.

#### 4.5.2 k-Nearest Neighbor with k=1 for PSD Coefficients

In the previous chapter, we've used the  $L_{max}*(L_{max}+1)$  SHT coefficients to classify the faces by computing the Euclidean distance between the coefficients. This time we'll use power spectral density of the SHT coefficients to classify the faces. And now we have only  $L_{max}$  power bands, and hence  $(L_{max}+1)$  coefficients. Since  $L_{max}$  is chosen as 200, it may seem to be that there are not enough parameters for a successful face classification. However, analyzing the PSD for all the training samples makes it clear that PSD coefficients contain enough information for a successful discrimination. Figure 4.44 shows the mean and variance of the PSD obtained using the training faces.

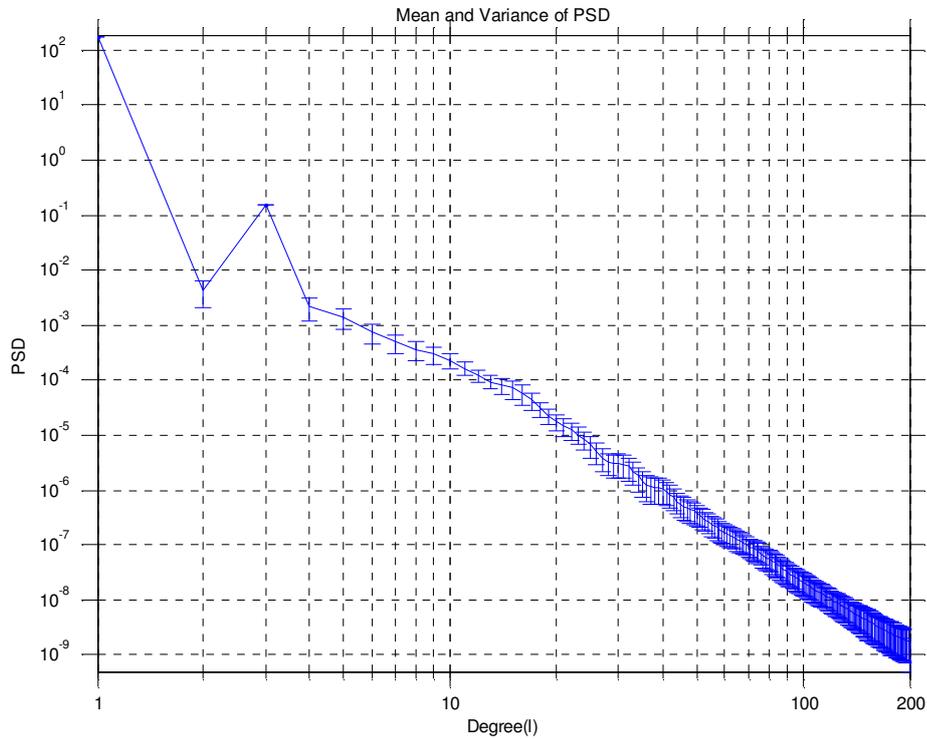


Figure 4.44 Mean and Variance of PSD

As can be observed on Figure 4.44, there is a large variance in the PSD coefficients. However,  $\text{PSD}(l=0)$  is nearly  $10^{10}$  times than  $\text{PSD}(l=200)$  and also the variance to mean ratio of  $\text{PSD}(l=0)$  is about 170 time smaller than the variance to mean ratio of  $\text{PSD}(l=200)$  as can be observed on Table 4.4.

Table 4.4 Mean and Variance of the PSD

Degree (l)	Mean of PSD	Variance of PSD	Variance/Mean Ratio
1	171,663308802358	0,717854922017014	0,00418176095419147
2	0,00759448400416351	0,00849890305109356	1,11908893960857
3	0,155566383038054	0,00640403133050346	0,0411659074758905
4	0,00369074399354123	0,00395971727794203	1,07287779506557
5	0,00210192046425806	0,00197747446795526	0,940794145916115
6	0,000970339568823109	0,000775902880011981	0,799619952583245
7	0,000478212175396498	0,000274085213339452	0,573145619122309
8	0,000319322437525565	0,000164054645009332	0,513758589219706
9	0,000292003177696955	0,000161184156018881	0,551994527217645
10	0,000258342362576950	0,000157582917598658	0,609977070840329
11	0,000192114212749387	0,000118792250329638	0,618341811517105
12	0,000124830246393072	6,53355300085602e-05	0,523395025616055
....	...	...	...

99	2,15592622398305e-08	1,31785628417568e-08	0,611271512686994
100	2,02623696122891e-08	1,21226476950376e-08	0,598283810186015
101	1,91310258328951e-08	1,12643084449230e-08	0,588797931868053
102	1,84289106892547e-08	1,09182655559561e-08	0,592453115653881
103	1,79972287659462e-08	1,07773723699043e-08	0,598835104563259
104	1,74016902844874e-08	1,03454354037475e-08	0,594507500973618
105	1,64309900675562e-08	9,46425663841730e-09	0,576000387043320
106	1,53569679420491e-08	8,50702582414719e-09	0,553952177034505
107	1,46361807829815e-08	7,99346672499309e-09	0,546144301134051
108	1,44143897018970e-08	8,08196627084592e-09	0,560687371299687
109	1,43810160478563e-08	8,39166742867458e-09	0,583523959694452
110	1,41006561977949e-08	8,38409607361470e-09	0,594589071317536
111	1,34578610406013e-08	7,86536884432891e-09	0,584444201095530
112	1,27174884854230e-08	7,17134955957408e-09	0,563896681942667
...	...	...	...
141	5,03789373993384e-09	2,92062636912378e-09	0,579731633871686
142	4,98533646470740e-09	2,95063894184922e-09	0,591863550782904
143	4,88198543241573e-09	2,93628968724718e-09	0,601454004297230
144	4,70923041840174e-09	2,82797960136707e-09	0,600518418108506
145	4,52498284899804e-09	2,71768371108810e-09	0,600595361745929
146	4,38602118663799e-09	2,65991876717806e-09	0,606453697780007
147	4,30331859929520e-09	2,64094321140054e-09	0,613699206894203
148	4,24211971889210e-09	2,64968674461776e-09	0,624613853498166
149	4,16431819550519e-09	2,63668240887441e-09	0,633160648415470
150	4,05558852906255e-09	2,58488830252815e-09	0,637364536368694
151	3,93431654196940e-09	2,50897401290699e-09	0,637715340426337
...			
...			
189	1,88531736070677e-09	1,32725888257927e-09	0,703997592257733
190	1,87514504722884e-09	1,33870760028206e-09	0,713922158854033
191	1,86547150915079e-09	1,34606444567003e-09	0,721567946262976
192	1,84550557925945e-09	1,33862082142314e-09	0,725340977815030
193	1,80712803998170e-09	1,30847808788622e-09	0,724064957732309
194	1,76015325308292e-09	1,26911261901594e-09	0,721023931747467
195	1,74316036533116e-09	1,27129316962950e-09	0,729303622841372
196	1,75581909356178e-09	1,31766024837717e-09	0,750453308776943
197	1,75382151747726e-09	1,32910542397300e-09	0,757833913387502
198	1,77666355871154e-09	1,32764056723988e-09	0,747266166815908
199	1,74023377226216e-09	1,29255735519203e-09	0,742749264951810
200	1,17946138381183e-09	8,55339034677712e-10	0,725194606976780

The previous table tells us that a transformation of PSD coefficients is needed, because there are bands with a large mean but a low variance. This means that while computing the distance between the faces, the impact of those bands on the distance computation will be large, although those bands are less discriminative because of low variance. As in previous chapter a whitening transform is applied to the PSD.

The distance between two faces using PSD is defined as follows;

$$D_{i,j} = \sum_{l=0}^{L_{max}} |PSD_i(l) - PSD_j(l)| \quad (4.10)$$

A distance table for 4 samples faces similar to the one in the previous chapter is given in Table 4.5.

Table 4.5 Face distances using PSD

	Face1	Face2	Face3	Face4
Face1	0	0.27	1.14	2.18
Face2	0.27	0	0.87	1.91
Face3	1.14	0.87	0	1.02
Face4	2.18	1.91	1.02	0

### 4.5.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) has been a popular method for extracting features which preserve class separability. The projection functions of LDA are commonly obtained by maximizing the between class covariance and simultaneously minimizing the within class covariance. In this study, LDA is applied for face classification. Two different feature vectors are used for LDA based classification.

- Use SHT coefficients as feature vector,
- Use PSD of SHT coefficients as feature vector.

The problem with the first approach is that the dimension of the covariance matrices is very large. For  $L_{max}=200$ , the dimension of the feature vector is  $200 \times 201 = 40200$ , and hence the covariance matrix would have a dimension of  $40200 \times 40200$ . The dimension of the feature vector is halved by combining the SHT coefficient for a particular order.  $m$  and  $-m$ , i.e. the definition of the SHT coefficient  $C(l, m)$  is modified as follows;

$$C'(l, m)_{\substack{0 \leq l \leq L_{max} \\ 0 \leq m \leq l}} = |C(l, m)| + |C(l, -m)| \quad (4.11)$$

In this case, there are  $\frac{(L_{max}+1)(L_{max}+2)}{2}$  coefficients. For  $L_{max} = 200$ , there are 20301 coefficients. The dimension of the covariance matrix is still very large. The computation of LDA involves dense matrices eigendecomposition which can be computationally expensive both in time and memory. Specifically, LDA has  $O(mnt+t^3)$  time complexity and requires  $O(mn + mt + nt)$  memory, where  $m$  is the number of samples,  $n$  is the number of features and  $t = \min(m, n)$ . When both  $m$  and  $n$  are large, it is infeasible to apply LDA. The method described in [85] is applied for large scale discriminant analysis. In [85], it is reported that LDA can be computed with  $O(ms)$  time and  $O(ms)$  memory, where  $s \leq n$  is the average number of non-zero features in each sample. This is a noticeable improvement over standard LDA computation method using eigenvalue decomposition. After LDA based dimension reduction, multi-class linear classifier using perceptron algorithm is applied for further classification.

#### 4.5.4 Support Vector Machines

Another method we applied for face classification is Support Vector Machines. Again we will use two different feature vectors as input to the SVM method. We will use the whole combined SHT coefficients and PSD coefficients as input feature vectors to SVM.

In this study, LIBSVM in [72] is used for SVM based classification. Main features of this library are as follows:

- Different SVM formulations
- Efficient multi-class classification
- Cross validation for model selection
- Probability estimates
- Weighted SVM for unbalanced data
- Both C++ and Java sources

We've used Java sources of this library. The implementation details of this library are given in [72].

SVM requires that the input feature vector is scaled so that it is zero mean and unit variance. The main advantage of scaling is to avoid attributes (features) in greater numeric ranges dominate those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation.

The documentation for LIBSVM suggests that in general Radial Basis Functions (RBF) kernel is a reasonable choice. The RBF kernel nonlinearly maps samples into a higher dimensional space, so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel is a special case of RBF as shows that the linear kernel with a penalty parameter  $C$  has the same performance as the RBF kernel with some parameters  $(C, \gamma)$ . In addition, the sigmoid kernel behaves like RBF for certain parameters. The second reason is the number of hyperparameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel. Finally, the RBF kernel has less numerical difficulties.

Like all other linear classifiers, SVM is also a binary classifier which outputs only for two-class problems. In order to apply SVM for multi-class problems as in the face recognition case, One-Against-All decomposition rule is implemented in this study, where there is a separate SVM discriminant function for every class in the gallery.

#### **4.6 Confusion Matrix**

A confusion matrix is a visualization tool typically used in supervised learning. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes [36].

A sample confusion matrix is given in Table 4.6. This table is calculated according to the method described in [86]. As can be seen on this table, a sample set of 67 faces is used for verification. In this case a set of 200 faces is used as gallery faces. The classification method based on Euclidean distance is used to generate the table. Instance scores are calculated as an average of true positives to false positives for each class label. Notice that, the table 4.6 corresponds to a point on the ROC curve, which will be given in next chapter.

Table 4.6 A sample confusion matrix

		Actual Value		total
		p	N	
Output	p'	True Positives=6	False Positives=2	P'=8
	n'	False Negatives=1	True Negatives=58	N'=59
total		P=7	N=60	

Using Table 4.6, False Acceptance Rate (False Positive Rate), or probability of identification can be calculated as follows;

$$FAR = \frac{FP}{N} = \frac{2}{60} = 0.033$$

$$\text{Probability of Identification} = \frac{TP}{P} = \frac{6}{7} = 0.857$$

In the above confusion experiment, the persons which are confused by the classifier are depicted in Figure 4.45.



Figure 4.45 Confused subjects

The two confused subjects have 8 pictures in the database. The PSD plot of the SHT coefficients of the all 8 pictures are given in Figure 4.46. As can be seen on the figure, the PSD of the two persons overlap at some regions, and are very close to each other and also they have very similar patterns. The reason why the Euclidean based classifier can be seen clearly.

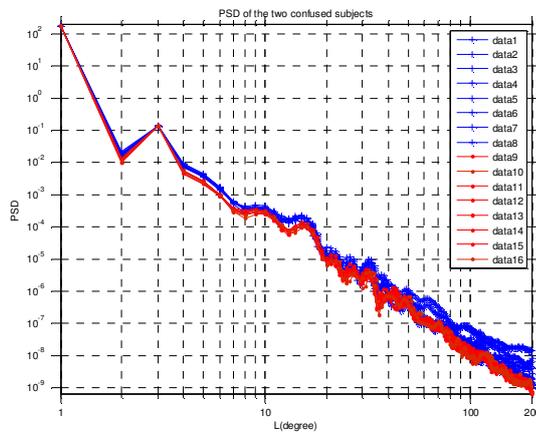


Figure 4.46 PSD plot of images of confused subjects

The classification still can be performed using LDA. For the two confused persons, an LDA can be applied for dimension reduction. Since there are only 2-classes, the reduced dimension will be only 1. The 1-D projection of the SHT coefficients can be seen on Figure 4.47. As it can be observed on this figure, a simple linear classifier is now able to classify the two persons.

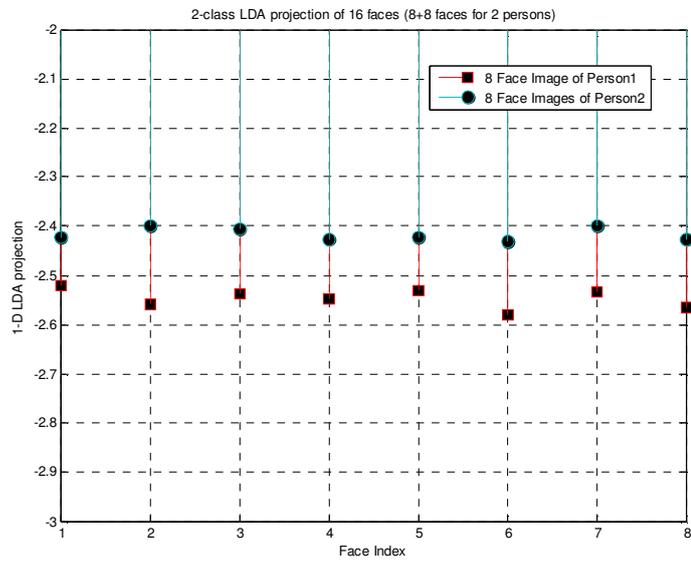


Figure 4.47 1-D LDA projection of the faces of confused subjects

# CHAPTER 5

## EXPERIMENTS

### 5.1 Face Database

The FRGC v1 database contains 943 2D images and 943 3D range data belonging to 275 different persons. People have different number of images in the database. The numbers are given in Table 5.1.

Table 5.1 FRGC 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

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. There are also some badly registered pairs in the database, that is to say, there are small translation differences about 10 pixels between the color image and depth image. These pairs are also removed from the database. There are also some face range images where the eyes have wrong depth values. Even, gap filling and noise filtering (spike removal) at the preprocessing step cannot correct these wrong values. These faces are also removed from the face. Another problem with some of the face images was the distance of the face to the 3D sensor. This caused large spherical variations in face data, hence these faces are also removed. All the experiments are carried out by the remaining 907 pairs. Notice

that, this count is different than the faces used in experiments in [19]. However, this deviation is low compared to the number of faces, and can be neglected.

In order to compare the efficiency of the proposed 3D face matching method, four  $T_n$  experiments are held where  $n$  denotes the number of images per person used in the training set (Table 5.2).

Table 5.2 Images used in experiments

# of images per person	# of persons (or classes)	# of training images	# of test images
1	197	197	666
2	165	330	469
3	119	357	304
4	86	344	185

To illustrate, for  $T_4$  experiments, all 86 people having more than 4 images in the database are used. The first 4 images from each person are used in the training step. In the test set, the rest of all the faces belonging to the persons participated in the training step are used.

All the experiments are performed for the following three methods:

- k-Nearest Neighbor with  $k=1$  using Euclidean distances of all SHT coefficients (labeled as SHT-EUC)
- k-Nearest Neighbor with  $k=1$  using Euclidean distance of PSD of SHT coefficients (labeled as PSD-EUC)
- Mahalanobis Distance of SHT coefficients (labeled as SHT-MAHA)
- Mahalanobis Distance of PSD of SHT coefficients (labeled as PSD-MAHA)
- Linear Discriminant Analysis (LDA) using SHT coefficients (labeled as SHT-LDA)
- Linear Discriminant Analysis (LDA) using PSD of SHT coefficients (labeled as PSD-LDA)

- Support Vector Machines using SHT coefficients as feature vector (labeled as SHT-SVM)
- Support Vector Machines using PSD of SHT coefficients as feature vector (labeled as PSD-SVM)

CMC curves of these methods are given in comparison with the results of the work in [19]. In [19], the effect of registration process is evaluated for 3D face recognition. ICP and some variants of ICP are implemented for 3D face registration, which is followed by 4 recognition algorithms, namely, Eigenface, Fisherface, Non-negative Matrix Factorization (NMF) and Independent Component Analysis (ICA).

## 5.2 Results

### 5.2.1 T1 Experiments

CMC curve for experiment T1 is given in Figure 5.1. Match scores for experiment T1 are given in Table 5.3.

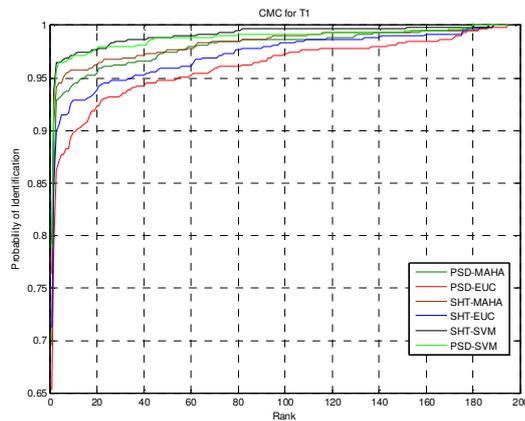


Figure 5.1 CMC curve for experiment T1

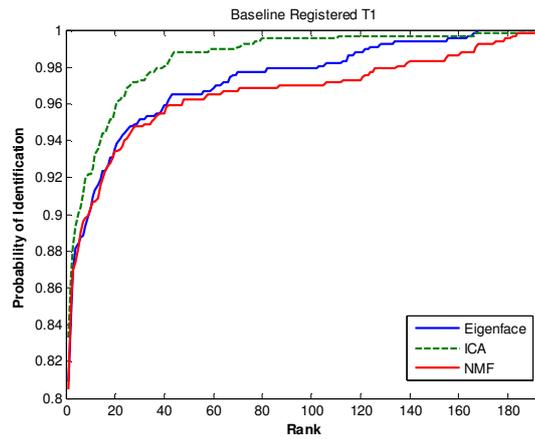


Figure 5.2 CMC curve for T1 from [19]

Table 5.3 Top-10 Rank scores for experiment T1

Rank	SHT-EUC	PSD-EUC	SHT-MAHA	PSD-MAHA	SHT-LDA	PSD-LDA	SHT-SVM	PSD-SVM
1	0,594	0,583	0,594	0,622	n/a	n/a	0,792	0,787
2	0,740	0,681	0,697	0,768	n/a	n/a	0,938	0,923
3	0,748	0,681	0,709	0,783	n/a	n/a	0,964	0,957
4	0,756	0,697	0,717	0,799	n/a	n/a	0,964	0,964
5	0,760	0,713	0,720	0,811	n/a	n/a	0,965	0,964
6	0,768	0,728	0,724	0,827	n/a	n/a	0,967	0,964
7	0,772	0,736	0,724	0,831	n/a	n/a	0,969	0,965
8	0,772	0,744	0,724	0,831	n/a	n/a	0,969	0,967
9	0,780	0,756	0,732	0,839	n/a	n/a	0,971	0,967
10	0,783	0,760	0,736	0,843	n/a	n/a	0,971	0,971

## 5.2.2 T2 Experiments

CMC curve for experiment T2 is given in Figure 5.3. Match scores for experiment T2 are given in Table 5.4.

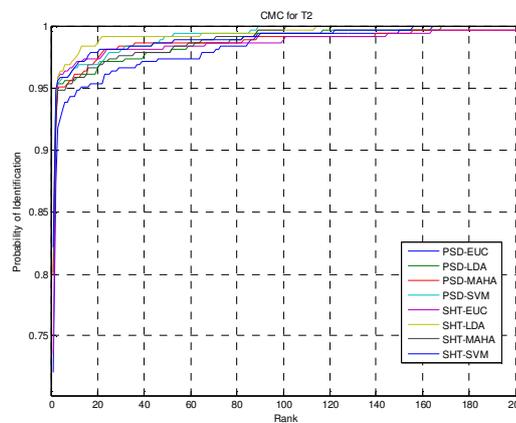


Figure 5.3 CMC Curve for Experiment T2

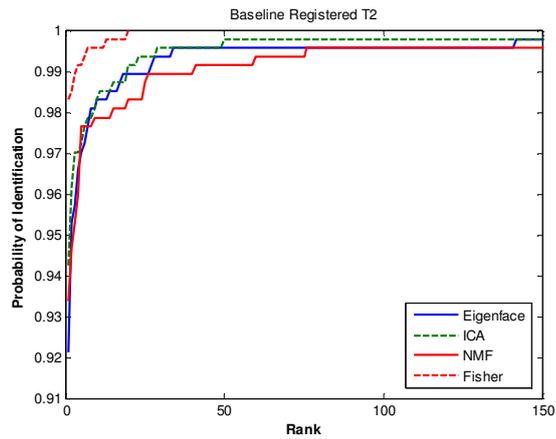


Figure 5.4 CMC Curve from [19]

Table 5.4 Top-10 Rank scores for experiment T2

Rank	SHT-EUC	PSD-EUC	SHT-MAHA	PSD-MAHA	SHT-LDA	PSD-LDA	SHT-SVM	PSD-SVM
1	0,735	0,720	0,811	0,798	0,824	0,801	0,821	0,819
2	0,918	0,872	0,921	0,928	0,944	0,926	0,949	0,944
3	0,959	0,918	0,949	0,951	0,959	0,954	0,956	0,954
4	0,961	0,926	0,949	0,951	0,964	0,954	0,959	0,956
5	0,961	0,933	0,949	0,951	0,964	0,954	0,959	0,959
6	0,964	0,938	0,949	0,951	0,969	0,956	0,959	0,959
7	0,964	0,938	0,954	0,954	0,969	0,956	0,959	0,959
8	0,966	0,944	0,954	0,956	0,969	0,956	0,961	0,961
9	0,966	0,944	0,954	0,956	0,972	0,956	0,964	0,964
10	0,969	0,944	0,959	0,961	0,974	0,956	0,966	0,966

### 5.2.3 T3 Experiments

CMC curve for experiment T3 is given in Figure 5.5. Match scores for experiment T3 are given in Table 5.5.

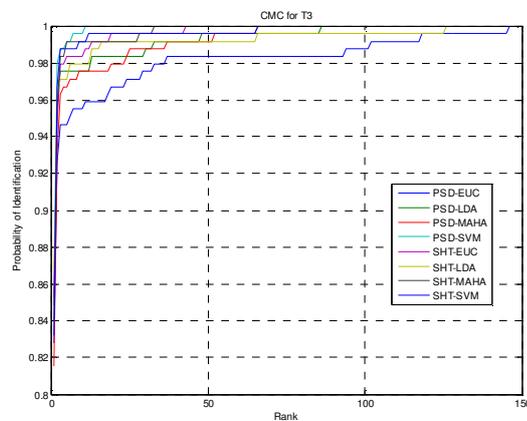


Figure 5.5 CMC curve for experiment T3

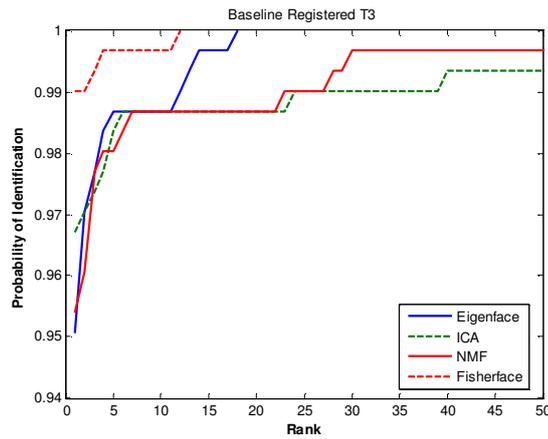


Figure 5.6 CMC curve from [19]

Table 5.5 Top-10 Rank scores for experiment T3

Rank	SHT-EUC	PSD-EUC	SHT-MAHA	PSD-MAHA	SHT-LDA	PSD-LDA	SHT-SVM	PSD-SVM
1	0,852	0,827	0,840	0,815	0,868	0,848	0,852	0,831
2	0,926	0,876	0,934	0,930	0,963	0,959	0,979	0,967
3	0,939	0,913	0,943	0,963	0,981	0,975	0,987	0,987
4	0,941	0,925	0,963	0,967	0,991	0,975	0,987	0,987
5	0,953	0,932	0,971	0,967	0,991	0,975	0,991	0,987
6	0,967	0,940	0,971	0,971	0,999	0,975	0,991	0,987
7	0,971	0,954	0,981	0,971	0,999	0,975	0,995	0,987
8	0,978	0,954	0,983	0,971	0,999	0,975	0,995	0,987
9	0,983	0,954	0,991	0,975	0,999	0,975	0,995	0,991
10	0,985	0,954	0,991	0,975	0,999	0,975	0,995	0,991

## 5.2.4 T4 Experiments

CMC curve for experiment T4 is given in Figure 5.7. Top match scores for experiment T4 are given in Table 5.6.

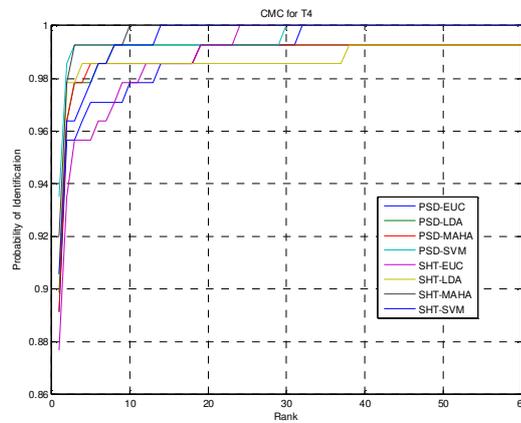


Figure 5.7 CMC curve for experiment T4

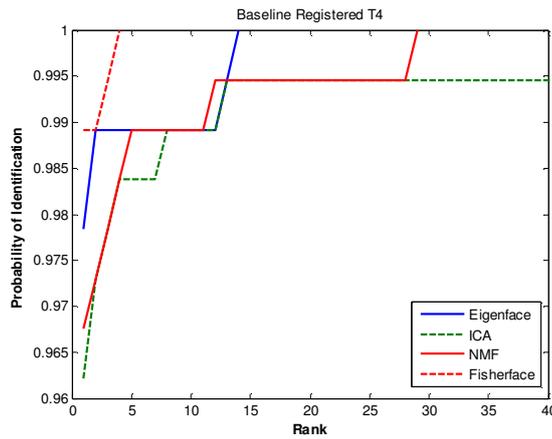


Figure 5.8 CMC Curve from [19]

Table 5.6 Top-10 Rank scores for experiment T4

Rank	SHT-EUC	PSD-EUC	SHT-MAHA	PSD-MAHA	SHT-LDA	PSD-LDA	SHT-SVM	PSD-SVM
1	0,891	0,876	0,920	0,891	0,905	0,898	0,934	0,905
2	0,956	0,934	0,956	0,963	0,978	0,963	0,985	0,963
3	0,956	0,956	0,972	0,978	0,978	0,978	0,992	0,963
4	0,963	0,956	0,987	0,978	0,985	0,978	0,992	0,971
5	0,971	0,956	0,987	0,985	0,985	0,978	0,992	0,978
6	0,971	0,963	0,987	0,985	0,985	0,985	0,992	0,985
7	0,971	0,963	0,991	0,985	0,985	0,985	0,992	0,985
8	0,971	0,971	0,991	0,985	0,985	0,992	0,992	0,992
9	0,971	0,978	0,992	0,985	0,985	0,992	0,992	0,992
10	0,978	0,978	0,992	0,985	0,985	0,992	0,992	0,992

### 5.3 ROC Analysis

In the case of a verification scenario, the False Acceptance Rate (FAR), False Rejection Rate (FRR), and Equal Error Rate are summarized in the ROC curve in Figure 5.9. The curve is generated using the method described in [86]. As can be seen on the figure, SVM based classification achieves lower EER than the other methods. The ROC curves for Euclidean based, LDA based and SVM based methods which use SHT coefficients as feature vectors are given in the figure.

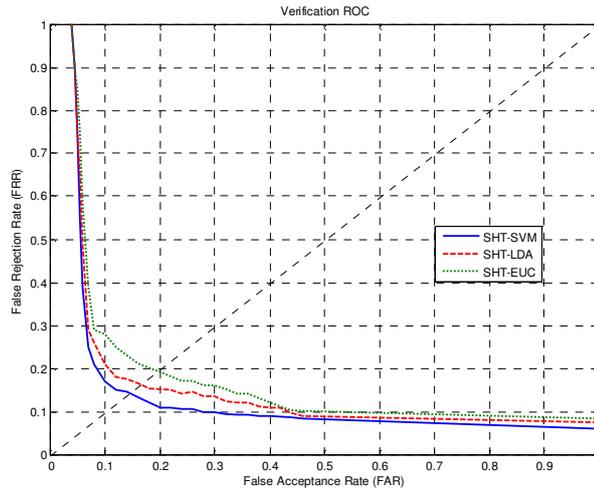


Figure 5.9 Verification ROC

The equal error rate (EER) for SVM based method is about 0.15, for LDA based method 0.17 and for Euclidean based method about 0.195.

#### 5.4 Results from Literature for Comparison

There are several published algorithms on face recognition algorithms. A robust comparison of the efficiency of these all algorithms should be based on a common face database. The face database used throughout this study is the version 1 of FRGC database [12], [87]. Actually, this database is a subset of the FRGC version 2 database. Philips et al. ([87]) has selected a PCA based method as the baseline algorithm in order to provide a minimum level of performance. The rank-1 score for the baseline algorithm for 1-Gallery, 1-Probe mode is given as 88.9%, and an EER score of 0.05 is published,

## CHAPTER 6

### CONCLUSION

In this study, a 3D face matching method using face range image data is proposed. The proposed method is implemented using Java and MATLAB. The efficiency of the proposed method is listed and compared with other methods in the previous chapter. As can be seen on the CMC and ROC plots in the previous chapter, our method gives very close results to the well established methods, such as Eigenface, Fisherface, ICA and LDA, compared to the results given in [19].

It is obvious that the 3D registration method proposed in this study which is based on the face geometry is easier to implement than ICP. ICP requires to preprocess the whole gallery faces in order to build up a model face by averaging the faces. And also ICP tries to register faces to the model face by rigid transformation using an iterative approach, which is computational very expensive.

Another issue which is worthwhile to mention is that the proposed method is open to improvements. For example, a better face to ellipsoid stitching method can be developed using higher order surface characteristics. In this study, linear interpolation is used to stitch the face surface to the ellipsoid which ensures first order (geometric) continuity. However, using a cubic interpolation would make the stitched band smoother which would result in better SNR values for SHT. And hence classification would improve.

As noted earlier, spherical harmonics based classification is not new to the face recognition community. Bronstein et al. [3] and Iordanis et al. [62] used

spherical harmonics decomposition of the so called ‘canonical images’ for face classification. However, the experiments they conducted in their publications are not comparable with the experiments held in this study. For example, Bronstein et al. held their experiments with a data-set of 104 faces consisted of 4 subjects. So, it is not feasible to compare their method with the one proposed in this study, since a quantitative comparison could not be given.

The spherical bandwidth  $L_{max}$  is another important factor which affects the efficiency of the proposed recognition method. In this study, observing the SNR vs. bandwidth plot, a bandwidth of 200 is chosen practically. Actually, the decision is mostly due to computation time required for the spherical harmonics transform. Although FFT is used for fast spherical transformation, it is still computationally not feasible to use a higher bandwidth. However, it is obvious that higher match scores could be obtained when higher bandwidth was chosen.

The results of the experiments given in the previous chapter make it clear that the methods based on LDA and SVM are slightly better than the one based on a distance metric. LDA is a powerful tool for dimension reduction where it projects the samples onto a lower dimension while it maximizes the probability of class separation in lower dimensions. However, it is also a fact that there are very few samples per class compared to the dimension of the feature vector. For example, for  $L_{max}=200$ , the dimension of the feature vector is  $201 \times 202 / 2 = 20301$  which is very high compared to the number of the samples  $\sim 900$  consisted of  $\sim 200$  subjects (see Table 5.1). Although it is expected that LDA and SVM based methods would give much better results, the abovementioned numbers make it practically impossible.

## **6.1 Future Study**

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, a new approach is proposed and implemented.

In this study, only the 3D range image of the face is used. For better recognition performance, the registered texture image should also be fused with the 3D range image. This is called multi-modal classification. Bowyer et al. ([13]) listed some conjectures about 3D range based and multi-modal classification. These are;

- The shape channel of one 3D image is more powerful for face recognition than one 2D image.
- Multimodal 2D+3D face recognition performs significantly better than using either 3D or 2D alone.
- Combining results from two or more 2D images using a similar fusion scheme as used in multimodal 2D+3D also improves performance over using a single 2D image.

As discussed above, the registration algorithm used in this study was a geometric one and was computationally more efficient. However, most of the published 3D face matching methods in the literature used ICP for registration. ICP seems to be a very efficient method for registering two 3D shapes. However convergence to global minimum is still cannot be ensured. In terms of face registration, instead of using a geometric one variants of ICP can be examined for 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.

In this study, the proposed 3D face recognition method used 2.5D face range images from FRGC v1 database. The method relies on stitching and mapping the range image to an ellipsoid in order to make the face surface star-shaped which is a requirement of spherical harmonics transform. However, instead of range images, if full-view 3D face meshes generated by a 3D scanner were used as input, better classification results would be obtained, since spherical harmonics based 3D shape matching gives very good results as discussed in Chapter 2.

Finally, it is expected that a solid face recognition system should be able to handle 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. The proposed method cannot handle with facial expressions. However, the spherical mapping part of the proposed method can be extended to overcome the facial expressions by mapping parts of the faces locally.

## REFERENCES

- [1] Ashbourn, J.: 2002, *Biometrics: advanced identity verification*. Springer-Verlag, Berlin Heidelberg New York.
- [2] J. Gonzalez-Rodriguez, J. Ortega-Garcia, and J.-L. Sanchez-Bote, "Forensic Identification Reporting using Automatic Biometric Systems", Chapter 7 in *Biometrics Solutions for Authentication in an E-World*, (D. Zhang, ed.), vol. 697, pp. 169-185, Kluwer Academic Publishers, July 2002. ISBN 1-4020-7142-6.
- [3] A. M. Bronstein, M. M. Bronstein and R. Kimmel . Three-Dimensional Face Recognition. *International Journal of Computer Vision* archive Volume 64 , Issue 1 (August 2005) Pages: 5 - 30 2005-ISSN:0920-5691
- [4] Face Recognition Vendor Test, <http://www.frvt.org> , last accessed: 07.08.2008
- [5] P. Papadakis, et al., Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation, *Pattern Recognition* (2007), doi: 10.1016/j.patcog.2006.12.026
- [6] S. Ertürk and T.J. Dennis, 3D Model Representation using Spherical Harmonics, *Electronics Letters*. 22nd May 1997 Vol. 33 No. 11
- [7] J. Fehr, H. Burkhardt, Harmonic Shape Histograms for 3D Shape Classification and Retrieval MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN
- [8] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz, Eurographics Symposium on Geometry Processing (2003) Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors
- [9] A. M. Bronstein, M. M. Bronstein and R. Kimmel, "Expression-invariant 3D face recognition," in *Proceedings of International Conference on Audio- and Video-based Biometric Person Authentication, Lecture Notes in Computer Science* **2688**, 62–70 (2003).
- [10] Pose-Encoded Spherical Harmonics for Face Recognition and Synthesis Using a Single Image Zhanfeng Yue,<sup>1</sup> Wenyi Zhao,<sup>2</sup> and Rama Chellappa<sup>1</sup>

- [11] Perronnin, F., Junqua, J. C., and Dugelay, J. L., Biometrics Person Authentication: From Theory to Practice, EURASIP Journal on applied Signal processing, 16 (1): 2--19, Mars 2005.
- [12] Phillips, P. J., P. Grother, R. J. Michaels, D. M. Blackburn, E. Tabassi, and J. Bone: 2003, 'FRVT 2002: overview and summary'. Online. Available at [www.frvt.org](http://www.frvt.org).
- [13] K. Bowyer, K. Chang, and P. Flynn. A Survey of Approaches and Challenges in 3D and Multi-Modal 3D+2D Face Recognition. In *CVIU*, volume 101, pages 1–15, 2006.
- [14] 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.
- [15] Blackburn, D.M., Face Recognition 101: A Brief Primer. 2003.
- [16] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Computing Survey*, pages 399–458, 2003.
- [17] <http://www.face-rec.org/> . Last accessed 09.08.2008
- [18] Pentland, M.T.a.A., Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 1991. Volume 3(Number 1): p. 71-86.
- [19] Üstün, Bülemd. MS Thesis. 3D face recognition. Supervisor Prof. Dr. Uğur Halıcı. Publication info. Ankara : METU ; 2007.
- [20] Wyse, M.R.a.L. *Face Recognition with Non-negative Matrix Factorization*. in *Visual Communications and Image Processing 2003*. 2003.
- [21] 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).
- [22] Seung, D.D.L.a.H.S., *Algorithms for Non-negative Matrix Factorization*.
- [23] Marian Stewart Bartlett, J.R.M., Terrence J. Sejnowski, *Face Recognition by Independent Component Analysis*. IEEE TRANSACTIONS ON NEURAL NETWORKS, 2002. Vol 13(No 6).
- [24] Curt Heshner; Anuj Srivastuvu, G.E., *A NOVEL TECHNIQUE FOR FACE RECOGNITION USING RANGE IMAGING*. IEEE, 2003.

- [25] Anuj Srivastava, X.L., Curt Heshner, *Face recognition using optimal linear components of range images*. Image and Vision Computing, 2005. 24: p. 291-299.
- [26] <http://mplab.ucsd.edu/~marni/code.html>, Bartlett, M.S., Movellan, J.R., & Sejnowski, T.J. (2002). Face recognition by independent component analysis. IEEE Transactions on Neural Networks 13(6) p. 1450-64.
- [27] L. Wiskott, J.-M. Fellous, N. Krueger, C. von der Malsburg, Face Recognition by Elastic Bunch Graph Matching, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 1997, pp. 775-779
- [28] M.-H. Yang, Face Recognition Using Kernel Methods, Advances in Neural Information Processing Systems, T. Diederich, S. Becker, Z. Ghahramani, Eds., 2002, vol. 14, 8 pages
- [29] B. Scholkopf, A. Smola, K.-R. Muller, Nonlinear Component Analysis as a Kernel Eigenvalue Problem, Technical Report No. 44, December 1996, 18 pages
- [30] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face Recognition Using Kernel Direct Discriminant Analysis Algorithms, IEEE Trans. on Neural Networks, Vol. 14, No. 1, January 2003, pp. 117-126
- [31] T.F. Cootes, C.J. Taylor, Statistical Models of Appearance for Computer Vision, Technical Report, University of Manchester, 125 pages
- [32] A. Kadyrov, M. Petrou, The Trace Transform and Its Applications, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 8, August 2001, pp. 811-828
- [33] S. Srisuk, M. Petrou, W. Kurutach and A. Kadyrov, Face Authentication using the Trace Transform, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), 16-22 June 2003, Madison, Wisconsin, USA, pp. 305-312
- [34] S. Srisuk and W. Kurutach, Face Recognition using a New Texture Representation of Face Images, Proceedings of Electrical Engineering Conference, Cha-am, Thailand, 06-07 November 2003, pp. 1097-1102
- [35] B. Moghaddam, C. Nastar, A. Pentland, A Bayesian Similarity Measure for Deformable Image Matching, Image and Vision Computing, Vol. 19, Issue 5, May 2001, pp. 235-244
- [36] [www.wikipedia.com](http://www.wikipedia.com). Last accessed 10.08.2008.

- [37] David Meyer, Friedrich Leisch, and Kurt Hornik. The support vector machine under test. *Neurocomputing* 55(1-2): 169-186, 2003  
[http://dx.doi.org/10.1016/S0925-2312\(03\)00431-4](http://dx.doi.org/10.1016/S0925-2312(03)00431-4)
- [38] Corinna Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning*, Volume 20, 1995, pages 273-297.
- [39] M. Aizerman, E. Braverman, and L. Rozonoer (1964). "Theoretical foundations of the potential function method in pattern recognition learning". *Automation and Remote Control* 25: 821–837.
- [40] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144-152, Pittsburgh, PA, 1992. ACM Press
- [41] Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola and Vladimir Vapnik (1997). "Support Vector Regression Machines". *Advances in Neural Information Processing Systems 9*, NIPS 1996, 155-161, MIT Press.
- [42] Johan A.K. Suykens, Tony Van Gestel, Jos de Brabanter, Bart De Moor and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific Publishing. 2002.
- [43] G. Guo, S.Z. Li, K. Chan, *Face Recognition by Support Vector Machines*, Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition, 26-30 March 2000, Grenoble, France, pp. 196-201
- [44] B. Heisele, P. Ho, T. Poggio, *Face Recognition with Support Vector Machines: Global versus Component-based Approach*, Proc. of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vol. 2, 09-12 July 2001, Vancouver, Canada, pp. 688-694
- [45] K. Jonsson, J. Matas, J. Kittler, Y.P. Li, *Learning Support Vectors for Face Verification and Recognition*, Proc. of the IEEE International Conference
- [46] J. C. Lee and E. Milios, "Matching range images of human faces," in *Proc. IEEE International Conference on Computer Vision*, pp. 722–726 (1990).
- [47] G. G. Gordon, "Face recognition based on depth maps and surface curvature," in *Geometric Methods in Computer Vision, SPIE Proceedings*, **1570**, 234–247 (1991). *3D Face Recognition from Range Data* 591

- [48] G. G. Gordon, "Face recognition based on depth and curvature features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 808–810 (June 1992).
- [49] Y. Yacoob and L. S. Davis, "Labeling of human face components from range data," *CVGIP: Image Understanding* **60**(2), 168–178 (September 1994).
- [50] H. T. Tanaka, M. Ikeda and H. Chiaki, "Curvature-based face surface recognition using spherical correlation — principal directions for curved object recognition," in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 372–377 (1998).
- [51] B.K.P. Horn, "Extended Gaussian Images", *Proc. IEEE*, vol. 72 pp. 1671-1686, Dec. 1984
- [52] K. Ikeuti, "Recognition of 3D Objects using the Extended Gaussian Image", in. *Proc. 7<sup>th</sup> Int. Joint Conf. Artificial Intel.*, Aug 1981, pp. 595-600
- [53] C. S. Chua, F. Han and Y. K. Ho, "3D human face recognition using point signature," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 233–238 (March 2000).
- [54]. Y. Wang, C.-S. Chua and Y.-K. Ho, "Facial feature detection and face recognition from 2D and 3D images," *Pattern Recognition Letters* **23**, 1191–1202 (2002).
- [55]. G. Pan, Y. Wang and Z. Wu, "Pose-invariant detection of facial features from range data," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Hyatt Regency, Washington D.C., pp. 4171–4175 (2003).
- [56] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(9), 1063–1074 (2003).
- [57] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Computer Graphics Proceedings SIGGRAPH'99*, pp. 187–194 (1999).
- [58] M. W. Lee and S. Ranganath, "Pose-invariant face recognition using a 3D deformable model," *Pattern Recognition* **36**(8), 1835–1846 (2003).

- [59] W. Zhao and R. Chellappa, "Illumination-insensitive face recognition using symmetric shape-from-shading," in *Proc. IEEE International Conference on Computer Vision*, pp. 1286–1293 (2000).
- [60] C. Beumier and M. Acheroy, "Automatic 3D face authentication," *Image and Vision Computing* **18**(4), 315–321 (2000).
- [61] Y. Lee, K. Park, J. Shim and T. Yi, "3D face recognition using statistical multiple features for the local depth information," in *Proc. 16th International Conference on Vision Interface*, Halifax, Canada (2003).
- [62] Iordanis Mpipieris, Sotiris Malassiotis, and Michael G. Strintzis. 3-D Face Recognition With the Geodesic Polar Representation. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 2, NO. 3, SEPTEMBER 2007
- [63] Xiaoguang Lu, Dirk Colbry, and Anil K. Jain. Three-Dimensional Model Based Face Recognition. Proc. of ICPR, August 2004
- [64] Richard Duda, Peter Hart, David Stork. Pattern Classification. John Wiley&Sons Inc. 2001
- [65] Peter-Pike Sloan, Jan Kautz, and John Snyder , Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments, ACM-SIGGRAPH 2002, July, 2002
- [66] Ravi Ramamoorthi, Modeling Illumination Variation with Spherical Harmonics Columbia University, Research Report, Email: ravir@cs.columbia.edu
- [67] R. Basri and D. Jacobs. Illumination Modeling for Face Recognition, chapter 5, Face Recognition Handbook. Springer Verlag, 2004.
- [68] L. Zhang and D. Samaras, "Face recognition under variable lighting using harmonic image exemplars," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03), vol. 1, pp. 19–25, Madison, Wis, USA, June 2003.
- [69] Wang, J.; Oliveira, M.M. A hole-filling strategy for reconstruction of smooth surfaces in range images. Computer Graphics and Image Processing, 2003. SIBGRAPI 2003. XVI Brazilian Symposium on Volume , Issue , 12-15 Oct. 2003 Page(s): 11 – 18
- [70] Face detection: A survey E Hjelm, BK Low - Computer Vision and Image Understanding, 2001

- [71] Christopher Bishop. Pattern Recognition and Machine Learning, Springer Science&Business, LLC, 2006
- [72] <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> . Last accessed 12.08.2008
- [73] McKay, P.J.B.a.N.D., A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. Vol.14(2): p. 239-256.
- [74] Xiaoguang Lu, A.K.J., and Dirk Colbry, Matching 2.5D Face Scans to 3D Models. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2005. Vol.28(No.1): p. 31-43.
- [75] Ajmal Mian, M.B., and Robyn Owens, 2D and 3D Multimodal Hybrid Face Recognition. ECCV 2006. Part III(LNCS 3953): p. 344-355.
- [76] A. Lorusso, D.W.E.a.R.B.F. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. in British Machine Vision Conference.
- [77] Levoy, S.R.a.M., Efficient Variants of the ICP Algorithm. IEEE, 2001: p. 145-152.
- [78] Takeshi Masuda, K.S., Naokazu Yokoya. Registration and Integration of Multiple Range Images for 3-D Model Constructioin. in ICPR'96. 1996.
- [79] S. Weik, Registration of 3-D Partial Surface Models Using Luminance and Depth Information. IEEE, 1997.
- [80] Medioni, Y.C.a.G. Object Modeling by Registration of Multiple Range Images. in IEEE International Conference on Robotics and Automation. 1991. Sacramento, California.
- [81] 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).
- [82] 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).
- [83] <http://math.nist.gov/javanumerics/jama/> . Last accessed 20.08.2008
- [84] Kyong I. Chang, Kevin W. Bowyer, and Patrick J. Flynn, "An evaluation of multi-modal 2d+3d face biometrics", IEEE Trans. PAMI, vol. 27, no. 4, pp. 619–624, 2005.

- [85] Deng Cai, Xiaofei He, and Jiawei Han. "Spectral Regression for Dimensionality Reduction", Department of Computer Science Technical Report No. 2856, University of Illinois at Urbana-Champaign (UIUCDCS-R-2007-2856), May 2007.
  
- [86] Tom Fawcett . An introduction to ROC analysis. Pattern Recognition Letters archive Volume 27 , Issue 8 (June 2006) Special issue: ROC analysis in pattern recognition. Pages 861-874 Year of Publication: 2006 ISSN:0167-8655.
  
- [87] P. Jonathon Phillips, Patrick J. Flynn, Todd Scruggs, Kevin W. Bowyer Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, William Worek.. Overview of the Face Recognition Grand Challenge. IEEE Conference on Computer Vision and Pattern Recognition 2005.

## APPENDIX A

### ITERATIVE CLOSEST POINT

The iterative closest point algorithm (ICP) is designed to fit points in a data set to points in a model set. The ultimate goal of the algorithm is to minimize the mean square error (MSE) with respect to the closest data points and their corresponding model points. It is important that an initial estimate is made regarding where the overlay of the two point sets should be. An appropriate transformation should be applied based on this initial estimate to align the point sets coarsely before ICP is applied. The base component of the algorithm calculates the smallest distance between each point in the data set to a point in the model set. These calculated points are then used to form a translation and rotation matrix that is applied over all points in the data set to adjust them towards the model set. This process is repeated numerous times, thus an iterative algorithm, with the end result being a data set with points that are within a specified squared error distance of their corresponding points in the model set. 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 A.1)

$$Data_{registered} = R * Data_{nonregistered} + T \quad (A.1)$$

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 A.1) and calculates the rotation and translation matrices iteratively. The algorithm repeats the following steps, until

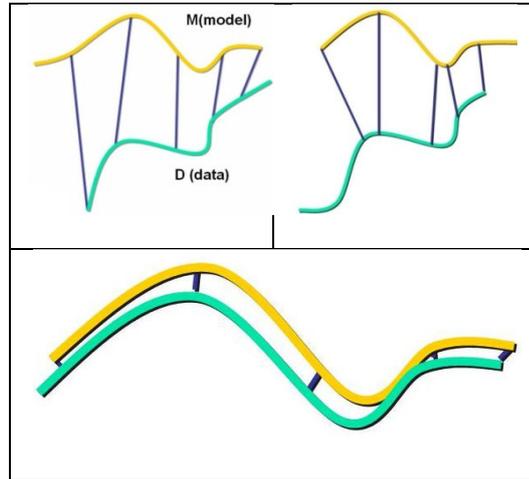


Figure A.1 ICP

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 sets, data and model set, are close enough.

ICP always converges monotonically to a local minimum [73] 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 [74, 75].

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 [76] 4 registration algorithms are evaluated and Singular Value Decomposition (SVD) has found to be better in general.

As can be seen in Figure A.1(top left), if the correct correspondences are known, which is not possible in general, registration can be handled without any problems. ICP considers the nearest points in the model (top-right). If starting point is good, the registration will give a good result (Figure A.1, bottom image).

## APPENDIX B

### NONNEGATIVE MATRIX FACTORIZATION (NMF)

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

$$V \approx W * H \quad (\text{B.1})$$

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

$$E(W, H) = \|V - WH\|^2 \quad (\text{B.2})$$

Convergence is assured if  $W$  and  $H$  are updated according to the equations below [22]:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}} \text{ where } a = 1, \dots, r \text{ and } \mu = 1, \dots, m \quad (\text{B.3})$$

and

$$W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}} \text{ where } i = 1, \dots, n \text{ and } a = 1, \dots, r \quad (\text{B.4})$$

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 B.1). This column is used to determine the similarity between a test image and a training image.

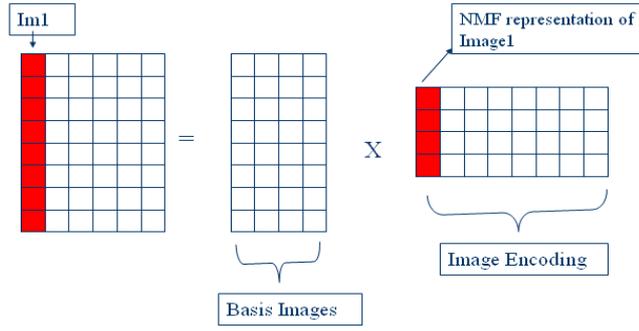


Figure B.1 NMF factorization

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.

$$Weight_{Training} = H(:, Training) \text{ (corresponding column of } H) \quad (B.5)$$

$$Weight_{Test} = pinv(W) * PCA(Testimage) \quad (B.6)$$

$$\alpha_{(Test, Training)} = \frac{dot(Weight_{Test}, Weight_{Training})}{\|Weight_{Test}\| * \|Weight_{Training}\|} \quad (B.7)$$

## APPENDIX C

### KERNEL PCA

Given a set of  $m$  centered (zero mean, unit variance) sample  $x_k, x_k=[x_{k1}, \dots, x_{kn}]^T$  in  $R^n$ , PCA aims to find the projection directions that maximize the variance,  $C$ , which is equivalent to finding the eigenvalues from the covariance matrix  $Cw=\lambda w$  for eigenvalues  $\lambda>0$  and eigenvectors  $w$  in  $R^n$ . In Kernel PCA, each vector  $x$  is projected from the input space,  $R^n$ , to a high dimensional feature space,  $R^f$ , by a non-linear mapping function  $\Phi: R^n \rightarrow R^f, f \gg n$ . Note that the dimensionality of the feature space can be arbitrarily large. In  $R^f$ , the corresponding eigenvalue problem is

$$\lambda w^\Phi = C^\Phi w^\Phi \quad (C.1)$$

where  $C^\Phi$  is a covariance matrix. All solutions  $w^\Phi$  with  $\lambda \neq 0$  lie in the span of  $\Phi(x_1), \dots, \Phi(x_m)$ , and there exist coefficients  $\alpha_i$  such that

$$w^\Phi = \sum_{i=1}^m \alpha_i \Phi(x_i) \quad (C.2)$$

Denoting an  $m \times m$  matrix  $K$  by

$$K_{ij} = k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (C.3)$$

The kernel PCA problem becomes,

$$m\lambda L\alpha = K^2\alpha \quad (C.4)$$

$$m\lambda\alpha = K\alpha \quad (C.5)$$

Where  $\alpha$  denotes a column vector with entries  $\alpha_1, \alpha_2, \dots, \alpha_m$ . The above derivations assume that all the projected samples  $\Phi(x)$  are centered in  $R^f$ .

Note that the conventional PCA is a special case of Kernel PCA with polynomial kernel of first order. In other words, Kernel PCA is a generalization of conventional PCA since different kernels can be utilized for different nonlinear projections.

We can now project the projects in  $R^f$  to a lower dimensional space spanned by the eigenvectors  $w^\Phi$ . Let  $x$  be a test sample whose projection is  $\Phi(x)$  in  $R^f$ , then the projection of  $\Phi(x)$  onto the eigenvectors  $w^\Phi$  is the nonlinear principal components corresponding to  $\Phi$ :

$$w^\Phi \cdot \Phi(x) = \sum_{i=1}^m \alpha_i (\Phi(x_i) \cdot \Phi(x)) = \sum_{i=1}^m \alpha_i k(x_i, x) \quad (C.6)$$

In other words, the first  $q$  ( $1 \leq q \leq m$ ) nonlinear principal components are extracted using the kernel function with the expensive operation that explicitly projects the samples to a high dimensional space  $R^f$ .

Similar to Kernel PCA, in Kernel FisherFace, the projected samples  $\Phi(x)$  are centered in  $R^f$ . As in FLD,  $S_W^\Phi$  and  $S_B^\Phi$  are defined as within-class and between-class scatter matrices. Applying FLD in kernel space, eigenvalues  $\lambda$  and eigenvectors  $w^\Phi$  are found as follows:

$$\lambda S_W^\Phi w^\Phi = S_B^\Phi w^\Phi \quad (C.7)$$

Which can be obtained by,

$$W_{OPT}^\Phi = \arg \underbrace{\max}_{w^\Phi} \frac{|(w^\Phi)^T S_B^\Phi w^\Phi|}{|(w^\Phi)^T S_W^\Phi w^\Phi|} = [w_1^\Phi \ w_2^\Phi \ \dots \ w_m^\Phi] \quad (C.8)$$

Where  $\{w_i^\Phi | i = 1, 2, \dots, m\}$  is the set of generalized eigenvectors corresponding to the  $m$  largest generalized eigenvalues  $\{\lambda_i | i = 1, 2, \dots, m\}$ .

For given classes  $t$  and  $u$  and their samples, we define the kernel function by

$$(k_{rs})_{tu} = k(x_{tr}, x_{us}) = \Phi(x_{tr}) \Phi(x_{us}) = \Phi(x_{tr})^T \Phi(x_{us}) \quad (C.9)$$

Let  $K$  be a  $m \times m$  matrix defined by the elements  $(K_{tu})_{u=1, \dots, c}^{t=1, \dots, c}$  where  $K_{tu}$  is a matrix composed of dot products in the feature space  $R^f$ , i.e.,

$$(K_{tu})_{u=1, \dots, c}^{t=1, \dots, c} \text{ where } K_{tu} = (k_{rs})_{s=1, \dots, l_u}^{r=1, \dots, l_t} \quad (C.10)$$

Note that  $K_{tu}$  is a  $l_t \times l_u$  matrix, and  $K$  is a  $m \times m$  symmetric matrix. We also define a matrix  $Z$ :

$$Z = (Z_t)_{t=1, \dots, c} \quad (C.11)$$

Where  $Z_t$  is a  $l_t \times l_t$  matrix with terms equal to  $\frac{1}{l_t}$ , i.e.  $Z$  is a  $m \times m$  block diagonal matrix. The between-class and within-class scatter matrices in a high dimensional feature space  $R^f$  are defined as

$$S_B^\Phi = \sum_{i=1}^c l_i \mu_i^\Phi (\mu_i^\Phi)^T \quad (\text{C.12})$$

$$S_W^\Phi = \sum_{i=1}^c \sum_{j=1}^{l_i} \Phi(x_{ij}) \Phi(x_{ij})^T \quad (\text{C.13})$$

Where  $\mu_i^\Phi$  is the mean of class  $i$  in  $R^f$ ,  $l_i$  is the number of samples belonging to class  $i$ . From the theory of reproducing kernels, any solution  $w^\Phi$  in  $R^f$  must lie in the span of all training samples in  $R^f$ , i.e.,

$$w^\Phi = \sum_{p=1}^c \sum_{q=1}^{l_p} \alpha_{pq} \Phi(x_{pq}) \quad (\text{C.14})$$

The solution can be obtained by solving;

$$\lambda K K \alpha = K Z K \alpha \quad (\text{C.15})$$

Consequently,

$$\begin{aligned} W_{OPT}^\Phi &= \arg \max_{W^\Phi} \frac{|(W^\Phi)^T S_B^\Phi W^\Phi|}{|(W^\Phi)^T S_W^\Phi W^\Phi|} \\ &= \arg \max_{W^\Phi} \frac{|\alpha K Z K \alpha|}{|\alpha K K \alpha|} \\ &= [w_1^\Phi \ w_2^\Phi \ \dots \ w_m^\Phi] \end{aligned} \quad (\text{C.16})$$

## APPENDIX D

### SPHERICAL HARMONICS LIGHTING

In computer graphics, the problem with the rendering equation is that it is difficult to compute, and definitely not a real-time friendly operation. It is an integral over a hemisphere of directions where  $L$  appears on both sides of the equation:

$$L(\mathbf{x}, \vec{\omega}_0) = L_e(\mathbf{x}, \vec{\omega}_0) + \int_S f_r(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_0) L(\mathbf{x}', \vec{\omega}_i) G(\mathbf{x}, \mathbf{x}') V(\mathbf{x}, \mathbf{x}') d\omega_i$$

where (D.1)

$L(\mathbf{x}, \vec{\omega}_0)$  = the intensity reflected from position  $\mathbf{x}$  in direction  $\vec{\omega}_0$

$L_e(\mathbf{x}, \vec{\omega}_0)$  = the intensity emitted from  $\mathbf{x}$  by the object itself

$f_r(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_0)$  = the Bidirectional Reflectance Distribution Function (BRDF) of the surface at point  $\mathbf{x}$ , transforming incoming light  $\vec{\omega}_i$  to reflected light  $\vec{\omega}_0$ . The BRDF is a surface property which defines how light is reflected at an opaque surface.

$L(\mathbf{x}', \vec{\omega}_i)$  = light from  $\mathbf{x}'$  on another object arriving along  $\vec{\omega}_i$

$G(\mathbf{x}, \mathbf{x}')$  = the geometric relationship between  $\mathbf{x}$  and  $\mathbf{x}'$ .

$V(\mathbf{x}, \mathbf{x}')$  = a visibility test, returns 1 if  $\mathbf{x}$  can see  $\mathbf{x}'$ , 0 otherwise

It is nearly impossible to integrate the above equation in real-time. And also the function itself is not friendly for the state of the art graphics hardware available today.

Spherical harmonics expansion is used to approximate the light transfer function at a surface. The Monte-Carlo integration is used to integrate the above equation for mesh surfaces (Figure D.1).

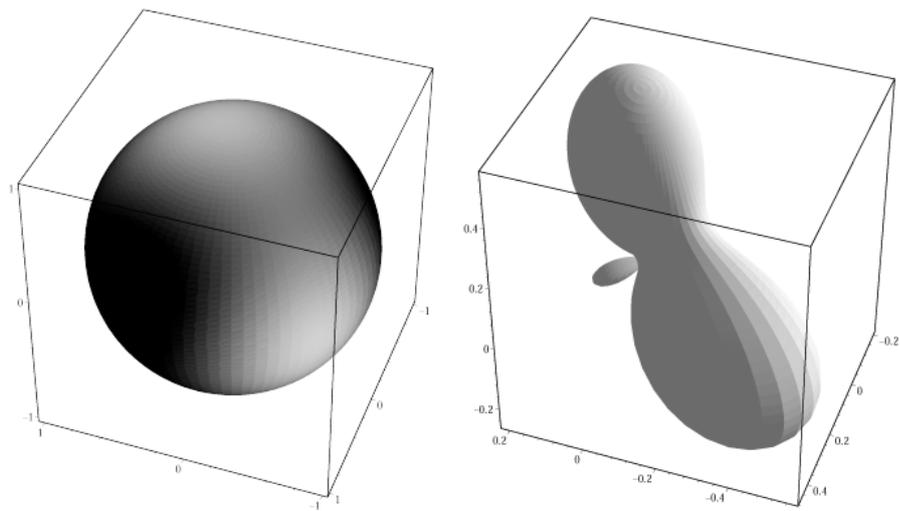


Figure D.1 Light transfer function (left), estimated spherical plot(right);

The real-time performance of spherical harmonics lighting is very good, and also the rendering quality with static shadows is compelling. Figure D.2 shows, a static scene rendered with OpenGL using the spherical harmonics lighting method.

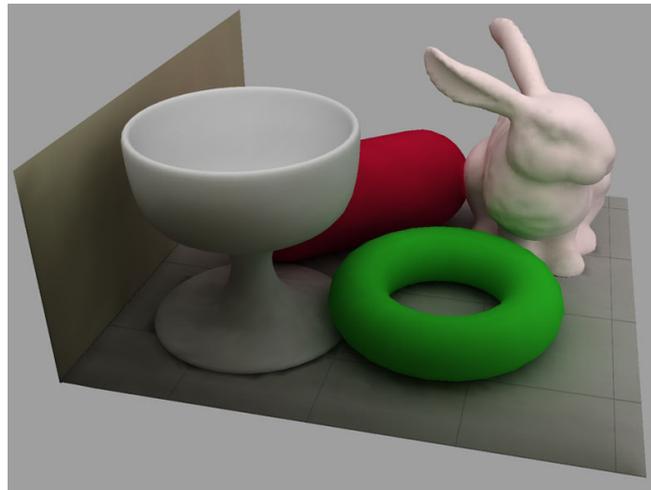


Figure D.2 Scene lit using shadowed spherical harmonics lighting method in real-time

# CURRICULUM VITAE

## PERSONAL INFORMATION

Surname, Name: Tunçer, Fahri  
Nationality: Turkish (TC)  
Date and Place of Birth: 25 May 1975, Ankara  
Phone: +90 505 299 87 47  
e-mail: fahri.tuncer@gmail.com

## EDUCATION

Degree	Institution	Year of Graduation
MS	METU E.E. Eng.	2002
BS	METU E.E. Eng.	1997
High School	Ankara Anadolu Lisesi	1992

## WORK EXPERIENCE

Year	Place	Enrollment
2000-today	IES-SEBIT A.Ş.	Sr. Software Design Eng.
1999-2000	Click Ent.	Sr. Software Developer
1998-1999	Aselsan A.Ş.	Hardware Design Engineer
1996-1998	SEBIT A.Ş.	Sr. Software Developer
1995-1996	TÜBİTAK-BİLTEN	Software Developer
1995 July	TÜBİTAK-BİLTEN	Intern Engineering Student

## FOREIGN LANGUAGES

Advanced English, Advanced German

## PUBLICATIONS

1. Tuncer, F., Halıcı, U., Ulusoy İ., Post-filtering Focus Blurring Formation Using Open GL, SIU 2002, pp. 212-215. (in Turkish)
2. Tuncer, F., Introduction to OpenGL, Game 2002-İstanbul.