Combining Probabilistic Shape-from-Shading & Statistical Facial Shape Models

Touqeer Ahmad

Submitted for the degree of Master of Science (M.Sc.)

Department of Computer Science

THE UNIVERSITY of York

December 2010

Abstract

In this thesis we present an approach to combine a probabilistic Shape-from-Shading algorithm with statistical models of facial shapes. Thesis presents how Fisher-Bingham (FB_8) distributions are sampled using Gibbs sampling to give normal distributions on the tangent plane. These normal distributions are in turn combined with normal distributions arising from statistical models of facial shapes.

Chapter 2 gives a brief review of Shape-from-Shading and statistical shape models. In Chapter 3 we formulate the problem under consideration and give an outline of our approach. In Chapter 4 we describe the probabilistic Shape-from-Shading algorithm based on Directional Statistics, Markov Random Fields and Belief Propagation. We describe the statistical facial shape models for needle maps and surface height in the same chapter based on the concepts of dimension reduction e.g. Principal Component Analysis, Principal Geodesic Analysis and tools of cartography e.g. Azimuthal Equidistant Projection.

Chapter 5 details sampling of Fisher-Bingham distributions using a slice sampling approach of the Gibbs sampler. In Chapter 6 we discuss the actual algorithm of combining the two types of surface normals using multivariate Gaussian distributions on the tangent plane. This chapter also details how the statistical surface height model is used to recover surface heights from surface normals. The Fisher criterion and smoothing are used to deal with outliers arising from the regions of shadow and specular reflectance. Chaper 7 lists our experiment results for synthetic and real images. The Iterative Closest Point algorithm is used to calculate the error difference between the groundtruth and recovered normals and surface heights. The results of experiments in this thesis have shown improvements in probabilistic shape-from-shading when statistical shape models are used.

Contents

List of Figures

List of Tables

Acknowledgements

First of all I would like to thank my supervisor Professor Richard Wilson for his kind guidance throughout the course of my MSc, helping me understand research material and proof reading my documents, presentation, reports and thesis. I am very thankful to EURECA[1] scholarship under the Erasmus Mundus scheme for fully funding my masters degree here at University of York.

I am grateful to Dr. William Smith and Dr. Tom Haines for providing their software, models and data set for my experiments. I cannot but acknowledge the wonderful multicultural and multilingual enviornment created by my fellow research students in Computer Science Department and CVPR group; especially Sravan Kumar, Abhishek Dutta, Eliza Xu, Lichi Zhang, Thomas Lampart, Zhihong Zhang, Mohammad Haseeb, Abdul Haseeb, Ahmad Shahid, Mian Asbat and all others. I would also like to mention Outdoor Society of York that provided me a chance to visit the surroundings of York which is no doubt a beautiful place.

I am thankful for all the love, prayers and best wishes from people back at home; my parents Prof. Muhammad Ikram, Kausar Ikram, my elder brother Touseef Ikram, my cute lovely niece Fatima Madni and my very close friends.

[1] http://www.mrtc.mdh.se/eureca/

Declaration

I declare that all the work in this thesis is solely my own, except where attributed and referred to other authors. Some material from this thesis have been previously presented and another is under review: please refer to Appendix for details.

Chapter 1

Introduction

The human visual system exploits a number of cues to infer 3D information from the 2D image projected onto the retina [1]. These cues include motion parallax, stereo vision, perspective, interposition, shadowing and accommodation. In absence of other cues it is still possible to have some perception of 3D shape from variations in shading especially in case of observing human faces. This thesis is concerned with a classical problem of Computer Vision i.e. to infer 3D shape from variations in shading in a single image; in our case a face image. In the Computer Vision community this problem is named as Shape-from-Shading and has been an active field of research for the last 4 decades. The earliest use of Shape-from-Shading was to recover information about the surface of the moon in preparation for moon landings [1, 56, 57] .

Like many other problems of Computer Vision, the aim of Shape-from-Shading is the inverse of Computer Graphics. In computer graphics the objective is to create realistic images from description of scenes and imaging conditions whereas the objective of Shape-from-Shading is to recover the shape of a scene (face) from a single intensity image. Since one of the dimensions (i.e. depth) is lost during the process of image formation, shape-from-shading is an ill-posed problem. Despite being a difficult challenge, Shape-from-Shading has broad ranging potential applications of great utility. Within the context of faces, Shape-from-Shading has numerous applications e.g. virtual reality, computer games, avatars, facial re-animation, face re-lighting, expression editing, pose normalization, virtualised reality displays, driver safety, human computer interaction and face recognition etc [1, 56, 57].

Since the shape information is independent of viewing conditions, the information extracted from an image using shape-from-shading can be used for facial recognition under dramatically different pose and illumination conditions. 3D shape information provides invariance to changes in lighting and pose but to recover 3D shape information in a way which maintains real world applicability is still an open problem. Laser scanners are available to capture 3D shape of a subject but these are very expensive; time consuming and need paticipation of the subject. The alternative to these equipments are number of techniques that use single or multiple images to recover 3D shape and collectively termed as Shape-from-X where X could be shading, stereo, texture, edges, motion and contours. Shape-from-Shading holds the possibility of recovering 3D shape from a single image unlike rest of these techniques except shape-from-texture. Shape-from-shading also provides a far richer shape information as compared to other Shape-from-X techniques as it uses information contained in each pixel i.e. surface orientation.

This research aims to improve a recent shape-from-shading algorithm for faces, based on directional distributions with the use of pre-existing statistical face shape models.

Following this chapter we provide a brief literature review of shape-from-shading and statistical models of faces. In chapter 3 we give details about the shape-from-shading algorithm which we intend to improve on; along with description of how statistical model of face shapes have been constructed and how we are going to use it to improve the shape-from-shading algorithm. Chapter 4 gives the details about the sampling method that has been used to sample the directional distributions outputted from shape-from-shading algorithm.

Chapter 5 talks about the technical details of getting mean vectors and covariance matrices for all pixels and how Gaussian distributions can be fitted and combined on a tangent plane. Details regarding the use of the Fisher criterion are also listed in this chapter. Chapter 6 provides the experimental details for different types of test data and describe the algorithm being used to calculate their errors. We conclude our thesis with conclusions in the last chapter.

Chapter 2

Literature Review

The aim of our work is to combine a probabilistic approach of Shape-from-Shading with statistical face shape models in a probabilistic framework. In this chapter we will give a brief review regarding these two major topics i.e. Shape-from-Shading and Statistical Shape models.

2.1 Shape-from-Shading

Shape-from-Shading is a classical 4-decade old problem of Computer Vision; first presented by Horn [15] in 1970. The problem can be stated as: 'to infer 3D shape of scene from a given single 2D intensity image'. Shape-from-Shading problem is stated under a number of assumptions about scene, capturing method and light source. Specifically these assumptions are Lambertian reflectance, known albedo, smooth surface, no inter-reflectance, orthographic projection and a single infinitely distant light source. Image irradiance only constrains one degree of freedom whereas the surface orientations to be inferred from shape-from-shading have two degrees of freedom. Further information is necessary to get a solution and typically a smoothness assumption is used to give a complete solution.

Ikeuchi and Horn [25] obtain a needle map by minimising an energy function consisting of sum of two per pixel costs. The two costs being the brightness cost and smoothness cost. The brightness cost measures the error between the actual sensed irradiance and irradiance calculated using the reflectance map. As its name states the smoothness cost measures the deviation from a smooth surface. Brooks and Horn [26] gave a set of algorithms, with the first minimizing the same cost as Ikeuchi and Horn but using a much simpler implementation.

In their review paper; Zhang et al. [27] compared a number of SfS algorithms available at that time in 1999. They placed Lee & Kuo [14] at the first place and Zheng & Chellappa [19] at the 2nd place. These both methods work with depth rather than surface orientations and have the advantage of enforcing integrability and thus not requiring an integration step to get surface height from surface orientations. Lee and Kuo [14] linearised the reflectance map and solved the resulting linear equation using multigrid method. They define costs for violating the Lambertian reflectance and smoothness assumption. The approach is then to minimize these costs using a multigrid method. Zheng et al. [27] also compared SfS algorithms from Bichsel & Pentand [28] , Lee & Rosenfeld [29] , Pentland [30] and Tsai & Shah [31].

Frankot & Chellappa [9] defined the integrability constraint as,

$$\frac{\delta^2 z}{\delta x \delta y} = \frac{\delta^2 z}{\delta y \delta x} \tag{2.1}$$

where, z is the surface depth and x and y are image coordinates. The integrability constraint enforces that a surface can be integrated from a needle map. An arbitrary needle map will not necessarily satisfy the integrability constraint. Frankot & Chellapa formulated a method of projecting a needle map onto nearest needle map that satisfies this constraint. This technique can be used with any iterative algorithm.

Worthington & Hancock [13] presented a brightness constraint to limit surface orientation to lie on a cone with half angle θ given by irradiance equation when Lambertian reflection is assumed,

$$\frac{I}{a} = \hat{n}.\hat{I} = \cos(\theta) \tag{2.2}$$

They gave a two step iterative algorithm; first a smoothness constraint is applied and the normals are projected back to the cone as shown in figure 2.1. This makes the irradiance constraint as a hard constraint. The advantage of the Worthington & Hancock approach is that the smoothness component can be changed independently of the rest of algorithm so a number of robust smoothness techniques can be tried and compared.

Robles-Kelly [32] presented an approach to add an integration constraint by modifying the smoothness method in Worthington & Hancock [13]. This is one of few algorithms having irradiance as well as integration as hard constraints. Unfortunately the performance of this algorithm is dependent on initialization. Louw & Nicolls [33, 34] presented a discrete approach to

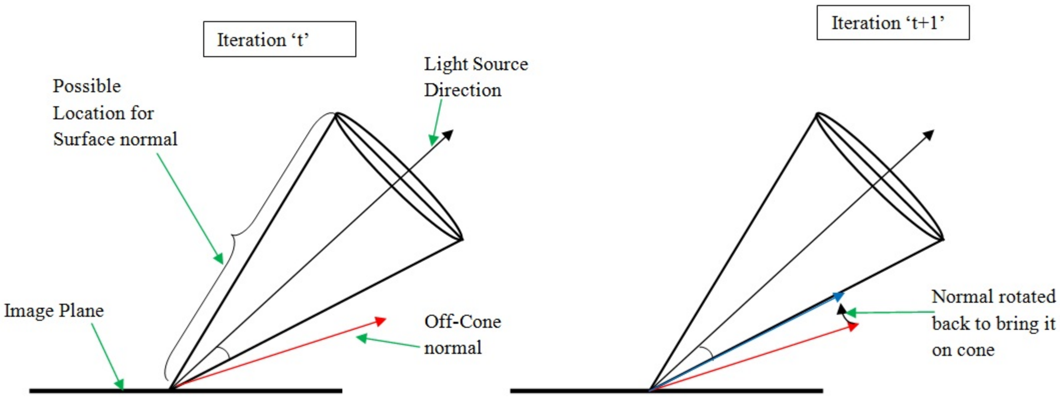

Figure 2.1: Worthington and Hancock : Cone Constraint

solve SfS using depth based on belief propagation. Unfortunately a discrete approach is not suitable for this continuous problem. Another algorithm having both constraints as hard constraints was presented by Potetz [35]. It makes use of belief propagation which serves to estimate the marginals of a multivariate probability distribution often presented by a graphical model. Potetz uses two variables $\frac{\delta x}{\delta z}$ and $\frac{\delta y}{\delta z}$ and uses various factor nodes to provide the reflectance information, smoothness assumption and integrability constraint. Whilst this algorithm takes a lot of resources but is probably the most capable available.

Haines & Wilson [3, 36, 37] presented a SfS approach in which directional statistics (specifically Fisher-Bingham distributions) are used to represent the surface orientations. Like Potetz [35], Haines & Wilson also used belief propagation to estimate marginals of Fisher-Bingham distributions over a pairwise Markov Random Field. We will give details of this algorithm later in Chapter 3. Statistical shape models have also been used by several authors for shape-from-shading to resolve ambiguity e.g. Atick et al. [38] Dovgrad & Barsi [39] and Smith & Hancock [1, 40 – 47]. We will give a brief review of usage of shape models in next section.

2.2 Statistical Shape Models

Image based statistical approaches assume that images of faces occupy a subspace or a facespace; in the space of all possible images [1]. From a training set of face images, a manifold learning technique is chosen to learn this subspace or facespace. Eigenfaces; one of the best known statistical technique proposed by Turk & Pentland [48] uses Principal Component Analysis (PCA) to derive a basis set of images. This basis set of images is used to express the training set of face images in fewer dimensions. PCA is a classical dimensionality reduction technique that has been used by many researchers for making statistical models. Kirby & Sirovich [49] were the first to apply PCA to images of faces. The eigenvalues and eigenvectors of the centered covariance matrix of data are found to perform PCA. Turk & Pentland coined the term eigenfaces to describe eigenvectors that themselves have a face-like appearance.

Unlike PCA, Linear Discriminant Analysis (another classical dinmesionality reduction scheme) has the capability to maintain the separability of different classes by maximizing between-class scatter while minimizing within-class scatter. Belhumeur et al. [50] were the first to apply LDA for statistical modeling of faces and they named their technique as Fisherfaces. The most remarkable work on 2D facial modeling have been undertaken by Cootes et al.[51, 52]. Cootes et al. built an Active Shape Model by applying PCA to the set of vectors describing the shapes in the training set [52]. Cootes extended this idea for appearance and presented the Active Appearance Model that makes use of intensity information contained within the shape [51]. Atick et al. [38] parameterized a general face with PCA and then uses SfS to determine parameters for individual specimens. Dovgrad & Barsi [39] extended this idea by utilizing the symmetry of human face.

Smith & Hancock [1, 40, 41, 42, 43] constructed a statistical model of face in directional domain of surface orientation i.e. they used surface normals to construct their statistical model instead of depth as done by several others for SfS problem. Initial work has distributions of direction for each pixel on the tangent plane of the mean surface orientation using PCA. The idea was then extended to nonlinear manifolds using Principal Geodesic Analysis. Smith & Hancock also constructed a statistical model of surface heights that can be used to infer surface heights from surface normals. Following the analogy of Active Appearance Model (AAM) and Coupled View AAM of Cootes et al.[53], Smith presented a coupled model that captured variations in both surface height and surface normal direction [1, 44, 45]. We will describe details of Smith's statistical models in next chapter.

Chapter 3

Problem Formulation

In this Chapter we formally define the objective of this thesis and detail the vocabulary that will be used throughout the thesis. We give a flow chart of the algorithm to show a compact description of our approach and then give a summary of the remaining chapters of the thesis.

3.1 Problem Statement

Shape-from-Shading is an interesting approach to the problem of finding the shape of a face becuase it only requires one image and no subject participation. However, SfS alone is not accurate enough to produce good shape models. Previously, SfS has been combined with shape models to produce realistic face reconstructions. In this work, we aim to improve the quality of such models by combining a probabilistic Shape-from-Shading model based on Fisher-Bingham 8-parameter distribution (FB_8) with a statistical model of 3D face shapes. The benefits are two-fold; firstly we can correctly weight the contributions of the data and model where the surface normals are uncertain, and secondly we can locate areas of shadow and facial hair using inconsistencies between the data and model. Our work is based on Haines' Shape-from-Shading algorithm [3] and Smith's statistical face shape models [1].

Haines' SfS model is based on directional statistics and belief propapgation. Haines' model takes an image, light source direction and albedo value as inputs and gives surface normal directions using Fisher-Bingham (FB_8) distributions as output. Smith built his face shape model using aligned 3D face shapes and his model gives a Gaussian distribution on the tangent plane of a unit sphere for each surface normal. These two models produce results on different spaces using different probability distributions. In order to combine these two models they should be in the same space and represented by tractable distributions. The best way to combine these methods is to transform the Gaussian distributions generated on the tangent plane by Smith's statistical model into Fisher-Bingham distributions on the unit sphere. This transformation however is less tractable and not much literature was found. We took the second best route and instead transformed Fisher-Bingham distributions into Gaussian distributions on the tangent plane.

We sample the Fisher-Bingham (FB_8) distribution using a Gibbs sampling algorithm of Kume & Walker [11]. These are then modelled as Gaussian distributions on the surface tangent plane defined by the model. The shape model provides a second Gaussian distribution describing the likely configurations of the model; these distributions are combined on the tangent plane of the directional sphere to give the most probable surface normal directions for all pixels. The Fisher criterion is used to locate inconsistencies between the two distributions and smoothing is used to deal with outliers originating in the shadowed and specular regions. A surface height model is then used to recover surface heights from surface normals.

3.2 Breakdown of the Model

We can breakdwon our approach in a number of major steps, as listed below. The flow chart of our approach is shown next in Figure 3.1.

Step 1 : FB_8 Distributions from SfS

First of all we use Haines' Shape-from-Shading model; provide it an input face image, light source direction as a vector and aibedo value as a scalar. The SfS algorit hm in-turn gives us Fisher-Bingham (FB_8) distributions for every pixel in the image.

Step 2 : Gaussian Distribution from Statistical Shape Model

Smith's statistical shape model originally gives a mean surface normal vector and a matrix of reduced dimensions. The mean surface normals can be used as it is for our approach; but to fit Gaussian distributions for each surface normal we need a covariance matrix. We find these individual covariance matrices for all pixels using Eigenvectors and Eigenvalues given by PGA used in Smith's approach as detailed in Chapter 4 and 6. In essence we get one Gaussian distribution per pixel from Smith's model.

Step 3 : Gaussian from Fisher-Bingham

We use slice sampling algorithm of Kume and Walker to generate samples from Fisher-Bingham (FB_8) distribution for each pixel from Haines' model. We use first few samples to compute the extrinsic mean. Then we use an iterative appoach to find the spherical median using newly generated samples and extrinsic mean as detailed in Chapter 4 and 6. Once the spherical median is found we use the samples from the last iteration to compute a covariance matrix on the tangent plane so that a Gaussian distribution can be fitted.

Step 4 : Combining the Gaussians

The two Gaussian distributions on the tangent planes resulting for each pixel are then combined using Logarithmic and Exponential mapping to give probabilistically a more accurate surface normal. The mean surface normals are used as base points for these mapping due to their statistical accuracy.

Step 5 : Dealing with Inconsistencies

We use the Fisher criterion to deal with the problematic pixels where probability distributions arising from these two models do not register with each other. A threshold is decided and then any pixel giving the Fisher criterion's value greater that this threshold is replaced with the mean surface normal from the statistical model. Discontinuties result in the regions of facial hair and specular reflection; these are dealt by smoothing the normals using Smith's statistical model.

Step 6 : Surface Height from Surface Normals

After step 5 we get smooth normals and then use these normals to get surface heights using Smith's height model as described in Chapter 6.

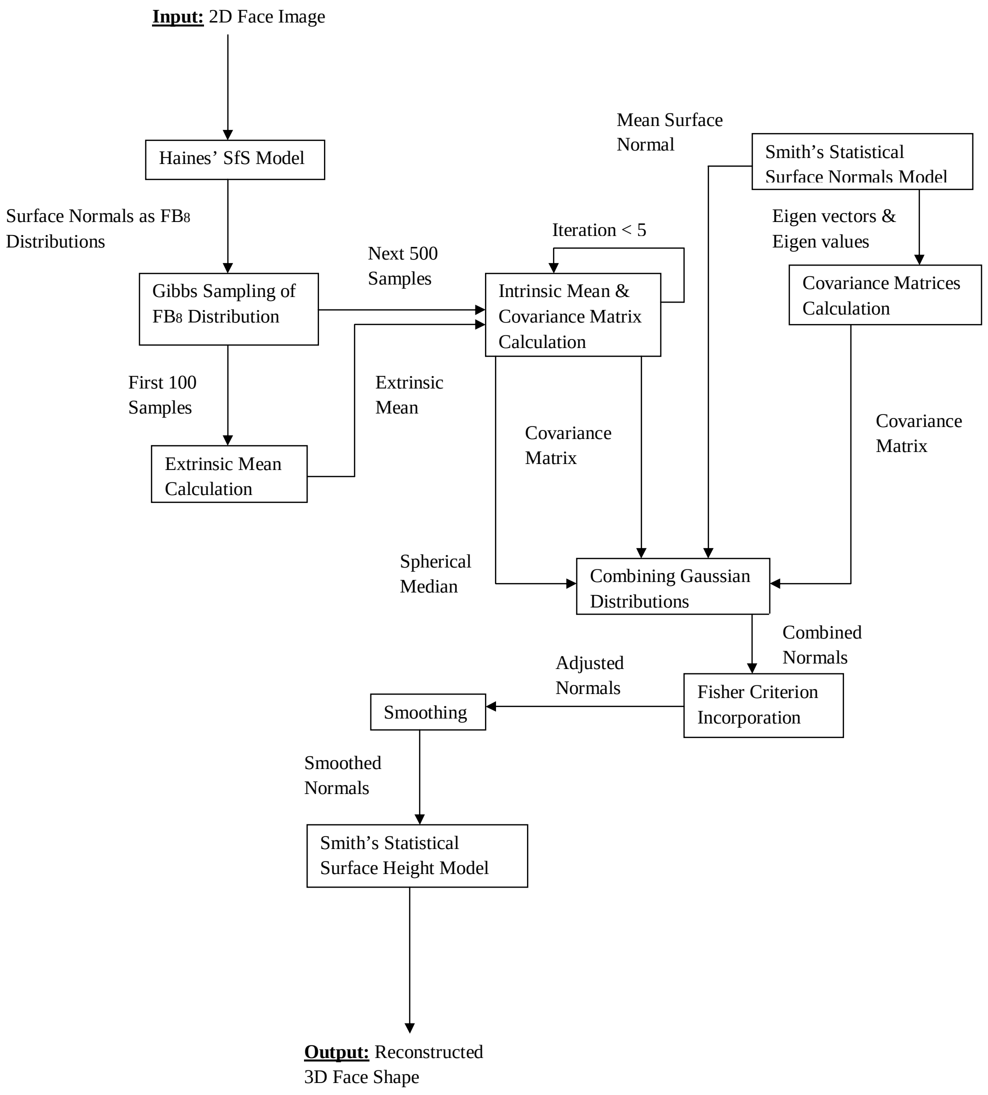
Input: 2D Face Image
Haines' SfS Model
Surface Normals as FB8 Distributions
Gibbs Sampling of FB8 Distribution
First 100 Samples
Extrinsic Mean Calculation
Extrinsic Mean
Next 500 Samples
Iteration < 5
Intrinsic Mean & Covariance Matrix Calculation
Mean Surface Normal
Smith's Statistical Surface Normals Model
Eigen vectors & Eigen values
Covariance Matrices Calculation
Covariance Matrix
Covariance Matrix
Spherical Median
Combining Gaussian Distributions
Combined Normals
Fisher Criterion Incorporation
Adjusted Normals
Smoothing
Smoothed Normals
Smith's Statistical Surface Height Model
Output: Reconstructed 3D Face Shape

Figure 3.1: Flow Chart of the Model

3.3 Chapter Details

3.3.1 Chapter 4

In Chapter 4 we give a brief description of the two models on which our approach is based i.e. Haines' Shape-from-Shading model and Smith's statistical shape model. We begin this chapter by giving general idea of Shape-from-Shading i.e. Lambert's law and then continue with Haines' specific Shape-from-Shading method based on directional statistics and belief propagation.

We give details about Smith's statistical model. We briefly discuss how the statistical model of surface normals is constructed from training data using dimension reduction techniques, Log/Exp mappings and spherical median. We also list the difference between PCA and PGA and the statistical models based on these two dimension reduction techniques. In the same chapter we also describe Smith's surface height model and how it can be used to reconstruct surface heights from surface normals.

3.3.2 Chapter 5

This chapter talks about the sampling of Fisher-Bingham distribution. We start this chapter with an introduction of Fisher-Bingham distribution and its effectiveness for representation of directional data. We give basics of Gibbs and slice sampling in general. The details about the Gibbs sampling of Fisher-Bingham distribution based on Kume & Walker's work are listed next. We conclude this chapter by showing the efficacy of this sampling method using a graphical example.

3.3.3 Chapter 6

We start chapter 6 by describing different options to solve the problem under consideration and argue why we have chosen this approach. We describe how the individual mean vectors and covariance matrices are acquired from Haines' and Smith's models to give Gaussian distributions on the tangent plane. Next, we detail how a product normal distribution results from Gaussian distributions. We talk about different issues like base point selection for Log/Exp mappings and accuracy of combined normals. We give graphical examples for verification of our arguments. Once the normals are combined, we list the ways to deal with outliers i.e. the incorporation of the Fisher criterion and smoothing of normals. We conclude this chapter by listing the details of surface height reconstruction from surface normals.

Chapter 4

Shape-from-Shading and Shape Models

4.1 Probabilistic Shape from Shading

Haines presented a probabilistic Shape-from-Shading algorithm based on Markov random fields and belief propogation [3] which will be used later in this thesis. Directional statistics, specifically Fisher Bingham distributions were used for the probabilistic representation of surface orientation. Haines approach generally performed well as compared to Lee & Kuo [14] and Worthington & Hancock [13]. In this chapter we will briefly present Haines shape-from-shading algorithm along with some basic ideas.

4.1.1 Lambert's Law

Haines' algorithm to solve the Shape from Shading problem follows the original assumptions of Horn [15] . Suppose that a facial surface $F \in R^3$ is projected orthographically onto the image plane and parameterised by the surface height function $z(x, y)$. The local surface normal at the pixel location (x, y) is given by $\hat{\mathbf{n}}_{(x,y)}$ and the measured intensity at this location is $I_{(x,y)}$. The relationship between the surface normal, light source direction and image intensity is described by the image irradiance equation. For the shape from shading problem it is assumed that the surface has Lambertian reflectance i.e. light falling on the surface is scattered equally in all the directions. For a perfectly diffuse surface the Lambert's Law is given as,

$$I_{(x,y)} = a\hat{\mathbf{L}}.\hat{\mathbf{n}}_{(x,y)} \tag{4.1}$$

where, $a \in [0, 1]$ is the diffuse albedo: the intrinsic reflectivity of surface i.e. what proportion of light is reflected from the surface. $\hat{\mathbf{L}} \in R^3$ is the infinitely distant light source of unit magnitude i.e. $|\hat{\mathbf{L}}| = 1$. Light source direction $\hat{\mathbf{L}}$, image intensity $I_{(x,y)}$ and albedo a are the inputs to the algorithm where as unit normal $\hat{\mathbf{n}}_{(x,y)} \in R^3$, $|\hat{\mathbf{n}}_{(x,y)}| = 1$ is the output to be inferred from the algorithm.

In general the surface normal $\hat{\mathbf{n}}_{(x,y)}$ can not be inferred from a single intensity value $I_{(x,y)}$ since it has two degrees of freedom i.e. azimuth angle and the elevation angle on the unit sphere. Lambert's Law gives a partial constraint on the direction of surface normals when the dot product between two vectors in Equation(4.1) is replaced with cosine,

$$\theta_{(x,y)} = \arccos\left(\frac{I_{(x,y)}}{a}\right) \tag{4.2}$$

Geometrically this means that the surface normal is constrained to lie on a right circular cone whose axis is the light source direction $\hat{\mathbf{L}}$ and half angle is $\theta_{(x,y)}$.

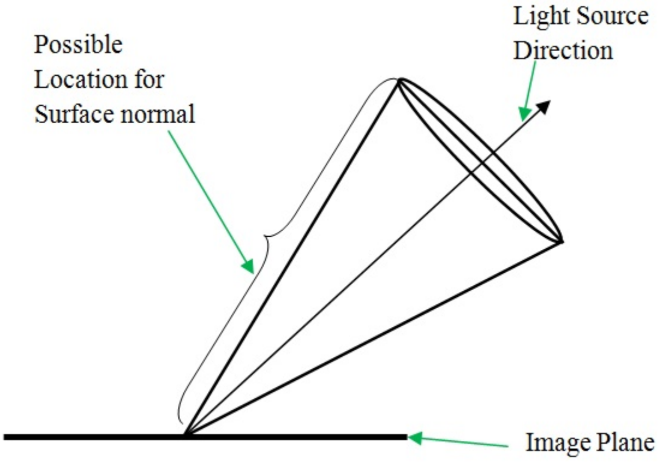

Figure 4.1: Cone Constraint

In SfS literature this constraint is named as cone constraint [13]. Haines takes the typical approach and uses smoothness as the further source of information to constrain the second degree of freedom.

4.2 Directional Statistics

The distinct feature of Haines algorithm is the use of directional distribution which allows the representation of surface orientation with a random direction instead of two random variables $\frac{\delta x}{\delta z}$ and $\frac{\delta y}{\delta z}$ that are used mostly in SfS literature. Specifically the eight paramter Fisher-Bingham(FB) distribution was used. A Fisher-Bingham distribution when multiplied with itself results in another FB distribution; this multiplication property was used for belief propagation. The sub-model of the Bingham-Mardia distribution was used to describe the cone constraint which is needed to provide the prior on each pixel.

4.2.1 Fisher-Bingham Distribution

The Fisher-Bingham distribution with 8 paramters FB$_8$ is simply the multiple of von-Mises-Fisher distribution and Bingham distribution and is given as,

$$P_{FB}(\hat{\mathbf{x}};\mathbf{A}) \propto \exp(\mathbf{u}^T\hat{\mathbf{x}} + \hat{\mathbf{x}}^T\mathbf{A}\hat{\mathbf{x}}) \tag{4.3}$$

where $\hat{\mathbf{x}}$ is the considered direction, $\mathbf{u}$ is the Fisher parameter and $\mathbf{A}$ is the inverse of the covariance matrix and is symmetric. For convenience FB$_8$ may be represented as,

$$\exp(\mathbf{u}^T\hat{\mathbf{x}} + \hat{\mathbf{x}}^T\mathbf{A}\hat{\mathbf{x}}) = \Omega[\mathbf{u},\mathbf{A}] \tag{4.4}$$

The multiplication of two FB$_8$ distributions then can be described as,

$$\Omega[\mathbf{w},\mathbf{C}] = \Omega[\mathbf{u},\mathbf{A}]\Omega[\mathbf{v},\mathbf{B}] = \Omega[\mathbf{u}+\mathbf{v},\mathbf{A}+\mathbf{B}] \tag{4.5}$$

This multiplication property of FB$_8$ distribution was used for belief propagation.

4.2.2 Bingham-Mardia Distribution

The Bingham-Madia distribution [16] is a sub-distribution of the FB$_8$ distribution i.e. it is a constrained form of the FB$_8$ distribution. Using the notation introduced for FB$_8$ distribution the Bingham-Mardia distribution is given as,

$$\exp\left(-k(\hat{\mathbf{u}}^T\mathbf{x} - \cos\theta)^2\right) = \Omega[2k\cos(\theta)\hat{\mathbf{u}}, -k\hat{\mathbf{u}}\hat{\mathbf{u}}^T] \tag{4.6}$$

where $\hat{\mathbf{u}}$ is the direction of the axis of a cone and θ is the angle of that cone. This conic representation of Bingham-Mardia distribution makes it capable of representing the cone constraint of shape-from-shading as shown in figure (4.1).

4.3 Belief Propogation

Belief propagation has two main advantages over graph cuts. Belief propagation can solve continuous problems whereas graph cuts is only limited to the discrete problems. Graph cuts can only solve the maximum likelihood estimation for all random variables whereas belief propagation can also solve for the posterior distribution for each random variable. Haines used this property of belief propagation for the continous problem of shape-from-shading to find posterior distribution of surface normals for each node in a Markov random field. Loopy belief propagation is a message passing algorithm that can be described by the equation,

$$P(\mathbf{x}) = \prod_{v \in V} \psi_v(\mathbf{y}_v) \tag{4.7}$$

where $\mathbf{x}$ is a set of random variables and $\forall v$; $\mathbf{y}_v \subset \mathbf{x}$. $P(\mathbf{x})$ is the joint probability distribution for all the random variables involved. This joint distribution can also be described as sum of costs that can be found by taking the negative log of probabilities.

$$C(\mathbf{x}) = -\ln(P(\mathbf{x})) = -\ln\left(\prod_{v \in V} \psi_v(\mathbf{y}_v)\right) = \sum_{v \in V} -\ln \psi_v(\mathbf{y}_v) = \sum_{v \in V} \phi_v \mathbf{y}_v \tag{4.8}$$

The two formulations of belief propagation are sum-product and min-sum. The sum-product formulation is used to find the marginal (posterior) distribution for each random variable where as the min-sum calculates the maximum likelihood estimation for all the random variables involved.

4.3.1 Pairwise Markov Random Fields

Equation (4.8) can be represented as a bipartite graph where each random variable and each function is represented by a node. The ψ nodes are called factors which are linked to the random variables which are used for their calculation; this graphical model representation is called a factor graph. The structure of this graphical model allows belief propagation to efficiently find a solution by sending messages along the edges of this graph. The time it takes to calculate the message for a particular factor depends on the number of random variables that factor is connected to. It is preferable to design a graphical model that has a clique of two at most for each factor. In the worst case when there is only one factor connected to all random variables the algorithm becomes to brute force in efficiency. A factor graph is shown in figure (4.2) where the circles are the random variables and the squares are factors. Factors having cliques size of one (δ, ϵ), two (α, γ) and three (β) are shown.

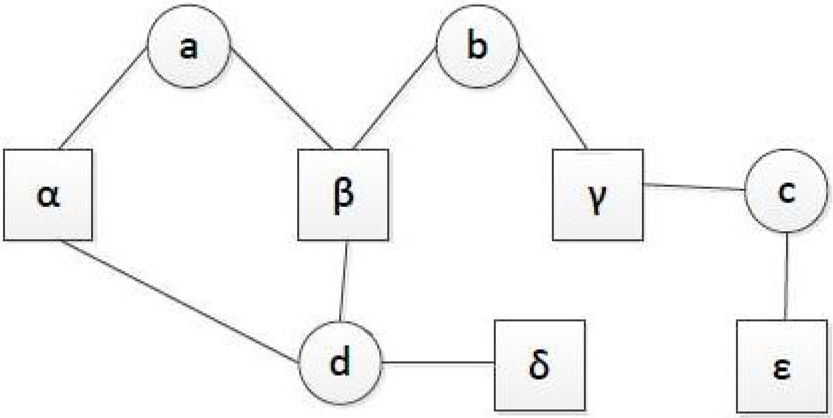

Figure 4.2: A Factor Graph

A Markov Random Field (MRF) is a set of random variables where a random variable's probability is only dependent on neighbouring random variables [3]. MRFs can be solved using belief propagation. A pairwise MRF is a Makov random field where the clique size of each factor is at most two. So in case where a factor is connected to one random variable, priors are presumed to exist for all random variables. The other case is when a factor has two random variables; in that case the factor is represented as an edge between the two involved random variables. This pairwise representation of MRF reduces the calculation time as number of nodes are far less than the actual graph cut representation. Haines expressed the shape-from-shading problem as pairwise Markov random fields. Next, we will describe the discrete and continuous formulations of belief propagation.

4.3.2 Discrete Formulation

The min-sum version of belief propagation is to find assignemts for all random variables that minimize the cost $C(\mathbf{x})$ given in equation (4.8).

$$\arg\min_{\mathbf{x}} \left(C(\mathbf{x}) \right) \tag{4.9}$$

A Markov chain is generally defined over time as a sequence of states where each state is dependent on the last state [3]. In such graphical model the factors are constrained to have a clique size of either one i.e. the prior (p) or two i.e. relationships between adjacent states (r). The factors having a clique of one can be merged with their respective random nodes. The message passed over this chain starts at one end t_0 and goes to the other end t_n while passing through the factors $r_{n,n+1}$ at each step. Message provides the cost for each state of the involved random variable. In the example shown in the figure (4.3) the message passed from variable t_0 to factor $r_{0,1}$ only includes the cost p_0; message passed from $r_{0,1}$ to t_1 includes both p_0 and $r_{0,1}$ and similarly message passing from t_1 to $r_{1,2}$ will now include all the costs resulting from p_0, p_1 and $r_{0,1}$.

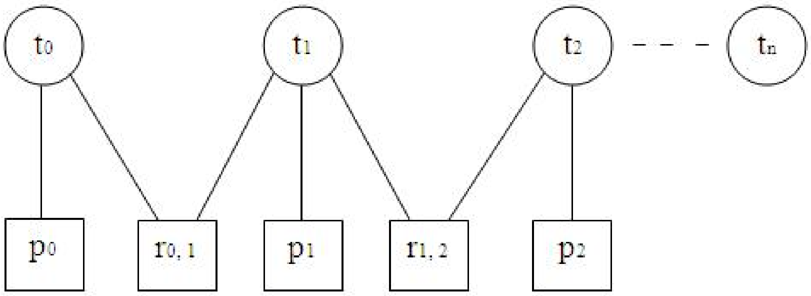

Figure 4.3: Markov Chain using Factor Graphs

For each state of t_1 we have to find the optimal costs for all previous factors leading to t_1. Let's define the cost $r_{a,b}(s_a, s_b)$ as selecting the state s_a from S_a for t_a and s_b from S_b for t_b. The optimal cost is then given by,

$$M_{r \to t_b} = \min(M_{t_a \to r}(s_a) + r_{a,b}(s_a, s_b)) \tag{4.10}$$

$M_{a \to b}$ indicates the message passing from node a to b; the sub scripts of r have been dropped for clarity. For each state of a random variable receiving this message the equation above calculates the states for previous variables and selects those result in the minimum cost. This cost factors in the previous states and factors. Let's consider the case of node t_1 which receives a message from $r_{0,1}$ i.e. $M_{r \to t_1}(s_1)$ and forwards a message to $r_{1,2}$ and also has the prior $p_1(s_1)$. In generic form this case can be written as,

$$M_{t_a \to r}(s_a) = M_{r \to t_a}(s_a) + p_a(s_a) \tag{4.11}$$

By using this message passing equations the optimal state for the final node is found. If we keep track at each node that which previous state resulted in the minimum cost for the next node we can back track this implied link list to get the optimal state for each node. This procedure is named as dynamic programming to which the belief propagation has simplified to in the case of Markov chains. We can extend this idea of belief propagation for trees as well, the difference will be we have to modify our message passing equations for arbitrary clique size. We will now list the message passing equations for arbitrary clique size; interested readers are advised to see the details in PhD thesis of Haines [3]. Given that a factor is connected to a set of random variables $t \in T$ where each variable has a state $s_t \in S_t$, the exteded equation of message passing to variable u is given by,

$$M_{r \to u}(s_u) = \min_{S_t; t \in T-u} \left(f(s_t; t \in T) + \sum_{t \in T-u} M_{t \to r}(s_t) \right) \tag{4.12}$$

where $\min_{S_t; t \in T-u}$ represents the minimization over all the states of all the connected random variables except the one we are sending the message to. f is the factor cost equation and requires the states of all the random variables as input. Now, the message passed from random variable u to a factor will be the sum of all the messages received from all the factors u is connected to. If $f \in F$ is the set of factors u is connected to and g is the factor we are sending the message from u, following the convention of equation (4.12) the message passed would be,

$$M_{u \to g}(s_u) = \sum_{f \in F-g} M_{f \to u} \tag{4.13}$$

Once, the belief propagation iterated a number of times the final belief at each node is given by the equation,

$$B_u(s_u) = \sum_{f \in F} M_{f \to u} \tag{4.14}$$

4.3.3 Continuous Formulation

With understanding of discrete formulation the continuous sum-product belief propagation can be described by just modifying equations of discrete formulation. So the equation describing the message passed from a factor to a variable will become,

$$M_{r\to u}(s_u) \propto \int_{S_t;t\in T-u} \left(f(s_t; t \in T) \prod_{t\in T-u} M_{t\to r}(s_t) \right) \tag{4.15}$$

Message passed from variable to factor would be,

$$M_{u\to g}(s_u) \propto \prod_{f\in F-g} M_{f\to u} \tag{4.16}$$

The final belief for each node after convergence would be,

$$B_u(s_u) \propto \prod_{f\in F} M_{f\to u} \tag{4.17}$$

For continuous formulation we have replaced summation with product; this is due to the fact that we are now dealing with probabilities instead of costs so the message passed over the variables and factors would be probability distribution and that is why we have not listed the normalization. The summation has been replaced with integral due to the example under consideration being continuous. The sum-product algorithm sums all the distributions and hence calculates the marginal.

4.4 Shape-from-Shading Algorithm

To describe shape-from-shading Haines [3] built a graphical model specifically a pairwise Markov random field on grid as shown in Figure (4.4). Each node of the graphical model is a random variable that represents the unknown surface orientation. For each node the prior is given by the cone constraint using the Bingham-Mardia distribution and the compatability between adjacent nodes provide smoothing using a Fisher distribution.

The pairwise MRF is given by,

$$P(\mathbf{x}) = \prod_{v\in V} \psi_v(\mathbf{y}_v) \tag{4.18}$$

where $\mathbf{x}$ is a set of random variables i.e. nodes of graphical model and $\forall v$; $\mathbf{y}_v \subset \mathbf{x}$. This is the equation (4.8) with additional requirement that ψ can not involve more than two variables. The belief propagation described above finds a solution that approximately maximises $P(\mathbf{x})$ marginalized over all involved random variables. The message passed from node p to q at iteration t is then given by,

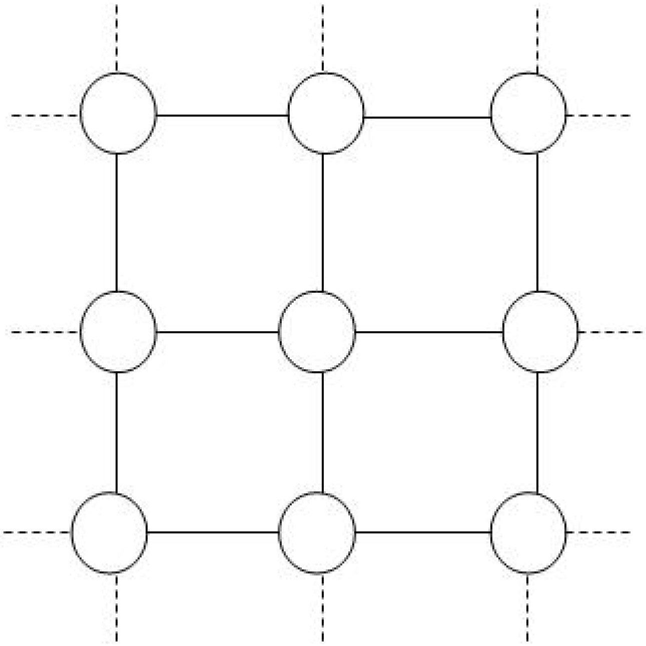

Figure 4.4: Markov Random Field: Grid Configuration

$$m^t_{p\to q}(\hat{\mathbf{x}}_q) = \int_{\hat{\mathbf{x}}_p} \psi_{pq}(\hat{\mathbf{x}}_p, \hat{\mathbf{x}}_q)\psi_p(\hat{\mathbf{x}}_p) \prod_{u\in(N-q)} m^{t-1}_{u\to p}(\hat{\mathbf{x}}_p) \tag{4.19}$$

In the equation above $\psi_{pq}(\hat{\mathbf{x}}_p, \hat{\mathbf{x}}_q)$ is the compatability between adjacent nodes p and q, and $\psi_p(\hat{\mathbf{x}}_p)$ is the prior on each node's orientation. N represents the neighbouring nodes. The prior on each node as stated before is given by the cone constraint and the compatibility between adjacent nodes expresses the smoothness. Haines [3] used the hierarchical method of Felzenszwal and Huttenlocher [18] in which instead of solving problem directly it was solved at lower resolution. The messages from lower resolution levels are then used to initialize the higher resolution levels. Once the belief propagation has converged the final belief (FB$_8$) at each node is given by using the equation,

$$b_p(\hat{\mathbf{x}}_p) = \psi_p(\hat{\mathbf{x}}_p) \prod_{u\in N} m_{u\to p}(\hat{\mathbf{x}}_p) \tag{4.20}$$

The graphical representation of this message passing for nodes p and q is shown below in figure (4.5).

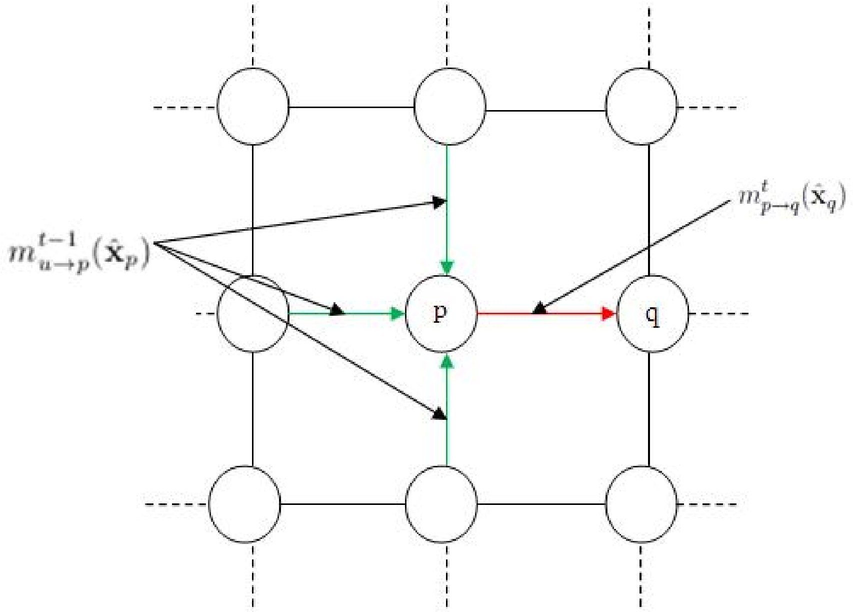
$m^{t-1}_{u\to p}(\hat{\mathbf{x}}_p)$
$m^{t}_{p\to q}(\hat{\mathbf{x}}_q)$
p
q

Figure 4.5: Message Passing over MRF

The actual output of Shape-from-Shading algorithm should be a direction rather than a distribution per node. Haines gave a sum-product formulation of belief propagation to choose among the two maximas of FB$_8$ distribution described later.

4.4.1 Priors on Nodes

Using irradiance equation i.e. equation (4.2) and Bingham-Mardia distribution given by equation (4.6) the cone constraint can be captured with a Bingham-Mardia distribution which provides prior for each node(pixel),

$$\Omega[2k_i \frac{I}{A} \hat{I}, -k_i \hat{I} \hat{I}^T] \qquad (4.21)$$

where I is the irradiance at a specific node(pixel) and $\hat{I}$ is the direction of unit light source as described earlier. Equation (4.21) can be used to assign priors to each node however Haines used boundary and gradient information as well before assigning prior to the nodes. The concentration parameter k_i in above equation is set on the fact that the extreme irradiance values (specular and non-lambertian areas) are less reliable as at these regions the lambertian assumption of shape-from-shading is compromised.

The surface normal will usually be on the disc containing the gradient and direction to the light source direction. Haines used a disc distribution rather than a direction as used by both Zheng & Chellappa [19] and Worthington & Hancock [13] so not biasing concave or convex regions. Given a normalised gradient direction $\hat{g}$ which is in the image plane $g_z = 0$ a Bingham distribution can be defined,

$$\Omega[0, -k_g \hat{d}\hat{d}^T] \tag{4.22}$$

where $\hat{d} = \frac{\hat{g}*\hat{I}}{|\hat{g}*\hat{I}|}$. The concentration parameter k_g is proportional to the gradient strength. Haines used boundary constraint to bias towards a convex or concave solution. The edge of an object is tangential to both the viewing direction and curve of the boundary. Boundaries are detected as non-zero pixels adjacent to pixels with zero values. A Fisher distribution was used to represent the boundary information.

$$\Omega[k_b b\hat{t}, 0] \tag{4.23}$$

where, b is 1 for boundaries and 0 elsewhere. Tangential direction $\hat{t}$ is $-\hat{g}$ i.e. negative gradient direction for object edges and $\frac{\hat{I}*\hat{d}}{|\hat{I}*\hat{d}|}$ for shadow edges. The concentration parameter k_b is constant over the image, a positive value for a convex bias and a negative bias for a concave bias. The equations $(4.21 - 4.23)$ are combined according to equation (4.5) to give prior for each node(pixel).

$$\psi_p(\hat{\mathbf{x}}_p) = \Omega[2k_i \frac{I}{A}\hat{I} + k_b b\hat{t}, -k_i \hat{I}\hat{I}^T - k_g \hat{d}\hat{d}^T] \tag{4.24}$$

4.4.2 Compatability between Adjacent Nodes

Adjacent pixels have small angular differences among their orientations, this is known as the smoothness constraint. The compatability between adjacent pixels can be described using a Fisher distribution,

$$\psi_{pq}(\hat{\mathbf{x}}_p, \hat{\mathbf{x}}_q) \propto \exp\left(k_s(\hat{\mathbf{x}}_p^T \hat{\mathbf{x}}_q^T)\right) \tag{4.25}$$

By substituting the Fisher distribution in the message passing equation (4.22),

$$m_{p\to q}^{t}(\hat{\mathbf{x}}_q) = \int_{S^2} \exp\left(k_s(\hat{\mathbf{x}}_p^T\hat{\mathbf{x}}_q^T)\right) t(\hat{\mathbf{x}}_p)\delta\hat{\mathbf{x}}_p \tag{4.26}$$

where,

$$t(\hat{\mathbf{x}}_p) = \psi_p(\hat{\mathbf{x}}_p) \prod_{u\in(N-q)} m_{u\to p}^{t-1}(\hat{\mathbf{x}}_p) \tag{4.27}$$

Hence the message passing consists of first multiplying FB_8 distributions and then convolving the resultant FB_8 distribution with a Fisher distribution. The result of convolution is not a Fisher-Bingham distribution so Haines [3] provided an approximation to Fisher-Bingham distribution; we however will not list the details of this approximation here. The concentration parameter k_s is found numerically and is highest for the areas of similar irradiance and lowest for nodes having greatly differing irradiance.

From Haines Shape-from-Shading algorithm we get surface orientations described by Fisher-Bingham distributions for every pixel. We then sample each of these Fisher-Bingham distributions using slice sampling algorithm of Kume & Walker [11] as described in Chapter 4.

4.5 Shape Model of Surface Normals

Smith [1] constructed statistical shape models of needle maps and surface heights by following the analogy of the Cootes active shape model [52]. In this section we will partially describe how Smith built his model and later we will show how we are using these models to combine them with the shape from shading algorithm of Haines [3]. Smith [1] described two methods for building the shape model of surface normals. One is based on Azimuthal Equidistant Projection(AEP) and Principal Component Analysis(PCA) while the other one is build using the concepts of Log & Exponential maps and Principal Geodesic Analysis(PGA). We will first go into the details of model based on AEP and PCA and then extend this description for the other method based on Log/Exp mapping and PGA.

4.5.1 Training Data

The model was constructed using the range images from the 3DFS data base [4] and Max Plank data base [5] . The range images have been aligned so that each specific pixel location in all the images addresses the same point. From the range images the fields of surface normals have been extracted using orthographical projection. So, Smith [1] had a training set comprising K needle maps where $n_k(x,y) = [n_k(x,y)_x, n_k(x,y)_y, n_k(x,y)_z]^T$ is the unit surface normal at pixel location (x,y) in kth training image.

4.6 PCA based Statistical Surface Normal Model

4.6.1 Azimuthal Equidistant Projection

In directional statistics the average of a set of unit vectors e.g. surface normals is named as *mean direction* and is given as,

$$\hat{n}_0(x,y) = \frac{\overline{n}_0(x,y)}{\|\overline{n}_0(x,y)\|} \tag{4.28}$$

where,

$$\overline{n}_0(x,y) = \frac{1}{K}\sum_{k=1}^{K} n_k(x,y). \tag{4.29}$$

There are other more suitable measures for the average of directional data if they are considered as points lying on sphere, called the spherical median [6] . We will describe this term later when we will address the model based on PGA. On the unit sphere a unit surface normal at location (x,y), $n_k(x,y)$ has the elevation angle of $\theta_k(x,y) = \frac{\pi}{2} - \arcsin[n_k(x,y)_z]$ and an azimuth angle $\alpha_k(x,y) = \arctan\frac{n_k(x,y)_y}{n_k(x,y)_x}$ while the mean surface normal at the same location has an elevation angle $\theta_0(x,y) = \frac{\pi}{2} - \arcsin[\hat{n}_0(x,y)_z]$ and azimuth angle $\alpha_0(x,y) = \arctan\frac{\hat{n}_0(x,y)_y}{\hat{n}_0(x,y)_x}$.

In order to transform the field of surface normals into a representation in which the standard PCA could be applicable Smith used the azimuthal equidistant projection. He commences AEP by constructing a tangent plane to the unit sphere at the location corresponding to the mean surface normal. A local co-ordinate system to this tangent plane is established. The origin is at the point of contact between the tangent plane and the unit sphere. The x-axis is aligned parallel to the local circle of latitude on the unit sphere.

Under the AEP at the location (x,y) the surface normal $n_k(x,y)$ maps to the point with co-ordinate vector $v_k(x,y) = (x_k(x,y), y_k(x,y))^T$.The transformations between the tangent plane and the unit sphere are given by the following equations,

$$x_k(x,y) = k' \cos\theta_k(x,y) \sin[\alpha_k(x,y) - \alpha_0(x,y)] \tag{4.30}$$

$$y_k(x,y) = k' \left(\cos\theta_0(x,y) \sin\alpha_k(x,y) - \sin\theta_0(x,y) \cos\theta_k(x,y) \cos[\alpha_k(x,y) - \alpha_0(x,y)]\right)$$

$$\cos c = \sin\theta_0(x,y) \sin\theta_k(x,y) + \cos\theta_0(x,y) \cos\theta_k(x,y) \cos[\alpha_k(x,y) - \alpha_0(x,y)]$$

$$k' = \frac{c}{\sin c}$$

The figure (4.6) gives a graphical representation of Azimuthal Equidistant Projection.

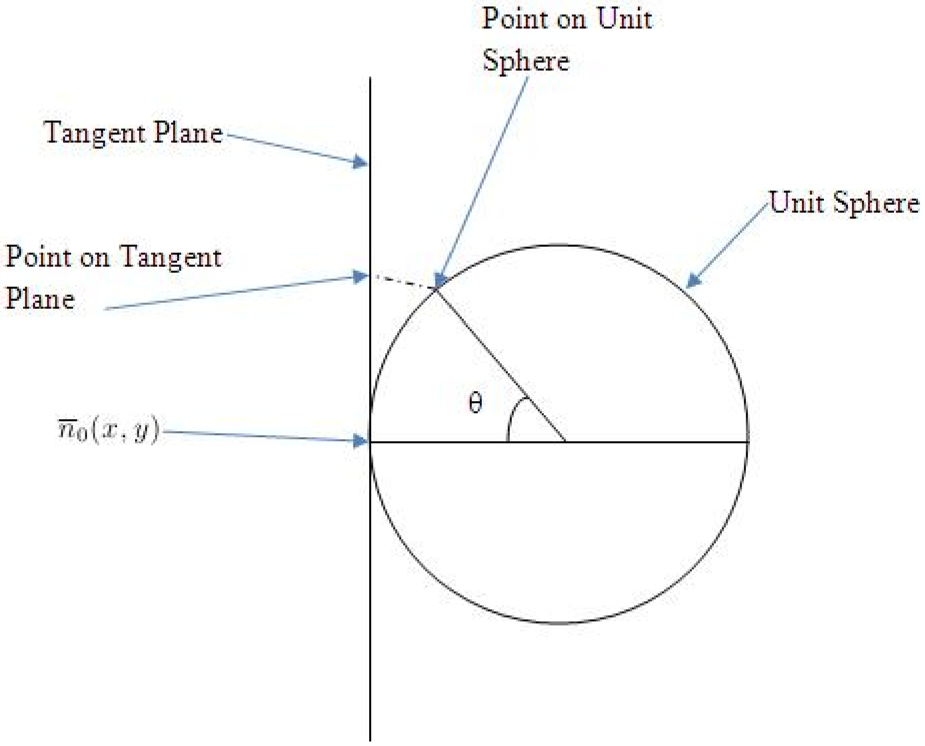

Figure 4.6: Azimuthal Equidistant Projection

The equations for the inverse azimuthal equidistant projection to transform from the tangent plane to the unit sphere are given by,

$$\theta_k(x,y) = \arcsin(\cos c \sin\theta_0(x,y) - \frac{1}{c} y_k(x,y)\sin c\cos\theta_0(x,y)) \tag{4.31}$$

$$\alpha_k(x,y) = \alpha_0(x,y) + \arctan\omega(x,y)$$

where,

$$\omega(x,y) = \begin{cases} \frac{x_k(x,y)\sin c}{c\cos\theta_0(x,y)\cos c - y_k(x,y)\sin\theta_0(x,y)\sin c}; & \theta_0(x,y) \neq \pm\frac{\pi}{2} \\ -\frac{x_k(x,y)}{y_k(x,y)} & ;\theta_0(x,y) = +\frac{\pi}{2} \\ \frac{x_k(x,y)}{y_k(x,y)} & ;\theta_0(x,y) = -\frac{\pi}{2} \end{cases} \tag{4.32}$$

and

$$c = \sqrt{x_k(x,y)^2 + y_k(x,y)^2}$$

4.6.2 Point Distribution Model

Each range image in the training data is converted to a field of surface normals, so the surface normal at pixel location (x, y) in kth training image is given by $n_k(x,y) = (n_k(x,y)_x, n_k(x,y)_y, n_k(x,y)_z)^T$. Using the azimuthal equidistant projection all the surface normals are converted to the vectors, like $v_k(x,y) = (x_k(x,y), y_k(x,y))^T$ is the vector on the tangent plane for surface normal at pixel location (x, y) in the kth training sample. If the range image has a resolution of $N = X_{res} \times Y_{res}$ pixels arranged in X_{res} rows and Y_{res} columns, its surface normal coordinates can be represented in a long vector $\mathbf{U}^k$ of length $2N$ ordered according to the raster scan,

$$\mathbf{U}^k = [x_k(1,1), x_k(1,2), ...x_k(X_{res}, Y_{res}), y_k(1,1), y_k(1,2), ..., y_k(X_{res}, Y_{res})]^T \tag{4.33}$$

So, in the above vector $\mathbf{U}^k$ the 1st N elements are the x-coordinates of the surface normals obtained by applying azimuthal equidistant projection and 2nd N elements are the y-coordinates; stacked into a column vector. Since the AEP involves centering the local coordinate system, the coordinates corresponding to the mean direction are $(0, 0)$ at each image location. Hence the long vector corresponding to the mean direction at each image location is zero. The K training samples are used to form the $2N \times K$ matrix,

$$\mathbf{D} = [\mathbf{U}^1|\mathbf{U}^2...|\mathbf{U}^k]. \tag{4.34}$$

The $2N \times 2N$ covariance matrix is given by,

$$\mathbf{L} = \frac{1}{K}\mathbf{D}\mathbf{D}^T. \tag{4.35}$$

Since $2N >> K$ Smith used the numerically efficient snap-shot method of Sirovich [1, 6] to calculate the principal directions of $\mathbf{L}$. Instead of finding the eigenvectors and eigenvalues of $\mathbf{L}$ the eigenvalues and eigenvectors of $\hat{\mathbf{L}} = \frac{1}{K}\mathbf{D}^T\mathbf{D}$ are found. The eigenvectors of $\hat{\mathbf{L}}$ are used to calculate the eigenvectors of $\hat{\mathbf{L}}$ by using the relationship, $\Psi_i = \mathbf{D}\hat{\Psi}_i$ where Ψ_i is the ith eigenvector of the matrix $\mathbf{L}$ and $\hat{\Psi}_i$ is the ith eigenvector of $\hat{\mathbf{L}}$. Similarly for $i \leq k$, $\lambda_i = \hat{\lambda}_i$ where λ is the eigenvalue. For $i > k$ eigenvalues for $\mathbf{L}$ are taken as zeros. The matrix comprised of eigenvectors of matrix $\mathbf{L}$ is then formed by stacking the eigenvectors as columns i.e. $\mathbf{P} = [\Psi_1|\Psi_2|\dots|\Psi_k]$.

4.7 PGA based Statistical Surface Normal Model

In Euclidian space shape variations are analysed using linear approaches i.e. linear averaging and PCA. When the points lying on a non-linear space, e.g. a sphere, are considered then concepts of spherical median and PGA [7] are used. A linear combination of unit vectors is not a unit vector so the analysis of surface normals cannot be effected in a linear way. In the previous section, we have shown how Smith addressed this problem by using the azimuthal equidistant projection; so he transformed the field of surface normals to the tangent plane where the linear PCA was applied to get the eigenvectors and eigenvalues for the model. Here Smith used tools of differential geometry and provided a different definition for average facial needle map. He chose to model a distribution of surface normals as a distribution of points lying on the spherical manifold and exploited the Principal Geodesic approach of Fletcher et al. [7] .

4.7.1 The Log and Exponential Maps

A unit vector $\mathbf{n} \in R^3$ may be considered as a point lying on a spherical manifold $n \in S^2$ where S^2 is a unit sphere. The two are related by the embedding $\Phi : S^2 \longmapsto R^3$, so $n = \Phi(\mathbf{n})$. If $v \in T_nS^2$ is a vector on the tangent plane to S^2 at $n \in S^2$ and $v \neq 0$, the exponential map $\exp_n$ of v is the point on the sphere S^2 along the geodesic in the direction of v at distance $\|v\|$ from n. Geometrically this is equivalent to mark a length $\|v\|$ along the geodesic that passes through n in the direction of v. The inverse of the exponential map is the logarithmic map, denoted by $\log_n$. So, in essence the exponential map takes a unit vector in the Euclidian space and converts to a point on a spherical manifold whereas the log map takes a point on the unit sphere and converts it to a unit vector on the tangent plane.

It is apparent that the log map is analogous to the azimuthal equidistant projection whereas exponential mapping is analogous to the inverse azimuthal equidistant projection.The geodesic distance between two points on the spherical manifold $n_1, n_2 \in S^2$ can be given by the log map i.e. $d(n_1, n_2) = \|\mathrm{Log}_{n_1}(n_2)\|$.

4.7.2 Spherical Median

If the unit vectors in the Euclidian space are considered as points distributed on a spherical manifold $n_1, n_2, ..., n_k \in S^2$ where $\Phi(n_k) = \mathbf{n_k}$; the mean direction is dependent on the embedding Φ and is the *extrinsic mean* of a distribution of spherical data; given by

$$\mu_\Phi = \underset{n \in S^2}{\mathrm{argmin}} \sum_{i=1}^{K} \|\Phi(n) - \Phi(n_i)\|^2 \tag{4.36}$$

If the projection from Euclidian to Spherical space $\pi : R^3 \longmapsto S^2$ is defined as,

$$\pi(\mathbf{n}) = \underset{n \in S^2}{\mathrm{argmin}} \|\Phi(n) - \mathbf{n}\|^2 \tag{4.37}$$

then it can be shown that the Euclidian average of the points on R^3 projected back on the spherical manifold is the extrinsic mean i.e.

$$\mu_\Phi = \pi(\overline{\mathbf{n}}) = \pi\left(\frac{1}{K}\sum_{i=1}^{K} \Phi(n_i)\right) \tag{4.38}$$

A more natural definition of the average of points on the unit sphere uses arc length as the choice of distance measure. This is given by the Riemannian distance $d(.,.)$ between a pair of points, $d(n_1, n_2) = \arccos(\Phi(n_1).\Phi(n_2))$. Using this definition of distance, the *intrinsic mean* will be a point that minimizes the distance to all the points on the sphere,

$$\mu = \underset{n \in S^2}{\mathrm{argmin}} \sum_{i=1}^{K} d^2(n, n_i) \tag{4.39}$$

For data lying on the sphere this *intrinsic mean* is named as *spherical median* [7] . This point cannot be found analytically, but can be solved iteratively using the gradient descent method of Pennec [8]. The estimate of this point is initialized with the extrinsic mean of the distribution, i.e. $\mu_{(0)} = \mu_\Phi$ and then updated iteratively by,

$$\mu_{j+1} = \mathrm{Exp}_{\mu_{(j)}} \left(\frac{1}{K} \sum_{i=1}^{K} \mathrm{Log}_{\mu_{(j)}} (n_i) \right) \tag{4.40}$$

4.7.3 PGA versus PCA

In PCA, we do linear dimension reduction to represent the high dimensional data in reduced space. In case of PGA, the data lies on the spherical manifold so the reduced lower dimensional non-linear space will be a sub manifold. In other words where in PCA each principal axis is a straight line; in PGA each principal axis is a geodesic curve. In case of a spherical manifold this geodesic curve will be a great circle. In PCA linear projection is used to project data from higher dimension to lower dimension. In PGA a different type of projection operator is required. Suppose H is a submanifold of a Riemannian manifold M. In order to project $x \in M$ onto a point on H that is closest to x in Riemannian distance Smith use the projection $\pi_H : M \longmapsto H$ given by

$$\pi_H(x) = \operatorname*{argmin}_{y \in H} d(x,y)^2 \tag{4.41}$$

If $H \subset M$ is a geodesic submanifold at point μ we can approximate π_H linearly on the tangent space of M at μ, $T_\mu M$. If $v_1, v_2, ..., v_k$ is the orthonormal basis for $T_\mu H$, then the projection operator π_H can be approximated in the tangent plane using,

$$\mathrm{Log}_\mu \left(\pi_H(x)\right) \approx \sum_{i=1}^{K} v_i \left(v_i . \mathrm{Log}_\mu (x) \right) \tag{4.42}$$

Fletcher et al. [7] observed that PGA can be approximated by applying the standard linear PCA by projecting spherical data to the tangent plane. Like PCA the objective of PGA is to find the nested geodesic submanifold that account for the decreasing amounts of variance in data. The principal geodesics $U_1, U_2, ..., U_k \in S^2(N)$ are defined by first constructing an orthonormal basis of tangent vectors $v_1, v_2, ..., v_k \in T_\mu S^2(N)$. These are the principal directions on the tangent plance $T_\mu S^2(N)$. The first principal direction is found using approximation of (4.42)

$$v_1 \approx \operatorname*{argmax}_{\|v\|=1} \sum_{i=1}^{N} \left(v . \mathrm{Log}_\mu (U_i) \right)^2 \tag{4.43}$$

The remining principal directions are updated using the approximation,

$$v_k \approx \underset{\|v\|=1}{\arg\max} \sum_{i=1}^{N} \sum_{j=1}^{K-1} (1 - \|v_j.v\|)^2 + \left(v.\mathrm{Log}_\mu\left(U_i\right)\right)^2 \tag{4.44}$$

In essence Smith transforms the distribution of unit vectors to points on the tangent plane to $S^2(N)$ at μ and applies standard linear PCA.

4.7.4 PGA of Needle Maps

For his model Smith used K training samples each comprised of a depth image from which he got the surface normals. The surface normal at pixel location (x, y) in kth training image is given by $n_k(x, y) = (n_k(x, y)_x, n_k(x, y)_y, n_k(x, y)_z)^T$. Smith calculates the *spherical median* $\mu_{(x,y)}$ at each pixel location (x, y) of the K surface normals from the training set using Equation (4.40). The surface normal at position (x, y) in kth training image is represented on the tangent plane $T_{\mu_{(x,y)}}S^2$ given by the log map, $v^k_{(x,y)} = \mathrm{Log}_{\mu_{(x,y)}}\left(\mathbf{n}^k_{(x,y)}\right) \in R^2$. From the vectors on the tangent plane the data matrix $\mathbf{D}$ is formed,

$$\mathbf{D} = [\mathbf{v}^1|\mathbf{v}^2|...|\mathbf{v}^k] \tag{4.45}$$

where, $\mathbf{v}^k = [v^k_{(1,1)}, v^k_{(1,2)}, ...v^k_{(X_{res},Y_{res})}]$

From the data matrix the covariance matrix is calculated,

$$\mathbf{L} = \frac{1}{K}\mathbf{D}\mathbf{D}^T \tag{4.46}$$

Smith finds the eigenvalues and eigenvectors of the covariance matrix $\mathbf{L}$ and perform PGA. Just like PCA the matrix comprised of eigenvectors of covariance martix is constructed as $\mathbf{P} = [\Psi_1|\Psi_2|\dots|\Psi_k]$.

4.8 Statistical Surface Height Model

We will now list the details of surface height model of Smith [1] that we use to infer surface heights from surface normals.

4.8.1 Relationship between Surface Height and Surface Normals

Recovering surface height from a field of surface normals or surface gradients is named as surface integration or the height from gradient problem. A field of surface normals is said to be integrable if the surface gradient $p_{(x,y)} = \frac{n_{(x,y)x}}{n_{(x,y)_z}}, q_{(x,y)} = \frac{n_{(x,y)y}}{n_{(x,y)_z}}$ recovered from surface normal $n_{(x,y)} = [n_{(x,y)_x}, n_{(x,y)_y}, n_{(x,y)_z})]^T$; follow the integraibility constraint given by,

$$\frac{\partial p_{(x,y)}}{\partial y} = \frac{\partial q_{(x,y)}}{\partial x}$$

Given the surface normals follow the integrability constraint getting the surface height is straightforward. It commences by initializing the surface height to a chosen starting point and then an arbitrary path of integration is followed through the field of surface normals and the height is summed from the surface gradients given by the surface normals. There are many approaches to solve these problems; the most famous being the global method of Frankot and Chellapa [9]. If the surface normals are noisy, difficulties arise while using these algorithm. We will not go into the details of different local and global surface integration method;the interested reader is suggested to look on section 5.1 of Smith's PhD thesis [1]. We will however use surface height model of Smith which gives the surface height implicitly without following any explicit surface integration method. The details of this model and its usability are given next.

4.8.2 Surface Height Model

The training set is comprised of K range images of size $N = X_{res} * Y_{res}$ pixels. Each training set can be represented by a long vector containing the depth at each pixel location stacked into a column vector according to the raster scan e.g. the long vector for kth training image would be,

$$\mathbf{z}^k = [z^k(1,1), z^k(1,2), ..., z^k(X_{res}, Y_{res})] \tag{4.47}$$

The mean surface height vector $\hat{\mathbf{z}}$ is calculated by $\hat{\mathbf{z}} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{z}^i$. Just like surface normal model Smith form the $N * K$ data matrix $\mathbf{D}_h$ of surface heights,

$$\mathbf{D}_h = [(\mathbf{z}^1 - \hat{\mathbf{z}})|(\mathbf{z}^2 - \hat{\mathbf{z}})|...|(\mathbf{z}^k - \hat{\mathbf{z}})] \tag{4.48}$$

Smith calculates the covariance matrix $\mathbf{L}_h$ of surface height and then apply the PCA to get the K orthogonal eigenvectors i.e. the pricipal modes of variation.

$$\mathbf{L}_h = \frac{1}{K}\mathbf{D}_h\mathbf{D}_h^T \tag{4.49}$$

4.8.3 Height from Normals

As mentioned previously there are number of algorithms available for surface integration but we will only focus on how Smith's surface height model can be used to get surface height from surface normals. We can express the facial surface $F \in R^3$ projected orthographically on the image plane and parameterized by the function $z(x, y)$ in terms of base surface $\hat{z}(x, y)$ plus a linear combination of K modes of variation Ψ_i,

$$z_b(x, y) = \hat{z}(x, y) + \sum_{i=1}^{K} b_i\Psi_i(x, y) \tag{4.50}$$

where b_i are the surface parameters. Conversely we can also express the surface normal at location (x, y) as sum of the gradient of the mean surface height $\hat{z}(x, y)$ and a linear combination of the gradients of the eigenvectors at the corresponding point.

$$\mathbf{n}_b(x, y) = \left\{ \begin{array}{c} \frac{\partial\hat{z}(x,y)}{\partial x} + \sum_{i=1}^{K} b_i\frac{\partial\Psi_i(x,y)}{\partial x} \\ \frac{\partial\hat{z}(x,y)}{\partial y} + \sum_{i=1}^{K} b_i\frac{\partial\Psi_i(x,y)}{\partial y} \\ 1 \end{array} \right\} \tag{4.51}$$

We can write (4.51) neatly if we define, $\hat{p}(x, y) = \frac{\partial\hat{z}(x,y)}{\partial x}$, $\hat{q}(x, y) = \frac{\partial\hat{z}(x,y)}{\partial y}$, $\hat{p}_i(x, y) = \frac{\partial\Psi_i(x,y)}{\partial x}$ and $\hat{q}_i(x, y) = \frac{\partial\Psi_i(x,y)}{\partial y}$ i.e. the partial derivatives of average surface height and eignevectors with respect to x and y. Equation (4.51) can now be written as,

$$\mathbf{n}_b(x, y) = \left\{ \begin{array}{c} \hat{p}(x, y) + \sum_{i=1}^{K} b_i\hat{p}_i(x, y) \\ \hat{q}(x, y) + \sum_{i=1}^{K} b_i\hat{q}_i(x, y) \\ 1 \end{array} \right\} \tag{4.52}$$

To get surface height $z(x, y)$ from a specific field of surface normals $\mathbf{n}(x, y)$; we first have to calculate the parameter vector b^* whose field of surface normals given by equation (4.52) minimises the distance to $\mathbf{n}(x, y)$. This is effectively the popular linear least squares problem. In next section we will describe this problem and its solution using matrix calculus.

4.8.4 Linear Least Squares and Normal Equation

The surface gradients of $\mathbf{n}(x, y)$ are given by $p(x, y) = \frac{n_x(x,y)}{n_z(x,y)}$ and $q(x, y) = \frac{n_y(x,y)}{n_z(x,y)}$. Using this definition of gradients and (4.52) we have to find b^* such that,

$$b^* = \underset{b}{\operatorname{argmin}} \sum_{x,y} \left[\left[\hat{p}(x, y) + \sum_{i=1}^{K} b_i \hat{p}_i(x, y) - p(x, y) \right]^2 + \left[\hat{q}(x, y) + \sum_{i=1}^{K} b_i \hat{q}_i(x, y) - q(x, y) \right]^2 \right] \tag{4.53}$$

If the field of surface normals is of $N = X_{res} \times Y_{res}$ dimensions; Smith subtracts the mean surface gradients from the surface gradients of surface normals and forms a vector $\mathbf{G}$ of length $2N$,

$$\mathbf{G} = \left\{ \begin{array}{c} p(1,1) - \hat{p}(1,1) \\ q(1,1) - \hat{q}(1,1) \\ ... \\ ... \\ p(X_{res}, Y_{res}) - \hat{p}(X_{res}, Y_{res}) \\ q(X_{res}, Y_{res}) - \hat{q}(X_{res}, Y_{res}) \end{array} \right\} \tag{4.54}$$

Next he forms the matrix $\mathbf{\Psi}$ of $2N \times K$ dimensions of surface gradients of eigen vectors where the kth column of this matrix is given by, $\Psi_k = [p_k(1,1), q_k(1,1), ...p_k(X_{res}, Y_{res}), q_k(X_{res}, Y_{res})]$. The *Least-Square cost function* is now given in the matrix form,

$$\mathbf{J}(b) = \frac{1}{2} \|\mathbf{\Psi} b - \mathbf{G}\|^2 \tag{4.55}$$

The scalar term $\frac{1}{2}$ in Equation (4.55) is being used only to make the derivation easy. We need to find surface parameter vector b^* such that $\mathbf{J}(b)$ is minimised i.e. $b^* = \operatorname{argmin}_b \mathbf{J}(b)$. This can be done by finding the derivative of $\mathbf{J}(b)$ and setting it to zero.

$$\bigtriangledown_b \mathbf{J}(b) = \mathbf{\Psi}^T \mathbf{\Psi} b - \mathbf{\Psi}^T \mathbf{G}$$

Now, setting this derivative equal to zero we get the *Normal equation* [10],

$$\mathbf{\Psi}^T \mathbf{\Psi} b = \mathbf{\Psi}^T \mathbf{G} \tag{4.56}$$

The complete derivation of Normal equation is listed in Appendix for interested readers. Thus value of surface parameters b^* that minimize $\mathbf{J}(b)$ is given by [1],

$$b^* = (\mathbf{\Psi}^T \mathbf{\Psi})^{-1} \mathbf{\Psi}^T \mathbf{G} \tag{4.57}$$

By using these surface parameters we calculate surface height using Equation (4.50) without explicitly integrating the surface normls.

Chapter 5

Sampling the Fisher Bingham Distribution

We have sampled Fisher-Bingham distribution for each pixel using the sampling technique of A. Kume & S.G. Walker [11]. In this chapter we will describe this sampling algorithm but before that we will list some preliminaries regarding Fisher-Bingham distribution, Gibbs sampler and slice sampling.

5.1 Preliminaries

5.1.1 Fisher Bingham Distribution

The Fisher-Bingham distribution is a multivariate normal distribution that is constrained to lie on the surface of a unit sphere. The Fisher-Bingham distribution is used to model the directional data on spheres and sometimes for shape analysis as Haines used it for Shape-from-Shading [3]. If $\mathbf{x} = (x_0, x_1, x_2)$ is a random variable from this distribution then according to the unit norm constraint $\|x\|^2 = 1$. Hence, $\mathbf{x}^2 = (x_0^2, x_1^2, x_2^2)$ lies on a simplex. The contribution of Kume et al. [11] is to transform $\mathbf{x}$ to (ω, s) and study the marginal and conditional distributions of ω and s; where $s = x_i^2$ and $\omega_i = \frac{x_i}{\|x_i\|}$, so ω can either be 1 or -1.The Fisher Bingham distribution which is obtained on unit sphere S^2 by constraining the multivariate normal distribution belonging to R^3 can be given with respect to the mean vector μ and inverse of covariance matrix Σ as,

$$f(x|\mu,\Sigma) = N(\mu,\Sigma)\exp(-(x-\mu)^T\Sigma(x-\mu)) \tag{5.1}$$

where, $N(\mu,\Sigma)$ is the normalizing constant, $x \in R^3$ and $x^Tx = 1$. We take that the inverse of the covariance matrix as diagonal without loss of generality i.e. $\Sigma = \Lambda = diag(\lambda_0, \lambda_1, \lambda_2)$. Since, x^Tx constraint is satisfied, the $\exp(-(x-\mu)^T\Sigma(x-\mu))$ can be expressed in $(\omega_0, \omega_1, \omega_2)$ and (s_1, s_2).

$$\begin{aligned}\exp(-(x-\mu)^T\Sigma(x-\mu)) &= x^T\Sigma x - 2x^T\Sigma\mu + \mu^T\Sigma\mu \\ &= \sum_{i=0}^{2}\lambda_i s_i - 2\sum_{i=0}^{2}\lambda_i\omega_i\mu_i\sqrt{s_i} + \mu^T\Sigma\mu\end{aligned}$$

$$\exp(-(x-\mu)^T\Sigma(x-\mu)) = \sum_{i=0}^{2}(-a_i s_i - b_i\omega_i\sqrt{s_i}) - b_0\omega_0\sqrt{1-s} + c \tag{5.2}$$

where, $a_i = \lambda_0 - \lambda_i$ for $i = 1, 2$; for $i = 0...2$, $b_i = 2\lambda_i\mu_i$, while $c = \mu^T\Sigma\mu + \lambda_0$ and $s = 1 - s_0 = s_1 + s_2$. Using the transformation given by Equation(5.2) the joint density of interest is given by,

$$f(\omega, s) \propto \exp\left[\sum_{i=0}^{2}(a_i s_i + b_i\omega_i\sqrt{s_i})\right] \times \exp\left[b_0\omega_0\sqrt{1-s}\right] \times \prod_{i=0}^{2}\frac{1}{\sqrt{s_i}}\frac{1}{\sqrt{1-s}}\mathbf{1}(s \leq 1) \tag{5.3}$$

where, $\mathbf{1}(s \leq 1)$ is the *indicator* variable. We take λ_0 to be the largest of all $\lambda' s$ so that $\lambda_i \geq 0$ for $i = 1$ and 2.

5.1.2 Slice Sampling

Suppose we want to sample from a distribution for a random variable say x, by taking values in some subset of R^n while the probability density function is proportional to some function $f(x)$ [54] . We can do this by sampling uniformly from the $n+1$ dimensional region that lies under the plot of $f(x)$. This can be done by introducing an auxiliary variable y and defining a joint distribution over x and y that is uniform over the region $U = \{x, y : 0 < y < f(x)\}$ under the surface $f(x)$. The joint density for (x,y) is then given as,

$$p(x,y) = \begin{cases} \frac{1}{Z}; & 0 < y < f(x) \\ 0; & \text{otherwise} \end{cases}$$

where $Z = \int f(x)\text{dx}$. The marginal distribution of x can be found by,

$$p(x) = \int_0^{f(x)} \frac{1}{Z} \quad \mathrm{dy} = \frac{f(x)}{Z}$$

The distribution can be sampled jointly for (x, y) and then y can be ignored to get samples for x only. Generating independent points drawn uniformly from region U may not be easy, so a Markov chain can be defined that will converge to this uniform distribution. Gibbs sampling is one of the possibilities which is presented next.

5.1.3 Gibbs Sampling

The Gibbs sampling is a technique for generating random vairables from a marginal distribution indirectly, without having to calculate the density [12]. This algorithm generates samples from a joint distribution of two or more random variables. These samples can be used to approximate the joint density of random variables. Gibbs sampling is used when the joint distribution is not explicitly known or very difficult to sample from directly; whereas the conditional densities are easy to sample. Gibbs sampling is an iterative process. Suppose we have a joint distribution of two random variables (X, Y) given as $f(x, y)$. The Gibbs sampling will generate a sample from $f(x, y)$ by sampling from the condition distributions $f(x|y)$ and $f(y|x)$. This is done by generating a Gibbs sequence of random variables $Y_0^{'}, X_0^{'}, Y_1^{'}, X_1^{'}, \ldots, Y_k^{'}, X_k^{'}$. The initial value $Y_0^{'}$ is specified as $Y_0^{'} = y_0^{'}$ and the rest are found iteratively using the conditional distributions alternatively,

$$x_i^{'} \sim f(x^{'}|Y_i^{'} = y_i^{'})$$

and

$$y_{i+1}^{'} \sim f(y|X_i^{'} = x_i^{'})$$

Under reasonable general conditions the distribution of $X_k^{'}$ converges to $f(x)$ as $k \mapsto \infty$. Thus if k is large enough then the kth obsevation $X_k^{'} = x_k^{'}$ would effectively be a sample from $f(x)$ [12]. We will use this concept of approximating a density from its samples for sampling of Fisher-Bingham distribution in the next section.

5.2 Gibbs Sampling of Fisher Bingham Distribution

As described earlier; in slice sampling Kume et al. [11] introduce latent variables to sample a distribution so we will follow the same scenario here. We will introduce three latent variables (u, v, w) and use Gibbs sampling to draw samples using the conditional distributions of these latent variables. The joint density of interest described in Equation (5.3) will become,

$$f(\omega, s, u, v, w) \propto \mathbf{1}\left[u < \exp\left(\sum_{i=1}^{2}(a_i s_i + b_i \omega_i \sqrt{s_i})\right)\right] \times \mathbf{1}\left[v < \exp\left(b_0 \omega_0 \sqrt{1-s}\right)\right]$$

$$\times\, \mathbf{1}\left[w < \frac{1}{\sqrt{1-s}}\right] \times \prod_{i=1}^{2} \frac{1}{\sqrt{s_i}} \mathbf{1}\,(s \leq 1) \tag{5.4}$$

The conditional densities of u, v and w all are uniform and can easily be sampled. We now describe the conditional densities for ω_i and s_i. The conditional mass function of ω_i is straight forward and for $i = 0, 1, 2$ is qiven as,

$$P(\omega_i = +1|\ldots) = \frac{\exp(b_i\sqrt{s_i})}{\exp(-b_i\sqrt{s_i}) + \exp(b_i\sqrt{s_i})} \tag{5.5}$$

We will now list the full conditional density of s_1; the conditional density for s_2 can be found switching the indices,

$$f(s_i|\ldots) \propto \mathbf{1}\left[A_u \cap A_v \cap A_w \cap A\right] \frac{1}{\sqrt{s_1}} \tag{5.6}$$

where, A_u, A_v and A_w are the sets formed by inverting the inequalities involving latent variables and given as,

$$A = \{0 < s_1 < s_2\}$$

From Equation(5.6),

$$A_w = \left[s1 : \frac{1}{\sqrt{(1-s)}} > w\right]$$

$$= \left[s_1 > 1 - s_2 - w^{-2}\right]$$

Similarly,

$$A_v = \left[s1 : v < \exp\left(b_0\omega_0\sqrt{(1-s)}\right)\right]$$

in this case we have two scenarios, if $b_0\omega_0 < 0$ we have $A_v = A_v^+$ otherwise if $b_0\omega_0 > 1$ then $A_v = A_v^-$. Whereas, there is no constraint from A_v if $0 < b_0\omega_0 < 1$. A_v^+ and A_v^- are given as

$$A_v^+ = \left[s_1 > 1 - s_2 - \left(\frac{-\log(v)}{b_0} \right)^2 \right]$$

$$A_v^- = \left[s_1 < 1 - s_2 - \mathbf{1}(v > 1) \left(\frac{\log(v)}{b_0} \right)^2 \right]$$

Finally,

$$\begin{aligned} A_u &= \left[s1 : u < \exp\left(\sum_{i=1}^{2} (a_i s_i + b_i \omega_i \sqrt{s_i}) \right) \right] \\ &= [s1 : \exp\left(a_1 s_1 + b_1 \omega_1 \sqrt{s_1}\right) > d] \end{aligned}$$

where,

$$d = \log(u) - \exp\left(a_2 s_2 + b_2 \omega_2 \sqrt{s_2}\right)$$

if $a_1 > 0$ then $A_u = [s1 : l_u < \sqrt{s_1}] \cup [s1 : \sqrt{s_1} > t_u]$ where,

$$l_u = \left[\sqrt{\frac{d}{a_1} + \left(\frac{b_1}{2a_1} \right)^2} - \frac{b_1 \omega_1}{(2a_1)} \right]$$

And

$$t_u = \left[-\sqrt{\frac{d}{a_1} + \left(\frac{b_1}{2a_1} \right)^2} - \frac{b_1 \omega_1}{(2a_1)} \right]$$

Clearly, if the square root term is < 0 then $A_u = (0, 1)$; if $b_1\omega_1 > 0$ then $t_u < 0$ so $A_u = \{s_1 : l_u < \sqrt{s_1}\}$ and in case $a_1 = 0$ then $A_u = \{s1 : b_1\omega_1\sqrt{s_1} > d\}$. Once all the intervals have been evaluated, the final interval is found as the intersection as described in Equation(5.6). The final interval is always nonempty or at most a union of two disjoint intervals. This is due to the fact that A_u might sometimes end up as a union of two disjoint intervals. If intersection $A_u \cap A_v \cap A_w \cap A$ is a single interval say (ζ_l, ζ_u) then conditional density of s_1 would be,

$$f(s_1|\ldots) \propto \mathbf{1}\{s_1 \in (\zeta_l, \zeta_u)\} s_1^{-\frac{1}{2}}$$

It is therefore possible to sample this density as,

$$s_1 = \left[\tau(\sqrt{\zeta_u} - \sqrt{\zeta_l}) + \sqrt{\zeta_l} \right]^2 \tag{5.7}$$

where τ is a uniform random variable from interval $(0, 1)$. If, however the intersection $A_u \cap A_v \cap A_w \cap A$ is a union of two disjoint intervals $(\zeta_l, \zeta_u) \cup (\eta_l . \eta_u)$ then s_1 can be simulated as described above once the interval bounds are randomly chosen as either (ζ_l, ζ_u) or (η_l, η_u) according to their respective lengths $(\sqrt{\zeta_l}, \sqrt{\zeta_u})$ and $(\sqrt{\eta_l}, \sqrt{\eta_u})$.

Once the value of s_1 is simulated, we use this value of s_1 and follow the same procedure to get new value of s_2. The value of s_0 is then computed by the relation $s_0 = 1 - s = 1 - s_1 - s_2$. s_0 , s_1 and s_2 are then used to give the newly sampled point $\mathbf{x}_n = [x_0, x_1, x_2]$ where $x_i = w_i \sqrt{s_i}$ and $w_i \in \{-1, +1\}$ according to the conditional distribution given in equation (5.5).

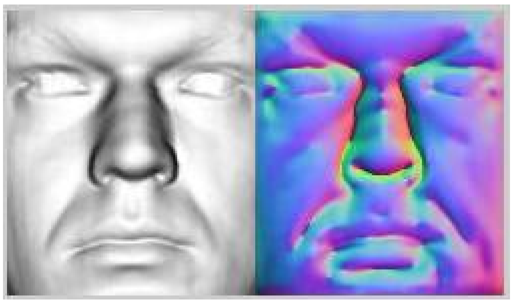

Figure 5.1: Shape-from-Shading Input & Output

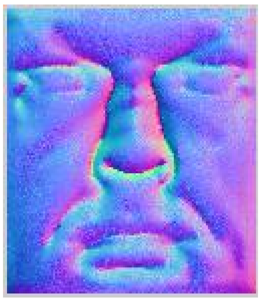

Figure 5.2: Fisher-Bingham Sampling

5.3 FB Sampling: An Example

The effectiveness of this sampling algorithm of Fisher-Bingham distribution can be represented with an example. Figure (5.1) on the left shows one of our synthetic test images. We used this image as an input for the probabilistic shape-from-shading algorithm of Haines which gives surface normals as Fisher-Bingham distributions for all pixels. The graphical output of the SfS is shown on the right in the same figure.

From the SfS algorithm we got a Fisher-Bingham distribution for each pixel that was sampled according to the sampling algorithm described above and resultant mean of samples is shown in figure (5.2). Comparing figures (5.1)&(5.2) it is apparent that Gibbs Fisher-Bingham sampling algorithm is effective and almost all the detail has been captured through sampling. We have displayed these figures here only to show a graphical result of Gibbs sampling algorithm. Actual details about getting spherical median from samples are given in next chapter.

Chapter 6

Combining the Models

As described in Chapter 4, Haines' probabilistic Shape-from-Shading method provides a Fisher-Bingham (FB_8) distribution for each pixel's surface normal. By using a statistical model of surface shape, we can improve the quality of surface normals produced by SfS. We can represent surface normals arising from Smith's statistical model of shapes as multivariate normal distributions on the tangent plane for all pixels as detailed later in section 6.1. Our objective is to get a most probable surface normal for each pixel based on these two techniques. We want to represent these individual surface normals arising from Haines' and Smith's approaches in such a way that they can be combined. One of the option would be to represent each surface normal arising from Smith's model with a Fisher-Bingham (FB_8) distribution and then combine the the two Fisher-Bingham distributions for each pixel. Estimating a Fisher-Bingham disribution from a multivariate Gaussian distribution is intractable and so can not be used to serve our objective. The second possible approach is opposite of the first one i.e. we modify the representation of Haines' surface normals instead of Smith's.

We adopted the second approach and used Gibbs sampling of Fisher-Bingham distributions given by Kume & Walker [11]. For each surface normal the Fisher-Bingham distribution provided by Haines SfS was sampled using the slice sampling algorithm as described in the previous chapter. After sampling, some issues still remain. The samples are directional (they lie on the surface of sphere) so combining them with Smith's model is not straight forward. The two models should be in the same plane and representation so that they can be combined. We use the machinery of the Exponential map to bring the samples on the tangent plane. The generated samples were then used to compute the mean vector and covariance matrix for each pixel. A bivariate normal distribution was fitted to each surface normal on the tangent plane and the two bivariate normal distributions were then combined to give a product normal distribution on the tangent plane. This resultant mean from product normal distribution when brought back to the unit sphere provides the most probable surface normal for each pixel.

6.1 Acquiring the Mean Vector and Covariance Matrix

Each surface normal from Smith's shape model and sampling of Fisher-Bingham distribution is represented by a bivariate Gaussian distribution on the tangent plane. To combine these surface normals resulting from Smith's statistical model and sampling of FB_8 distribution we need the mean vectors and covariance matrices. We will describe how we achieve these vectors and matrices from their respective origins.

6.1.1 From Smith's Model

From Smith's model of surface normals we get an $N \times 3$ vector of mean surface normals μ_1, $2N \times k$ matrix $\mathbf{P}$ of reduced principal dimensions (eigenvectors) and a k dimensional diagonal matrix $\mathbf{\Lambda}$ of eigenvalues. We can use the mean surface normals as they are. For our experiments we are interested in the actual covariance matrix from which Smith derived the eigenvectors and eigenvalues. So, using the matrix $\mathbf{P}$ and $\mathbf{\Lambda}$ we get the actual covaraince matrix of $2N \times 2N$ dimensions.

$$\mathbf{L}_r = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T \tag{6.1}$$

The matirx $\mathbf{L}_r$ gives us a $2N \times 2N$ matrix of all the pixels but we are interested in individual 2×2 covariance matrices for each pixel. From section (4.7.2) we know that Smith constructed covariance matrix $\mathbf{L}$ using data matrix $\mathbf{D}$ in which all the training examples were stacked as columns and each training example's column in turn was comprised of $2N$ elements. In these $2N$ elements of vector $\mathbf{U}$ the first N elements were the x-coordinates and the next N elements were the y-coordinates of that training example on the tangent plane. To make the individual covariance matrices for all pixels we have to get the elements of these matrices from the covariance matrix $\mathbf{L}_r$. We will now describe the method to get the individual covariance matrix $\mathbf{C}_{1(x,y)}$ for pixel (x, y).

$$\mathbf{C}_{1(x,y)} = \begin{bmatrix} \mathbf{L}_{r(x,y)} & \mathbf{L}_{r(x+N,y)} \\ \mathbf{L}_{r(x,y+N)} & \mathbf{L}_{r(x+N,y+N)} \end{bmatrix} \tag{6.2}$$

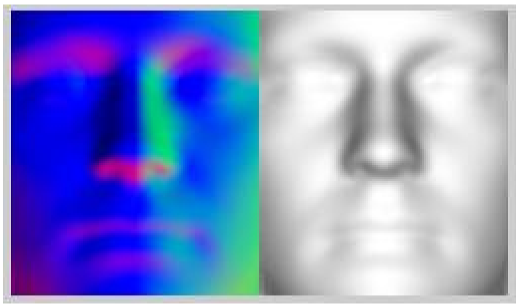

Figure 6.1: Statistical Surface Normal Model, Normals Illuminated

Smith's statistical model of surface normals is shown on the left in Figure (5.1) whereas the left image shows when these normals are re-illuminated with a unit magnitude light source from [0 0 1] direction.

6.1.2 From Haines SfS

We sampled the Fisher-Bingham (FB_8) distribution for each pixel provided by the Haines [3] shape-from-shading algorithm using Gibbs sampling of A. Kume & S. G. Walker [11] as described in Chapter 5. Kume's sampling method takes the Fisher and Bingham parameters and returns the sample for FB_8 distribution. From Haines' SfS method we get Fisher vector and Bingham matrix for each pixel that we use for sampling of FB_8 distribution. The distribution used for sampling by Kume & Walker as given in equation (5.1) is,

$$f(x) \propto \exp\left(-x^T \Sigma x + 2x^T \Sigma \mu - \mu^T \Sigma \mu\right) \tag{6.3}$$

and $b_i = 2\lambda_i \mu_i$. Whereas Haines used FB_8 distribution given by equation (4.3),

$$f(x) \propto \exp\left(x^T A x + x^T \mu\right) \tag{6.4}$$

So, in order to use Kume's formulation we adjusted Fisher and Bingham parameters as $\Sigma = -A$ and $b_i = \mu_i$ where A is the Bingham matrix and μ is the Fisher vector resulting for each pixel from SfS, here μ should not be confused with μ_1 or μ_2 used for surface normals representation . For each pixel we first generate k samples using FB_8 sampling and get a $k \times 3$ vector of samples $\mathbf{S}$ i.e.

$$\mathbf{S} = [p_1, p_2, \cdots, p_k]^T \tag{6.5}$$

where, p_k is the kth sample having three components i.e. $p_k = [x_0 \quad x_1 \quad x_2]$ and $x_i = w_i\sqrt{s_i}$ for $i = 0-2$ from FB_8 sampling. We calculate the extrinsic mean of the surface normals (samples) using,

$$\hat{p}_0 = \frac{1}{K}\sum_{k=1}^{K} p_k \tag{6.6}$$

We then use this extrinsic mean and generate new samples to calculate the intrinsic mean using the iterative process as described in section (4.8).

$$\hat{p}_{j+1} = \mathrm{Exp}_{\hat{p}_j}\left(\frac{1}{K}\sum_{i=1}^{K} \mathrm{Log}_{\hat{p}_j}(p_i)\right) \tag{6.7}$$

We use $5-10$ iterations to calculate the spherical median and for each iteration we generate about 100 samples. Once the spherical median is found we use this surface normal $\mu_2 = \hat{p}_{final}$ as the base point and convert the k samples $[p_1, p_2, \cdots, p_k]^T$ from last iteration to the vectors $[v_1, v_2, \cdots, v_k]^T$ on the tangent plane using the log map for the sphere i.e.

$$v_k = \mathrm{Log}_{\mu_2}(p_k) \tag{6.8}$$

So, we get a $k \times 2$ dimesnion vector $\mathbf{V} = [v_1, v_2, \cdots, v_k]^T$ on the tangent plane. Using these points on the tangent plane we calculate the 2×2 covariance matrix $\mathbf{C}_{2(x,y)}$ which we will use to combine with the surface normals from Smith's model.

$$\mathbf{C}_{2(x,y)} = \mathrm{Cov}(\mathbf{V}) \tag{6.9}$$

where Cov gives the covariance of a matrix. In Figure (6.2) we have shown one of our test images from synthetic data, output from Haines probabilistic SfS algorithm and when the SfS output is illuminated with unit light source. Figure (6.3) on the other hand shows the surface normals generated through the sampling of FB distributions for this particular test image and when these normals are illuminated with unit light source [0 0 1] along with the test image itself. Comparing Figures (6.2) and (6.3) we can notice the effectiveness of slice sampling algorithm used for sampling of Fisher-Bingham distributions.

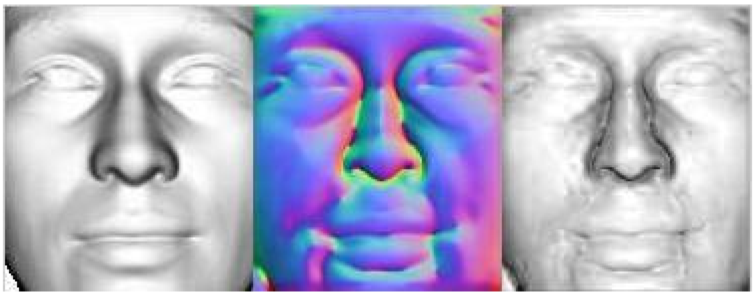

Figure 6.2: SfS Input, SfS Output, Illuminated Output

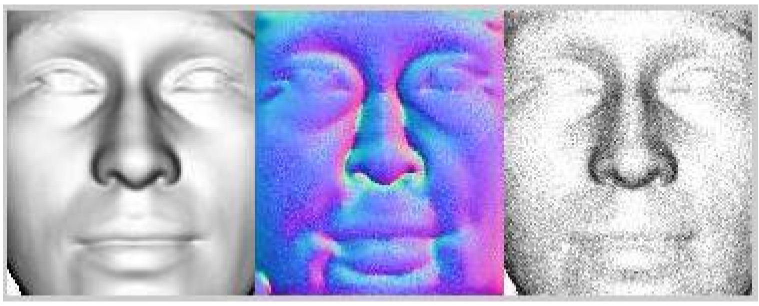

Figure 6.3: Test Image, FB Sampled Normals, Illuminated Normals

6.2 Product Normal Distribution

Let's assume we have two $2D$ Gaussian distributions Ω_1 and Ω_2 having different respective means μ_1 , μ_2 and covariance matrices Σ_1 and Σ_2. μ_1 and μ_2 are 2×1 vectors whereas Σ_1 and Σ_2 are 2×2 matrices. We can formulate these normal distributions as $\Omega_1[P_1\mu_1, P_1]$ and $\Omega_2[P_2\mu_2, P_2]$ where P_1 and P_2 are the inverse of respective covariance matrices. These two Gaussian distributions can be combined to give product normal distribution,

$$\Omega_T[P^*\mu^*, P^*] = \Omega[P_1\mu_1 + P_2\mu_2, P_1 + P_2] \tag{6.10}$$

From the product normal distribution we can calculate the combined (resultant) mean of this distribution which is given by,

$$\mu^* = (P^*)^{-1}(P^*\mu^*) \tag{6.11}$$

We combine the Gaussian distributions resulting from Smith's model (Sub-Section 6.1.1) and FB sampling (Sub-Section 6.1.2) using Equations 6.10 and 6.11. We use the surface normals from Smith's model μ_1 as a base point for Log-exp mapping and convert the surface normals resulting from FB sampling μ_2 to the tangent plane using,

$$v_{2(x,y)} = \mathrm{Log}_{u_{1(x,y)}}\left(u_{2(x,y)}\right) \tag{6.12}$$

The choice for selecting μ_1 as the base point for Log-exp mapping was made due to the fact that this is a statistical surface normal and hence more reliable as compared to μ_2 which belongs to a single subject and has comparatively more probability of being erronous due to local features or other discontinuties e.g. albedo and specular reflection etc. Another appropriate choice for the base point could be the summation of two surface normals i.e. $\mu_t = \mu_1 + \mu_2$. We however used μ_1 for our calculations and experiments. Once, we bring the FB sampled normal on the tangent plane, the two Gaussian distributions are then combined using equation (6.10) i.e.

$$\Omega_T[P^*v^*, P^*] = \Omega[P_1v_1 + P_2v_2, P_1 + P_2] \tag{6.13}$$

where, v_1 is 2×1 zero vector. Using the convention of equation (6.11) the combined normal v^* on the tangent plane is found.

$$v^* = (P^*)^{-1}(P^*v^*) \tag{6.14}$$

The normal on the tangent plane is then converted back on the unit-sphere using the $\mu_1(x, y)$ with exponential mapping i.e.

$$\mu^* = \mathrm{Exp}_{u_{1(x,y)}}\left(v^*_{(x,y)}\right) \tag{6.15}$$

We continue using same test image and combine the surface normals generated through the sampling of FB distributions i.e. μ_2 with those emerging from Smith's model i.e. μ_1 and results are display below in figure (6.4). The images on the left show the sampled surface normals μ_1 and when they are illuminated where as on the right we show the resultant combined surface normals μ^* generated through product normal distribution using equation (6.15).

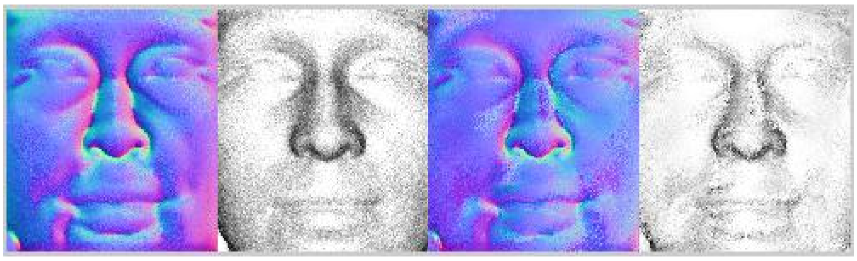

Figure 6.4: μ_2 , μ_2 Illuminated , μ^* , μ^* Illuminated

6.3 Dealing with Outliers

Outliers arise in the regions where surface normals resulting from Smith's model do not comply with the surface normals generated through the sampling of the Fisher-Bingham distribution. This is due to the fact that we are so far ignoring the effects of albedo i.e. the irradiance equation assumes unit albedo and also the shadowed regions do not contribute towards the surface orientation. We use unit albedo for synthetic data in our experiments however we estimate non-unit value of albedo for real images; we will talk about albedo in next chapter when we will give experiment details.

The other phenomenon responsible for outliers is shadowing. When a surface is illuminated with unit light source; a point on the surface is said to be in shadow if no light reaches to that point. There are two scenarios of a point on a surface being in shadow i.e. cast shadow and attached shadow. A self shadow or attached shadow occurs when a point is oriented away from the light source and thereby not being illuminated [1]. A cast shadow on the other hand is a point that is being occluded from light source due to another region of the surface. These shadowed regions are responsible for surface normals from SfS not being registered with surface normals from shape model. To deal with these outliers we use the Fisher criterion. The Fisher criterion is one of the basic pattern classifiers and it is suitable for our algorithm due to the fact that we have separate class distributions to represent surface normals from shape model and FB sampling and in particular the knowledge of the covariance matrices makes it the best choice.

As a final step we also perform smoothing to get best fit normals before using Smith's height model to recover surface height from surface normals. Smoothing is performed to achieve a globally consistent solution and nulling out any local glitches or noise in the resultant combined surface normals arising from product normal distribution after incorporation of the Fisher criterion.

6.3.1 Incorporating the Fisher Criterion

The Fisher criterion is a linear pattern classifier that evaluates between-class variance relative to the within-class variance [20]. The idea of the Fisher criterion lies in finding such a vector d that the patterns belonging to opposite classes would be optimally separated after projecting them onto d [21]. The Fisher criterion for two classes can be expressed as [21, 22],

$$F(d) = \frac{d^T B d}{d^T \Sigma d} \tag{6.16}$$

where, B is between class scatter matrix given by $B = \Delta\Delta^T$ and $\Delta = \mu_1 - \mu_2$. μ_1, μ_2 are the mean vectors. Σ_1, Σ_2 are the covariance matrices of classes 1 and 2. $\Sigma = p_1\Sigma_1 + p_2\Sigma_2$ where p_1 and p_2 are the a priori probabilities of the two classes which are taken as 0.5 while giving equal weight to both classes. The optimal value of Fisher discriminant d_{opt} maximises the value of F. To find this optimal value of d we take the first derivative of $F(d)$ and equate it to zero i.e. $F'(d) = 0$ which gives us,

$$d_{opt} = \Sigma^{-1}\Delta \tag{6.17}$$

We used Fisher Criterion to deal with the outliers as follows. The outliers exist at the pixels where the two surface normals from FB sampling and Smith's model do not register with each other normally because the SfS surface normals are in error because of the albedo or surface shadows. We use the two class Fisher criterion to decide whether the two surface normal distributions are in agreement or not. We decide a threshold value for each specific example and then replace that particular pixel's surface normal μ^* with Smith's surface normal μ_1 at that specific location if the value of $F(d)$ is higher than decided threshold.

The stem graph of Fisher criterion for the test image under consideration is shown in figure (6.5). The threshold value for this particular example was chosen as 350 so all surface normals μ^* giving the Fisher criterion value greater than 350 were replaced with statistical surface normals at respective pixels (x, y).

The number of normals affected by the Fisher criterion depends upon the threshold; tightening the threshold will increase the number of affected normals and vice versa. In this particular example the choice of 350 as a threshold has affected 64 normals. The binary representation of the Fisher

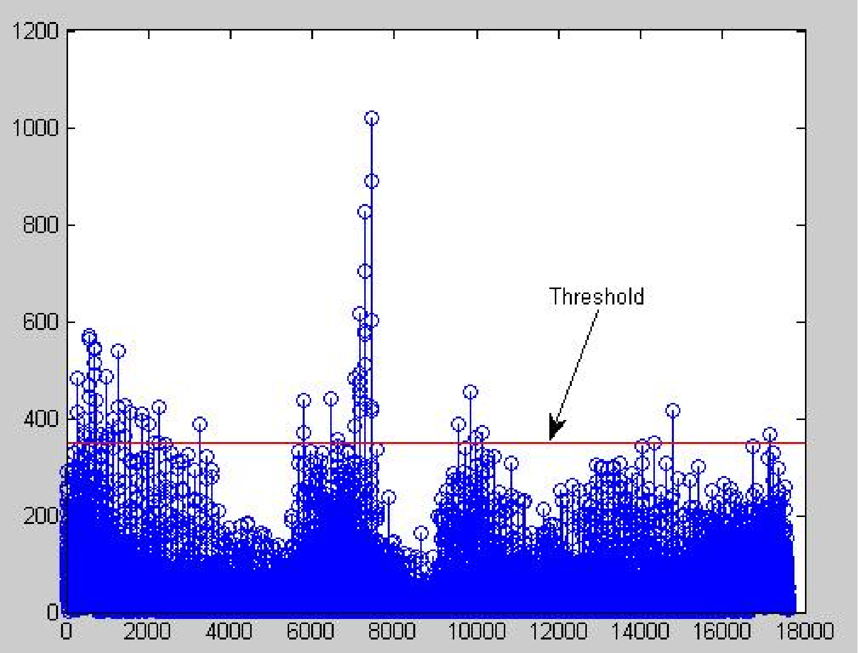

Figure 6.5: Stem plot of Fisher Criterion showing Threshold

criterion and pixels being affected in our particular example are given in the figure 6.6. In the left image white dots represent where the value of the Fisher criterion is higher than the threshold whereas the red dots in the right image identify the pixels being affected due to outliers.

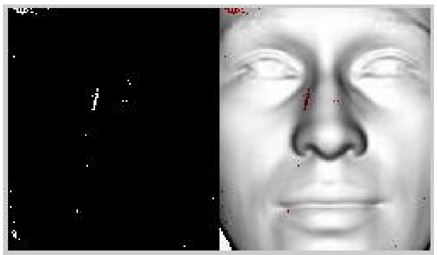

Figure 6.6: Fisher Criterion specifying Outliers

6.3.2 Smoothing the Normals

As described earlier in this Chapter, smoothing is required to get a globally consistent surface and filter out any local discontinuties in the combined surface normals. After all the surface normals have been combined using product normal distribution and incorporation of Fisher criterion, we perform smoothing on these normals. Smoothing is required as it makes the normals best fit to the statistical model. The normals are first converted on the tangent plane using the Smith's statistical normals as the base points using Log map i.e.

$$u_{(x,y)} = \mathrm{Log}_{\mu_1}(\mu^*) \tag{6.18}$$

where μ_1 and μ^* are the statistical surface normal from shape model and combined surface normal at location (x, y), the subscripts have been dropped for clarity. The points $[u_1, u_2, \cdots, u_N]$ on the tangent plane are then stacked together to give a $N \times 2$ vector $\mathbf{U}_c$ of combined normals on the tangent plane. The best fit vector on the tangent plane is then found using,

$$\mathbf{U}_r = \mathbf{P}\mathbf{P}^T\mathbf{U}_c \tag{6.19}$$

where, $\mathbf{P}$ is the matrix of principal directions from Smith's statistical model. $\mathbf{U}_c$ is the vector comprised of combined surface normals on the tangent plane and $\mathbf{U}_r$ is the vector of recovered smoothed surface normals. The surface normals on the tangent plane are then brought back to the unit sphere through exponential mapping using Smith's statistical needle map as base points.

$$\mu^*_{(x,y)} = \mathrm{Exp}_{\mu_1}(u_{x,y}) \tag{6.20}$$

Figure (6.7) shows the smoothed surface normals on the left using equation (6.20) for the same test image; in right image the smoothed surface normals are illuminated with unit light source.

6.4 Getting Height from Combined Normals

To get surface height from surface normals we use Smith's statistical height model as described in section (4.9). From combined surface normals we construct the surface gradient vector $\mathbf{G}$ as given by,

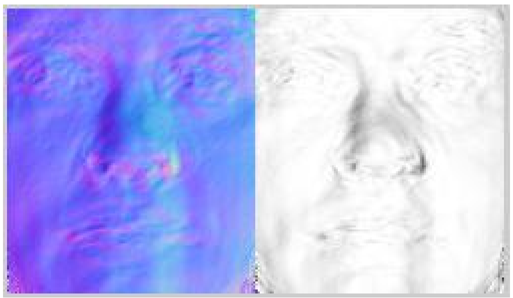

Figure 6.7: Smoothed Surface Normals

$$\mathbf{G} = \left\{ \begin{array}{c} p(1,1) - \hat{p}(1,1) \\ q(1,1) - \hat{q}(1,1) \\ \cdots \\ \cdots \\ p(X_{res}, Y_{res}) - \hat{p}(X_{res}, Y_{res}) \\ q(X_{res}, Y_{res}) - \hat{q}(X_{res}, Y_{res}) \end{array} \right\} \tag{6.21}$$

Using the eigenvectors of statistical height model $\mathbf{\Psi}$ and gradient vector $\mathbf{G}$ we calculate the surface parameter b^* following equation (4.61).

$$b^* = (\mathbf{\Psi}^T \mathbf{\Psi})^{-1} \mathbf{\Psi}^T \mathbf{G} \tag{6.22}$$

The surface parameter b^* along with average surface height $\hat{z}(x, y)$ and k eigenmodes of height model $\Psi_i(x, y)$ are then used to give the surface height according to equation.

$$z_b(x, y) = \hat{z}(x, y) + \sum_{i=1}^{K} b_i \Psi_i(x, y) \tag{6.23}$$

In figure (6.8) average surface height from Smith's Height model, original height for the test image we have been working with and recovered height using smoothed normals are shown from left to right respectively. To compare the recovered height with the actual height for test images Iterative Closest Point algorithm is used which will be described in the next chapter when we will list experiment results.

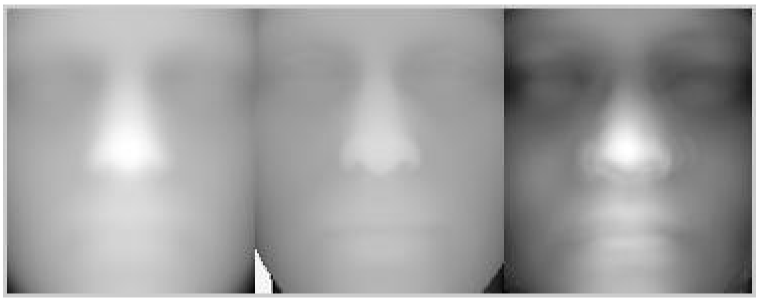

Figure 6.8: Average, Original and Recovered Surface Height

Chapter 7

Experiment Details

We tested our algorithm on synthetic as well real images. Range images from 3DFS data base [4] and Max Plank data base [5] were used to test the algorithm for synthetic data. For real data we ourselves collected images under Shape-from-Shading assumptions and 3D scans using Cyberware 3030PS laser head scanner. We also use images from Yale B data base for visual results. We used Iterative Closest Point algorithm to compute the root mean square distance between the actual & recovered surface normals and actual & recovered surface heights with and without smoothing.

7.1 Iterative Closest Point Algorithm

Iterative Closest Point algorithm as the name states is an *iterative* algorithm for matching point-sets [23]. ICP is used to compute a matching between two point sets that minimizes the root mean squared distance (RMSD). Consider 2 point sets $A, B \subseteq R^d$ where $|A| = n$ and $|B| = m$, we use ICP to find a one-to-one matching function $p : A \to B$ that minimizes the root mean squared distance between A and B [24]. The root mean squared distance can be mathematically defined as,

$$RMSD(A, B, p) = \sqrt{\frac{1}{n} \sum_{a \in A} ||a - p(a)||^2} \tag{7.1}$$

To get the perfect matching for point sets we want to minimize the function RMSD. If we introduce rotation (R) and translation (T) the minimization problem for equation (7.1) can be written as,

$$\min_{p:A\rightarrow B,T\in R^d,R\in SO(d)} \sum_{a\in A} ||Ra - T - p(a)||^2 \tag{7.2}$$

The ICP algorithm seeks to minimize the RMSD by alternating between a matching step and a transformation step. In the matching step the optimal matching is found for the given rotation and translation by calculating RMSD. Conversely in the transformation step an optimal transformation i.e. rotation matrix and translation vector are computed given a matching. This alternating process terminates when the matching remains unchanged in successive iterations.

As stated earlier we use ICP algorithm to compute the error between the ground truth surface normals and resultant surface normals from algorithms. ICP is also used to compute the error between actual and recovered surface heights. Since the point-sets might have different dimensions; ICP is suitable to compute errors for both synthetic as well as real data. In case of synthetic images we have point sets A and B having same dimensions but in case of real data the ground truth $3D$ scans have higher dimensions than the recovered surface height vectors.

7.2 Synthetic Data

The data base of range images based on 3DFS [4] and Max Plank [5] contain 200 images. We will use a smaller subset of this data base comprising 10 test images for presentation purposes. However the error differences are computed for complete data base and will be presented for all 200 images using graphs. Each of the range images give ground truth surface normals which are then illuminated with unit light source to give the synthetic input images for Shape-from-Shading algorithm.

Figure (7.1) shows 10 range images which will be used later to compare the recovered surface heights. From these range images ground truth surface normals are generated which are shown in figure (7.2). These ground truth normals will be used to compare with (1) normals resulting from Shape-from-Shading and when (2) SfS normals are combined with shape models using product normal distribution as described in previous chapter. When ground truth normals are rendered with a frontal unit light source [0 0 1] they result into synthetic images shown in figure (7.3). These synthetic images are used as inputs for probabilistic Shape-from-Shading of Haines. Since, these images

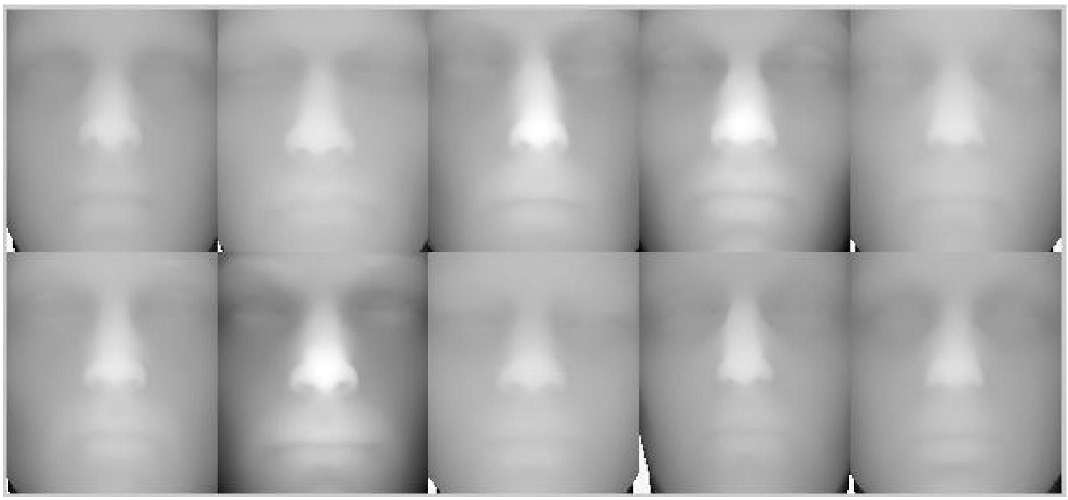

Figure 7.1: Range Images

have only Lambertian reflectance; the effects of albedo can be ignored and its value can be taken as unity.

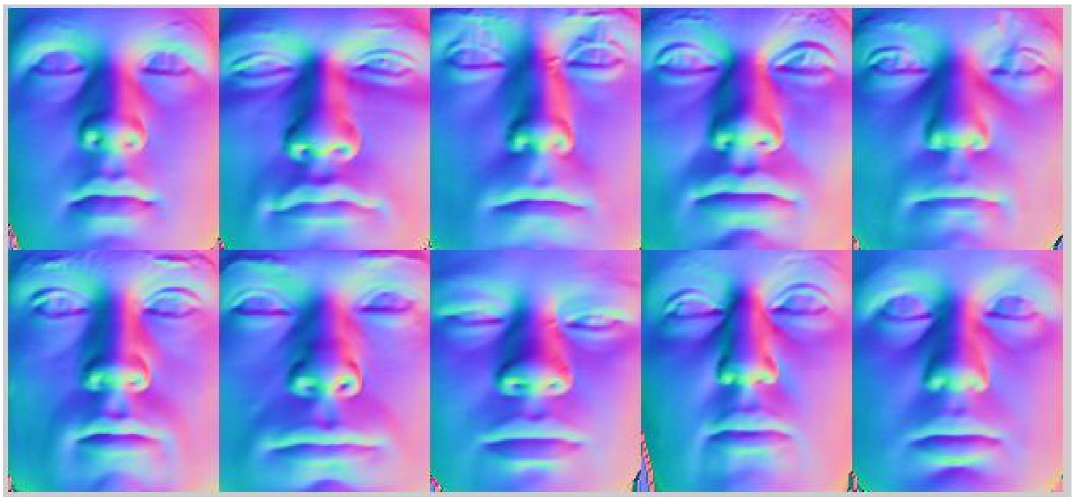

Figure 7.2: Surface Normals

7.2.1 Error in Surface Normals

We use rendered ground truth surface normals shown in figure (7.3) as input for Haines Shape-from-Shading method. As described earlier for synthetic data unit albedo is assumed. The graphical results from Haines Shape-from-Shading algorithm are shown below in figure (7.4).

As stated number of times before Haines Shape-from-Shading algorithm outputs surface normals represented by Fisher-Bingham distributions. For each of these 10 test images we take the Fisher-Bingham distributions and sample them according to the algorithm described in Chapter 5 and 6. For each Fisher-Bingham distribution we take 500 samples in eact iteration to compute spherical median. The normals resulting from sampling of Fisher-Bingham distributions for all these 10 images are shown in figure (7.5).

To give a better visual comparison of FB sampled surface normals and ground truth surface normals we have stacked them together in figure (7.6). Row 1 shows the first five ground truth surface normals illuminated with unit light source while row 2 shows the Fisher-Bingham sampled surface normals illuminated with [0 0 1] light source. Similarly row 3 and 4 display the illuminated ground truth and illuminated sampled surface normals for remaining 5 subjects.

By comparing figures (7.4) and (7.5) we can notice that Fisher-Bingham sampling process is performing quite well. All the shape and major features for all faces have been preserved during sampling. Sampling however

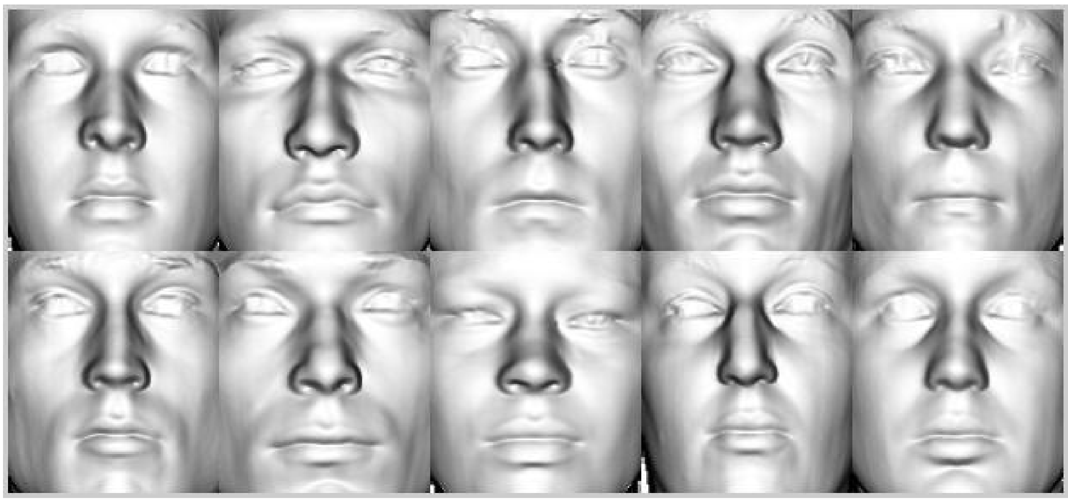

Figure 7.3: Normals Illuminated

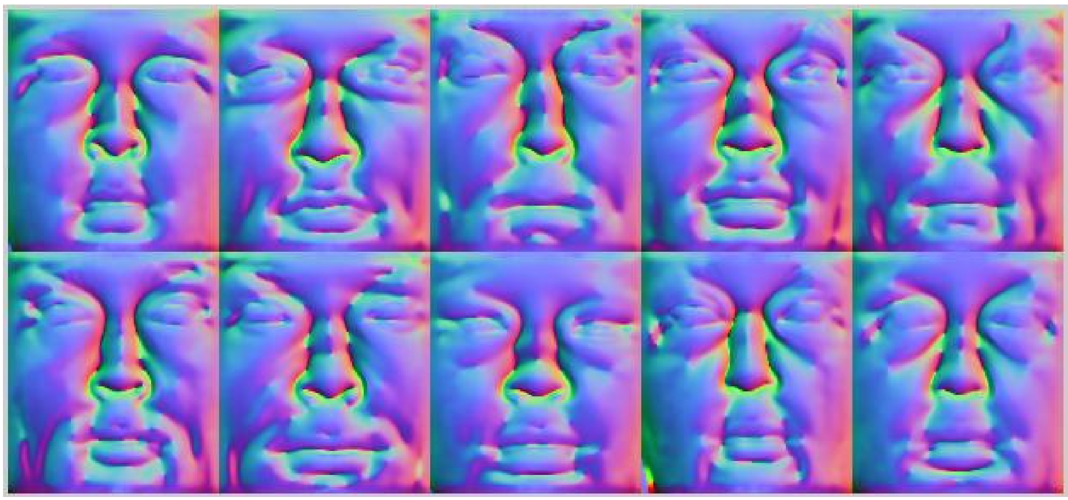

Figure 7.4: Shape-from-Shading Output

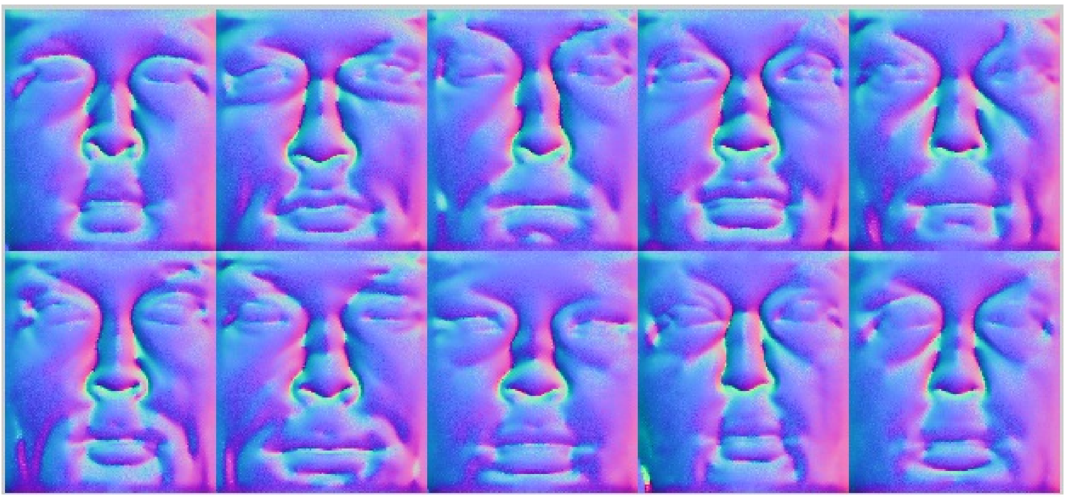

Figure 7.5: Fisher-Bingham Sampled Normals

introduces some effects that look like salt and pepper noise. These discontinuties are however taken care of when we combine the Fisher-Bingham sampled normals with the statistical normals and then the usage of Fisher criterion and smoothing.

As described in Chapter 5 the surface normals arising from sampling of Fisher-Bingham distributions μ_2 are combined with surface normals from Smith's shape model μ_1 to give most probable suface normals μ^*. The Fisher criterion is then incorporated to deal with the outliers emerging due to shadowed regions. The combined surface normals for the under consideration synthetic images are shown in figure (7.7). The Fisher criterion has also been incorporated in these combined normals.

Illuminated combined surface normals are also shown in figure (7.8) along with illuminated groundtruth surface normals and sampled Fisher-Bingham normals. In figure (7.8) the first row gives the illuminated groundtruth normals for first five subjects, 2nd row gives the Fisher-Bingham sampled normals and 3rd shows the combined normals. Similarly rows 4, 5 and 6 repeat the same scenario for next five subjects.

We can see the smoothness improvement in the resultant normals when Fisher-Bingham normals are combined with statistical normals. The effects of discontinuties have been filterred out however the shapes of faces are still preserved.

We have calculated the Root mean squared error between (a). ground truth surface normals and Fisher-Bingham sampled surface normals and

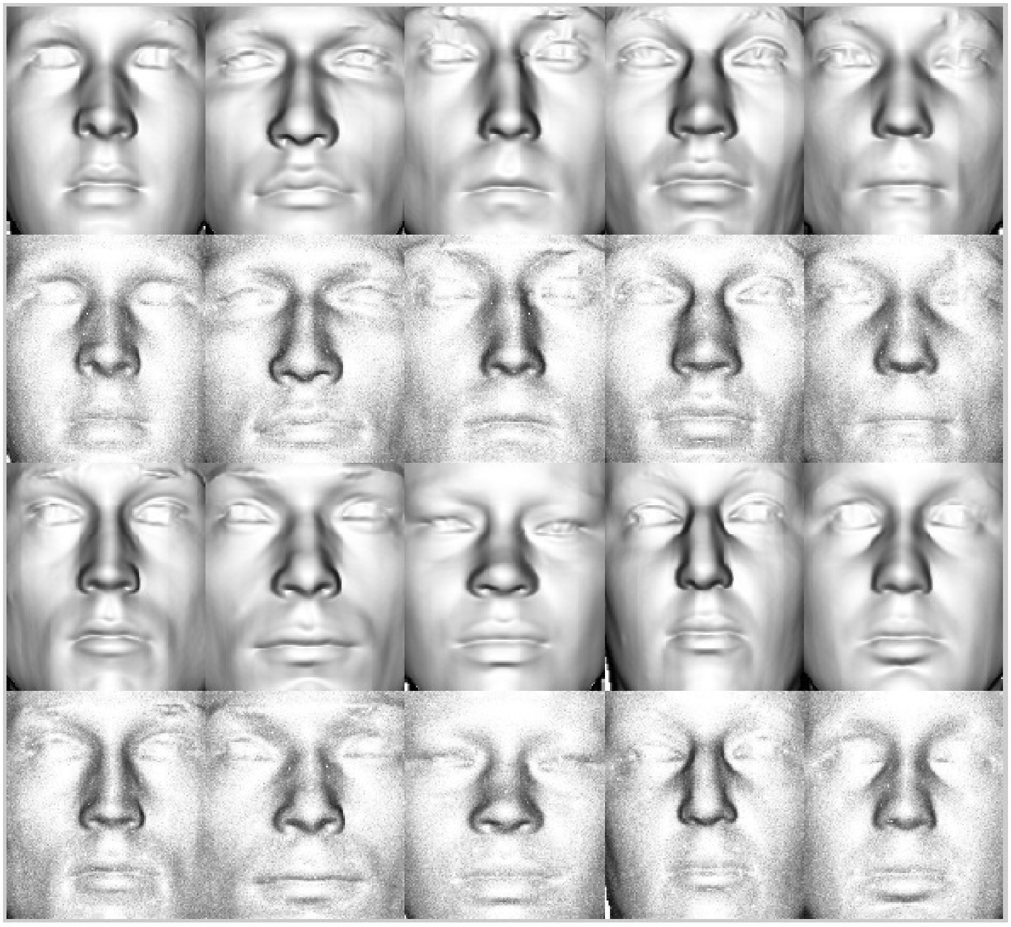

Figure 7.6: Groundtruth Vs Fisher-Bingham Sampled Normals

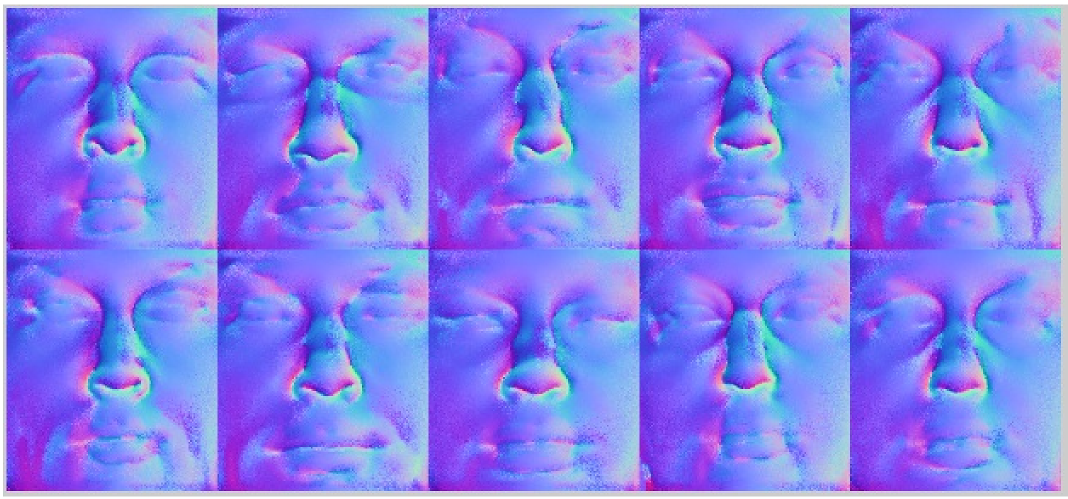

Figure 7.7: Combined Surface Normals

(*b*). ground truth surface normals and combined surface normals for all 200 synthetic images. By combined surface normals we mean here the Fisher-Bingham sampled surface normals combined with Smith's statistical surface normals along with the incorporation of Fisher criterion. For numerical error computations we have used 25 samples per iteration for computing spherical median for each pixel to show the effectiveness of our algorithm even when FB sampled normals are relatively erroneous. The error results are displayed in figure (7.9). The subjects were arranged according to ascending error rate between the ground truth surface normals and sampled surface normals. The red curve ($\mathrm{E}_{\mathrm{FBSampled}}$) shows the error computed between ground truth suface normals and Fisher-Bingham sampled surface normals from SfS using ICP algorithm. Whereas the green curve ($\mathrm{E}_{\mathrm{Combined}}$) shows the error between ground truth normals and combined surface normals. The frequency bar charts for both errors are shown in figure (7.10).

It is apparent that the error has been reduced with the use of Smith's statistical surface normals. Most number of subjects were lying in the error window of 4 & 7 when using only sampled Fisher-Bingham surface normals from SfS. Whereas with the use of statistical surface normals and Fisher criterion the error has reduced for all the subjects considerably.

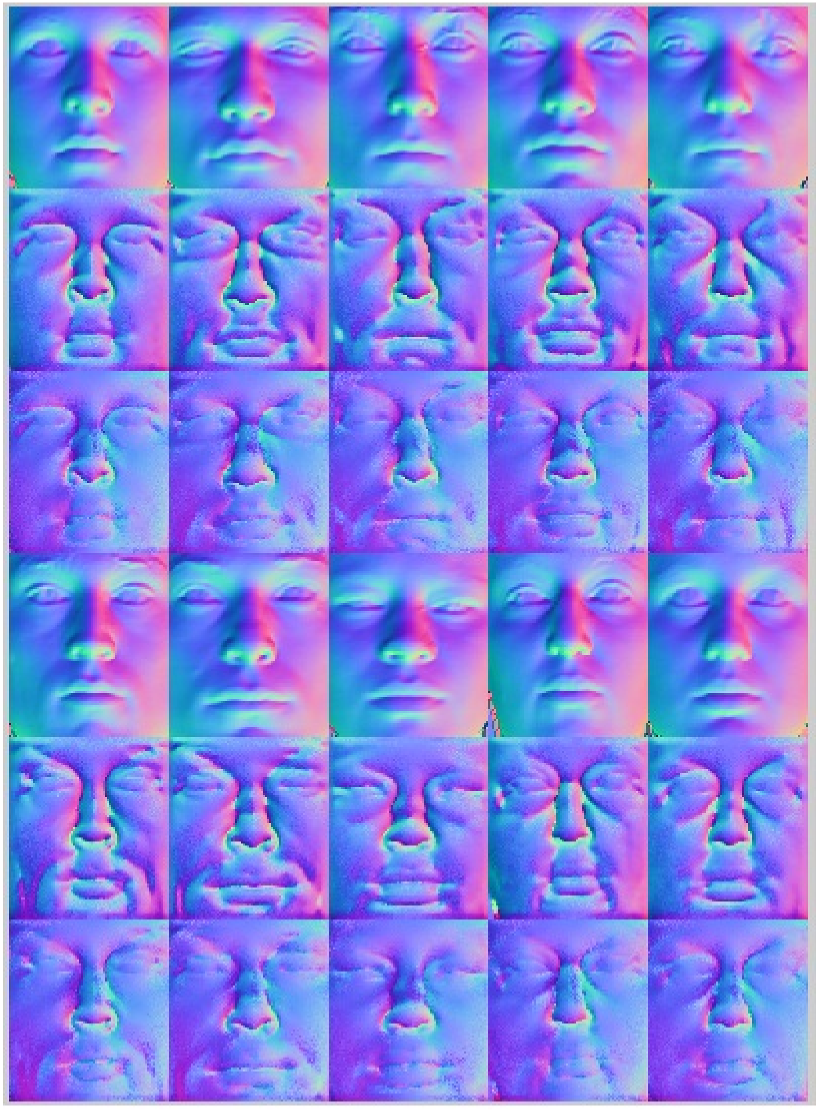

Figure 7.8: Ground Truth, Fisher-Bingham Sampled and Combined Normals shown together

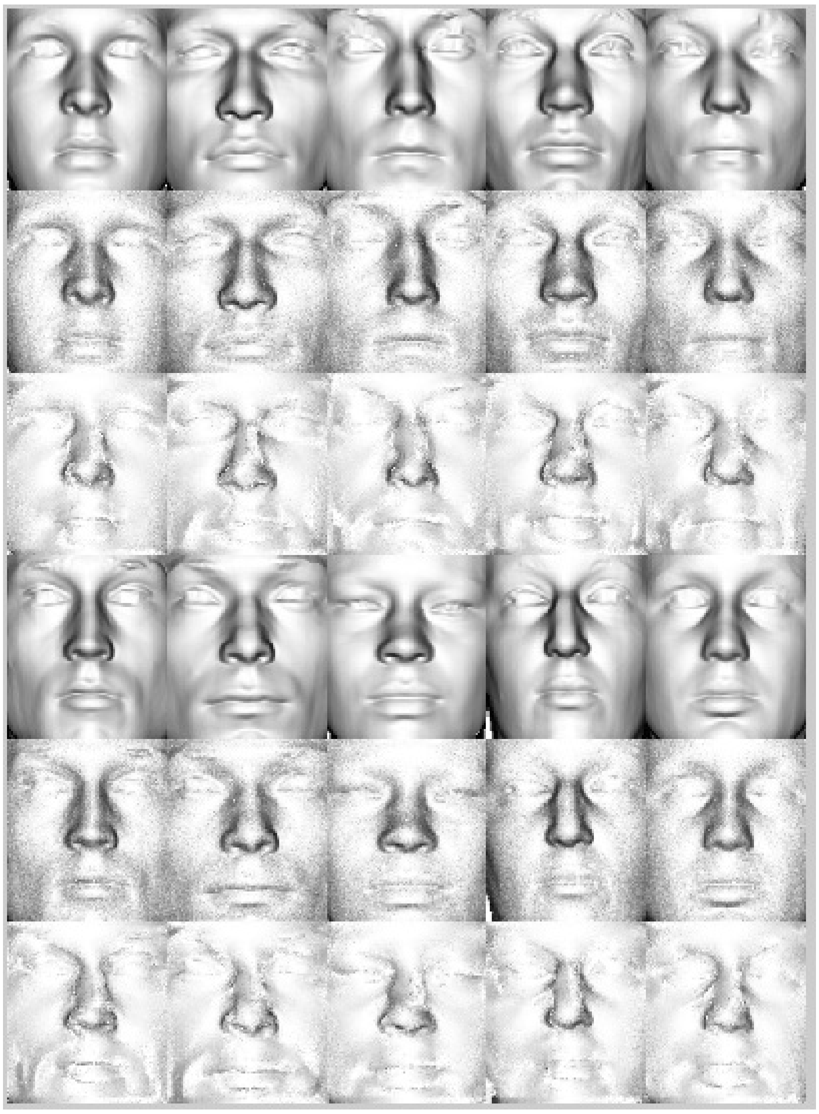

Figure 7.9: Illuminated Ground Truth, Fisher-Bingham Sampled and Combined Normals

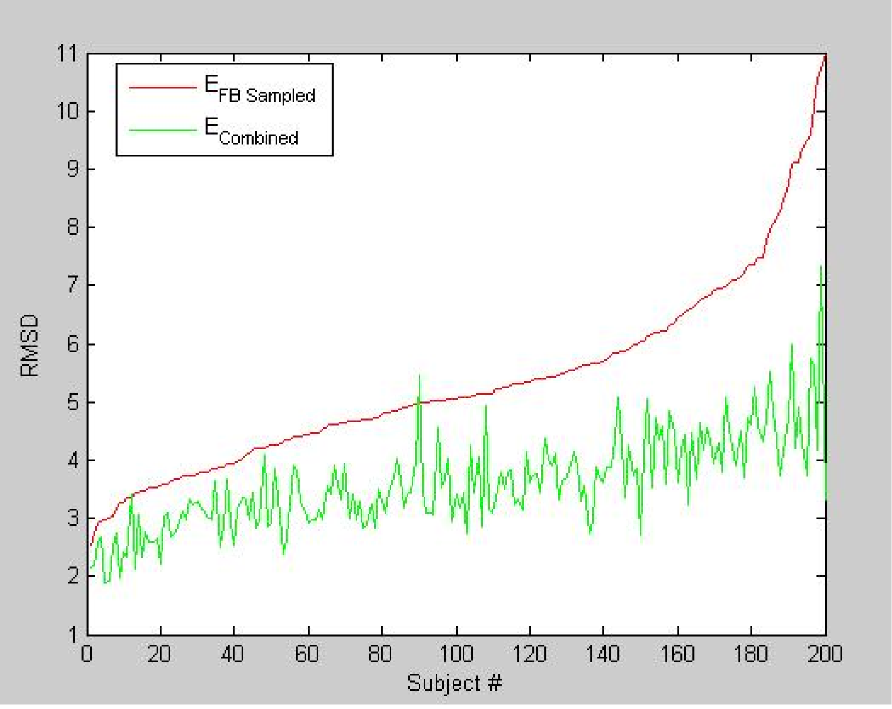

Figure 7.10: Error between Ground truth & Sampled Fisher-Bingham Normals

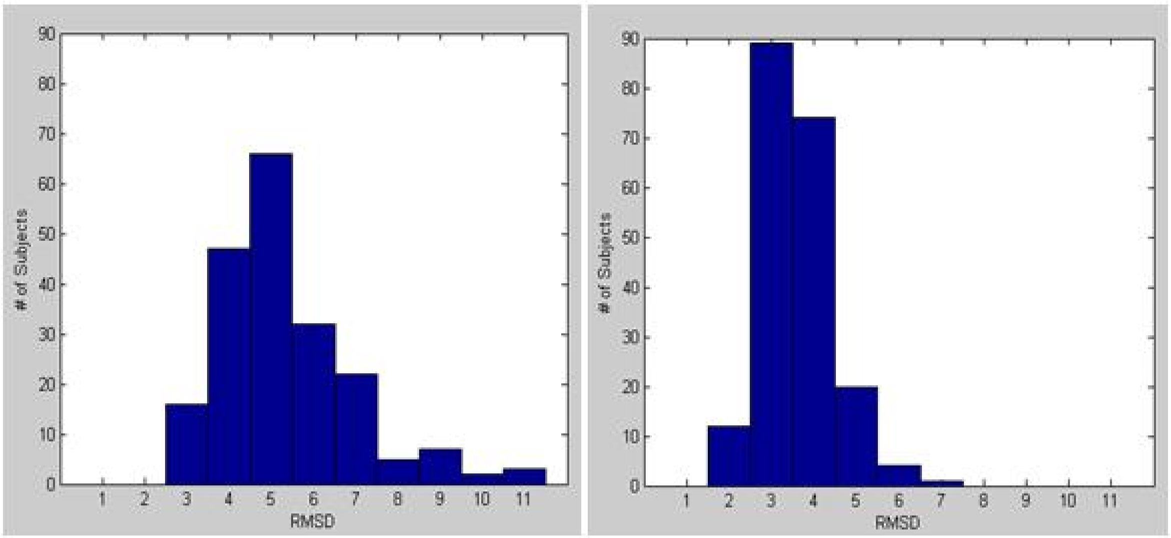

Figure 7.11: Frequency Histogrms for $\mathrm{E}_{\mathrm{FBSampled}}$ and $\mathrm{E}_{\mathrm{Combined}}$

7.2.2 Error in Surface Heights without Smoothing

As detailed in Chapter 5 after combining the Fisher-Bingham sampled and statistical surface normals we incorporate Fisher criterion and then use smoothing to get best fit surface normals. The best fit normals are then used to get surface height using Smith's height model. For this specific experiment we will not perform the smoothing step. Instead we will use the combined normals including the step of incorporation of the Fisher criterion and generate suface height using Smith's surface height model as discussed in Chapter 4 and 6. We continue to use same 10 subjects and recover height from the combined normals without smoothing.

Again we compared the surface heights recovered with only sampled surface normals and combined normals with the actual height of the subjects and the results are shown in the graph displayed in figure (7.11). The subjects are arranged according to the ascending error between ground truth surface heights and surface heights recovered from sampled normals. The red curve ($\mathrm{SurfaceHeightError}_{\mathrm{FBSampled}}$) represents the surface height error between the ground truth surface and surface recovered from sampled normals. The green curve ($\mathrm{SurfaceHeightError}_{\mathrm{Combined}}$) on the other hand shows the error between the ground truth surface height and suface height recovered using combined normals.

Like surface normals we have also represented the subject frequencies that

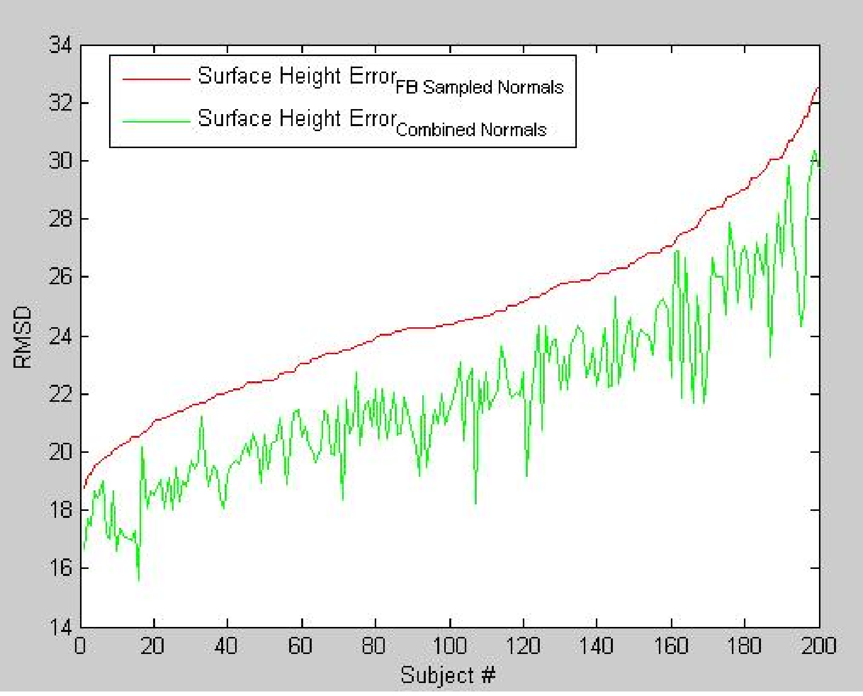

Figure 7.12: Surface Heights Error without Smoothing

lie in specific bins of error using histograms. Improvements in the surface height error are visible even when the smoothing is not performed on the combined surface normals.

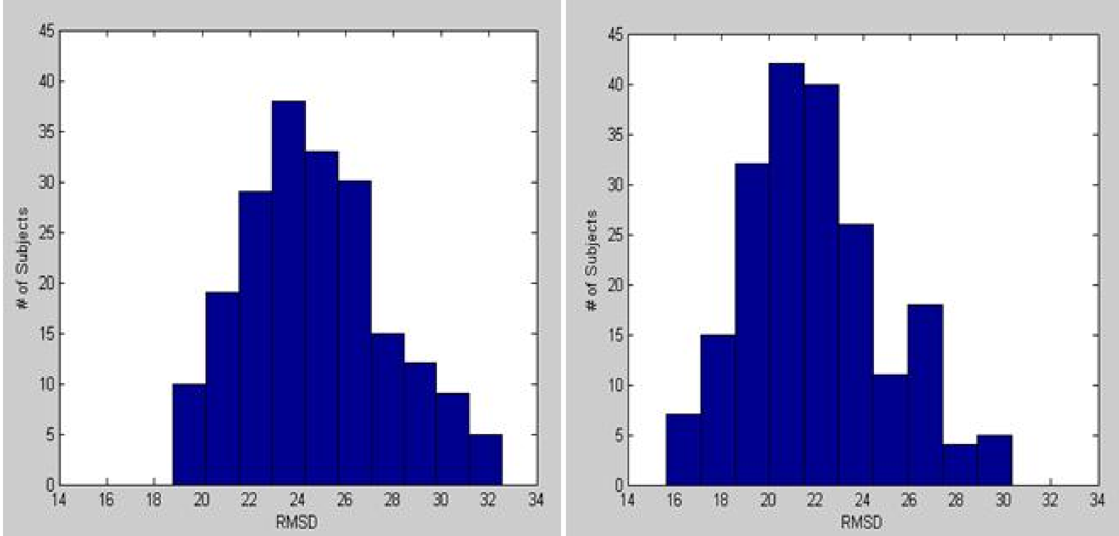

Figure 7.13: Frequency Histograms for Surface Height Errors

7.2.3 Error in Surface Heights with Smoothing

Next we show the results when surface heights were recovered from sampled as well as combined normals and smoothing step have also been performed on these normals. Again the RMS errors between the groundtruth surface heights and surface heights recovered from FB sampled smoothed normals and RMS between groundtruth surface height and combined smoothed surface normals are shown in figure (7.13). The red and green curves are for SurfaceHeightError$_{\text{(FBSampled)}}$ and SurfaceHeightError$_{\text{(Combined)}}$ respectively.

The histogram charts of subject frequencies for SurfaceHeightError$_{\text{(FBSampled)}}$ and SurfaceHeightError$_{\text{(Combined)}}$ are shown in figure (6.13). Smoothing even has reduced the error for sampled normals but still the height recovered from combined normals are better performing than heights recovered from sampled normals and are more close to the ground truth heights.

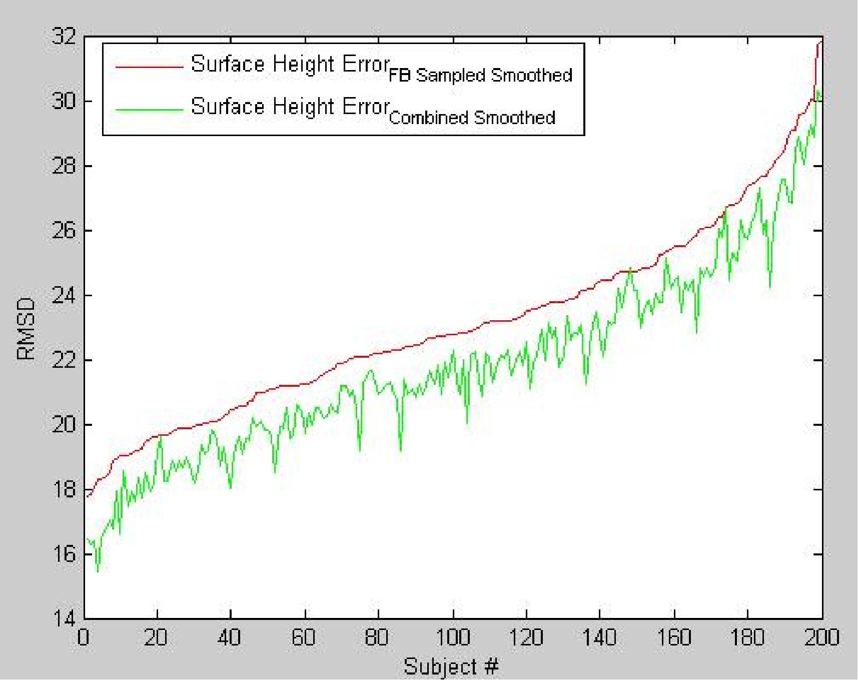

Figure 7.14: Surface Heights with Smoothing

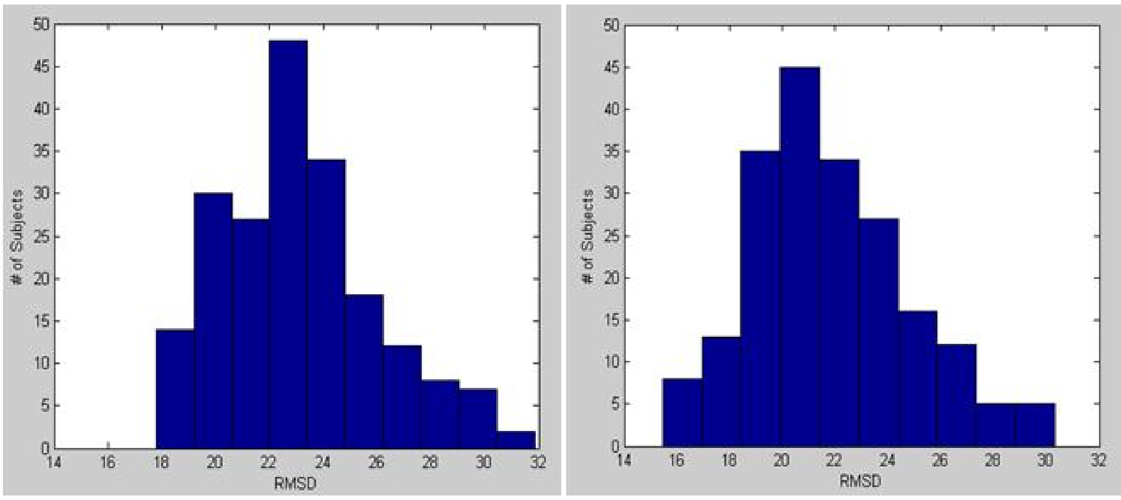

Figure 7.15: Frequency Histograms for Surface Height Errors

7.2.4 Effects of number of Samples

The only time consuming component in our technique is the slice sampling of the FB_8 distributions. For each pixel first 200 samples are used for burn in period then extrinsic mean is calculated with next 100 samples. We use $5-10$ iterations to calculate intrinsic mean; each iteration in turn may use 25, 100, 250 or 500 samples. In experiment results presented above for graphical results we have used 500 samples per iteration whereas for numerical results 25 samples per iteration are used. Increasing the number of samples increases the global consistency of sampled FB_8 surface normals resulting from SfS at the expense of time. Figure (7.16) shows one of the test images that has been used as input to the SfS algorithm; the outputted FB_8 distributions from SfS were sampled using Kume & Walker [9]. In figure (7.17) row 1 we have shown the resulting sampled illuminated surface normals when 25, 100, 250 and 500 samples were used per iteration for computing spherical median. Row 2 of figure (7.17) shows the combined normals when SfS sampled normals are combined with statistical model of needle maps. No fisher criterion or smoothing yet have been performed.

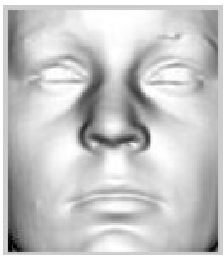

Figure 7.16: Illuminated Groundtruth Surface Normals

Table 1 shows the computation time for corresponding number of samples per iterations along with the error distances between the Groundtruth normals and sampled FB_8 normals ($E_{\mathrm{FBSampled}}$); and between Groundtruth normals and combined normals (E_{Combined}).

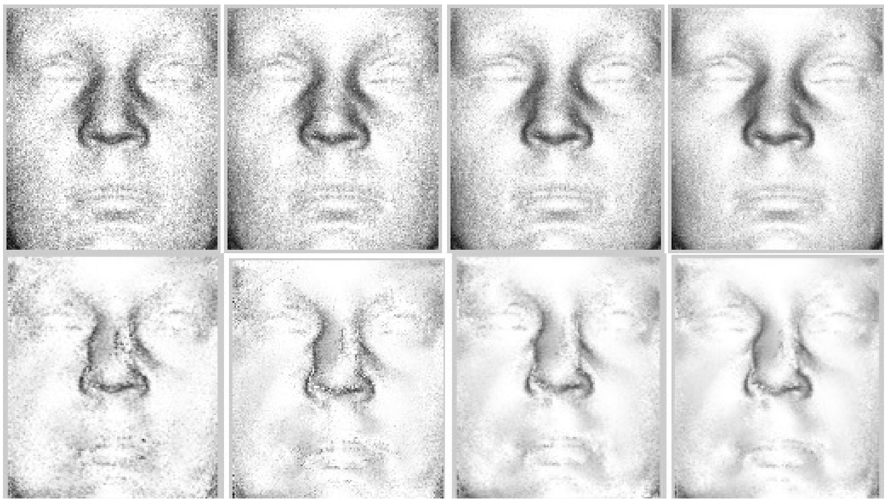

Figure 7.17: Illuminated FB Sampled (Row1) and Combined Normals (Row2)

Table 7.1: Effects of # of Samples per Iteration

Samples	Time (min)	$E_{FBSampled}$	$E_{Combined}$
25	10.5106	4.8501	4.2025
100	25.0088	4.3254	3.1110
250	54.5500	4.0267	2.8732
500	104.8722	3.7096	2.9293

7.3 Real Data

We have two types of real data available one which has ground truth available and we have captured it ourselves and other for which ground truth is not available and is based on Yale B Data base [55]. For real data lacking ground truth we can not provide numerical results and hence just using it to show visual results.

7.3.1 Subjects having Ground Truth

For real data we have taken images of five subjects fulfilling assumptions of Shape-from-Shading. The images were taken with a camera having linear response function and using single light source. These images were cropped and aligned with Smith's shape models. Figure 7.18 shows these images.

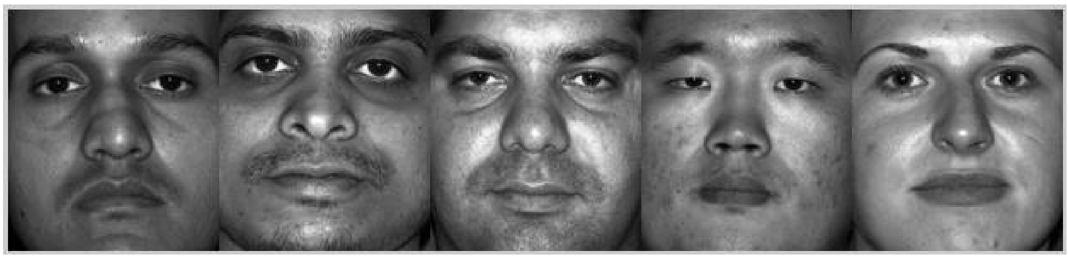

Figure 7.18: Real Test Images

We also taken $3D$ scans of these subjects using Cyberware 30303PS laser scanner. The $3D$ scans were also cropped and some meshes were filled using software MeshLab v1.2.3 to minimise the error due to the holes in the original $3D$ scans. These $3D$ scans have been shown in figure 7.19.

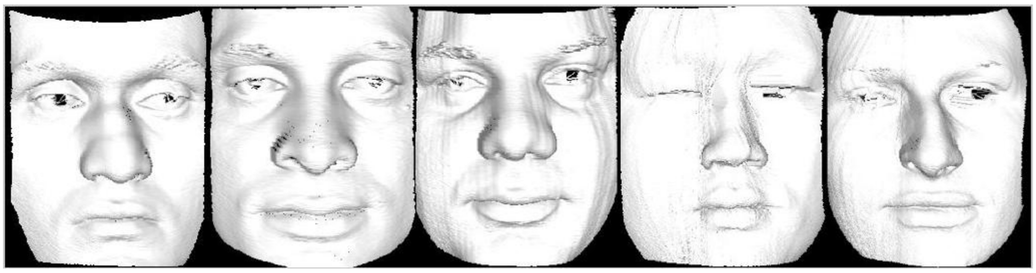

Figure 7.19: 3D Scans of the Subjects

7.3.2 Applying the Algorithm

To run the complete process for real images; we start by applying Haines Shape-from-Shading algorithm on these. Since, we are now dealing with real images the unit albedo assumption does not hold anymore. We estimated the individual albedo values for all 5 test images ignoring the white regions around the iris in the eyes and regions having specular reflection e.g. tip of the nose and forehead. The calculated individual albedo values for all these images are tabulated in table (7.1).

Table 7.2: Estimated Albedo Values for Real Images

Subject # 1	Albedo Value
1	0.73
2	0.90
3	0.98
4	0.90
5	0.90

Once the shape-from-shading is done we sampled the resulting Fisher-Bingham distributions for all images. Fisher-Bingham sampling as well as Shape-from-Shading results are shown in figure (7.20). The first row shows the outputs from the shape-from-shading algorithm and 2nd row shows resulting FB-sampled surface normals.

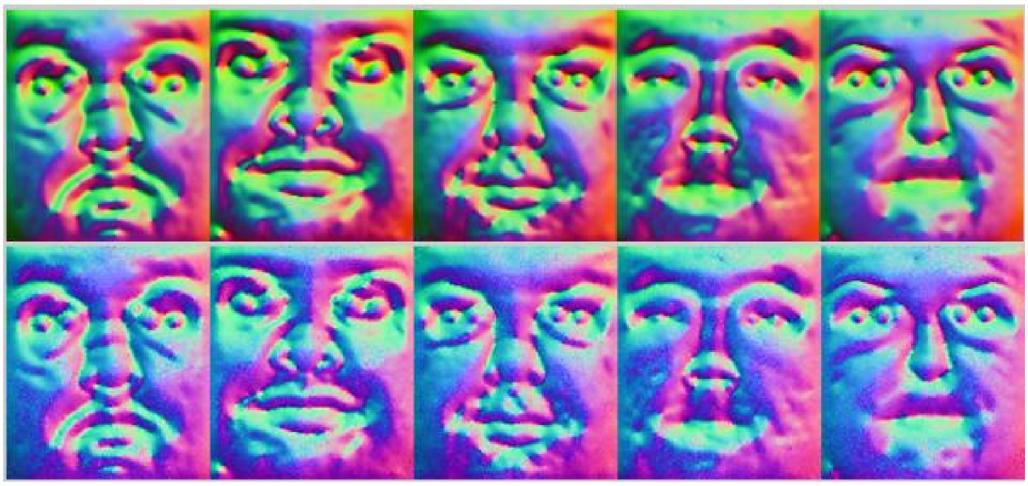

Figure 7.20: SfS Output & Sampled FB Normals

We combined Fisher-Bingham normals with Smith's statistical normal and introduce Fisher criterion. In figure (7.21) we have shown the results when Fisher-Bingham sampled and combined normals are renderred with unit light source [0 0 1] and their individual albedo values. Row 1 shows the renderred FB normals where as row 2 shows the combined normals. Be noted that no smoothing has performed yet. We can see how combining the normals has preserved the shape of subject yet got rid of salt n pepper effects of Fisher-Bingham sampling.

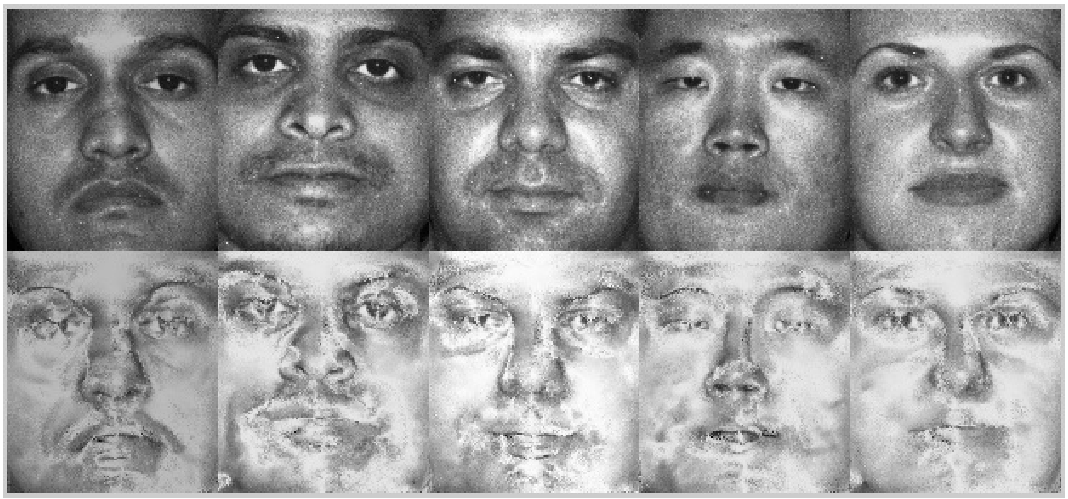

Figure 7.21: Renderred FB and Combined Normals

7.3.3 Error in Surface Height

We calculate the respective surface heights from the combined as well as Fisher-Bingham sampled normals. For presentation purposes we retrieve the normals again from these surface heights. These illuminated normals along with real images are shown in figure (7.22). Row 2 and 3 show the illuminated retrieved normals for Fisher-Bingham and coupled normals respectively.

Some readers might not notice any difference between all the images shown in one row e.g. row 2. This is because of the fact all the images share same statistical needle map and surface height model; so they have same base on which variations are added later. However, images are different regardless of their visual similarity e.g. the difference between first and second image shown in row 2 is shown in figure (7.23).

The root mean squared difference between ground truth 3D surfaces scanned by Cyberware 3030PS scanner, surface heights based on Fisher-

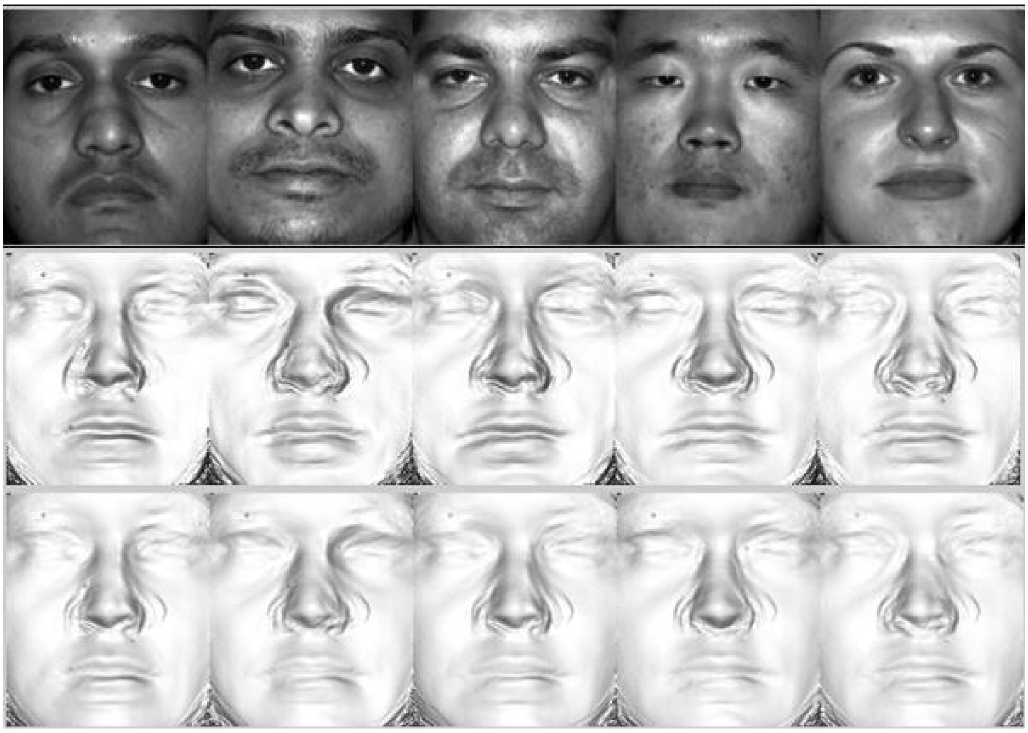

Figure 7.22: Input Images and Retrieved Normals

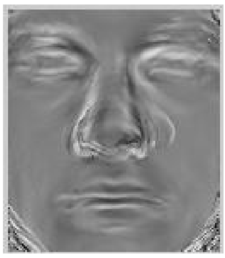

Figure 7.23: Difference between Subject 1 and Subject 2

Bingham normals and surface heights based on coupled normals are tabulated in table (7.2).

Table 7.3: Surface Height Errors

Subject #	Ground truth vs FB Sampled	Ground truth vs Coupled
1	5.2500×10^7	5.2673×10^7
2	5.2638×10^7	5.2771×10^7
3	5.2634×10^7	5.2859×10^7
4	5.2614×10^7	5.2686×10^7
5	5.2603×10^7	5.2677×10^7

7.3.4 Subjects lacking Ground Truth

We have used first 10 images from Yale B Data Base [55] as shown in figure (7.24) . The data base has cropped and aligned images so we only have to resize these images to bring them in correspondence with Smith's models. Since, these are real images so we have computed their respective albedo values ignoring the specular regions and white pigment of eyes. The estimated albedo values for these 10 images are shown in Table (7.3).

Table 7.4: Estimated Albedo Values for Yale B Images

Subject #	Albedo Value	Subject #	Albedo Value
1	0.8824	6	0.8627
2	0.8980	7	0.7098
3	0.9490	8	0.7843
4	0.6196	9	0.7961
5	0.8784	10	0.9686

The output from SfS for these real images and sampled Fisher-Bingham normals are shown in figures (7.25) and (7.26) respectively.

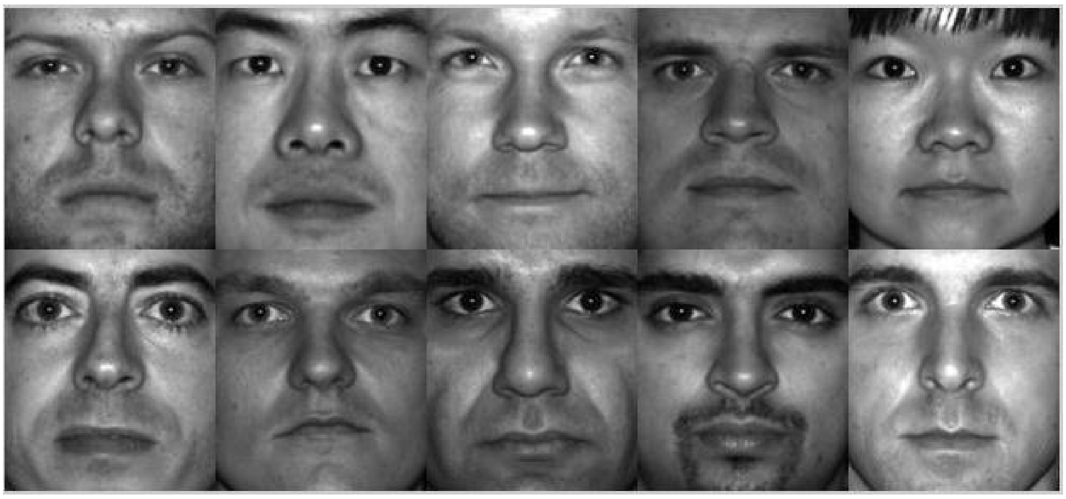

Figure 7.24: Yale B 1st 10 Subjects

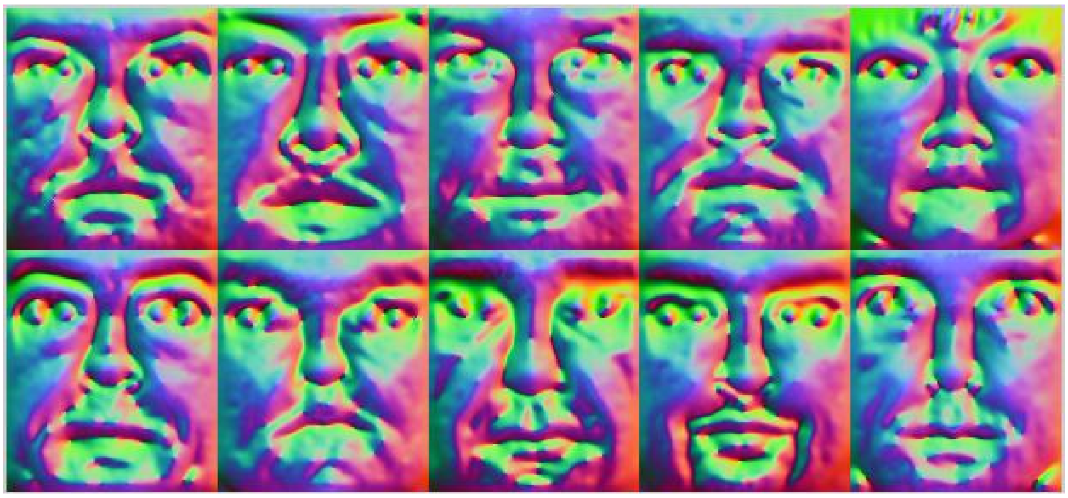

Figure 7.25: SfS Output for Yale B Images

We have shown the sampled as combined surface normals illuminated with their respective albedo values and unit light source [0 0 1] in figure (7.27).

Next the smoothing is performed on the normals and then surface heights are calculated but unfortunately we do not have ground truth surface heights to compare with.

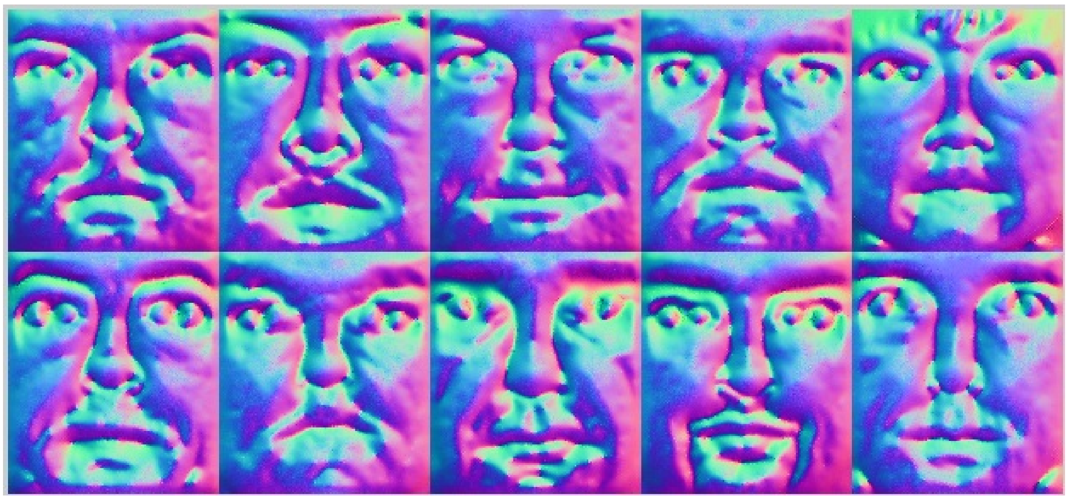

Figure 7.26: Sampled Fisher-Bingham Normals

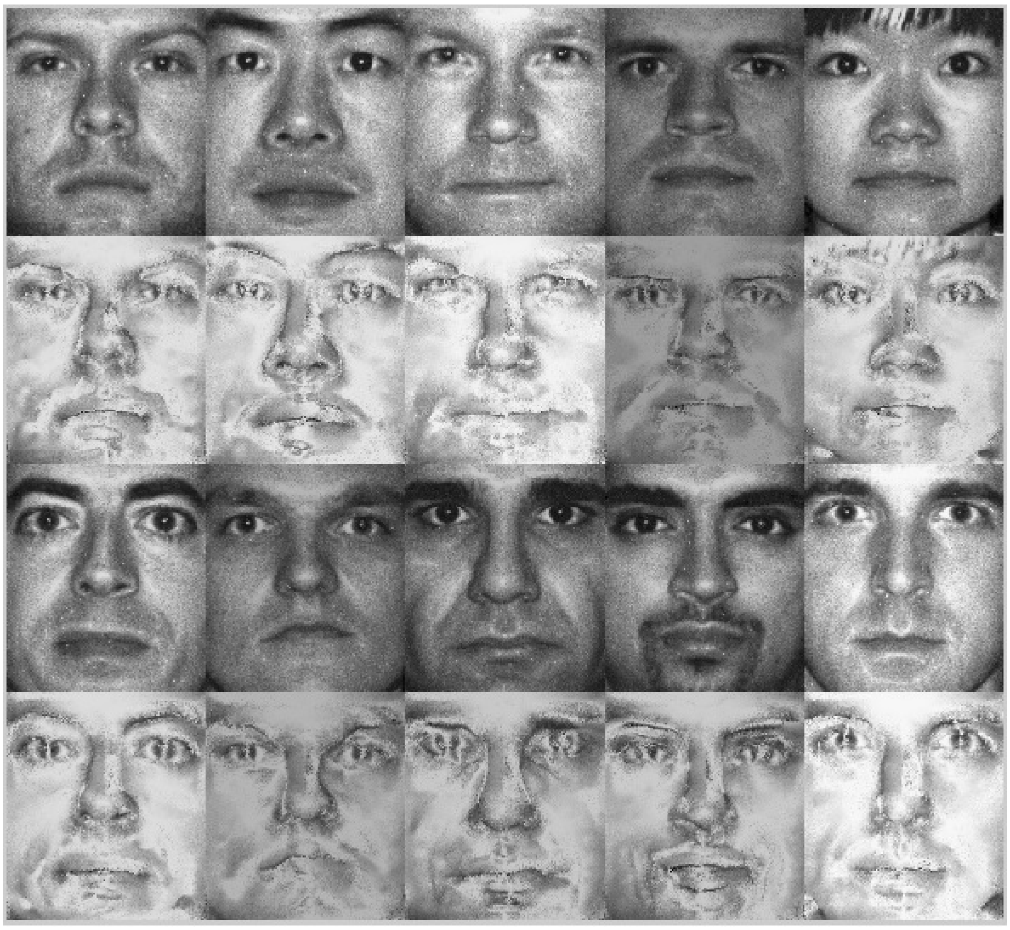

Figure 7.27: Renderred FB and Combined Normals

Chapter 8

Conclusions

In this thesis we have listed a brief review of Shape-from-Shading and statistical models of face shapes. We have particularly discussed a recent approach of Shape-from-Shading based on directional distributions and statistical models of needle maps and surface heights.

We have combined this probabilistic Shape-from-Shading algorithm with statistical models of face shapes in a probabilistic framework. A slice sampling algorithm was used to generate samples from Fisher-Bingham distributions that were used to represent surface orientations in shape-from-shading algorithm. Concepts of differential geometry, specifically logarithmic and exponential mapping, have been used to transfer points lying on the spherical manifold to the tangent plane and vice versa. Independent Gaussian distributions were fitted to the surface orientations arising from Shape-from-Shading and Shape models. On the tangent plane these Gaussian distributions were combined to give more probable surface orientation in the form of mean of resultant product normal distribution.

The Fisher criterion was used along with smoothing to deal with outliers and discontinuties arising from sampling of distributions. We have tested the combined represenatation of surface orientations for synthetic as well as real data. We have shown how the use of statistical model of face shapes have improved error difference between the actual and recovered surface normals and surface heights using Iterative closest point algorithm. We have discussed how the number of samples used per iteration for the computation of spherical median affect the global consistency of surface normals from shape-from-shading and hence improve the visual quality of combined normals. For visual presentation of results we have used 500 samples per iteration whereas for numerical results 25 samples per iteration were used to compute spherical median of FB samples. We have shown even using less number of samples per iteration for computation of spherical median results in the improvement of surface normal quality. Increasing the number of samples at the expense of computation time though helps denoising the surface normals and hence increases their quality even more. We have used two types of data sets for real images i.e. one having the ground truth and one lacking the ground truth. For real images the effects of albedo were also discussed. The improvement in the quality of combined resultant normals was presented using visual results.

8.1 Future Work

The two models that we have combined in this thesis originally give results separately in different planes and using differnt distributions. Haines Shape-from-Shading algorithm [3] gives surface normals using Fisher-Bingham distributions on a unit sphere whereas Smith's statistical model [1] gives surface normals using Gaussian distributions on the tangent plane of a unit sphere. To combine these two models their respective outputs should be on the same plane and represented by same distributions. So we represented Haines surface normals on the tangent plane using Gaussian distributions with the help of slice sampling of Fisher-Bingham distribution and machinary of Log and Exponential mapping. The route we took to combine these two approaches was the second best of the possible solutions. Fisher-Bingham distribution being a directional distribution models directional data such as surface normals more accurately as compared to an ordinary distribution such as Gaussian.

So the best way to combine these normals is to represent Smith's surface normal model using Fisher-Bingham distributions instead of using Gaussian distributions on the tangent plane to represent Haines surface normals. Unfortunately we could not find a proper way to fit Fisher-Bingham distributions to statistical normals and hence we opt the second best possible route. In our future work we will explore this possibility more deeply and try to use Fisher-Bingham or any other more suitable directional distributions to represent Smith's statistical normals. Our prelimiary ideas in this respect can be grouped as two different tracks.

8.1.1 Approach 1

To fit a Fisher distribution to the statistical normal instead of Fisher-Bingham distribution seems more tractable; so in our first attempt we will try to explore this possibility. This approach can be listed as a 4 step algorithm.

1. Fit a Fisher distribution to the each surface normal from Smith's statistical model.
2. Approximate each FB_8 distribution from Haies Shape-from-Shading method as a sum of Fisher distributions.
3. Convolve the Fisher distributions resulting from Step 1&2.
4. Fit a Fisher-Bingham (FB_8) distribution to the mixture model of Fisher distributions arising in Step 3.

8.1.2 Approach 2

A more attractive but less tractable approach would be to fit Fisher-Bingham distributions directly to the statistical surface normals and then combine the two FB_8 for each pixel. We will also explore this track in our future work.

Appendix

Origin of the Fisher-Bingham Distribution

A constrained distribution works with the vector representation of a direction, taking a classical statistics distribution on the vectors and constraining it to unit vector lengths only [3]. The von-Mises distribution [16] is an example of a constrained distribution. This is a $2D$ Gaussian distribution having an arbitrary mean and co-variance matrix a multiple of the identity matrix. This restricted Gaussian distribution can be re-paramaterised as the PDF,

$$P_{vM}(\hat{\mathbf{x}}; \mathbf{u}) \propto \exp(\mathbf{u}^T \mathbf{x})$$

where, $\mathbf{x} \in R^2$,$|\mathbf{x}|$ is the considered direction and $\mathbf{u} \in R^2$ is the parameter.If the parameter vector is zero we will have a uniform distribution otherwise $\frac{\mathbf{u}}{|\mathbf{u}|}$ indicates the direction with highest probability while $-\frac{\mathbf{u}}{|\mathbf{u}|}$ gives the direction with lowest probability. The concentration parameter k is defined as $k = |\mathbf{u}|$. The higher the concentration k the more the distribution is concentrated around $\frac{\mathbf{u}}{|\mathbf{u}|}$. Using the concentration parameter k the distribution can also be described as,

$$P_{vM}(\hat{\mathbf{x}}; \mathbf{u}) \propto e^{k \cos \theta}$$

where, θ is the angle between two directions. Fisher [17] extended this idea to projecting a $3D$ distribution for directions in R^3.The definition of the Fisher distribution is same as 1st Equation except it uses $3D$ vectors rather than $2D$ vectors. This idea can be extended to arbitrary dimensions and in the n dimensional case referred to jointly as the von-Mises-Fisher distribution [3, 16].The Bingham distribution is based on a Gaussian distribution where the mean is set to the zero vector but the co-variance matrix is unconstrained,

$$P_B(\hat{\mathbf{x}}; \mathbf{A}) \propto \exp(\hat{\mathbf{x}}^T \mathbf{A} \hat{\mathbf{x}})$$

where matrix $\mathbf{A}$ is the inverse of the covariance matrix and is symmetric.

Deriving Normal Equation

We now list some facts about matrix derivatives without going into their details.
For a function $f : R^{m*n} \longmapsto R$ mapping from m-by-n matrix to the real numbers, we define the derivative of f with respect to A as,

$$\bigtriangledown_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} \cdots \frac{\partial f}{\partial A_{1n}} \\ \vdots \quad \ddots \quad \vdots \\ \frac{\partial f}{\partial A_{m1}} \cdots \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

Hence, the gradient $\bigtriangledown_A f(A)$ is itself an m-by-n matrix, whose element (i, j) is given by $\frac{\partial f}{\partial A_{ij}}$. Next we define the *trace* operator written as "tr" and defined for a square matrix A as, $trA = \sum_{i=1}^{n} A_{ii}$ i.e. sum of the diangonal elements of a square matrix. We will now list a few properties of trace operator,

$$tr\, a = a$$

where 'a' is a scalar.

$$tr\, AB = tr\, BA$$

A , B etc. are square matrices, similarly

$$tr\, ABC = tr\, CAB = tr\, BCA$$

$$tr\, ABCD = tr\, DABC = tr\, CDAB = tr\, BCDA$$

$$tr\, A = tr\, A^T$$

$$tr\, (A + B) = tr\, A + tr\, B$$

$$tr\, aA = atr\, A^T$$

Here are some facts about matrix derivatives,

$$\bigtriangledown_A tr\, AB = B^T$$

$$\bigtriangledown_{A^T} f(A) = \bigtriangledown_A f(A)^T$$

$$\bigtriangledown_A tr\, ABA^T C = CAB + C^T AB^T$$

From Equation (4.55) we have

$$\mathbf{J}(b) = \frac{1}{2}\|\mathbf{\Psi} b - \mathbf{G}\|^2 = \frac{1}{2}(\mathbf{\Psi} b - \mathbf{G})^T(\mathbf{\Psi} b - \mathbf{G})$$

So,

$$\bigtriangledown_b \mathbf{J}(b) = \frac{1}{2} \bigtriangledown_b (b^T \boldsymbol{\Psi}^T \boldsymbol{\Psi} b - b^T \boldsymbol{\Psi}^T \mathbf{G} - \mathbf{G}^T \boldsymbol{\Psi} b + \mathbf{G}^T \mathbf{G})$$

$$\bigtriangledown_b \mathbf{J}(b) = \frac{1}{2} \bigtriangledown_b \, tr \, (b^T \boldsymbol{\Psi}^T \boldsymbol{\Psi} b - b^T \boldsymbol{\Psi}^T \mathbf{G} - \mathbf{G}^T \boldsymbol{\Psi} b + \mathbf{G}^T \mathbf{G})$$

$$\bigtriangledown_b \mathbf{J}(b) = \frac{1}{2} \bigtriangledown_b \, (tr \, b^T \boldsymbol{\Psi}^T \boldsymbol{\Psi} b - 2tr \, \mathbf{G}^T \boldsymbol{\Psi} b)$$

$$\bigtriangledown_b \mathbf{J}(b) = \frac{1}{2} (\boldsymbol{\Psi}^T \boldsymbol{\Psi} b + \boldsymbol{\Psi}^T \boldsymbol{\Psi} b - 2\boldsymbol{\Psi}^T \mathbf{G})$$

Presentations

1. Touqeer Ahmad, Richard C. Wilson. *Shape-from-Shading: Combining Probabilistic & Statistical Models.* Poster presentation of the intermediate progress of MSc work in ICVSS 2010, Italy.

2. Touqeer Ahmad, Richard C. Wilson, William A. P. Smith, Tom S. F. Haines. *Combining Probabilistic Shape-from-Shading & Statistical Facial Shape Models.* Accepted for publication and presentation in ICIAP 2011-Ravenna, Italy.

Bibliography

[1] William A. P. Smith. *Statistical Methods For Facial Shape-from-Shading and Recognition.* PhD thesis, University of York, 2007.

[2] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Comput. Vis. Image Underst.*, 61:39-59, 1995.

[3] Tom S. F. Haines *Integrating Shape-from-Shading & Stereopsis.* PhD thesis, University of York, 2009.

[4] USF HumanID 3D Face Database, Courtesy of Sundeep. Sarkar, University of South Florida, Tampa, FL.

[5] V. Blanz and T. Vetter. Face Recognition based on fitting a 3D morphabale model. *IEEE Trans. Pattern Anal. March. Intell.* , 25(9):1063 - 1074, 2003.

[6] L. Sirovich. Turbulence and the dynamics of coherent structures. *Quart. Appl. Math.*, XLV(3):561 - 590, 1987.

[7] P. T. Fletcher, S. Joshi, C. Lu, and S. M. Pizer. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Trans. Med. Imaging.* , 23(8):995 - 1005, 2004.

[8] X. Pennec. Probabilities and statistics on Riemannian manifolds: basic tools for geometric measurements. In *Proc. IEEE Workshop on Nonlinear Signal and Image Processing*, 1999.

[9] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern Anal. March. Intell.* , 10(4):439 - 451, 1988.

[10] Andrew Ng. Lecture #1 of Machine Learning. Stanford engineering everywhere, Stanford University.

[11] A. Kume and S. G. Walker. On the Fisher-Bingham distribution. *Stat. and Comput.*, 19:167 - 172, 2009.

[12] G. Casella and E. I. George Explaining the Gibbs Sampler *The American Stat.*, 46(3):167 - 174, 1992.

[13] P. L. Worthington and E. R. Hancock. New constraints on data-closeness and needle map consistency for shape-from-shading. *IEEE Trans. Pattern Anal. March. Intell.*, 21(12):1250 - 1267, 1999.

[14] K. M. Lee and C. J. Kuo. Shape from shading with perspective projection. *CVGIP: Image Understanding*, 59(2):202 - 212, 1994.

[15] B. K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View.* PhD thesis, Massachusetts Institute of Technology, 1970.

[16] K. V. Mardia and P. E. Jupp. *Directional Statistics.* John Wiley and Sons Ltd, 2000.

[17] R. Fisher. Dispersion on a sphere *Proceedings of the Royal Society of London, Series A, Math. and Physical Sciences*, 217:295 - 305, 1953.

[18] P. F. Felzenszwalb and D. P. Huttenlocher Efficient belief propagation for early vision. *Computer Vision and Pattern Recognition*, 1:261 - 268, 2004.

[19] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *Pattern Analysis and Machine Intelligence*, 13(7):680 - 702, 1991.

[20] K. Luebke and C. Weihs. Improving Feature Extraction by Replacing the Fisher Criterion by an Upper Error Bound. *Pattern Recognition*, 38(2005):2220 - 2223, 2005.

[21] S. Maciej and M. Witold. Versatile Pattern Recognition System Based on Fisher Criterion *Proceedings of the KOSYR*, 2003:343 - 348, 2003.

[22] J. Sammon An Optimal Discriminnat Plane *IEEE Transactions on Computers*, 19:826 - 829, 1970.

[23] P. J. Besl and N. D. Mckay. A method for registration of 3-d shapes. *IEEE Transactions on Pat. Analysis and Mach. Intelligence.*, 14(2):239 - 256, 1992.

[24] J. M. Phillips, R. Liu and C. Tomasi. Outlier Robust ICP for Minimizing Fractional RMSD *6th International Conference on 3-D Digital Imaging and Modeling.*, 2007:427-434, 2007.

[25] K. Ikeuchi and B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence.*, 1981:141 -184, 1981.

[26] M. J. Brooks and B. K. P. Horn. Shape and source from shading. *Artificial Intelligence.*, 1985:932 -936, 1985.

[27] R. Zhang, P-S. Tsai, J. E. Cryer, and M. Shah. Shape from Shading: A survey. *Analysis and Machine Intelligence.*, 21(8):690 -706, 1999.

[28] M. Bichsel and A. P. Pentland. A simple algorithm for shape from shading. *Computer Vision and Pattern Recognition.*, 1992:459 -465, 1992.

[29] C-H. Lee and A. Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence.*, 26(2):125 -143, 1985.

[30] A. Pentland. Shape information from shading: A theory about human perception. *Computer Vision.*, 1988: 404 -413, 1988.

[31] P-S. Tsai and M. Shah. Shape from shading using linear approximation. *Image and Vision Computing.*, 12:487 -498, 1994.

[32] A. Robles-Kelly and E. R. Hancock. A graph-spectral approach to shape-from-shading. *Image Processing.*, 13 (7):912 -926, 2004.

[33] M. Louw and F. Nicolls. A loopy belief propagation approach to the shape from shading problem. *International Conference on Computer Vision Theory and Applications.*, 2007.

[34] M. Louw and F. Nicolls. A spatially multiresolution, MRF optimization based approach to the shape from shading problem. *Visualization, Imaging, and Image Processing.*, 2007.

[35] B. Potetz. Efficient belief propagation for vision using linear constraint nodes. *Computer Vision and Pattern Recognition.*, 2007:1-8, 2007.

[36] T. S. F. Haines and R. C. Wilson. Combining Shape-From-Shading and Stereo Using Gaussian-Markov Random Fields. *International Conference on Pattern Recognition.*, 2008.

[37] T. S. F. Haines and R. C.Wilson. Belief Propagation with Directional Statistics for solving the Shape-from-Shading problem. *European Conference on Computer Vision.*, 2008.

[38] J. J. Atick, P. A. Griffin, and A. N. Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computing.*, 8(6):1321 - 1340, 1996.

[39] R. Dovgard and R. Basri. Statistical symmetric shape from shading for 3d structure recovery of faces. *European Conference on Computer Vision.*, 2:99113, 2004.

[40] W. A. P. Smith and E. R. Hancock Recovering Facial Shape using a Statistical Model of Surface Normal Direction *IEEE Trans. on Pattern Analysis and Machine Intelligence.*, 28(12): 1914-1930, 2006.

[41] W. A. P. Smith and E. R. Hancock Recovering Facial Shape using a Statistical Surface Normal Model *International Conference on Image Processing*, 2: 113 -116, 2005.

[42] W. A. P. Smith and E. R. Hancock Recovering Facial Shape and Albedo using a Statistical Model of Surface Normal Direction *International Conference on Computer Vision*, 2005: 1550 -5499, 2005.

[43] W. A. P. Smith and E. R. Hancock Coupled Statistical Face Reconstruction *Comp. Analysis of Images and Patterns*, 3691: 153 -161, 2005.

[44] W. A. P. Smith and E. R. Hancock A Model-Based Method for Face Shape Recovery *Pattern Recognition and Image Analysis*, 3522: 268 - 276, 2005.

[45] M. Castelan, W. A. P. Smith and E. R. Hancock Approximating 3D Facial Shape from Photographs Using Coupled Statistical Models *Progress in Pattern Recognition, Image Analysis and Applications*, 4225: 89 -98, 2006.

[46] M. Castelan, W. A. P. Smith and E. R. Hancock A Coupled Statistical Model for Face Shape Recovery *Structural, Syntactic and Statistical Pattern Recognition* , 4109: 898 -906, 2006.

[47] W. A. P. Smith and E. R. Hancock Statistical Methods for Surface Integration *Mathematics of Surfaces*, 4647: 427 -441, 2007.

[48] M. Turk and A. Pentland. Face recognition using eigenfaces *In Proc. of Computer Vision and Pattern Recognition*, 1991: 586 -591, 1991.

[49] L. Sirovich. Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics* , 45: 561 -571, 1987.

[50] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 17(7):711 -720, 1997.

[51] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Euorpean Conference on Computer Vision*, 1998:484 -498, 1998.

[52] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61:38 -59, 1995.

[53] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. Coupled-view active appearance models. *In Proc. of British Machine Vision Conference*, 1:52 -61, 2000.

[54] R. M. Neal. Slice Sampling *Department of Statistics, University of Toronto*, Technical Report # 2005, 2000.

[55] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643 -660, 2001.

[56] T. Rindfleisch. Photometric method for lunar topography. *Photogramm. Eng.*, 32:262 -276, 1966.

[57] J. van Diggelen. A photometric investigation of the slopes and heights of the ranges of hills in the Maria of the moon. *Bull. Astron. Inst. Netherlands.*, 11, July 1951