# การสร้างภาพหน้ามุมมองสามมิติจากภาพหน้าสองมิติภาพเดียวโดยใช้วิธีหาแนวตั้งฉากของนอร์แมลเวกเตอร์และวายเรโชว์

นางสาวณัฐชามญช์ ศรีจำเริญรัตนา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และ
วิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

# STEREOSCOPIC FACE RECONSTRUCTION FROM A SINGLE 2D FACE IMAGE USING ORTHOGONALITY OF NORMAL SURFACE AND Y-RATIO

Miss Natchamol Srichumroenrattana

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy Program in Computer Science and Information Technology

Department of Mathematic and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2012

| | |
|---|---|
| Thesis Title | STEREOSCOPIC FACE RECONSTRUCTION FROM A SINGLE 2D FACE IMAGE USING ORTHOGONALITY OF NORMAL SURFACE AND Y-RATIO |
| By | Miss Natchamol Srichumroenrattana |
| Field of Study | Computer Science and Information Technology |
| Thesis Advisor | Assistant Professor Rajalida Lipikorn, Ph.D. |
| Thesis Co-advisor | Professor Chidchanok Lursinsap, Ph.D. |

---

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

.....Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr. rer. nat.)

THESIS COMMITTEE

.....Chairman

(Assistant Professor Nagul Cooharojananone, Ph.D.)

.....Thesis Advisor

(Assistant Professor Rajalida Lipikorn, Ph.D.)

.....Thesis Co-advisor

(Professor Chidchanok Lursinsap, Ph.D.)

.....Examiner

(Suphakant Phimoltares, Ph.D.)

.....Examiner

(Associate Professor Wiwat Vatanawood, Ph.D.)

.....External Examiner

(Assistant Professor Amornrit Puttipipatkajorn, Ph.D.)

ณัฐชามญฑ์ ศรีจำเริญรัตนา : การสร้างภาพหน้ามุมมองสามมิติจากภาพหน้าสองมิติภาพเดียวโดยใช้วิธีหาแนวตั้งฉากของนอร์แมลเวกเตอร์และวายเรโชว์. (STEREOSCOPIC FACE RECONSTRUCTION FROM A SINGLE 2D FACE IMAGE USING ORTHOGONALITY OF NORMAL SURFACE AND Y-RATIO) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.รัชลิดา ลิปิกรณ์,อ. ที่ปรึกษาวิทยานิพนธ์ร่วม:ศ.ดร. ชิดชนก เหลือสินทรัพย์,93 หน้า.

วิทยานิพนธ์นี้เสนอการสร้างภาพสามมิติจากภาพถ่ายใบหน้าคนสองมิติโดยใช้ภาพถ่ายเพียงภาพเดียว ภายใต้โมเดลของแลมเบอร์ท ซึ่งประกอบไปด้วย 5 ส่วนหลัก ได้แก่ ส่วนการเตรียมข้อมูลจากฐานข้อมูลภาพ ส่วนการประมาณค่าเวกเตอร์ที่ตั้งฉากกับพื้นผิว ส่วนการคำนวณค่าความสามารถในการสะท้อนแสงของพื้นผิว ส่วนการคำนวณหาค่าเวกเตอร์ที่ตั้งฉากกับพื้นผิว และส่วนการคำนวณหาค่าความสูงจริงของแต่ละจุดบนภาพ ในส่วนแรกคือส่วนการเตรียมข้อมูลจากฐานข้อมูลภาพ เป็นการสร้างรูปร่างและภาพใบหน้าเฉลี่ยจากฐานข้อมูลภาพ ส่วนที่สองคือส่วนกระประมาณค่าเวกเตอร์ที่ตั้งฉากกับพื้นผิวของภาพ กระทำด้วยวิธีการแปลงค่าเวกเตอร์ที่ตั้งฉากกับพื้นผิวของภาพโมเดลเฉลี่ยมาสู่ภาพใบหน้าที่ต้องการสร้าง ส่วนที่สามเป็นส่วนการคำนวณค่าความสะท้อนแสง ด้วยการคำนวณหาค่าความสามารถในการสะท้อนแสงของแต่ละจุดบนภาพโดยต้องรู้ข้อมูลของทิศทางแหล่งกำเนิดแสง และค่าความสว่างของแต่ละจุดบนภาพใบหน้าที่ต้องการ ส่วนที่สี่เป็นการคำนวณค่าเวกเตอร์ที่ตั้งฉากกับพื้นผิว โดยเริ่มจากการหาอัตราส่วนของค่าเวกเตอร์ที่ตั้งฉากกับพื้นผิวระหว่างทิศทางในแนวแกนตั้งเมื่อเปรียบเทียบกับในแนวแกนนอน แล้วจึงทำการคำนวณหาค่าเวกเตอร์ที่ตั้งฉากกับพื้นผิว จากนั้นจึงดำเนินการในส่วนที่ห้าคือคำนวณหาความสูงแต่ละจุดบนใบหน้า หลังจากนั้นจึงนำผลมาคำนวณค่าความถูกต้องและแสดงผล ซึ่งได้ทำการทดสอบกับภาพใบหน้าหนึ่งร้อยสิบสี่ภาพ จากฐานข้อมูลใบหน้าสองชุด เปรียบเทียบกับวิธีการ minimization พบว่าวิธีการที่นำเสนอนี้ให้ผลลัพธ์ที่ถูกต้องมากกว่า และใช้เวลาประมวลผลน้อยกว่ามาก

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2555 ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม .....

# # 5073918623 : MAJOR COMPUTER SCIENCE
KEYWORDS : SHAPE FROM SHADING/ LAMBERT MODEL/ NORMAL SURFACE/ SINGLE IMAGE/ 3D FACE RECONSTRUCTION

NATCHAMOL SRICHUMROENRATTANA: STEREOSCOPIC FACE RECONSTRUCTION FROM A SINGLE 2D FACE IMAGE USING ORTHOGONALITY OF NORMAL SURFACE AND Y-RATIO, THESIS ADVISOR: ASST. PROF RAJALIDA LIPIKORN, Ph.D., CO-ADVISOR : PROF. CHIDCHANOK LURSINSAP, Ph.D., 93 pp.

This research proposes a 3D face reconstruction method with single image and condition of Lambertian model. This method can be divided into 4 parts: surface albedo estimation, normal surface calculation, actual height calculation and accuracy evaluation. The first part which is surface albedo estimation started with initial normal surface from Average face data and morphed it to match an input image by my pattern morphing method. Then the estimated normal surface, Intensity map and Light source vector of input image are used to estimate the albedo of each point on input image. The second part, normal surface calculation, started with finding the relation equation of Normal vector in X and Y axis and calculate the normal surface. The third part is pixels' height calculation. And the last part is a accuracy evaluation of the method and surface display. The experiment was done with 114 face's images from 2 face's database compared with the minimization approach and the result shown that my approach gave the better accuracy and consumed the less process time.

Department : Mathematics and Computer Science Student's Signature .....
Field of Study : Computer Science ..... Advisor's Signature .....
Academic Year : 2012 ..... Co-advisor's Signature .....

# Acknowledgements

This research has been supported by the grant under the strategic scholarships for Frontier Research Network for the Ph.D. Program Thai Doctoral degree from the office of the Higher Education Commission, Thailand.

I would like to express my deepest gratitude to my advisor, Assistant Professor Dr. Rajalida Lipikorm, for encouraging and supporting me during my study. She gives me advices and ideas to derive the optimal solution for my research.

I would like to thank Professor Dr. Chidchanok Lursinsap for accepting me as a member of his research group Advanced Virtual and Intelligence Computing(AVIC) Center.

I would like to thank Dr. Kairat Jaroenrat for his support, continuous encouragement, understanding, and love.

Finally, I would like to dedicate my Ph.D. to my family for giving me the opportunity to pursue the degree from the best institution and supporting me throughout my life.

# Contents

Chapter Page

# List of Figures

Figure Page

Figure Page

# List of Tables

Table Page

# CHAPTER I

# INTRODUCTION

## 1.1 Introduction

### 1.1.1 Overview

From the past decades, human has tried to communicate and trained the computer to imitate human behaviors. One of them is to recover three-dimensional contour of objects from a two-dimensional image, which in this research, I consider only human face images and to let a computer to reconstruct three-dimension contour from two-dimension into three-dimension contour. The contour recovery of human face converts a two-dimensional image to a three-dimensional contour image model.

In general, a two-dimensional image from a digital camera can keep details of the target objects such as eye color, nose and mouth of a human face image and display it on a computer monitor or other display. But one thing that missing is the depth of an object because it disappears after taking a picture(photo).

Human eyes can distinguish objects in the environment around them, such as near or far objects, the movement of objects even the color of objects. Humans can understand the depth of vision from reflection of light from object surface to the eyes and estimate image's depth immediately but we do not know how human brain processes the data.

The objective of three-dimensional face reconstruction from a human face image in my research is the accuracy of the three-dimension face reconstruction process. The input data of my research is limited to be a frontal face image under fixed direction of light source and diffuse surface reflectance to recover normal surface. The final output of my research are depth map of two-dimensional image that can be reconstruct to three-dimensional contour which are as close to the depth as seen by human eyes.

To support this idea, there are several ways to obtain the contour of an object such as shape from shade(SFS) and shape from motion(SFM). I classify input data for reconstruction process into 2 types. The first type is data from special tools or equipments such as

three-dimensional scanner and x-ray machine. The second type is data from two-dimension images and construct a three-dimensional image from a single two-dimension image or multiple two-dimension images.

### 1.1.2 Literature Review

In 1970, Berthold Horn has done his doctoral dissertation [1] which is the starting point of Shape From Shade researches. Partial differential equation was used to analyse line distance of surface curves and initial a singular point and ambiguity line of curve. He used the curve of spherical and characteristic strips that grow from initial surface with a similar point of intensity to calculate the depth from shading. SFS techniques can be divided into four groups [2] : minimization approaches, shape propagation approaches, local approaches and linear approaches. Untill 2009, Jochen Wilhelmy [3] has specified the fifth group that is a group of belief propagation approaches. Minimization approaches solve a problem by minimizing an energy function. Propagation approaches propagate the shape information from a set of surface points (e.g., singular points) to the whole image. Local approaches obtain shape depended on the assumption of surface type. Linear approaches calculate the solution depended on the conversion from non linear to linear equation of the reflectance map. Belief propagation approaches use a factor graph to calculate the most probable solution.

1. Minimization approaches

   Minimization approaches solve the problem by iterative minimizing of energy function with surface constrains. Firstly, Ikeuchi and Horn [4] proposed minimization approaches with surface gradients retrieve. This propose requires the brightness constraint to reconstruct shape which has the same brightness as the input image in every surface point. Another constraint is moothness constraint which force normal vector and it's neighbor to be smooth reconstructed. A drawback of this approach is that it tends to make too smooth of the recovered normal surface, then Zheng and Chellappa [5] avoid the too smooth reconstruction by discarding the smoothness term of the energy function and making the gradients of the reconstructed image to be close to the gradients of the input image instead. Horn and Brooks [6] found a variational smoothness approach constraint and use it for Shape-from-Shading with a drawback that the gradient along some closed curve has

to be specified. Because of the difficulties on the convergence of the algorithm, Frankot and Chellappa [7] integrate Brooks and Horn's algorithm with using the discrete Fourier transform to recover integrable surfaces by using orthogonal projections. But this algorithm has a drawback that it is a very complex transformation of whole image data and cannot be seperated to parallel implement in grid computing. After that, Horn [8]changed the smoothness constraint in his approach into an integrability constraint to have a new term for the cost function that integrate by fusing height and gradient recovery together. Unfortunately, by using this function, there is a problem of slow convergence. Moreover, linearizing the reflectance map and re-introducing a smoothness term are attempts to solve the problem of instability. In the part of findng the direction of the light source, there is a method proposed by Brooks and Horn [9] that is an iterative method that can estimates both surface shape and light source direction. Later, Zheng and Chellappa [5] proposed a method of estimating the light source direction and the surface albedo. Harrison, A. and Joseph [10] presented depth-map reconstruction method which is used to make maximum likelihood (ML) and image object to directly relate between a surface and gradients fields of noise under reconstruct depth of Gaussian model of image. They suggest this ML method to estimate depth maps when shadowed pixel are filtered from surface normal estimation. W. Smith and E. Handcock [11] describe a statistical model that can be used to recover a facial albedo map from a single image before reconstructing a three-dimension shape. Atkinson, G.A. and Hancock, E.R.[12] surface normal using Fresnel theory by estimate correspondence from 2 view and matching the reflectance properties between surface orientation and pixel bright. Gang Zeng and Paris, S [13] proposed surface representation patchwork with using multiple image to extract feature of corner point but the limitation of this propose is to get image from many cameras for compute feature corresponding between image. Biswas, S., Aggarwal, G. and Chellappa, R. [14] estimate recovery shape with robust of albedo, know light source direction and Normal surface. This technique use the linear minimum mean square error(LMMSE) to estimate initial albedo then apply nonstationary mean and nonstationary variance(NMNV) field to produce a normalized image for normal surface estimation.[15] used minimise the difference between intensity of input image and shading with the

weighting coefficients, the second term is to apply smoothness constraints to the reconstructed shape.

2. Shape propagation approaches

   Shape propagation approaches propagate the shape information from a set of surface points which are khown of solution to the whole image. Firstly, Horn [1] proposed the characteristic strip method which is the first propagation method. Rouy and Tourin [16] proposed a solution of using Hamilton-Jacobi-Bellman equations and viscosity solutions theories to retrieve a unique solution. Oliensis and Dupuis [17] propose a optimal control related algorithm which reconstructs the surface of input image when singular points that the surface is locally concave have been identified, and their heights are known. After that Bichsel and Pentland [18] proposed a simple form of this algorithm and the improvement in speed was presented by Sethian [19] with using the Fast Marching method. Kimmel and Sethian [20] modified the Fast Marching algorithm to be an optimal algorithm for SFS. Tankus et al. [21] extended the Fast Marching method of SFS with perspective projection. Baek, S.-Y. and Kim, B.-Y [22] apply ASM Model for feature extraction and create landmark 114 points distribution model to estimate depth of facial face image from database of human facial face model for three-dimension reconstruction. Itsik Horovitz and Nahum Kiryati [23] surface point three dimension acquired from laser point from Weighted Least Square(WLS) and compute depth from gradient and add interpolation surface. Ohtake et al. [24] introduced a surface representation that shares some common properties. To recover a surface from a dense set of points, they locally fit quadratic patches. The main drawback of this group is some points of the surface is need to be known before the surface has to be reconstructed.

3. Local approaches

   Local approaches reconstruct shape with assumption that is the local simple shapes such as plane, sphere, or paraboloid. These approaches use the information from intensity derivative and assume the spherical surface. Lee and Rosenfeld [25] proposed the local surface regions approximation by spherical patches which they calculated the slant and tilt of the surface at light source coordinate and then thansformed them back to the viewer coordinate. Zhongquan Wu and Lingxiao Li[26] propose height map recovery from

surface normals present a line - integration between relative depths at initial point to every point(x,y) from a single image then the actual shape of the object can be described using its depth or its height by $p = \frac{\partial z}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x, y) - f(x, y))}{\Delta x}$ and $q = \frac{\partial z}{\partial y} = \lim_{\Delta y \to 0} \frac{f(x, y+\Delta y) - f(x, y))}{\Delta y}$

So the assumption of the surface approximating by local simple shapes is very limitting, using these approaches for general objects gives inaccurate results.

4. Linear approaches

   The image irradiance equation is a nonlinear equation because of the nonlinearity of the reflectance map function. So, this linear approaches is used to convert nonlinear to linear problem with the linearization of the reflectance map. Pentland [27] uses Taylor series expansion and ignored the high order terms to take the reflectance map as a function of the surface gradient. And calculate the surface depth by using Fourier transform and inverse Fourier transform. But there is a problem if nonlinear terms are large, then Tsai and Shah [28] used the same method that is Taylor expansion to linearize the reflectance map in depth z(x, y) without using the surface gradient. Instead of using surface gradient, they used the discrete approximations with the assumption that the surface will be darker when it is far away from the camera and calculated the depth by using Jacobi iterative scheme.[29] improve the depth estimation accuracy by using 22 feature point to estimation depth base on ICA model for reconstruction the three dimension of face from two dimension image with method rotate facial feature points and use ICA algorithm under the linear mixture the non-gaussian process to recovery depth. Antonio using linear heat-equation smoothing process to recovery the surface normal and the second observation the problem of surface height recovering that embedding surface normals on a maniflod in three-dimension space.

5. Belief propagation approaches

   Belief propagation approaches is a method that use a factor graph to calculate the most probable solution. Petrovic et al. [30] uses belief propagation in a gridlike factor graph with nodes that enforce the integrability. Potetz [31] describes techniques to improve the efficiency of belief propagation by constraint on estimates of surface gradients. The main drawback of this group is the need of input surface gradients , then this algorithm is not a complete SFS algorithm because it needs inputs from the result of another SFS algorithm.

Besides of these 4 groups, there are other methods of three-dimension shape reconstruction, such as reconstruction of three-dimension shape objects from multiple images. Woodham [32] uses multiple images at the same camera position under different lighting conditions to avoid the ambiguity inherent in a single measurement of image brightness. In case of Lambertian surface, only three images are needed to determine the surface orientation and the albedo. Onn and Bruckstein [33] reconstruct the surface orientation which is Lambertian surfaces from only two images by exploiting the integrability constraint. Fan and Wolff [34] presented a method for reconstructing surface curvature and shape from multiple unknown light directions. The method works for certain types of reflectance maps including Lambertian. Hailin Jin and etc. [35] recovery shape by assume same constant albedo to background and foreground with multiple image view and minimizze the gradient direction. Wu and Tang [36] proposed a method for dense photometric stereo, that images from many light directions are recorded where the lighting direction is estimated using a mirror sphere. While the solution is solved by graph cut method, the main drawback is that many ( > 1000) images are evaluated to find the solution. Another method of three-dimension reconstruction is depth order estimation as presented by Natchamol S.[37], that propose the depth order estimation base on intensity and shadow using Hillerest-Valley classification and otsu.

### 1.1.3 Statement of Propose

This aim of this dissertation is to recovery height map from single two-dimensional face image. Even if there are many researches in this area have been developed continuously, there are still difficulties in human face shape recovery more than other shape type. Human face height map recovery is a special case of depth recovery because there is very little different of height map in human face. Then the challenge is how to separate height map degree of human face. This height map recovery is an interested research that can recover face shape from single straight face image. In this research, images that were used are grayscale that have less information than RGB image. The main theory in this research is Lambertian law of reflection with main objective to create 3D shape by height map recovery.

The statement of propose of this dissertation can be categorized into three issues:

1. To solve surfaces Normal vector direction problem.

2. To verify accuracy of reconstructed shape.

3. To estimate three-dimensional facial shape by using only single two-dimensional input face image.

### 1.1.4 Scope and Proposed Solutions

**1. Scope**

(a) This dissertation is a three-dimensional face reconstruction from a single two-dimensional image.

(b) Input images, image from human face must be a frontal face when the direction of light source is the same as the direction of a camera.

(c) The image size 256 x 256 pixel.

(d) The image consist of a human face only one.

(e) The type of image is grayscale.

(f) Scaling of average face model and ground truth model to fit input face image was done with simple scaling of range in x and y axis. But in z axis scaling was done with average range of scaled x and y axis to retain true proportion of face shape as an original model.

**2. Proposed Solutions**

Shape from shade is used in this research to reconstruct three-dimensional face from a two-dimensional image. Our focus is on solving problems with good quality and high speed of 2 important processes.

(a) Face albedo estimation: This process is used to produce a normalized image of two-dimensional human face image and enter a normalized image into the Normal surface estimation process.

(b) Normal surface estimation: This process is used to find normal surface which is close to ground truth's normal surface and processes in high speed. By using additional equations of normal vector in x-axis and normal vector in y-axis, the problem can be solved without using optimization method.

# CHAPTER II

# BACKGROUND AND RELATED WORKS

This research is estimating the shape of face from shading. I recover the 3D shape of a human face using a single image and focus on recovering on height map of face. I assume the surface of object exhibits lambertian reflecance. In this chapter we will give a brief related propose many of information to extract what actually looking for generating height maps.

## 2.1 Projection

The projection of a surface point of the three-dimensional scene onto the two-dimensional image plane can be described by parallel projection, a perspective projection, or an orthographic projection.

### 2.1.1 Parallel Projection

Parallel projection is the method of drawing parallel lines of an point from the object to the plane of incidence (view plane), regardless of the distance, the rendering is done by parallel projection which maintains the ratio of an object without considering depth. That is to draw a line, which parallels to the z-axis, to appear on the screen (no depth of an object) as show in Figure 2.1.

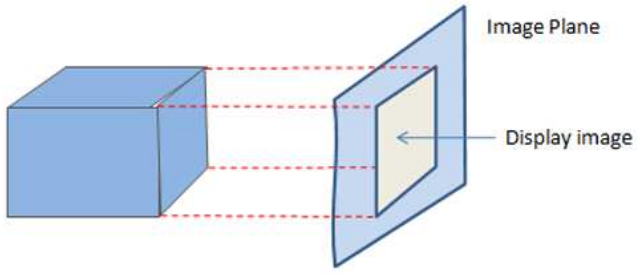



Figure 2.1 : Parallel Projection

### 2.1.2 Perspective Projection

Perspective projection describes how to display an image from taking a picture of

three-dimensional object.

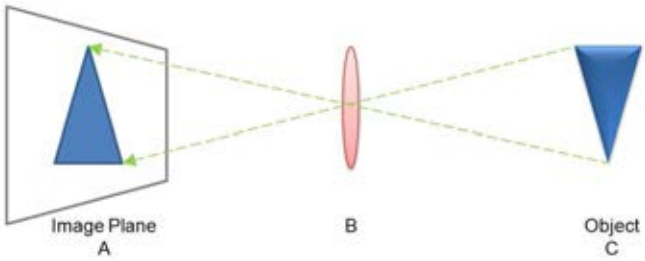



Figure 2.2: A camera produces an image that is a perspective projection

As shown in Figure 2.2 when the light reflects from an object surface onto an image plane and three-dimensional object is projected onto two-dimensional image, it is called perspective projection.

The technique proposed by Milan Sonka, Vaclav hlavac and et.al.[38], that camera use with object and images screen, is parallel projections. Perspective view can explain about two-dimensional displayed image that was projected from a three-dimensional object and displayed image's size is depends on the distance between an objects and an image plane position as shown in Figure 2.3.

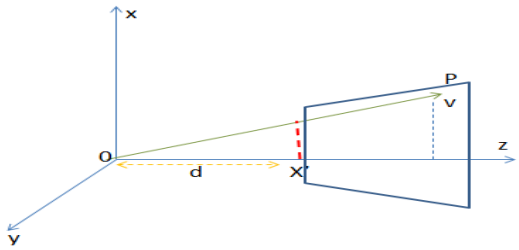
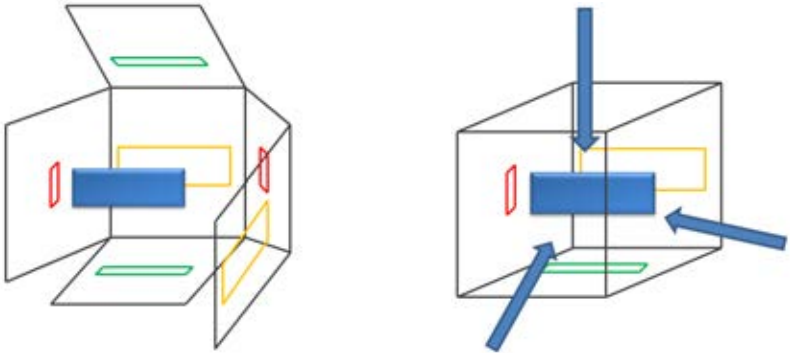



Figure 2.3: Perspective view

### 2.1.3 Orthographic Projection

A three-dimensional object is projected perpendicularly onto image planes with parallel projectors as shown in Figure 2.4



(a) a projection with 6 parallel projectors (b) a projection with 3 parallel projectors

Figure 2.4: Orthographic Projection

Figure2.4(a) display a projection with 6 parallel projectors and Figure 2.4(b) display a projection with 3 parallel projectors

Beside these techniques of three-dimensional to two-dimensional projection, there are special equipments that can get three-dimensional data directly from an object such as a laser scanner.

## 2.2 Laser Scanners and Laser Scanning Data

Laser scanning system [39] is a popular technique used to collect data of object and environment which can be reconstructed to three-dimensional model. The data, sometimes called 2.5D surface model, contain a lot of random-distributed details such that model can reconstruct some sides of surface. The quality of laser scanning data depends on laser scanner and three-dimensional application. A the laser scanner can operate in different style upon the user requirement such as "point density". The point density is a density of data point that depends on object's depth, system platform, field of view and sampling frequency. The point density should be adapted to the reconstruction application.

## 2.3 Lambertian Reflectance Models

An image is the result of interaction between surface materials and light sources. Michael Oren and Shree K. Nayar. [40] Surface model can approximation the reflection by lambertion. Having good models to describe the reflection properties of different surfaces helps in achieving a better understanding for the image formation process. Surfaces can be classified according to their

reflection properties to specular, diffuse, or hybrid surfaces. This research focuses on human face reconstruction using Lambertian reflectance model.

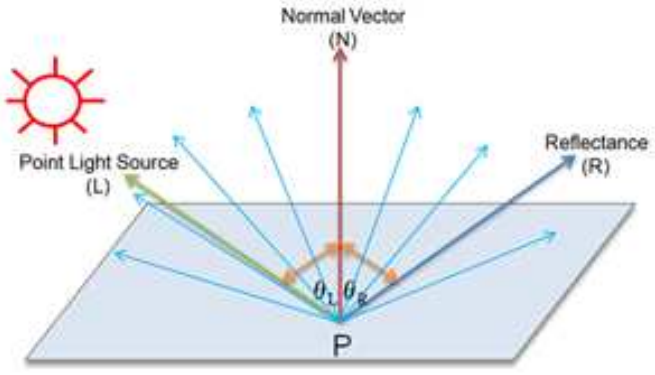
The brightness of a Lambertian surface is proportional to the energy of the incident light. The amount of light energy falling on a surface element is proportional to the area of the surface element from the light source position.

Lambertian surface reflection of light occurs whenever light incidents on a surface which can be classified into two incident :

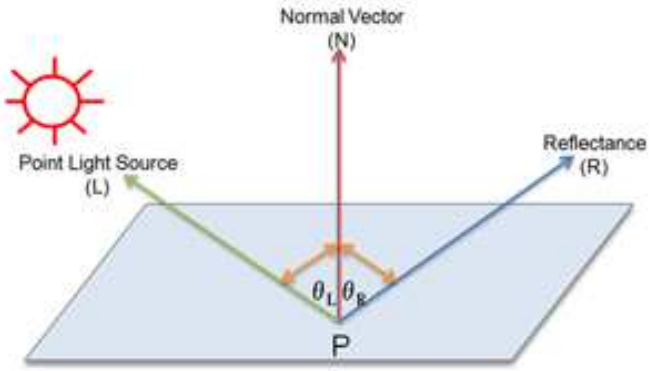
The first type is when the light reflects in all directions from a surface is called diffuse reflection or Lambertion reflection. The phenomenon of diffuse reflection is shown in Figure 2.5(a). Let be $L$ be a point light source, $N$ be a normal vector of the surface on point $P$, $\rho$ be an angle between light source direction and the normal vector of the surface when $L$ illuminates straightly to the surface, the diffuse reflection reflects incoming light equally in all directions and angle of light reflected in all directions can be defined with the same angle for all positions of an observer. The foreshortened area is a cosine function of an angle between the surface orientation and light source direction.

The second type is the incident of light reflected on surface when the reflection direction are equal to the direction of light source, is called specular reflection. The phenomenon of specular reflection is illustrated in Fig 2.5(b) using mirror-like reflection. The point of light source direction and the reflection direction are in the same angle.

Therefore, the Lambertian surface can be modeled as show in Figure 2.5:




(a) diffuse reflection




(b) specular reflectance

Figure 2.5: Geometry of reflectance

In this research surface reflections are categorized as Lambertian and non-Lambertian. The Lambertian model is when the light falls on to a surface and a surface diffuses light equally in all directions therefore the brightness of a surface point can be expressed as a scalar product of a light direction vector and a surface normal vector at a point, multiplied by the albedo. The non-Lambertian model considers the specular component in addition to diffuse component in the Lambertian model.

In this research, I have focused on diffuse reflection for reconstruction 3D human face from a single image.

## 2.4 Light Source Direction Vector

Light is an important component of an image that an image must have at least one or more light sources. Two types of light source are point and area light sources [41].

**Area point light source** is a single position of light source (small light bulb) or light source is at infinity e.g. the sun. Moreover, a point light source's image has intensity which cover from various light source directions.

**Area light source** is not a point of light source but an area of light source such as fluorescent lamp. However, in this researcher assume that the light source is a point light source.

### 2.4.1 Directional Lights

Light source distance can be assumed as infinity; therefore a single direction vector can be

used for every point in the scene. From this assumption, the light source direction can be calculated because the direction of light source is the same for every point in the scene. From Figure. 2.5 let's denote that:

- $L$ is a point light source vector.
- $R$ is reflectance vector.
- $N$ is normal vector.
- $P$ is point of incidence.
- $\Theta_L$ is the angle between the light source direction and the surface normal.
- $\Theta_R$ is the angle between the viewing direction and the surface normal.

The angle between the light source direction and the surface normal is in the range $[-90^{\circ}, 90^{\circ}]$. Then this value is set to the cosine of the angle between the light source direction and the surface normal (that is greater than 0). Our directional light function assumes that the vectors of interest are normalized, thus the dot product between two vectors results as the cosine of the angle between them.

## 2.5 Albedo of Human Face

Albedo is a fraction of light from the light source which surface can reflect as described in the reflectance rule and light which is not reflected is absorbed by the surface. Many surfaces in nature are composed of elements that have their own reflective properties, such as eyebrow and hair in human face. Actually, reflectance for such surfaces can be derived by radiative transfer calculations [42]. The reflected radiances adopt an angular pattern which is typical light that has suffered multiple scatterings. If the elements are not strongly absorbing, then light reflected by one surface element may be reflected by another. On the other hand, if a surface element is strongly absorbed, then the reflection approximates single scattering, that the angular distribution of the reflected light can be described in terms of a single angle.

Image's intensity can be restored by using the difference between observed angles to estimate the albedo map. From a Lambertian reflection model, if intensity light source vector and normal vectors are all known, we can estimate the albedo at each pixel by using the formula:

$$\rho(x, y) = I(x, y)/L * N(x, y) \tag{2.1}$$

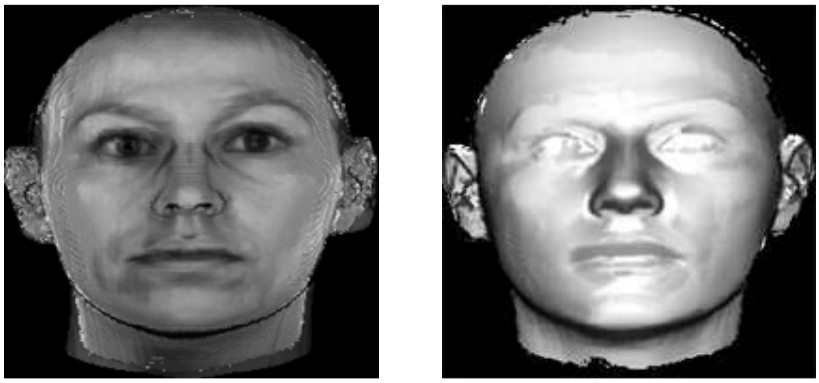
where

$(x, y)$ is a pixel coordinate (x=1,...end of width; y=1,...end of column),

$\rho(x, y)$ is the surface albedo $(0 \leq \rho \leq 1)$,

$I(x, y)$ is the intensity of each pixel and $L.N$ is its illumination.

For gray scale shading image, the value of each pixel's albedo Figure. 2.6(a) can be estimated that allows albedo map to be estimated as close to real albedo of an image under various of illumination. The normalized face image Figure. 2.6(b) is after derived albedo face from input face image.



(a) Derived albedo face (b) normalize face image

Figure 2.6: Example albedo map and normalize face image

## 2.6 Normal Surface

From Lambert model a normal surface of any object can be reconstructed from normal vector of points on a surface. Normal vector gives a line that is perpendicular to the curve at any point, $P$ as shown in Figure. 2.7

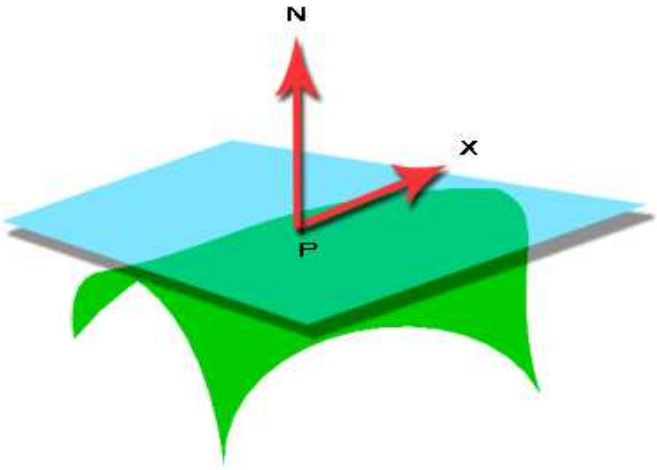



Figure 2.7: Normal vector

Normally, an image shape model consists of a pixels set which can be represented in x,y,z coordinate. If this points set is projected onto the x-y plane, there will be a z-value (and normal vector) of each $x, y$ coordinates. Then a point, $P$ at coordinates( $x, y$ )on surface will be represented by the depth(x,y) and its normal vector( $x, y$ ). For each point on the surface, normal vector that perpendicular to the surface can be calculated.

The vector $\vec{n} = (N_x, N_y, N_z)$ is a normal vector perpendicular to the plane.

If $P(x_0, y_0, z_0)$ is a point on the plane, then vector $\vec{P}P = (x - x_0, y - y_0, z - z_0)$ is perpendicular to vector $\vec{n}$ and a normal vector can be calculated from $\vec{P}P.\vec{n} = 0$

But if the light source vector, intensity and albedo are known the normal vector can be calculated from: $\vec{n} = I/(p * L)$ and the unit normal vector at every pixel (x, y) is $N = \frac{N}{|N|}$

## 2.7 Relative Height

The height map cannot be estimated using only a surface normal because it lacks of necessary data in order to extract the exact height information. Thus it needs an additional iterative estimation technique.

Starting from the known-height pixel, the algorithm finds the heights corresponding to other pixels. At each iteration, the algorithm updates the heights of four neighbors (north, east, west and south) for any reference pixel. The updating is done progressively starting with the nearest neighbors and continuing with their corresponding neighbors of unknown heights until the entire image.

Considering the projection of the surface normal in the direction of one of its neighbors, components of the surface normal vector can be denoted by $N(N_x, N_y, N_z)$ which is considered to be normalized :

$$N_x^2 + N_y^2 + N_z^2 = 1 \tag{2}$$

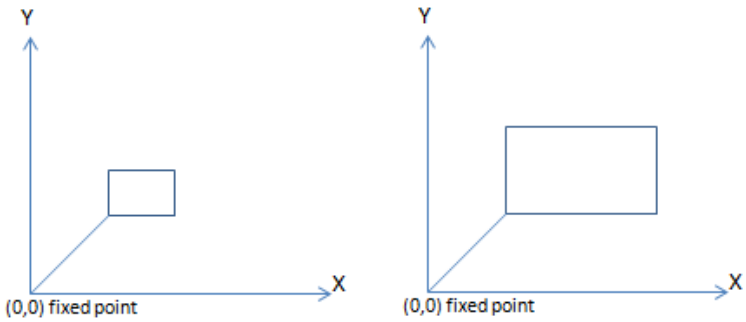
Gavin D.J. Smith and Adrian G. Bors [52] presented the height difference in x direction, $\Delta_x H$ , and in y direction, $\Delta_y H$ , between the reference and the unknown height pixel which can be calculated from :

$$\Delta_x H = k \frac{N_x}{N_z} \tag{3}$$

$$\Delta_y H = k \frac{N_y}{N_z} \tag{4}$$

The updating considers the component $N_x$, of a surface normal for the case when the pixels,two sites are located on the same row and the $N_y$ component when the two sites pixel are located on the same column. The unknown-height pixels to be considered are those distances, are equal to one k = 1 while all the normal vectors are unit vector as in section 2.2.

## 2.8 Scale




(a) image before resize (b) image after resize

Figure 2.8: model of image resize

The size of an image can be changed by changing the distance between points. A multiply value of every point that leads to longer or shorter distance is called "scaling factor". A scaling factor with value greater than one will enlarge an image, a scaling factor with value less than one will shrink an image ,and scaling factor with value equal to 1 does not affect the image's size.

When an image is shrunk or enlarged, there is a reference point of shrinking or enlarging that is called "a fixed point".

If the origin point (0,0) is a fixed point, any (x,y,z) point can be resized by multiplying with $S_x$ factor to x-axis value, $S_y$ factor to y-axis value, and $S_z$ factor to z-axis value. Then the new point $\left(x^{'}, y^{'}, z^{'}\right)$ is :

$$[x, y, z, 1] * \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & S_k \\ & & & \end{pmatrix} \begin{matrix} x' & = x * S_x \\ = y' & = y * S_y \\ z' & = z * S_z \end{matrix}$$

# CHAPTER III

# PROPOSED 3D FACE RECONSTRUCTION METHOD

There exist several 3D face reconstruction methods, however, the computation time of these methods are not the main concern thus, a new 3D face reconstruction from a singe 2D image using height map is proposed to solve such a problem. The proposed method consists of 5 steps; i.e. preprocessing method, albedo calculation normal, actual normal surface calculation, height computation, reconstruction. Since a 2D image doesn't provide information about normal surface, the proposed method uses normal surface of an average face image to estimate normal surface of each input image. The average face image and the head model of each image are provided by "Basel Face Model(BFM)" from Basel University" and "Max Planck Institute for Biological Cybernetics" as ground truth data. These databases consist of face images, average face image and head model of each face image as ground truth data. Head model data of each face image contains the following description.

1. Object Vertex data

   **The geometric vertices(v)** are represented by vx,vy,vz which specifiy the coordinates of each pixel.

   **The texture vertices(vt)** is texture coordinates, represented by vtx,vty when vtx represents row of a texture image and vty represents column of a texture image.

2. Texture image is a color face image of size 512 x 512.

The vertices conversion of database show in Figure 3.1 is flow intensity of image by the relation of geometric vertices given the coordinate of x-axis and y-axis and z-axis which relate to texture vertices with x-axis and y-axis of texture vertice and texture image.

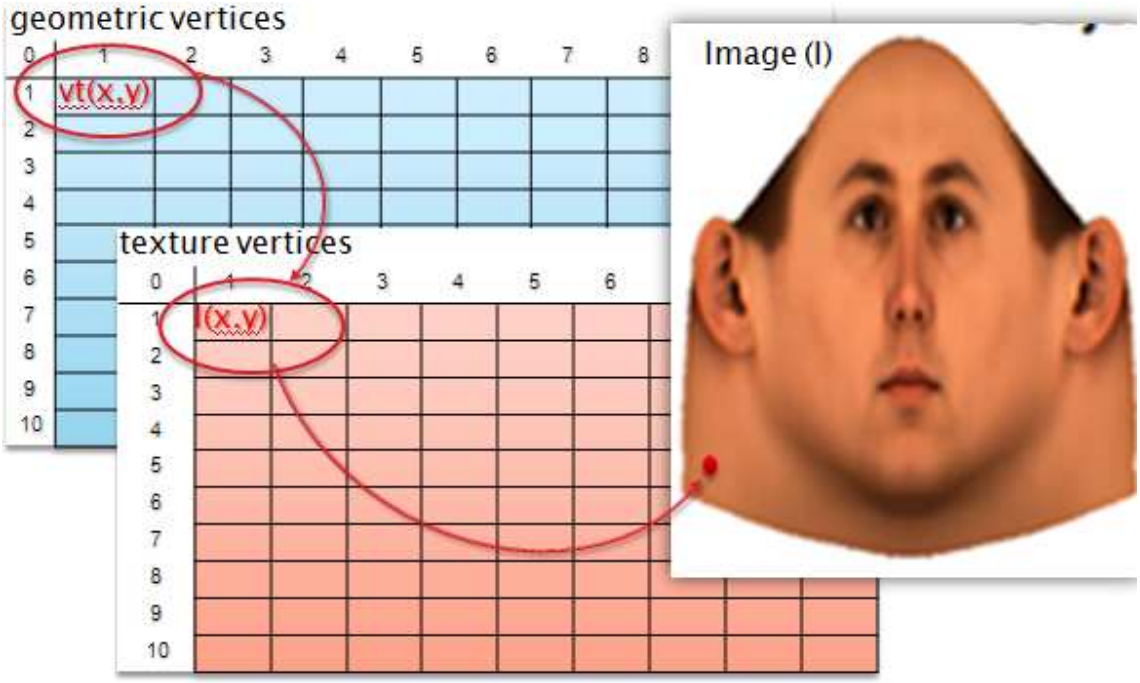



Figure 3.1: Structure of data model

There are five main processes in the proposed 3D face reconstruction method: (1) Preprocessing method, (2) Normal Surface Estimation, (3) Calculate Albedo and Normalize Input, (4) Normal Surface Calculation, and (5) Calculate Actual Height as show in Figure 3.2.

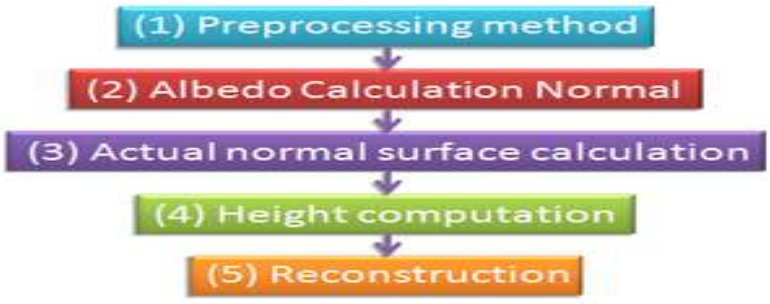
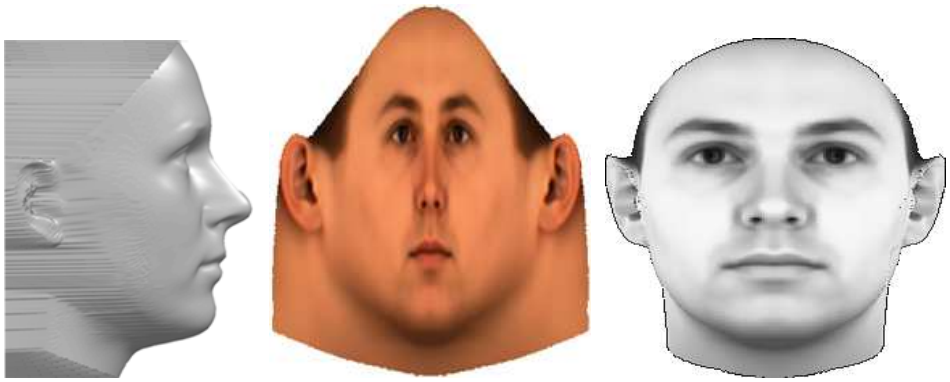



Figure 3.2: Process of the proposed method

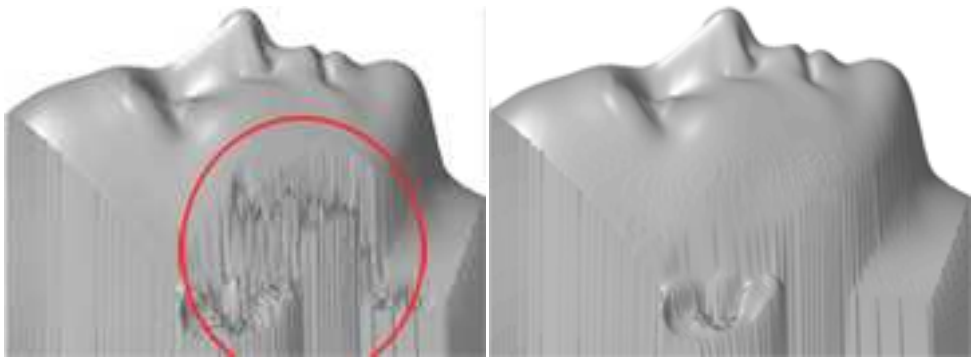
## 3.1 Preprocessing Step



(a) height image (b) texture image (c) intensity image

Figure 3.3: Transform Height of average to 2D image

Figure 3.3 describes how to construct 2D face from Basel Face database that provides texture vertices and geometric vertices scanned around the x-axis. For this research, the data of a frontal face (as shown in Figure 3.3(a)) an extracted from geometric vertices of a head and are used to make a height map with a texture image (as shown in Figure 3.3(b)) to create an intensity image (Figure 3.3(c)).



(a) height map with error] (b) height map

Figure 3.4: Comparison of height derived from different data

Figure 3.4 shows images of height map, Figure 3.4(a) shows errors that are caused by mapping of more than one height values to the same coordinates. These duplicated values are the values of the back and the front of a head at the same coordinates. Figure 3.4(b) shows the result of

discarding redundant height from each coordinates. In preprocessing step, we thus select to keep the larger height value and discard the smaller height value according to the position of frontal face from xy plane. The preprocessing steps are as follows.

### 3.1.1 Convert Height and Intensity of A Face

After extracting geometric vertices of each frontal face, the coordinates of each vertex are used to map the height of each vertex with the texture of each vertex from texture image to form an intensity image as follow

Because the coordinates of face model in face database are measured by Euclidean distance, thus the range of pixel is converted according to the range by:

1. For geometric vertices(v) used for height transform

   (a) Find x and y coordinates for each pixel.

   (b) Scale the range of x-axis using equation (3.1), y-axis using equation (3.2) and z-axis using equation (3.3) to be between 0 and 256 value.

$$\text{range of x} = \text{maximum of x} - \text{minimum of x}$$

$$\text{New geometric vertex of x} = \frac{\text{geometric vertex of x}}{\text{range of x}} \text{x256} \quad (3.1)$$

$$\text{range of y} = \text{maximum of y} - \text{minimum of y}$$

$$\text{New geometric vertex of y} = \frac{\text{geometric vertex of y}}{\text{range of y}} \text{x256} \quad (3.2)$$

$$\text{range of z} = \text{maximum of z} - \text{minimum of z}$$

$$\text{New geometric vertex of z} = \frac{\text{geometric vertex of z}}{\text{range of z}} \text{x256} \quad (3.3)$$

   (c) Remove duplicate points from the same and keep only the maximum z-value.

   (d) Transform geometric vertex to height and interpolatet the unknown geometric vertex from known coordinate values and the result is shown in Figure 3.5(a)

2. Texture vertices(vt) used for intensity transform.

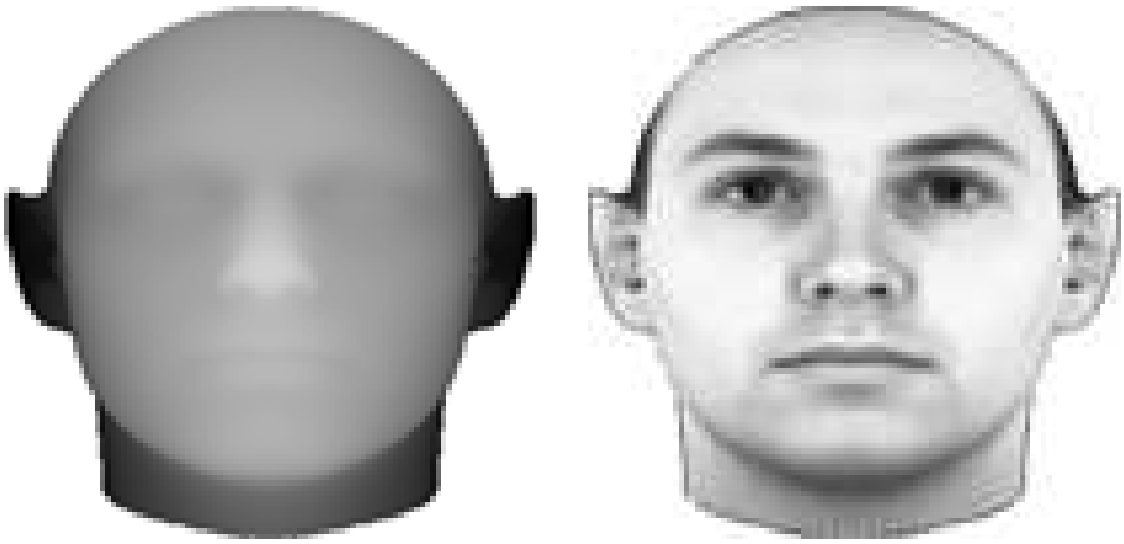
   (a) Get data size in x,y axis from texture vertices.

   (b) Change the range of x,y-axis to be between 0 and 512.

$$\text{New texture vertex of } x = \text{texture vertices of } x * 512 \quad (3.4)$$

$$\text{New texture vertex of } y = 512(512 - \text{texture vertices of } y) \quad (3.5)$$

(c) Get the intensity of geometric vertex from face image texture from a set of data geometric vertex, texture vertices that are related with reference point of a face image.



(a) Height from Face Average (b) Intensity from Face Image Texture

Figure 3.5: data after transform

### 3.1.2 Create Height Map of Face by Linear Interpolation

From previous step, the range vertex data are not the actual coordinate values because x values and y values are not integer number. Then the z values of integer x and y values are approximated by linear interpolation method which interpolates data values by assuming that the data vary linearly between original data values. For example, Figure 3.6 shows the x,z data points which have been shifted from the previous step and Figure 3.7 shows linearly interpolated values between these points.

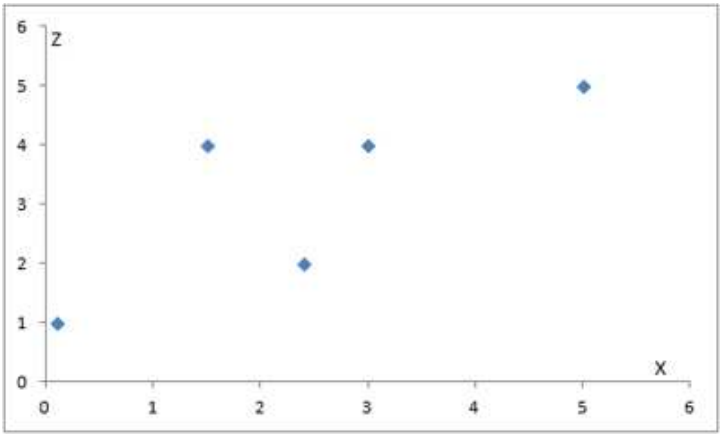



Figure 3.6: data points

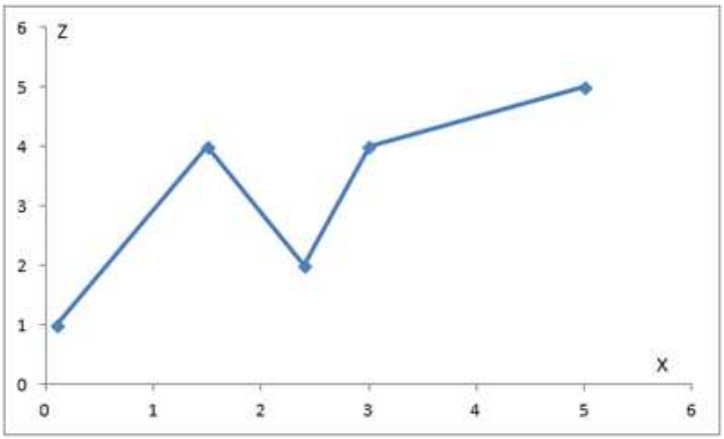



Figure 3.7: linearly interpolated values

Figure 3.8 illustrates interpolated values at integer points in x-axis. Then the complete height map can be generated.

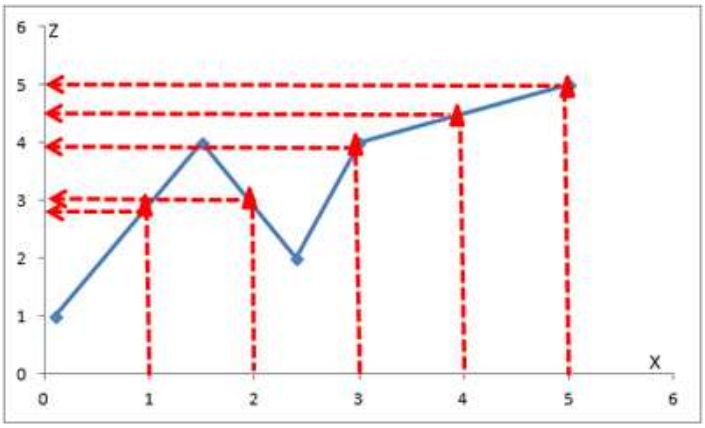



Figure 3.8: interpolated values at integer points

## 3.2 Normal Surface of Transformed Average Face Calculation

An input image is a 2D intensity map without any other data. The proposed method used height map as shown in Figure. 3.5(a) and intensity of an average face image as shown in Figure. 3.5(b). if the normal surface of an average image is transformed to match an input image, normal surface of an input image can be estimated. The average face is transformed by using four reference points of an average face image and an input image as the following steps:

1. Assign four reference points to the same face positions of an average face image and an input image. Four reference point are 1.Right eye, 2.Left eye, 3. Top head, 4.Chin.
2. Transform height map (AVh) of an average face image to match an input image by aligning four reference points of an average face height map to be at the same(x,y) coordinates.
3. Transform intensity map (avI) of an average face image to match an input image by aligning four reference points of an average face and an input image at the same(x,y) coordinates.
4. Calculate normal surface from height map (Figure. 3.9(d)) of a transformed image (Figure. 3.9(a)) as shown in Figure. 3.9(e).

(a) input face image

(b) Intensity from face image texture

(c) intensity of face average and size of input image

(d) height map from face average

(e)height map of face average and size of input image

Figure 3.9: Intensity and height map transform

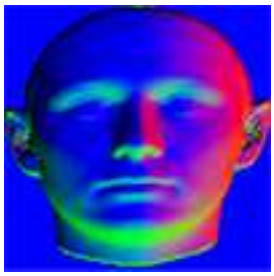
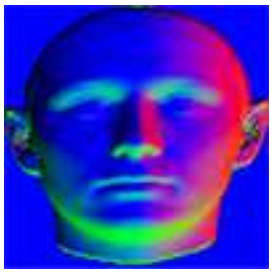


Figure 3.10: normal surface from height map

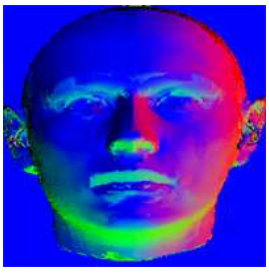
## 3.3 Normal Surface of Input Image Estimation

In this step, the albedo of an input face image is calculated but the only known value is the average face normal surface. If ' $N_{avg}$ ' is an average face's normal surface, then albedo of an input face image, $\rho_i$, can be computed from $\rho_i = I/N_i L$ when $N_i$ can be estimated from $N_{avg}$.



(a) Normal surface of an average face image

(b) Input face image



(c) Estimated normal surface of input face image

Figure 3.11: process of normal surface estimation

### Data preparation

The first step in morphing process is data preparation

1. Adjust histogram of an average face image(Figure. 3.12(a)) to be equal to an input face image(Figure. 3.12(c)).
2. Add light from ' $L$ ' direction (light source direction of an input image) to texture on average face's height map and get new intensity map (Figure. 3.12(b)).

(a) (b) (c)

Figure 3.12: (a) an intensity of an average face image, (b) an average face image add light and histogram from input face image, (c) an input face image.

Then perform morphological operations to transform normal surface of average face to normal surface of input face by the best matched pattern of normal vector in z-axis.

### 3.3.1 Pattern Method to Estimate Normal Surface

Inputs of the process are an average face image and an input face image.

**concept**

Match z-axis normal vector pattern of an input face pixel with an average face pixel which is nearby and get normal vector from height map average face pixel to input face pixel using a 3x3 pixel pattern size. The estimation process of input image's normal surface is as the following steps:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 88 | 60 | 43 | 87 | 59 | 95 | 50 | 68 | 50 | 71 | 45 | 32 | |
| 2 | 99 | 92 | 39 | 53 | 35 | 47 | 50 | 63 | 78 | 57 | 72 | 94 | |
| 3 | 29 | 69 | 80 | 23 | 72 | 59 | 35 | 29 | 45 | 90 | 92 | 22 | |
| 4 | 76 | 61 | 94 | 52 | 74 | 83 | 40 | 82 | 52 | 51 | 71 | 63 | |
| 5 | 36 | 44 | 52 | 86 | 79 | 59 | 38 | 39 | 58 | 98 | 92 | 89 | |
| 6 | 89 | 83 | 65 | 71 | 30 | 82 | 96 | 59 | 57 | 87 | 32 | 28 | |
| 7 | 73 | 57 | 81 | 87 | 62 | 48 | 96 | 40 | 34 | 59 | 29 | 88 | |
| 8 | 41 | 34 | 77 | 55 | 61 | 82 | 80 | 27 | 50 | 68 | 97 | 68 | |
| 9 | 23 | 33 | 86 | 50 | 26 | 21 | 70 | 70 | 63 | 41 | 34 | 30 | |
| 10 | 95 | 98 | 26 | 38 | 76 | 48 | 35 | 58 | 46 | 79 | 97 | 27 | |
| 11 | 56 | 51 | 55 | 99 | 36 | 97 | 41 | 88 | 75 | 41 | 89 | 65 | |
| 12 | 97 | 55 | 96 | 60 | 62 | 82 | 33 | 24 | 34 | 39 | 35 | 48 | |
| X | | | | | | | | | | | | | |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 27 | 38 | 39 | 25 | 49 | 20 | 45 | 1 | 36 | 30 | 36 | 45 | |
| 2 | 11 | 21 | 48 | 42 | 1 | 24 | 8 | 26 | 47 | 27 | 27 | 29 | |
| 3 | 39 | 40 | 44 | 14 | 47 | 28 | 40 | 28 | 19 | 2 | 31 | 30 | |
| 4 | 37 | 18 | 34 | 36 | 1 | 27 | 4 | 42 | 28 | 19 | 48 | 32 | |
| 5 | 49 | 46 | 15 | 31 | 23 | 2 | 8 | 20 | 41 | 44 | 36 | 16 | |
| 6 | 32 | 10 | 35 | 7 | 29 | 14 | 20 | 13 | 20 | 50 | 37 | 40 | |
| 7 | 10 | 18 | 38 | 19 | 36 | 49 | 46 | 22 | 21 | 48 | 34 | 36 | |
| 8 | 23 | 43 | 3 | 19 | 37 | 30 | 15 | 15 | 23 | 43 | 1 | 34 | |
| 9 | 33 | 15 | 18 | 13 | 23 | 20 | 16 | 5 | 20 | 8 | 6 | 35 | |
| 10 | 2 | 23 | 10 | 3 | 49 | 20 | 8 | 30 | 22 | 25 | 9 | 46 | |
| 11 | 35 | 29 | 24 | 33 | 17 | 5 | 13 | 22 | 7 | 14 | 50 | 7 | |
| 12 | 6 | 43 | 47 | 30 | 36 | 49 | 3 | 13 | 50 | 27 | 26 | 1 | |
| X | | | | | | | | | | | | | |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 23 | 75 | 23 | 47 | 94 | 26 | 93 | 69 | 24 | 32 | 54 | 32 | |
| 2 | 69 | 84 | 60 | 61 | 63 | 44 | 23 | 41 | 54 | 30 | 30 | 38 | |
| 3 | 50 | 81 | 99 | 39 | 33 | 69 | 55 | 51 | 82 | 24 | 72 | 75 | |
| 4 | 55 | 61 | 45 | 90 | 22 | 77 | 85 | 62 | 97 | 46 | 56 | 47 | |
| 5 | 97 | 68 | 44 | 41 | 68 | 99 | 72 | 39 | 47 | 21 | 89 | 33 | |
| 6 | 25 | 24 | 44 | 63 | 52 | 29 | 88 | 33 | 43 | 76 | 80 | 51 | |
| 7 | 80 | 22 | 65 | 69 | 39 | 61 | 58 | 74 | 70 | 98 | 56 | 80 | |
| 8 | 63 | 57 | 40 | 64 | 61 | 85 | 31 | 80 | 84 | 22 | 83 | 84 | |
| 9 | 27 | 38 | 27 | 25 | 47 | 35 | 37 | 89 | 84 | 61 | 69 | 90 | |
| 10 | 68 | 78 | 44 | 77 | 90 | 72 | 86 | 95 | 24 | 88 | 25 | 63 | |
| 11 | 59 | 95 | 94 | 27 | 81 | 23 | 77 | 52 | 99 | 46 | 69 | 64 | |
| 12 | 31 | 83 | 78 | 45 | 43 | 82 | 26 | 96 | 43 | 93 | 37 | 73 | |
| X | | | | | | | | | | | | | |

(a)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 48 | 23 | 21 | 25 | 9 | 21 | 46 | 42 | 35 | 23 | 39 | 27 | |
| 2 | 18 | 41 | 47 | 31 | 11 | 43 | 33 | 11 | 28 | 39 | 19 | 10 | |
| 3 | 48 | 5 | 46 | 26 | 49 | 17 | 33 | 50 | 40 | 39 | 28 | 8 | |
| 4 | 42 | 20 | 36 | 2 | 9 | 2 | 1 | 44 | 1 | 9 | 40 | 13 | |
| 5 | 5 | 44 | 9 | 29 | 26 | 50 | 43 | 16 | 8 | 47 | 41 | 12 | |
| 6 | 14 | 49 | 7 | 13 | 15 | 17 | 8 | 1 | 43 | 46 | 8 | 19 | |
| 7 | 7 | 7 | 30 | 24 | 30 | 3 | 9 | 50 | 38 | 13 | 25 | 43 | |
| 8 | 36 | 19 | 24 | 11 | 4 | 32 | 31 | 21 | 1 | 7 | 27 | 6 | |
| 9 | 9 | 21 | 2 | 37 | 20 | 4 | 18 | 20 | 20 | 23 | 24 | 35 | |
| 10 | 18 | 6 | 6 | 28 | 50 | 21 | 16 | 8 | 39 | 32 | 7 | 5 | |
| 11 | 29 | 42 | 14 | 44 | 8 | 2 | 30 | 13 | 17 | 18 | 29 | 47 | |
| 12 | 41 | 25 | 38 | 46 | 11 | 24 | 19 | 26 | 3 | 35 | 19 | 46 | |
| X | | | | | | | | | | | | | |

(b)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.3 | 2.3 | 1.5 | 1.5 | 2.6 | 1.3 | 2.9 | 2.0 | 1.5 | 1.5 | 2.2 | 2.0 | |
| 2 | 1.7 | 1.9 | 2.5 | 2.2 | 2.8 | 1.9 | 1.5 | 1.7 | 1.7 | 1.5 | 1.4 | 1.4 | |
| 3 | 2.7 | 2.2 | 2.2 | 2.7 | 1.5 | 2.2 | 2.6 | 2.8 | 2.8 | 1.3 | 1.8 | 4.4 | |
| 4 | 1.7 | 2.0 | 1.5 | 2.7 | 1.3 | 1.9 | 3.1 | 1.8 | 2.9 | 1.9 | 1.8 | 1.7 | |
| 5 | 3.7 | 2.5 | 1.8 | 1.5 | 1.9 | 2.7 | 2.9 | 2.0 | 1.8 | 1.2 | 2.0 | 1.4 | |
| 6 | 1.3 | 1.3 | 1.7 | 1.9 | 2.7 | 1.4 | 1.9 | 1.6 | 1.8 | 1.9 | 3.5 | 2.8 | |
| 7 | 2.1 | 1.4 | 1.8 | 1.8 | 1.6 | 2.3 | 1.6 | 2.9 | 3.1 | 2.7 | 2.9 | 1.9 | |
| 8 | 2.5 | 2.7 | 1.5 | 2.2 | 2.0 | 2.0 | 1.4 | 4.0 | 2.7 | 1.3 | 1.9 | 2.2 | |
| 9 | 2.2 | 2.2 | 1.3 | 1.5 | 2.8 | 2.7 | 1.5 | 2.3 | 2.3 | 2.5 | 3.0 | 4.0 | |
| 10 | 1.7 | 1.8 | 2.7 | 3.0 | 2.2 | 2.5 | 3.5 | 2.6 | 1.5 | 2.1 | 1.3 | 3.3 | |
| 11 | 2.1 | 2.9 | 2.7 | 1.3 | 3.3 | 1.2 | 2.9 | 1.6 | 2.3 | 2.1 | 1.8 | 2.0 | |
| 12 | 1.3 | 2.5 | 1.8 | 1.8 | 1.7 | 2.0 | 1.8 | 5.0 | 2.3 | 3.4 | 2.1 | 2.5 | |
| X | | | | | | | | | | | | | |

(c)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 61 | 52 | 32 | 39 | 23 | 27 | 132 | 85 | 52 | 33 | 86 | 54 | |
| 2 | 31 | 78 | 119 | 67 | 31 | 83 | 48 | 18 | 47 | 60 | 27 | 14 | |
| 3 | 131 | 11 | 103 | 70 | 71 | 37 | 85 | 138 | 113 | 49 | 50 | 35 | |
| 4 | 72 | 40 | 53 | 5.5 | 12 | 3.9 | 3.1 | 77 | 2.9 | 17 | 72 | 23 | |
| 5 | 18 | 112 | 17 | 43 | 48 | 134 | 124 | 32 | 14 | 57 | 81 | 16 | |
| 6 | 18 | 63 | 12 | 25 | 41 | 23 | 15 | 1.6 | 75 | 86 | 28 | 54 | |
| 7 | 15 | 9.7 | 54 | 43 | 49 | 6.8 | 14 | 143 | 116 | 35 | 73 | 82 | |
| 8 | 91 | 51 | 36 | 24 | 8 | 65 | 43 | 83 | 2.7 | 9.3 | 50 | 13 | |
| 9 | 20 | 45 | 2.6 | 56 | 56 | 11 | 28 | 45 | 47 | 57 | 73 | 140 | |
| 10 | 31 | 11 | 16 | 85 | 109 | 53 | 55 | 21 | 59 | 68 | 8.8 | 17 | |
| 11 | 60 | 120 | 38 | 56 | 26 | 2.5 | 86 | 21 | 39 | 38 | 51 | 93 | |
| 12 | 54 | 63 | 69 | 81 | 19 | 48 | 34 | 130 | 6.8 | 118 | 39 | 116 | |
| X | | | | | | | | | | | | | |

(d)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 61 | 52 | 32 | 39 | 23 | 27 | 132 | 85 | 52 | 33 | 86 | 54 | |
| 2 | 31 | 78 | 119 | 67 | 31 | 83 | 48 | 18 | 47 | 60 | 27 | 14 | |
| 3 | 131 | 11 | 103 | 70 | 71 | 37 | 85 | 138 | 113 | 49 | 50 | 35 | |
| 4 | 72 | 40 | 53 | 5.5 | 12 | 3.9 | 3.1 | 77 | 2.9 | 17 | 72 | 23 | |
| 5 | 18 | 112 | 17 | 43 | 48 | 134 | 124 | 32 | 14 | 57 | 81 | 16 | |
| 6 | 18 | 63 | 12 | 25 | 41 | 23 | 15 | 1.6 | 75 | 86 | 28 | 54 | |
| 7 | 15 | 9.7 | 54 | 43 | 49 | 6.8 | 14 | 143 | 116 | 35 | 73 | 82 | |
| 8 | 91 | 51 | 36 | 24 | 8 | 65 | 43 | 83 | 2.7 | 9.3 | 50 | 13 | |
| 9 | 20 | 45 | 2.6 | 56 | 56 | 11 | 28 | 45 | 47 | 57 | 73 | 140 | |
| 10 | 31 | 11 | 16 | 85 | 109 | 53 | 55 | 21 | 59 | 68 | 8.8 | 17 | |
| 11 | 60 | 120 | 38 | 56 | 26 | 2.5 | 86 | 21 | 39 | 38 | 51 | 93 | |
| 12 | 54 | 63 | 69 | 81 | 19 | 48 | 34 | 130 | 6.8 | 118 | 39 | 116 | |
| X | | | | | | | | | | | | | |

(e)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 48 | 23 | 21 | 25 | 9 | 21 | 46 | 42 | 35 | 23 | 39 | 27 | |
| 2 | 18 | 41 | 47 | 31 | 11 | 43 | 33 | 11 | 28 | 39 | 19 | 10 | |
| 3 | 48 | 5 | 46 | 26 | 49 | 17 | 33 | 50 | 40 | 39 | 28 | 8 | |
| 4 | 42 | 20 | 36 | 2 | 9 | 2 | 1 | 44 | 1 | 9 | 40 | 13 | |
| 5 | 5 | 44 | 9 | 29 | 26 | 50 | 43 | 16 | 8 | 47 | 41 | 12 | |
| 6 | 14 | 49 | 7 | 13 | 15 | 17 | 8 | 1 | 43 | 46 | 8 | 19 | |
| 7 | 7 | 7 | 30 | 24 | 30 | 3 | 9 | 50 | 38 | 13 | 25 | 43 | |
| 8 | 36 | 19 | 24 | 11 | 4 | 32 | 31 | 21 | 1 | 7 | 27 | 6 | |
| 9 | 9 | 21 | 2 | 37 | 20 | 4 | 18 | 20 | 20 | 23 | 24 | 35 | |
| 10 | 18 | 6 | 6 | 28 | 50 | 21 | 16 | 8 | 39 | 32 | 7 | 5 | |
| 11 | 29 | 42 | 14 | 44 | 8 | 2 | 30 | 13 | 17 | 18 | 29 | 47 | |
| 12 | 41 | 25 | 38 | 46 | 11 | 24 | 19 | 26 | 3 | 35 | 19 | 46 | |
| X | | | | | | | | | | | | | |

(f)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | |
| 2 | | 159 | -265 | -265 | -232 | -262 | -296 | -318 | -307 | -290 | -232 | | |
| 3 | | -303 | -254 | -257 | -190 | -198 | -234 | -241 | -261 | -243 | -205 | | |
| 4 | | -253 | -214 | -228 | -205 | -224 | -249 | -228 | -245 | -243 | -226 | | |
| 5 | | -226 | -209 | -146 | -163 | -171 | -182 | -165 | -215 | -243 | -235 | | |
| 6 | | -13 | -477 | -448 | -439 | -463 | -493 | -534 | -569 | -559 | -486 | | |
| 7 | | -496 | -438 | -415 | -339 | -347 | -406 | -443 | -481 | -451 | -399 | | |
| 8 | | -408 | -389 | -410 | -370 | -375 | -437 | -436 | -438 | -421 | -429 | | |
| 9 | | -367 | -363 | -328 | -370 | -367 | -353 | -339 | -386 | -423 | -401 | | |
| 10 | | -160 | -677 | -657 | -653 | -632 | -625 | -715 | -759 | -768 | -706 | | |
| 11 | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | |
| X | | | | | | | | | | | | | |

(g)

(h)

(i)

Figure 3.13: Estimate normal surface

1. Input intensity of average face image(avI) as an example in Figure. 3.13(a).
2. Input height of average face image(AVh) as an example in Figure. 3.13(b)
3. Get new intensity of average face image(avI) after transforming average face image to fit input face image($I_i$) and apply histogram and light from input face image as shown in Figure. 3.13(c).
4. Get new height of average face image(AVh) after shape transform and calculation normal surface from height as shown in Figure. 3.13(d).
5. Calculate coefficient value for morphological method of average image to input image shown in Figure. 3.13(e).

$$\text{input face image after transformation} = \text{COEF} * \text{average face image} \tag{3.6}$$

where COEF is a value, that is used to multiply with intensity of an average face image, for morphing from an average face image to an input face image. This coefficient can be calculated from

$$\text{COEF} = \frac{(\text{intensity of input face ratio} + \text{intensity of average face ratio})}{\text{average face intensity}} \qquad (3.7)$$

when

- intensity of an average face ratio equals to multiplication of average face image and 1 minus an input face image's similarity proportion factor .
- input face image's similarity proportion factor is how much similarity (0 to 100%) of image that need to be morphed to an input face image.
- intensity of an input face ratio equals to multiplication of an input face image and a input face image's similarity proportion factor.

6. Calculate normal surface of input face image that equals to COEF Figure. 3.13(e) value multiply with normal surface of average face image.

**Morphological method** we estimate normal surface of an input face image from normal surface of an average face image to normal surface of an input image by choosing the minimum sum of difference between patern of average face image and pattern of input face image when patern of any image is the 3x3 neighbors' data alignment of each pixel's data.

1. Get an x coordinate and a y coordinate from an input face image pixel Figure. 3.13(g)
2. Compute the sum of the intensities of neighbor of size 3x3 pixels
3. Get an x coordinate and a y coordinate of an average face image of the same position in an input face image Figure. 3.13(h).
4. Get pattern of average face image of size 11x11 pixels Figure. 3.13(h)
5. Choose the minimum sum from pattern average face image of size 3x3 with the sum of pattern average minus the sum of pattern input image Figure. 3.13(i)
6. Get normal vector of x,y and z of an average face image at (x,y) coordinate with the minimum sum of difference value to normal vector at coordinate x and coordinate y of an input face image.

After this process, an input face image's normal surface is estimated.

### 3.3.2 Result of Each Step to The Estimate Pattern Normal

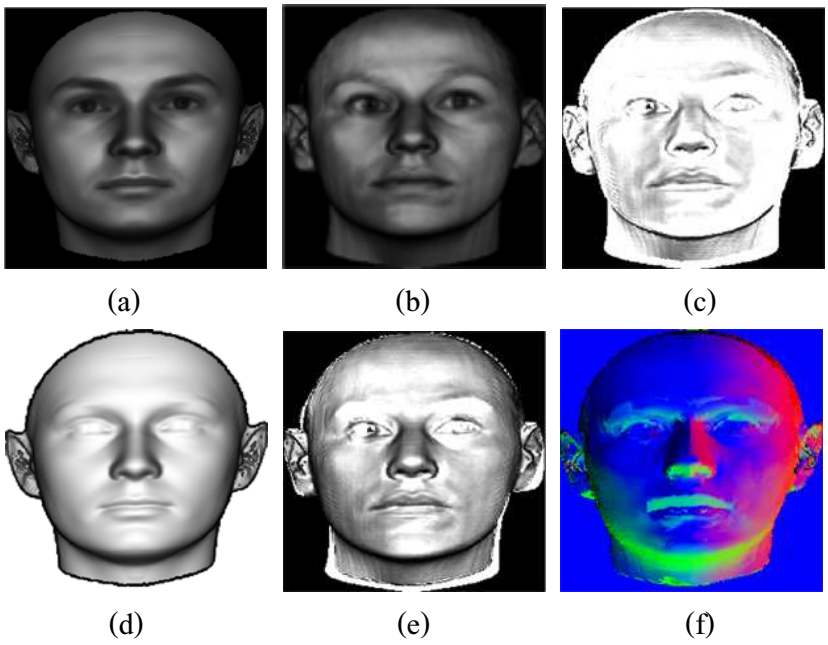



Figure 3.14: result of each step

1. Figure. 3.14(a) is a shape of average face image with light and histogram as same as input face.
2. Figure. 3.14(b) is a shape of input face image.
3. Figure. 3.14(c) is a coefficient of average face for morphing to input face image:Figure. 3.14(b).
4. Figure. 3.14(d) is a normal surface of z-axis of an average face image.
5. Normal surface of z-axis of input face image in Figure. 3.14(e) came from normal surface of z-axis of average face image in Figure. 3.14(d) multiply with Coefficient in Figure. 3.14(c).
6. Figure. 3.14(f) is an estimated normal surface of input image.

### 3.4 Albedo Calculation and Input Image Normalization

1. Enter estimated normal surface from previous step into Lambert model equation, to receive $\rho$ (albedo) of an input face image.

$$\rho = I/(L * N) \tag{3.8}$$

Figure 3.15: result from albedo calculation

2. Divide an input face image by $\rho$, to get an image with unity albedo that is called 'normalized input face'

$$\text{normalized input face} = \text{input face image} / \rho \quad (3.9)$$

Figure 3.16: result from normalization

## 3.5 Y-Ratio Calculation

**Concept**

Y-Ratio is a vlaue to be used for finding a relation between x-axis and y-axis of normal vector. The Y-Ratio relation can be used as a third equation, with unit vector equation and Lambert's equation, to solve and calculate a surface normal vector.

At each pixel, Y-Ratio is proportion between slope of normal vector in y-axis and x-axis which suppose to be calculated from estimated normal vector in albedo estimation process :

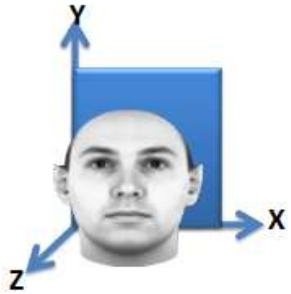
$$Yratio = n_y / (|n_x| + |n_y|)$$

**Input** an input image comes from a human face and existed an average face model, input of this process is used to estimate normal surface of an input face image.

**Process**

$Yratio = n_y / (|n_x| + |n_y|)$ that can be calculated directly from x-axis normal vector and

y-axis normal vector. Y-Ratio is calculated from normal x and normal y of each pixel as in the example below.




| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 6 | [0.2,0.4,0.4] | [0.4,0.4,0.2] | [0.1,0.6,0.3] | [0.3,0.3,0.4] |
| 5 | [0.6,0.1,0.3] | [0.3,0.3,0.4] | [0.2,0.4,0.4] | [0.7,0.2,0.1] |
| 4 | [0.2,0.3,0.5] | [0.6,0.1,0.3] | [0.4,0.3,0.3] | [0.6,0.1,0.3] |
| 3 | [0.7,0.2,0.1] | [0.7,0.2,0.1] | [0.2,0.3,0.5] | [0.2,0.4,0.4] |
| 2 | [0.4,0.4,0.2] | [0.1,0.6,0.3] | [0.7,0.2,0.1] | [0.1,0.6,0.3] |
| 1 | [0.3,0.1,0.7] | [0.2,0.4,0.4] | [0.3,0.3,0.4] | [0.2,0.3,0.5] |

Figure 3.17: input for Y-Ratio

**Example 1** from input data in Figure 3.17

Yratio of coordinate (3,2) can be calculated by

$$\text{Y - Ratio} = 0.2/(|0.7| + |0.2|) = 0.2/0.9 = 0.22$$

$$\text{And X - Ratio} + \text{Y - Ratio} = 1$$

$$\text{Then X - Ratio} = 1 - \text{Yratio} = 1 - 0.22 = 0.78$$

Thus, the relation of Y-Ratio and X-Ratio can be shown in Figure 3.18

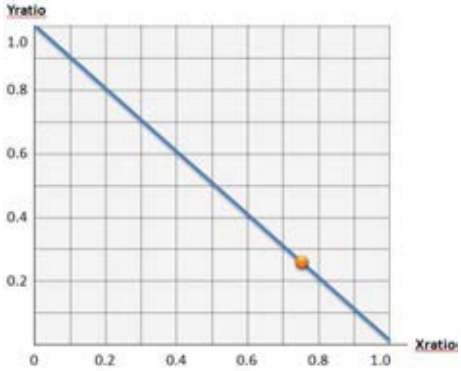



Figure 3.18: result process relation of Y-Ratio and X-Ratio

**Output** Y-Ratio equation can be used as the third equation(3.12) in set of relation equations for calculating normal vector.

$$n_x^2 + n_y^2 + n_z^2 = 1 \tag{3.10}$$

$$I = p\left(n_x * l_x + n_y * l_y + n_z * l_z\right) \tag{3.11}$$

$$Yratio = n_y / (|n_x| + |n_y|) \quad (3.12)$$

If only equation (3.10) and equation (3.11) are used , used normal vector cannot be solved except using optimization method that is very slow (eg. TaipangWu paper).

## 3.6 Actual Normal Surface Calculation

1. **concept**

   Normal vector is a unit vector that is perpendicular to the surface

2. **input**

   (a) Y-Ratio from the previous step.

   (b) An input face image is a normalized input face image.

3. **Process**

There are two areas of normal surface calculation.

(a) Background area: background pixels of an input face image that have intensities equal to 0, have normal vector =[0,0,1].

(b) Foreground area: normal surface can becalculated from an input face image of foreground and there are in 4 cases as described below.

### 3.6.1 Orthogonality of Normal Surface

The proposed methoddology mainly consists of four parts,i.e.,..

define parameter

$l_1$ = Light source in x-axis direction

$l_2$ = Light source in y-axis direction

$l_3$ = Light source in z-axis direction

k = intensity from a normalized inut face image

$y$ = Y-Ration

- $y \neq \{-1,0,1\}(A)$
  - $l_3 \neq 0(A1)$
  - $l_3 = 0(A2)$ and $l_1 + \beta l_2 \neq 0(A2.1)$

- $y = 1(B)$
  - $l_3 \neq 0(B1)$
  - $l_3 = 0(B2)$ and $l_2 \neq 0(B2.1)$

- $y = 0(C)$
  - $l_3 \neq 0(C1)$
  - $l_3 = 0(C2)$ and $l_1 \neq 0(C2.1)$

- $y = -1(D)$
  - $l_3 \neq 0(D1)$
  - $l_3 = 0(D2)$ and $l_2 \neq 0(D2.1)$

#### 3.6.1.1 Equation of Normal Surface Estimation

The proposed methodology mainly consist of four parts.

define parameter

$n_1$ = Normal vector x axis

$n_2$ = Normal vector y axis

$n_3$ = Normal vector z axis

$$n_1^2 + n_2^2 + n_3^2 = 1 \tag{3.13}$$

$$l_1 n_1 + l_2 n_2 + l_3 n_3 = k \tag{3.14}$$

$$Yratio = \begin{cases} 0; n_1 = n_2 = 0 \\ \dfrac{n_2}{|n_1| + |n_2|} otherwise \end{cases} \tag{3.15}$$

consider (3.13) (3.14) (3.15),we have

$$k = \frac{I}{\rho} \tag{3.16}$$

$$n_2 = \beta n_1 \tag{3.17}$$

From unit vector equation (3.13) and Lambert equation (3.14) there are two equation

with three unknow variables then Y-Ratio equation is the third equation for Y-Ratio equation (3.15), where $n_2 = \beta n_1$ for relationship between $n_2$ *and* $n_1$

From $y = \dfrac{n_2}{|n_1| + |n_2|}$

$$y(|n_1| + |n_2|) = n_2$$
$$y|n_1| + y|n_2| = n_2 \qquad (3.18)$$

Equation (3.18) has four solutions

**case 1** if $y > 0$ then $n_2 > 0$ and $n_1 > 0$

from (3.18) $n_2 = yn_1 + yn_2$

$$\begin{aligned} n_2 - yn_2 &= yn_1 \\ (1-y)n_2 &= yn_1 \\ n_2 &= \frac{yn_1}{(1-y)} \\ \text{Then}\beta &= \frac{y}{1-y} \end{aligned}$$

**case 2** if $y > 0$ then $n_2 > 0$ and $n_1 < 0$

from (3.18) $n_2 = y(-n_1) + yn_2$

$$\begin{aligned} n_2 - yn_2 &= y(-n_1) \\ (1-y)n_2 &= y(-n_1) \\ n_2 &= \frac{y(-n_1)}{(1-y)} \\ \text{Then}\beta &= \frac{-y}{(1-y)} \end{aligned}$$

**case 3** if $y < 0$ then $n_2 < 0$ and $n_1 > 0$

from (3.18) $n_2 = yn_1 + y(-n_2)$

$$\begin{aligned} n_2 + yn_2 &= yn_1 \\ (1+y)n_2 &= yn_1 \\ n_2 &= \frac{yn_1}{1+y} \\ \text{Then}\beta &= \frac{y}{(1+y)} \end{aligned}$$

**case 4** if $y < 0$ then $n_2 < 0$ and $n_1 < 0$

from (3.18) $n_2 = y(-n_1) + y(-n_2)$

$$\begin{aligned} n_2 &= (-y)n_1 - yn_2 \\ n_2 + yn_2 &= (-y)n_1 \\ (1+y)n_2 &= (-y)n_1 \\ n_2 &= \frac{(-y)n_1}{(1+y)} \\ \text{Then}\beta &= \frac{-y}{(1+y)} \end{aligned}$$

**Assume** $n_1 > 0$ for case 1-4 of Y-Ratio because in the current process the value of normal vector x,y and z are not known.

$$if = \begin{cases} \text{Yratio} \in (0,1) \quad \text{and} \quad n_1 > 0; \quad \Rightarrow \beta = \dfrac{y}{1-y} \\ \text{Yratio} \in (0,1) \quad \text{and} \quad n_1 < 0; \quad \Rightarrow \beta = \dfrac{-y}{1-y} \\ \text{Yratio} \in (-1,0) \quad \text{and} \quad n_1 > 0; \quad \Rightarrow \beta = \dfrac{y}{1+y} \\ \text{Yratio} \in (-1,0) \quad \text{and} \quad n_1 < 0; \quad \Rightarrow \beta = \dfrac{-y}{1+y} \\ \text{Yratio} = 1; \quad \Rightarrow \quad n_1 = 0; \quad n_2 > 0 \\ \text{Yratio} = -1; \quad \Rightarrow \quad n_1 = 0; \quad n_2 < 0 \\ \text{Yratio} = 0; \quad \Rightarrow \quad n_1 > 0; \quad n_2 = 0 \\ \text{Yratio} = 0; \quad \Rightarrow \quad n_1 < 0; \quad n_2 = 0 \end{cases}$$

$\beta$ has different values for these cases.

1. $\beta = \left(\dfrac{-y}{1-y}\right); y \in (0,1)$; then find value $n_1$ when $n_1 > 0$;
2. $\beta = \left(\dfrac{-y}{1-y}\right); y \in (0,1)$; then find value $n_1$ when $n_1 < 0$;
3. $\beta = \left(\dfrac{y}{1+y}\right); y \in (-1,0)$; then find value $n_1$ when $n_1 > 0$;
4. $\beta = \left(\dfrac{-y}{1+y}\right); y \in (-1,0)$; then find value $n_1$ when $n_1 < 0$;

Then we solve these equations:

$$n_1^2 + n_2^2 + n_3^2 = 1 \qquad (3.13)$$

$$l_1 n_1 + l_2 n_2 + l_3 n_3 = k \qquad (3.14)$$

$$n_2 = \beta n_1 \tag{3.19}$$

There Y-Ratio equation (3.15) changes to equation (3.19)

Ther are four main solution :

(A) when $y \neq \{-1,0,1\}$

(A1) case $l_3 \neq 0;$

From (3.14)

$$\begin{aligned} l_1 n_1 + l_2 n_2 + l_3 n_3 &= k \\ l_1 n_1 + \beta l_2 n_1 + l_3 n_3 &= k \\ \textit{then} \qquad n_3 &= \frac{k - (l_1 + \beta l_2) n_1}{l_3} \end{aligned}$$

From (3.13)

$$\begin{aligned} n_1^2 + n_2^2 + n_3^2 &= 1 \\ n_1^2 + (\beta n_1)^2 + \left( \frac{k - (l_1 + \beta l_2) n_1}{l_3} \right)^2 &= 1 \\ (1+\beta^2) n_1^2 + \frac{1}{l_3^2}\left[ k^2 - 2k(l_1 + \beta l_2) n_1 + (l_1 + \beta l_2)^2 n_1^2 \right] &= 1 \\ l_3^2 (1+\beta^2) n_1^2 + (l_1 + \beta l_2)^2 n_1^2 - 2k(l_1 + \beta l_2) n_1 + k^2 &= l_3^2 \\ \left[ l_3^2 (1+\beta^2) + (l_1 + \beta l_2)^2 \right] n_1^2 + \left[ -2k(l_1 + \beta l_2) \right] n_1 + \left[ k^2 - l_3^2 \right] &= 0 \end{aligned}$$

from $ax^2 + bx + c = 0$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\text{then } n_1 = \frac{2k(l_1 + \beta l_2) \pm \sqrt{\left[ -2k(l_1 + \beta l_2) \right]^2 - 4(k^2 - l_3^2)\left[ l_3^2 (1+\beta^2) + (l_1 + \beta l_2)^2 \right]}}{2\left[ l_3^2 (1+\beta^2) + (l_1 + \beta l_2)^2 \right]}$$

Divide both numeration and denorminator by 2 of the equation

$$\begin{aligned} n_1 &= \frac{\frac{2k(l_1 + \beta l_2)}{2} \pm \frac{1}{2}\sqrt{4k^2 (l_1 + \beta l_2)^2 - 4(k^2 - l_3^2)\left[ l_3^2 (1+\beta^2) + (l_1 + \beta l_2)^2 \right]}}{\left[ l_3^2 (1+\beta^2) + (l_1 + \beta l_2)^2 \right]} \\ n_1 &= \frac{k(l_1 + \beta l_2) \pm \frac{2}{2}\sqrt{k^2 (l_1 + \beta l_2)^2 - (k^2 - l_3^2)\left[ l_3^2 (1+\beta^2) + (l_1 + \beta l_2)^2 \right]}}{l_3^2 (1+\beta^2) + (l_1 + \beta l_2)^2} \end{aligned}$$

$$n_1 = \frac{k(l_1+\beta l_2) \pm \sqrt{k^2(l_1+\beta l_2)^2 - (k^2 - l_3^2)\left[l_3^2(1+\beta^2)+(l_1+\beta l_2)^2\right]}}{l_3^2(1+\beta^2)+(l_1+\beta l_2)^2}$$

$$n_2 = \beta n_1$$

$$n_3 = \frac{k-(l_1+\beta l_2)[n_1]}{l_3}$$

(A2) case $l_3 = 0$

$$l_1 n1 + l_2 n_2 = k$$

$$l_1 n1 + \beta l_2 n_1 = k$$

$$(l_1+\beta l_2) n_1 = k$$

then $n_1 = \frac{k}{(l_1+\beta l_2)}$ with $\beta = \left(\frac{y}{1-y}\right)$

and $n_2 = \beta n_1$

when(A2.1) $(l_1+\beta l_2) \neq 0$

from(3.13) $n_1^2 + n_2^2 + n_3^2 = 1$

$$n_3 = \pm\sqrt{1-n_1^2-n_2^2}$$

$$n_3 = \pm\sqrt{1-\left(\frac{k}{l_1+\beta l_2}\right)^2 - \left(\frac{\beta k}{l_1+\beta l_2}\right)^2}$$

then $n_3 = \pm\sqrt{1-(1-\beta^2)\left(\frac{k}{l_1+\beta l_2}\right)^2}$

$$n_1 = \frac{k}{(l_1+\beta l_2)}$$

$$n_2 = \beta n_1$$

$$n_3 = \pm\sqrt{1-(1+\beta^2)\left(\frac{k}{l_1+\beta l_2}\right)^2}$$

(B) case y=1;

$n_1 = 0; n_2 > 0$

from $n_1^2 + n_2^2 + n_3^2 = 1$ (3.13)

$n_2^2 + n_3^2 = 1$ (3.20)

from $l_1 n_1 + l_2 n_2 + l_3 n_3 = k$ (3.14)

$$l_2 n_2 + l_3 n_3 = k \qquad (3.21)$$

when $n_1 = 0; \text{and} n_2 > 0$

from (3.21) $l_3 n_3 = k - l_2 n_2$

$$n_3 = \frac{k - l_2 n_2}{l_3}$$

(B1) case $l_3 \neq 0$

$$n_2^2 + n_3^2 = 1 \qquad (3.20)$$

$$n_2^2 + \left(\frac{k - l_2 n_2}{l_3}\right)^2 = 1$$

$$n_2^2 + \frac{1}{l_3^2}\left(l_2^2 n_2^2 - 2kl_2 n_2 + k^2\right) = 1$$

$$l_3^2 n_2^2 + l_2^2 n_2^2 - 2kl_2 n_2 + k^2 = l_3^2$$

$$\left(l_3^2 + l_2^2\right) n_2^2 + \left(-2kl_2\right) n_2 + \left(k^2 - l_3^2\right) = 0$$

From $ax^2 + bx + c = 0$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\text{then } n_2 = \frac{-(-2kl_2) \pm \sqrt{(-2kl_2)^2 - 4(l_3^2 + l_2^2)(k^2 - l_3^2)}}{2(l_3^2 + l_2^2)}$$

$$= \frac{2kl_2 \pm \sqrt{2^2(kl_2)^2 - 2^2(l_3^2 + l_2^2)(k^2 - l_3^2)}}{2(l_3^2 + l_2^2)}$$

$$= \frac{2kl_2 \pm \sqrt{2^2(kl_2)^2 - (l_3^2 + l_2^2)(k^2 - l_3^2)}}{2(l_3^2 + l_2^2)}$$

$$= \frac{2kl_2 \pm 2\sqrt{(kl_2)^2 - (l_3^2 + l_2^2)(k^2 - l_3^2)}}{2(l_3^2 + l_2^2)}$$

$$n_2 = \frac{kl_2 \pm \sqrt{k^2 l_2^2 - \left(k^2 - l_3^2\right)\left(l_2^2 + l_3^2\right)}}{l_3^2 + l_2^2}$$

$$n_3 = \frac{k}{l_3} - \frac{l_2}{l_3}\left(n_2\right)$$

$$n_1 = 0$$

(B2) case $l_3 = 0$

From (3.21) $l_2 n_2 = k;$

$$n_2 = \frac{k}{l_2};$$

(B2.1) case $l_2 \neq 0;$

From (3.20) $n_2^2 + n_3^2 = 1;$

$$n3 = \pm\sqrt{1-n_2^2}$$

$$= \pm\sqrt{1-\left(\frac{k}{l_2}\right)^2}$$

(C) case $y = 0; n_2 = 0;$

from $n_1^2 + n_2^2 + n_3^2 = 1$ (3.13)

$$n_1^2 + n_3^2 = 1 \qquad (3.22)$$

$l_1 n_1 + l_2 n_2 + l_3 n_3 =$ k (3.14)

$$l_1 n_1 + l_3 n_3 = k \qquad (3.23)$$

when $y = 0;$ and $n_2 = 0;$

for (3.23) $l_3 n_3 = k - l_1 n_1$

$$n_3 = \frac{k - l_1 n_1}{l_3}$$

(C1) case $l_3 \neq 0$

for(3.22) $n_1^2 + n_3^2 = 1$

$$n_1^2 + \left(\frac{k - l_1 n_1}{l_3}\right)^2 = 1$$

$$l_3^2 n_1^2 + \left(k^2 - 2kl_1 n_1 + l_1^2 n_1^2\right) = l_3^2$$

$$\left(l_1^2 + l_3^2\right) n_1^2 + \left(-2kl_1\right) n_1 + \left(k^2 - l_3\right)^2 = 0$$

$$n_1 = \frac{kl_1 \pm \sqrt{k^2 l_1^2 - \left(k^2 - l_3^2\right)\left(l_2^2 + l_3^2\right)}}{l_1^2 + l_3^2}$$

$$n_3 = \frac{k}{l_3} - \frac{l_1}{l_3}\left(n_1\right)$$

$$n_2 = 0$$

(C2) case $l_3 = 0$

From (3.23) $l_1 n_1 + l_3 n_3 = k$

$$l_1 n_1 = k;$$

$$n_1 = \frac{k}{l_1};$$

when (C2.1) $l_1 \neq 0;$

From (3.22) $n_1^2 + n_3^2 = 1$

$$n_3 = \pm\sqrt{1 - n_1^2} = \pm\sqrt{1 - \left(\frac{k}{l_1}\right)^2}$$

(D) case $y = -1; n_1 = 0; n_2 < 0$

from $n_1^2 + n_2^2 + n_3^2 = 1$ (3.13)

$$n_2^2 + n_3^2 = 1 \qquad (3.24)$$

$l_1 n_1 + l_2 n_2 + l_3 n_3 = k$ (3.14)

$$l_2 n_2 + l_3 n_3 = k \qquad (3.25)$$

when $n_1 = 0; n_2 < 0$

from (3.25) we get $l_2 n_2 + l_3 n_3 = k$

$$n_3 = \frac{k - l_2 n_2}{l_3}$$

(D1) case $l_3 \neq 0$

$$\begin{aligned} n_2^2 + n_3^2 &= 1 \\ n_2^2 + \left(\frac{k - l_2 n_2}{l_3}\right)^2 &= 1 \\ l_3^2 n_2^2 + \left[k^2 - 2k l_2 n_2 - l_2^2 n_2^2\right] &= l_3^2 \\ \left(l_3^2 + l_2^2\right) n_2^2 + \left(-2k l_2\right) n_2 + \left(k^2 - l_3^2\right) &= 0 \end{aligned}$$

n2 = $n_2 = \dfrac{k l_2 \pm \sqrt{k^2 l_2^{\,2} - \left(k^2 - l_3^2\right)\left(l_3^2 + l_2^2\right)}}{\left(l_3^2 + l_2^2\right)}$ ; when $n_2$ is negative

$$n_3 = \frac{k}{l_3} - \frac{l_2}{l_3}(n_2)$$

$$n_1 = 0$$

(D2) case $l_3 = 0$

From (3.25) $l_2 n_2 = k;$

$n2 = k / l_2$ when $l_2 \neq 0; (D2.1)$

if $\frac{k}{l_2} \geq 0$ no solution for $n_3$

$$n_3 = \pm\sqrt{1-n_2^2}$$

else

$$= \pm\sqrt{1-\left(\frac{k}{l_2}\right)^2}$$

end

**Example 1.** Y-Ratio calculation when y>0 and $l_3 \neq 0$

we get data from Example 1 of section 3.5 ( Y-Ratio calculation) Y-Ratio = 0.22

Light($l_1, l_2, l_3$) = (0,0,1).

k is intensity of normalized face image and assume to be 0.2

Then

$$n1 = \frac{k(l_1+\beta l_2) \pm \sqrt{k^2(l_1+\beta l_2)^2 - (k^2 - l_3^2)\left[l_3^2(1+\beta^2)+(l_1+\beta l_2)^2\right]}}{l_3^2(1+\beta^2)+(l_1+\beta l_2)^2}$$

$$n2 = \beta n_1$$

$$n3 = \frac{k-(l_1+\beta l_2)[n_1]}{l_3}$$

Because $n_1$ value is not know. Then for $\beta$ value calculation, $n_1$ is assigned to be more than zero

$$\beta = \frac{y}{1-y} = \frac{0.22}{1-0.22} = 0.28$$

$$n_1 = \frac{\sqrt{-(k^2 - l_3^2)\left[l_3^2(1+\beta^2)\right]}}{l_3^2(1+\beta^2)} = 0.94$$

$$n_2 = \beta n_1 = 0.27$$

$$n_3 = \frac{k}{l_3} = 0.20$$

Normal surface equation $n_1^2 + n_2^2 + n_3^2 = 1$

Then $(0.94)^2 + (0.27)^2 + (0.20)^2 = 1$

**Example 2.** Y-Ratio calculation when y < 0 and $l_3 \neq 0$

we get data from Example 1 of section 3.5 (Y-Ratio calculation) but Y-Rratio = -0.22

Light($l_1, l_2, l_3$) = (0,0,1).

k is intensity of normalized face image and assume to be 0.2

Then $\beta = \frac{y}{1+y} = -0.28$

$n_1 = 0.94$

$n_2 = -0.27$

$n_3 = 0.2$

Normal surface equation $n_1^2 + n_2^2 + n_3^2 = 1$

Then $(0.94)^2 + (-0.27)^2 + (0.20)^2 = 1$

## 3.7 Normal Vector Correction

### 1. concept

This step is a process that corrects sign of normal vectors. The normal vector is a unit vector of each pixel that is perpendicular to the surface but from the step of Y-Ratio calculation, the result normal vectors may have wrong sign (+ or - ) .

### 2. input

(a) Normal surface from an actual normal surface equation.

(b) Height map of an input face image.

### 3. Process

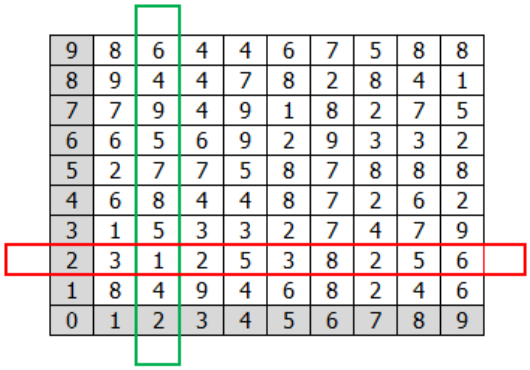
(a) This process runs in every pixel of the height map in both x and y axis from left to right and bottom to top. The computed value from minus of current coordinate pixel and previous coordinate pixel.

(b) If height difference is less than 0, assign the normal vector this pixel as

$$\text{normal vector} = - \left|\text{normal vector}\right|$$

(c) If height difference is equal or more than 0, assign this pixel is

$$\text{normal vector} = \left|\text{normal vector}\right|$$



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 9 | 8 | 6 | 4 | 4 | 6 | 7 | 5 | 8 | 8 |
| 8 | 9 | 4 | 4 | 7 | 8 | 2 | 8 | 4 | 1 |
| 7 | 7 | 9 | 4 | 9 | 1 | 8 | 2 | 7 | 5 |
| 6 | 6 | 5 | 6 | 9 | 2 | 9 | 3 | 3 | 2 |
| 5 | 2 | 7 | 7 | 5 | 8 | 7 | 8 | 8 | 8 |
| 4 | 6 | 8 | 4 | 4 | 8 | 7 | 2 | 6 | 2 |
| 3 | 1 | 5 | 3 | 3 | 2 | 7 | 4 | 7 | 9 |
| 2 | 3 | 1 | 2 | 5 | 3 | 8 | 2 | 5 | 6 |
| 1 | 8 | 4 | 9 | 4 | 6 | 8 | 2 | 4 | 6 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Figure 3.19: Height from input face image

**EXAMPLE 3: CORRECTION OF NORMAL VECTOR IN X-AXIS**

1. Consider height value of x-axis row number 2 and column number 2 height value is 1.
2. Consider previous x-axis row number 2 and column number 1 height value is 3
3. Difference of value is 1 - 3 = -2 then the answer in x axis should be less than zero thus the sign is minus(-).
4. Update sign of normal vector x-axis from actual normal vector in current axis consider.

**EXAMPLE 4 CORRECTION OF NORMAL VECTOR IN Y AXIS**

1. Consider height value of y-axis row number 4 and column number 2 height value is 8.
2. Consider previous y-axis row number 3 and column number 2 height value is 5
3. Difference of value is 8 - 5 = 3 then the answer in y axis should be more than zero thus the sign is plus(+).
4. Update sign of normal vector y-axis from actual normal vector in current axis consider.

Figure. 3.20 shows normal vector where the veary vector are shown with green is show with green arrow and actual normal vector show with red arrow after change sign of vector Figure. 3.21.

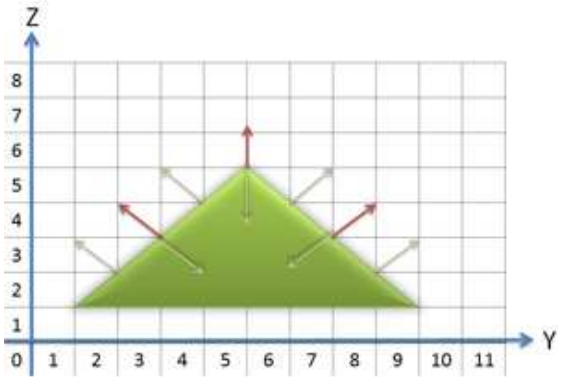



Figure 3.20: before process normal vector

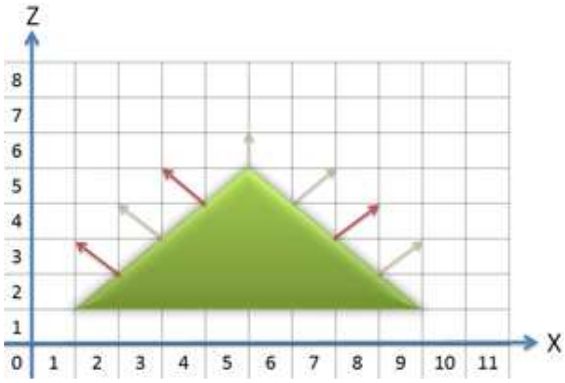



Figure 3.21: after process actual normal vector

### 3.8 The Minimization Approach

The minimization approach uses for computing normal surface as follow

From objection function or energy function

$$E_2 = fitness + smoothness \tag{3.26}$$

The first term of the energy function measures the *fitness* of the image model. The second term of the energy function is *smoothness* constraint on the normals. The minimize energy function is used for computing normal surface from Lambert equation to fit the image model and is used for smoothing the neighbor pixel.

The data value follow from lambert equation

$$I = \rho \cdot NL$$

Normal surface is computed from the normalized face image then $\rho$ is set to equal to one. Then the above equation become

$$I = NL$$

and the condition of light(L), the direction of light is defined as [0,0,1] and thus

$$I = N_1L_1 + N_2L_2 + N_3L_3$$

so... $N_1L_1 = 0, N_2L_2 = 0$

$$I = N_3 \text{or } N_3 = I$$

define $N_3$ to equal to inensity of an image

From a unit vector

$$N_1^2 + N_2^2 + N_3^2 = 1$$

when $N_3 = I$

$$N_2^2 = 1 - N_3^2 + N_1^2$$

$$N_2 = \sqrt{1 - N_3^2 - N_1^2} \tag{3.27}$$

from equation lambert equation

$$I = \rho \cdot NL$$

$$\frac{I}{\rho} = NL$$

$$\frac{I}{\rho} - NL = 0$$

$$\text{Fitness} = \sum_i \left\| \rho I_i - NL \right\|$$

$$smoothness = \lambda \sum \| (N_i - N_j) \|^2$$

Where $i$ = index of a pixel

$j$ = neighbor of i

$\lambda$ = weight of smoothness

Next find the minimum of function using genetic algorithm which consists of 7 steps:[43] [44]

1. **Random initialization and first generation population evaluation** This step is to initiate the first generation of population by selecting randomly a solution random set and then evaluating their fitness values with an objective function.
2. **Selection** This step is to select some population to be processed in the next step. The most commonly used selection method is Roulette Wheel Selection which is like a random spinning of roulette wheel with unequal slot size. That means that each population has unequal probability to be selected into the next generation population. Better fitness value, more probability to be selected into the next generation and the number of selection depends on the number of required population.
3. **Crossing-Over** Crossing-Over is the step to transform population which has been selected from the selection step.
4. **Mutation** This step is to transform population by changing a few of solution data and

turns it into a new solution. The changing method is to invert some bit of binary data.

5. **Fitness evaluation** This step is a process to put each new population data into an objective function equation and to get fitness value of it. For minimization problem, a good solution is a population that has small fitness value.
6. **Population replacement** After evaluation of each population, the old population is replaced with the new population.
7. **End condition checking** If these conditions are not matched, go back to the start process again at step 2 until the ending condition is true.

## 3.9 Relative Height Computation

Consider the reference height and unknow height of neighbor pixel on horizontal plane from local height estimation [45]

1. **Concept**

   Relative height is the difference of height between this pixel and its neighbor pixel.

2. **Input**

   Normal Surface of an input image

   Height difference in x axis direction $d_x H = \left(N_x / N_z\right)$ and

   Height difference in y axis direction $d_y H = \left(N_y / N_z\right)$

## 3.10 Actual Height Computation

### 3.10.1 Affine transformation

Amit Agrawal, Ramesh Raskar , and Rama Chellappa [46] derived a continuum of solvers: $\alpha$ -surface (binary weights) which allows tradeoff between smoothness and robustness, Regularization and M-estimators (continuous weights) and diffusion (affine transformation on gradients). Their results showed that diffusion method gives consistently better feature preserving reconstructions.

Let p(y, x) and q(y, x) denote the given non-integrable gradient field over this grid, Amit Agrawal, Ramesh Raskar, and Rama Chellappa [46] propose to generalize the Poisson solver using D as

$$div\left( D\begin{bmatrix} Z_x \\ Z_y \end{bmatrix} \right) = div\left( D\begin{bmatrix} p \\ q \end{bmatrix} \right)$$

where

$$D(y,x) = \begin{bmatrix} d_{11}(y,x) & d_{12}(y,x) \\ d_{21}(y,x) & d_{22}(y,x) \end{bmatrix}$$

That can be written as

$$div(d_{11}Z_x + d_{12}Z_y, d_{21}Z_x + d_{22}Z_y) = div(d_{11}p + d_{12}q, d_{21}p + d_{22}q).$$

Thus, diffusion corresponds to the function being affine in their arguments.

This process uses relative height ($\nabla_x H, \nabla_y H$) from previous process (process of relative height computation) to calculate actual height by Affine Transformation of gradients using diffusion tensors from [46]. For an example face, the input face, ground truth and actual height of input face was show in Figure.3.22.

(a) ground trut

(b) input face image

(b) height of face ground truth

(d) height of input face image

(e) height of input face image

Figure 3.22: Input and output of height computation

## 3.11 Height Normalization

1. Transform reference model to fit Height map of an input face using the average face model for the reference model
2. Adjust level of an input face image's height map to equal to the reference model with perspective calculation for compensate to perspective effect.

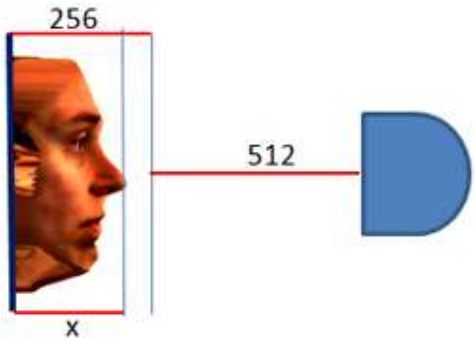



Figure 3.23: model of distance from camera of face

**process**

1. Add perspective effect and adjust height range of an input face image to equal to the average face image

   (a) Change height of an input face input image to back side of image plane (prepare for computing of perspective effect)

   $$h_0 = \text{height of each pixel} - 256$$

   (b) Compute the distance from camera to input face image by

   $$d = 512 + 256 - \text{height of each pixel}$$

(c) From the concept of perspective an image size depends on the distance between an object and an image plane position then change to

actual height(h) = apparent height($h_0$)*distance(d)

h=h0*d

(d) Calculate the range of height of an input face image and average face image from

Height range of input =( maximum of height of input- minimum of height of input)

Height range of average face=( maximum of height of average face - minimum of height of average face)

(e) Shift the value of actual height(h) to starting from 0 by

h=h-min(h)

(f) Adjust actual height(h) =(actual height/range height of input)* range height of average face

2. Adjust an angle of a face (angle of elevation, angle of depression) by using

Point of reference from preprocessing section 3.2, point number 3 and point number 4,that are forehead and chin.

(a) height of point of reference number 3 of input face image was changed height, call "IA3", to equal to height of point reference number 3 of average face, call "GB3" by height of input face image= height of input face image + (GB3-IA3 )

(b) Compute slope of point reference number 3 and 4 of an average face image and input face image

From Equation of a Straight Line y=mx+b, slope(m) = $(y_2-y_1)/(x_2-x_1) = (z_4-z_3)/(y_4-y_3)$

Slope of an average face = $(z_4-z_3)_{gt}/(y_4-y_3)_{gt}$

where

- $z_4$ and $z_3$ are height values of an average face point number 4 and point number3
- $y_4$ and $y_3$ are coordinate point reference number 4 and 3 from aaverage face image

Slope of input face image = $(z_4-z_3)_i/(y_4-y_3)_i$

- $z_4$ and $z_3$ are height value of an input face image point number 4 and point number 3
- $y_4$ and $y_3$ are coordinate point reference number 4 and 3 from an input face image

(c) Compute distance between the slope line of an average face and the slope line of an

input face image

distance slope = (slope of average face - slope of input face image)*(point distance from point of reference number 3)

where point distance from point of reference number 3 is in the range of 1 to 256 minus y-axis point reference number 3

(d) Create plane height of a face image

plane height image = (every column of plane height image = value of slope line at current row).

(e) Calcutate actual height of input face image from

actual height of input face image = height of input face image + plane depth image

## 3.12 Error Estimation

The error of the proposed approach can be measured by Height Difference Error of face region.

The errors are calculated per pixel as

$|\text{height}(z(x,y))\text{of groundtruth} - \text{height}(z(x,y))\text{ of reconstructed face}|$ /height(z(x,y)) of ground truth, In addition, we present the mean values of errors multiplied by 100 to be a percentage.

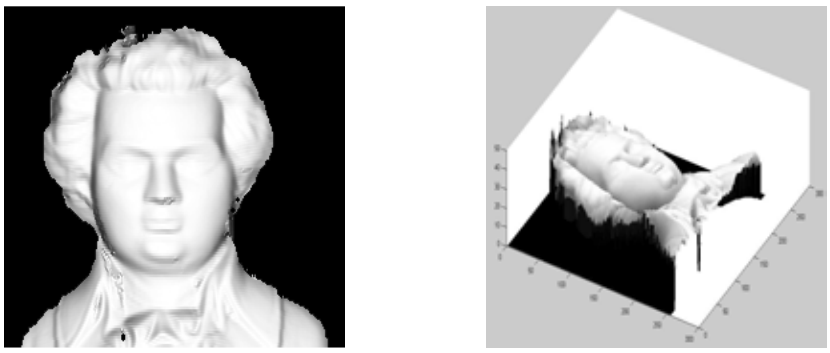
# CHAPTER IV

# EXPERIMENTAL RESULTS

In this chapter the performance of my proposed SFS algorithm is evaluated using human face and synthetic object images. The synthetic object image is a synthetic Mozart's face as shown in Figure. 4.1. that was provided by Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer and Mubarak Shah [2] This object has been used frequently in the SFS literatures as standard test. For human face data, the test set consists of several real images.

## 4.1 Shape From A Synthetic Object Image

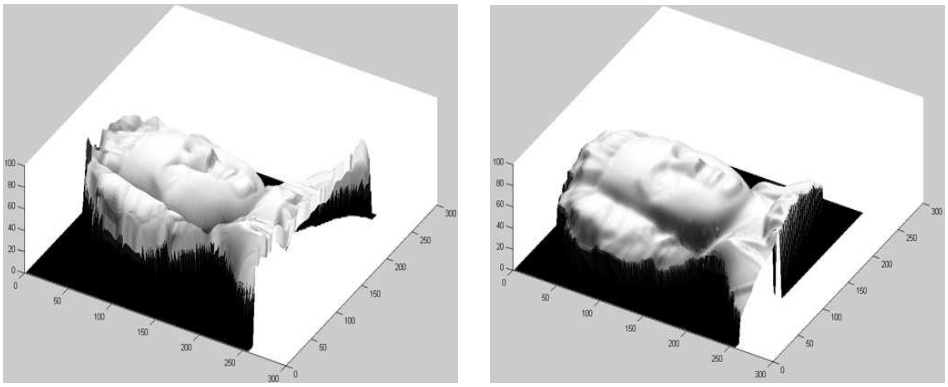
Figure.4.1(a), Mozart's face image, was the first test image with my approach. The proposed of the reconstruction method without an average face model is shown in Figure.4.1(b) and light vector is assumed to be 0,0,1. It is found that this method is not suitable for face reconstruction (error =24.4328%) even if it is an image of a synthetic face model.



(a) Mozart's face image (b) Reconstruction face

Figure 4.1: Mozart synthetic face reconstruction without average face model

So the Mozart face image was test again by reconstruction method with average face model and the result was shown in Figure.4.2(a) with Height Difference Error = 6.6366% compared with the ground truth face image in Figure 4.2(b).

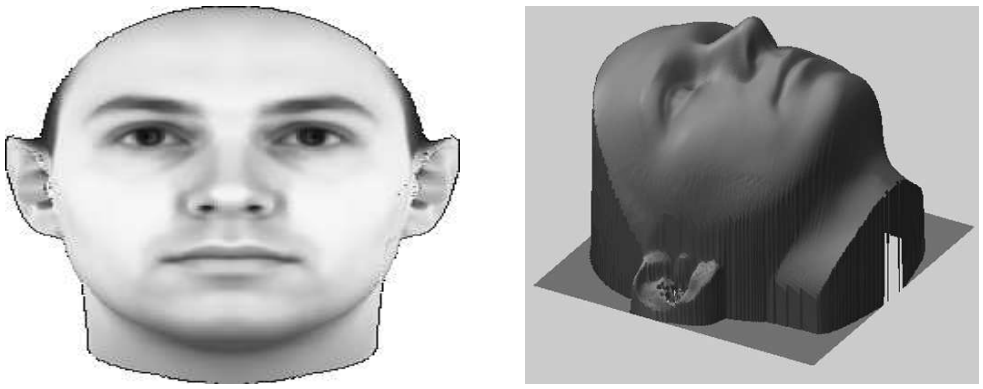


(a) Reconstruction face (b) Ground truth

Figure 4.2: Mozart synthetic face reconstruction with an average face model.

## 4.2 Shape From a Human Face Image

From the previous section, the human face image should be reconstructed with additional data, which are an average face image and an average face height map as shown in Figure. 4.3 The average face data is an average of 200 faces from the database that was provided by the Max-Planck Institute for Biological Cybernetics in Tuebingen, Germany [47].

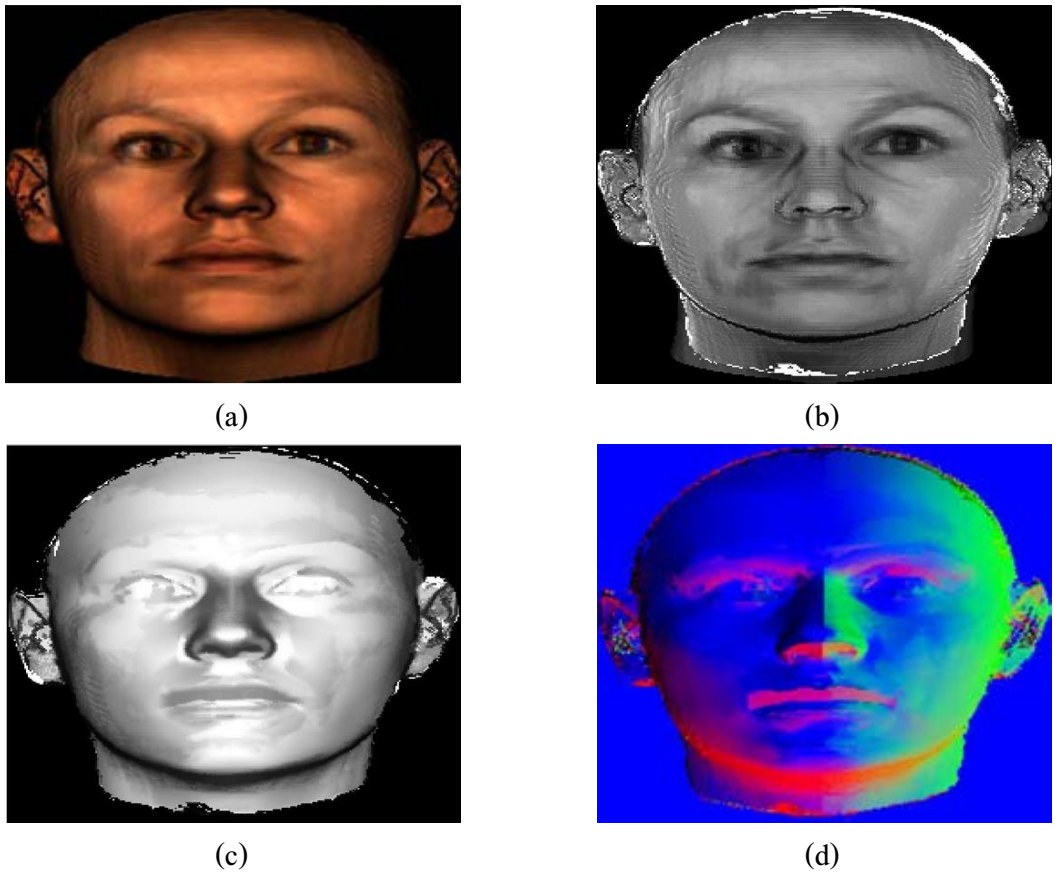


(a) Average face image (b) Average face Height map

Figure 4.3 : Average Face

By using Fast Shape Reconstruction method, which was described in Chapter 3, the first step is to match an average face image and an input face image. The experiment used 256x256 pixels image in "barbara.png", Figure.4.4(a) , from face database of Max-Planck Institute for Biological Cybernetics and then estimated an albedo that is shown in Figure.4.4(b). After the image albedo was estimated, the normalized can be calculated as in Figure.4.4(c) with process time = 25.0255 seconds. The next step is to calculate the normal surface that was done with

Orthogonality of Normal Surface equation with process time = 1.3542 seconds. The normal surface of barbara.png image is shown in Figure.4.4(d).

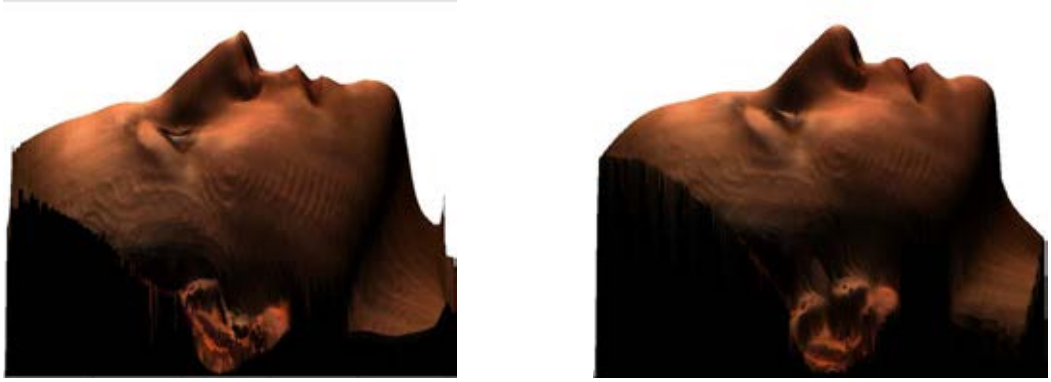


(a) (b)

(c) (d)

Figure 4.4: Input face image and post-processes images

After that, the height map was calculated with Affine transformation [46] and compared with the ground truth as shown in Figure.4.5. With the Height Difference Error= 6.1073% and the total process time = 31.2002 seconds.



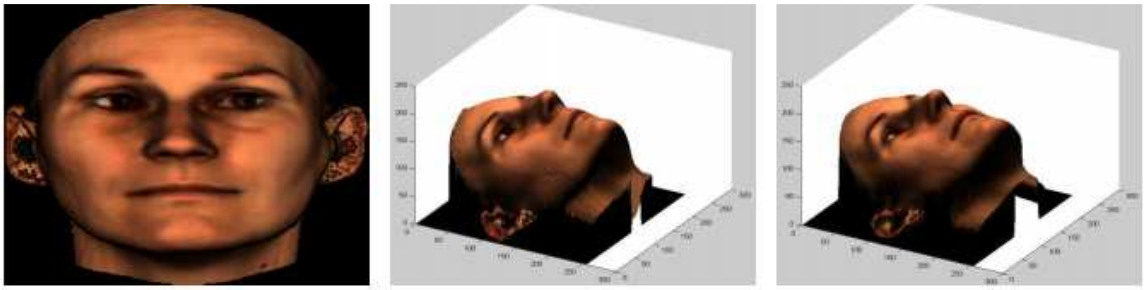
(a) Reconstructed surface of input face image (b) Ground truth

Figure 4.5: Reconstructed surface and ground truth of barbara

## 4.3 Experimental Results of Shape Reconstruction Method

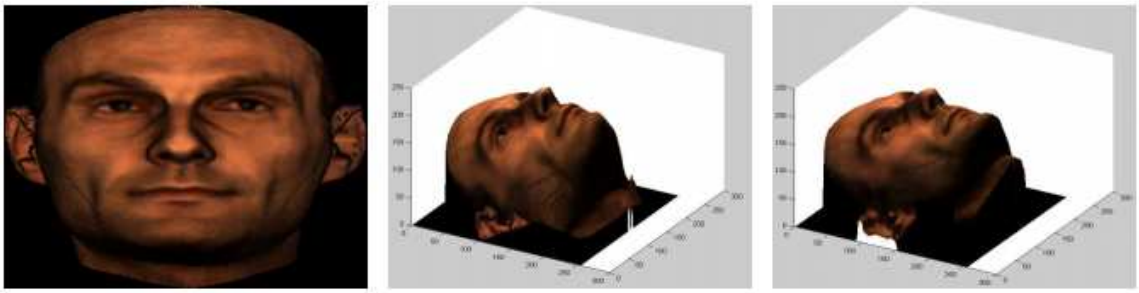
### 4.3.1 Real Human Face Image Experiments

My experiments were done with 4 human face images from face database of Max-Planck Institute for Biological Cybernetics and 10 human face images from Face database of University of Basel. Figure 4.6 to Figure 4.18 show the results of these face reconstruction compared with their ground truth face images.



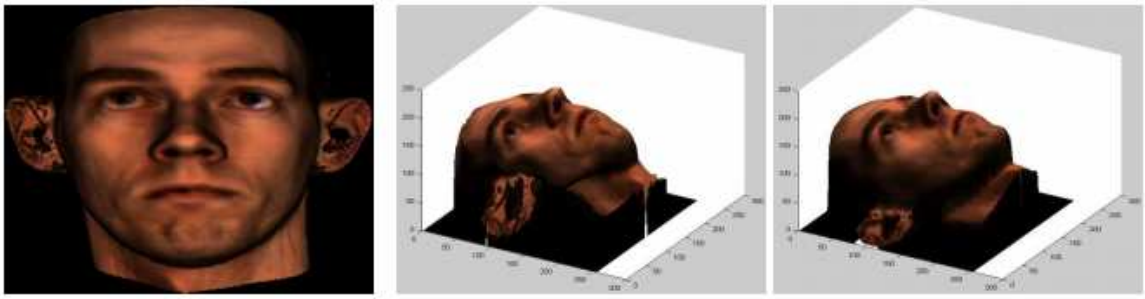
(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.6: Reconstructed surface and ground truth of isabelle



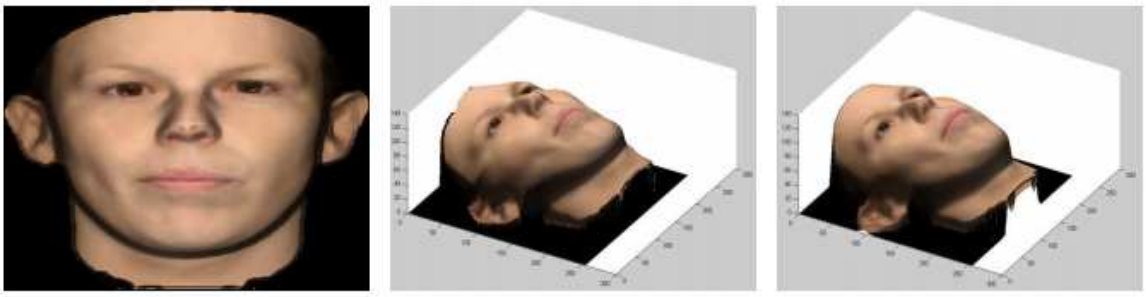
(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.7: Reconstructed surface and ground truth of thomas



(a) Input face image (b) Reconstructed surface of input (c) Ground truth

Figure 4.8: Reconstructed surface and ground truth of volker

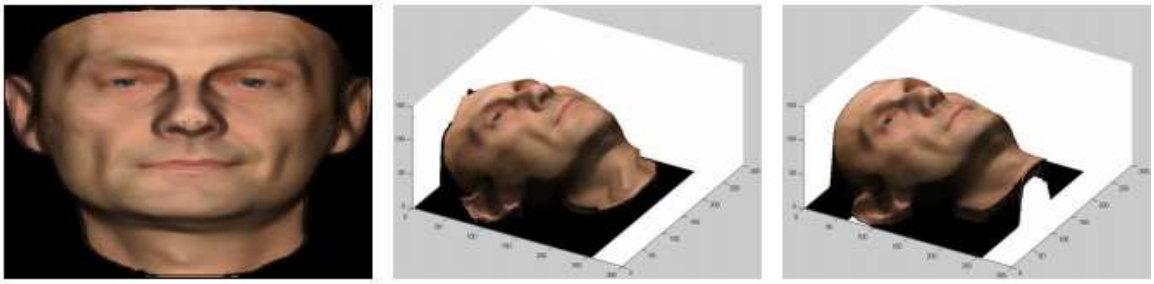


(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.9: Reconstructed surface and ground truth of 001

(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.10: Reconstructed surface and ground truth of 002

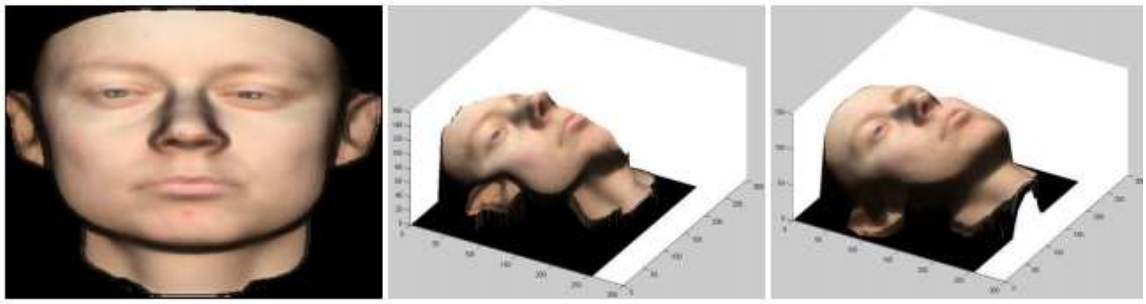


(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.11: Reconstructed surface and ground truth of 006

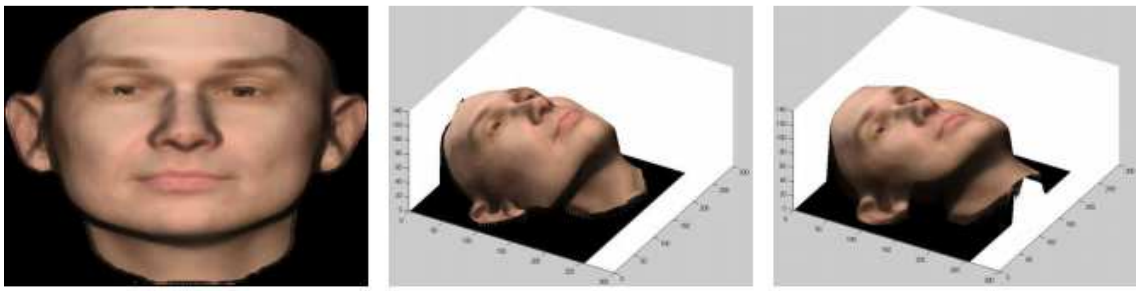
(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.12: Reconstructed surface and ground truth of 014



(a) Input face image (b) Reconstructed Surface of input face image (c) Ground truth

Figure 4.13: Reconstructed surface and ground truth of 017

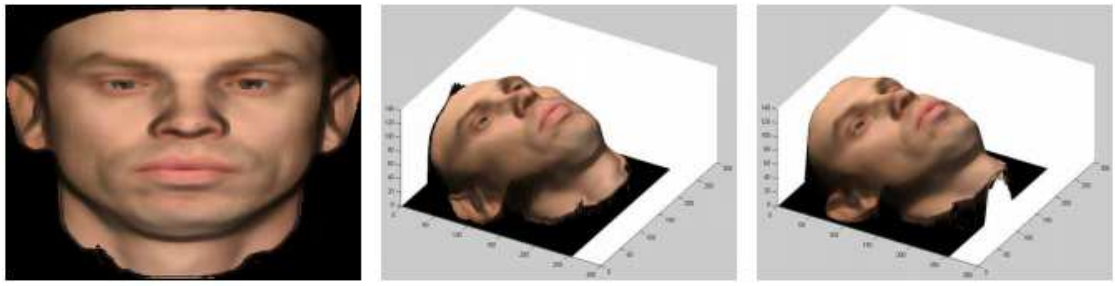


(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.14: Reconstructed surface and ground truth of 022

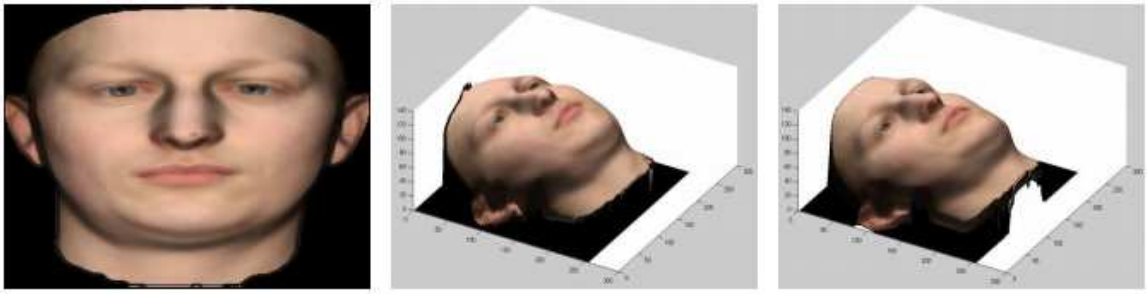
(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.15: Reconstructed surface and ground truth of 052



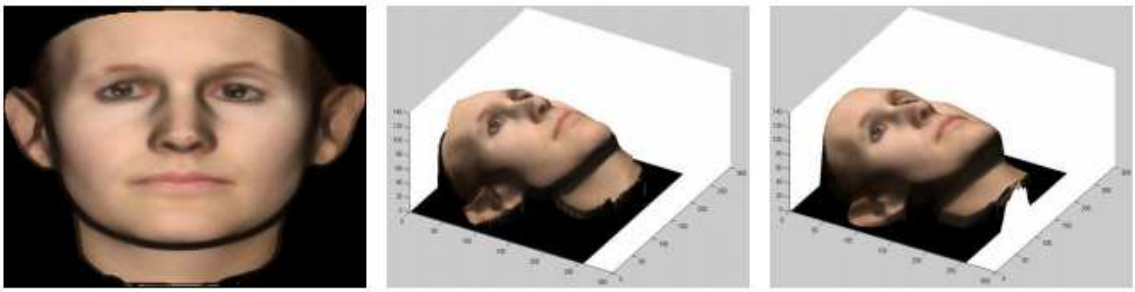
(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.16: Reconstructed surface and ground truth of 053



(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.17: Reconstructed surface and ground truth of 293



(a) Input face image (b) Reconstructed surface of input face image (c) Ground truth

Figure 4.18: Reconstructed surface and ground truth of 323

The result of Height Difference Error(HDE) and main processes time are concluded in Table 4.1.

Table 4.1: Reconstruction result by face reconstruction method

| Image name | Albedo time(sec) | Normal time(sec) | total time(sec) | HDE(%) |
|---|---|---|---|---|
| Barbara.png | 25.026 | 1.354 | 31.200 | 6.107 |
| isabelle.png | 24.593 | 1.300 | 30.587 | 3.517 |
| thomas.png | 24.990 | 1.264 | 30.785 | 4.585 |
| volker.png | 23.760 | 1.210 | 29.412 | 3.221 |
| 001.png | 25.134 | 1.282 | 31.111 | 3.989 |
| 002.png | 25.350 | 1.264 | 31.164 | 2.036 |
| 006.png | 25.242 | 1.282 | 31.002 | 4.774 |
| 014.png | 26.542 | 1.336 | 32.500 | 8.380 |
| 017.png | 25.120 | 1.281 | 30.861 | 5.354 |
| 022.png | 24.456 | 1.228 | 30.288 | 3.275 |
| 052.png | 24.339 | 1.245 | 30.117 | 3.706 |
| 053.png | 25.819 | 1.264 | 31.597 | 3.393 |
| 293.png | 25.838 | 1.300 | 31.688 | 3.630 |
| 323.png | 24.213 | 1.245 | 29.990 | 2.846 |
| Mean | 25.030 | 1.275 | 30.879 | 4.201 |

### 4.3.2 Extended Experiments With 100 Synthetic Face Images

After the experiments of 14 human face images from Face database of Max-Planck Institute for Biological Cybernetics and University of Basel were performaed, additional experiments was performed done with 100 synthetic human face images which are shown in Figure 4.19. The synthetic human face images were synthesized by morphable model of Volker Blanz and Thomas Vetter [47].

The result of each face albedo estimating time, normal surface calculation time, total time and percentage of height different error are shown in Table 4.2 to Table 4.8.

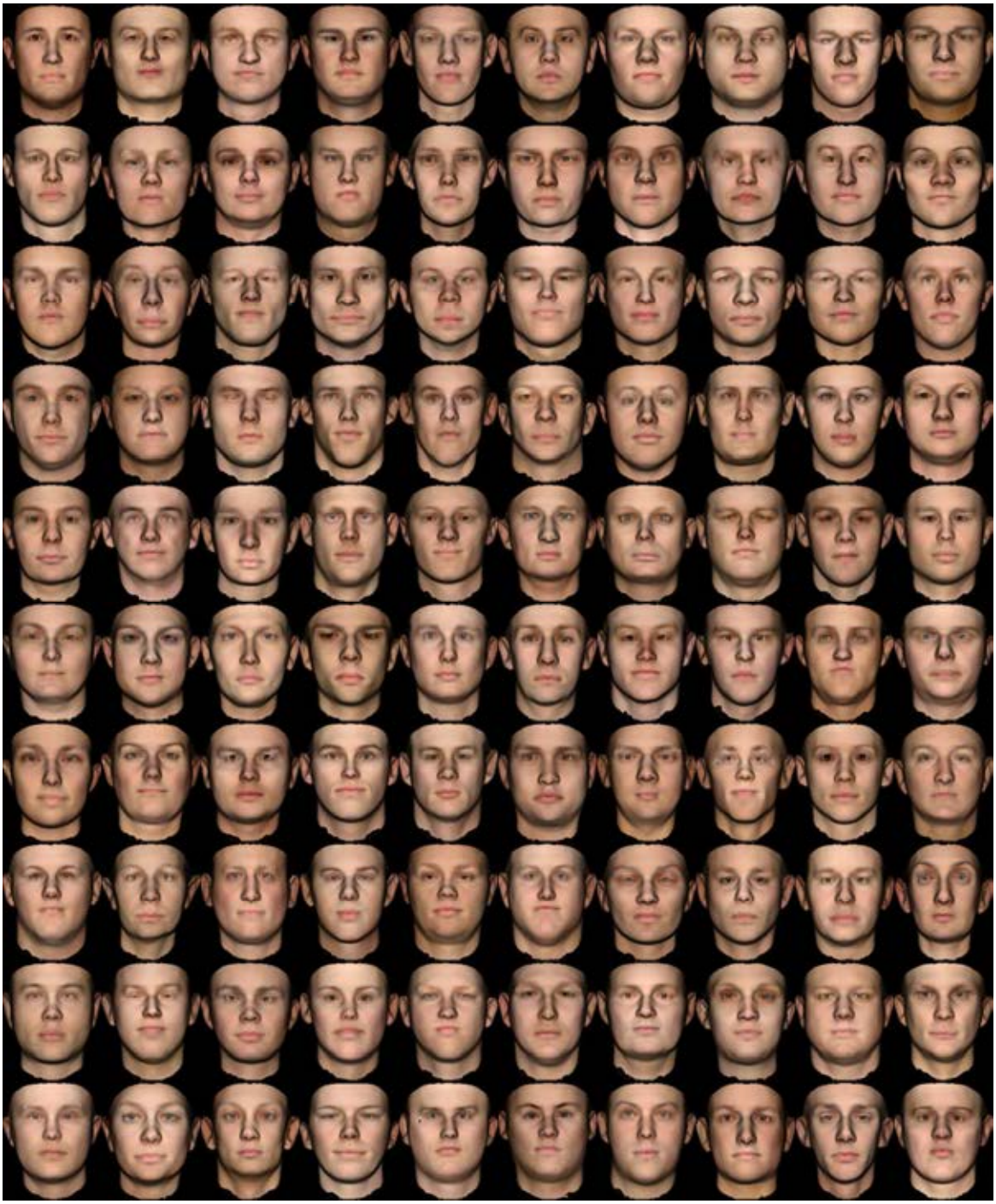


Figure 4.19: 100 human face images

Table 4.2: Reconstruction result by face reconstruction method(1)

| Random face reconstruction | | | | |
|---|---|---|---|---|
| NO. | Albedo time(sec) | Normal time(sec) | total time(sec) | HDE(%) |
| 1 | 24.90 | 1.30 | 30.69 | 3.33 |
| 2 | 25.93 | 1.32 | 31.87 | 5.79 |
| 3 | 25.48 | 1.26 | 31.25 | 3.28 |
| 4 | 25.96 | 1.30 | 31.74 | 6.73 |
| 5 | 24.34 | 1.25 | 30.01 | 8.10 |
| 6 | 25.12 | 1.32 | 30.98 | 2.37 |
| 7 | 26.34 | 1.34 | 32.19 | 4.97 |
| 8 | 26.65 | 1.37 | 32.55 | 5.82 |
| 9 | 25.01 | 1.26 | 30.93 | 2.86 |
| 10 | 26.83 | 1.34 | 32.68 | 3.57 |
| 11 | 25.37 | 1.32 | 31.09 | 5.96 |
| 12 | 26.34 | 1.28 | 32.12 | 5.71 |
| 13 | 25.78 | 1.41 | 31.92 | 3.42 |
| 14 | 25.19 | 1.41 | 31.34 | 6.51 |
| 15 | 23.53 | 1.30 | 29.61 | 4.16 |
| 16 | 26.00 | 1.43 | 32.12 | 6.21 |
| 17 | 25.80 | 1.41 | 31.83 | 5.05 |
| 18 | 24.86 | 1.37 | 31.09 | 4.53 |
| 19 | 25.82 | 1.41 | 32.01 | 4.46 |
| 20 | 25.98 | 1.43 | 32.10 | 7.91 |
| 21 | 26.14 | 1.41 | 32.39 | 2.80 |
| 22 | 24.92 | 1.35 | 31.07 | 3.02 |
| 23 | 26.25 | 1.43 | 32.41 | 3.56 |
| 24 | 26.14 | 1.43 | 32.36 | 3.75 |
| 25 | 24.99 | 1.37 | 31.07 | 3.69 |
| 26 | 26.33 | 1.46 | 32.57 | 4.42 |

Table 4.3: Reconstruction result by face reconstruction method(2)

| Random synthetic face reconstruction | | | | |
|---|---|---|---|---|
| NO. | Albedo time(sec) | Normal time(sec) | total time(sec) | HDE(%) |
| 27 | 25.39 | 1.41 | 31.67 | 2.01 |
| 28 | 26.23 | 1.43 | 32.50 | 5.85 |
| 29 | 25.64 | 1.43 | 31.80 | 4.13 |
| 30 | 25.30 | 1.39 | 31.51 | 2.89 |
| 31 | 25.96 | 1.44 | 32.17 | 3.72 |
| 32 | 25.44 | 1.39 | 31.58 | 3.30 |
| 33 | 25.96 | 1.43 | 32.17 | 2.77 |
| 34 | 25.77 | 1.41 | 31.96 | 3.18 |
| 35 | 24.77 | 1.35 | 30.93 | 3.14 |
| 36 | 28.00 | 1.47 | 34.39 | 3.94 |
| 37 | 25.51 | 1.41 | 31.67 | 5.88 |
| 38 | 24.90 | 1.37 | 30.98 | 5.06 |
| 39 | 24.83 | 1.39 | 30.97 | 4.23 |
| 40 | 25.95 | 1.41 | 32.03 | 3.02 |
| 41 | 25.49 | 1.41 | 31.69 | 6.80 |
| 42 | 26.92 | 1.50 | 33.18 | 2.17 |
| 43 | 25.13 | 1.39 | 31.27 | 3.03 |
| 44 | 25.73 | 1.37 | 31.87 | 3.72 |
| 45 | 25.33 | 1.39 | 31.49 | 2.76 |
| 46 | 24.61 | 1.35 | 30.77 | 2.64 |
| 47 | 25.58 | 1.41 | 31.76 | 3.37 |
| 48 | 26.18 | 1.44 | 32.41 | 2.50 |
| 49 | 25.37 | 1.37 | 31.54 | 3.20 |
| 50 | 24.86 | 1.37 | 31.07 | 2.26 |
| 51 | 26.00 | 1.43 | 32.17 | 5.28 |
| 52 | 24.93 | 1.39 | 31.06 | 3.83 |

Table 4.4 Reconstruction result by face reconstruction method(3)

| Random synthetic face reconstruction | | | | |
|---|---|---|---|---|
| NO. | Albedo time(sec) | Normal time(sec) | total time(sec) | HDE(%) |
| 53 | 25.15 | 1.41 | 31.29 | 4.76 |
| 54 | 26.14 | 1.41 | 32.28 | 13.97 |
| 55 | 25.33 | 1.43 | 31.49 | 3.36 |
| 56 | 24.66 | 1.55 | 31.06 | 3.27 |
| 57 | 25.53 | 1.44 | 31.81 | 5.69 |
| 58 | 25.04 | 1.35 | 31.09 | 5.15 |
| 59 | 24.88 | 1.35 | 31.02 | 3.08 |
| 60 | 25.93 | 1.44 | 32.16 | 4.62 |
| 61 | 25.51 | 1.43 | 31.65 | 6.87 |
| 62 | 25.80 | 1.41 | 31.96 | 3.87 |
| 63 | 25.64 | 1.43 | 31.83 | 6.04 |
| 64 | 25.40 | 1.39 | 31.62 | 5.93 |
| 65 | 25.19 | 1.39 | 31.40 | 2.92 |
| 66 | 25.86 | 1.43 | 32.10 | 2.76 |
| 67 | 24.99 | 1.37 | 31.15 | 4.68 |
| 68 | 24.50 | 1.35 | 30.62 | 4.33 |
| 69 | 25.30 | 1.44 | 31.51 | 3.28 |
| 70 | 26.20 | 1.44 | 32.46 | 3.23 |
| 71 | 26.52 | 1.43 | 32.75 | 6.24 |
| 72 | 23.44 | 1.30 | 29.56 | 4.04 |
| 73 | 25.13 | 1.39 | 31.33 | 3.33 |
| 74 | 24.45 | 1.35 | 30.62 | 3.46 |
| 75 | 25.78 | 1.41 | 31.92 | 3.00 |
| 76 | 25.40 | 1.41 | 31.71 | 4.05 |
| 77 | 25.53 | 2.11 | 33.18 | 4.59 |
| 78 | 24.99 | 1.37 | 31.09 | 2.39 |
| 79 | 25.03 | 1.39 | 31.09 | 5.92 |

Table 4.5: Reconstruction result by face reconstruction method(4)

| Random synthetic face reconstruction | | | | |
|---|---|---|---|---|
| NO. | Albedo time(sec) | Normal time(sec) | total time(sec) | HDE(%) |
| 80 | 25.63 | 1.34 | 31.89 | 5.17 |
| 81 | 25.86 | 1.41 | 32.07 | 6.44 |
| 82 | 24.92 | 1.39 | 31.09 | 2.30 |
| 83 | 25.48 | 1.37 | 31.56 | 6.16 |
| 84 | 24.86 | 1.37 | 30.98 | 2.67 |
| 85 | 27.13 | 1.43 | 33.42 | 3.01 |
| 86 | 25.37 | 1.39 | 31.45 | 3.92 |
| 87 | 25.21 | 1.41 | 31.38 | 6.34 |
| 88 | 24.99 | 1.39 | 31.04 | 3.14 |
| 89 | 25.93 | 1.44 | 32.07 | 3.61 |
| 90 | 25.06 | 1.37 | 31.25 | 4.12 |
| 91 | 25.30 | 1.39 | 31.44 | 9.61 |
| 92 | 25.48 | 1.39 | 31.65 | 4.74 |
| 93 | 24.66 | 1.41 | 30.84 | 3.03 |
| 94 | 25.51 | 1.39 | 31.63 | 4.85 |
| 95 | 25.44 | 1.39 | 31.56 | 3.65 |
| 96 | 26.05 | 1.43 | 32.41 | 7.31 |
| 97 | 25.19 | 1.41 | 31.31 | 4.08 |
| 98 | 25.53 | 1.39 | 31.71 | 4.13 |
| 99 | 24.25 | 1.34 | 30.31 | 4.84 |
| 100 | 25.80 | 1.41 | 31.96 | 4.39 |
| Average | 25.49 | 1.40 | 31.63 | 4.39 |

From Tables 4.1, 4.2, 4.3, 4.4 and 4.5, the total results of all face images were retrieved and there show that the average of normal surface calculation time is only 1.3822 seconds and average of height difference error is as low as 4.3669 % with standard deviation = 1.7461. Then a graph of height difference error and a graph of time of normal surface process and a graph of time

of albedo shown process are in Figure 4.20, Figure 4.21 and Figure 4.22

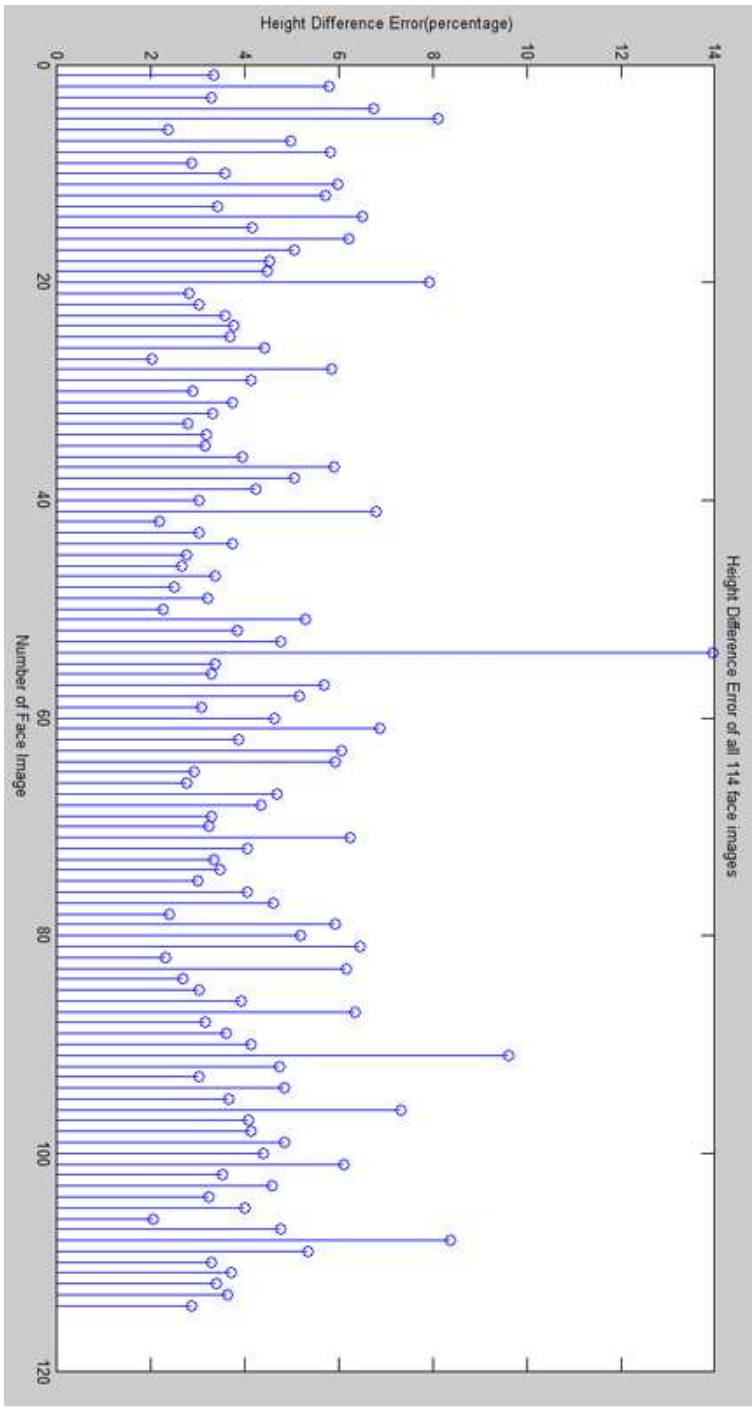



Figure 4.20: Height difference error of all 114 face image

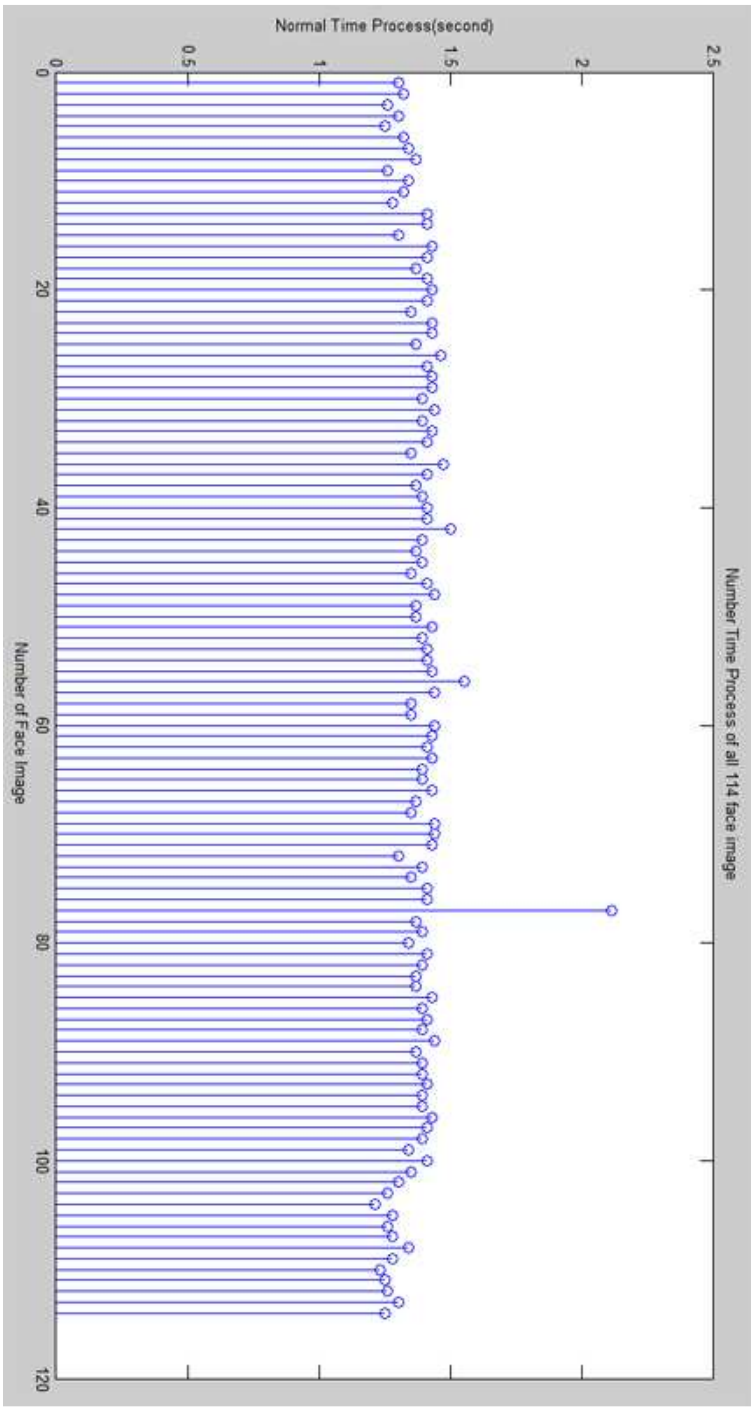



Figure 4.21: Time of process normal surface

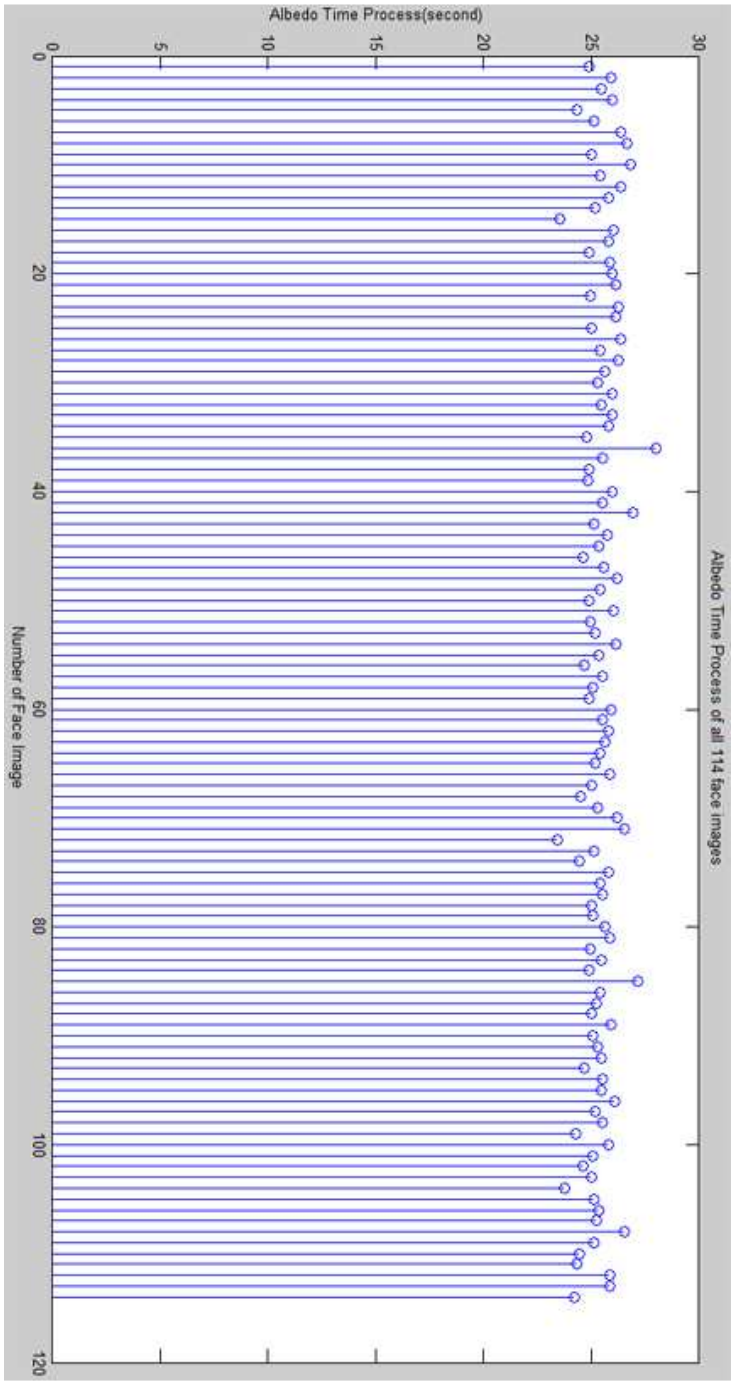



Figure 4.22: Time of process albedo

## 4.4 Efficiency of Normal Surface Calculation by Orthogonality of Normal Surface Equation

The main important process in shape reconstruction is normal surface calculation which usually uses traditional method of energy function minimization. Generally, the minimization algorithm consumes very long time process; therefore I propose the normal surface calculation algorithm that was described in chapter 3.

In this experiment, I minimize this energy function.

$$E = \sum_{i \varepsilon P} \left\| \rho^{'} I_i - N_i^T L \right\|^2 + \lambda \sum_{\{i,j\}} \left\| N_i - N_j \right\| \tag{4.1}$$

Where $P$ is the foreground region to be processed, $i, j$ are the first order neighbor pair, $I$ is the pixel's intensity of an input image, $L$ is the light source direction unit vector, $\lambda$ is a smoothness factor, $\rho = \rho^{-1}$, and $N$ is the result of minimization process that is a normal vector of an input image.

As an example, Mozart's face image has been reconstructed by using traditional minimization method and the reconstructed face is shown in Figure. 4.23(a) compared with using my method(Figure 4.23(b)).

(a) Reconstructed face by minimization method

(b) Reconstructed face by my method

Figure 4.23: Mozart synthetic face image reconstruction

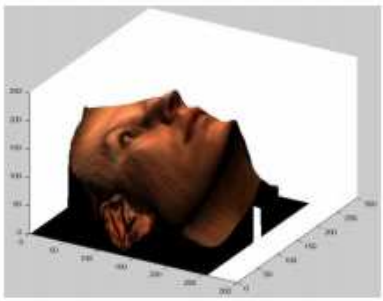
Table 4.6 shows the comparison of results by using minimization method, the mozart's reconstructed face has the height different error = 14.93%, little worse than that was reconstructed by my method 6.6366 % (from section 4.1: shape from a synthetic object image). But the normal surface calculation time is 3.9848 minutes that is 200 time of normal surface calculation of my method.

The human face experimental results with the traditional minimization method are shown in Figure 4.24 to Figure 4.37 compared with the ground truth and reconstructed surface by my approach. Selection of minimization method in this research is Zheng and Chellappa [5] approach, which is implemented by Ruo Zhang and Ping-Sing Tsai and el.[2] and Chen,D and Dong,F[15] with addition process of albedo estimation.

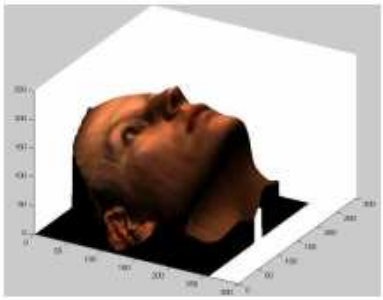
Table 4.6: Compare results of our method, D.Chen,F.Dong and Zheng el.

| | minimization method | | our method | |
|---|---|---|---|---|
| | height difference error(%) | time(s) | height difference error(%) | time(s) |
| D.Chen, F.Dong | 9.6 | - | | |
| zheng and chellappa | 18.4 | - | | |
| Our propose | 14.93 | 239.057 | 6.6366 | 1.1921 |

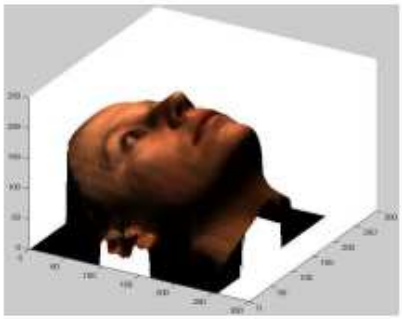
(a) babara.png



(b) minimized approach

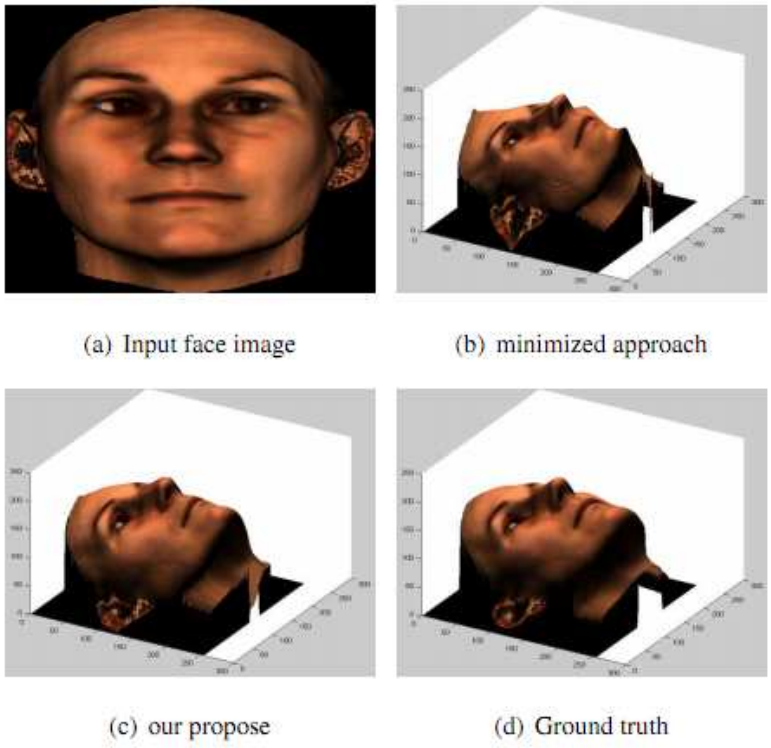


(c) our propose



(d) Ground truth

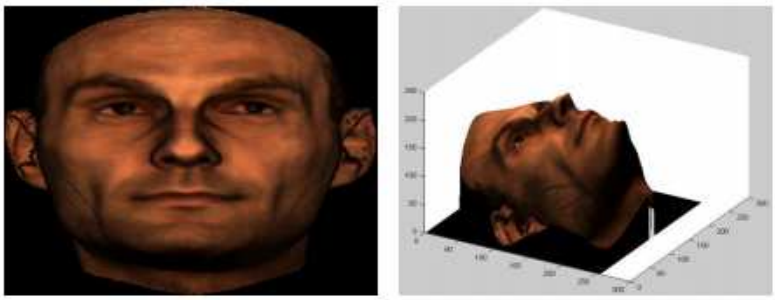
Figure 4.24: babara.png



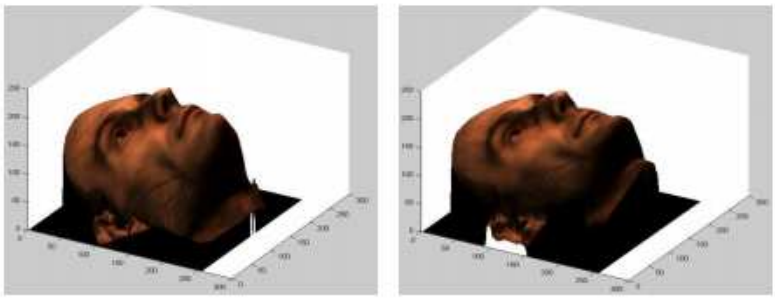
(a) Input face image (b) minimized approach

(c) our propose (d) Ground truth

Figure 4.25: Reconstructed surface of isabelle

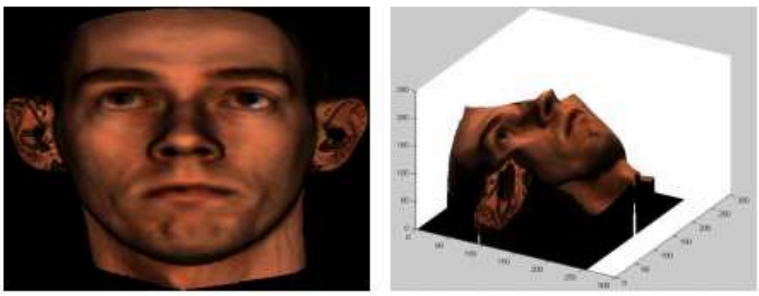


(a) Input face image (b) minimized approach

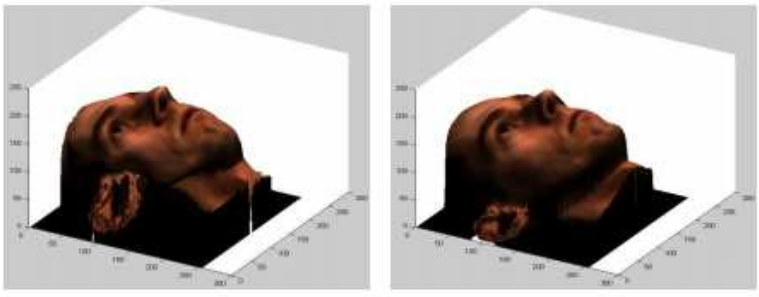


(c) Our propose (d) Ground truth

Figure 4.26: Reconstructed surface of thomas

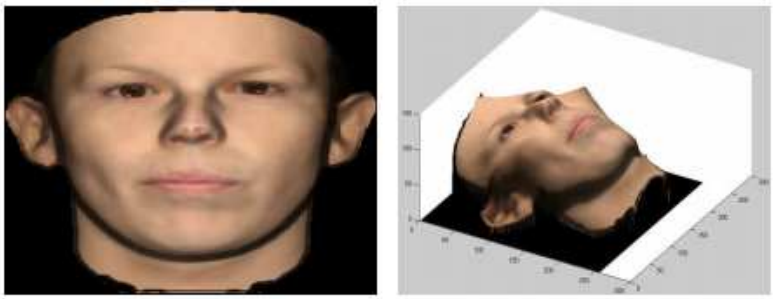


(a) Input face image (b) minimized approach

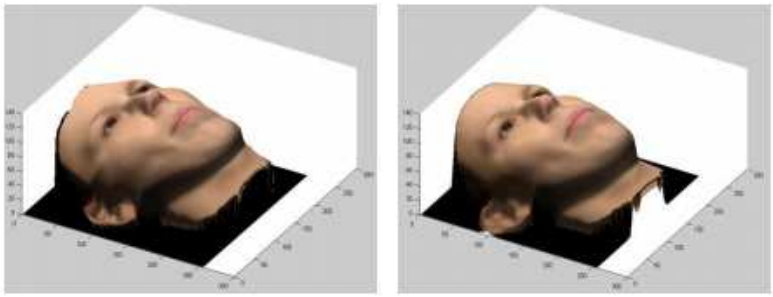


(c) Our propose (d) Ground truth

Figure 4.27: Reconstructed surface of volker

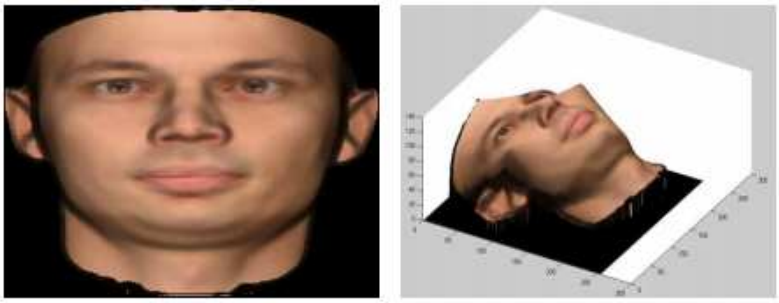


(a) Input face image (b) minimized approach

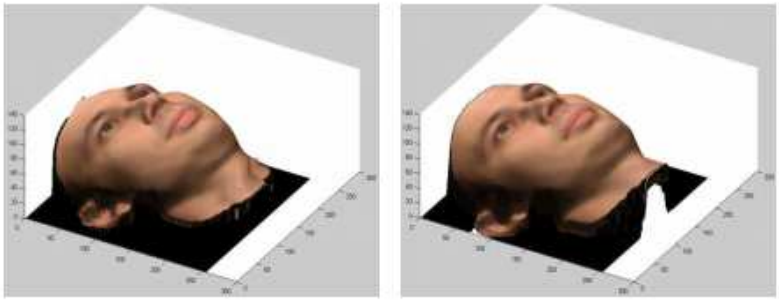


(c) Our propose (d) Ground truth

Figure 4.28: Reconstructed surface of 001

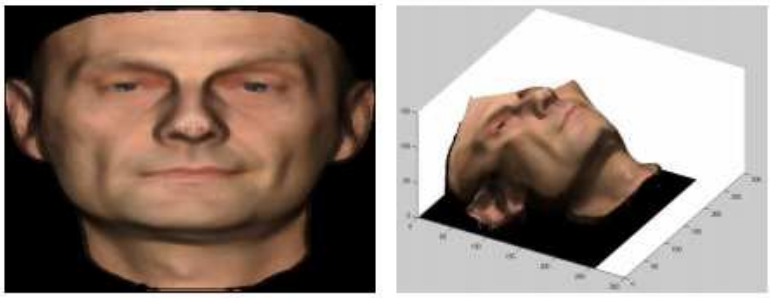


(a) Input face image (b) minimized approach

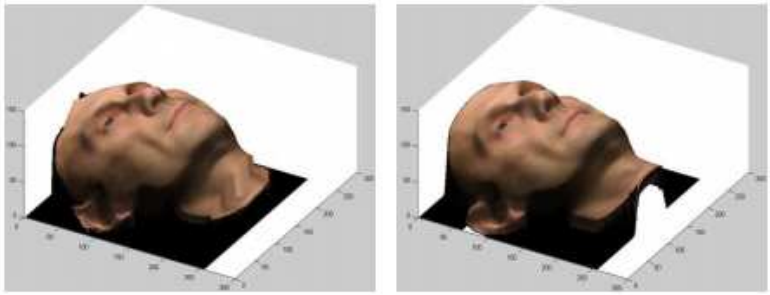


(c) Our propose (d) Ground truth

Figure 4.29: Reconstructed surface of 002

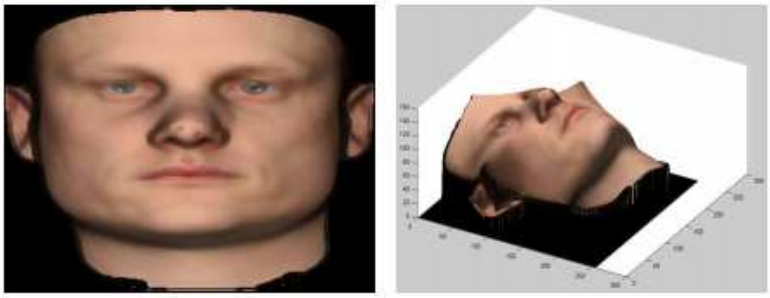


(a) Input face image (b) minimized approach

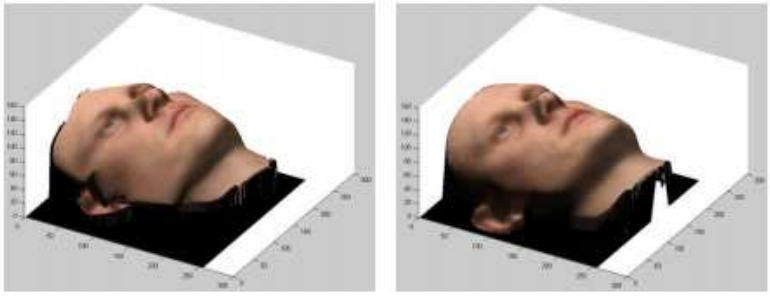


(c) our propose (d) Ground truth

Figure 4.30: Reconstructed surface of 006

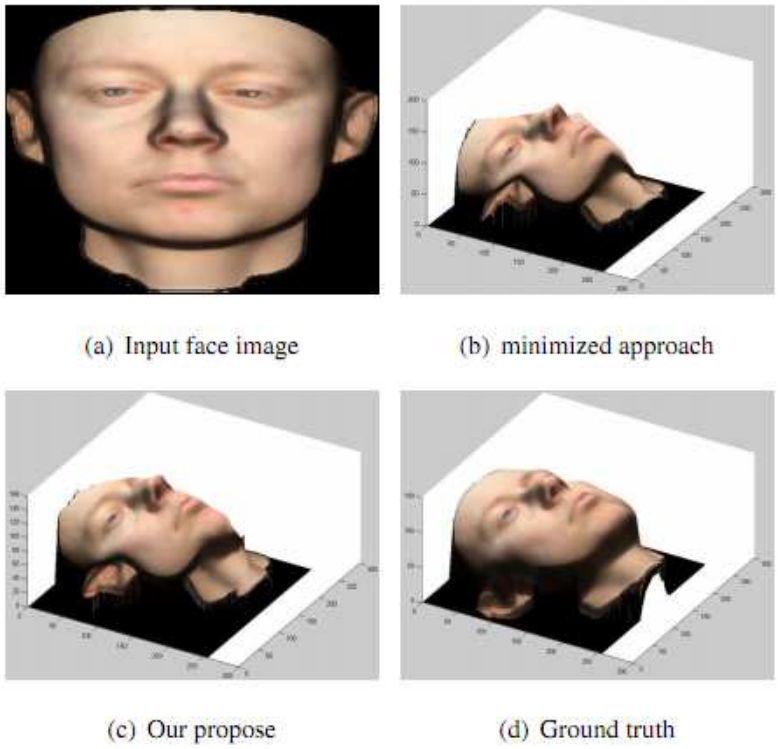


(a) Input face image (b) minimized approach



(c) Our propose (d) Ground truth

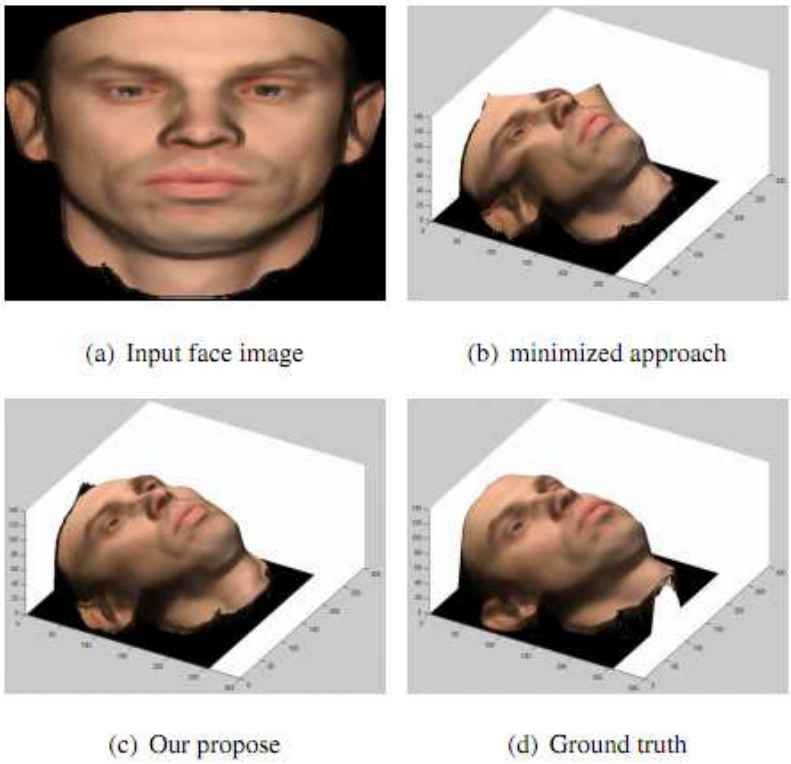
Figure 4.31: Reconstructed surface of 014



(a) Input face image (b) minimized approach

(c) Our propose (d) Ground truth

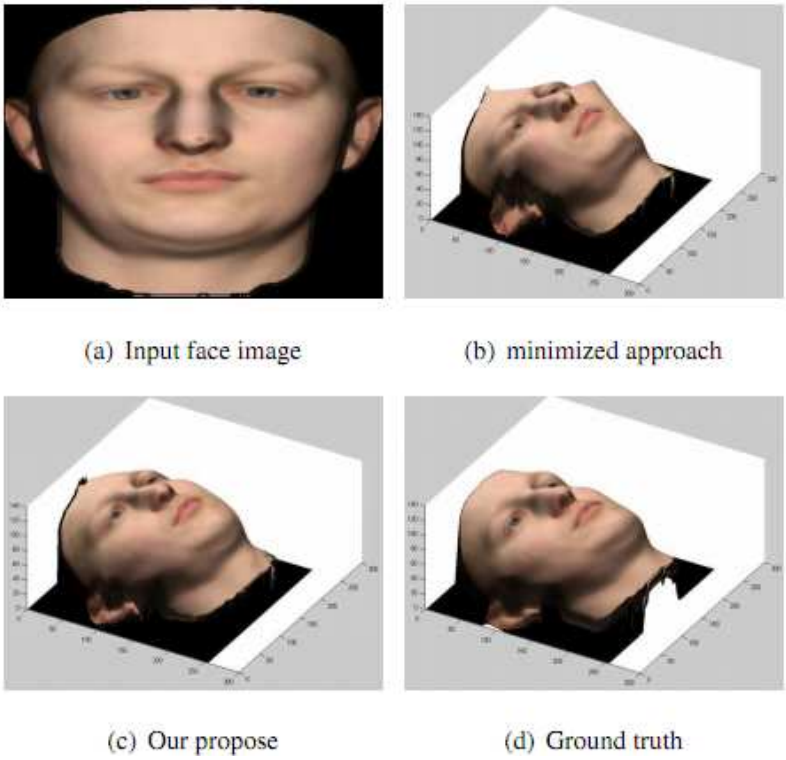
Figure 4.33: Reconstructed surface of 022



(a) Input face image (b) minimized approach

(c) Our propose (d) Ground truth

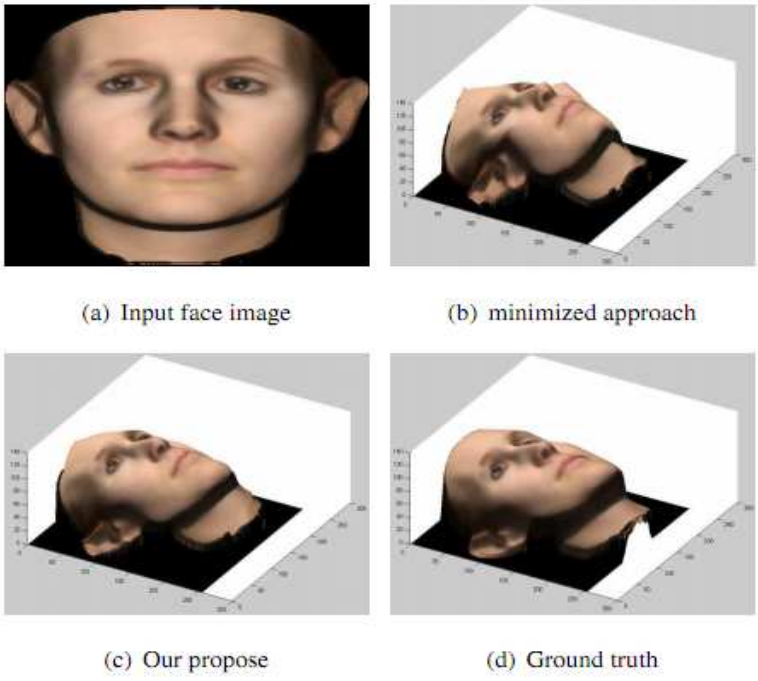
Figure 4.35: Reconstructed surface of 053



(a) Input face image (b) minimized approach

(c) Our propose (d) Ground truth

Figure 4.36: Reconstructed surface of 293



(a) Input face image (b) minimized approach

(c) Our propose (d) Ground truth

Figure 4.37: Reconstructed surface of 323

The Height Difference Error of each approach and normal surface calculation process time are concluded in Table 4.7.

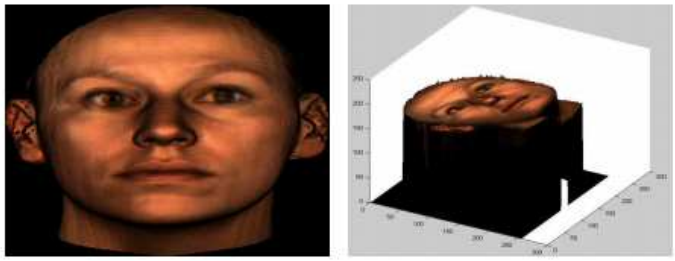
Table 4.7: Comparation of my approach (Face reconstruction method : FR) and minimization reconstruction approach (MR)

| Image name | FFR HDE (%) | MR HDE (%) | FR Normals calculation time(sec) | MR Normals calculation time(sec) |
|---|---|---|---|---|
| Barbara.png | 6.11 | 6.96 | 1.35 | 246.86 |
| isabelle.png | 3.52 | 4.08 | 1.30 | 324.32 |
| thomas.png | 4.58 | 3.91 | 1.26 | 341.42 |
| volker.png | 3.22 | 6.74 | 1.21 | 417.16 |
| 001.png | 3.99 | 4.88 | 1.28 | 249.68 |
| 002.png | 2.04 | 3.45 | 1.26 | 271.65 |
| 006.png | 4.77 | 5.07 | 1.28 | 494.61 |
| 014.png | 8.38 | 6.86 | 1.34 | 307.72 |
| 017.png | 5.35 | 10.59 | 1.28 | 304.12 |
| 022.png | 3.27 | 5.83 | 1.23 | 318.12 |
| 052.png | 3.71 | 9.73 | 1.25 | 975.93 |
| 053.png | 3.39 | 4.97 | 1.26 | 302.57 |
| 293.png | 3.63 | 5.30 | 1.30 | 318.84 |
| 323.png | 2.85 | 5.52 | 1.25 | 345.56 |
| **Mean** | **4.20** | **5.99** | **1.28** | **372.75** |

## 4.5 Efficiency of Albedo Estimation Algorithm

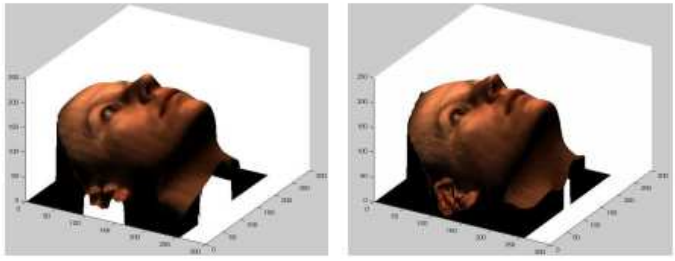
Since there are many dissimilar reflectance property areas in a human face, so the normal surface calculation should not be done with original face image. The face image have to be normalized by using its albedo, therefore, the albedo estimating process is necessary. The albedo estimating process has been described in chapter 3.

I have run the experiment that compared the result from normalized image and non-normalized image and present the result in Figure 4.38 to Figure 4.51



(a) babara.png

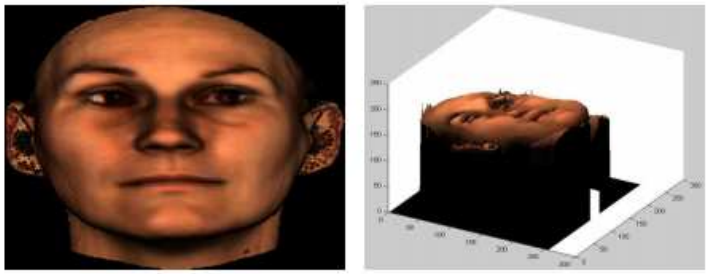
(b) Reconstructed surface by minimized approach



(c) Ground truth

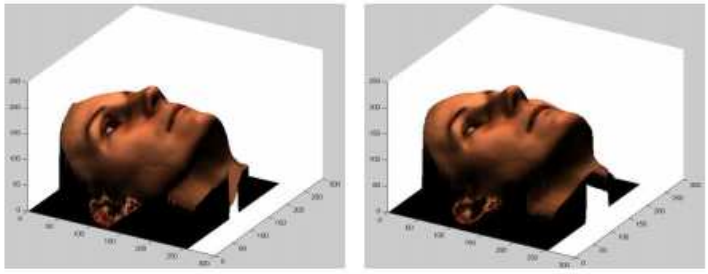
(d) Reconstructed Surface of input face image

Figure 4.38: Reconstructed surface and ground truth of babara.png



(a) Input face image

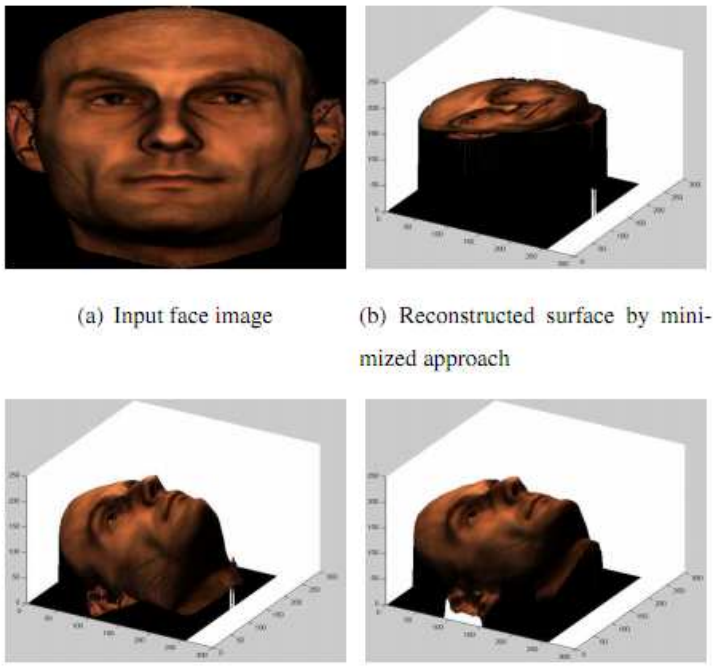
(b) Reconstructed surface by minimized approach



(c) Reconstructed Surface of input face image

(d) Ground truth

Figure 4.39: Reconstructed Surface and Ground truth of isabelle



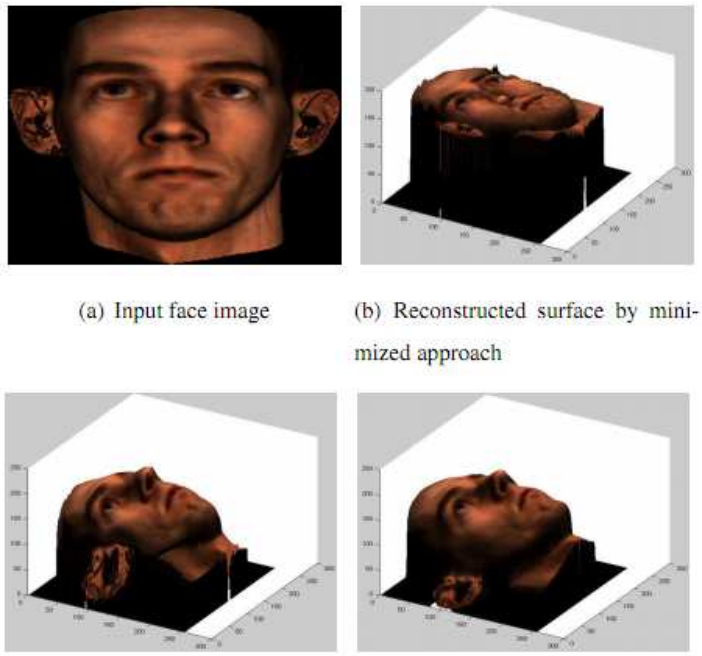
(a) Input face image

(b) Reconstructed surface by minimized approach

(c) Reconstructed Surface of input face image

(d) Ground truth

Figure 4.40: Reconstructed surface and ground truth of thomas



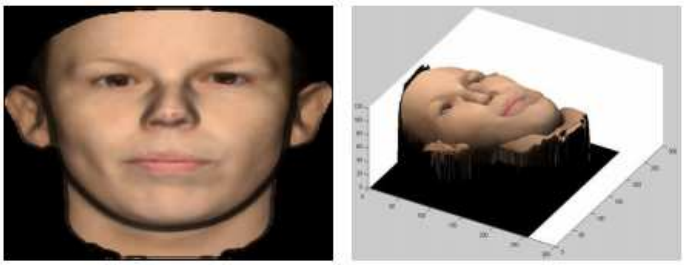
(a) Input face image

(b) Reconstructed surface by minimized approach

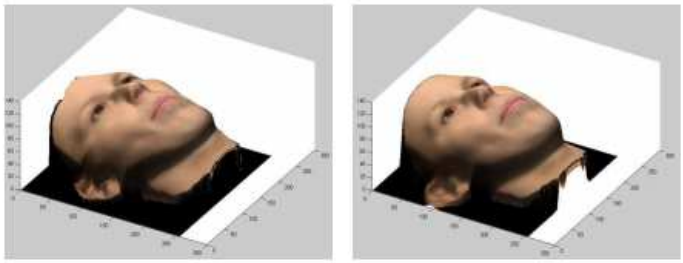
(c) Reconstructed Surface of input face image

(d) Ground truth

Figure 4.41: Reconstructed Surface and Ground truth of volker

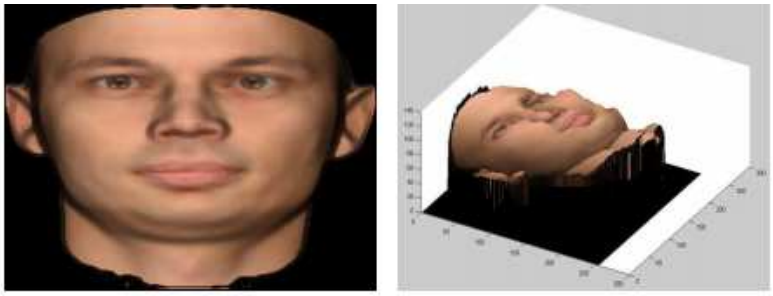


(a) Input face image (b) Reconstructed surface by minimized approach

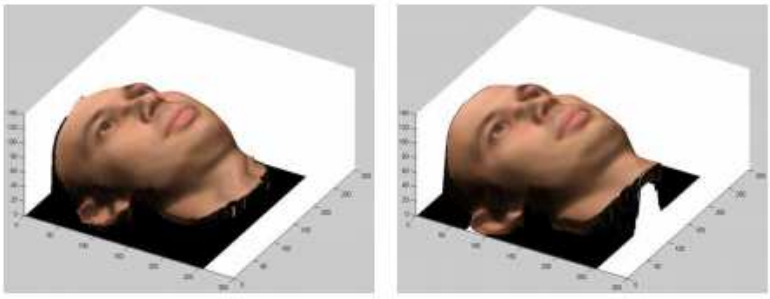


(c) Reconstructed Surface of input face image (d) Ground truth

Figure 4.42: Reconstructed Surface and Ground truth of 001

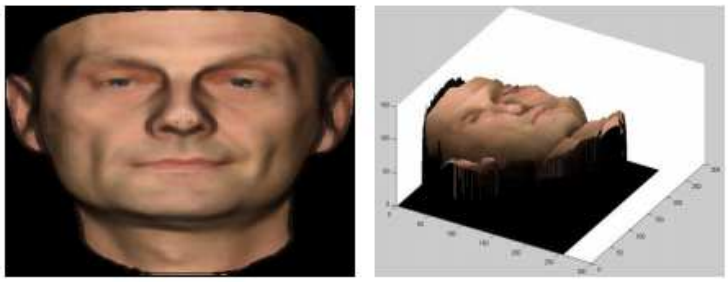


(a) Input face image (b) Reconstructed surface by minimized approach

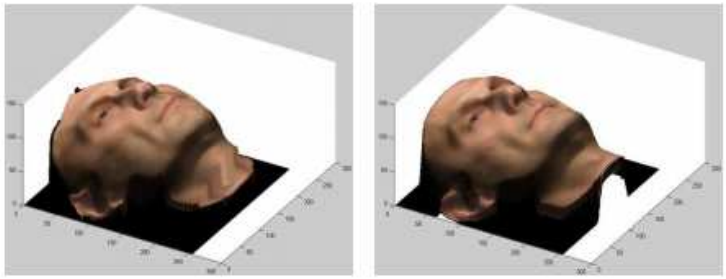


(c) Reconstructed Surface of input face image (d) Ground truth

Figure 4.43: Reconstructed Surface and Ground truth of 002

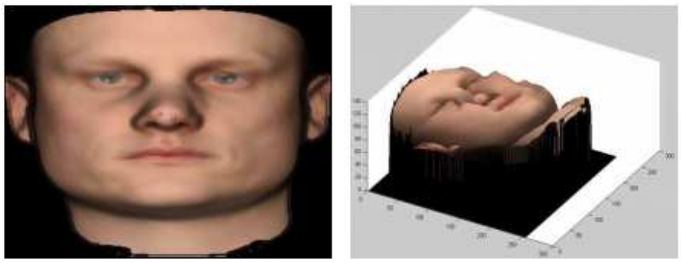


(a) Input face image (b) Reconstructed surface by minimized approach

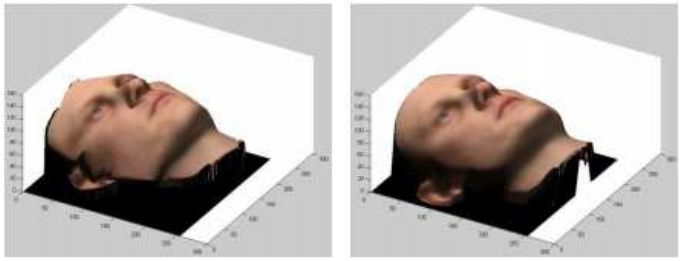


(c) Reconstructed Surface of input face image (d) Ground truth

Figure 4.44: Reconstructed Surface and Ground truth of 006

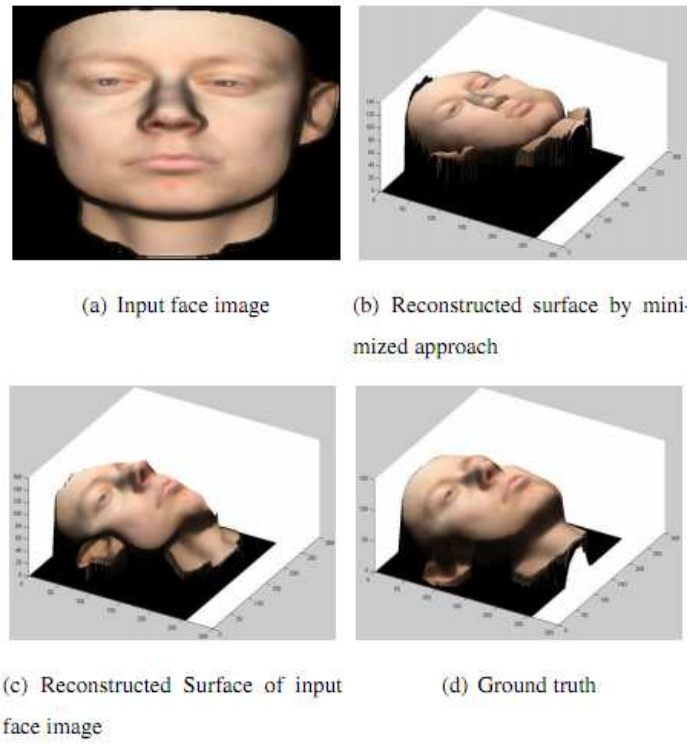


(a) Input face image (b) Reconstructed surface by minimized approach



(c) Reconstructed Surface of input face image (d) Ground truth

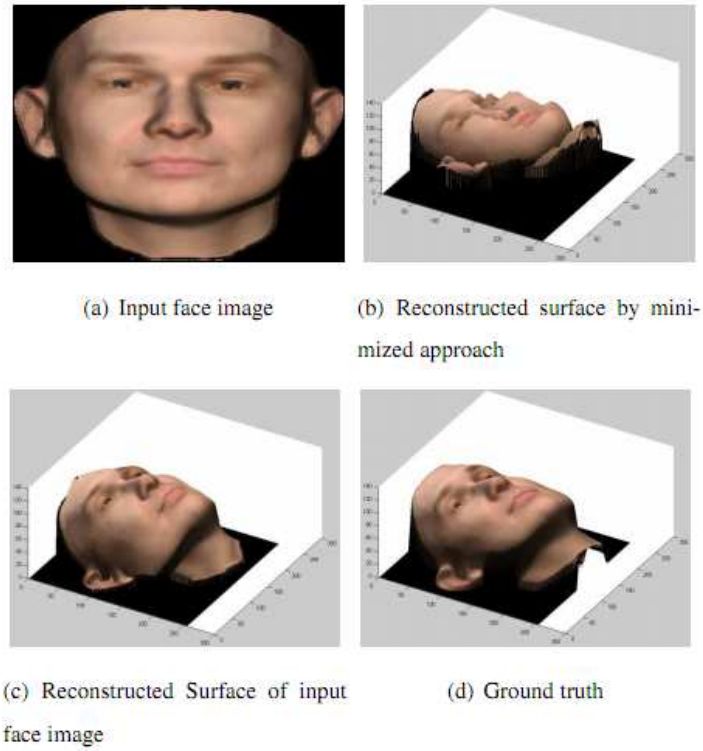
Figure 4.45: Reconstructed Surface and Ground truth of 014



(a) Input face image (b) Reconstructed surface by minimized approach

(c) Reconstructed Surface of input face image (d) Ground truth

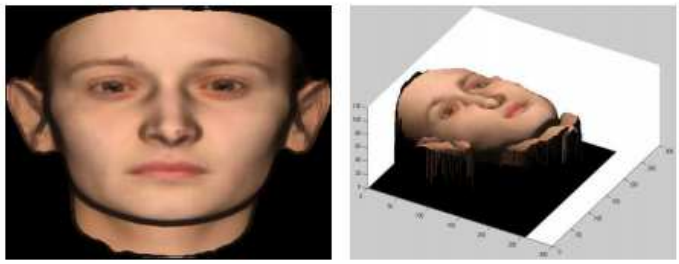
Figure 4.46: Reconstructed Surface and Ground truth of 017



(a) Input face image (b) Reconstructed surface by minimized approach

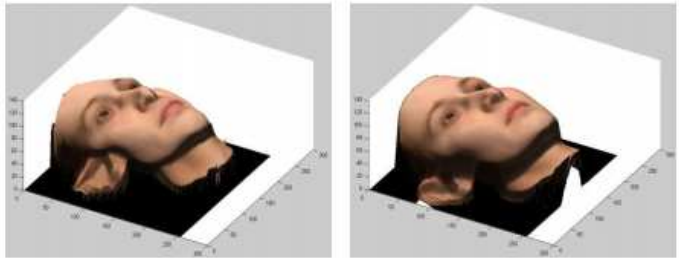
(c) Reconstructed Surface of input face image (d) Ground truth

Figure 4.47: Reconstructed Surface and Ground truth of 022



(a) Input face image

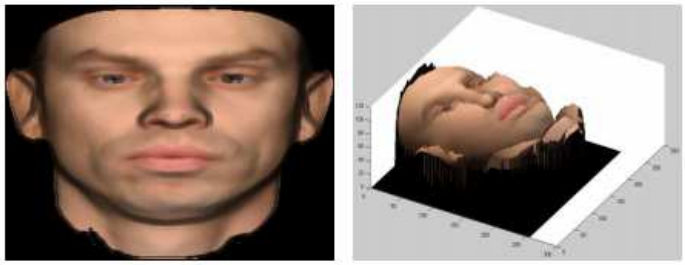
(b) Reconstructed surface by minimized approach



(c) Reconstructed Surface of input face image

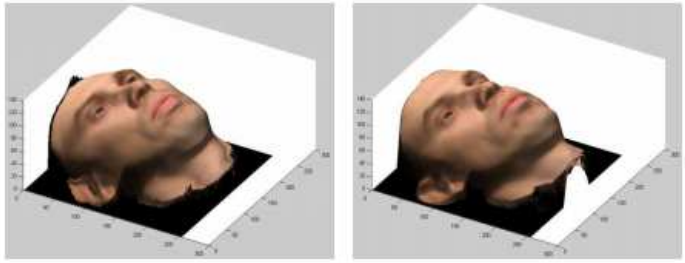
(d) Ground truth

Figure 4.48: Reconstructed Surface and Ground truth of 052



(a) Input face image

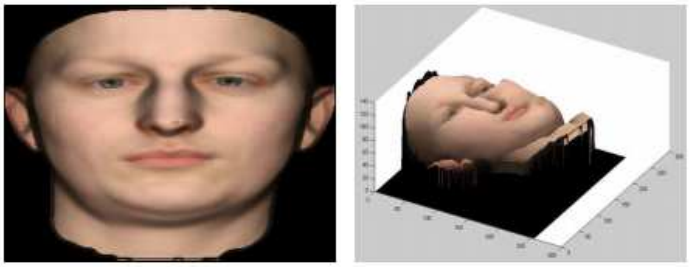
(b) Reconstructed surface by minimized approach



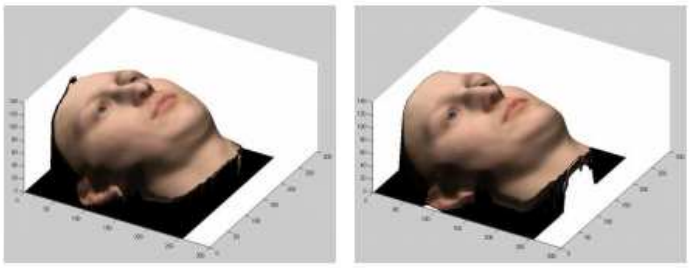
(c) Reconstructed Surface of input face image

(d) Ground truth

Figure 4.49: Reconstructed Surface and Ground truth of 053

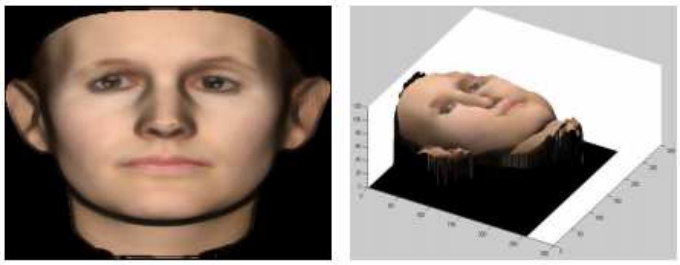


(a) Input face image (b) Reconstructed surface by minimized approach

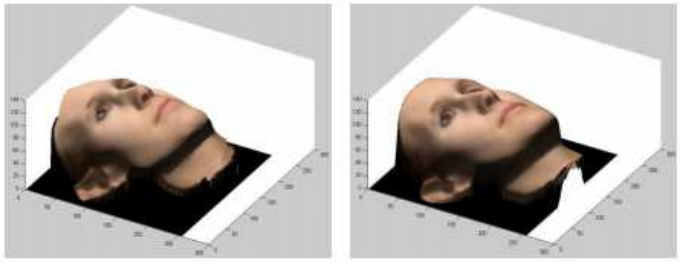


(c) Reconstructed Surface of input face image (d) Ground truth

Figure 4.50: Reconstructed Surface and Ground truth of 293



(a) Input face image (b) Reconstructed surface by minimized approach



(c) Reconstructed Surface of input face image (d) Ground truth

Figure 4.51: Reconstructed Surface and Ground truth of 323

The Height Difference Error of reconstructed surface by FR approach with normalized image and non-normalized are concluded in Table 4.8.

Table 4.8 Comparation of using normalized image and non-normalized image in my approach

| Image name | Normalized FR HDE (%) | Non-normalized FR HDE (%) |
|---|---|---|
| Barbara.png | 6.11 | 45.10 |
| isabelle.png | 3.52 | 44.23 |
| thomas.png | 4.58 | 44.82 |
| volker.png | 3.22 | 83.56 |
| 001.png | 3.99 | 31.81 |
| 002.png | 2.04 | 28.08 |
| 006.png | 4.77 | 26.36 |
| 014.png | 8.38 | 134.41 |
| 017.png | 5.35 | 31.39 |
| 022.png | 3.27 | 396.86 |
| 052.png | 3.71 | 30.17 |
| 053.png | 3.39 | 84.66 |
| 293.png | 3.63 | 28.23 |
| 323.png | 2.85 | 404.31 |
| Mean | 4.20 | 101.00 |

## 4.6 Experiment Analysis

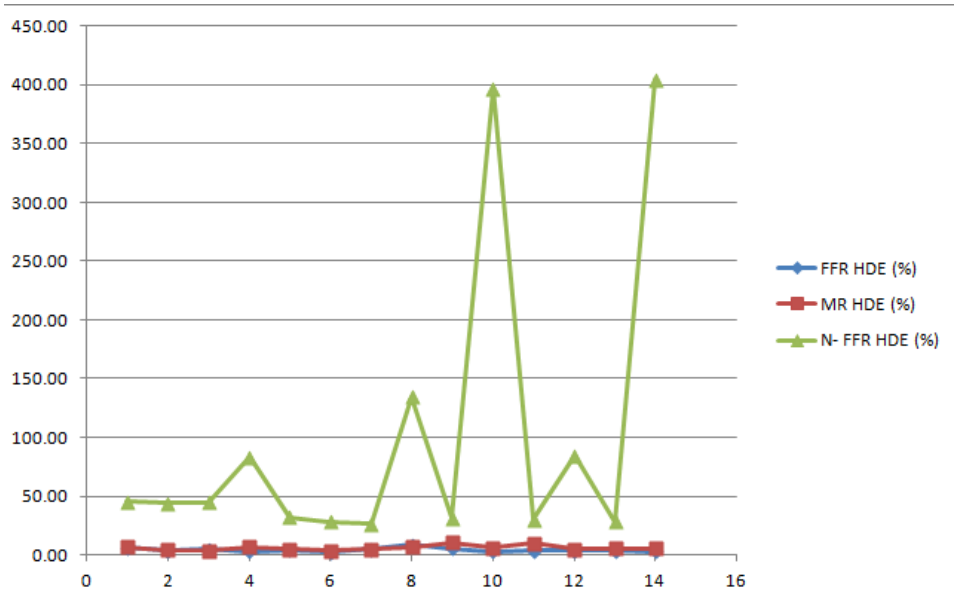
### 4.6.1 Height Different Error

From section 4.3, 4.4 and 4.5, the height different error (HDE) was retrieved and then the comparison can be shown in table 4.9.

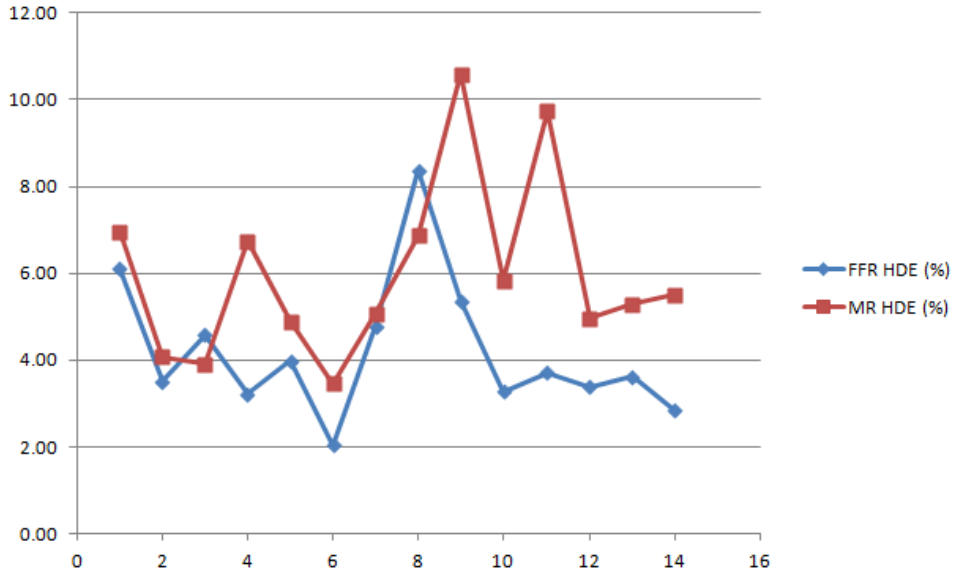
Table 4.9: Height different error comparison

| Image name | FR HDE (%) | MR HDE (%) | N-FR HDE (%) |
|---|---|---|---|
| Barbara.png | 6.11 | 6.96 | 45.10 |
| isabelle.png | 3.52 | 4.08 | 44.23 |
| thomas.png | 4.58 | 3.91 | 44.82 |
| volker.png | 3.22 | 6.74 | 83.56 |
| 001.png | 3.99 | 4.88 | 31.81 |
| 002.png | 2.04 | 3.45 | 28.08 |
| 006.png | 4.77 | 5.07 | 26.36 |
| 014.png | 8.38 | 6.86 | 134.41 |
| 017.png | 5.35 | 10.59 | 31.39 |
| 022.png | 3.27 | 5.83 | 396.86 |
| 052.png | 3.71 | 9.73 | 30.17 |
| 053.png | 3.39 | 4.97 | 84.66 |
| 293.png | 3.63 | 5.30 | 28.23 |
| 323.png | 2.85 | 5.52 | 404.31 |
| Mean | 4.20 | 5.99 | 101.00 |

FFR mean to my approach that is face reconstruction method, MR mean to face reconstruction method which use minimize-method to calculate normal surface and N-FFR mean to my approach without albedo estimation process. Then a graph in Figure. 4.52(a) was plotted and clearly show that without albedo estimation process is the worst case with very high error. To compare 2 remain method, a graph is Figure 4.52(b) was plotted and shown that my approach is the best case with less error.




(a) Without albedo estimation process




(b) albedo estimation process

Figure 4.52: Compare with Albedo estimation process

## 4.7 Normal Surface Calculation Time

From the result of processing time in section 4.3 and 4.4, the comparison of normal surface calculation time between my approach (FFR) and minimization approach (MR) was shown in table 4.10.

Table 4.10: Normal surface calculation time comparison

| Image name | FFR Normal surface calculation time (sec) | MR Normal surface calculation time (sec) |
|---|---|---|
| Barbara.png | 1.35 | 246.86 |
| isabelle.png | 1.30 | 324.32 |
| thomas.png | 1.26 | 341.42 |
| volker.png | 1.21 | 417.16 |
| 001.png | 1.28 | 249.68 |
| 002.png | 1.26 | 271.65 |
| 006.png | 1.28 | 494.61 |
| 014.png | 1.34 | 307.72 |
| 017.png | 1.28 | 304.12 |
| 022.png | 1.23 | 318.12 |
| 052.png | 1.25 | 975.93 |
| 053.png | 1.26 | 302.57 |
| 293.png | 1.30 | 318.84 |
| 323.png | 1.25 | 345.56 |
| Mean | 1.28 | 372.75 |

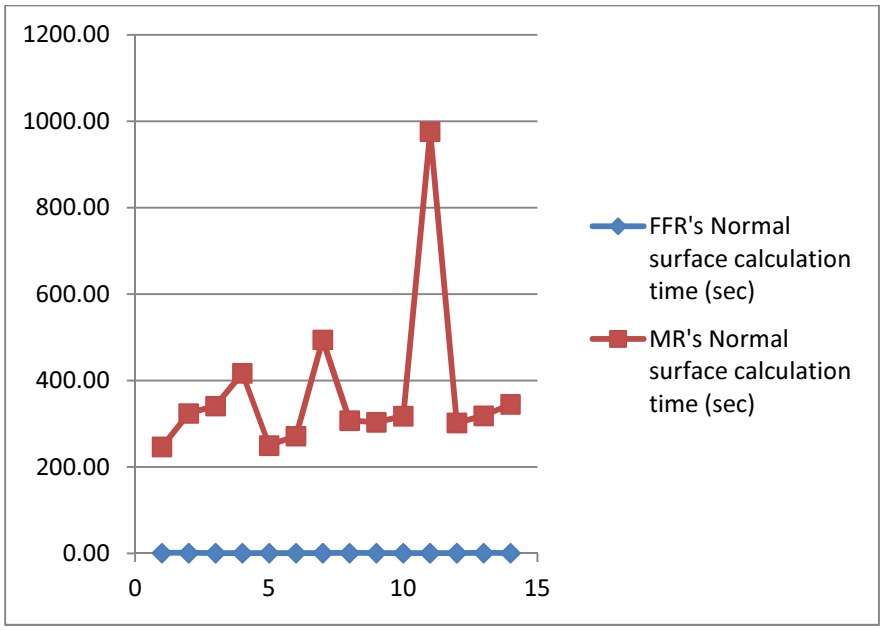



Figure 4.53: Normal surface calculation time comparision

## 4.8 Total Calculation Time

From the result of processing time in section 4.3 and 4.4, the comparison of total calculation time between my approach (FFR) and minimization approach (MR) was shown in table 4.11.

Table 4.12: Total calculation time comparison

| Image name | FFR total calculation time (sec) | MR total calculation time (sec) |
|---|---|---|
| Barbara.png | 31.20 | 276.76 |
| isabelle.png | 30.59 | 353.60 |
| thomas.png | 30.78 | 370.90 |
| volker.png | 29.41 | 445.27 |
| 001.png | 31.11 | 279.49 |
| 002.png | 31.16 | 301.51 |
| 006.png | 31.00 | 524.32 |
| 014.png | 32.50 | 338.92 |
| 017.png | 30.86 | 333.68 |
| 022.png | 30.29 | 347.11 |
| 052.png | 30.12 | 1004.75 |
| 053.png | 31.60 | 332.87 |
| 293.png | 31.69 | 349.23 |
| 323.png | 29.99 | 374.25 |
| Mean | 30.88 | 402.33 |

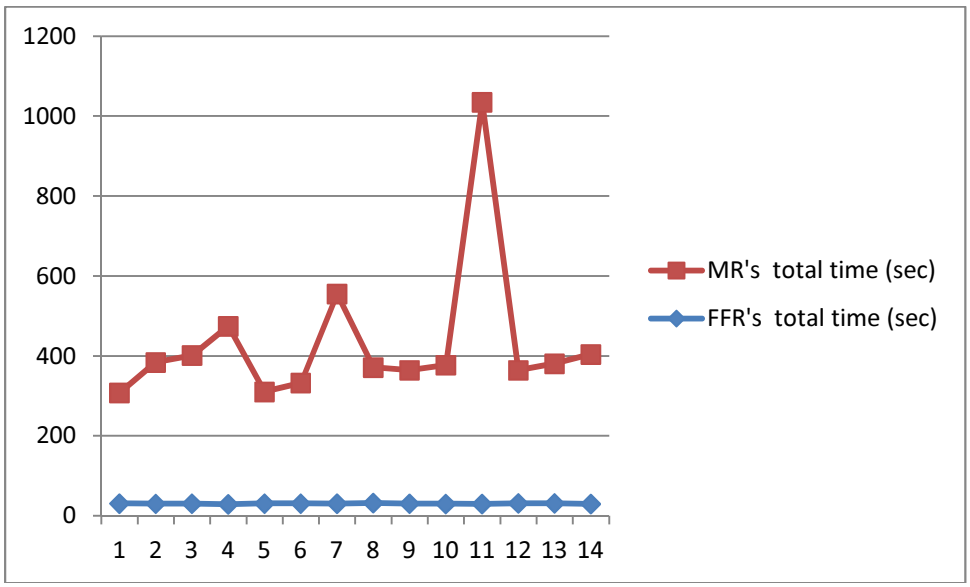



Figure 4.53: Total calculation time comparision

# CHAPTER V

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusions

In this research, the SFS problem has been formulated with a single image and condition of Lambertian model. Two object types have been considered in this work: (1) Smooth surface objects with unity reflectance ability and (2) Human faces with various reflectance abilities. In Lambertian model in equation 5.1, light is the first needed condition.

$$I = \rho N^T L \tag{5.1}$$

To obtain the light source direction vector (if unknown), is needs to assign initial normal vector for a few points (at least 3 points) and minimize the energy function from [48]. However, the human face images which were used in my experiment were obtained from face database of Max-Planck Institute for Biological Cybernetics [47], [49] and Basel face model from Basel University [50] whose the light source vector is known as 0,0,1.

For human face image reconstruction, the surface reflectance abilities is needed to be known that is called "Albedo". The results in section 4.5 show that human face has various albedo values in each point and face reconstruction will be done with lots of error if I use unity albedo to the entire surface. Then the albedo estimation process starts with initial normal surface from average face data, which is an average from 200 faces that obtained from a Face database of Max-Planck Institute for Biological Cybernetics in Tuebingen, Germany. After that the initial normal surface has been morph to match an input image by my pattern morphing method that is described in chapter 3 and then used as estimated normal surface. This estimated normal surface, Intensity map and Light source vector of input image are used to find albedo of each point on an input image by solving equation 5.1. Next, the real normal surface of an input image is calculated by my method of solving Orthogonality of Normal Surface equation that is described in chapter3. This method is easy to solve than minimization method as shown in section 4.4. The principle of this method is to find relation equation of Normal vector in $x$ and $y$ axes, which is described in

chapter 3, that is no need to use the slow method of minimization approach anymore. Finally the relative height (or different height) is calculated from gradient in $x$ and $y$ axes of normal surface by using height difference equation of Gavin Smith and Adrian Bors [45] and used it to calculate actual height by method of affine transformation on gradients [46].

## 5.2 Future works

Several important directions of further research is application of 3D reconstruction for human face recognition that can be the future work of this research.

# REFERENCE

[1] Horn, B. K., SHAPE FROM SHADING: A METHOD FOR OBTAINING THE SHAPE OF ASMOOTH OPAQUE OBJECT FROM ONE VIEW. Doctoral dissertation Massachusetts Institute of Technology, 1970.

[2] Zhang, R., Tsai, P.-S., Cryer, J., and Shah, M. Shape-from-shading: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (aug 1999): 690 -706.

[3] Wilhelmy, J. and Kr ¨ uger, J. Shape from Shading Using Probability Functions and Belief Propagation. Int. J. Comput. Vision 84 (Sept. 2009): 269–287.

[4] Ikeuchi, K. and Horn, B. K. P., Numerical Shape from Shading and Occluding Boundaries., 1981.

[5] Zheng, Q. and Chellappa, R. Estimation of Illuminant Direction, Albedo, and Shape from Shading. IEEE Trans. Pattern Anal. Mach. Intell. 13 (July 1991): 680–702.

[6] Horn, B. K. P. and Brooks, M. J. The variational approach to shape from shading. Computer Vision, Graphics, and Image Processing 33 , 2 (1986): 174–208.

[7] Frankot, R. T., Chellappa, R., and Member, S. A Method for enforcing integrability in shape from shading algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (1988): 439–451.

[8] Horn, B. K. P. Height and gradient from shading. International Journal of Computer Vision 5 , 1(1990): 37–75.

[9] Brooks, M. J. and Horn, B. K. P., Shape and source from shading. Proceedings of the 9th interna-tional joint conference on Artificial intelligence - Volume 2, IJCAI'85, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., (1985): 932–936.

[10] Harrison, A. and Joseph, D. Maximum Likelihood Estimation of Depth Maps using Photometric Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence PP , 99 (2011):1.

[11] Smith, W. and Hancock, E. Recovering Facial Shape Using a Statistical Model of Surface Normal Direction. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (dec.2006): 1914 -1930.

[12] Atkinson, G. and Hancock, E. Shape Estimation Using Polarization and Shading from Two Views. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (nov. 2007): 2001-2017.

[13] Zeng, G., Paris, S., Quan, L., and Sillion, F. Accurate and Scalable Surface Representation andReconstruction from Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (jan. 2007): 141 -158.

[14] Biswas, S., Aggarwal, G., and Chellappa, R. Robust Estimation of Albedo for Illumination-Invariant Matching and Shape Recovery. IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009): 884-899.

[15] Chen, D. and Dong, F., Shape from Shading Using Wavelets. Proceedings of the Second International Workshop on Semantic Media Adaptation and Personalization, SMAP '07, Washington, DC, USA, IEEE Computer Society, (2007): 86–91.

[16] Rouy, E. and Tourin, A. A Viscosity Solutions Approach to Shape-From-Shading. SIAM Journal on Numerical Analysis 29 , 3 (1992): 867–884.

[17] Oliensis, J. and Dupuis, P. Direct method for reconstructing shape from shading. 1570 (1991): 116-128.

[18] Bichsel, M. and Pentland, A. P. A simple algorithm for shape from shading. Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on (1992): 459–465.

[19] Sethian, J. A. Fast Marching Methods. SIAM Rev. 41 (June 1999): 199–235.

[20] Kimmel, R. and Sethian, J. A. Optimal Algorithm for Shape from Shading and Path Planning. J.Math. Imaging Vis. 14 (May 2001): 237–244.

[21] Tankus, A., Sochen, N., and Yeshurun, Y. Shape-from-Shading Under Perspective Projection. Int. J.Comput. Vision 63 (June 2005): 21–43.

[22] Baek, S.-Y., Kim, B.-Y., and Lee, K., 3D face model reconstruction from single 2D frontal image. Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry, VRCAI '09, New York, NY, USA, ACM, (2009): 95–101.

[23] Horovitz, I. and Kiryati, N. Depth from Gradient Fields and Control Points: Bias Correction in Photometric Stereo. Image and Vision Computing 22 (2004): 681–694.

[24] Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., and Seidel, H.-P., Multi-level partition of unity implicits. ACM SIGGRAPH 2003 Papers, SIGGRAPH '03, New York, NY, USA, ACM,(2003): 463–470.

[25] Lee, C.-H. and Rosenfeld, A. Improved methods of estimating shape from shading using the light source coordinate system. Artif. Intell. 26 (May 1985): 125–143.

[26] Wu, Z. and Li, L. A line-integration based method for depth recovery from surface normals. Comput.Vision Graph. Image Process. 43 (July 1988): 53–66.

[27] Pentland, A. Shape Information From Shading:A Theory About Human Perception. Computer Vision., Second International Conference on (1988): 404–413.

[28] Tsai, P. and Shah, M. Shape From Shading Using Linear Approximation. Image and Vision Computing 12.

[29] Sun, Z.-L. and Lam, K.-M. Depth Estimation of Face Images Based on the Constrained ICA Model. IEEE Transactions on Information Forensics and Security 6 (june 2011): 360 -370.

[30] Petrovic, N., Cohen, I., Frey, B. J., Koetter, R., and Huang, T. S., Enforcing Integrability for Surface Reconstruction Algorithms Using Belief Propagation in Graphical Models. In: Proc.Conf. Computer Vision and Pattern Recognition, (2001): 743–748.

[31]Potetz,B., Ecient Belief Propagation for Vision Using Linear Constraint Nodes. 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA, IEEE Computer Society.

[32] Woodham, R. J., Shape from shading. ch. Photometric method for determining surface orientation from multiple images, pp. 513–531, Cambridge, MA, USA : MIT Press, 1989.

[33] Onn, R. and Bruckstein, A. Integrability disambiguates surface recovery in two-image photometric stereo. Int. J. Comput. Vision 5 (Sept. 1990): 105–.

[34] Fan, J. and Wolff, L. B. Surface curvature and shape reconstruction from unknown multiple illumination and integrability. Comput. Vis. Image Underst. 65 (Feb. 1997): 347–359.

[35] Jin, H., Cremers, D., Wang, D., Prados, E., Yezzi, A., and Soatto, S. 3-D Reconstruction of Shaded Objects from Multiple Images Under Unknown Illumination. Int. J. Comput. Vision 76(Mar. 2008): 245–256.

[36] Wu, T. and Tang, C., Dense Photometric Stereo Using a Mirror Sphere and Graph Cut. (2005): I: 140-147.

[37] Srichumroenrattana, N., Lursinsap, C., and Lipikorn, R., 2D Face image depth ordering using adaptive Hillcrest-Valley classification and Otsu. 2010 IEEE 10th International Conference on Signal Processing (ICSP) (oct. 2010): 645-648.

[38] Sonka, M., Hlavác, V., and Boyle, R. Image processing, analysis and and machine vision (3. ed.).Thomson, 2008.

[39] Axelsson, P. Processing of laser scanner data – algorithms and applications. Journal Of Photogrammetry And Remote Sensing 54 , 2-3 (1999): 138–147.

[40] Oren, M. and Nayar, S. K., Generalization of Lambert's refiectance model. Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94, New York, NY, USA, ACM, (1994): 239–246.

[41] Szeliski, R. Computer vision. Algorithms and applications. London: Springer, 2011.

[42]Coakley, J., REFLECTANCE AND ALBEDO, SURFACE. in Encyclopedia of Atmospheric Sciences (Chief: James R. Holton, E., ed.), pp. 1914 – 1923, Oxford : Academic Press,2003.

[43] Kukanok, S., Genetic Algorithm with Adaptive Operators. Ph.D. dissertation King Mongkut's University of Technology North Bangkok, 2550.

[44] Chambers, L. D. The Practical Handbook of Genetic Algorithms: Applications, Second Edition. Boca Raton, FL, USA : CRC Press, Inc., 2nd ed., 2000.

[45] Smith, G. D. J. and Bors, A. G., Height estimation from vector fields of surface normals. 2 (July 2002): 1031–1034.

[46] Agrawal, A., Raskar, R., and Chellappa, R., What is the range of surface reconstructions from a gradient field?. Proceedings of the 9th European conference on Computer Vision - Vol-ume Part I, ECCV'06, Berlin, Heidelberg, Springer-Verlag, (2006): 578–591.

[47] Blanz, V. and Vetter, T., A morphable model for the synthesis of 3D faces. Proceedings of

the 26thannual conference on Computer graphics and interactive techniques, SIGGRAPH '99, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co., (1999): 187–194.

[48] Wu, T.-P., Sun, J., Tang, C.-K., and Shum, H.-Y. Interactive normal reconstruction from a single image. ACM Trans. Graph. 27 , 5 (2008): 1–9.

[49] Troje, N. F. and Bültho, H. H. Face recognition under varying poses: The role of texture and shape.Vision Research 36 (1996): 1761–1771.

[50] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T., A 3D Face Model for Pose and Illumination Invariant Face Recognition. AVSS '09. Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (sept. 2009): 296 -301.

# Biography

**Name** : Miss Natchamol Srichumroenrattana

**Education:**

- Ph.D. Program in Computer Science, Chulalongkorn University, Thailand (October 2007 - September 2012).

**Publication:**

- Srichumroenrattana, N., Lursinsap, C., and Lipikorn, R., 2D Face image depth ordering using adap- tive Hillcrest-Valley classification and Otsu. 2010 IEEE 10th International Conference on Signal Processing (ICSP) (oct. 2010): 645-648.
- Srichumroenrattana, N., Lursinsap, C., and Lipikorn, R., Facial Feature Detection Using Multires- olution Decomposition and Hillcrest-Valley Classification with Adaptive Mean Filter. Computer Sciences and Convergence Information Technology, 2009. ICCIT 09. Fourth International Conference on, nov. 2009.