University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

8-2011

3D facial shape estimation from a single image under arbitrary pose and illumination.

Ham Rara 1981-
University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

Recommended Citation

Rara, Ham 1981-, "3D facial shape estimation from a single image under arbitrary pose and illumination." (2011). *Electronic Theses and Dissertations.* Paper 1181.
https://doi.org/10.18297/etd/1181

This Master's Thesis is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

3D FACIAL SHAPE ESTIMATION FROM A SINGLE IMAGE UNDER ARBITRARY POSE AND ILLUMINATION

By

Ham Rara
M.Sc. 2006, EE, University of Louisville

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering
University of Louisville
Louisville, Kentucky

August 2011

3D FACIAL SHAPE ESTIMATION FROM A SINGLE IMAGE UNDER ARBITRARY POSE AND ILLUMINATION

By

Ham Rara
M.Sc., 2006, EE, University of Louisville

A Dissertation Approved on

July 11, 2011

By the following Reading and Examination Committee:

Aly A. Farag, Ph.D., Dissertation Director

Thomas L. Starr, Ph.D.

Dar-Jen Chang, Ph.D.

John F. Naber, Ph.D.

Hossam Eldin H Abd El Munim, Ph.D.

DEDICATION

To my dearest wife, Reza, and my son, Zachary

ACKNOWLEDGMENTS

First of all, my deepest thanks are due to the Almighty God, for the uncountable blessings given to me.

I would like to express my deepest gratitude to my advisor, Prof. Aly A. Farag, for giving me the opportunity to be a member in his research group, for his continuous encouragement, and for his support over the course of this work. He provided a very rich working environment with many opportunities to develop new ideas, work in promising applications, get experience in diverse areas, and meet well-known people in the field.

I would like to thank Dean Thomas L. Starr for giving me the opportunity to be a member of his research group in the face recognition at-a-distance (FRAD) project, and for useful discussions.

I would like to thank Dr. John Naber, Dr. Dar-Jen Chang, and Dr. Hossam Eldin H Abd El Munim for agreeing to be on my dissertation committee, for the useful consultation and the fruitful discussions.

I would like to thank Ms. Shireen Elhabian for useful discussions and her assistance in publishing in respected conferences. She has never hesitated to share her expertise in various topics related to computer vision and image processing. She has been a good teammate and a great friend.

I would like to thank all the members of Computer Vision and Image Processing Laboratory at University of Louisville, both past and present. Special thanks to Mr. Mike Miller for his continuous dedication to help and for his support in hard times. Also, I would like to thank Mr. Chuck Sites for his valuable technical help during the work in the lab projects.

Very special thanks to my family for their encouragement and support, without which this dissertation and research would not have been possible. My deepest gratitude to my parents, brothers, sister, and son Zachary. Finally, words cannot describe how I am indebted to my wife for her pain, suffering and sacrifices made during the journey of this study.

ABSTRACT

3D FACIAL SHAPE ESTIMATION FROM A SINGLE IMAGE UNDER ARBITRARY POSE AND ILLUMINATION

Ham Moso Rara

June 27, 2011

Humans have the uncanny ability to perceive the world in three dimensions (3D), otherwise known as depth perception. The amazing thing about this ability to determine distances is that it depends only on a simple two-dimensional (2D) image in the retina. It is an interesting problem to explain and mimic this phenomenon of getting a three-dimensional perception of a scene from a flat 2D image of the retina. The main objective of this dissertation is the computational aspect of this human ability to reconstruct the world in 3D using only 2D images from the retina.

Specifically, the goal of this work is to recover 3D facial shape information from a single image of unknown pose and illumination. Prior shape and texture models from real data, which are metric in nature, are incorporated into the 3D shape recovery framework. The output recovered shape, likewise, is metric, unlike previous shape-from-shading (SFS) approaches that only provide relative shape.

This work starts first with the simpler case of general illumination and fixed frontal pose. Three optimization approaches were developed to solve this 3D shape recovery problem, starting from a brute-force iterative approach to a computationally efficient regression method (Method II-PCR), where the classical shape-from-shading equation is cast as a regression framework. Results show that the output of the regression-like approach is faster in timing and similar in error metrics when compared to its iterative counterpart.

The best of the three algorithms above, Method II-PCR, is compared to its two predecessors, namely: (a) Castelan et al. [1] and (b) Ahmed et al. [2]. Experimental results show that the proposed method (Method II-PCR) is superior in all aspects compared to the previous state-of-the-art. Robust statistics was also incorporated into the shape recovery framework to deal with noise and occlusion.

Using multiple-view geometry concepts [3], the fixed frontal pose was relaxed to arbitrary pose. The best of the three algorithms above, Method II-PCR, once again is used as the primary 3D shape recovery method. Results show that the pose-invariant 3D shape recovery version (for input with pose) has similar error values compared to the frontal-pose version (for input with frontal pose), for input images of the same subject. Sensitivity experiments indicate that the proposed method is, indeed, invariant to pose, at least for the pan angle range of ($-50°$ to $50°$).

The next major part of this work is the development of 3D facial shape recovery methods, given only the input 2D shape information, instead of both texture and 2D shape. The simpler case of output 3D sparse shapes was dealt with, initially. The proposed method, which also use a regression-based optimization approach, was compared with state-of-the art algorithms, showing decent performance. There were five conclusions that drawn from the sparse experiments, namely, the proposed approach: (a) is competitive due to its linear and non-iterative nature, (b) does not need explicit training, as opposed to [4], (c) has comparable results to [4], at a shorter computational time, (d) better in all aspects than Zhang and Samaras [5], and (e) has the limitation, together with [4] and [5], in terms of the need to manually annotate the input 2D feature points.

The proposed method was then extended to output 3D dense shapes simply by replacing the sparse model with its dense equivalent, in the regression framework inside the 3D face recovery approach. The numerical values of the mean height and surface orientation error indicate that even if shading information is unavailable, a decent 3D dense reconstruction is still possible.

TABLE OF CONTENTS

DEDICATION . . . iii
ACKNOWLEDGMENTS . . . iv
ABSTRACT . . . vi
TABLE OF CONTENTS . . . ix
LIST OF TABLES . . . xii
LIST OF FIGURES . . . xiv
LIST OF ALGORITHMS . . . xxvi

CHAPTER
I. INTRODUCTION . . . 1
- A. Pyschology of Depth Perception . . . 2
  - 1. Motion-based Cues . . . 3
  - 2. Pictorial Cues . . . 3
- B. Shape-from-X . . . 7
  - 1. Image Formation . . . 8
  - 2. Shape-from-Stereo . . . 8
  - 3. Shape-from-Shading (SFS) . . . 10
- C. Related Methods . . . 13
  - 1. Statistical Model-based Approaches . . . 13
- D. Why This Work Is Needed . . . 16
- E. Dissertation Contributions . . . 16
- F. Document Layout . . . 18

II. MODELING IMAGES OF OBJECTS UNDER VARIABLE ILLUMINATION USING SPHERICAL HARMONICS . . . 20
- A. Lambertian Model . . . 20
- B. Spherical Harmonic (SH) Analysis . . . 21
  - 1. Spherical Harmonics Projection . . . 23
  - 2. Spherical Harmonics Reconstruction . . . 23
  - 3. SH Analysis of Lambertian Reflectance and Light Source Function . . . 24
  - 4. Spherical Harmonic Projection (SHP) Images . . . 28
- C. Robust Spherical Harmonics Projection (SHP) Images . . . 30
  - 1. Robust Estimation . . . 32
  - 2. Robust Estimation Framework for Spherical Harmonic Projection (SHP) Images . . . 34

3. Simultaneous Over-Relaxation 35
4. Graduated Non-Convexity 35

D. Robust SHP Image Experiments 37
1. Results from Least-Squares Estimation 38
2. Results from Robust Estimation 40

E. Summary 44

III. MODEL-BASED SHAPE RECOVERY FROM SINGLE IMAGES OF GENERAL AND UNKNOWN LIGHTING AND POSE 46

A. Model Construction 46
1. USF Database of 3D Scans 47
2. Dense Correspondence 51
3. Statistical Modeling 52

B. Spherical Harmonics Illumination 54

C. Shape-from-shading 55
1. Brightness Constraint with Spherical Harmonics 56

D. Model-based 3D Face Shape Recovery: An Iterative Approach (Frontal Pose) 58

E. Model-based 3D Face Recovery: A Coupled Statistical Model Approach (Frontal Pose) 61
1. Coupled Statistical Models 62
2. 3D Face/Albedo Recovery 63
   a. **Method I** 63
   b. **Method II** 66

F. Comparison of Model-based SFS Methods 67
1. Algorithmic Comparison 69
2. Experimental Results Comparison 71
3. Related Algorithms Comparison 73

G. Robust Model-based Shape-from-shading Framework 82
1. Experimental Results 85

H. Model-based 3D Face Recovery: A Coupled Statistical Model Approach (General Illumination and Unknown Pose) 95
1. Preprocessing and Postprocessing Steps 95
2. Experimental Results 98

I. Summary 101

IV. MODEL-BASED SHAPE RECOVERY FROM SINGLE IMAGES OF UNKNOWN POSE USING A SMALL NUMBER OF FEATURE POINTS109

A. Basic Definitions and Notations 109
1. Experimental Results 111

B. Related Methods 111
1. Iterative Methods 113
2. Linear Methods 114

C. Proposed Approach 116
  1. Preliminaries 117
  2. Model-based Framework (Case I: Output 3D Sparse Shape) 117
  3. Experimental Results 120
  4. Model-based Framework (Case II: Output 3D Dense Shape) 124
  5. Experimental Results 125

D. Conclusion 128

V. CONCLUSIONS AND FUTURE DIRECTIONS 141

A. Future Directions 142

REFERENCES 143

APPENDIX

I. 2D/3D Deformable Models 149

A. Statistical Shape/Appearance Models: AAM 149
  1. Model Construction 150
  2. Combined Shape/Appearance Models (Combined AAM) . 151
  3. Independent Shape and Appearance Models (Independent AAM) 151
  4. AAM Model Search (Fitting) 152
  5. Combined Model 152
  6. Independent AAM Model: 153

B. 3D Morphable Models (3DMM) 154
  1. Face Vectors 155
  2. Image Synthesis 157
  3. Fitting the 3D Morphable Model to an Image 158

CURRICULUM VITAE 159

A. CONTACT INFORMATION 159

B. RESEARCH INTERESTS 159

C. EDUCATION 159

D. HONORS AND AWARDS 160

E. PUBLICATIONS 160

F. PROFESSIONAL ACTIVITIES 161

G. SOFTWARE PROGRAMMING 161

H. LANGUAGES 161

I. MEMBERSHIP 162

LIST OF TABLES

TABLE PAGE

1. Numerical Values for Reconstruction Errors with Non-Groundtruth SH Basis Images 42
2. Average Computational Times of Model-based 3D Face Recovery Approaches 71
3. Mean and standard deviation of the mean height error for 80 out-of-training USF samples 72
4. Mean and standard deviation of the mean surface orientation error for 80 out-of-training USF samples 73
5. Related Algorithms Comparison: Mean and standard deviation of the mean height error for 80 out-of-training USF samples 74
6. Related Algorithms Comparison: Mean and standard deviation of the mean surface orientation error for 80 out-of-training USF samples 74
7. Circular Occlusion Experiments: Mean and standard deviation of the mean height error for 80 out-of-training USF samples 87
8. Circular Occlusion Experiments: Mean and standard deviation of the mean surface orientation error for 80 out-of-training USF samples 88
9. Contiguous Occlusion Experiments: Mean and standard deviation of the mean height error for 80 out-of-training USF samples 89
10. Contiguous Occlusion Experiments: Mean and standard deviation of the mean surface orientation error for 80 out-of-training USF samples 89
11. Unknown Pose Experiments: Mean and standard deviation of the mean height error for 80 out-of-training USF samples 103
12. Unknown Pose Experiments: Mean and standard deviation of the mean surface orientation error for 80 out-of-training USF samples . 103

13. Sparse-to-Dense Experiments: Mean and standard deviation of the mean height error for 80 out-of-training USF samples 126

14. Sparse-to-Dense Experiments: Mean and standard deviation of the mean surface orientation error for 80 out-of-training USF samples . 127

LIST OF FIGURES

FIGURE PAGE

1. 3D facial shape estimation problem under general illumination and pose. The goal is to recover 3D facial shape from images of unknown lighting and pose conditions, using a 3D estimation black box. 2

2. The mountains in the background [6], partially hidden by the barn, are perceived to be farther away than the latter due to a type of pictorial cue called occlusion. 4

3. The painting (Paris Street, Rainy Day) by Gustave Caillebotte [7] illustrates the cue of texture gradient, where elements that are equally spaced in a closer scene appear to be more closely packed as distance increases. 5

4. A fogged roadway [8] illustrating the cue of atmospheric perspective. With fog interfering with the view, the more distant car appears less sharp. 6

5. A painting (The Lacemaker) by Johannes Vermeer [9], where the cues of lighting and shading are used to illustrate depth. 6

6. Distant light source and Lambertian surface illustration. The intensity of a surface point (with albedo λ) depends on the angle between the surface normal $\mathbf{n}$ and light source direction $\mathbf{l}$, no matter what the viewing angle is. 9

7. A simple stereo system illustrating how depth can be extracted given the binocular disparity. The resulting relationship shows that depth is inversely proportional to disparity. 9

8. Shape-from-shading problem 11

9. Shape-from-shading results of Ahmed and Farag [10] [11] on an illuminated image of a human face. 12

10. Recovered 3D shape using Castelan et al. [1] on an input with frontal pose and unknown illumination. 14

11. Recovered shapes, together with the input image and ground-truth (GT) shape, using the proposed method in the next chapters. Reconstruction is performed in terms of both albedo (defined as the scale that connects the image irradiance (E) and intensity (I), i.e., $I = \lambda E$) and shape. From a visual comparative perspective, the reconstructions are close to its groundtruth (GT) counterparts. . . . 17

12. Lambertian surface illustration. The intensity of a surface point (with albedo λ) depends on the angle between the surface normal $\mathbf{n}$ and light source direction $\mathbf{l}$, no matter what the viewing angle is. . 21

13. Illustration of the first four spherical harmonic bands, $l = 0, \cdots, 3$. Green and red parts indicate positive and negative values, respectively. 22

14. Sample spherical function taken from [12], displayed as a spherical plot. The actual function definition is $f(\theta, \phi) = \max(0, 5\cos(\theta) - 4) + \max(0, -4\sin(\theta - \pi) * \cos(\phi - 2.5) - 3)$. 24

15. Reconstruction of the spherical function at Fig. 14 using the first 10, 15, and 25 SH coefficients. Notice that the reconstruction using 25 coefficients is visually close enough to the original function. 24

16. Input image corruped with (a) salt & pepper noise and (b) occlusion. 26

17. Visualization of first nine harmonic images. Given the albedo, λ, and surface normals (derived from the 3D shape), $\mathbf{n} = (n_x, n_y, n_z)$, of a certain subject, the nine harmonic images can be computed easily using (19). The harmonic images are arranged in a pyramidal form similar to Figure 13. 27

18. Visualization of first nine harmonic images of various subjects from the USF Database [13]. Given the albedo, λ, and surface normals (derived from the 3D shape), $\mathbf{n} = (n_x, n_y, n_z)$, of a certain subject, the nine harmonic images can be computed easily using (19). The harmonic images are arranged in a pyramidal form similar to Figure 13. 28

19. Spherical Harmonics Projection (SHP) Images Illustration I: Input image is reconstructed by projecting it first to the nine harmonic basis images (i.e., solve for the coefficients) and then taking the reverse process by summing the scaled basis images (using the computed coefficients). The reconstructed image (now called SHP image) is visually similar to the input image. 30

20. Spherical Harmonics Projection (SHP) Images Illustration II: The pipeline in Figure 19 is extended to all subjects of the USF database. Notice that the SHP images encode the illumination of the input but retains the identity of the subject involved. 31

21. Input image corruped with (a) salt & pepper noise and (b) occlusion. 31

22. Quadratic ρ and ψ functions: (a) $\rho(x) = x^2$ and (b) $\psi(x) = 2x$. 32

23. Visualization of the truncated quadratic (a) ρ and (b) ψ functions. . . 32

24. Geman-McClure ρ and ψ functions: (a) $\rho(x) = \frac{x^2}{(\sigma+x^2)}$ and (b) $\psi(x) = \frac{2x\sigma}{(\sigma+x^2)^2}$. 33

25. Lorentzian ρ and ψ functions: (a) $\rho(x) = \log(1 + \frac{1}{2}(\frac{x}{\sigma})^2)$ and (b) $\psi(x) = \frac{2x}{2\sigma^2+x^2}$. 34

26. Graduated Non-Convexity. Geman-McClure $\rho(x,\sigma)$ and $\psi(x,\sigma)$ plotted for thresholds $\tau\epsilon\{16,8,4,2,1\}$. (a) Error measure, $\rho(x,\sigma)$. (b) Influence, $\psi(x,\sigma)$. 37

27. Graduated Non-Convexity. Lorentzian $\rho(x,\sigma)$ and $\psi(x,\sigma)$ plotted for thresholds $\tau\epsilon\{16,8,4,2,1\}$. (a) Error measure, $\rho(x,\sigma)$. (b) Influence, $\psi(x,\sigma)$. 37

28. Least-square (LS) estimation results for the reconstruction of Fig. 21(b) using groundtruth SH basis images. (a) Input image, I_g. (b) Occluded version of input image, I_{gc}. (c) LS reconstruction of the input, $\tilde{I}_g$ (d) LS reconstruction of the occluded input, $\tilde{I}_{gc}$. 40

29. Robust estimation (Geman-McClure) results for the reconstruction of Figure 21(b) using groundtruth SH basis images. (a) Input image, I_g. (b) Occluded version of input image, I_{gc}. (c) LS reconstruction of the input, $\tilde{I}_g$ (d) Robust reconstruction (Geman-McClure) of the occluded input, $\tilde{I}_{gc}$. 41

30. Occlusion estimate of Algorithm **2** (Step 6) using the Geman-McClure ρ-function, expressed as a binary image. White pixels indicate outliers. 41

31. Robust estimation (Lorentzian) results for the reconstruction of Fig. 21(b) using groundtruth SH basis images. (a) Input image, I_g. (b) Occluded version of input image, I_{gc}. (c) LS reconstruction of the input, $\tilde{I}_g$ (d) Robust reconstruction (Lorentzian) of the occluded input, $\tilde{I}_{gc}$. 41

32. Occlusion estimate of Algorithm **2** (Step 6) using the Lorentzian ρ-function, expressed as a binary image. White pixels indicate outliers. 41

33. Reconstruction results when basis images are computed from other subjects' shape and albedo. First and second rows contain the original input and its occluded version, respectively. Albedo and shape of various subjects are found in the third and fourth rows. Fifth and sixth rows show the reconstruction of the input image and its occluded version using the least-squares estimate. Seventh row shows the reconstruction of the occluded input using robust estimation. Last row shows the occlusion estimate, with white pixels indicating outliers. 43

34. Sample 3D laser scan parameterized by cylindrical coordinates, (h, ϕ). 48

35. Inverse mapping to infer pixel values in the image frame using barycentric coordinates. 50

36. Sample texture (top row) and height maps (bottom row) from the original USF data. 50

37. Sample annotations of the USF database. There is a total of 76 manually annotated landmark points, including both anatomical and pseudo-landmarks. 51

38. (**Upper Left**) Mean shape together with its convex hull. The convex hull forms the boundary of the mask. (**Lower**) Each pixel position within the mask corresponds to a certain point on a sample scan in the USF database. The vector $\mathbf{I}_{i,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$ refers to the shape and texture information at the *k*th vertex of the *i*th sample scan. (**Upper Right**)Similarly, each pixel position within the mask corresponds to a certain point on the input image. Hence, there is correspondence between the USF database samples and the input image, through the mean shape mask. 53

39. **(Top row)** $\bar{\mathbf{t}}$, $\bar{\mathbf{t}} - 0.5\beta_1\mathbf{t}_1$, $\bar{\mathbf{t}} - 0.5\beta_2\mathbf{t}_2$, $\bar{\mathbf{t}} - 0.5\beta_1\mathbf{t}_1 - 0.5\beta_2\mathbf{t}_2$, $\bar{\mathbf{t}} + 0.5\beta_1\mathbf{s}_1 + 0.5\beta_2\mathbf{s}_2$. **(Bottom row)** $\bar{\mathbf{s}}$, $\bar{\mathbf{s}} - 0.5\sigma_1\mathbf{s}_1$, $\bar{\mathbf{s}} - 0.5\sigma_2\mathbf{s}_2$, $\bar{\mathbf{s}} - 0.5\sigma_1\mathbf{s}_1 - 0.5\sigma_2\mathbf{s}_2$, $\bar{\mathbf{s}} + 0.5\sigma_1\mathbf{s}_1 + 0.5\sigma_2\mathbf{s}_2$. 54

40. (a) Image synthesized from $Z(x, y, h_t) = h_t\sqrt{1 - (\frac{x^2}{a^2} + \frac{y^2}{b^2})}$ with $a = b = 32, h_t = 64$, and light source direction $l = (0, 0, 1)$. (b) Nine basis images of $Z(x, y)$ with $\lambda = 1$. 57

41. Plot of $\varepsilon(h_t)$ and $\frac{d\varepsilon(h_t)}{dh_t}$ for some values of h_t. Note that $\varepsilon(h_t)$ approaches the minimum as $h_t = 64$ and the direction of $-\frac{d\varepsilon(h_t)}{dh_t}$ points toward the minimum. Using a simple gradient descent algorithm, the value of h_t such that $\varepsilon(h_t)$ is minimum can be found in a few iterations. 58

42. Coupled statistical model illustration. A coupled statistical model can be constructed from this figure, which links coefficients of the intensity, shape and spherical harmonics projection (SHP) images of faces of various subjects. 63

43. The coupled model illustration in Fig. 42 can be decomposed into two parts, which can then be cast as a regression framework. There are two regression models at work here, namely: (a) SHP-to-shape and (b) SHP-to-texture. 66

44. (a) Multivariate multiple linear regression: A model ($\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$) is directly fitted between the independent data (**X**) and dependent data (**Y**). (b) Principal component regression: Instead of directly fitting between **X** and **Y**, they are first transformed to a low-dimensional subspace, forming **T** and **U**, e.g. $\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}_0, \mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F}_0$, where **P** and **Q** are eigenvectors. Actual multiple linear regression ($\mathbf{U} = \mathbf{T}\mathbf{C} + \mathbf{F}$) is done between **T** and **U**. 69

45. Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Most samples fall under the $(2 - 8\%)$ mean height error range, for all three methods. The coupled statistical approaches, **Method I** (Combined) and **II** (PCR), have similar error values. Average error for all 80 out-of-training samples is 3.71%, 3.33%, and 3.33% for the Iterative, Method I, and Method II approaches, respectively. 75

46. Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Most samples fall under the $(0.04 - 0.14\,rad)$ surface orientation error range, for all three methods. The coupled statistical approaches, **Method I** (Combined) and **II** (PCR), have similar error values. Average error for all 80 out-of-training samples is 0.06 rad, 0.06 rad, and 0.06 rad for the Iterative, Method I, and Method II approaches, respectively. 76

47. Recovered shapes, together with the input image and ground-truth (GT) shape, for the three model-based methods (Iterative, Method I, and Method II). The results are very close, visually. 77

48. Recovered albedo, together with the input image and ground-truth (GT) albedo, for the three model-based methods (Iterative, Method I, and Method II). The results are very close, visually. 78

49. (**Upper Left**) Mean shape together with its convex hull. The convex hull forms the boundary of the mask. (**Lower**) Each pixel position within the mask corresponds to a certain point on a sample scan in the USF database. The vector $\mathbf{I}_{i,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$ refers to the shape and texture information at the kth vertex of the ith sample scan. (**Upper Right**) Similarly, each pixel position within the mask corresponds to a certain point on the input image. Hence, there is correspondence between the USF database samples and the input image, through the mean shape mask. 79

50. Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Note that the proposed method (Method II-PCR) outperforms its predecessors. Average error for all 80 out-of-training samples is 3.33%, 9.73%, and 6.07% for the PCR-Method II, Castelan, and Ahmed approaches, respectively. 80

51. Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Note that the proposed method (Method II-PCR) outperforms its predecessors. Average error for all 80 out-of-training samples is 0.06 rad, 0.11 rad, and 0.06 rad for the PCR-Method II, Castelan, and Ahmed approaches, respectively. 81

52. Recovered shapes, together with the input image and ground-truth (GT) shape, for the three model-based methods. Methods III and IV refer to Castelan et al. [1] [14] and Ahmed et al. [2], respectively. The results between Methods II and IV are close visually. Castelan's version (Method III) suffers with input images that have illumination. 83

53. Recovered albedo, together with the input image and ground-truth (GT) albedo, for the three model-based methods. Methods III and IV refer to Castelan et al. [1] [14] and Ahmed et al. [2], respectively. The results between Methods II and IV are close visually. Castelan's version (Method III) suffers with input images that have illumination. 84

54. (a) True occlusion mask. (b) Occlusion mask estimates m_i, for each h_i. (c) Occlusion mask estimate using (66). 84

55. Two types of occlusion: (a) circular occlusion with uniform color, (b) contiguous occlusion with non-uniform texture. 86

56. Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 3.33%, 3.61%, and 4.46% for the Case I, Case II, and Case III experiments, respectively. It is clear that the modified robust framework for shape recovery (Algorithm 6) can handle this type of occlusion, i.e., Case II results are closer to Case I. Without the robust framework (Case III), shape recovery results have larger errors. 90

57. Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 0.06 rad, 0.06 rad, and 0.08 rad% for the Case I, Case II, and Case III experiments, respectively. 91

58. Reconstruction results of five out-out-training USF samples for the three cases above, using the circular occlusion. Input images are in the first row. Second row contains the estimated occlusion in the input image. Third row displays the ground-truth albedo of the input. Fourth to sixth rows contain the recovered albedos for **Cases I, II,** and **III,** respectively. Seventh row contains the ground-truth shape. Eighth to tenth row illustrate the recovered shapes for **Cases I, II,** and **III,** respectively. 92

59. Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 3.33%, 3.52%, and 3.84% for the Iterative, Method I, and Method II approaches, respectively. 93

60. Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 0.06 rad, 0.06 rad, and 0.06 rad% for the Case I, Case II, and Case III experiments, respectively. 94

61. Reconstruction results of five out-out-training USF samples for the three cases above, using contiguous occlusion. Input images are in the first row. Second row contains the estimated occlusion in the input image. Third row displays the ground-truth albedo of the input. Fourth to sixth rows contain the recovered albedos for **Cases I, II,** and **III,** respectively. Seventh row contains the ground-truth shape. Eighth to tenth row illustrate the recovered shapes for **Cases I, II,** and **III,** respectively. 96

62. (**Upper Left**) Mean shape together with its convex hull. The convex hull forms the boundary of the mask. (**Lower**) Each pixel position within the mask corresponds to a certain point on a sample scan in the USF database. The vector $\mathbf{I}_{i,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$ refers to the shape and texture information at the kth vertex of the ith sample scan. (**Upper Right**) Similarly, each pixel position within the mask corresponds to a certain point on the input image. Hence, there is correspondence between the USF database samples and the input image, through the mean shape mask. 97

63. Pan angle illustration. The face moves left-to-right or right-to-left sideways. 99

64. Recovered shapes and albedo, together with the input image and ground-truth (GT) shapes and albedo. 102

65. Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are General Pose (Case I) and Frontal (Case II). Note that the proposed method for general pose behaves similarly to Method II-PCR for frontal poses. Average error for all 80 out-of-training samples is 2.34% and 2.28% for the Case I and Case II experiments, respectively. 104

66. Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are General Pose (Case I) and Frontal (Case II). Note that the proposed method for general pose behaves similarly to Method II-PCR for frontal poses. Average error for all 80 out-of-training samples is 0.04 rad and 0.04 rad for the Case I and Case II experiments, respectively. 105

67. Bar graph of the average mean height error with respect to pose changes, i.e., pan angle range of ($-20°$ to $20°$). The graph indicates that the proposed approach is insensitive to pose changes. 106

68. Zoomed version of Figure 67. The plot shows that there is a slight increase in error as the pose angle is increased. 106

69. Bar graph of the average mean surface orientation error with respect to pose changes, i.e., pan angle range of ($-20°$ to $20°$). The graph indicates that the proposed approach is insensitive to pose changes. 106

70. Zoomed version of Figure 69. The plot shows that there is a slight increase in error as the pose angle is increased. 107

71. Stem plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (-90° to 90°). The plot indicates that the proposed approach is insensitive to pose changes only up to $\pm 50^\circ$. 107

72. Stem plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (-90° to 90°). The graph indicates that the proposed approach is insensitive to pose changes only up to $\pm 50^\circ$. 107

73. Plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (-90° to 90°). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change (5°). At pan angle -55° and 55°, there is a change of about -1.5% and 1.5% error, respectively. The plot indicates that the proposed approach is insensitive to pose changes only up to about $\pm 50^\circ$. 108

74. Plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (-90° to 90°). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change (5°). At pan angle -60° and 55°, there is a change of about $-0.01rad$ and $0.01rad$ error, respectively. The plot indicates that the proposed approach is insensitive to pose changes only up to about $\pm 50^\circ$. 108

75. The problem of estimating 3D shape given only the 2D feature points of an input image under unknown pose. The output 3D shape of the estimation method can be sparse or dense. Since prior shape and texture models from real data, which are metric in nature, are incorporated into the 3D shape recovery framework, the output 3D shape are metric, as well. 110

76. **(Upper)** Solve for the camera projection matrix $\mathbf{C}$ given the point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$. Note that $\mathbf{x}_i$ was synthetically generated from $\mathbf{X}_i$. **(Lower)** After projecting $\mathbf{X}_i$ to the 2D space using the computed projection matrix $\mathbf{C}$, the resulting shape fits perfectly to the input 2D feature points. 112

77. (**Upper**) Solve for the camera projection matrix **C** given the point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$. Note that $\mathbf{x}_i$ was *not* synthetically generated from $\mathbf{X}_i$. (**Lower**) After projecting $\mathbf{X}_i$ to the 2D space using the computed projection matrix **C**, the resulting shape does not fit perfectly to the input 2D feature points. However, they are rigidly registered together. 112

78. (**Upper Left**) Albedo (texture) and 3D dense shape of a USF database sample, expressed as Monge patches, i.e., $(x, y, Z(x, y), T(x, y))$. (**Upper Right**) 3D sparse shape corresponding to the annotated positions of the albedo image. (**Lower**) A series of 3D dense shape with its corresponding 3D sparse shapes. 118

79. (**Left**) Compute the camera projection matrix **C** between the 3D sparse mean shape and the input 2D shape ($\mathbf{x}_{inp}$). The mean shape substitutes for the unknown 3D sparse shape where $\mathbf{x}_{inp}$ comes from. (**Right**) Using the computed matrix **C**, the 3D mean shape and a sample USF subject are projected to the 2D space. The results are plotted together with the input 2D feature points. 119

80. Given the input 2D feature points ($\mathbf{x}_{inp}$), a camera projection matrix **P** can be estimated using the 3D mean sparse shape. This camera projection matrix can be used to project all USF database samples to the 2D space. 119

81. Pan angle illustration. The face moves left-to-right or right-to-left sideways. 122

82. Recovered 3D sparse shape of the same input using the three algorithms (proposed, [4], and [5]), together with the ground truth 3D sparse shape. (View 2) represents the projection of the recovered 3D shapes to the *x*-*y* plane. Notice that the results of the proposed method and that of Aldrian et al. [4] are similar. 123

83. Side-by-side stem plot of the time (in seconds) needed to recover the 3D sparse shape for inputs generated from 80 samples of the USF database. The x-axis refers to the sample index of the USF Database. Notice that the proposed method is computationally faster than the others, due to its simplistic regression framework. Average time is 0.03, 0.10, and 0.28 seconds for the proposed, Aldrian et al., and Zhang et al. approaches, respectively. Note that offline training time is not included in Aldrian et al. 130

84. Side-by-side stem plot of the 2D shape projection error (84) for inputs generated from 80 samples of the USF database. Average 2D shape projection error is 5.55, 5.48, and 15.43 seconds for the proposed, Aldrian et al., and Zhang et al. approaches, respectively. . . . 131

85. Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average 3D shape error is 29.00, 28.79, and 31.80 seconds for the proposed, Aldrian et al., and Zhang et al. approaches, respectively. 132

86. Given the input 2D feature points ($\mathbf{x}_{inp}$), a camera projection matrix **P** can be estimated using the 3D mean sparse shape. This camera projection matrix can be used to project all USF database samples to the 2D space. 133

87. Recovered shapes, together with the input image and ground-truth (GT) shape, for the model-based 3D shape recovery framework in Algorithm 8. 135

88. Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are Sparse-to-Dense (Case I) and General Pose (Case II). Average error for all 80 out-of-training samples is 2.51% and 2.34% for the Case I and Case II experiments, respectively. 136

89. Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are Sparse-to-Dense (Case I) and General Pose (Case II). Average error for all 80 out-of-training samples is 0.04 rad and 0.04 rad for the Case I and Case II experiments, respectively. 137

90. 2D-Sparse to 3D-Dense Experiments. Bar graph of the average mean height error with respect to pose changes, i.e., pan angle range of ($-20°$ to $20°$). The graph indicates that the proposed approach is insensitive to pose changes. 138

91. 2D-Sparse to 3D-Dense Experiments. Bar graph of the average mean surface orientation error with respect to pose changes, i.e., pan angle range of ($-20°$ to $20°$). The graph indicates that the proposed approach is insensitive to pose changes. 138

92. 2D-Sparse to 3D-Dense Experiments. Stem plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of ($-90°$ to $90°$). The plot indicates that the proposed approach is insensitive to pose changes, with frontal-pose images having slightly higher errors than non-frontal ones. 139

93. 2D-Sparse to 3D-Dense Experiments. Stem plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (-90° to 90°). The plot indicates that the proposed approach is insensitive to pose changes, with frontal-pose images having slightly higher errors than non-frontal ones. 139

94. Plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (-90° to 90°). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change (5°). It appears that there is no breaking point of pose invariance since the change in error per pan angle incremental change (5°) is close to 0% as the pan angle approaces $\pm 90^\circ$. 140

95. Plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (-90° to 90°). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change (5°). It appears that there is no breaking point of pose invariance since the change in error per pan angle incremental change (5°) is close to 0 rad as the pan angle approaces $\pm 90^\circ$. 140

96. Sample annotations of the USF database. There is a total of 76 manually annotated landmark points, including both anatomical and pseudo-landmarks. 150

97. Dense correspondence between two 3D scans in the cyclograph (parameterized by h and ϕ) domain. 155

98. The mean face of the USF database. (a) mean face in RGB format parameterized by h and ϕ, (b) radius data of the mean face parameterized by h and ϕ, (c) mean face expressed in Cartesian coordinates, visualized as an OBJ file [15]. 156

LIST OF ALGORITHMS

ALGORITHM PAGE

1 Compute Spherical Harmonics Projection (SHP) Image 29

2 Compute Robust Spherical Harmonics Projection (SHP) Image 39

3 Iterative Approach to Model-based 3D Face Shape Recovery 62

4 Coupled Model Approach to Model-based 3D Face Shape Recovery . 65

5 Principal Component Regression (PCR) Framework for 3D Shape Recovery 68

6 Robust Model-based 3D Shape Recovery Framework 86

7 Principal Component Regression (PCR) Framework for 3D Sparse Shape Recovery 121

8 Principal Component Regression (PCR) Framework for 3D Dense Shape Recovery 134

CHAPTER I

INTRODUCTION

Humans have the exceptional ability to perceive the world in three dimensions (3D), otherwise known as depth perception [16]. The amazing thing about this ability to determine distances is that it depends only on a simple two-dimensional (2D) image in the retina. It is an interesting problem to explain and mimic this phenomenon of getting a three-dimensional perception of a scene from a flat 2D image of the retina.

The main objective of this dissertation is the computational aspect of this human ability to reconstruct the world in 3D using only 2D images from the retina. Specifically, the goal of this work is to solve the problem of recovering 3D facial shape information from a single image of unknown pose and illumination. Figure 1 sums up this whole document into a single illustration. The input, which can be images of arbitrary pose and illumination, is fed into a 3D facial shape estimation algorithm that outputs the desired 3D facial shape.

This chapter starts with a discussion on how psychologists approach the problem of depth perception. The next topic talks about the computer vision approach to depth perception, which is known as shape recovery. It then describes the prior art of recovering 3D facial shape information from a single image of unknown pose and illumination. A section is allotted to defend why this work is needed and relevant. The rest of this chapter includes a list of dissertation contributions and the layout of the whole dissertation document.

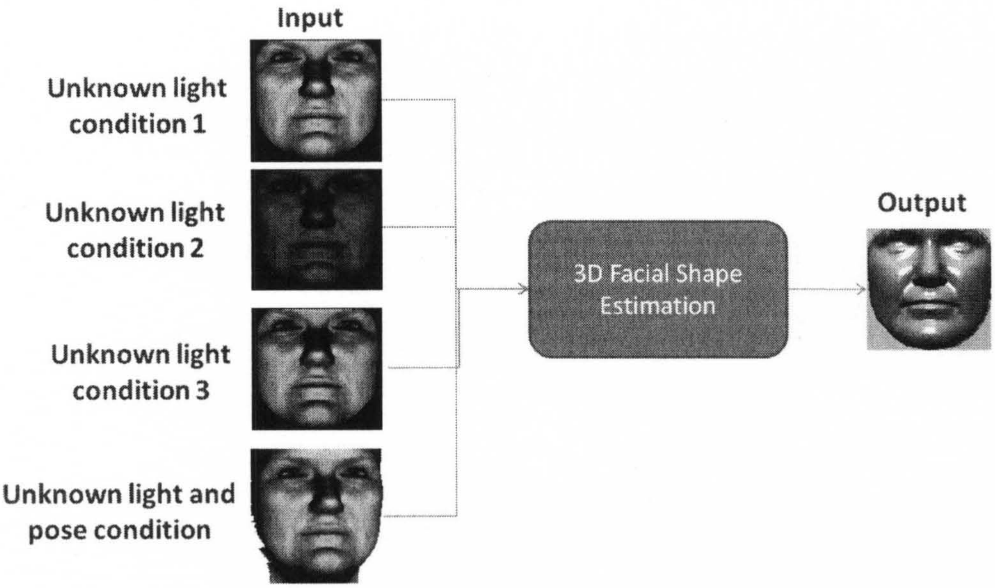

Input
Unknown light condition 1
Unknown light condition 2
Unknown light condition 3
Unknown light and pose condition
3D Facial Shape Estimation
Output

FIGURE 1 – 3D facial shape estimation problem under general illumination and pose. The goal is to recover 3D facial shape from images of unknown lighting and pose conditions, using a 3D estimation black box.

A. Pyschology of Depth Perception

Psychologists approach this problem by asking what information is available from the 2D image that enables humans to perceive depth. This is referred to as the cue approach to depth perception. According to the cue theory [17], humans learn the connection between this cue and depth through previous experiences. The link between a particular cue and its corresponding depth becomes automatic after these learning experiences that when presented with depth cues in the future, the world is experienced as three-dimensional.

These cues can be divided into three major classes: (a) oculomotor - cues based on the ability to the sense eye positions and the tension in the eye muscles, (b) monocular - cues that work with one eye, and (c) binocular - cues that depend on two eyes. Subsequent discussions will focus on both binocular and monocular cues for depth perception.

The difference in the viewpoint of the two eyes, which are about six centimeters apart in adults, creates the cue of binocular disparity. The impression of depth that results from the information provided by binocular disparity is called stereopsis.

In contrast to binocular cues, monocular cues work with a single eye. Pictorial cues, which are depth information stored in a two-dimensional image, and motion-based cues, which refer to depth information created by movement, are examples of monocular cues. The next section covers several examples of pictorial and motion-based cues.

1. Motion-based Cues

A common example of a motion-based cue is the concept of motion parallax. It occurs when, as the observer moves, near objects appear to rapidly pass by while farther objects seem to move slowly. Motion parallax is one of the most important sources of depth information for many animals [16], not just humans.

2. Pictorial Cues

Occlusion is a type of pictorial cue. It happens when one object partially hides another object from the viewer. The partially hidden object is considered to be farther away, as illustrated in Figure 2, where the mountains are perceived to be farther away than the barn. Occlusion only provides relative distance, as opposed to absolute distance, i.e., no information can be further extracted except that the occluded object is farther from the item in front.

The cue of familiar size refers to the ability of humans to determine distance using prior knowledge of the sizes of objects. An experiment conducted by William Epstein [18] showed that under certain conditions, prior knowledge of an object's size can influence perception of that object's distance. The stimuli in Epstein's experiment were same-size images of a dime, a quarter, and a half dollar, which were placed in the same distance from an observer. Epstein made an illusion that the images were real coins by putting the images in a darkened room, illumi-

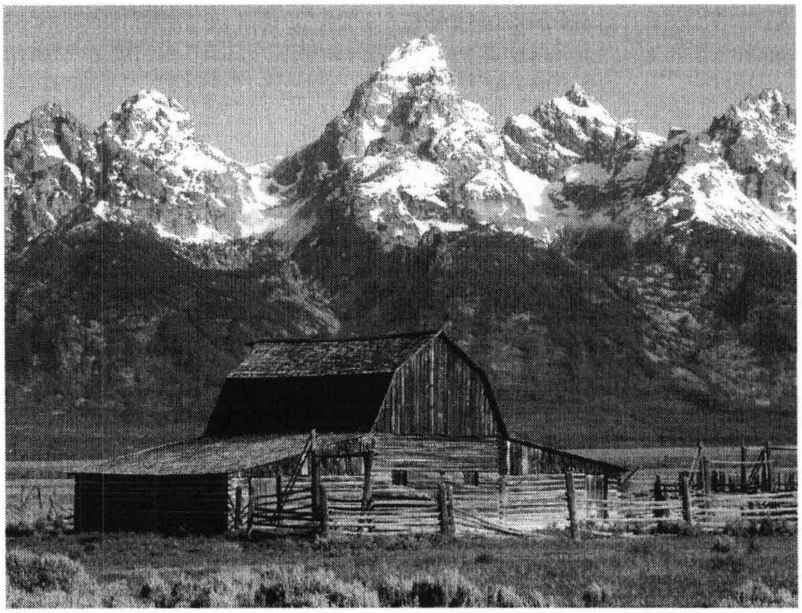


FIGURE 2 – The mountains in the background [6], partially hidden by the barn, are perceived to be farther away than the latter due to a type of pictorial cue called occlusion.

nating them with a spot light, and having the observers view the images with only one eye.

The subjects estimated that the dime was closest, the half-dollar being the farthest, and the quarter in the middle. These answers were influenced by their knowledge of actual dimes, quarters, and half-dollars. Interestingly, this result did not happen when the observers' two eyes were open, since the use of both eyes provided additional information indicating that the coins were at the same distance from the viewer. The conclusion of Epstein's study is that the cue of familiar size is most effective when other information about depth is absent.

Another important source of depth information is the texture gradient. Elements that are equally spaced in a closer scene appear to be more closely packed as distance increases. This cue is closely related to another one, which is the cue of relative size that states that more distant objects take up less of the observer's field of view. This is the reason behind the tightly-packed nature of the faraway elements of the texture. This cue is illustrated in Figure 3.

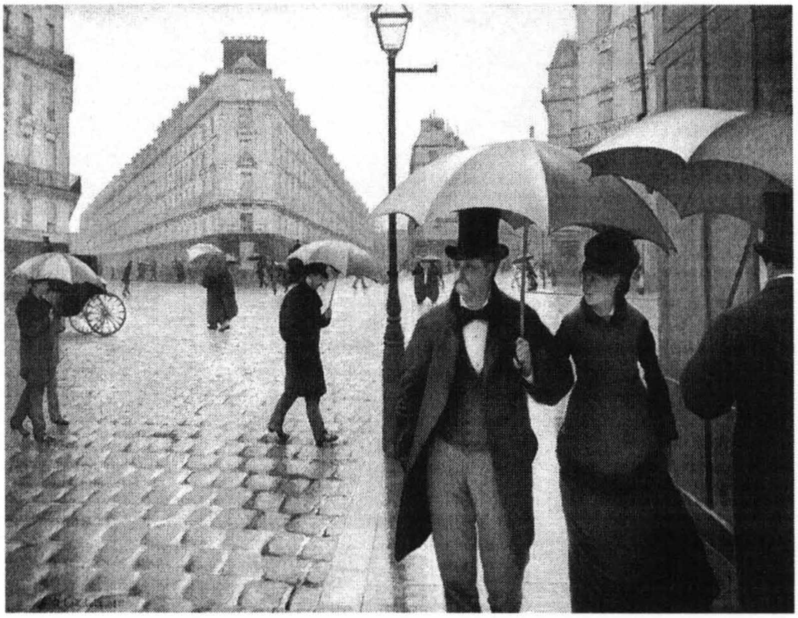


FIGURE 3 – The painting (Paris Street, Rainy Day) by Gustave Caillebotte [7] illustrates the cue of texture gradient, where elements that are equally spaced in a closer scene appear to be more closely packed as distance increases.

Atmospheric perspective refers to the case when more distant objects appear less sharp and usually have a slight blue tint. The more distant an object is, the more air and particles (e.g., dust, water droplets, and pollution) interfere with the view, making distant objects look less sharp and bluer than closer objects. Figure 4 shows an example of atmospheric perspective.

The cues of lighting, shading and shadows represent the way light falls on an object and how it is reflected by its surface. In perceiving depth using these cues, it is commonly assumed [19] that there is only one light source and light comes from above, retinally (i.e., light comes from the same direction as the top of the head). Since the closest surface of an object to the light is brightest, humans can easily infer about the depth and shape of an object from the pattern of shading and lighting alone [20]. Artists have long used this technique to illustrate depth in two-dimensional images, as illustrated in Fig. 5.

The majority of depth cues discussed earlier provide highly correlated quantitative information, e.g., texture gradients and the dynamic cues of motion paral-

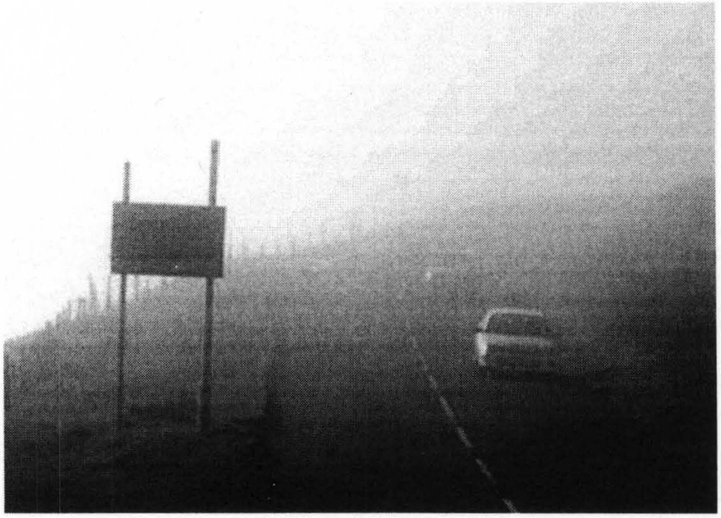


FIGURE 4 – A fogged roadway [8] illustrating the cue of atmospheric perspective. With fog interfering with the view, the more distant car appears less sharp.

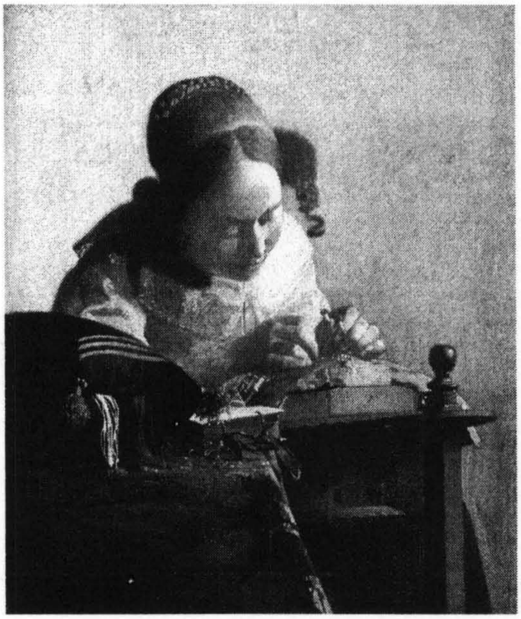


FIGURE 5 – A painting (The Lacemaker) by Johannes Vermeer [9], where the cues of lighting and shading are used to illustrate depth.

lax vary systematically with distance according to a power function determined by the observer's height [21]. Hence, different depth cues can be combined together to come up with a better depth estimate.

The methods in this dissertation fall under the monocular class, pictorial-based cues, specifically, the cues of lighting and shading, as well as the cue of familiar size.

B. Shape-from-X

The previous sections discussed the psychology of depth perception, where numerous concepts such as monocular and binocular cues for depth perception were mentioned. Of special importance are the monocular pictorial cues that will be used throughout this thesis.

In computer vision, there is an analogous term to depth perception, which is *shape recovery*. Shape recovery is, basically, the computational equivalent of the depth perception concepts from psychology.

Shape recovery is a classic problem in the computer vision field, where the goal is to derive a three-dimensional scene description from one or more two-dimensional images [22]. The recovered shape can be expressed in several ways: (a) depth $Z(x, y)$, (b) surface normal (n_x, n_y, n_z), (c) surface gradient (p, q), and (d) surface slant (ϕ) and tilt (θ).

The depth, $Z(x, y)$, can be considered either as a relative distance from the camera to surface points, or the relative height above the x-y plane. The surface gradient, $(p, q) = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y})$, is the rate of change of depth in the x and y directions. The surface slant, (ϕ), and tilt, (θ), are related to the surface normals using a Cartesian-to-spherical coordinate system relationship [23].

The class of algorithms that deals with different types of shape recovery is conveniently named as shape-from-**X** techniques, where **X** can be stereo, motion, shading, texture, etc. Notice that **X** has corresponding analogues to the different cues of depth perception from the previous section.

1. Image Formation

The next sections would be incomplete without a discussion on image formation, an integral concept in computer vision. Image formation refers to the response of an image sensor (in a camera system) to incoming light. There are two parts to consider in the image formation process: (a) the geometry of image formation and (b) physics of light.

The geometry of image formation determines where in the image plane the projection of a scene point is located. Two common projection models are the *perspective* and *orthographic* projection models [3]. This work uses the orthographic projection model, which is discussed thoroughly in Chapter 3, to describe the transformation from a 3D point in a scene to a 2D point in the image.

The physics of light determines the brightness of a point in the image plane as a function of illumination and surface properties. For this work, distant light sources are used, together with a Lambertian surface reflectance model. Chapter 2 discusses in detail the Lambertian model, which assumes that each surface point appears equally bright from all directions. Figure 6 illustrates a simple image formation model where the scene is illuminated by a single distant light source and the surface follows that of a Lambertian reflectance model.

2. Shape-from-Stereo

This section illustrates how binocular cues translate into actual depth through shape-from-stereo. The diagram in Figure 7 shows a simple stereo system composed of two pinhole [24] cameras. The left and right image planes are coplanar and represented by the segments I_l and I_r. O_l and O_r are the centers of projection and the optical axes are parallel. Assuming that the correspondence problem [25] has now been solved, the next step is to perform reconstruction, i.e., expressing

$$I(\mathbf{p}) = l\lambda \max(\mathbf{n} \cdot \mathbf{l}, 0) = l\lambda \max(\cos\theta, 0)$$

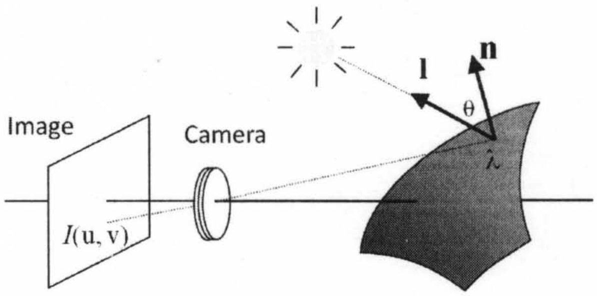


FIGURE 6 – Distant light source and Lambertian surface illustration. The intensity of a surface point (with albedo λ) depends on the angle between the surface normal **n** and light source direction **l**, no matter what the viewing angle is.

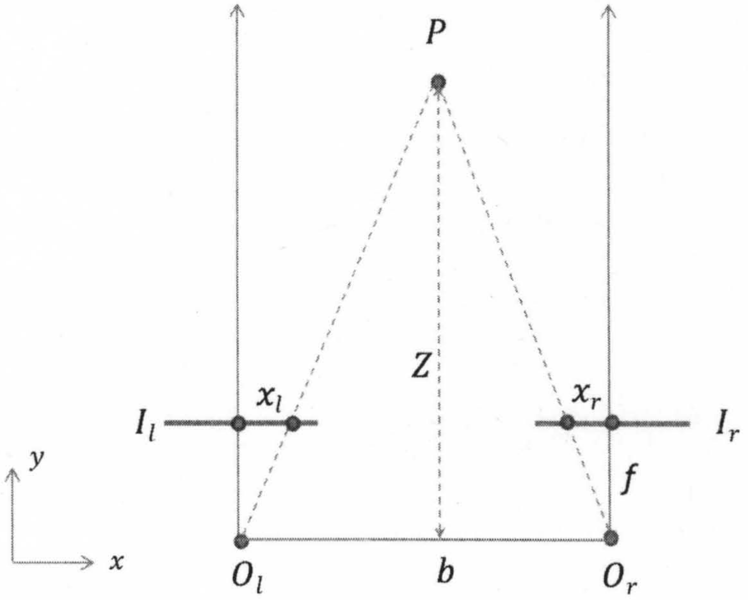


FIGURE 7 – A simple stereo system illustrating how depth can be extracted given the binocular disparity. The resulting relationship shows that depth is inversely proportional to disparity.

point P in 3D.

Let x_l and x_r be the image projections of point P to the left and right cameras. The distance b between the cameras' centers of projection is called the baseline. f is the common focal length of the two cameras. The depth, Z, is the perpendicular distance between the baseline and point P, which the shape-from-stereo problem is trying to solve. From the similar rectangles, (x_l, P, x_r) and (O_l, P, O_r),

$$\frac{b - (x_l - x_r)}{Z - f} = \frac{b}{Z} \tag{1}$$

Solving for Z in (1) leads to a simple relationship

$$Z = f\frac{b}{d} \tag{2}$$

where $d = x_l - x_r$ is the disparity, the difference in retinal position between the corresponding two points in the left and right images. Notice that depth is inversely proportional to disparity. This relationship can be used to explain motion parallax, the phenomenon where distant objects seem to move slower than closer ones [24].

3. Shape-from-Shading (SFS)

The shape-from-shading problem is an interesting field in computer vision that involves recovering the 3D shape of an object using the cues of lighting and shading discussed previously. SFS was formally introduced to the computer vision literature by the seminal work of Horn [26], over three decades ago.

Figure 8 illustrates the shape-from-shading problem. The image formation process in Figure 8 takes a 3D shape as input and outputs the 2D image. The SFS problem, basically, takes the opposite direction such that the input is the 2D image and the output is the recovered 3D shape.

Two tasks need to be accomplished to solve the SFS problem [27], namely: (a) formulate an imaging model that describes the relationship between the surface and image brightness and (b) after creating the imaging model, a numerical

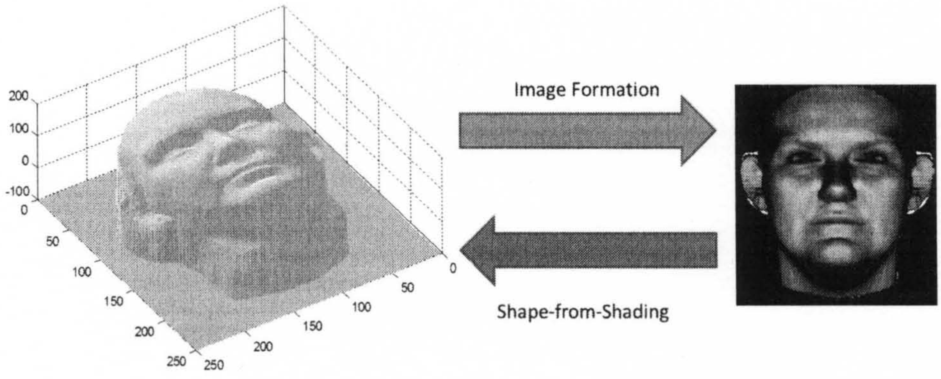


FIGURE 8 – Shape-from-shading problem

algorithm needs to be developed to reconstruct the 3D shape from the image.

The Lambertian model is the simplest and most widely used among the imaging models. It assumes that each surface point appears equally bright from all viewing directions. Various materials with rough and nonspecular surfaces, such as matte paint and paper, exhibit this type of behavior. Formally, with the Lambertian model, the intensity I at point $\mathbf{p}$, $I(\mathbf{p})$ is determined by taking the dot product between the surface normal $\mathbf{n}$ and the lighting direction $\mathbf{l}$, and scaled by the albedo λ

$$I(\mathbf{p}) = l\lambda(\mathbf{n} \cdot \mathbf{l}) = l\lambda\cos(\theta) \tag{3}$$

where θ is the angle between the surface normal and light direction and l is the intensity of the light ray.

Horn obtained a nonlinear first-order partial differential equation, called the *image irradiance equation*, to describe the SFS problem. This image irradiance equation models the relationship between the object shape and its corresponding image brightness under known illumination conditions. The image irradiance equation can be written as

$$E(\mathbf{x}) = R(\mathbf{n}(\mathbf{x})) \tag{4}$$

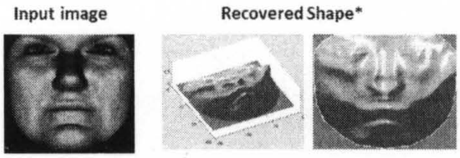


FIGURE 9 – Shape-from-shading results of Ahmed and Farag [10] [11] on an illuminated image of a human face.

where $E(\mathbf{x})$ is the image irradiance at point **x** and $R(.)$ is the radiance of the surface patch with unit normal $\mathbf{n}(\mathbf{x})$. For convenience, most approaches in the SFS literature, assume that the surface reflect light with the simplistic Lambertian model discussed above.

Minimization approaches [28] to SFS solve for the solution that minimizes an energy function over the whole image. This energy function involves the brightness constraint, which is derived directly from the image irradiance equation,

$$\int\int (I - R)^2 dxdy \tag{5}$$

where I is the measured intensity and is a scaled version of $E(\mathbf{x})$, and R is the estimated reflectance. Several constraints that regularize the solution, such as smoothness, integrability, and intensity gradient constraints, can be added to help in the minimization procedure.

However, real images, especially that of human faces, do not always follow the Lambertian assumption. It is with this motivation that Ahmed and Farag [10] [11] incorporated several existing physics-based models into the SFS framework to deal with non-Lambertian conditions. Their results are successful with images of several objects but does not work with human faces, as illustrated in Figure 9.

There are several serious limitations of current SFS algorithms, namely: (a) the light source direction should be known in advance and (b) there should be a single light source. However, real-life images usually are formed not only with multiple light sources but the sources themselves are unknown, in nature. This dissertation has the advantage over previous SFS algorithms by not having these limitations.

C. Related Methods

As discussed previously, shape-from-shading methods (SFS) provide an alternative means of recovering 3D shape from images; the other approaches being shape-from-X techniques (X can be stereo, motion, etc.) [22].

Previous works have shown that constraining the shape-from-shading algorithm to a specific class of objects can improve the accuracy of the recovered 3D shape [29]. There is particularly a huge interest in the 3D shape recovery of human faces from intensity images. Zhao and Chellapa [30] used the known bilateral symmetry of frontal faces as a geometric constraint in their approach.

1. Statistical Model-based Approaches

The next set of methods use statistical prior models to perform better in shape recovery. Furthermore, the use of prior models is rooted in the cue theory of depth perception, which states that humans learn the connection between cue and depth through previous experiences. The work of Knill [31] studied prior models of monocular cues for depth perception, from a psychology point of view.

Atick et al. [32] proposed the first statistical SFS method by parameterizing the set of all possible facial surfaces using principal component analysis (PCA). Dovgard and Basri [33] introduced a statistical symmetric SFS method by taking into account both the statistical constraint of [32] and the geometric constraint of symmetry in [30]. Recently, Smith and Hancock [34] embedded a statistical model of surface normals within a shape-from-shading framework.

There has been a substantial amount of work regarding statistical face processing in the computer vision literature. The morphable model framework of [35] estimates the shape and texture coefficients from an input 2D image, together with other scene parameters, using an optimization method based on stochastic gra-

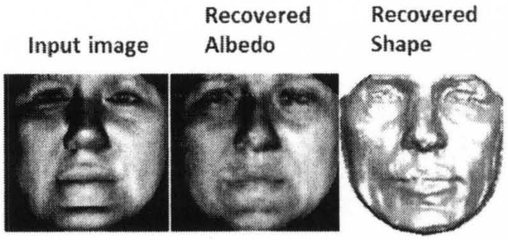


FIGURE 10 – Recovered 3D shape using Castelan et al. [1] on an input with frontal pose and unknown illumination.

dient descent. It is a 3D extension of the seminal work of Cootes et al. [36] on Active Appearance Models (AAM), where a coupled statistical model is generated to describe the 2D shape and appearance (albedo) information of faces. Appendix I provides a full description of both AAM and morphable model algorithms.

Castelan et al. [1] developed a coupled statistical model, which is a variant of the combined AAM [36], that can recover 3D shape from intensity images with a frontal pose. The shape and intensity models in Castelan's work is similar to that of the AAM model discussed in the appendix. Note that in the shape recovery literature, albedo can be used, interchangeably, with the term intensity. The primary difference in Castelan's approach is that the 2D shape model in AAM is replaced with a 3D shape (height map) model [37].

The main advantage of the Castelan approach over the 3D morphable model framework is in the straightforward recovery of the 3D face shape, which does not go through a costly iterative optimization process, i.e., shape recovery can be performed using a series of matrix operations. Figure 10 shows the recovered shape on a sample input with unknown illumination. The results are not acceptable since no illumination is included in the Castelan approach.

To alleviate this problem, Ahmed and Farag [2] incorporated an illumination model to the original coupled model of Castelan, to deal with frontal images of arbitrary lighting. This is made possible through the use of the concept of *spherical harmonics*. Basri et al. [38] and Ramamoorthi [39] independently proved that the set of images of a convex Lambertian object can be approximated accurately by low-dimensional linear subspace based on spherical harmonics (SH). The proposed methods in this thesis rely on this conclusion to perform shape recovery from 2D images of general and unknown lighting.

The coupled model of Ahmed and Farag [2] can be considered as the foundation of this work and will be used for comparison purposes in later chapters. This dissertation extends [2] in several aspects.

Basically, the shape recovery algorithms, based on coupled models mentioned above, are examples of statistical shape-from-shading. However, it is not evident how they are related to the classical SFS formulation due its coupled model nature. This dissertation is successful in connecting the coupled model approaches to the classical SFS equations [40] [41]. Along the way, two related algorithms of statistical SFS were formulated to perform shape recovery in this work, which are discussed thoroughly in the next chapters.

The alignment procedure in [2] only uses three points (two eyes and mouth positions). As mentioned in the morphable model framework of [35], dense correspondence is crucial for successful reconstruction. This dissertation uses 76 points as control points to find more accurate dense correspondence between the reconstruction database samples. Chapter 3 provides a comprehensive discussion about this process.

Robust statistics are incorporated into this work to deal with non-ideal conditions during the reconstruction process. Specifically, if outliers are present in the input images, the computation of spherical harmonic coefficients are affected. This dissertation [42] solves this problem with the use of Geman-McClure and Lorentzian functions [43].

There is no pose invariance in both Castelan [1] and Ahmed et al. [2] approaches, i.e., only frontal poses are considered. This dissertation can handle unknown pose with the help of multiple-view geometry [3] concepts. Therefore, the methods under this dissertation can be considered to be both pose and illumination invariant.

Figure 11 shows sample reconstruction results of the proposed method in later chapters on several inputs under unknown pose and illumination. Reconstruction is performed in terms of both albedo (defined as the scale that connects the image irradiance (E) and intensity (I), i.e., $I = \lambda E$) and shape. From a visual comparative perspective, the reconstructions are close to its groundtruth (GT) counterparts.

D. Why This Work Is Needed

3D shape recovery methods from single 2D images of any generic object have various real-world applications. Recently, Bustard et al. [44] developed an algorithm to extract 3D information of ears from a 2D image. Blanz et al. [45] used this class of methods to infer 3D shape from dental images.

The proposed methods here are developed specifically for faces, with the intended application to be face recognition at-a-distance problems [46]. The rationale behind this is that at long distances, it is difficult to use 3D sensors to infer 3D data and help improve recognition. These algorithms will help solve this problem. Further, even though the methods here are for a specific type of problem, it can be applied to other objects as well, such as ears and teeth, as mentioned before.

E. Dissertation Contributions

The main goal of this work is to recover 3D facial shape information from a single image of arbitrary pose and illumination. The main contributions of this dissertation can be categorized into two classes:

- 3D Facial Shape Estimation using Texture and Shape Information:

1. The classical shape-from-shading iterative equation is cast as a regression framework, which can be solved efficiently using Principal Component Regression (PCR).

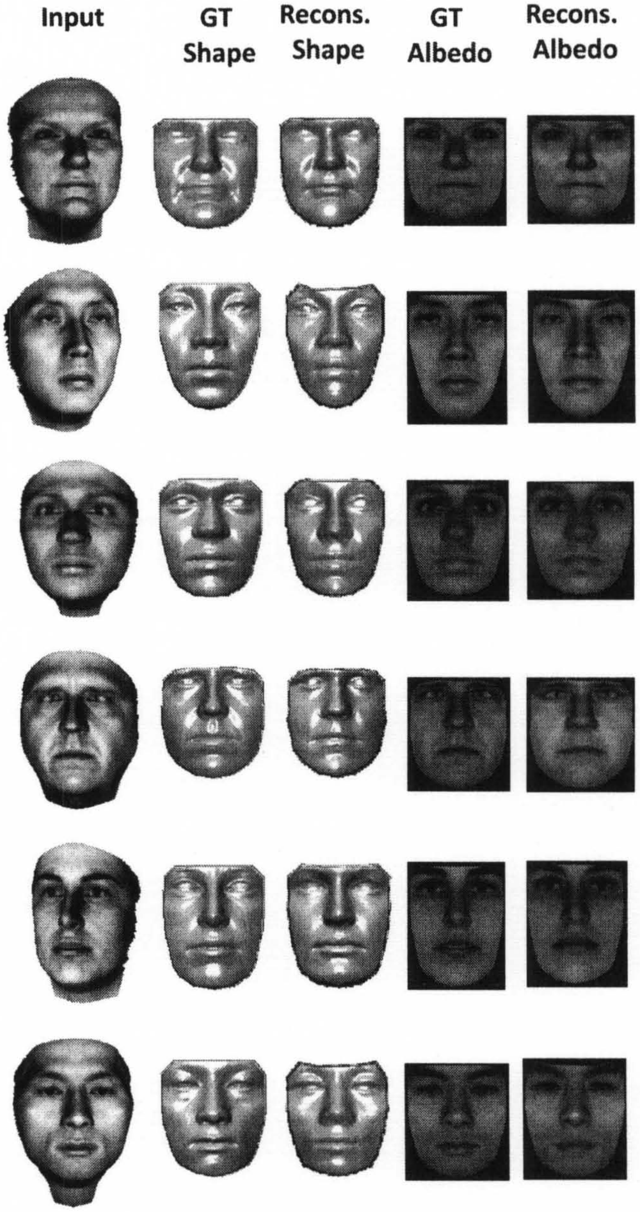


FIGURE 11 – Recovered shapes, together with the input image and ground-truth (GT) shape, using the proposed method in the next chapters. Reconstruction is performed in terms of both albedo (defined as the scale that connects the image irradiance (E) and intensity (I), i.e., $I = \lambda E$) and shape. From a visual comparative perspective, the reconstructions are close to its groundtruth (GT) counterparts.

2. Robust statistics is incorporated into the computation of Spherical Harmonics Projection (SHP) images, to deal with non-ideal situations, e.g., noise and occlusion.

3. Pose-invariance is added to the model-based shape recovery framework to handle both unknown illumination and pose.

- *3D Facial Shape Estimation using Shape Information Alone:*

1. Estimation 3D facial shape given only the 2D input feature points.

2. The problem is cast as a regression framework, similar to model-based SFS framework above, allowing it be computationally efficient.

3. Estimate both 3D sparse and dense shapes given only 2D input sparse points.

F. Document Layout

This document is presented in four chapters. The following remarks summarize the scope of each chapter.

Chapter II discusses the basics of spherical harmonics and how it can be applied to image formation with a Lambertian surface reflectance model. The concept of Spherical Harmonics Projection (SHP) images is introduced. Robust statistics is incorporated to deal with non-ideal conditions when computing for the SHP image.

Chapter III starts first with the model-based 3D shape recovery framework for arbitrary illumination and fixed pose. The classical shape-from-shading equation is cast into a regression framework, which can then be solved using Principal Component Regression (PCR). The fixed frontal pose limitation is relaxed to include general pose, making it a 3D estimation algorithm for general pose and illumination.

Chapter IV deals with the interesting question if 3D shape can be recovered using shape information alone, unlike in the previous chapter where both shape and texture are needed. Results show acceptable 3D shape recovery compared to the ground truth and previous methods.

CHAPTER II

MODELING IMAGES OF OBJECTS UNDER VARIABLE ILLUMINATION USING SPHERICAL HARMONICS

This chapter discusses the use of spherical harmonics to model images of objects under arbitrary illumination. For this work, it is assumed that the object surface exhibits Lambertian reflectance. The SH concepts discussed here are then used to model the Lambertian surface and any unknown lighting function. The end result is a simplifying conclusion that the set of images of objects under varying illumination can be efficiently represented as a linear combination of harmonic images [38].

This chapter will review the basic concepts of spherical harmonics and Lambertian surfaces, and how to find the low-dimensional subspace that describes the images of the convex Lambertian object. The concept of Spherical Harmonics Projection (SHP) images, which is integral to later algorithms in this dissertation, will be discussed. In addition, a robust way to compute SHP images in non-ideal situations (e.g., noise and occlusion) will be described.

A. Lambertian Model

A surface reflectance model describes the way in which a surface reflects incident light [24]. A well-known example is the Lambertian model, which assumes that each surface point appears equally bright from all viewing directions. Various materials with rough and nonspecular surfaces, such as matte paint and paper, exhibit this type of behavior. Formally, the intensity I at point $\mathbf{p}$, $I(\mathbf{p})$ is determined by taking the dot product between the surface normal $\mathbf{n}$ and the lighting direction $\mathbf{l}$, and scaled by the albedo λ

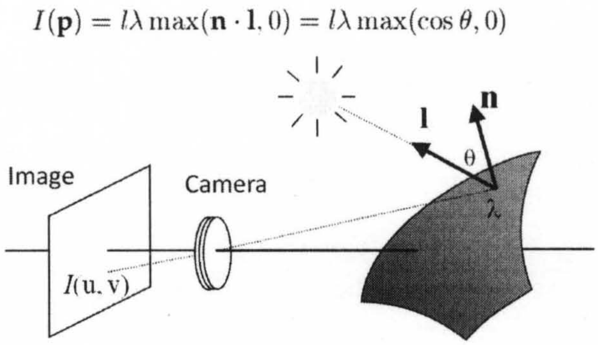

$I(\mathbf{p}) = l\lambda\max(\mathbf{n}\cdot\mathbf{l}, 0) = l\lambda\max(\cos\theta, 0)$
Image
Camera
I(u, v)
l
n
θ
λ

FIGURE 12 – Lambertian surface illustration. The intensity of a surface point (with albedo λ) depends on the angle between the surface normal $\mathbf{n}$ and light source direction $\mathbf{l}$, no matter what the viewing angle is.

$$I(\mathbf{p}) = l\lambda(\mathbf{n}\cdot\mathbf{l}) = l\lambda\cos(\theta) \tag{6}$$

where θ is the angle between the surface normal and light direction and l is the intensity of the light ray. Figure 12 illustrates the Lambertian model.

By definition, attached shadows (also known as self-shadows) occur when a surface faces away from the light source direction. The effect of attached shadows on (6) is a negative intensity value. To solve this problem, the modified equation for Lambertian reflectance is

$$I(\mathbf{p}) = l\lambda\max(\mathbf{n}\cdot\mathbf{l}, 0) = l\lambda\max(\cos\theta, 0) \tag{7}$$

where the function $\max()$ ensures that intensity values are greater than or equal to zero.

B. Spherical Harmonic (SH) Analysis

The surface spherical harmonics are a set of functions that form an orthonormal basis for any spherical function described on the surface of a sphere [47]. It is analogous to the common Fourier basis functions of linear functions. Any piecewise continuous spherical function, $f : \mathbb{S}^2 \to \mathbb{R}$, can be expressed as a linear combination of an infinite series of harmonics

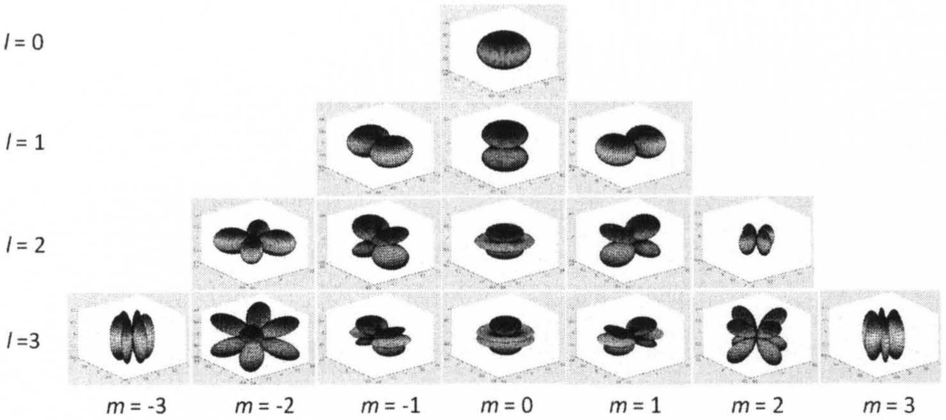


FIGURE 13 – Illustration of the first four spherical harmonic bands, $l = 0, \cdots, 3$. Green and red parts indicate positive and negative values, respectively.

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_{lm} Y_{lm}(\theta, \phi) \tag{8}$$

where $\mathbb{S}^2$ refers to points on the surface of a unit sphere, f_{lm} are scalar values (commonly known as spherical harmonic coefficients) and $Y_{lm} : \mathbb{S}^2 \rightarrow \mathbb{R}$ are the spherical harmonics functions. The spherical harmonic Y_{lm} can be written as

$$\begin{aligned} N_{lm} &= \sqrt{\tfrac{2l+1}{4\pi} \tfrac{(l-m)!}{(l+m)!}} \\ Y_{lm}(\theta, \phi) &= N_{lm} P_l^m(\cos\theta) e^{Im\phi} \end{aligned} \tag{9}$$

where N_{lm} is a normalization factor, P_l^m are Associated Legendre Polynomials (ALP). The indices l and m break the family of ALP into bands of functions. The argument l is referred to as the *band index* and takes the values, $l \geq 0$. The argument m is related to l with the following relation, $-l \leq m \leq l$. It is interesting to visualize how the basis functions look like when plotted as spherical functions. Figure 13 illustrates the first four spherical harmonic bands, $l = 0, \cdots, 3$. Green and red parts indicate positive and negative values, respectively.

1. Spherical Harmonics Projection

Basis functions can be informally considered as small pieces of signals that can be scaled and combined to produce an approximation to an original function. The process of working out how much of each basis function to sum is called *projection* [12].

An arbitrary function can be approximated using basis functions by coming up with a scalar value that that represents how much the original function is similar to each basis function, e.g., $Y_{lm}(\theta, \phi)$ in the spherical harmonics case. This is done by integrating the product of the original function and the basis function over the full domain of the original function. This projection process over all basis functions returns a vector of coefficients.

The process of projecting a spherical function into spherical harmonics (SH) coefficients is straightforward. The harmonic coefficients can be computed as

$$f_{lm} = \int_{\mathbb{S}^2} f(u) Y_{lm}(u)\, du \tag{10}$$

where $u = (\theta, \phi)$.

2. Spherical Harmonics Reconstruction

To reconstruct the approximated function, $\tilde{f}(u)$, one just take the reverse process and sum the scaled (using the computed SH coefficients, f_{lm}) SH basis functions. Suppose the first four bands are used, i.e., $l = 0, \cdots, 3$, the reconstructed signal is computed as

$$\tilde{f}(u) = \sum_{l=0}^{3} \sum_{m=-l}^{l} f_{lm} Y_{lm}(u) \tag{11}$$

Figure 14 is a visualization of the spherical function taken from an example in [12]. This function will be reconstructed using the first n SH coefficients described in (10). The integration is performed numerically using *Monte Carlo In-*

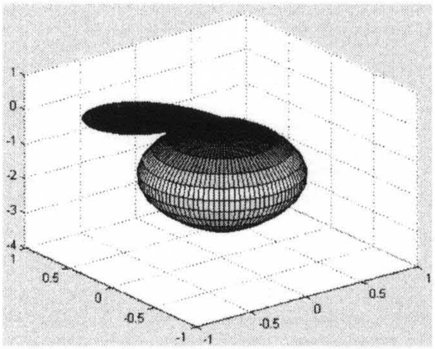
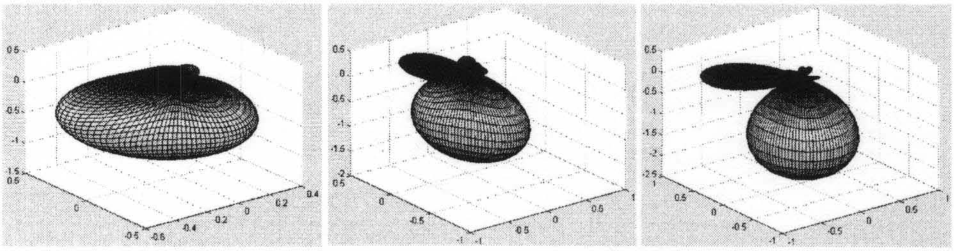


FIGURE 14 – Sample spherical function taken from [12], displayed as a spherical plot. The actual function definition is $f(\theta,\phi) = \max(0, 5\cos(\theta) - 4) + \max(0, -4\sin(\theta-\pi) * \cos(\phi - 2.5) - 3)$.



$n = 10$ $n = 15$ $n = 25$

FIGURE 15 – Reconstruction of the spherical function at Fig. 14 using the first 10, 15, and 25 SH coefficients. Notice that the reconstruction using 25 coefficients is visually close enough to the original function.

tegration. Figure 15 shows the reconstruction using the first 10, 15, and 25 SH coefficients. Notice that the reconstructed function approaches the original one (Figure 14) as more coefficients are used.

3. SH Analysis of Lambertian Reflectance and Light Source Function

With a change in variables, equation (7) can be written as a product of two functions, namely, the Lambertian kernel, $k()$, and light source function, $l()$

$$I(\mathbf{p}) = \lambda k(\theta) l(\mathbf{l}) \tag{12}$$

where the albedo λ is usually set to 1, $k(\theta) = max(\cos\theta, 0)$ and $l(\mathbf{l})$ takes into consideration both light source intensity and direction. If there are multiple light sources, the light reflected by the point would be an integral over the contribution of each light direction. Let $d\mathbf{l}$ be a differential light direction, $\mathbf{n}$ be the specific normal at a certain surface point $\mathbf{p}$, the intensity $I(\mathbf{p})$ is

$$I(\mathbf{p}) = \lambda \int_{S^2} k(\theta) l(\mathbf{l})\, d\mathbf{l} \tag{13}$$

Equation (13) can be regarded as a convolution on the sphere, i.e., $I(\mathbf{p}) = \lambda k(\theta) * l(\mathbf{l})$.

Both the Lambertian kernel and light source function can be written as a linear combination of spherical harmonics basis functions, i.e.,

$$l(\mathbf{l}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} l_{lm} Y_{lm} \qquad k(\theta) = \sum_{l=0}^{\infty} k_l Y_{l0} \tag{14}$$

where the harmonic expansion of $k(\theta)$ considers only the instances of Y_{lm} at $m = 0$ since $k(\theta)$ is circular symmetric about the north pole and $k_{lm} = 0$ at $m \neq 0$. According to the Funk-Hecke theorem [38], the harmonic expansion of (13) can be written as

$$I(\mathbf{p}) = \lambda k(\theta) * l(\mathbf{l}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} (\alpha_l l_{lm}) Y_{lm} \tag{15}$$

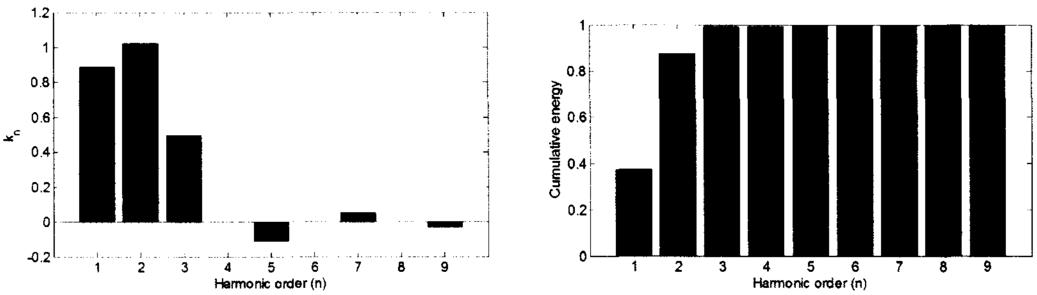
where $\alpha_l = \sqrt{\frac{4\pi}{2n+1}} k_l$.

The first few harmonic coefficients of the Lambertian kernel are

$$\begin{aligned} &k_0 \approx 0.8862 && k_1 \approx 1.0233 \\ &k_2 \approx 0.4954 && k_4 \approx -0.1108 \\ &k_6 \approx 0.0499 && k_8 \approx -0.0285 \end{aligned} \tag{16}$$

where the rest of the terms (k_3, k_5, k_7) are all equal to zero.

The energy captured by every harmonic term is defined as the square of the corresponding coefficient divided by the total squared energy of the transformed



(a) (b)

FIGURE 16 – Input image corruped with (a) salt & pepper noise and (b) occlusion.

function. The total squared energy of the Lambertian kernel (a half-cosine function) is $2\pi/3$. Figure 16 shows graphically the plot of the first nine harmonic coefficients together with the cumulative energy of the Lambertian kernel.

Looking at Figure 16(a), it is clear that the Lambertian kernel k acts as a low-pass filter, with high-frequency light components having little effect on $I(\mathbf{p})$. Therefore, it is possible to get a low-dimensional approximation of $I(\mathbf{p})$ by neglecting the higher-order terms in (15), i.e.,

$$I(\mathbf{p}) = \lambda \sum_{l=0}^{\infty} \sum_{m=-l}^{l} (\alpha_l l_{lm}) Y_{lm} \approx \lambda \sum_{l=0}^{N} \sum_{m=-l}^{l} l_{lm}(\alpha_l Y_{lm}) = \lambda \sum_{l=0}^{N} \sum_{m=-l}^{l} l_{lm}(r_{lm}) \tag{17}$$

where $r_{lm} = \alpha_l Y_{lm}$.

From (17), any image under varying illumination can be represented by a finite set of r_{lm}. Going further by combining the albedo λ and r_{lm},

$$I(\mathbf{p}) \approx \lambda \sum_{l=0}^{N} \sum_{m=-l}^{l} l_{lm}(r_{lm}) = \sum_{l=0}^{N} \sum_{m=-l}^{l} l_{lm}(\lambda r_{lm}) = \sum_{l=0}^{N} \sum_{m=-l}^{l} l_{lm}(b_{lm}(\mathbf{p})) \tag{18}$$

where $b_{lm}(\mathbf{p}) = \lambda r_{lm}$ is the harmonic image. Therefore, any image of an object under varying illumination can be efficiently approximated as a linear combination of harmonic images.

As a consequence of Figure 16, the first nine harmonic images are enough to reconstruct any image under arbitrary lighting, i.e., $I(\mathbf{p}) = \sum_{i=0}^{8} \alpha_i b_i(\mathbf{p})$. The

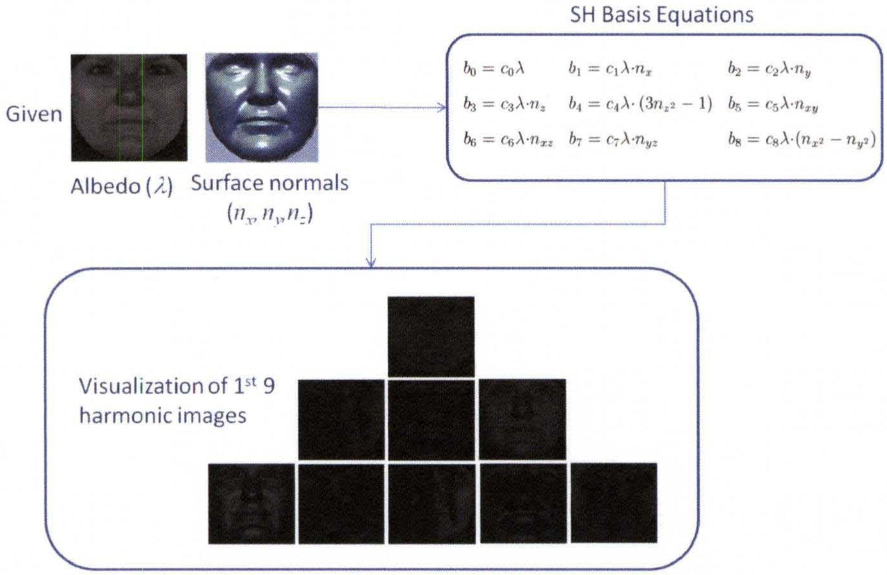

SH Basis Equations
$b_0 = c_0\lambda$
$b_1 = c_1\lambda\cdot n_x$
$b_2 = c_2\lambda\cdot n_y$
$b_3 = c_3\lambda\cdot n_z$
$b_4 = c_4\lambda\cdot(3n_{z^2} - 1)$
$b_5 = c_5\lambda\cdot n_{xy}$
$b_6 = c_6\lambda\cdot n_{xz}$
$b_7 = c_7\lambda\cdot n_{yz}$
$b_8 = c_8\lambda\cdot(n_{x^2} - n_{y^2})$
Given
Albedo (λ)
Surface normals (n_x, n_y, n_z)
Visualization of 1st 9 harmonic images

FIGURE 17 – Visualization of first nine harmonic images. Given the albedo, λ, and surface normals (derived from the 3D shape), $\mathbf{n} = (n_x, n_y, n_z)$, of a certain subject, the nine harmonic images can be computed easily using (19). The harmonic images are arranged in a pyramidal form similar to Figure 13.

actual equations for the nine harmonic images, independently derived by both Basri and Jacobs [38] and Ramamoorthi [39], are

$$\begin{array}{lll} b_0 = c_0\lambda & b_1 = c_1\lambda\cdot n_x & b_2 = c_2\lambda\cdot n_y \\ b_3 = c_3\lambda\cdot n_z & b_4 = c_4\lambda\cdot\left(3n_{z^2} - 1\right) & b_5 = c_5\lambda\cdot n_{xy} \\ b_6 = c_6\lambda\cdot n_{xz} & b_7 = c_7\lambda\cdot n_{yz} & b_8 = c_8\lambda\cdot(n_{x^2} - n_{y^2}) \end{array} \tag{19}$$

where the surface normal is $\mathbf{n} = (n_x, n_y, n_z)$, $(\cdot)$ is a component-wise operator, $n_{x^2} = n_x \cdot n_x$, $n_{xy} = n_x \cdot n_y$, and c_i's are constants [38]. Figure 17 shows how these nine harmonic images look like.

The rest of the chapters in this document contain predominantly model-based algorithms. The models are built using the USF HumanID 3D Face Database [13], which contains 100 subjects of various nationalities and gender. The next chapter contains a comprehensive summary about the description and usage of this database. For now, it is sufficient to know that each subject in the database contains albedo (texture) and shape information. For each subject in the USF database,

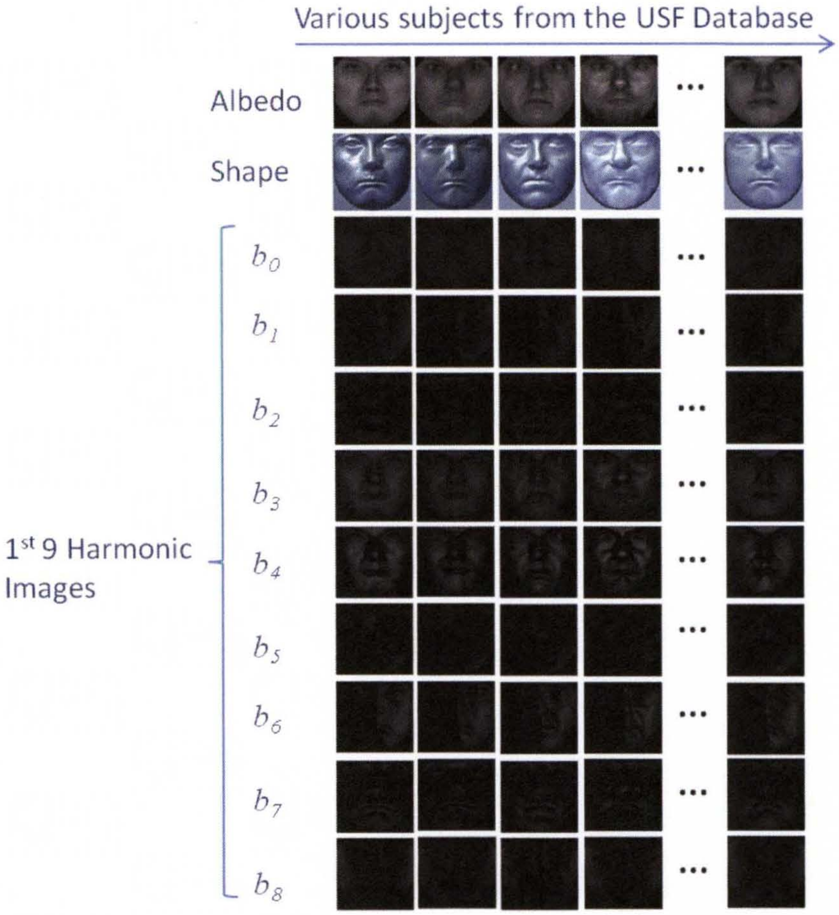


FIGURE 18 – Visualization of first nine harmonic images of various subjects from the USF Database [13]. Given the albedo, λ, and surface normals (derived from the 3D shape), $\mathbf{n} = (n_x, n_y, n_z)$, of a certain subject, the nine harmonic images can be computed easily using (19). The harmonic images are arranged in a pyramidal form similar to Figure 13.

we can generate nine harmonic images, using the process described in Figure 17. Figure 18 visualizes the hamonic images of several USF database subjects.

4. Spherical Harmonic Projection (SHP) Images

In Figure 15, several reconstructions of the original function (Figure 14) are shown, using the first n SH basis functions visualized in Fig. 13. One can extend this process to images of unknown illumination. The goal is to reconstruct the input image by projecting it first to the nine harmonic basis images (i.e., solve for the harmonic coefficients) and then taking the reverse process by summing the scaled basis images (using the computed harmonic coefficients).

In matrix notation, let the input image, I, be a $(d \times 1)$ image vector with d pixels and $\mathbf{B} = [b_0, \cdots, b_8]$ be a $(d \times 9)$ matrix of basis images (at its columns). To *project* the input image to the harmonic basis images, one needs to solve for the (9×1) coefficient vector $\boldsymbol{\alpha}$ from the linear system of equations, $I = \mathbf{B}\boldsymbol{\alpha}$. This is an overdetermined system since the number of equations (d) is much greater than the number of unknowns (9). The minimal solution is obtained using Singular Value Decomposition (SVD), i.e., $\tilde{\boldsymbol{\alpha}} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T I$, where $\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, the first nine columns of $\mathbf{U}$ are used, and $\mathbf{S}$ is (9×9). After computing the coefficient vector $\tilde{\boldsymbol{\alpha}}$, the reconstructed image (otherwise known as SHP image), h, is solved using the following equation, $h = \mathbf{B}\tilde{\boldsymbol{\alpha}}$. These steps are enumerated in Algorithm **1**.

Algorithm 1 Compute Spherical Harmonics Projection (SHP) Image

```
INPUT: (a) Input image, $I$ (b) Matrix of basis images, $\mathbf{B}$
OUTPUT: SHP image, $h$
Get SVD decomposition of $B$: $[\mathbf{U},\mathbf{S},\mathbf{V}] = svd(\mathbf{B})$
Get the number of columns ($n$) of $\mathbf{B}$: $n = size(\mathbf{B}, 2)$
Retain the first $n$ columns of U: $\mathbf{U} = \mathbf{U}(:, 1:n)$
Retain the first $n$ rows and columns of S: $\mathbf{S} = \mathbf{S}(1:n, 1:n)$
Solve for the SH coefficient vector $\tilde{\alpha}$: $\tilde{\alpha} = (\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T)I$
Compute the SHP image, $h$: $h = \mathbf{B}\tilde{\alpha}$
```

Figure 19 illustrates this procedure. Notice that the reconstructed image is visually similar to the input image. Hereupon, the reconstructed image will be known as the Spherical Harmonics Projection (SHP) image.

One can extend the pipeline in Figure 19 to all subjects of the USF database (Figure 20). The SHP images, here and onwards, will be denoted as, $h_1, \cdots, h_n$. Notice that the SHP images encode the illumination of the input but retains the identity of the subject involved. This property will be exploited in the next chap-

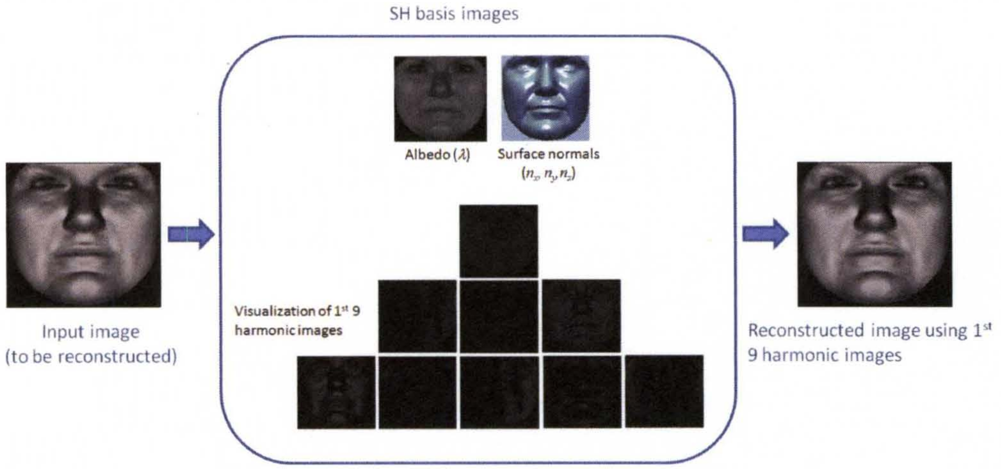

SH basis images
Albedo (λ)
Surface normals (n_x, n_y, n_z)
Input image (to be reconstructed)
Visualization of 1st 9 harmonic images
Reconstructed image using 1st 9 harmonic images

FIGURE 19 – Spherical Harmonics Projection (SHP) Images Illustration I: Input image is reconstructed by projecting it first to the nine harmonic basis images (i.e., solve for the coefficients) and then taking the reverse process by summing the scaled basis images (using the computed coefficients). The reconstructed image (now called SHP image) is visually similar to the input image.

ters.

C. Robust Spherical Harmonics Projection (SHP) Images

The previous section discussed how to generate Spherical Harmonics Projection (SHP) images, given an input image I and a matrix of basis images $\mathbf{B}$. There are instances, however, when input images are corrupted by some form of non-ideal situations, e.g., noise and occlusion (Figure 21). These non-ideal situations can be viewed, in a statistical context, as *outliers*.

The author proposes to use the field of *robust statistics* to solve this problem of estimating the coefficients when outliers exist in the input image. The main objective of *robust statistics* [48] is to recover the structure that best fits the majority of the data and, at the same time, identifying and rejecting *outliers* or *deviating substructures*. There is a growing interest in the use of robust statistics in the literature [49] [50] to cope with instances wherein computer vision models were not designed. Robust statistical methods have been applied before to the optical flow estimation problem with multiple image motions [43].

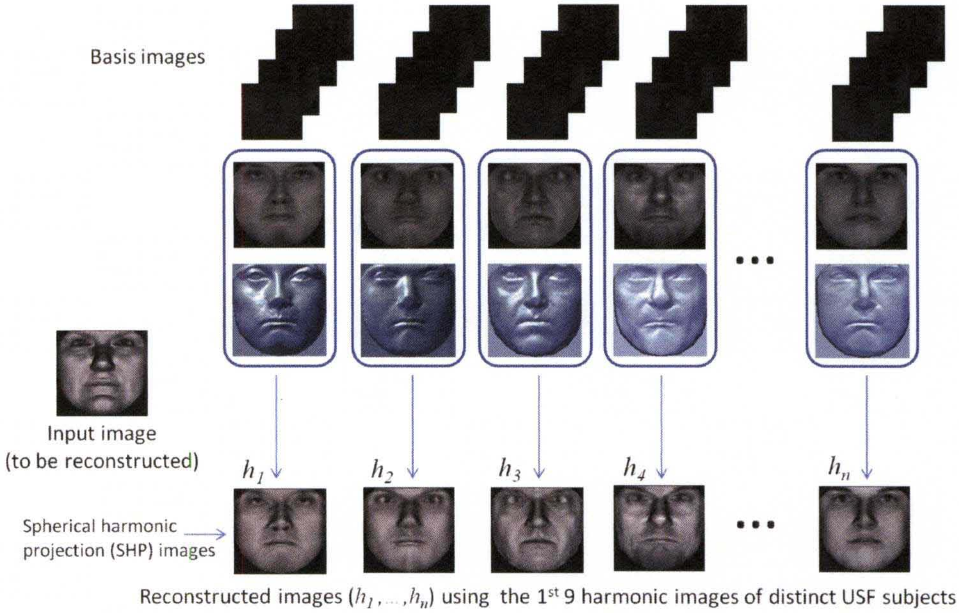
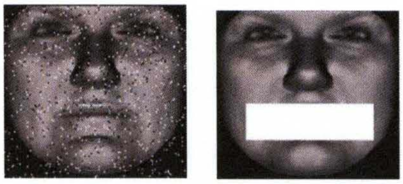
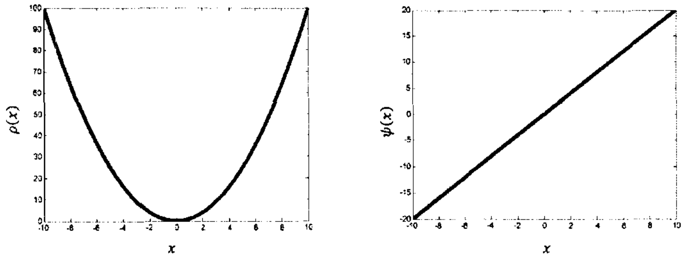

Basis images
Input image
(to be reconstructed)
h_1
h_2
h_3
h_4
h_n
Spherical harmonic projection (SHP) images
Reconstructed images ($h_1, \ldots, h_n$) using the 1st 9 harmonic images of distinct USF subjects

FIGURE 20 – Spherical Harmonics Projection (SHP) Images Illustration II: The pipeline in Figure 19 is extended to all subjects of the USF database. Notice that the SHP images encode the illumination of the input but retains the identity of the subject involved.



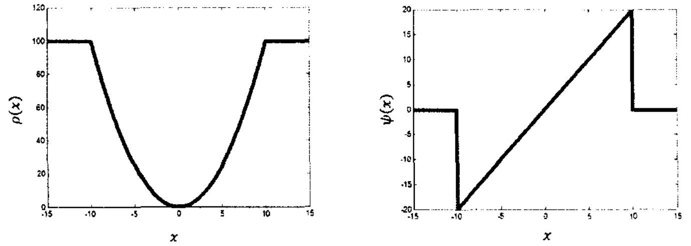
(a) (b)

FIGURE 21 – Input image corruped with (a) salt & pepper noise and (b) occlusion.



(a) (b)

FIGURE 22 – Quadratic ρ and ψ functions: (a) $\rho(x) = x^2$ and (b) $\psi(x) = 2x$.



(a) (b)

FIGURE 23 – Visualization of the truncated quadratic (a) ρ and (b) ψ functions.

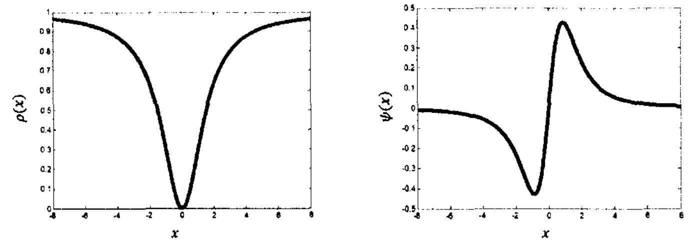
1. Robust Estimation

Robust estimation [48] addresses the problem of solving for the parameters $\mathbf{a} = (a_0, \cdots, a_n)$ that best fits the model, $\mathbf{u}(s; \mathbf{a})$, to a set of data measurements, $\mathbf{d} = \{d_0, \cdots, d_s\}$, in conditions where the data differs statistically from the model assumptions. To fit a model, the goal is to compute the parameters $\mathbf{a}$ that minimize the residual errors, i.e.,

$$\min_{\mathbf{a}} \sum_{s \epsilon S} \rho(d_s - \mathbf{u}(s; \mathbf{a}), \sigma_s) \tag{20}$$

where σ_s is a scale parameter and ρ is the error norm. The minima of (20) is an *M-estimate* since this corresponds to a *maximum likelihood* estimation [51]. Different ρ-functions result into various robust estimators; the *robustness* of an estimator is related to its insensitivity to outliers [43].

When the measurement errors are normally distributed, the optimal ρ-function



(a) (b)

FIGURE 24 – Geman-McClure ρ and ψ functions: (a) $\rho(x) = \frac{x^2}{(\sigma + x^2)}$ and (b) $\psi(x) = \frac{2x\sigma}{(\sigma + x^2)^2}$.

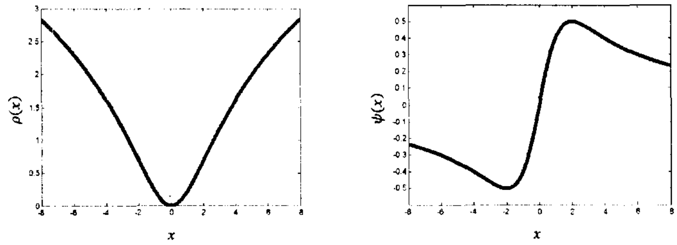
is the quadratic

$$\rho(d_s - \mathbf{u}(s; \mathbf{a}), \sigma_s) = \frac{(d_s - \mathbf{u}(s; \mathbf{a}))^2}{2\sigma_s^2} \tag{21}$$

which leads to the standard *least-squares* estimation problem. Least-squares (LS) estimation is not appropriate when outliers such as the ones visualized in Figure 21 exist. The problem with LS estimation results is that outlying measurements are assigned a large *weight* by the quadratic ρ-function, as illustrated in Figure 22(a). This is much clearer by looking at the *influence function*, ψ, associated with a particular ρ-function. The influence function describes the bias that a particular measurement has on the solution and is proportional to the derivative of the ρ-function [51] [43]. In the quadratic case (least-square sense), the influence of data points increases linearly and without bound (Figure 22(b)).

To enhance robustness, a ρ-function must lessen the effect of outlying measurements. One of the most common robust ρ-function in computer vision is the truncated quadratic (Figure 23(a)), where up to a fixed threshold, errors are weighted quadratically, but after that, errors are assigned a constant value. A closer look at the influence function (Figure 23(b)) reveals that the effect of outliers goes to zero beyond the threshold.

There are several other ρ-functions in the literature, each with different motivations and strengths [52]. Their common property, however, is their ability to



(a) (b)

FIGURE 25 – Lorentzian ρ and ψ functions: (a) $\rho(x) = \log(1+\frac{1}{2}(\frac{x}{\sigma})^2)$ and (b) $\psi(x) = \frac{2x}{2\sigma^2+x^2}$.

reduce the effect of outliers. Two examples that will be used in this work are the Geman-McClure (Figure 24) and Lorentzian ρ-functions (Figure 25). The advantage of these functions over the truncated quadratic (Figure 23) is in their differentiability that provides a more gradual transition between inliers and outliers than does the truncated quadratic.

2. Robust Estimation Framework for Spherical Harmonic Projection (SHP) Images

An earlier section showed how to compute SHP images (Algorithm **1**), given an input image, I, and a matrix of basis images, $\mathbf{B}$. In matrix notation, let the input image, I, be a $(d \times 1)$ image vector with d pixels and $\mathbf{B} = [b_0, \cdots, b_8]$ be a $(d \times 9)$ matrix of basis images (at its columns). To *project* the input image to the harmonic basis images, one needs to solve for the (9×1) coefficient vector $\boldsymbol{\alpha}$ from the linear system of equations, $I = \mathbf{B}\boldsymbol{\alpha}$. This system of equations can be translated into the robust estimation framework of (20) with the following minimization

$$\min_{\boldsymbol{\alpha}} E_D(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}} \sum_{s \in S} \rho(I - \mathbf{B}\boldsymbol{\alpha}, \sigma_s) \tag{22}$$

where σ_s is a scale parameter and ρ is the robust ρ-function. After computing the coefficient vector $\tilde{\boldsymbol{\alpha}}$, the reconstructed image (now known as the SHP image), $\tilde{I}$, is solved using the following equation, $\tilde{I} = \mathbf{B}\tilde{\boldsymbol{\alpha}}$.

There are various optimization methods that can be used to recover the coefficient vector α from the robust formulation in (22). In general, robust formulations do not have closed-form solutions and have objective functions that are nonconvex in nature. The deterministic continuation methods used in [43] will be utilized in this work to solve the robust framework in (22).

The two robust ρ-functions chosen for this work (i.e., Geman-McClure and Lorentzian) are both twice differentiable and any gradient descent approach such as *simultaneous over-relaxation* can be used to find the local minima. These ρ-functions also have scale parameters σ_s, which makes it attractive to use continuation methods.

3. Simultaneous Over-Relaxation

Simultaneous over-relaxation (SOR) belongs to a family of relaxation algorithms that include both Jacobi's method and Gauss-Seidel method [53] [54]. The iterative update equations for minimizing (22) with respect to the nth coefficient, α_n, is

$$\alpha_n^{i+1} = \alpha_n^i - \omega \frac{1}{T(\alpha_n)} \frac{\partial E}{\partial \alpha_n} \tag{23}$$

where $0 < \omega < 2$ is an overrelaxation parameter that is used to overcorrect the estimate of α_n^{i+1}. The term $T(\alpha_n)$ is an upper bound on the second partial derivative of E_D, i.e., $T(\alpha_n) \geq \frac{\partial^2 E_D}{\partial \alpha_n^2}$.

4. Graduated Non-Convexity

Continuation methods can be used to find a globally optimal solution of a nonconvex objective function such as the robust ρ-functions mentioned above. The main idea is to construct a convex approximation out of the nonconvex objective function and minimize it with the SOR technique (or any descent approach). Successively better convex approximations of the original objective function are then constructed and minimized starting from the solution of the previous approximation. The challenge in these type of approaches is in the construction of the sequence of approximations.

Blake and Zisserman [55] developed the Graduated Non-Convexity algorithm, a type of continuation method that constructs a parameterized piecewise polynomial approximation to the truncated quadratic (Figure 23). For this work, instead of using the truncated quadratic, the author utilizes both Geman-McClure and Lorentzian ρ-functions and create convex approximations out of them.

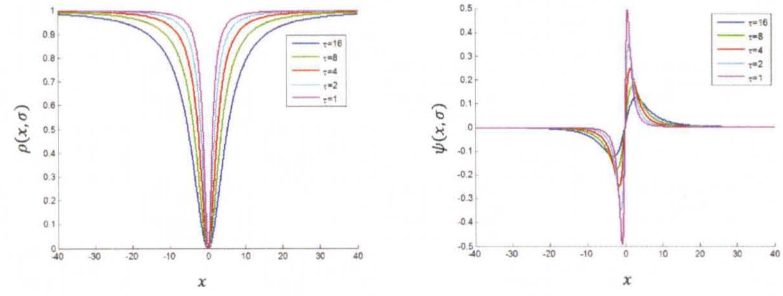
Formally, the objective function E is convex when the *Hessian* matrix, H, of E is positive definite. Furthermore, E_D is locally convex when $\rho(x)'' \geq 0$, i.e., σ_s is chosen such that there are no outliers [43]. Measurements greater than a set threshold, τ, can be considered outliers, i.e., ρ-functions begin to reduce the influence of measurements.

To construct a convex approximation of $E_D(\boldsymbol{\alpha})$, all measurements are to be treated as inliers. The point at which the influence of outliers first begins to decrease as the magnitude of the residuals increases from zero occurs when the second derivative of the ρ-function is zero. For the Lorentzian (Figure 25), the second derivative is

$$\frac{\partial^2 \rho}{\partial x^2} = \frac{\partial \psi}{\partial x} = \frac{2(2\sigma^2 - x^2)}{(2\sigma^2 + x^2)^2} \tag{24}$$

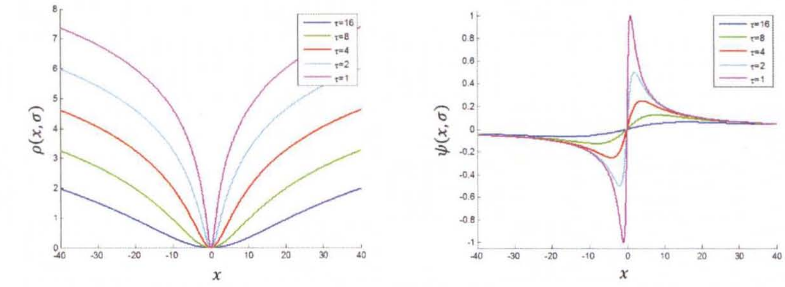
and is equal to zero when $\tau = \pm\sqrt{2}\sigma$. If the maximum expected residual is τ, then choosing $\sigma = \frac{\tau}{\sqrt{2}}$ will result into a convex optimization problem. Similarly, for the Geman-McClure ρ-function (Figure 24), $\sigma = \tau\sqrt{3}$ will lead to a convex optimization problem [43]. Notice that these τ's can be used as a simple test whether to consider a residual to be an outlier. In the case of the Lorentzian, a residual x is an outlier if $|x| \geq \sqrt{2}\sigma$.

The minimization commences with the convex optimization and the result-



(a) (b)

FIGURE 26 – Graduated Non-Convexity. Geman-McClure $\rho(x, \sigma)$ and $\psi(x, \sigma)$ plotted for thresholds $\tau \epsilon \{16, 8, 4, 2, 1\}$. (a) Error measure, $\rho(x, \sigma)$. (b) Influence, $\psi(x, \sigma)$.



(a) (b)

FIGURE 27 – Graduated Non-Convexity. Lorentzian $\rho(x, \sigma)$ and $\psi(x, \sigma)$ plotted for thresholds $\tau \epsilon \{16, 8, 4, 2, 1\}$. (a) Error measure, $\rho(x, \sigma)$. (b) Influence, $\psi(x, \sigma)$.

ing estimate will contain no outliers. This is similar to getting the least-squares estimate. Outliers can be gradually introduced when the value of σ is lowered and the minimzation is repeated. Figure 26 shows the Geman-McClure error function and its ψ-function for different values of $\tau = \frac{\sigma}{\sqrt{3}}$. Figure 27 illustrates the Lorentzian error function and its corresponding ψ-function for various values of $\tau = \sqrt{2}\sigma$.

D. Robust SHP Image Experiments

The next step is to solve for the coefficient vector $\boldsymbol{\alpha}$ in (22) using the robust estimation framework from the previous section. The term $E_D(\boldsymbol{\alpha})$ in (22) can be expanded in terms of the elements of $\boldsymbol{\alpha}$, i.e.,

$$E_D(\boldsymbol{\alpha}) = \sum_{p \in P} \rho(I(p) - \alpha_0 b_0(p) - \cdots - \alpha_8 b_8(p), \sigma_s) \tag{25}$$

where p refers to pixel positions. The iterative update equations (23) for the minimization of (25) are repeated here for convenience, i.e., $\alpha_n^{i+1} = \alpha_n^i - \omega \frac{1}{T(\alpha_n)} \frac{\partial E_D}{\partial \alpha_n}$. The term $\frac{\partial E_D}{\partial \alpha_n}$ is actually

$$\frac{\partial E_D}{\partial \alpha_n} = \sum_{p \in P} -b_n(p) \psi(I(p) - \alpha_0 b_0(p) - \cdots - \alpha_8 b_8(p), \sigma_s) \tag{26}$$

The term $T(\alpha_n)$ in the iterative update equation is an upper bound on the second partial derivative of E_D, i.e., $T(\alpha_n) \geq \frac{\partial^2 E_D}{\partial \alpha_n^2}$. The term $\frac{\partial^2 E_D}{\partial \alpha_n^2}$ is

$$\frac{\partial^2 E_D}{\partial \alpha_n^2} = \sum_{p \in P} b_n^2(p) \psi'(I(p) - \alpha_0 b_0(p) - \cdots - \alpha_8 b_8(p), \sigma_s) \tag{27}$$

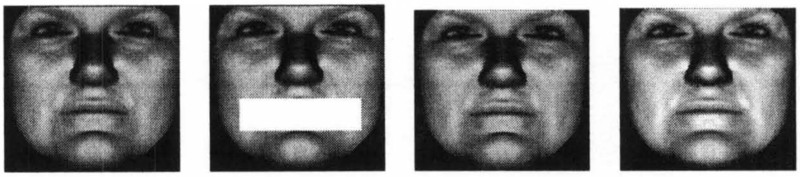
Therefore, $T(\alpha_n) = \max \psi'(x) \sum_p b_n^2(p)$. For the Geman-McClure estimator, $\psi(x, \sigma) = \frac{2x\sigma}{(\sigma + x^2)^2}$ and $\max \psi'(x, \sigma) = \frac{2}{\sigma^2}$, which leads to $T(\alpha_n) = \frac{2}{\sigma^2} \sum_p b_n^2(p)$. Likewise, for the Lorentzian estimator, $\psi(x) = \frac{2x}{2\sigma^2 + x^2}$ and $\max \psi'(x, \sigma) = \frac{1}{\sigma^2}$, which takes us to $T(\alpha_n) = \frac{1}{\sigma^2} \sum_p b_n^2(p)$. The complete algorithm to compute robust SHP images is shown in Algorithm **2**.

1. Results from Least-Squares Estimation

Using Algorithm **1**, a form of least-squares estimation, one can reconstruct Figure 21(b) using the basis images found in Figure 19. Figure 28 shows the least-squares estimation results. There is no problem in using the LS approach when reconstructing an input image without any outlier (Figure 28(c)), i.e., $\|I_g - \tilde{I}_g\| = 7.15 \times 10^{-013}$, where I_g is the input image and $\tilde{I}_g$ is the reconstructed image. However, when the occluded version of the input image is reconstructed, the result ($\tilde{I}_{gc}$) appears to be brighter than the original input image (Figure 28(d)), i.e., $\|I_g - \tilde{I}_{gc}\| = 22.64$. The robust estimation framework discussed in this chapter is supposed to

Algorithm 2 Compute Robust Spherical Harmonics Projection (SHP) Image

```
INPUT: (a) Input image, $I$ (b) Matrix of basis images, $\mathbf{B}$
OUTPUT: SHP image, $h$
1: Initial SH coefficients: Solve for $\boldsymbol{\alpha}^0$ using the least-squares estimate of $I = \mathbf{B}\boldsymbol{\alpha}^0$, i.e., Algorithm 1
2: Compute partial derivatives: Compute $\{\frac{\partial E_D}{\partial \alpha_n}\}_{n=0,\cdots,8}$
3: SH coefficient vector update: Update using $\alpha_n^{i+1} = \alpha_n^i - \omega \frac{1}{T(\alpha_n)} \frac{\partial E_D}{\partial \alpha_n}$, where $\omega$ is the overrelaxation parameter, $T(\alpha_n)$ is the upper bound of $\frac{\partial^2 E_D}{\partial \alpha_n^2}$
4: Robust function parameter update (Graduated Non-Convexity): $\sigma_s^{i+1} = k\sigma_s^i$, with $k \in (0,1)$
5: Repeat steps 2-4 until termination criterion is satisfied.
6: Test for outliers: Determine outlying measurements using, $|I(p) - \alpha_0 b_0(p) - \cdots - \alpha_8 b_8(p)| \geq \tau$, where $\tau$ is determined by the error norm and the control parameter, $\sigma$
7: Compute SH coefficients for non-outlier pixels: Solve for $\tilde{\alpha}$ in $I = \mathbf{B}\tilde{\alpha}$ using Algorithm 1 for non-outlier pixels
8: Compute the SHP image: $h = \mathbf{B}\tilde{\alpha}$
```



(a) (b) (c) (d)

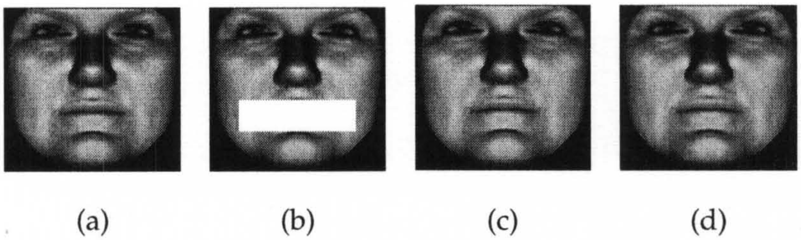
FIGURE 28 – Least-square (LS) estimation results for the reconstruction of Fig. 21(b) using groundtruth SH basis images. (a) Input image, I_g. (b) Occluded version of input image, I_{gc}. (c) LS reconstruction of the input, $\tilde{I}_g$ (d) LS reconstruction of the occluded input, $\tilde{I}_{gc}$.

overcome this problem; even in the presence of outliers, the resulting reconstruction $\tilde{I}_{gc}$ should be as close to the input image, I_g, i.e., $\|I_g - \tilde{I}_{gc}\| \approx 0$.

2. Results from Robust Estimation

The robust estimation framework for computing SHP images, even with the presence of outliers, is compactly described in Algorithm **2**. The parameters are set as follows: $\omega = 1.995$, $k = 0.95$, $\sigma_s = 1$. Figure 29 shows the robust estimation results for the Geman-McClure norm. In contrast to Figure 28(d), the reconstruction in Figure 29(d) is very close to the input image, i.e., $\|I_g - \tilde{I}_{gc}\| = 5.71 \times 10^{-014}$. The occlusion estimate in Algorithm **2** (Step 6) is given in Figure 30, which is visually close enough to the input occlusion in Figure 29(b). These Geman-McClure estimation results are done within 30 iterations.

Figure 29 shows the robust estimation results for the Lorentzian norm at 50 iterations. Similar to Fig. 29(d), the reconstruction in Figure 31(d) is very close to the input with $\|I_g - \tilde{I}_{gc}\| \approx 0$. However, as opposed to the Geman-McClure error norm, the occlusion estimate in Figure 32 is not that visually accurate. The reason may lie in the smooth and undefined transition between inliers and outliers for the Lorentzian error norm (Figure 27), compared to the defined transition of the Geman-McClure norm (Figure 26). For this reason, robust algorithms for the rest of this work will refer to the Geman-McClure ρ-function.



(a) (b) (c) (d)

FIGURE 29 – Robust estimation (Geman-McClure) results for the reconstruction of Figure 21(b) using groundtruth SH basis images. (a) Input image, I_g. (b) Occluded version of input image, I_{gc}. (c) LS reconstruction of the input, $\tilde{I}_g$ (d) Robust reconstruction (Geman-McClure) of the occluded input, $\tilde{I}_{gc}$.

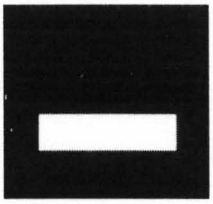
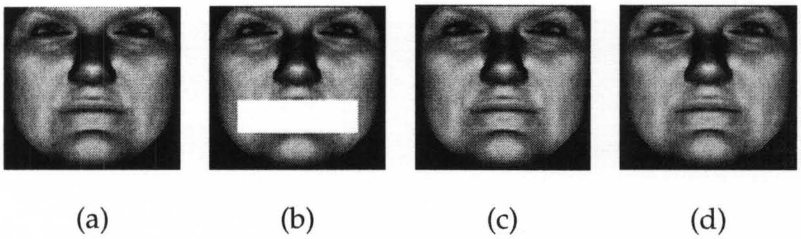


FIGURE 30 – Occlusion estimate of Algorithm **2** (Step 6) using the Geman-McClure ρ-function, expressed as a binary image. White pixels indicate outliers.



(a) (b) (c) (d)

FIGURE 31 – Robust estimation (Lorentzian) results for the reconstruction of Fig. 21(b) using groundtruth SH basis images. (a) Input image, I_g. (b) Occluded version of input image, I_{gc}. (c) LS reconstruction of the input, $\tilde{I}_g$ (d) Robust reconstruction (Lorentzian) of the occluded input, $\tilde{I}_{gc}$.

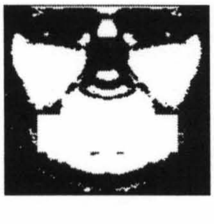


FIGURE 32 – Occlusion estimate of Algorithm **2** (Step 6) using the Lorentzian ρ-function, expressed as a binary image. White pixels indicate outliers.

The reconstructions in Figures 28, 29, and 31 use the basis images of Figure 19, which are essentially derived from the ground-truth shape and albedo of the input image. It is interesting to see the robust estimation results for reconstructions using a different subject's shape and albedo, similar to the one performed in Figure 20. Figure 33 shows the reconstruction results when the basis images are computed from other subjects' shape and albedo. The first and second rows contain the original input and its occluded version, respectively. The albedo and shape of various subjects (from where the basis images are derived (19)) are found in the third and fourth rows. The fifth and sixth rows show the reconstruction of the original input image (without occlusion) and its occluded version using the least-squares estimate. Notice that the sixth row appears brighter than the fifth, similar to Figure 28. The seventh row shows the reconstruction of the occluded input using robust estimation (Geman-McClure). The last row shows the occlusion estimate expressed as a binary image, with white pixels indicating outliers.

TABLE 1
NUMERICAL VALUES FOR RECONSTRUCTION ERRORS WITH NON-GROUNDTRUTH SH BASIS IMAGES

Case	Subject 1	Subject 2	Subject 3	Subject 4	$\cdots$	Subject n
I (LS, $\|\|I_g - \tilde{I}_g\|\|$)	12.39	11.95	13.01	14.91	$\cdots$	12.65
II (LS, $\|\|I_g - \tilde{I}_{gc}\|\|$)	17.52	16.74	18.38	18.49	$\cdots$	17.80
III (Robust, $\|\|I_g - \tilde{I}_{gc}\|\|$)	13.90	12.68	14.75	15.60	$\cdots$	13.45

Table 1 shows numerical values for several cases of reconstruction errors in Figure 33. Normally, when the ground-truth basis images are used to reconstruct an input image without occlusion, the reconstruction error is close to zero, i.e., $||I_g - \tilde{I}_g|| \approx 0$, where I_g is the input image and $\tilde{I}_g$ is the reconstructed image. However, when a different subject's shape and albedo are used to create the basis images, the reconstruction error, with respect to an input image without occlusion, is still minimum (Algorithm 1) but much greater than zero, i.e., $||I_g - \tilde{I}_g|| \gg 0$.

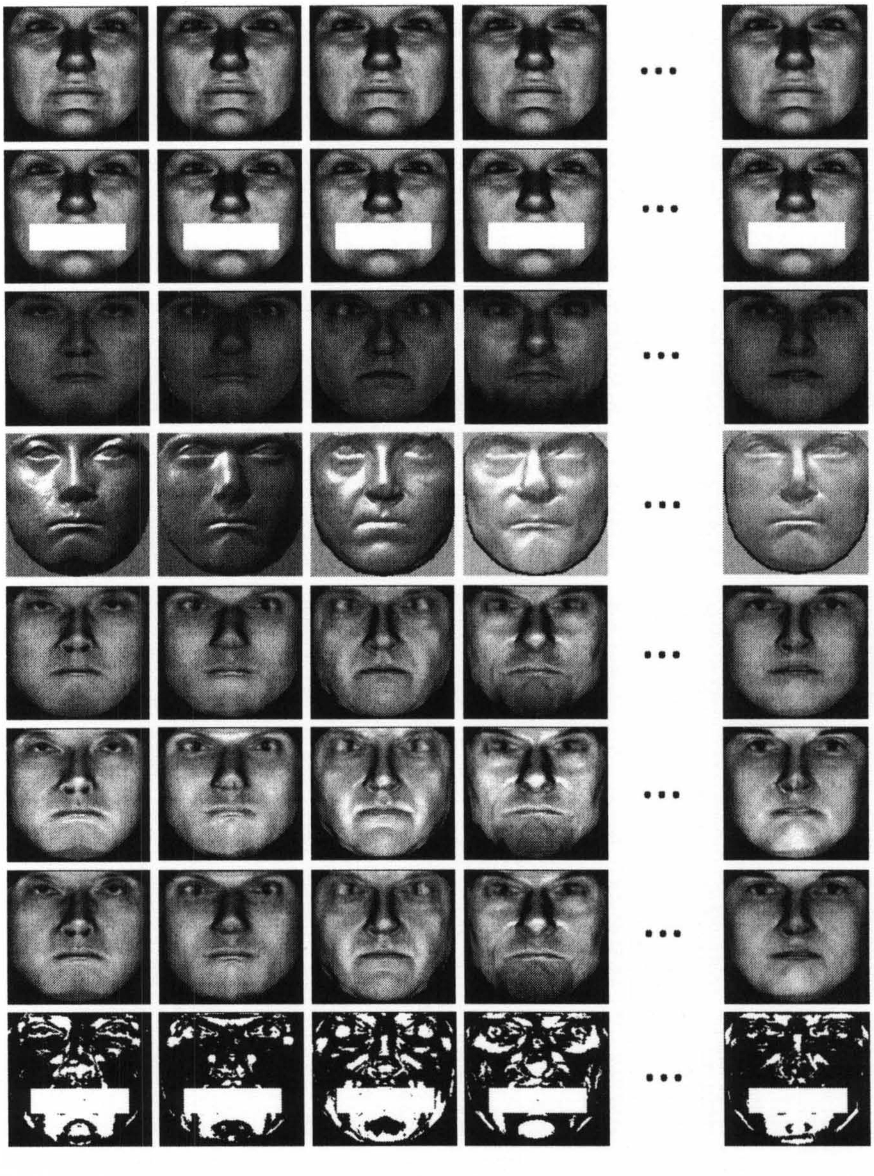


FIGURE 33 – Reconstruction results when basis images are computed from other subjects' shape and albedo. First and second rows contain the original input and its occluded version, respectively. Albedo and shape of various subjects are found in the third and fourth rows. Fifth and sixth rows show the reconstruction of the input image and its occluded version using the least-squares estimate. Seventh row shows the reconstruction of the occluded input using robust estimation. Last row shows the occlusion estimate, with white pixels indicating outliers.

The reconstruction error of the fifth row (LS estimate) of Figure 33 with respect to the first row (input image, without occlusion) is shown as the first row (Case I) in Table 1. Notice the numerical values being much greater than zero.

The second row (Case II) in Table 1 lists the reconstruction error of the sixth row in Figure 33 (reconstruction using LS of the occluded input) with respect to the first row (original input image). The numerical values are greater than the first row, as expected, since the outliers are not taken into account. The third row (Case III) of Table 1 shows the reconstruction error of the seventh row in Figure 33 (reconstruction using robust estimation of the occluded input) with respect to the first row (original input image). The numerical values are now lower than the second row and is closer to the first row since outliers in occluded input (seventh row) are taken into consideration.

E. Summary

This chapter started with basic definitions and equations related to spherical harmonics. Spherical harmonics (SH) are basically the extension of the Fourier basis of linear functions to the sphere. Any piecewise continuous spherical function can be approximated as a linear combination of a finite set of spherical harmonics. A sample spherical function was reconstructed using the first n SH basis functions to illustrate the utility of spherical harmonics.

Derivations were shown along the way to show that any image of a Lambertian object under unknown illumination can be compactly represented by a linear combination of a finite number of harmonic images. A sample input image was reconstructed using harmonic basis images of several individuals. There is a unique reconstruction from the input image for each individual. One important property of the reconstructed images is that it captures the illumination of the input image but retains the identity of the individual (Figure 20).

This chapter also discussed how to compute SHP images when the input image has outliers present. The algorithm involved should be robust enough to recover the SHP image as if there were no outliers present. To solve this problem, this chapter proposed to use the field of robust statistics to reformulate the equations (22) needed to get the SHP images. Specifically, the original objective function is incorporated into a robust ρ-function, which reduces the effect of outliers.

Several experiments were performed involving reconstructions of occluded input images using both groundtruth (Figures 28, 29, and 31) and non-groundtruth (Fig. 33) SH basis images. The robust estimation framework, in all cases, gives more accurate reconstructions compared to its least-squares variant (Algorithm 1).

CHAPTER III

MODEL-BASED SHAPE RECOVERY FROM SINGLE IMAGES OF GENERAL AND UNKNOWN LIGHTING AND POSE

This chapter proposes a new statistical shape-from-shading framework for images of unknown illumination and pose. The object (e.g., face) to be reconstructed is described by a parametric model. To deal with arbitrary illumination, the framework makes use of recent results that general lighting can be expressed using low-order spherical harmonics for convex Lambertian objects. The classical shape-from-shading (SFS) equation is modified according to this framework.

Before going to a complete SFS framework for general illumination and pose, this chapter starts first with the simpler case of unknown illumination and frontal pose. Three algorithms are described to solve this SFS problem, starting from a brute-force iterative approach to the efficient regression method. The findings of these three methods will be used to generalize the SFS framework to general illumination and pose. In addition, the robust version of SHP images from the previous chapter will be incorporated to the SFS framework to deal with non-ideal situations.

A. Model Construction

The starting point of any model-based approach is the construction of the model itself. The SFS framework in this chapter involves two models, namely, the shape and texture models. The process of constructing the shape and texture models for this model-based framework involves two major steps: (a) establishing a dense correspondence between database samples and (b) statistical modeling. The database samples are assumed to be globally aligned to each other, similar to [35], which negates the need to perform 3D shape alignment. The shape and texture model construction in this dissertation is similar to the morphable model construction in [35] [56].

1. USF Database of 3D Scans

The shape and texture models in this paper will be derived from 3D scans of the USF database. Originally, there are 100 3D scans in that database of various gender and race. Out of these 100 samples, 80 subjects were deemed to be acceptable and were subsequently chosen to build the shape and texture models.

The original data from the 3D laser scans, which is metric in nature, represent the face shape in terms of cylindrical coordinates relative to a vertical axis centered with respect to the head [35]. Figure 34 shows an illustration of a sample 3D scan. In 512 angular steps, ϕ covers $0 - 360°$ and there are 512 vertical steps h at a spacing of 0.615 mm. At each grid position (h, ϕ), the laser scans provide four measurements related to the radius r as well as the red, green and blue (RGB) components for texture information. Therefore, a sample scan **I** can be represented as

$$\mathbf{I} = (r(h,\phi), R(h,\phi), G(h,\phi), B(h,\phi))^T \qquad h, \phi \in \{0, \cdots, 511\} \tag{28}$$

If Cartesian coordinates are desired, it is straightforward to compute the equivalent (X, Y, Z) coordinates from the original cylindrical (r, h, ϕ) data. In addition to the raw 3D scans, the USF database contains the 3D morphable faces [35] derived from the raw scans. Specifically, it contains the mean shape and texture, shape and texture eigenvectors, and a triangle list for rendering purposes. The starting point for building the shape and texture models in this thesis is not the raw original data from the USF database. Instead, the original 3D scans are converted into a Monge patch format. A Monge patch [37] represents the surface as

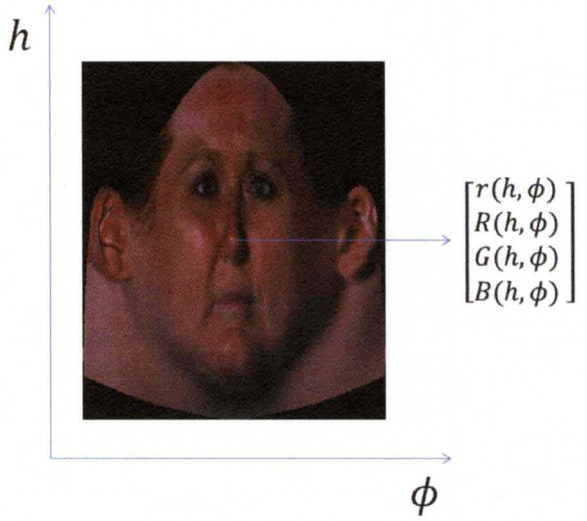

h
ϕ
$\begin{bmatrix} r(h,\phi) \\ R(h,\phi) \\ G(h,\phi) \\ B(h,\phi) \end{bmatrix}$

FIGURE 34 – Sample 3D laser scan parameterized by cylindrical coordinates, (h, ϕ).

$(x, y, f(x, y))$. This representation is convenient because a unique point on the surface can be determined by specifying only the image coordinates.

Castelan et al. [1] uses the term, height map, for a similar concept as the Monge patch. The height map can be formally defined as the function $Z(x, y)$, where the (x, y) coordinates represent a position in the image plane. The texture map can be similarly expressed as $T(x, y)$, where each function value can be in the grayscale or RGB format.

To get the height map and the corresponding image data, the front (for image information) and depth buffers (for height information) within the frame buffer in a computer graphics system are simulated [57]. If the 3D face data is rendered using OpenGL, built-in functions in OpenGL can readily extract the front and depth buffers [58].

This work will use the object-oriented approach [57], which has an outer loop of the form: *for(each object), render(object)*. The objects in question here are the triangles in the triangle list of the 3D morphable model. The triangles undergo a geometric transformation and affect any pixels in the frame buffer during the rasterization process.

Geometric Transformation Denote as $\mathbf{v}$ the (X, Y, Z) coordinates of a vertex of 3D shape $\mathbf{s}_{3D}$. The first rigid transformation is a translation by $\mathbf{t}_{3D}$ and a rotation about the X, Y, and Z axes with angles ϕ, γ, and θ, respectively. The translation $\mathbf{t}_{3D}$ sets the origin at the center of the 3D shape, i.e., $\mathbf{t}_{3D} = \sum_j \mathbf{s}_{3D,j}$. The result of both rotation and translation is

$$\mathbf{w} = R_\phi R_\gamma R_\theta (\mathbf{v} + \mathbf{t}_{3D}) \tag{29}$$

The 2D image coordinates of the vertex, denoted by $\mathbf{s}_{2D}$, are solved using orthographic projection, which is acceptable when the distance from the camera to the rendered 3D shape is much larger than its dimension. After the projection, a 2D translation $\mathbf{t}_{2D}$ and scaling of f are applied

$$\mathbf{s}_{2D} = f \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \mathbf{w} + \mathbf{t}_{2D} \tag{30}$$

The previous equations can be expressed into one overall equation for all vertices, expressing the relationship between the 2D and 3D vertices

$$\mathbf{s}_{2D} = fPR(\mathbf{v} + \mathbf{t}_{3D}) + \mathbf{t}_{2D} \tag{31}$$

where $\mathbf{t}_{3D}$ and $\mathbf{t}_{2D}$ are concatenated translation vectors of length $3N$ and $2N$, respectively. R is a $3N \times 3N$ block diagonal matrix which performs the combined rotation $R_\phi R_\gamma R_\theta$ for all N vertices. P is a $2N \times 3N$ orthographic projection matrix for the set of vertices. Note that only a subset of the original set of vertices will be visible. The z-buffer algorithm [59] is used for hidden-surface removal.

Image Synthesis The previous geometric transformation mapped the vertices (described in the triangle list) from the 3D space to the image frame. To synthesize an image, an inverse mapping is done, i.e., in order to know what color value must be assigned to a pixel, it is necessary to know where this pixel is mapped into the 3D space (or reference frame using the terminology of morphable models). Figure 35 illustrates this inverse mapping. The inverse mapping

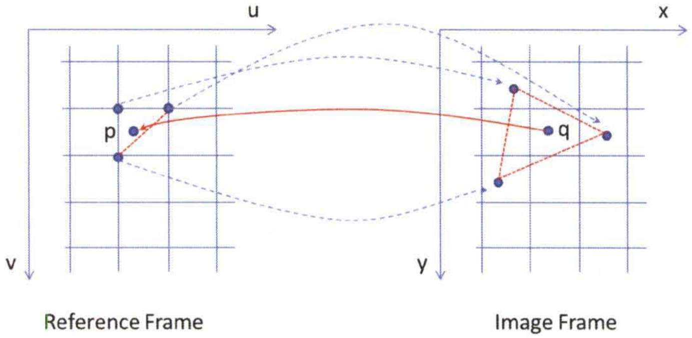


FIGURE 35 – Inverse mapping to infer pixel values in the image frame using barycentric coordinates.

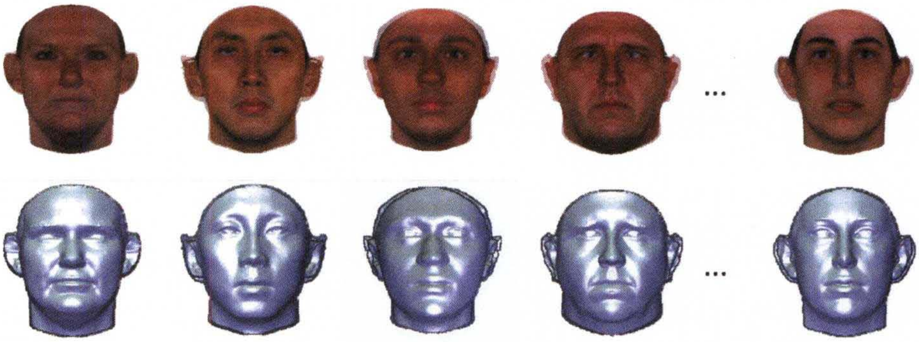


FIGURE 36 – Sample texture (top row) and height maps (bottom row) from the original USF data.

is realized using barycentric coordinates [60]. To get the corresponding position in the reference frame (p) of pixel position q, the barycentric coordinates of q are first computed. These coordinates are then used to infer the position of p in the reference frame. The interpolated color value of p in the reference frame is assigned to pixel position q.

The height map can be generated using the same process as the image data. Instead of using color values, the actual z values (depth information) is inferred from the reference frame and assigned to position q in the image frame. Figure 36 shows several height maps and corresponding image data generated from the original USF data.

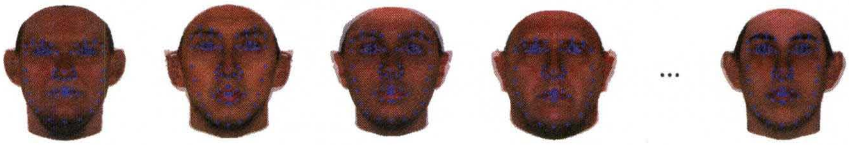


FIGURE 37– Sample annotations of the USF database. There is a total of 76 manually annotated landmark points, including both anatomical and pseudo-landmarks.

2. Dense Correspondence

To get the dense correspondence between database samples in the original morphable model framework of Blanz and Vetter [35], a modified optical flow algorithm is used. The correspondences are established by matching regions of similar texture and topography between each sample 3D scan and a reference face. The main advantage of this approach is that dense correspondence can be determined with little manual intervention.

The approach for finding dense correspondence in this dissertation follows that of Patel et al. [56]. A set of sparse landmark points, (x'_i, y'_i), are manually annotated on the resulting image data, $T(x, y)$ from the previous section (image synthesis) for all database samples. The landmark points are chosen such that they can be easily located on all database samples, e.g., anatomical landmarks [61]. In addition to the anatomical landmarks, additional pseudo-landmark points [61] are added to ensure smoothness of the facial contours. There is a total of 76 manually annotated landmark points, including both anatomical and pseudo-landmarks. Figure 37 illustrates several sample annotations of the USF database. To get the mean shape of this ensemble of 76 landmark points, Generalized Procrustes Analysis (GPA) [62] is first performed to align the set of shapes to a common reference frame. The mean shape is simply computed as

$$\mathbf{x}_0 = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \tag{32}$$

where $\mathbf{x}_i = (x_1, y_1, z_1, \cdots, x_n, y_n, z_n)^T$ is the *i*th shape after alignment and m is the total number of shapes. Since we are dealing with height maps, $\mathbf{x}_i$ can be represented as $\mathbf{x}_i = (x_1, y_1, Z(x_1, y_1), \cdots, x_n, y_n, Z(x_n, y_n))^T$.

The mean shape is crucial in establishing dense correspondence between USF samples. Figure 38(upper left) shows the mean shape together with its convex hull. A mask is created such that its pixels are within the convex hull of the mean shape. Each pixel within the mask is supposed to correspond to a certain point on each USF sample scan. Therefore, the mean shape mask exists as some form of reference frame which holds the correspondence among all USF 3D shapes. Fig. 38 illustrates this process of finding the dense correspondence. Note that $\mathbf{I}_i = (x_1, y_1, z_1, R_1, G_1, B_1, \cdots, x_k, y_k, z_k, R_k, G_k, B_k, \cdots, x_N, y_N, z_N, R_N, G_N, B_N)^T$ and there are a maximum of 100 such $\mathbf{I}_i$'s for the USF database.

To get the map from the mean shape mask to a USF sample scan, a warping function, $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, based on physically motivated thin-plate splines [63] [62] is constructed using the landmark points (e.g., (x_i, y_i) from the mean shape and another set (x_i', y_i') for each sample scan) as control points, i.e., $\mathbf{f}(\mathbf{x}) = \mathbf{x}'$, where $\mathbf{x} = [x_i, y_i]^T$ and $\mathbf{x}' = [x_i', y_i']^T$. Once the warping function is solved, this warp is applied to all pixels from the mean shape mask, determining their corresponding locations on each USF sample scan. Note that the warping operation is performed in 2D space.

A similar procedure is done between the mean shape mask and the frontal input image, as shown in Figure 38, forming a dense correspondence between the USF database samples and the input image, through the mean shape mask that serves a reference frame.

3. Statistical Modeling

The shape and texture information for each $\mathbf{I}_i$ can be separated into two separate vectors, $\mathbf{S}_i$ and $\mathbf{T}_i$, i.e.,

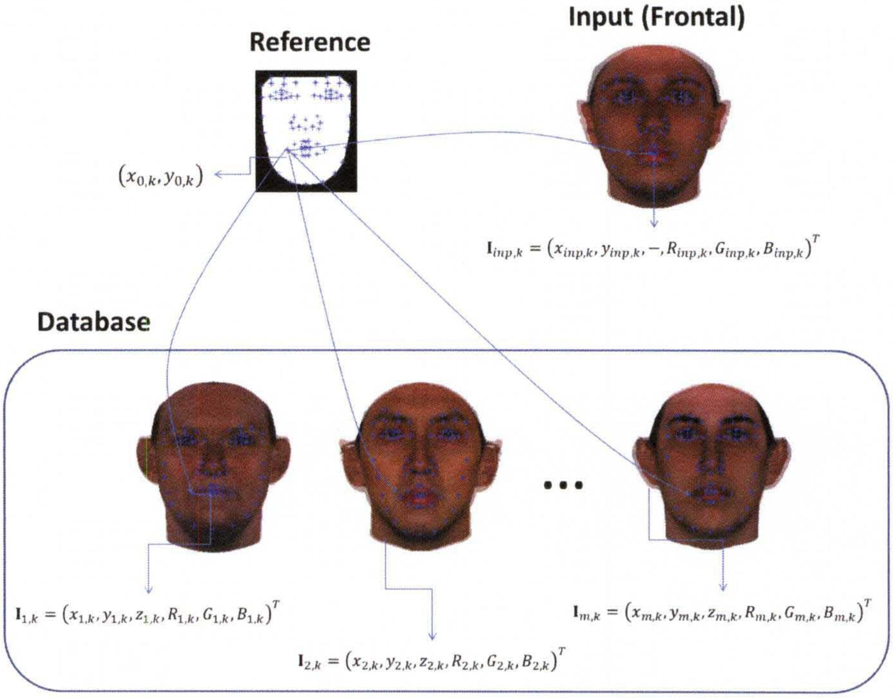

Input (Frontal)
Reference
$(x_{0,k}, y_{0,k})$
$\mathbf{I}_{inp,k} = (x_{inp,k}, y_{inp,k}, -, R_{inp,k}, G_{inp,k}, B_{inp,k})^T$
Database
...
$\mathbf{I}_{1,k} = (x_{1,k}, y_{1,k}, z_{1,k}, R_{1,k}, G_{1,k}, B_{1,k})^T$
$\mathbf{I}_{m,k} = (x_{m,k}, y_{m,k}, z_{m,k}, R_{m,k}, G_{m,k}, B_{m,k})^T$
$\mathbf{I}_{2,k} = (x_{2,k}, y_{2,k}, z_{2,k}, R_{2,k}, G_{2,k}, B_{2,k})^T$

FIGURE 38 – (**Upper Left**) Mean shape together with its convex hull. The convex hull forms the boundary of the mask. (**Lower**) Each pixel position within the mask corresponds to a certain point on a sample scan in the USF database. The vector $\mathbf{I}_{i,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$ refers to the shape and texture information at the *k*th vertex of the *i*th sample scan. (**Upper Right**)Similarly, each pixel position within the mask corresponds to a certain point on the input image. Hence, there is correspondence between the USF database samples and the input image, through the mean shape mask.

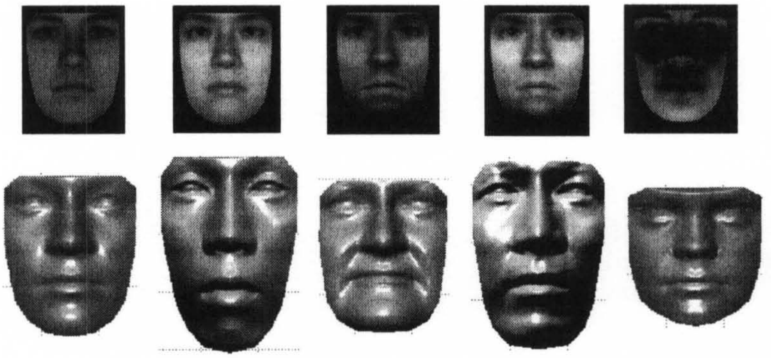


FIGURE 39 – **(Top row)** $\bar{\mathbf{t}}$, $\bar{\mathbf{t}} - 0.5\beta_1\mathbf{t}_1$, $\bar{\mathbf{t}} - 0.5\beta_2\mathbf{t}_2$, $\bar{\mathbf{t}} - 0.5\beta_1\mathbf{t}_1 - 0.5\beta_2\mathbf{t}_2$, $\bar{\mathbf{t}} + 0.5\beta_1\mathbf{s}_1 + 0.5\beta_2\mathbf{s}_2$. **(Bottom row)** $\bar{\mathbf{s}}$, $\bar{\mathbf{s}} - 0.5\sigma_1\mathbf{s}_1$, $\bar{\mathbf{s}} - 0.5\sigma_2\mathbf{s}_2$, $\bar{\mathbf{s}} - 0.5\sigma_1\mathbf{s}_1 - 0.5\sigma_2\mathbf{s}_2$, $\bar{\mathbf{s}} + 0.5\sigma_1\mathbf{s}_1 + 0.5\sigma_2\mathbf{s}_2$.

$$\mathbf{S}_i = (x_1, y_1, z_1, \cdots, x_N, y_N, z_N)^T \tag{33}$$

$$\mathbf{T}_i = (R_1, G_1, B_1, \cdots, R_N, G_N, B_N)^T \tag{34}$$

Principal Component Analysis (PCA) [64] [65] is performed on the set of shape and texture vectors, $\mathbf{S}_i$ and $\mathbf{T}_i$. For the shape information, the average ($\bar{\mathbf{s}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{S}_i$) is first subtracted from each shape vector, $\mathbf{d}_i = \mathbf{S}_i - \bar{\mathbf{s}}$, and form the data matrix $\mathbf{D} = (\mathbf{d}_1, \cdots, \mathbf{d}_m)$.

The main step of PCA is to compute the eigenvectors $(\mathbf{s}_1, \mathbf{s}_2, \cdots)$ of the covariance matrix ($\mathbf{C} = \frac{1}{m}\mathbf{A}\mathbf{A}^T$), which can be achieved using Singular Value Decomposition (SVD) [66] of $\mathbf{A}$. The eigenvalues $(\sigma_{s,i}^2)$ of $\mathbf{C}$ are related to the variance of the data across each eigenvector direction. Exactly the same procedure is done to obtain the texture eigenvectors $(\mathbf{t}_i)$ and variances $(\sigma_{t,i}^2)$. Figure 39 visualizes the PCA results. Note that the pixel positions of the texture visualization (Figure 39 (**Top row**)) are in the reference frame. The resulting shape and texture models are

$$\mathbf{s} = \bar{\mathbf{s}} + \sum_{i=1}^{m-1} \alpha_i \cdot \mathbf{s}_i, \qquad \mathbf{t} = \bar{\mathbf{t}} + \sum_{i=1}^{m-1} \beta_i \cdot \mathbf{t}_i \tag{35}$$

B. Spherical Harmonics Illumination

Under the assumption that the viewer and light source are far from the object, the image irradiance equation can be written as follows

$$E(\mathbf{p}) = R(\mathbf{n}(\mathbf{p})) \tag{36}$$

where $E(\mathbf{p})$ is the image irradiance at point $\mathbf{p}$ and $R(.)$ is the radiance of the surface patch with unit normal $\mathbf{n}(\mathbf{p})$. After scaling E with the surface albedo λ, the image intensity I at pixel $\mathbf{x}$ is $I(\mathbf{x}) = \lambda E(\mathbf{p})$. Under the framework of [38], this becomes

$$I(\mathbf{x}) = \mathbf{B}(\mathbf{x})\boldsymbol{\alpha} \tag{37}$$

which states that the that the pixel intensity at $\mathbf{x}$ is the weighted combination of the basis images $\mathbf{B}(\mathbf{x})$, where $\boldsymbol{\alpha}$ is the vector of illumination coefficients. The equations for the nine spherical harmonics basis images are

$$\begin{array}{lll} b_0 = c_0\lambda & b_1 = c_1\lambda{\cdot}n_x & b_2 = c_2\lambda{\cdot}n_y \\ b_3 = c_3\lambda{\cdot}n_z & b_4 = c_4\lambda{\cdot}\left(3n_{z^2} - 1\right) & b_5 = c_5\lambda{\cdot}n_{xy} \\ b_6 = c_6\lambda{\cdot}n_{xz} & b_7 = c_7\lambda{\cdot}n_{yz} & b_8 = c_8\lambda{\cdot}(n_{x^2} - n_{y^2}) \end{array} \tag{38}$$

where the surface normal is $\mathbf{n} = (n_x, n_y, n_z)$, $(\cdot)$ is a component-wise operator, $n_{x^2} = n_x \cdot n_x$, $n_{xy} = n_x \cdot n_y$, and c_i's are constants [38].

C. Shape-from-shading

The classic brightness constraint in shape-from-shading indicates the total brightness error of the reconstructed image compared to the input image, I [22]. The brightness constraint equation is defined as follows

$$\varepsilon = \iint \left(I\left(\mathbf{x}\right) - \lambda R\left(\mathbf{n}\left(\mathbf{x}\right)\right)\right)^2 d\mathbf{x} \tag{39}$$

The brightness constraint in the discrete case, following (37), becomes

$$\varepsilon = \sum_x (I(\mathbf{x}) - \mathbf{B}(\mathbf{x})\boldsymbol{\alpha})^2 \tag{40}$$

if SH basis images are used. In matrix notation, let I be the $(d \times 1)$ image vector with d pixels, $\mathbf{B} = [b_0(\mathbf{x}), \ldots, b_1(\mathbf{x})]$ be the $(d \times n)$ matrix of basis images (at its columns), where n is the number of basis images $(n = 4, 9)$, and $\boldsymbol{\alpha}$ the $(n \times 1)$ vector of coefficients, ε can be expressed as

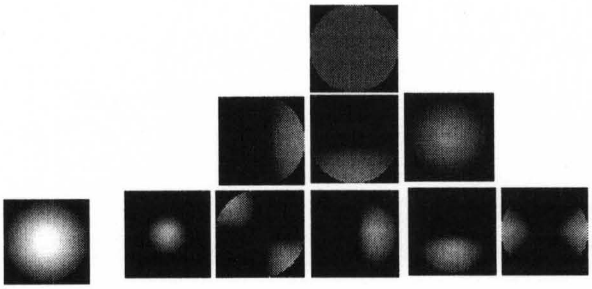
$$\varepsilon = \|I - \mathbf{B}\boldsymbol{\alpha}\|^2 \tag{41}$$

1. Brightness Constraint with Spherical Harmonics

For this work, assuming the albedo is known, the brightness constraint version for spherical harmonics illumination has two unknowns, namely: (a) the coefficients $\boldsymbol{\alpha}$ and (b) the exact surface normals $\mathbf{n} = (n_x, n_y, n_z)$. Typically, in face recognition applications using spherical harmonics [38], the surface normals of each subject is available. The author investigates the performance of ε given an input image I (synthesized from a known surface), but the surface normals used to solve the basis images are inaccurate, although close to the ground truth values.

The input image is synthesized from an ellipsoid with equation $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{h_t^2} = 1$, where a and b are fixed dimensions related to the image size, and h_t is the variable surface height. The surface is $Z(x, y, h_t) = h_t\sqrt{1 - (\frac{x^2}{a^2} + \frac{y^2}{b^2})}$. The surface normals, $\mathbf{n} = \frac{(-p, -q, 1)}{\sqrt{p^2+q^2+1}}$, where $p = \frac{\partial Z}{\partial x}$ and $q = \frac{\partial Z}{\partial y}$, can be easily determined with a closed-form solution. It is straightforward to solve for the basis images b_i (38), given the surface normals. Figure 40 shows the synthesized input image with $a = b = 32, h_t = 64$, and light source direction $l = (0, 0, 1)$, together with the nine basis images for this surface.

The author examines the values of ε in (41), using the synthesized input image I, as h_t is varied about the ground-truth value of $h_t = 64$. To solve for ε, one still needs to determine the suitable $\boldsymbol{\alpha}$, given I and $\mathbf{B}$. The equation, $I = \mathbf{B}\boldsymbol{\alpha}$, is an overdetermined linear system of equations, since the number of equations (pixel positions) is greater than the unknowns $(\boldsymbol{\alpha})$. The minimal solution is $\boldsymbol{\alpha} =$



(a) (b)

FIGURE 40– (a) Image synthesized from $Z(x, y, h_t) = h_t\sqrt{1 - (\frac{x^2}{a^2} + \frac{y^2}{b^2})}$ with $a = b = 32, h_t = 64$, and light source direction $l = (0, 0, 1)$. (b) Nine basis images of $Z(x, y)$ with $\lambda = 1$.

$\mathbf{VS}^{-1}\mathbf{U}^T\mathbf{I}$, where $\mathbf{B} = \mathbf{USV}^T$, the first n columns of $\mathbf{U}$ are used, and $\mathbf{S}$ is $(n \times n)$. For each h_t, a $\mathbf{B}$ matrix is computed, resulting to a new $\boldsymbol{\alpha}$, and eventually ε.

Figure 41 shows a plot of $\varepsilon(h_t)$ with respect to some values of h_t. Note that $\varepsilon(h_t)$ approaches the minimum as h_t approaches $h_t = 64$. It is interesting to know the values of $-\frac{d\varepsilon(h_t)}{dh_t}$, which is needed in gradient descent algorithms. Deriving $-\frac{d\varepsilon(h_t)}{dh_t}$,

$$\frac{d\varepsilon(h_t)}{dh} = 2\sum_x [I(\mathbf{x}) - \alpha_0 b_0(\mathbf{x}) - \ldots - \alpha_n b_n(\mathbf{x})][-\alpha_1 \frac{db_1(\mathbf{x})}{dh_t} - \ldots - \alpha_n \frac{db_n(\mathbf{x})}{dh_t}] \quad (42)$$

where $I(\mathbf{x})$ and $b_0(\mathbf{x})$ are not functions of h_t. The basis images $b_1(\mathbf{x}), \ldots, b_n(\mathbf{x})$ are functions of h_t since they are derived from the surface normals. Before $\frac{d\varepsilon(h)}{dh_t}$ can be solved, the value of $\boldsymbol{\alpha}$ needs to be determined using the method outlined above. The plot of $\frac{d\varepsilon(h_t)}{dh_t}$ is shown in Figure 41 for some values of h_t. Notice that its direction points toward the minimum. Using a simple gradient descent algorithm, the value of h_t such that $\varepsilon(h_t)$ is minimum can be found in a few iterations.

One can reach two conclusions: (1) The functions $\varepsilon(h_t)$ and $-\frac{d\varepsilon(h_t)}{dh_t}$ are unique because both $\mathbf{B}$ and $\boldsymbol{\alpha}$ change for each value of h_t. The matrix $\mathbf{B}$ is expected to change with h_t since most of its elements are functions of h_t. But $\boldsymbol{\alpha}$ is not directly a function of h_t; instead, it is determined by first solving the overdetermined linear system, $I = \mathbf{B}\boldsymbol{\alpha}$. Using this $\boldsymbol{\alpha}$, $\varepsilon(h_t)$ and $-\frac{d\varepsilon(h_t)}{dh_t}$ behave expectedly, i.e., $\varepsilon(h_t)$

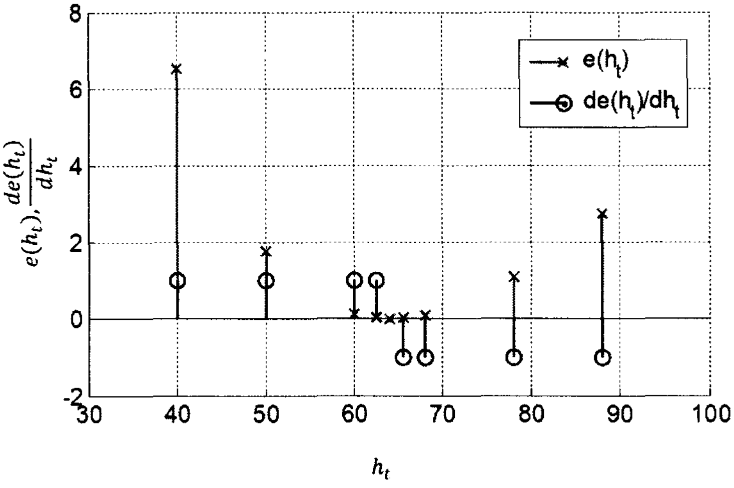


FIGURE 41 – Plot of $\varepsilon(h_t)$ and $\frac{d\varepsilon(h_t)}{dh_t}$ for some values of h_t. Note that $\varepsilon(h_t)$ approaches the minimum as $h_t = 64$ and the direction of $-\frac{d\varepsilon(h_t)}{dh_t}$ points toward the minimum. Using a simple gradient descent algorithm, the value of h_t such that $\varepsilon(h_t)$ is minimum can be found in a few iterations.

approaches the minimum as h_t gets closer to $h_t = 64$ and $-\frac{d\varepsilon(h_t)}{dh_t}$ points to the minimum h_t. (2) If there is prior knowledge (e.g., the surface Z as a function of model parameters like h_t) about the object to be reconstructed from the input image, minimizing the brightness constraint ε with respect to the model parameters, results to the best surface such that the image generated using this surface is as close to the input image. Therefore, shape recovery is performed in the process.

D. Model-based 3D Face Shape Recovery: An Iterative Approach (Frontal Pose)

The approach for the ellipsoid (parameterized by h_t) can be extended to 3D faces, which are parameterized by both shape ($\mathbf{b}_s$) and albedo ($\mathbf{b}_a$) coefficients from the shape and albedo models. In particular, minimizing the brightness constraint ε with respect to both shape ($\mathbf{b}_s$) and albedo ($\mathbf{b}_a$) coefficients, yields the best surface that generated the input image.

The 3D shape model (height map) is constructed by performing Principal Component Analysis (PCA) on aligned samples from the USF 3D Database [67].

The resulting shape model is

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s \mathbf{b}_s \tag{43}$$

where $\bar{\mathbf{s}}$ is the shape mean, $\mathbf{P}_s$ are the shape eigenvectors, and $\mathbf{b}_s$ is the set of shape coefficients. It is the matrix form of the shape model in (35), where $\mathbf{P}_s = [\mathbf{s}_1, \cdots, \mathbf{s}_{m-1}]$ and $\mathbf{b}_s = [\alpha_1, \cdots, \alpha_{m-1}]^T$.

Preprocessing and Postprocessing Steps: Note that only the z-component of the USF shape samples are involved in the shape model above, following the work of [1], i.e., the aligned samples that form the model are $Z_i = (z_1, z_2, \cdots, z_N)^T$, $i = 1, \cdots, m$, where m is the number of samples and N is the number of vertices. Since only the z-component is involved, the xy information is missing. There are inherent preprocessing and postprocessing steps that deal with this missing xy information.

Denote the set of control points in the mean shape and input image as $(x_{i,c}, y_{i,c})$ and $(x'_{i,c}, y'_{i,c})$, respectively. The preprocessing step involves solving for the vector-valued warping function, $\mathbf{f}(x_{i,c}, y_{i,c}) = (x'_{i,c}, y'_{i,c})$. Then the input image I is sampled, i.e., $I_s = I(\mathbf{f}(x_i, y_i))$, where (x_i, y_i) are pixels inside the mean shape mask (reference frame). All subsequent computations are then performed in the reference frame.

After all computations are done, the postprocessing step involves solving the inverse warping function, $\mathbf{f}'(x'_{i,c}, y'_{i,c}) = (x_{i,c}, y_{i,c})$. The pixel values (shape or texture) within the mean shape mask are then sampled, i.e., in the case of the shape output, $s_{out} = s_{ref}(\mathbf{f}'(x'_i, y'_i))$, where (x'_i, y'_i) are pixels inside the predefined output image. The output shape s_{out} can then be visualized as the recovered shape.

Each height map comes with a corresponding albedo. The albedo (texture) model can also be reconstructed using the same approach, resulting to

$$\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a \mathbf{b}_a \tag{44}$$

where $\bar{\mathbf{a}}$ is the mean albedo, $\mathbf{P}_a$ are the albedo eigenvectors, and $\mathbf{b}_a$ is the set of albedo coefficients. It is the matrix form of the albedo model in (35), where $\mathbf{P}_a = [\mathbf{t}_1, \cdots, \mathbf{t}_{m-1}]$ and $\mathbf{b}_a = [\beta_1, \cdots, \beta_{m-1}]^T$. There is a change of notation from $\mathbf{t}$ to $\mathbf{a}$.

Recall the brightness constraint in (41). The matrix of basis image, $\mathbf{B}$, can actually be expressed as a function of both shape $\mathbf{s}$ and albedo $\mathbf{a}$, i.e., $\mathbf{B} = \mathbf{f}(\mathbf{s}, \mathbf{a})$. This is clear in Figure 17 and equation (38), where the basis images are derived from the surface normals (calculated from shape) and albedo. From the shape (43) and albedo (44) PCA models, $\mathbf{B}$ can also be expressed as a function of shape ($\mathbf{b}_s$) and albedo coefficients ($\mathbf{b}_a$), i.e., $\mathbf{B} = \mathbf{f}(\mathbf{b}_s, \mathbf{b}_a)$. Therefore, the brightness constraint equation for human faces can be expressed as

$$\varepsilon(\mathbf{b}_s, \mathbf{b}_a) = \|I - \mathbf{f}(\mathbf{b}_s, \mathbf{b}_a)\boldsymbol{\alpha}\|^2 \tag{45}$$

and minimizing this with respect to $\mathbf{b}_s$ and $\mathbf{b}_a$ yields the best shape ($\tilde{\mathbf{s}}$) and albedo ($\tilde{\mathbf{a}}$) that generated the input image I, i.e., $(\tilde{\mathbf{b}}_s, \tilde{\mathbf{b}}_a) = \min_{(\mathbf{b}_s, \mathbf{b}_a)} \varepsilon(\mathbf{b}_s, \mathbf{b}_a)$, $\tilde{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{P}_s\tilde{\mathbf{b}}_s$, $\tilde{\mathbf{a}} = \bar{\mathbf{a}} + \mathbf{P}_a\tilde{\mathbf{b}}_a$.

Given a 3D face surface $Z(x, y)$, the normals can be approximated using the surface slopes in the x and y directions. Finite difference approximations for $p = \frac{\partial Z}{\partial x}$ and $q = \frac{\partial Z}{\partial y}$ are

$$\begin{aligned} p &= Z(x+1, y) - Z(x, y) \\ q &= Z(x, y+1) - Z(x, y) \end{aligned} \tag{46}$$

The basis images $b_i(\mathbf{x})$ can then be computed from p and q using the equations in (46), assuming the albedo is known. Given an input image, and suppose the real albedo is available, we can find the shape parameters $\mathbf{b}_s$ such that the brightness constraint $\varepsilon(\mathbf{b}_s)$ 39 is minimum by performing gradient descent with

$$\frac{d\varepsilon}{db_{si}} = 2\sum_x [I(\mathbf{x}) - \alpha_0 b_0(\mathbf{x}) - \ldots - \alpha_n b_n(\mathbf{x})][-\alpha_1 \frac{db_1(\mathbf{x})}{db_{si}} - \ldots - \alpha_n \frac{db_n(\mathbf{x})}{db_{si}}] \tag{47}$$

To solve $\frac{db_1(\mathbf{x})}{db_{si}}$, recall that $b_1 = \sqrt{\frac{3}{4\pi}\lambda \circ n_x}$, $n_x = -\frac{p}{\sqrt{p^2+q^2+1}}$ and $p = Z(x +$

$1, y) - Z(x, y)$. $Z(x, y)$ and $Z(x+1, y)$ correspond to certain positions (e.g., *j*th element in the vector **s**) in the shape model ($\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s\mathbf{b}_s$) vector. Then, $\frac{ds_j}{db_{si}} = P_{si,j}$. Taking all these into account, $\frac{db_1(\mathbf{x})}{db_{si}}$ can be determined. The other $\frac{db_i(\mathbf{x})}{db_{si}}$ terms can be solved using the same approach.

Alternatively, suppose the real shape is known, the albedo parameters that correspond to the minimum $\varepsilon(\mathbf{b}_a)$ can be determined by performing the following gradient descent equations

$$\frac{d\varepsilon}{db_{ai}} = 2\sum_{x}[I(\mathbf{x}) - \alpha_0 b_0(\mathbf{x}) - \ldots - \alpha_n b_n(\mathbf{x})][-\alpha_1\frac{db_1(\mathbf{x})}{db_{ai}} - \ldots - \alpha_n\frac{db_n(\mathbf{x})}{db_{ai}}] \qquad (48)$$

Using a similar approach as above, but this time with the albedo model, $\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a\mathbf{b}_a$, it is straightforward to get $\frac{db_i(\mathbf{x})}{db_{ai}}$. However, the brightness constraint ε 39 is a function of both parameters, $\mathbf{b}_s$ and $\mathbf{b}_a$, for real faces, e.g., $\varepsilon(\mathbf{b}_s, \mathbf{b}_a)$. To solve this problem, the author follows the *project-out* approach in [60] by dividing the optimization into two steps: (a) minimizing $\varepsilon(\mathbf{b}_s, \mathbf{0})$ with respect to the shape parameters $\mathbf{b}_s$ and (b) using this result as the optimal $\mathbf{b}_s$ to minimize $\varepsilon(\mathbf{b}_s, \mathbf{b}_a)$ with respect to the albedo parameters $\mathbf{b}_a$. Steps (a) and (b) corresponds to (47) and (48), respectively. Algorithm 3 best describes this model-based iterative approach. The minimization process in steps 2 and 3 of Algorithm 3 can be done using the built-in *fminunc* function in Matlab, which performs unconstrained optimization.

E. Model-based 3D Face Recovery: A Coupled Statistical Model Approach (Frontal Pose)

Castelan et al. [1] [14] developed a coupled statistical model based on the AAM [36] concept, which can recover the 3D shape from intensity images with frontal light source. The 2D shape model in [36] is replaced with a 3D shape model composed of height maps. The goal of this work is to formulate a 3D shape recovery method, by modifying the framework of [14] to handle images of general lighting.

Algorithm 3 Iterative Approach to Model-based 3D Face Shape Recovery

```
INPUT: (a) Input image, $I$ (b) Shape model, $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s\mathbf{b}_s$ (c) Albedo model, $\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a\mathbf{b}_a$
OUTPUT: (a) Recovered shape, $\tilde{\mathbf{s}}$ (b) Recovered albedo, $\tilde{\mathbf{a}}$
1: Set initial albedo coefficients to zero: Set $\tilde{\mathbf{b}}_a = \mathbf{0}$
2: Minimize brightness constraint with respect to shape parameters: Solve for $\tilde{\mathbf{b}}_s = \min_{\mathbf{b}_s} \varepsilon(\mathbf{b}_s, \tilde{\mathbf{b}}_a) = \min_{\mathbf{b}_s} \|I - \mathbf{f}(\mathbf{b}_s, \tilde{\mathbf{b}}_a)\boldsymbol{\alpha}\|^2$
3: Use the solved shape coefficient and minimize brightness constraint with respect to albedo parameters: Solve for $\tilde{\mathbf{b}}_a = \min_{\mathbf{b}_a} \varepsilon(\tilde{\mathbf{b}}_s, \mathbf{b}_a) = \min_{\mathbf{b}_a} \|I - \mathbf{f}(\tilde{\mathbf{b}}_s, \mathbf{b}_a)\boldsymbol{\alpha}\|^2$
4: Repeat steps 2-3 until termination criterion is satisfied.
5: Solve for the recovered shape and albedo: Solve for $\tilde{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{P}_s\tilde{\mathbf{b}}_s$ and $\tilde{\mathbf{a}} = \bar{\mathbf{a}} + \mathbf{P}_a\tilde{\mathbf{b}}_a$
```

Consider Figure 42, which is basically a modified version of Figure 20 that follows the discussion in Section II.B.4. A coupled statistical model can be constructed from this figure, which links coefficients of the intensity, shape and spherical harmonics projection (SHP) images of faces of various subjects. By fitting the SHP model to the input image, the coupled model can be used to recover the corresponding shape and albedo parameters of the input face. The next sections will discuss two ways how this fitting and recovery process can be performed.

1. Coupled Statistical Models

The coupled statistical model in Figure 42 is composed of three distinct but related models. The first two were previously mentioned in Section III.D, namely, the 3D shape (height map) and texture (albedo) models constructed using Principal Component Analysis (PCA) on aligned samples of the USF Database [67].

The third model corresponds to the spherical harmonics projection (SHP) images, $h_1, \cdots, h_n$, in Fig.42. One can build an SHP model

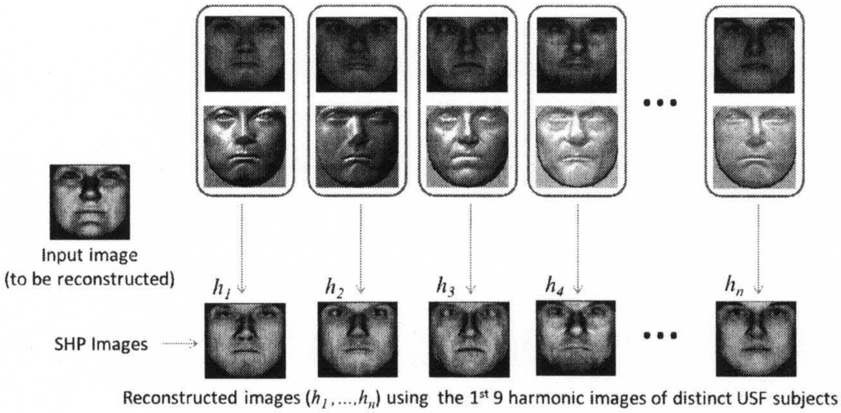

Input image
(to be reconstructed)
SHP Images
h_1
h_2
h_3
h_4
h_n
...
Reconstructed images ($h_1,\dots,h_n$) using the 1st 9 harmonic images of distinct USF subjects

FIGURE 42 – Coupled statistical model illustration. A coupled statistical model can be constructed from this figure, which links coefficients of the intensity, shape and spherical harmonics projection (SHP) images of faces of various subjects.

$$\mathbf{h} = \bar{\mathbf{h}} + \mathbf{P}_h \mathbf{b}_h \tag{49}$$

similar to the shape and texture models in Sec. III.D, where $\bar{\mathbf{h}}$ is the SHP mean, $\mathbf{P}_h$ are the SHP eigenvectors, and $\mathbf{b}_h$ is the set of SHP coefficients.

2. 3D Face/Albedo Recovery

a. ***Method I*** The face 3D shape (height map) and appearance of an input image can be described by the coefficients, $\mathbf{b}_s$, $\mathbf{b}_a$, and $\mathbf{b}_h$, from the shape ($\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s \mathbf{b}_s$), albedo ($\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a \mathbf{b}_a$) and SHP ($\mathbf{h} = \bar{\mathbf{h}} + \mathbf{P}_h \mathbf{b}_h$) models, respectively. This section will describe how to combine the vector coefficients into a single model that can be used to recover the height map and albedo of the input image under arbitrary illumination. For the kth training sample, a combined vector can be formed as follows

$$\mathbf{b}_c = \begin{pmatrix} \mathbf{W}_h \mathbf{b}_h \\ \mathbf{W}_a \mathbf{b}_a \\ \mathbf{b}_s \end{pmatrix} = \begin{pmatrix} \mathbf{W}_h \mathbf{P}_h^T (\mathbf{h} - \bar{\mathbf{h}}) \\ \mathbf{W}_a \mathbf{P}_a^T (\mathbf{a} - \bar{\mathbf{a}}) \\ \mathbf{P}_s^T (\mathbf{s} - \bar{\mathbf{s}}) \end{pmatrix} \tag{50}$$

where $\mathbf{W}_h$ and $\mathbf{W}_a$ are diagonal matrices of weights for the SHP and albedo models. The weights compensate for the difference in units between the shape, intensity and SHP vectors. The concatenated vectors, $\mathbf{b}_c$, for all training samples are combined into a matrix, and PCA is performed, resulting to a single model that links the variations between three models

$$\mathbf{b} = \mathbf{F}\mathbf{c} = \begin{pmatrix} \mathbf{F}_h \\ \mathbf{F}_a \\ \mathbf{F}_s \end{pmatrix} \mathbf{c} \tag{51}$$

where $\mathbf{F}$ are the eigenvectors and $\mathbf{c}$ is a vector of model parameters controlling the SHP, albedo, and shape (height map) of the model, simultaneously. The matrices, $\mathbf{F}_h$, $\mathbf{F}_a$, and $\mathbf{F}_s$, represent the eigenvectors corresponding to the SHP, albedo, and shape (height map) subspaces, respectively. The three models can be expressed independently as functions of $\mathbf{c}$,

$$\mathbf{h} = \bar{\mathbf{h}} + \mathbf{P}_h \mathbf{W}_h^{-1} \mathbf{F}_h \mathbf{c} \tag{52}$$

$$\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a \mathbf{W}_a^{-1} \mathbf{F}_a \mathbf{c} \tag{53}$$

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s \mathbf{F}_s \mathbf{c} \tag{54}$$

Given a 2D input image I_{inp}, the 3D shape and albedo can be recovered by first solving the SHP model coefficients $\mathbf{b}_h$, using the equation

$$\mathbf{b}_h = \mathbf{P}_h^T (I_{inp} - \bar{\mathbf{h}}) \tag{55}$$

The term $\mathbf{b}_h$ corresponds to the term $(\mathbf{W}_h^{-1}\mathbf{F}_h\mathbf{c})$ in (52) of the combined model. The parameter $\mathbf{c}$ can be estimated by performing the optimization

$$\mathbf{c} = arg \min_{\mathbf{c}} (\mathbf{b}_h - \mathbf{W}_h^{-1}\mathbf{F}_h\mathbf{c})^T (\mathbf{b}_h - \mathbf{W}_h^{-1}\mathbf{F}_h\mathbf{c}) \tag{56}$$

The 3D shape can be recovered from (52), using the solved parameter $\mathbf{c}$. Similarly, the albedo can be solved using (53). This method is summarized in the algorithm listing below.

Algorithm 4 Coupled Model Approach to Model-based 3D Face Shape Recovery

```
**INPUT**: (a) Input image, $I_{inp}$ (b) Shape and albedo samples: $(\mathbf{s}_1, \mathbf{a}_1)$ to $(\mathbf{s}_n, \mathbf{a}_n)$
**OUTPUT**: (a) Recovered shape, $\tilde{\mathbf{s}}$ (b) Recovered albedo, $\tilde{\mathbf{a}}$
**Build the shape and albedo models from the samples using PCA**: Construct $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s\mathbf{b}_s$ and $\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a\mathbf{b}_a$.
**Construct the basis images for each pair**: $\mathbf{B}_i = [b_1, \cdots, b_9]$ for each pair $(\mathbf{s}_i, \mathbf{a}_i)$
**Build the SHP model**: Given an input $I_{inp}$, solve for the SHP images, $(h_1, h_2, \cdots, h_n)$, for all samples using Algorithm 1 and then construct, $\mathbf{h} = \bar{\mathbf{h}} + \mathbf{P}_h\mathbf{b}_h$
**Replace the shape samples with its coefficients**: Solve for $\mathbf{b}_{si} = P_s^T(\mathbf{s}_i - \bar{\mathbf{s}})$
**Replace the texture samples with its coefficients**: Solve for $\mathbf{b}_{ai} = P_a^T(\mathbf{a}_i - \bar{\mathbf{a}})$
**Replace the SHP images with its coefficients**: Solve for $\mathbf{b}_{hi} = P_h^T(\mathbf{h}_i - \bar{\mathbf{h}})$
**Form the combined vector for the each training sample**: Construct $\mathbf{b}_{ck} = (\mathbf{W}_h\mathbf{b}_h, \mathbf{W}_a\mathbf{b}_a, \mathbf{b}_s)^T$
**Perform PCA on the combined vectors according to (51).**
**Solve for the SHP coefficients of the input image**: Get $\mathbf{b}_{h,inp} = P_h^T(I_{inp} - \bar{\mathbf{h}})$.
**Estimate the parameter c using (56)**: $\tilde{\mathbf{c}} = arg\min_{\mathbf{c}}(\mathbf{b}_h - \mathbf{W}_h^{-1}\mathbf{F}_h\mathbf{c})^T(\mathbf{b}_h - \mathbf{W}_h^{-1}\mathbf{F}_h\mathbf{c})$
**Solve for the recovered shape and albedo using (54) and (53)**: $\tilde{\mathbf{a}} = \bar{\mathbf{a}} + \mathbf{P}_a\mathbf{W}_a^{-1}\mathbf{F}_a\tilde{\mathbf{c}}$ and $\tilde{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{P}_s\mathbf{F}_s\tilde{\mathbf{c}}$
```

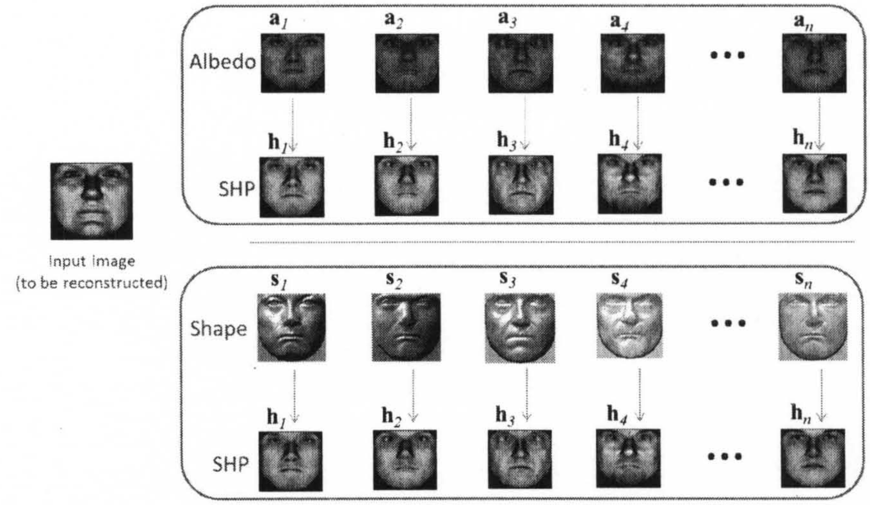


FIGURE 43 – The coupled model illustration in Fig. 42 can be decomposed into two parts, which can then be cast as a regression framework. There are two regression models at work here, namely: (a) SHP-to-shape and (b) SHP-to-texture.

b. ***Method II*** The coupled model illustration in Figure 42 can actually be decomposed into two diagrams, as shown in Figure 43. From Figure 43, the iterative framework of Section III.D can be cast as a regression framework. There are two regression models at work here, namely: (a) SHP-to-shape and (b) SHP-to-texture. The SHP coefficients in both models are considered the independent data (**X**). The shape and texture coefficients are the dependent data (**Y**). Note that the SHP model is computed (and can be done in a short amount of time) each time a new input image I_{inp} comes in.

Multivariate multiple linear regression (MLR) cannot be applied directly to fit a model between the original data matrices **X** and **Y** due to their high-dimensional nature. Specifically, let **X** and **Y** be of size $(n \times p)$, where p is the number of pixels, n is the number of samples, and $n \ll p$. For MLR to be successful [68], the number of samples must be greater than the number of variables, i.e., $n > p$.

In this paper, instead of using MLR, a related method called Principal Component Regression (PCR) is utilized to get the relationship between **X** and **Y**. The basic idea of PCR is to decompose **X** and **Y** into a low-dimensional subspace, i.e., replacing the high-dimensional vectors $\mathbf{s}_i$, $\mathbf{a}_i$, and $\mathbf{h}_i$ by their respective PCA coefficients ($\mathbf{b}_{s_i}$, $\mathbf{b}_{a_i}$, and $\mathbf{b}_{h_i}$). Then, standard MLR is performed between the low-dimensional representations of **X** and **Y**. Figure 44 explains the difference between MLR and PCR.

For the SHP-to-shape regression model, let $\mathbf{T} = [\mathbf{b}_{h_1}, \cdots, \mathbf{b}_{h_n}]$ and $\mathbf{U} = [\mathbf{b}_{s_1}, \cdots, \mathbf{b}_{s_n}]$ be the low-dimensional representations of **X** and **Y**. No preprocessing steps such as centering are needed for the PCA coefficients (see (51)). Performing MLR leads to

$$\mathbf{U} = \mathbf{T}\mathbf{C} + \mathbf{F} \tag{57}$$

where **C** is the matrix of regression coefficients and **F** is the matrix of random noise errors. The least squares method then gives

$$\tilde{\mathbf{C}} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U} \tag{58}$$

Given a 2D input image I_{inp}, the SHP model coefficients $\tilde{\mathbf{b}}_h$ can be solved using (55). The shape coefficient $\tilde{\mathbf{b}}_s$ can be predicted with

$$\tilde{\mathbf{b}}_s = (\tilde{\mathbf{b}}_h^T \tilde{\mathbf{C}})^T \tag{59}$$

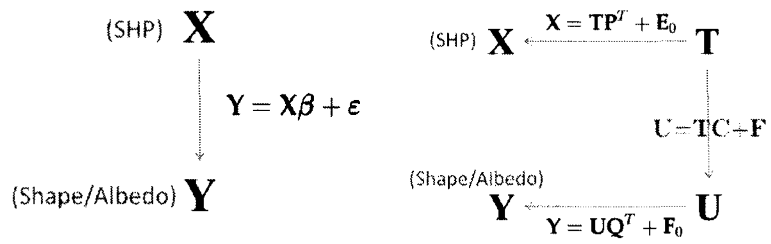
The recovered shape can be determined by substituting the solved shape coefficient, $\tilde{\mathbf{b}}_s$, into the shape model in (52),i.e., $\tilde{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{P}_s\tilde{\mathbf{b}}_s$. The recovered albedo can be solved by following the same steps as above, replacing $\tilde{\mathbf{b}}_s$ with $\tilde{\mathbf{b}}_a$ in **U**. This method is summarized in the algorithm listing below.

F. Comparison of Model-based SFS Methods

This section compares the three previously discussed model-based 3D face recovery methods, starting from a mathematical and algorithmic standpoint to side-by-side comparisons of experimental results. The goal of these approaches

Algorithm 5 Principal Component Regression (PCR) Framework for 3D Shape Recovery

```
INPUT: (a) Input image, $I_{inp}$ (b) Shape and albedo samples: $(\mathbf{s}_1, \mathbf{a}_1)$ to $(\mathbf{s}_n, \mathbf{a}_n)$
OUTPUT: (a) Recovered shape, $\tilde{\mathbf{s}}$ (b) Recovered albedo, $\tilde{\mathbf{a}}$
Build the shape and albedo models from the samples using PCA: Construct $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s\mathbf{b}_s$ and $\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a\mathbf{b}_a$.
Construct the basis images for each pair: $\mathbf{B}_i = [b_1, \cdots, b_9]$ for each pair $(\mathbf{s}_i, \mathbf{a}_i)$
Build the SHP model: Given an input $I_{inp}$, solve for the SHP images, $(h_1, h_2, \cdots, h_n)$, for all samples using Algorithm 1 and then construct, $\mathbf{h} = \bar{\mathbf{h}} + \mathbf{P}_h\mathbf{b}_h$
Replace the shape samples with its coefficients: Solve for $\mathbf{b}_{si} = P_s^T(\mathbf{s}_i - \bar{\mathbf{s}})$
Replace the texture samples with its coefficients: Solve for $\mathbf{b}_{ai} = P_a^T(\mathbf{a}_i - \bar{\mathbf{a}})$
Replace the SHP images with its coefficients: Solve for $\mathbf{b}_{hi} = P_h^T(\mathbf{h}_i - \bar{\mathbf{h}})$
Setup matrices for PCR: $\mathbf{T} = [\mathbf{b}_{h_1}^T, \cdots, \mathbf{b}_{h_n}^T]$, $\mathbf{U}_{sh} = [\mathbf{b}_{s_1}^T, \cdots, \mathbf{b}_{s_n}^T]$ and $\mathbf{U}_{al} = [\mathbf{b}_{a_1}^T, \cdots, \mathbf{b}_{a_n}^T]$
Build two PCR models: Construct $\tilde{\mathbf{C}}_{sh} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U}_{sh}$ and $\tilde{\mathbf{C}}_{al} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U}_{al}$
Solve for the SHP coefficients of the input image: Get $\mathbf{b}_{h,inp} = P_h^T(I_{inp} - \bar{\mathbf{h}})$
Solve for the shape and texture coefficients: Get $\tilde{\mathbf{b}}_s^T = \tilde{\mathbf{b}}_h^T\tilde{\mathbf{C}}_{sh}$ and $\tilde{\mathbf{b}}_a^T = \tilde{\mathbf{b}}_h^T\tilde{\mathbf{C}}_{al}$
Solve for the recovered shape and albedo: $\tilde{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{P}_s\tilde{\mathbf{b}}_s$ and $\tilde{\mathbf{a}} = \bar{\mathbf{a}} + \mathbf{P}_a\tilde{\mathbf{b}}_a$
```



(a) (b)

FIGURE 44 – (a) Multivariate multiple linear regression: A model ($\mathbf{Y} = \mathbf{X}\beta + \varepsilon$) is directly fitted between the independent data (**X**) and dependent data (**Y**). (b) Principal component regression: Instead of directly fitting between **X** and **Y**, they are first transformed to a low-dimensional subspace, forming **T** and **U**, e.g. $\mathbf{X} = \mathbf{TP}^T + \mathbf{E}_0, \mathbf{Y} = \mathbf{UQ}^T + \mathbf{F}_0$, where **P** and **Q** are eigenvectors. Actual multiple linear regression ($\mathbf{U} = \mathbf{TC} + \mathbf{F}$) is done between **T** and **U**.

is to recover 3D shape (and albedo as byproduct) from single images of general and unknown illumination.

1. Algorithmic Comparison

Each method has its own way of minimizing the brightness constraint equation, with respect to shape ($\mathbf{b}_s$) and albedo ($\mathbf{b}_a$) coefficients, i.e.,

$$(\tilde{\mathbf{b}}_s, \tilde{\mathbf{b}}_a) = \min_{\mathbf{b}_s, \mathbf{b}_a} \varepsilon(\mathbf{b}_s, \mathbf{b}_a) = \min_{\mathbf{b}_s, \mathbf{b}_a} \|I - \mathbf{f}(\mathbf{b}_s, \mathbf{b}_a)\boldsymbol{\alpha}\|^2 \qquad (60)$$

The iterative approach uses gradient descent to minimize (60), getting first the gradients $\frac{\partial \varepsilon}{\partial \mathbf{b}_s}$ and $\frac{\partial \varepsilon}{\partial \mathbf{b}_a}$, and tries to find the solution with the help of the gradient directions. Due to its iterative nature, this method is expected to take a lot of computational effort and time to arrive at the solution.

The coupled statical model approaches (**Methods I and II**) differ from the iterative approach by incorporating *regression-like* algorithms in the minimization process. Moreover, there is an intermediary step before the actual minimization with respect to $\mathbf{b}_s$ and $\mathbf{b}_a$ is performed. This intermediary step is done by introducing the concept of an SHP image and minimizing first with respect to the SHP coefficients, i.e.,

$$\tilde{\mathbf{b}}_h = \min_{\mathbf{b}_h} \|I - (\bar{\mathbf{h}} + \mathbf{P}_h \mathbf{b}_h)\|^2 \tag{61}$$

The solution is closed-form, as expressed in (55). The reconstructed SHP image closest to the input image I in (61) is $\tilde{\mathbf{h}} = \bar{\mathbf{h}} + \mathbf{P}_h \tilde{\mathbf{b}}_h$.

A SHP image is directly related to a certain shape (**s**) and albedo (**a**), as illustrated in Figure 42 and Algorithm **1**. From Figure 42, it is possible to form n data pairs, $(\mathbf{h}_i, (\mathbf{s}_i, \mathbf{a}_i))$, $i = 1, \cdots, n$, where $\mathbf{h}_i$ is the independent variable and $(\mathbf{s}_i, \mathbf{a}_i)$ is the dependent variable. Both coupled model approaches replace the high-dimensional vectors, $\mathbf{h}_i$, $\mathbf{s}_i$, and $\mathbf{a}_i$, with their respective PCA coefficients, $\mathbf{b}_{hi}$, $\mathbf{b}_{si}$, and $\mathbf{b}_{ai}$. The two methods differ, however, in the way they predict the corresponding $(\tilde{\mathbf{b}}_{si}, \tilde{\mathbf{b}}_{ai})$ for the solved $\tilde{\mathbf{b}}_{hi}$ in (55).

The first method combines all three PCA coefficients (SHP, shape and albedo) into a single vector (50) and performs PCA on them (51). The result after PCA is a single model that links the three models in terms of a single vector **c**. Recovering **c** independently from the SHP model (55) can lead to closed-form solutions for the recovered shape (54) and albedo (53).

The second method uses multiple linear regression (MLR) concepts (specifically, principal component regression (PCR)) to arrive at the corresponding $(\tilde{\mathbf{b}}_{si}, \tilde{\mathbf{b}}_{ai})$, given a $\mathbf{h}_i$. Starting from the n data pairs, $(\mathbf{h}_i, (\mathbf{s}_i, \mathbf{a}_i))$, two n data pairs, $(\mathbf{h}_i, \mathbf{s}_i)$ and $(\mathbf{h}_i, \mathbf{a}_i)$, $i = 1, \cdots, n$, can be formed. PCR requires the original data pairs to be replaced with their respective PCA coefficients, $(\tilde{\mathbf{b}}_{hi}, \tilde{\mathbf{b}}_{si})$ and $(\tilde{\mathbf{b}}_{hi}, \tilde{\mathbf{b}}_{ai})$. Two regression models can be trained from these data pairs using (58). It is straightforward to solve the shape and albedo coefficients by employing the common *prediction equation* for regression problems (59).

In contrast to the iterative nature of the first algorithm, the coupled statistical model approaches use mostly a sequence of matrix operations (except for the quasi-Newton optimization step (56) for **Method I**), making them computationally efficient. The table below compares the execution time of the three model-based SFS methods, under the same machine. Since **Method II** consists of purely matrix operations, it is the fastest among the three methods.

TABLE 2
AVERAGE COMPUTATIONAL TIMES OF MODEL-BASED 3D FACE RECOVERY APPROACHES

	Iterative	Method I	Method II
Time (seconds)	123.83	0.73	0.45

2. Experimental Results Comparison

This section shows experiments to evaluate the performance of the proposed methods (Iterative, **Method I**, **Method II**) in recovering the 3D face shape. The face models are built using the USF 3D Face Database [67], which contains 100 subjects of diverse gender and ethnicity. Out of these 100 samples, 80 subjects were deemed to be acceptable and were subsequently chosen to build the shape and texture models.

To quantify the reconstruction accuracy, we recover the 3D shape for 80 out-of-training USF samples illuminated with combined light source directions of $(0, 0, 1)$ and $(0, 0.5, 0.9)$. For each, we compute the following measures: (a) Height Error - the recovered height map is compared with the ground truth height and the mean absolute error is computed as

$$\bar{s}_{err} = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{|s_i - s_{GT,i}|}{s_{GT,i}} \tag{62}$$

where N_p is the number of pixels, s_i and $s_{GT,i}$ are height values at the ith pixel position for the recovered shape and the ground-truth shape, respectively, and (b) Surface Orientation Error - the directions of the recovered normal vectors are compared with the ground truth data. The average of the difference angle is computed as

$$\bar{\theta}_{err} = \frac{1}{N_p} \sum_{i=1}^{N_p} cos^{-1}\left(\frac{\mathbf{n}_i \cdot \mathbf{n}_{GT,i}}{\|\mathbf{n}_i\| \|\mathbf{n}_{GT,i}\|}\right) \tag{63}$$

where N_p is the number of pixels, $\mathbf{n}_i$ and $\mathbf{n}_{GT,i}$ are normal vectors at the ith pixel position for the recovered shape and the ground-truth shape, respectively.

The comparison of experimental results starts with a side-by-side visualization of the mean height (Figure 45) and surface orientation (Figure 46) error stem plots. For the mean height error, most samples fall under the $(2 - 8\%)$ range for the three methods. It is worth noting that coupled statistical approaches (**Method I** and **II**) have similar height error values. Most samples are within the $(0.04 - 0.14\ rad)$ range for the mean surface orientation error stem plots and **Methods I** and **II** have similar surface orientation error values, as well.

The actual mean and standard deviation of the mean height error and mean surface orientation error for the 80 out-of-training USF samples are shown in Tables 3 and 4. Notice the similarity of the numerical values, which means that the output of the three proposed methods are similar, as well.

TABLE 3
MEAN AND STANDARD DEVIATION OF THE MEAN HEIGHT ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Iterative	Method I	Method II
$\mu_{\bar{s}_{err}}$ (%)	2.69	2.28	2.28
$\sigma_{\bar{s}_{err}}$ (%)	0.82	0.72	0.73

The next step is to place alongside each other the recovered shapes and albedo for the three model-based algorithms. Figure 47 shows the recovered shapes together with the input and ground-truth shape. The results are very close visually. Likewise, Figure 48 displays the recovered albedo. Notice that the results are very difficult to differentiate from each other.

TABLE 4
MEAN AND STANDARD DEVIATION OF THE MEAN SURFACE ORIENTATION ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Iterative	Method I	Method II
$\mu_{\bar{\theta}_{err}}$ (rad)	0.05	0.04	0.04
$\sigma_{\bar{\theta}_{err}}$ (rad)	0.01	0.01	0.01

3. Related Algorithms Comparison

The three methods proposed in this chapter will be compared to their two predecessors, namely: (a) Castelan et al. [1] [14] and (b) Ahmed et al. [2]. Since the three proposed methods have similar results, Method II (PCR) is chosen to represent the proposed methods and will be compared alongside the earlier approaches.

The first difference between the proposed methods and their predecessors is in the determination of correspondence between the USF database samples and the input image. Instead of 76 control points, only three control points are used to determine the map between the database samples and the input, through the mean shape, as shown in Figure 49. In addition, the warping function used in the predecessors' algorithms is rigid (Euclidean) compared to the nonrigid (thin-plate splines) approach for the proposed methods.

Castelan's methods [1] [14] have an additional major difference in that they make use only of shape and texture models. There is no spherical harmonics projection (SHP) model that can deal with illumination in the input image. Ahmed et al. [2] has all three models (shape, texture, and SHP) and uses Method I to recover 3D facial shapes.

Similar to the previous section, the comparison starts with a side-by-side visualization of the mean height (Figure 50) and surface orientation (Figure 51) error stem plots. The actual mean and standard deviation of the mean height error and mean surface orientation error of the 80 out-of-training USF samples, for this related algorithms comparison, are shown in Tables 5 and 6. It is clear that the proposed algorithm (PCR-**Method II**) is superior compared to the previous state-of-the-art.

The next step is to place alongside each other the recovered shapes and albedo for the proposed method (Method II-PCR), Castelan's (Method III) and Ahmed's (Method IV). Figure 52 shows the recovered shapes together with the input and ground-truth shape. Likewise, Figure 53 displays the recovered albedo. The results between Methods II and IV are close visually. However, it is obvious that Castelan's version (Method III) suffers with input images that have illumination.

TABLE 5
RELATED ALGORITHMS COMPARISON: MEAN AND STANDARD DEVIATION OF THE MEAN HEIGHT ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	PCR	Castelan	Ahmed
$\mu_{\bar{s}_{err}}$ (%)	2.28	8.32	3.40
$\sigma_{\bar{s}_{err}}$ (%)	1.00	3.40	1.11

TABLE 6
RELATED ALGORITHMS COMPARISON: MEAN AND STANDARD DEVIATION OF THE MEAN SURFACE ORIENTATION ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	PCR	Castelan	Ahmed
$\mu_{\bar{\theta}_{err}}$ (rad)	0.04	0.11	0.06
$\sigma_{\bar{\theta}_{err}}$ (rad)	0.01	0.02	0.02

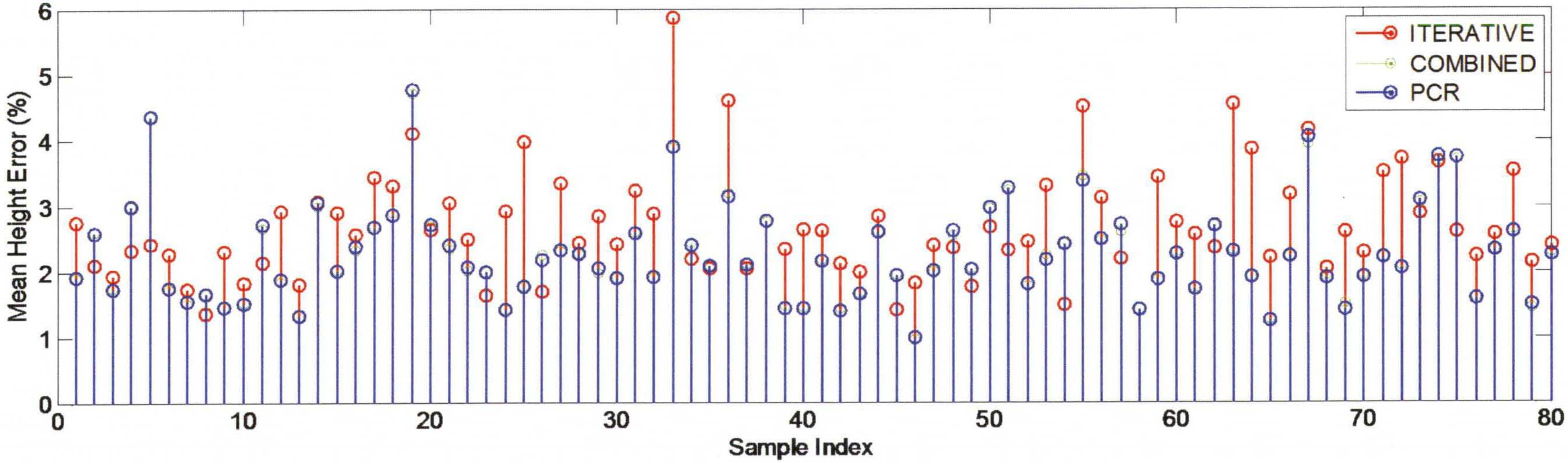


FIGURE 45– Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Most samples fall under the $(2 - 8\%)$ mean height error range, for all three methods. The coupled statistical approaches, **Method I** (Combined) and **II** (PCR), have similar error values. Average error for all 80 out-of-training samples is 3.71%, 3.33%, and 3.33% for the Iterative, Method I, and Method II approaches, respectively.

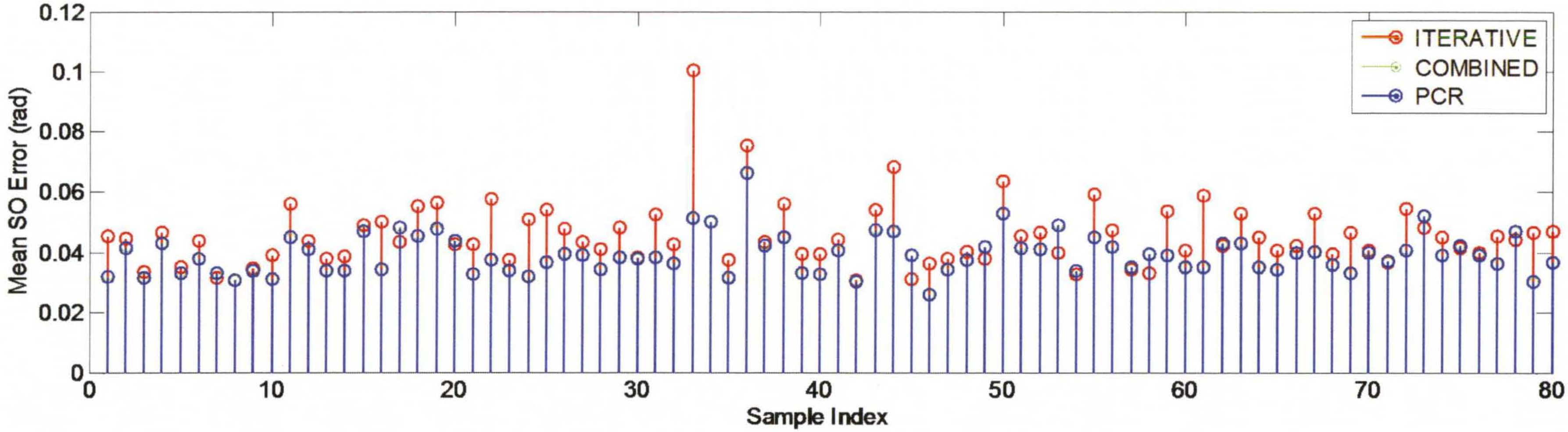


FIGURE 46 – Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Most samples fall under the $(0.04 - 0.14\,rad)$ surface orientation error range, for all three methods. The coupled statistical approaches, **Method I** (Combined) and **II** (PCR), have similar error values. Average error for all 80 out-of-training samples is 0.06 rad, 0.06 rad, and 0.06 rad for the Iterative, Method I, and Method II approaches, respectively.

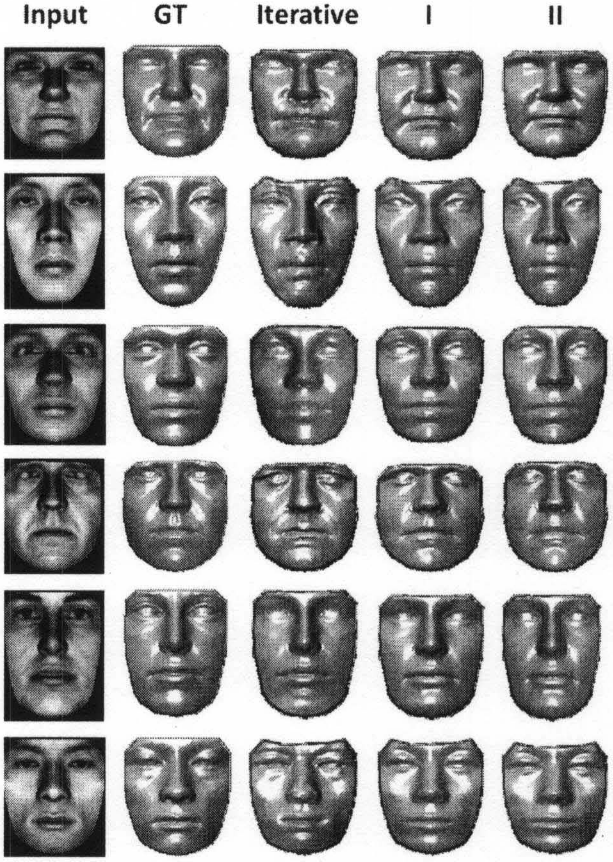


FIGURE 47 – Recovered shapes, together with the input image and ground-truth (GT) shape, for the three model-based methods (Iterative, Method I, and Method II). The results are very close, visually.

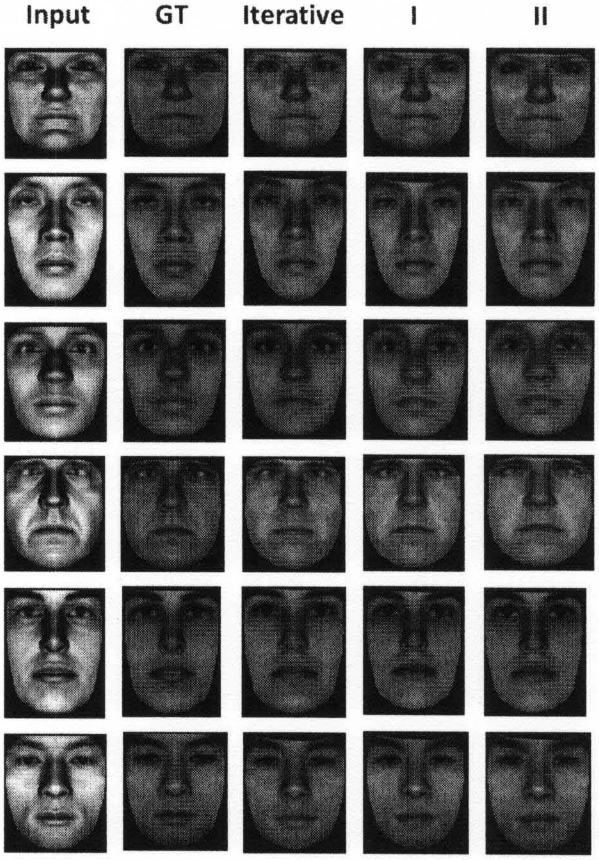


FIGURE 48 – Recovered albedo, together with the input image and ground-truth (GT) albedo, for the three model-based methods (Iterative, Method I, and Method II). The results are very close, visually.

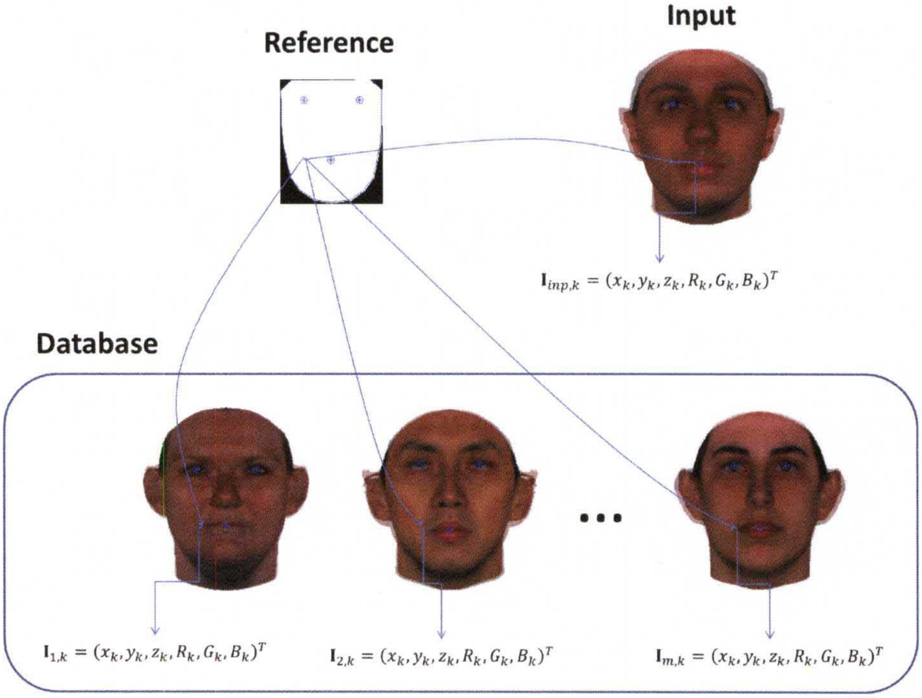

Reference
Input
$\mathbf{I}_{inp,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$
Database
...
$\mathbf{I}_{1,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$
$\mathbf{I}_{2,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$
$\mathbf{I}_{m,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$

FIGURE 49 – (**Upper Left**) Mean shape together with its convex hull. The convex hull forms the boundary of the mask. (**Lower**) Each pixel position within the mask corresponds to a certain point on a sample scan in the USF database. The vector $\mathbf{I}_{i,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$ refers to the shape and texture information at the kth vertex of the ith sample scan. (**Upper Right**) Similarly, each pixel position within the mask corresponds to a certain point on the input image. Hence, there is correspondence between the USF database samples and the input image, through the mean shape mask.

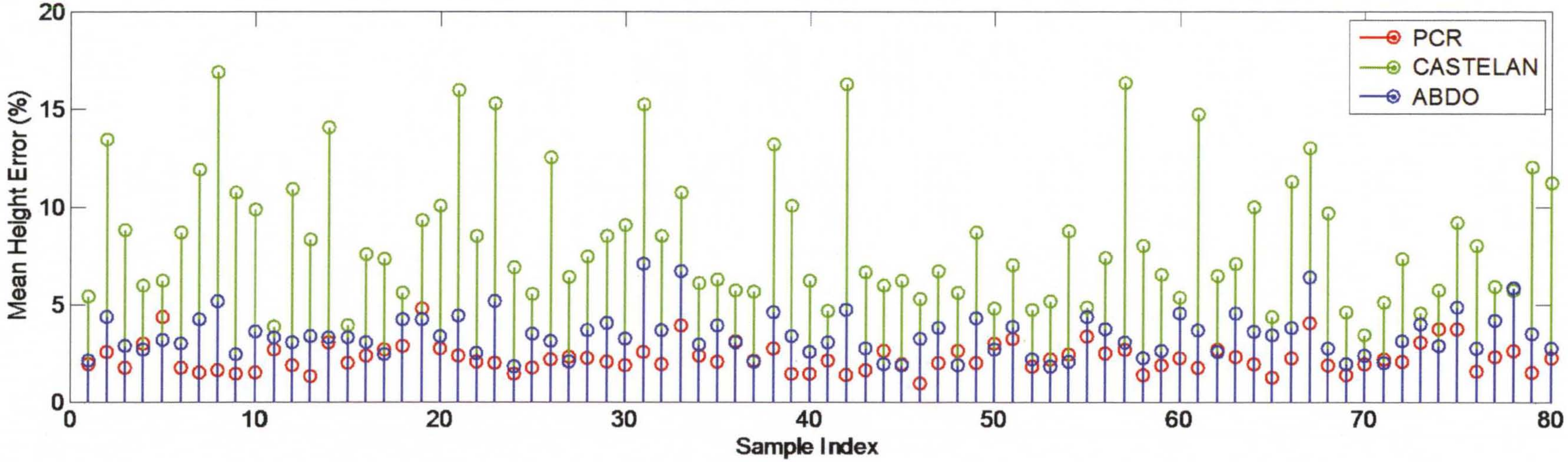


FIGURE 50 – Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Note that the proposed method (Method II-PCR) outperforms its predecessors. Average error for all 80 out-of-training samples is 3.33%, 9.73%, and 6.07% for the PCR-Method II, Castelan, and Ahmed approaches, respectively.

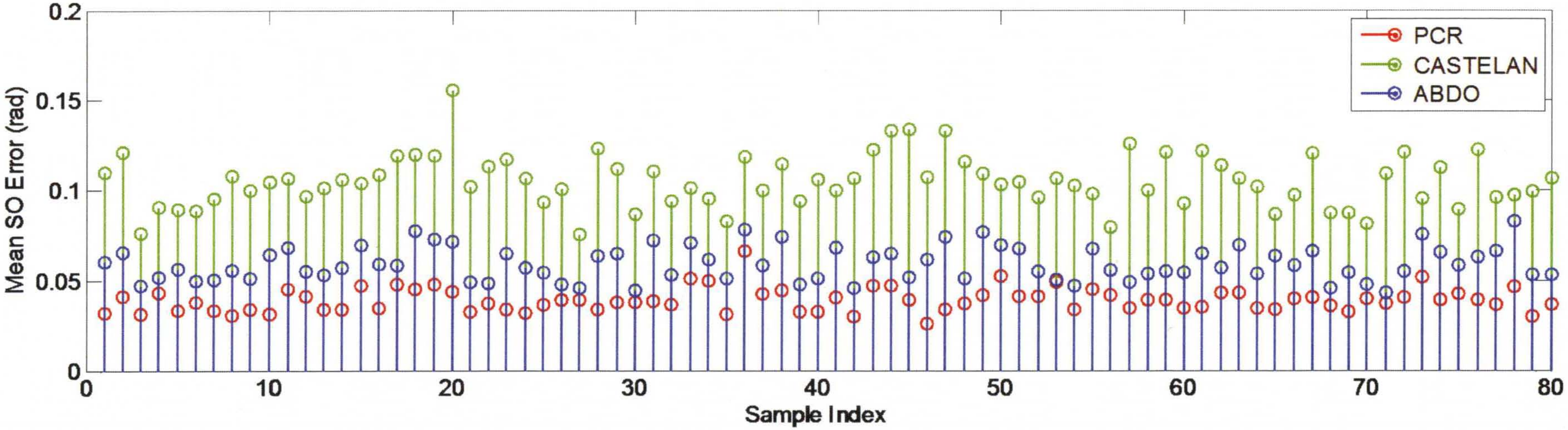


FIGURE 51 – Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Note that the proposed method (Method II-PCR) outperforms its predecessors. Average error for all 80 out-of-training samples is 0.06 rad, 0.11 rad, and 0.06 rad for the PCR-Method II, Castelan, and Ahmed approaches, respectively.

G. Robust Model-based Shape-from-shading Framework

The next sections will discuss several ways on how to extend the basic framework of model-based shape-from-shading under fixed pose and general lighting. Since the experimental results of the three algorithms from the previous chapter yield similar results, the fastest one (**Method II**), which is based on Principal Component Regression (PCR), is chosen as the primary 3D face recovery method. The first extension deals with how to incorporate the concept of robust spherical harmonics projection (SHP) images from the previous chapter into the model-based shape recovery framework. The second extension will incorporate variable pose into the framework, transforming it into an algorithm that can deal with both general pose and lighting.

The previous chapter on robust SHP images dealt with instances when the input images are corrupted by some form of non-ideal conditions, be it noise or occlusion. Recall that Algorithm **1** solves for the SHP image, $h = \mathbf{B}\tilde{\boldsymbol{\alpha}}$, where

$$\tilde{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \|I - \mathbf{B}\boldsymbol{\alpha}\| \tag{64}$$

$\tilde{\boldsymbol{\alpha}}$ is the solved SHP coefficient, I is the input image, and $\mathbf{B}$ is the matrix of basis images. This formulation is problematic in the presence of outliers. Algorithm **2** tries to remedy this problem by modifying (64) with

$$\min_{\boldsymbol{\alpha}} E_D(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}} \sum_{s \in S} \rho(I - \mathbf{B}\boldsymbol{\alpha}, \sigma_s) \tag{65}$$

where σ_s is a scale parameter and ρ is the robust ρ-function. The robust function decreases the effect of outliers, giving a solution as if the outliers did not exist.

The third step of the PCR framework for 3D shape recovery involves building the SHP model; Algorithm **2** will be used to solve for the SHP images, instead of Algorithm **1**. In addition, the occlusion estimate m_i (Figure 33) for each computed h_i is stored for later use. The final occlusion estimate m_f is determined by binary *and*-ing all occlusion estimates for each h_i, i.e.,

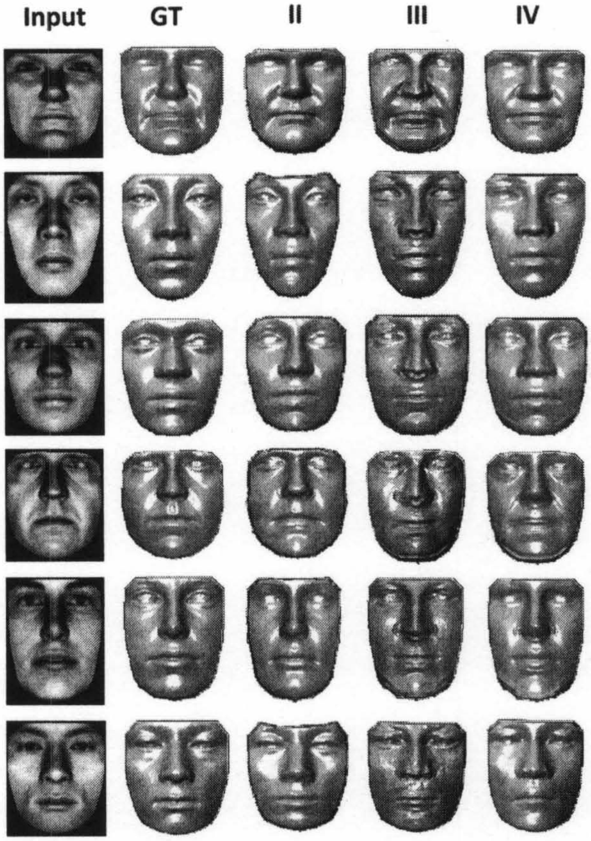


FIGURE 52 – Recovered shapes, together with the input image and ground-truth (GT) shape, for the three model-based methods. Methods III and IV refer to Castelan et al. [1] [14] and Ahmed et al. [2], respectively. The results between Methods II and IV are close visually. Castelan's version (Method III) suffers with input images that have illumination.

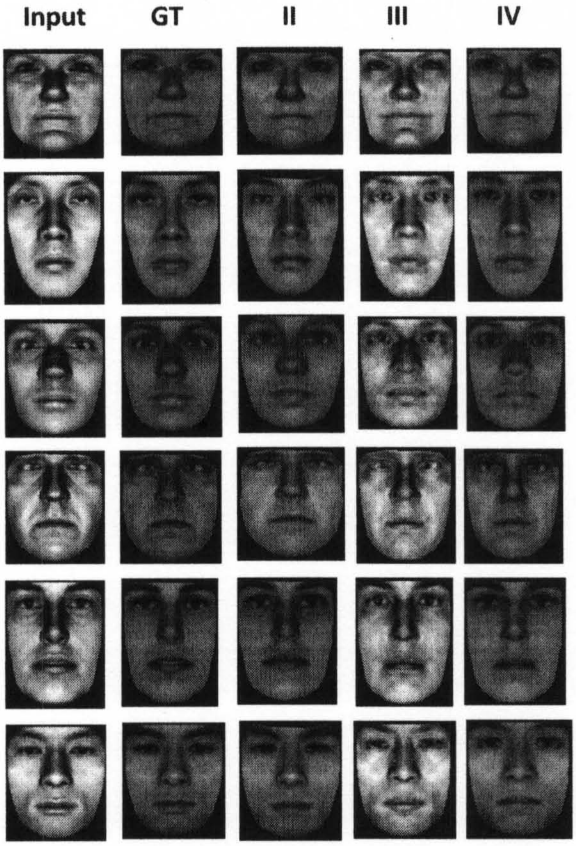
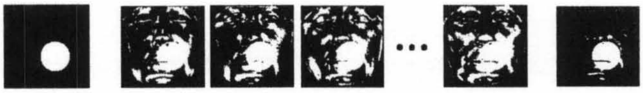


FIGURE 53 – Recovered albedo, together with the input image and ground-truth (GT) albedo, for the three model-based methods. Methods III and IV refer to Castelan et al. [1] [14] and Ahmed et al. [2], respectively. The results between Methods II and IV are close visually. Castelan's version (Method III) suffers with input images that have illumination.



(a) (b) (c)

FIGURE 54 – (a) True occlusion mask. (b) Occlusion mask estimates m_i, for each h_i. (c) Occlusion mask estimate using (66).

$$m_f = m_1 \bullet m_2 \bullet \cdots \bullet m_n \tag{66}$$

Figure 54 illustrates the final occlusion estimate m_f, together with the true occlusion mask. In the computation of SHP coefficients for the input image at the ninth step of Algorithm **5**, the occluded pixels (from the occlusion estimate) are neglected, i.e.,

$$\mathbf{b}_{h,inp} = P_{h,noc}^T (I_{inp,noc} - \bar{\mathbf{h}}_{noc}) \tag{67}$$

where noc refers to rows corresponding to non-occluded pixels. The algorithm listing below shows the robust model-based framework, a modified version of Algorithm **5**.

1. Experimental Results

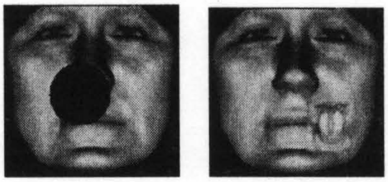
This section shows experimental results of the robust model-based shape-from-shading framework in Algorithm **6**. Just like in the previous sections, the face model is built using the USF 3D Face Database [67]. To quantify reconstruction accuracy, the recovered shapes of 80 out-of-training USF samples, illuminated with combined light sources of $(0, 0, 1)$ and $(0, 0.5, 0.9)$ are studied. For each shape recovery, two measures related to height (62) and surface orientation (63) error are used.

The two types of occlusion used in these experiments are illustrated in Figure 55. Figure 55(a) is a circular occlusion region with all-black color (i.e., gray-level value is 0) and Figure 55(b) is a contiguous type of occlusion [69] that is, in this case, a baboon image. The radius of the circular occlusion is 25 pixels and one side of the square contiguous occlusion is 40 pixels. Both occlusion types are placed randomly inside the input image region of the 80 out-of-training USF samples.

There are three sets of experiments that will be compared in this section,

Algorithm 6 Robust Model-based 3D Shape Recovery Framework

```
INPUT: (a) Input image, $I_{inp}$ (b) Shape and albedo samples: $(\mathbf{s}_1, \mathbf{a}_1)$ to $(\mathbf{s}_n, \mathbf{a}_n)$
OUTPUT: (a) Recovered shape, $\tilde{\mathbf{s}}$ (b) Recovered albedo, $\tilde{\mathbf{a}}$
1: Build the shape and albedo models from the samples using PCA: Construct $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s\mathbf{b}_s$ and $\mathbf{a} = \bar{\mathbf{a}} + \mathbf{P}_a\mathbf{b}_a$.
2: Construct the basis images for each pair: $\mathbf{B}_i = [b_1, \cdots, b_9]$ for each pair $(\mathbf{s}_i, \mathbf{a}_i)$
3: Build the SHP model: Given an input $I_{inp}$, solve for the SHP images, $(h_1, h_2, \cdots, h_n)$, for all samples using Algorithm 2, save occlusion estimate, $m_i$ for each sample, and then construct, $\mathbf{h} = \bar{\mathbf{h}} + \mathbf{P}_h\mathbf{b}_h$
4: Get final occlusion estimate: $m_f = m_1 \bullet m_2 \bullet \cdots \bullet m_n$
5: Replace the shape samples with its coefficients: Solve for $\mathbf{b}_{si} = P_s^T(\mathbf{s}_i - \bar{\mathbf{s}})$
6: Replace the texture samples with its coefficients: Solve for $\mathbf{b}_{ai} = P_a^T(\mathbf{a}_i - \bar{\mathbf{a}})$
7: Replace the SHP images with its coefficients: Solve for $\mathbf{b}_{hi} = P_h^T(\mathbf{h}_i - \bar{\mathbf{h}})$
8: Setup matrices for PCR: $\mathbf{T} = [\mathbf{b}_{h_1}^T, \cdots, \mathbf{b}_{h_n}^T]$, $\mathbf{U}_{sh} = [\mathbf{b}_{s_1}^T, \cdots, \mathbf{b}_{s_n}^T]$ and $\mathbf{U}_{al} = [\mathbf{b}_{a_1}^T, \cdots, \mathbf{b}_{a_n}^T]$
9: Build two PCR models: Construct $\tilde{\mathbf{C}}_{sh} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U}_{sh}$ and $\tilde{\mathbf{C}}_{al} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U}_{al}$
10: Solve for the SHP coefficients of the input image using only non-occluded pixels from Step 4: Get $\mathbf{b}_{h,inp} = P_{h,noc}^T(I_{inp,noc} - \bar{\mathbf{h}}_{noc})$
11: Solve for the shape and texture coefficients: Get $\tilde{\mathbf{b}}_s^T = \tilde{\mathbf{b}}_h^T\tilde{\mathbf{C}}_{sh}$ and $\tilde{\mathbf{b}}_a^T = \tilde{\mathbf{b}}_h^T\tilde{\mathbf{C}}_{al}$
12: Solve for the recovered shape and albedo: $\tilde{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{P}_s\tilde{\mathbf{b}}_s$ and $\tilde{\mathbf{a}} = \bar{\mathbf{a}} + \mathbf{P}_a\tilde{\mathbf{b}}_a$
```



(a) (b)

FIGURE 55 – Two types of occlusion: (a) circular occlusion with uniform color, (b) contiguous occlusion with non-uniform texture.

namely: (a) **Case I** - using the original PCR framework in Algorithm **5** on input images with no occlusion, (b) **Case II** - using the modified robust framework in Algorithm **6** on input images with occlusion, and (c) **Case III** - using the original PCR framework (without robust formulation) in Algorithm **5** on input images with occlusion.

Similar to the previous sections, the comparison starts with a side-by-side visualization of the mean height (Figure 56) and surface orientation (Figure 57) error stem plots, for the three sets of experiments. The occlusion type is a circular region (Figure 55a). Cases I, II, and III refer to PCR-none-none, PCR-circle-Geman-McClure, and PCR-circle-none, respectively, in the plot legends.

The actual mean and standard deviation of the mean height error and mean surface orientation error of the 80 out-of-training USF samples, for the above experiments, are shown in Tables 7 and 8. It is clear that the modified robust framework for shape recovery (Algorithm 6) can handle this type of occlusion, i.e., Case II results are closer to Case I. Without the robust framework (Case III), shape recovery results have larger errors, as seen in Tables 7 and 8.

TABLE 7
CIRCULAR OCCLUSION EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN HEIGHT ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II	Case III
$\mu_{\bar{s}_{err}}$ (%)	2.28	2.62	3.45
$\sigma_{\bar{s}_{err}}$ (%)	0.74	0.89	0.87

Figure 58 shows the reconstruction results, visually, of five out-out-training USF samples for the three cases above, using the circular occlusion in Figure 55(a). The input images are in the first row. Note the random position of the circular occlusion in the images. The second row contains the estimated occlusion in the input image. The third row displays the ground-truth albedo of the input. Fourth

TABLE 8
CIRCULAR OCCLUSION EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN SURFACE ORIENTATION ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II	Case III
$\mu_{\bar{\theta}_{err}}$ (rad)	0.04	0.04	0.06
$\sigma_{\bar{\theta}_{err}}$ (rad)	0.01	0.01	0.01

to sixth rows contain the recovered albedos for **Cases I, II,** and **III,** respectively. The seventh row contains the ground-truth shape. Eighth to tenth row illustrate the recovered shapes for **Cases I, II,** and **III,** respectively. Notice that the recovered albedo and shapes for **Cases I** and **II** are similar visually, reflecting the similar errors in Tables 7 and 8. The recovered shapes and albedo, however, for **Case III** appear to be degraded, reflecting the large error values in Tables 7 and 8.

For the contiguous occlusion type (Figure 55b), the comparison commences with a side-by-side visualization of the mean height (Figure 59) and surface orientation (Figure 60) error stem plots, for the three sets of experiments. Cases I, II, and III refer to PCR-none-none, PCR-circle-Geman-McClure, and PCR-circle-none, respectively, in the plot legends.

The actual mean and standard deviation of the mean height error and mean surface orientation error of the 80 out-of-training USF samples, for the above experiments, are shown in Tables 9 and 10. Similar to the circular type of occlusion, it is clear that the modified robust framework for shape recovery (Algorithm 6) can handle this type of occlusion, i.e., Case II results are closer to Case I. Without the robust framework (Case III), shape recovery results have larger errors, as seen in Tables 9 and 10.

Figure 61 shows the reconstruction results, visually, of five out-out-training USF samples for the three cases above, using the contiguous occlusion in Figure 55(b). The input images are in the first row. Note the random position of the rect-

TABLE 9
CONTIGUOUS OCCLUSION EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN HEIGHT ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II	Case III
$\mu_{\bar{s}_{err}}$ (%)	2.28	2.47	2.80
$\sigma_{\bar{s}_{err}}$ (%)	0.74	0.80	0.89

TABLE 10
CONTIGUOUS OCCLUSION EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN SURFACE ORIENTATION ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II	Case III
$\mu_{\bar{\theta}_{err}}$ (rad)	0.04	0.04	0.05
$\sigma_{\bar{\theta}_{err}}$ (rad)	0.01	0.01	0.01

angular contiguous occlusion in the input. The second row contains the estimated occlusion in the input image. The third row displays the ground-truth albedo of the input. Fourth to sixth rows contain the recovered albedos for **Cases I, II,** and **III**, respectively. The seventh row contains the ground-truth shape. Eighth to tenth row illustrate the recovered shapes for **Cases I, II,** and **III**, respectively. Notice that the recovered albedo and shapes for **Cases I** and **II** are similar visually.

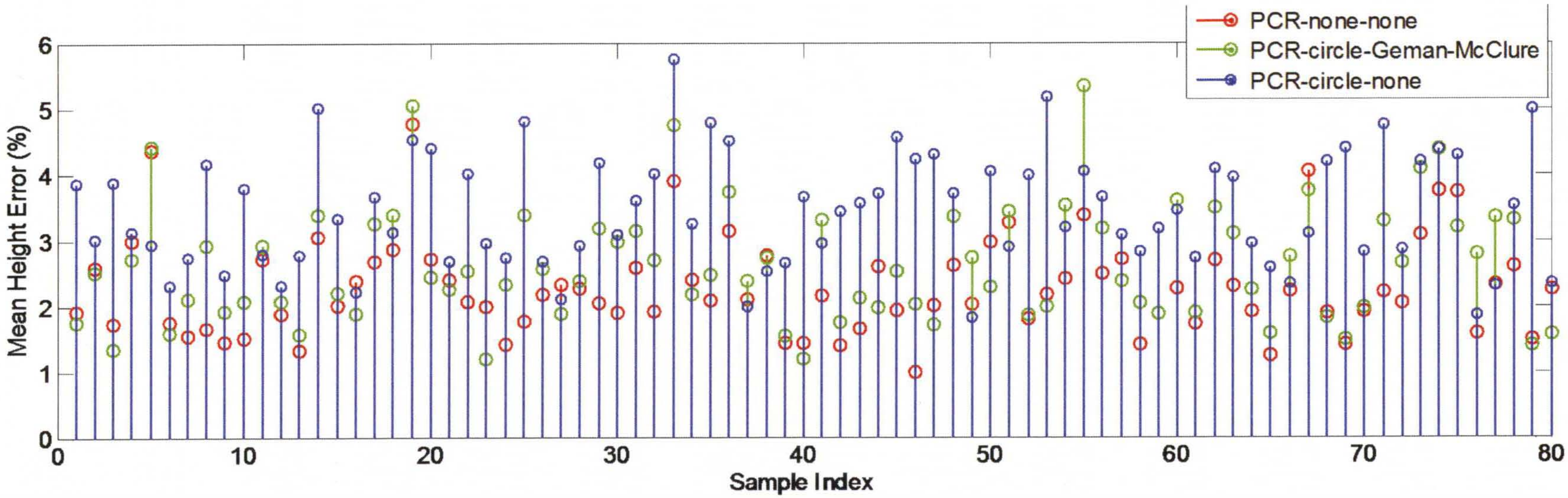


FIGURE 56– Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 3.33%, 3.61%, and 4.46% for the Case I, Case II, and Case III experiments, respectively. It is clear that the modified robust framework for shape recovery (Algorithm 6) can handle this type of occlusion, i.e., Case II results are closer to Case I. Without the robust framework (Case III), shape recovery results have larger errors.

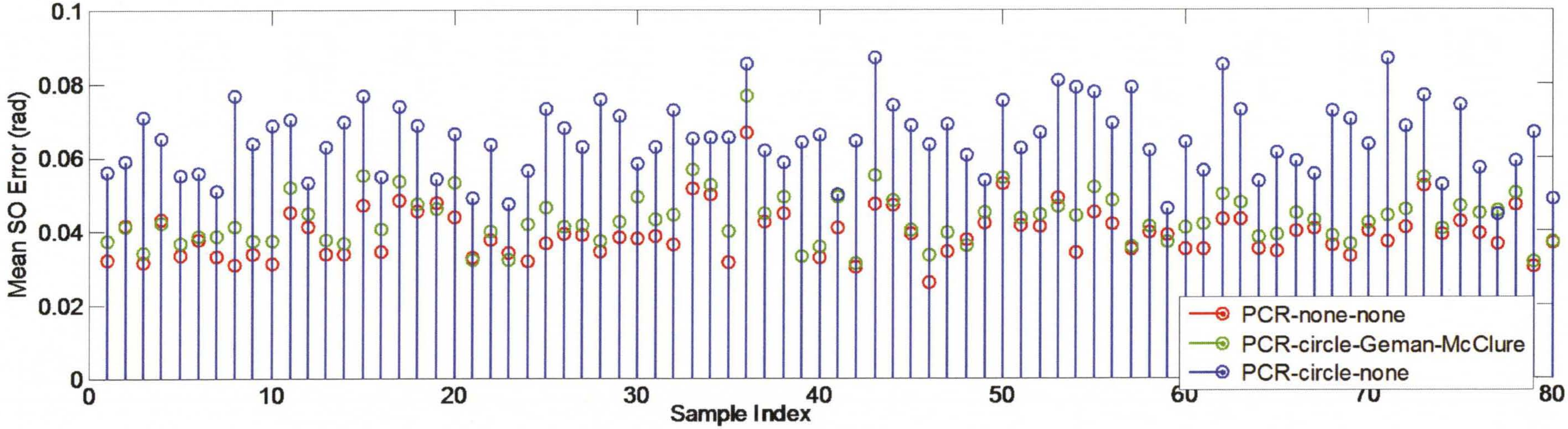


FIGURE 57 – Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 0.06 rad, 0.06 rad, and 0.08 rad% for the Case I, Case II, and Case III experiments, respectively.

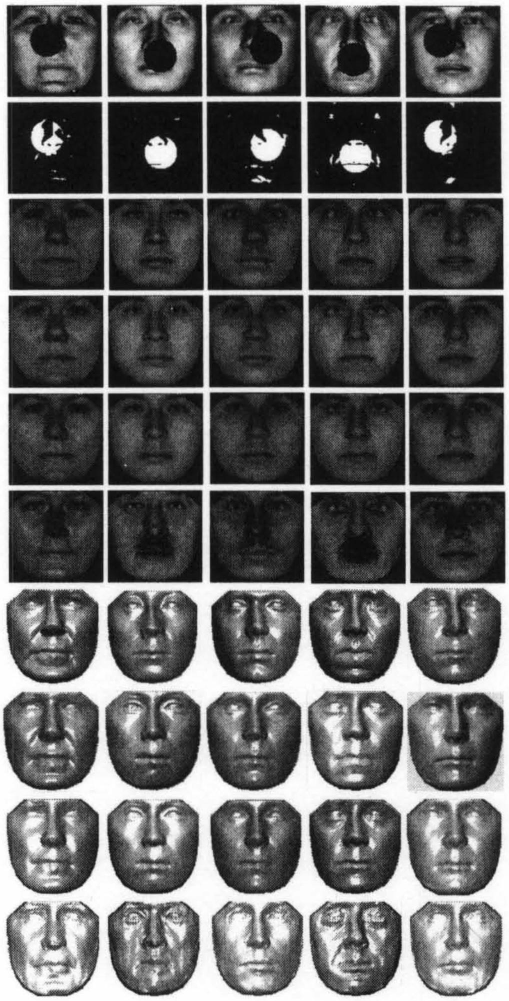


FIGURE 58 – Reconstruction results of five out-out-training USF samples for the three cases above, using the circular occlusion. Input images are in the first row. Second row contains the estimated occlusion in the input image. Third row displays the ground-truth albedo of the input. Fourth to sixth rows contain the recovered albedos for **Cases I**, **II**, and **III**, respectively. Seventh row contains the ground-truth shape. Eighth to tenth row illustrate the recovered shapes for **Cases I**, **II**, and **III**, respectively.

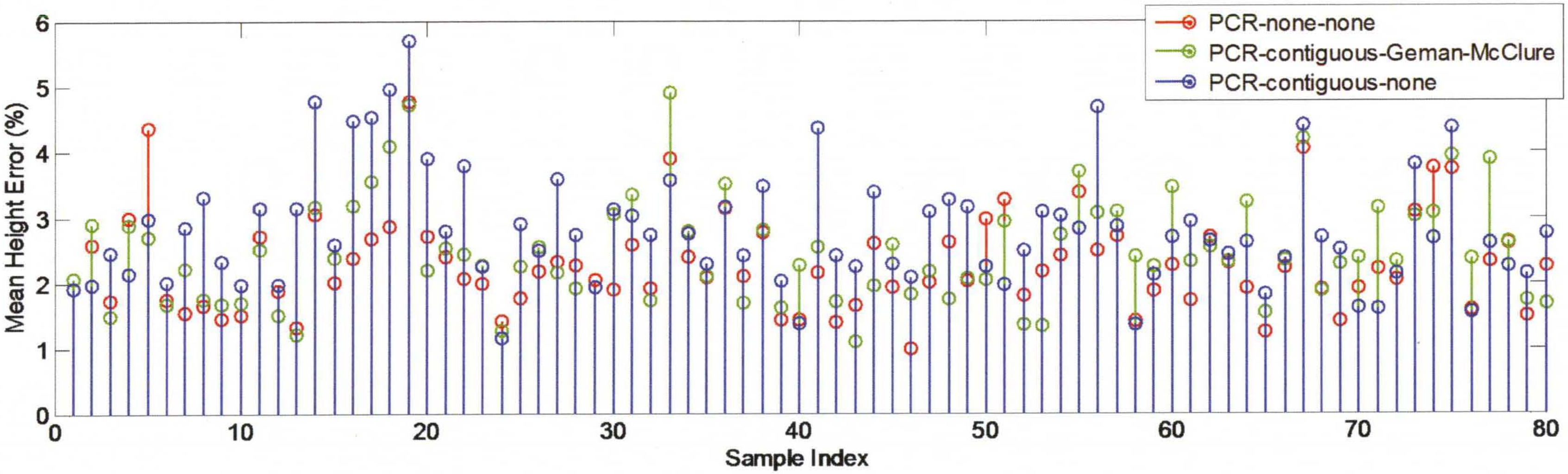


FIGURE 59 – Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 3.33%, 3.52%, and 3.84% for the Iterative, Method I, and Method II approaches, respectively.

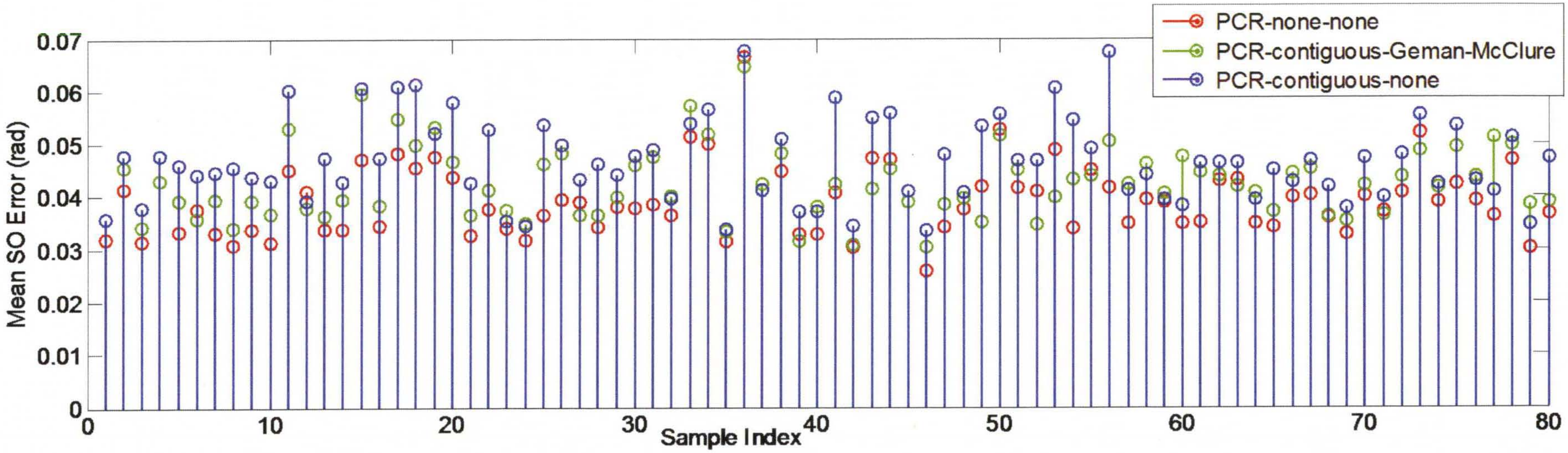


FIGURE 60– Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average error for all 80 out-of-training samples is 0.06 rad, 0.06 rad, and 0.06 rad% for the Case I, Case II, and Case III experiments, respectively.

H. Model-based 3D Face Recovery: A Coupled Statistical Model Approach (General Illumination and Unknown Pose)

The goal of this section is to extend the proposed 3D shape recovery approaches from the previous sections to handle variable pose, as opposed to the limiting case of fixed frontal pose. Just like in the robust version of the model-based SFS approach, the best one (Method II) is chosen as the primary 3D face recovery method.

For the general pose extension, Figure 38 of the fixed frontal pose approach can be updated to Figure 62. Note that the main difference is in the upper right corner of the figure, i.e., there is a presence of pose in the input. However, the dense correspondence between the USF database samples and the input image, through the mean shape mask (acting as the reference frame) still exists. Recall that in the fixed frontal pose, this dense correspondence is achieved through the use of the thin-plate spines (TPS) warping function.

1. Preprocessing and Postprocessing Steps

Recall in the frontal pose case previously that the set of control points in the mean shape and input image are denoted as $(x_{i,c}, y_{i,c})$ and $(x'_{i,c}, y'_{i,c})$, respectively. The preprocessing step involves solving for the vector-valued warping function, $\mathbf{f}(x_{i,c}, y_{i,c}) = (x'_{i,c}, y'_{i,c})$.

For this general pose case, the 2D locations of the input image feature points are represented as $\mathbf{x}_i \in \mathbb{R}^3$, and the corresponding 3D locations of the feature points of the mean shape as $\mathbf{X}_i \in \mathbb{R}^4$, using homogeneous coordinates. The goal is to find the (3×4) camera projection matrix **C**, i.e., $\mathbf{x}_i = \mathbf{C}\mathbf{X}_i$.

To estimate this projection matrix, the normalized versions of the feature points are needed, i.e., $\tilde{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$ and $\tilde{\mathbf{X}}_i = \mathbf{U}\mathbf{X}_i$, where $\mathbf{T} \in \mathbb{R}^{3\times4}$ and $\mathbf{U} \in \mathbb{R}^{4\times4}$ are similarity transforms that translate the centroid of the 2D and 3D feature points to the origin and scale them such that the RMS distance from the origin is $\sqrt(2)$ for

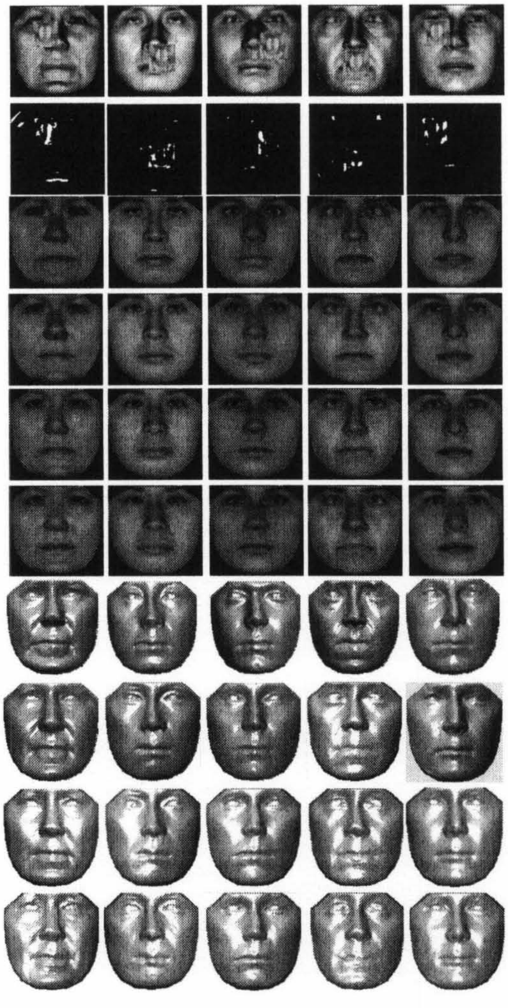


FIGURE 61 – Reconstruction results of five out-out-training USF samples for the three cases above, using contiguous occlusion. Input images are in the first row. Second row contains the estimated occlusion in the input image. Third row displays the ground-truth albedo of the input. Fourth to sixth rows contain the recovered albedos for **Cases I, II,** and **III**, respectively. Seventh row contains the ground-truth shape. Eighth to tenth row illustrate the recovered shapes for **Cases I, II,** and **III**, respectively.

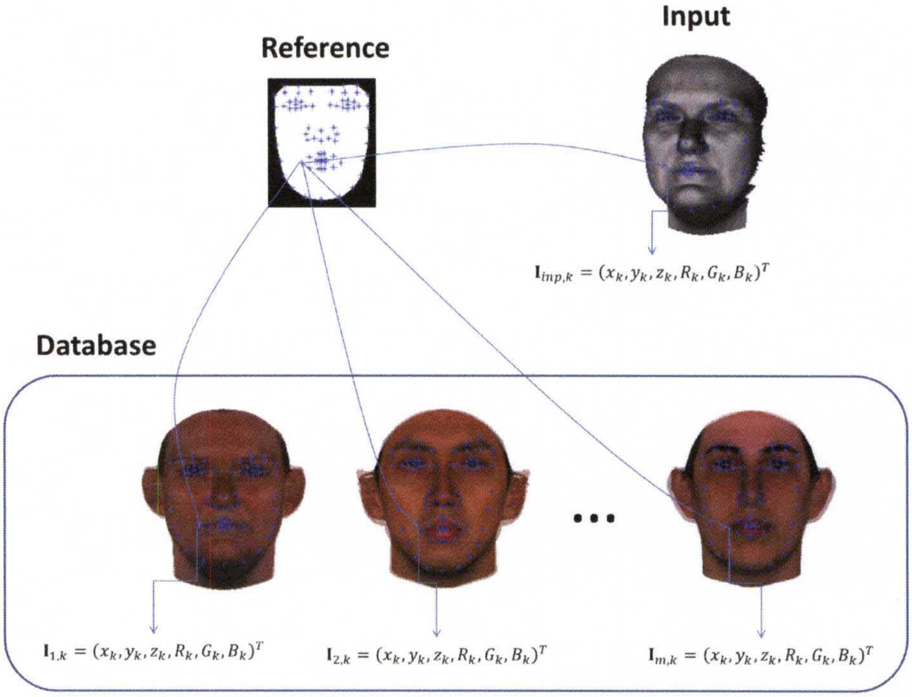

Input
Reference
$\mathbf{I}_{inp,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$
Database
$\mathbf{I}_{1,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$
$\mathbf{I}_{2,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$
…
$\mathbf{I}_{m,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$

FIGURE 62 – (**Upper Left**) Mean shape together with its convex hull. The convex hull forms the boundary of the mask. (**Lower**) Each pixel position within the mask corresponds to a certain point on a sample scan in the USF database. The vector $\mathbf{I}_{i,k} = (x_k, y_k, z_k, R_k, G_k, B_k)^T$ refers to the shape and texture information at the kth vertex of the ith sample scan. (**Upper Right**) Similarly, each pixel position within the mask corresponds to a certain point on the input image. Hence, there is correspondence between the USF database samples and the input image, through the mean shape mask.

the 2D case and $\sqrt(3)$ for the 3D case.

This work will assume an affine camera; the Gold Standard Algorithm (Algorithm 7.1) [3] can compute the normalized projection matrix $\tilde{\mathbf{C}}$. The desired camera projection matrix can be solved from $\tilde{\mathbf{C}}$ following a denormalization step, $\mathbf{C} = \mathbf{T}^{-1}\tilde{\mathbf{C}}\mathbf{U}$.

The solved camera projection matrix **C**, in conjunction with the z-buffer test, is used to determine hidden triangle faces. Only vertices belonging to the visible triangle faces are considered. Figure 77 illustrates the computation of camera projection matrix. Notice that the projected 3D feature points do not coincide with the input image feature points. A warping function (**f**) using thin-plate splines (TPS) is established between the projected 3D and input image feature points. Therefore, a 3D point in the mean face is first projected to 2D space using the camera projection matrix **C**, after which it is warped using **f**. The input image I is then sampled, i.e., $I_s = I(\mathbf{g}(x_i, y_i))$, where (x_i, y_i) are pixels inside the mean shape mask (reference frame) and **g** represents the combined camera projection matrix and TPS warp.

For the fixed frontal case, the postprocessing step involves solving the inverse warping function, $\mathbf{f}'(x'_{i,c}, y'_{i,c}) = (x_{i,c}, y_{i,c})$. This is not possible for the general pose case because input image feature points are with pose; it is desirable to get the equivalent frontal pose of the image feature points, first, before performing the inverse warping function. This work makes use of the recent results of [4] to get the equivalent frontal pose of the image feature points with pose. Related methods of [4] will be pursued in the next chapter.

2. Experimental Results

This section shows experiments to evaluate the performance of the proposed method in recovering the 3D face shape. The face models are built using the USF 3D Face Database [67], which contains 100 subjects of diverse gender and ethnicity. Out of these 100 samples, 80 subjects were deemed to be acceptable and

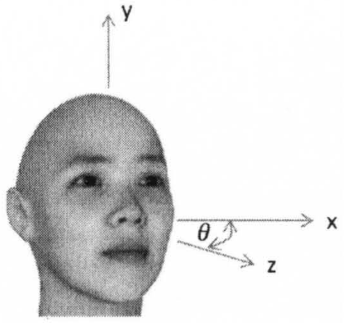


FIGURE 63 – Pan angle illustration. The face moves left-to-right or right-to-left sideways.

were subsequently chosen to build the shape and texture models.

Experiment A To quantify the reconstruction accuracy, the author recovers the 3D shape for 80 out-of-training USF samples illuminated with combined light source directions of $(0, 0, 1)$ and $(0, 0.5, 0.9)$. Two types of input images are generated: (a) Case I - an input image is generated with a random pan angle within the range of (-20° to 20°) and (b) Case II - another input image from the same subject is generated with a frontal pose, i.e., pan angle is 0°. For each input image, the two error measures, mean height error (62) and mean surface orientation error (63), are measured.

Figures 65 and 66 show stem plots of the height and surface orientation errors for two cases: (a) Case I - input image contains pose and is recovered by the model-based approach for general pose and illumination and (b) Case II - input image from the same subject but without pose and reconstructed by Method II. Notice that the results are similar, i.e., the newer framework can recover the 3D shape even if there is pose involved.

The actual mean and standard deviation of the mean height error and mean surface orientation error of the 80 out-of-training USF samples, for these experiments, are shown in Tables 11 and 12. The numerical values indicate that the pose-invariant 3D shape recovery version (for input images with pose) has similar error values compared to the frontal-pose version (for input images with frontal pose), for input images of the same subject.

The next step is to place alongside each other the recovered shapes and albedo, together with the input, ground-truth shape and albedo, as illustrated in Figure 64. The results are very close visually, indicating the the variable pose do not affect the 3D shape recovery framework.

Experiment B To investigate the sensitivity of the proposed approach with respect to pose, a similar experiment is performed as described, except that for a specific pose, the recovered 3D shapes of all 80 out-of-training input images are analyzed. Specifically, at pan angle of x°, generate 80 out-of-sample input images and compute the average mean height error and surface orientation error across all 80 input images, where $x \in \{-20^\circ, -15^\circ, \cdots, 15^\circ, 20^\circ\}$.

Figures 67 and 69 plot the average mean height error and average mean surface orientation error, respectively, with respect to a pan angle range of (-20° to 20°). The bar graphs indicate that the proposed method is not sensitive to pose changes, at least for the pan angle range of $\{-20^\circ, -15^\circ, \cdots, 15^\circ, 20^\circ\}$.

Experiment C The previous experiment does not show at what point pose begins to have an effect. The next obvious step is to increase the pan angle range, i.e., $x \in \{-90^\circ, -85^\circ, \cdots, 85^\circ, 90^\circ\}$, and perform Experiment B. However, this experiment will take a considerable amount of time to perform.

A quick solution is to choose only one subject (sample index #1 in this experiment), instead of all 80 subjects, from the USF database and then perform Experiment B but with a pan angle range of $x \in \{-90^\circ, -85^\circ, \cdots, 85^\circ, 90^\circ\}$.

Figures 71 and 72 show both mean height error and surface orientation error across various pan angles. To highlight the change in error per pan angle incremental change (5°), Figures 73 and 74 show the forward-difference numerical differentiations of the mean height error and surface orientation error plots. From a visual perspective, it appears that the breaking point of pose invariance is approximately at $\pm 50^\circ$. Hence, there is only pose invariance inside the pan angle range of (-50° to 50°); outside of this range, the recovered 3D shape is not accurate due to the effect of pose.

I. Summary

This chapter proposed a 3D facial shape recovery method for images under general pose and illumination. The first part of the chapter deals with fixed frontal pose only. Several ideas in this chapter are published in [40] [41] [42].

Three algorithms were developed to solve this 3D shape recovery problem, starting from a brute-force iterative approach to a computationally efficient regression method, where the classical shape-from-shading equation is cast as a regression framework. Results show that the output of the regression-like approach is faster in timing and similar in error metrics when compared to its iterative counterpart.

The best of the three algorithms above, Method II-PCR, is compared to its two predecessors, namely: (a) Castelan et al. [1] and (b) Ahmed et al. [2]. It is clear from the experimental results that the proposed method (Method II-PCR) is superior in all aspects compared to the previous state-of-the-art. Robust statistics was also incorporated into the shape recovery framework to deal with noise and occlusion.

Using multiple-view geometry concepts [3], the fixed frontal pose was relaxed to arbitrary pose. The best of the three algorithms above, Method II-PCR, once again is used as the primary 3D shape recovery method. Results show that the pose-invariant 3D shape recovery version (for input with pose) has similar error values compared to the frontal-pose version (for input with frontal pose), for input images of the same subject. Sensitivity experiments indicate that the proposed method is, indeed, invariant to pose, at least for the pan angle range of $(-50^\circ \text{ to } 50^\circ)$.

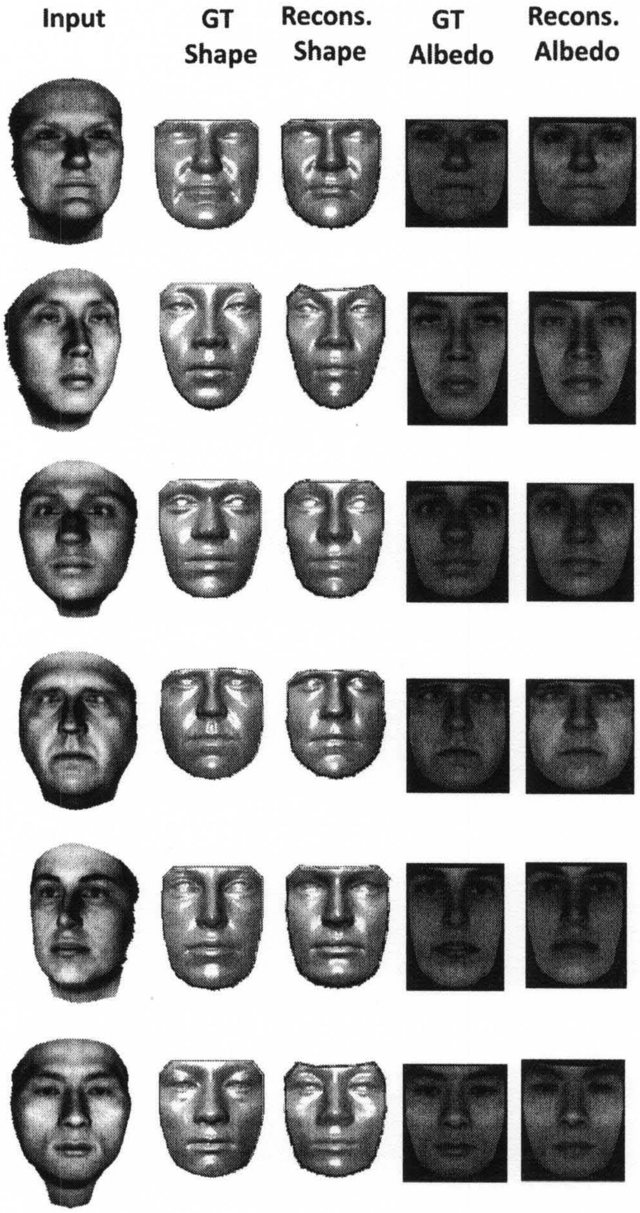


FIGURE 64 – Recovered shapes and albedo, together with the input image and ground-truth (GT) shapes and albedo.

TABLE 11
UNKNOWN POSE EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN HEIGHT ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II
$\mu_{\bar{s}_{err}}$ (%)	2.34	2.28
$\sigma_{\bar{s}_{err}}$ (%)	0.73	0.74

TABLE 12
UNKNOWN POSE EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN SURFACE ORIENTATION ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II
$\mu_{\bar{\theta}_{err}}$ (rad)	0.039	0.040
$\sigma_{\bar{\theta}_{err}}$ (rad)	0.010	0.010

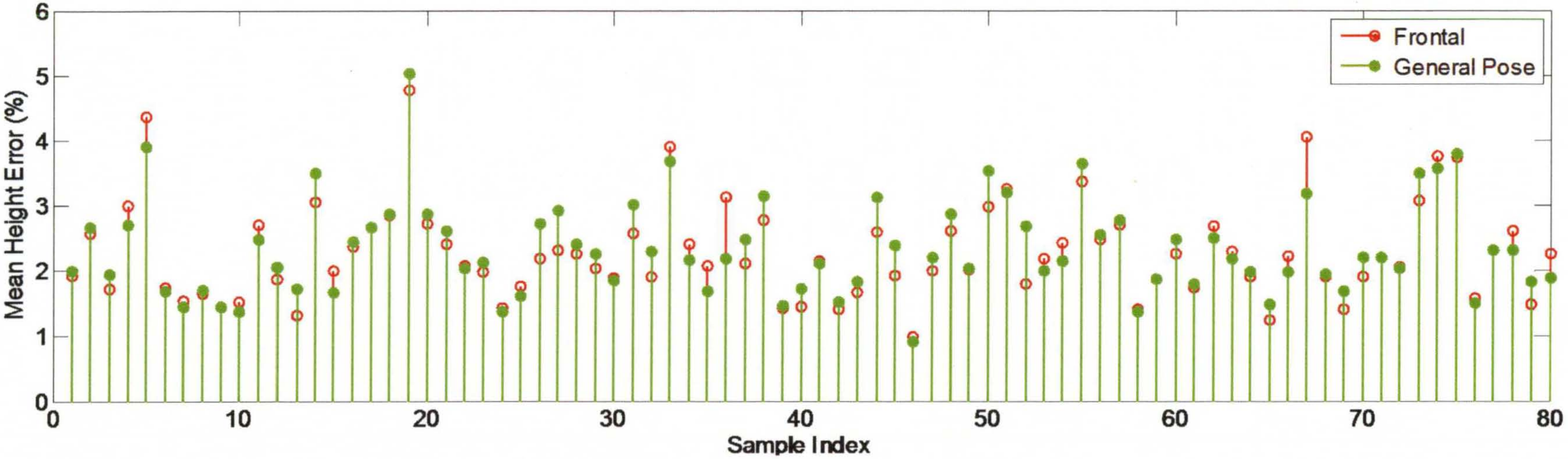


FIGURE 65 – Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are General Pose (Case I) and Frontal (Case II). Note that the proposed method for general pose behaves similarly to Method II-PCR for frontal poses. Average error for all 80 out-of-training samples is 2.34% and 2.28% for the Case I and Case II experiments, respectively.

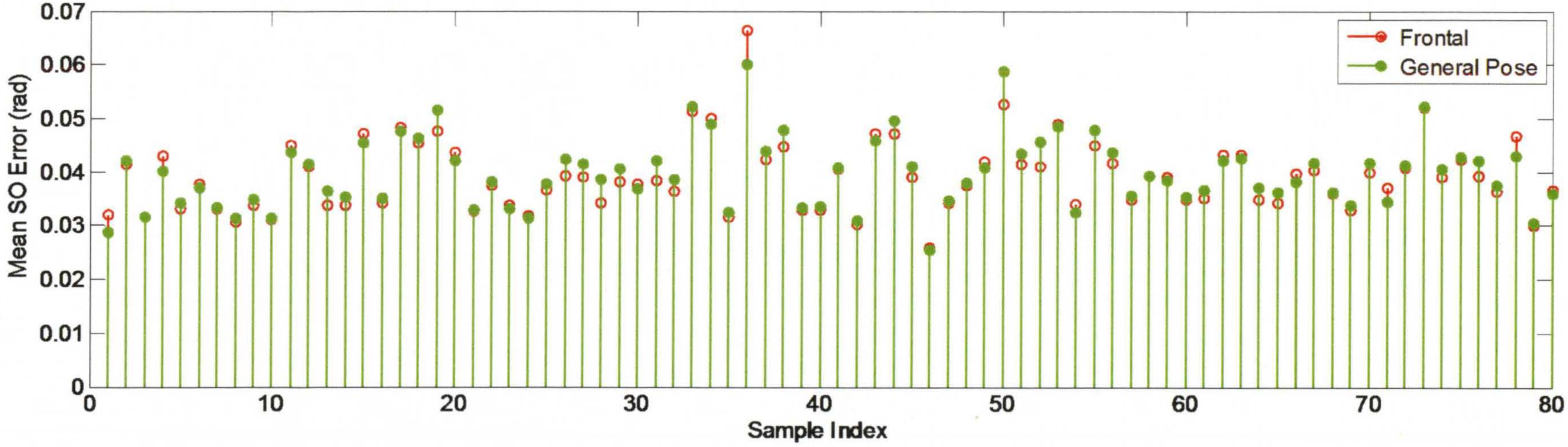


FIGURE 66– Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are General Pose (Case I) and Frontal (Case II). Note that the proposed method for general pose behaves similarly to Method II-PCR for frontal poses. Average error for all 80 out-of-training samples is 0.04 rad and 0.04 rad for the Case I and Case II experiments, respectively.

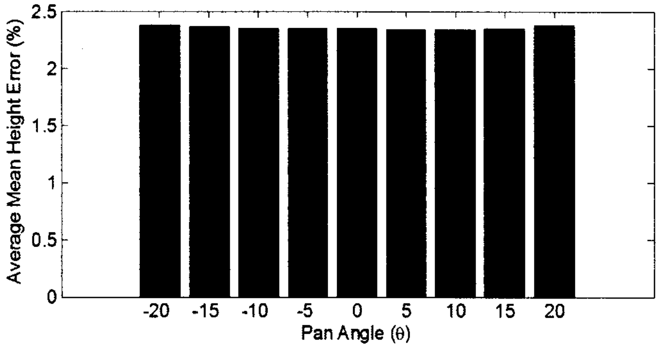


FIGURE 67– Bar graph of the average mean height error with respect to pose changes, i.e., pan angle range of ($-20°$ to $20°$). The graph indicates that the proposed approach is insensitive to pose changes.

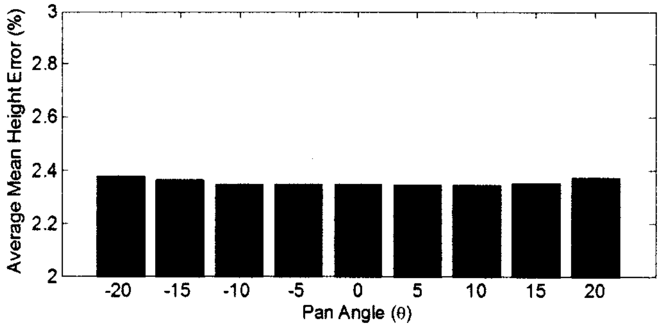


FIGURE 68– Zoomed version of Figure 67. The plot shows that there is a slight increase in error as the pose angle is increased.

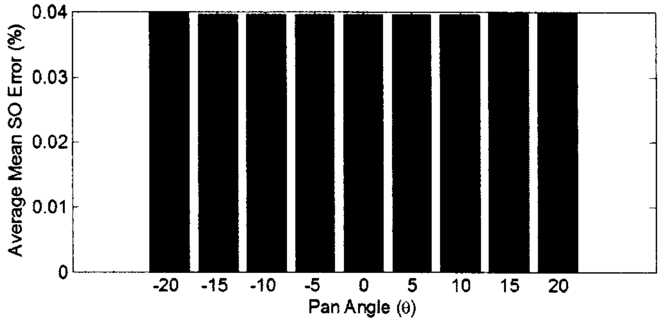


FIGURE 69– Bar graph of the average mean surface orientation error with respect to pose changes, i.e., pan angle range of ($-20°$ to $20°$). The graph indicates that the proposed approach is insensitive to pose changes.

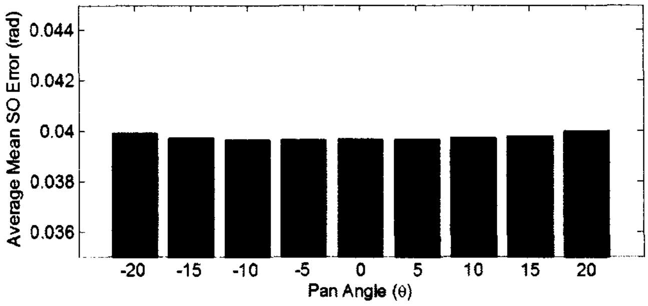


FIGURE 70 – Zoomed version of Figure 69. The plot shows that there is a slight increase in error as the pose angle is increased.

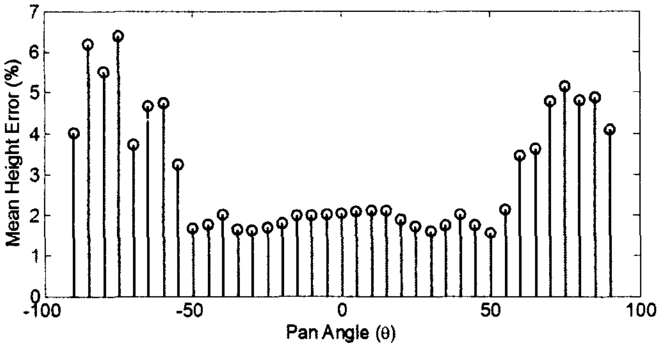


FIGURE 71 – Stem plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (−90° to 90°). The plot indicates that the proposed approach is insensitive to pose changes only up to ±50°.

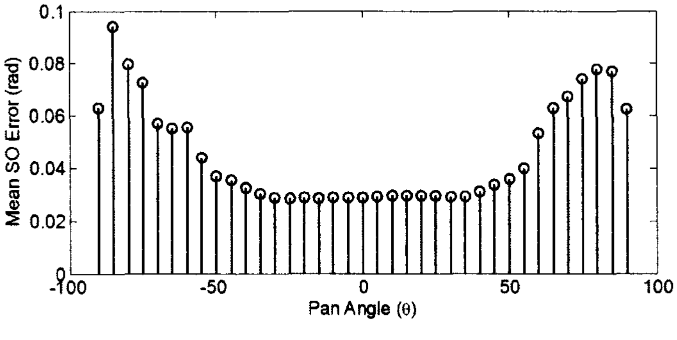


FIGURE 72 – Stem plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (−90° to 90°). The graph indicates that the proposed approach is insensitive to pose changes only up to ±50°.

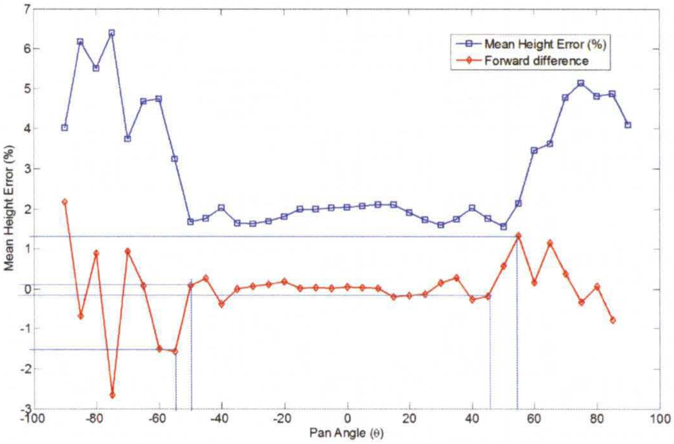


FIGURE 73 – Plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of ($-90°$ to $90°$). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change ($5°$). At pan angle $-55°$ and $55°$, there is a change of about -1.5% and 1.5% error, respectively. The plot indicates that the proposed approach is insensitive to pose changes only up to about $\pm 50°$.

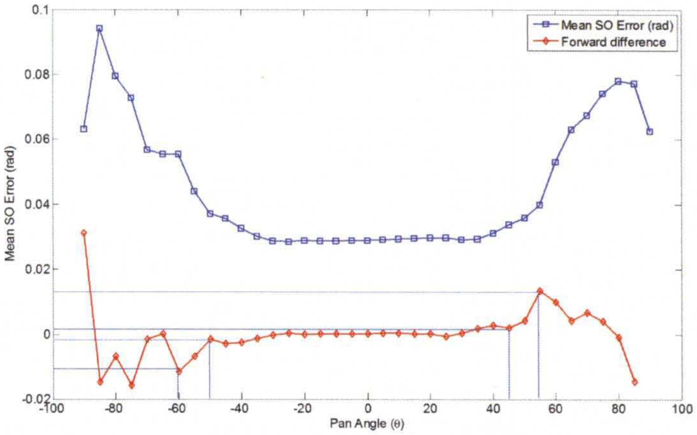


FIGURE 74 – Plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of ($-90°$ to $90°$). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change ($5°$). At pan angle $-60°$ and $55°$, there is a change of about $-0.01rad$ and $0.01rad$ error, respectively. The plot indicates that the proposed approach is insensitive to pose changes only up to about $\pm 50°$.

CHAPTER IV

MODEL-BASED SHAPE RECOVERY FROM SINGLE IMAGES OF UNKNOWN POSE USING A SMALL NUMBER OF FEATURE POINTS

This chapter proposes a model-based approach for face reconstruction using a small set of feature points from an input image of unknown pose. The model-based approaches proposed in the previous chapter require both texture (shading) and 2D shape information from the input image in order to perform 3D facial shape recovery. However, the methods discussed here need only the 2D feature points in an image to reconstruct the 3D shape.

Figure 75 illustrates this problem succinctly. The input is a 2D image with annotated feature points. Only the information from the feature points is given to a 3D estimation black box. The output can be one of the the two cases: (a) 3D sparse shape and (b) 3D dense shape.

A. Basic Definitions and Notations

The geometry of a face is represented as a shape vector that contains the XYZ coordinates of its vertices, i,e., $S = (X_1, Y_1, Z_1, \cdots, Y_n, Z_n)^T$, where n is the number of vertices. Similar to the previous chapter, the shape model can be constructed using a data set of m samples; a sample shape i is represented by the shape vector S_i. Novel shapes **s** can be generated from convex combinations of the m sample shapes, i.e., $\mathbf{s} = \sum_{i=1}^{m} a_i S_i$, $\sum_{i=1}^{m} a_i = 1$.

It is computationally convenient to reduce the dimensionality of the shape space, especially when dealing with high-dimensional shape vectors. Using Principal Component Analysis (PCA) on the data matrix provides us with $m-1$ eigenvectors $\mathbf{S}_i$, their corresponding eigenvalues (variances) σ_i^2, and the mean shape $\bar{\mathbf{s}}$.

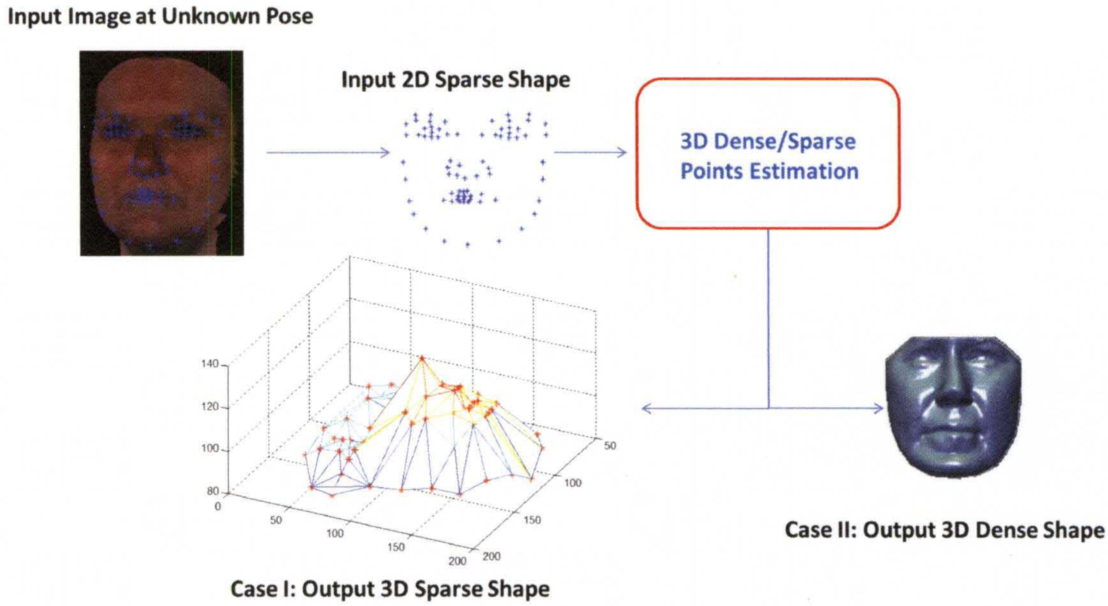

Input Image at Unknown Pose
Input 2D Sparse Shape
3D Dense/Sparse Points Estimation
Case I: Output 3D Sparse Shape
Case II: Output 3D Dense Shape

FIGURE 75 – The problem of estimating 3D shape given only the 2D feature points of an input image under unknown pose. The output 3D shape of the estimation method can be sparse or dense. Since prior shape and texture models from real data, which are metric in nature, are incorporated into the 3D shape recovery framework, the output 3D shape are metric, as well.

New shapes **s** can be derived from an equivalent model, i.e.,

$$\mathbf{s} = \bar{\mathbf{s}} + \sum_{i=1}^{m-1} a_i \mathbf{S}_i \tag{68}$$

where $\mathbf{a} = (a_1, \cdots, a_{m-1})^T$ is the shape parameter vector. In matrix notation, the above equation can be expressed as $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{Sa}$.

A realistic 2D face, $\mathbf{s}_{2D}$, can be generated from the 3D shape produced by the PCA model [70], i.e.,

$$\mathbf{s}_{2D} = fPR(\mathbf{s} + \mathbf{t}_{3D}) + \mathbf{t}_{2D} \tag{69}$$

where $\mathbf{t}_{3D}$ and $\mathbf{t}_{2D}$ are concatenated translation vectors of length $3N$ and $2N$, respectively. R is a $3N \times 3N$ block diagonal matrix which performs the combined rotation $R_\phi R_\gamma R_\theta$ for all N vertices. P is a $2N \times 3N$ orthographic projection matrix for the set of vertices. Note that only a subset of the original set of vertices will be visible. The z-buffer algorithm [59] is used for hidden-surface removal.

The same concepts above are expressed in a different terminology in [3]. 2D shape is represented as $\mathbf{x}_i \in \mathbb{R}^3$, and the corresponding 3D shape as $\mathbf{X}_i \in \mathbb{R}^4$, using homogeneous coordinates. The goal is to find the (3×4) camera projection matrix $\mathbf{C} \in \mathbb{R}^{3\times4}$, i.e., $\mathbf{x}_i = \mathbf{C}\mathbf{X}_i$. Assuming an affine camera, the camera projection matrix **C** can be solved using the Gold Standard Algorithm (Algorithm 7.2) in [3].

1. Experimental Results

The goal here is to visualize concepts related to the previous section. Throughout the experiments, there is always a set of point correspondences, $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D points $\mathbf{x}_i$. Two cases will be considered here, namely: (a) $\mathbf{x}_i$ was synthetically generated from $\mathbf{X}_i$ through an unknown projection matrix **C** (**Case I**) and (b) $\mathbf{x}_i$ does not come from $\mathbf{X}_i$ (**Case II**), i.e., a different $\mathbf{X}'_i$ was projected using an unknown **C** to get $\mathbf{x}_i$. In both cases, the camera projection matrix is computed and is used to project $\mathbf{X}_i$ to the 2D space.

Figures 76 and 77 illustrate the first and second case, respectively. Notice that after projection of the 3D shape $\mathbf{X}_i$ using the projection matrix **C**, the resulting 2D shape $\tilde{\mathbf{x}}_i = \mathbf{C}\mathbf{X}_i$ fits perfectly to $\mathbf{x}_i$. This is not the scenario with the second case since $\mathbf{x}_i$ and $\mathbf{X}_i$ do not come from the same subject. The second scenario will be exploited in later algorithms in this chapter.

B. Related Methods

Previous methods that perform face reconstruction from a small number of feature points can be classified into two groups, namely: (a) iterative and (b) linear approaches (non-iterative). The latest iterative method is found in the work of [5] and [71]. The linear approach of [4] calculates 3D shape using only a series of matrix operations. This section will briefly summarize both iterative and non-iterative methods to gain insight at how 3D face recovery is done.

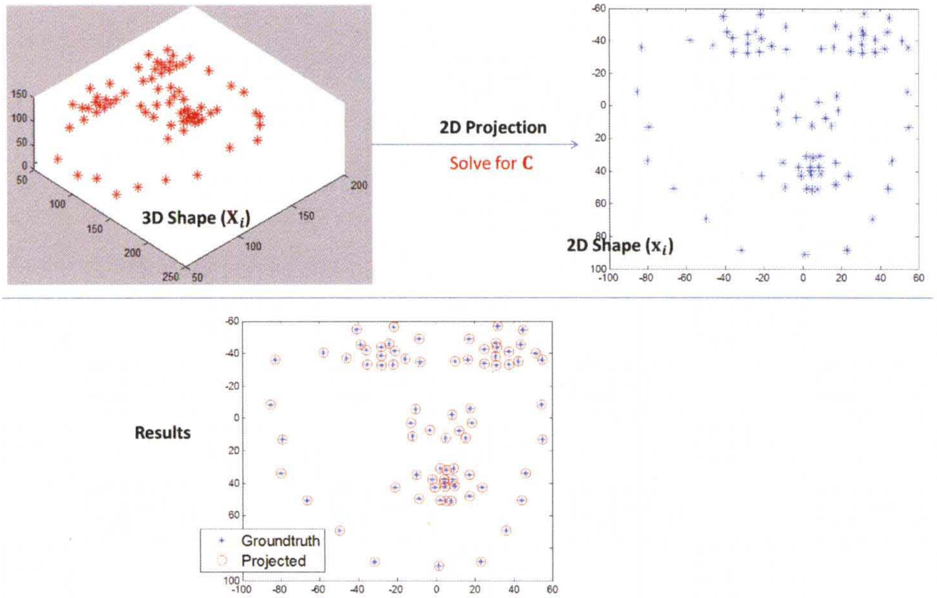


FIGURE 76 – (**Upper**) Solve for the camera projection matrix $\mathbf{C}$ given the point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$. Note that $\mathbf{x}_i$ was synthetically generated from $\mathbf{X}_i$. (**Lower**) After projecting $\mathbf{X}_i$ to the 2D space using the computed projection matrix $\mathbf{C}$, the resulting shape fits perfectly to the input 2D feature points.

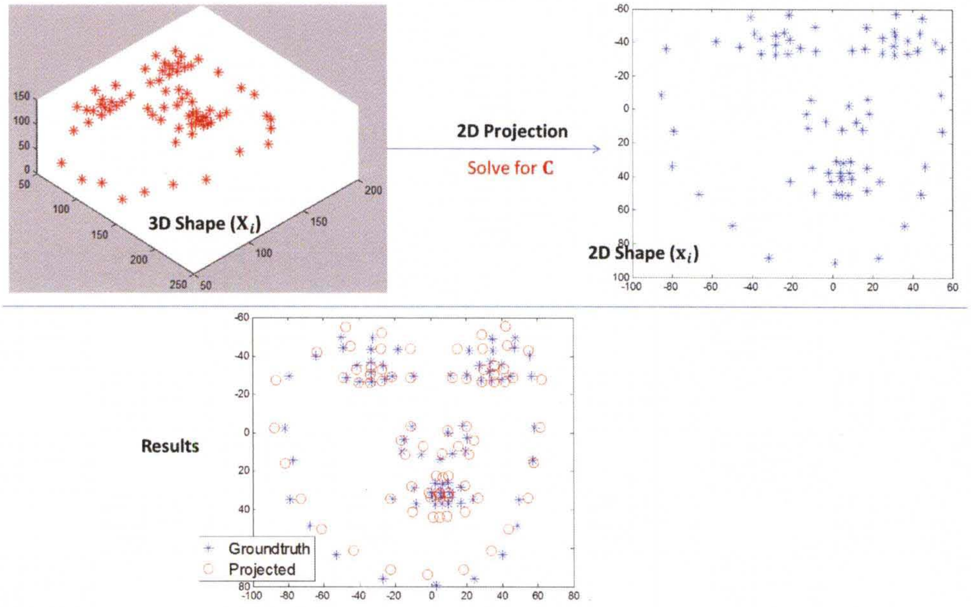


FIGURE 77 – (**Upper**) Solve for the camera projection matrix $\mathbf{C}$ given the point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$. Note that $\mathbf{x}_i$ was *not* synthetically generated from $\mathbf{X}_i$. (**Lower**) After projecting $\mathbf{X}_i$ to the 2D space using the computed projection matrix $\mathbf{C}$, the resulting shape does not fit perfectly to the input 2D feature points. However, they are rigidly registered together.

1. Iterative Methods

The algorithms in [5] and [71] use the 3D-to-2D projection equation of (69). They replace the 3D shape $\mathbf{s}$ with the shape model from PCA ($\mathbf{s} = \bar{\mathbf{s}} + \mathbf{Sa}$) to allow minimization with respect to the shape parameter $\mathbf{a}$, i.e.,

$$\mathbf{s}_{2D} = fPR(\bar{\mathbf{s}} + \mathbf{Sa} + \mathbf{t}_{3D}) + \mathbf{t}_{2D} \tag{70}$$

where f is a scale parameter, P is an orthographic projection matrix, R is the rotation matrix, $\mathbf{t}_{2D}$ and $\mathbf{t}_{3D}$ are translation vectors in 2D and 3D, respectively.

Notice that if the rendering process is inverted, the shape parameters $\mathbf{a}$ can be recovered from the shape error. As long as f, P, and R are kept constant, the relation between the shape $\mathbf{s}_{2D}$ and $\mathbf{a}$ is linear, i.e.,

$$\frac{\partial \mathbf{s}_{2D}}{\partial \mathbf{a}} = fPR\mathbf{S} \tag{71}$$

The above equation comes from differentiating $\mathbf{s}_{2D}$ with respect to $\mathbf{a}$ in (70). Transferring $\partial\mathbf{a}$ to the other side of the equation, we have a linear system, $\partial\mathbf{s}_{2D} = fPR\mathbf{S}(\partial\mathbf{a})$. Therefore, given the shape error $\partial\mathbf{s}_{2D}$, estimated by the displacement of a set of feature points, the update of $\mathbf{a}$ can be determined. The shape reconstruction goes through the following steps iteratively:

Model Initialization: Initialize the shape parameter to $\mathbf{a} = \mathbf{0}$. The pose parameters, f, R, and t_{2D}, are also manually set.

Feature Correspondence: Manually pick a set of feature points in the input image, denoted as $\mathbf{s}_{2D}^{im}$. There should be a corresponding set of these feature points in the shape model, denoted as $\mathbf{s}_{3D}^{mod}$. Projecting $\mathbf{s}_{3D}^{mod}$ to the 2D space will result into 2D points denoted as $\mathbf{s}_{2D}^{r}$.

Rotation, Translation, and Scale Parameters Update: Solve for parameters f, R, and t_{2D} by minimizing the following objective function using Levenberg-Marquadt optimization [3] [66].

$$\arg\min_{f,P,R} \|\mathbf{s}_{2D}^{im} - (fPR(\bar{\mathbf{s}} + \mathbf{Sa} + \mathbf{t}_{3D}) + \mathbf{t}_{2D})\|^2 = (\tilde{f}, \tilde{R}, \tilde{\mathbf{t}}_{2D}) \tag{72}$$

Shape Parameter Update: The shape error can be defined as the difference between $\mathbf{s}_{2D}^{im}$ and $\mathbf{s}_{2D}^{r} = \tilde{f}P\tilde{R}(\bar{\mathbf{s}} + \mathbf{Sa} + \mathbf{t}_{3D}) + \tilde{\mathbf{t}}_{2D}$. The vector of shape parameters can be updated with $\delta\mathbf{a}$ by solving the linear system of equations

$$\partial\mathbf{s}_{2D} = fPR\mathbf{S}(\partial\mathbf{a}) \tag{73}$$

2. Linear Methods

Recently, Aldrian et al. [4] proposed a shape recovery method that can extract 3D facial shape using only a sequence of matrix operations. This work will represent the class of linear approaches for this problem of 3D face reconstruction from a small set of feature points.

Before going to the actual algorithm, a brief summary of some changes in notation will be discussed. The 2D projection of the 3D feature points ($\mathbf{s}_{3D}^{mod}$) in the shape model is now referred to as $y_{mod2D,i}$, instead of $\mathbf{s}_{2D}^{r}$. The 2D feature points in the input image is now denoted as y_i, instead of $\mathbf{s}_{2D}^{im}$.

Another set of important notations in [4] is related to the eigenvector matrix $\mathbf{S}$ after applying PCA on shape data and the camera projection matrix solved using the Gold Standard Algorithm in [3]. Let $\tilde{\mathbf{S}} \in \mathbb{R}^{3N \times m-1}$ be the matrix formed after subselecting the rows of the eigenvector matrix $\mathbf{S}$ associated with the N feature points. Since we are dealing with homogeneous coordinates, a row of zeros will be inserted every after third row of $\hat{\mathbf{S}}_h$, giving it a dimension of $(4N \times m - 1)$. For the camera projection matrix, a new block diagonal matrix $\mathbf{P} \in \mathbb{R}^{3N \times 4N}$, where the camera projection matrix $\mathbf{C}$ is placed on the diagonal.

Instead of minimizing with respect to the shape parameter $\mathbf{a}$ in (68), the method in [4] minimizes with respect to a related variable, namely, the variance normalized shape parameter $\mathbf{c}_s = (\frac{a_1}{\sigma_1}, \cdots, \frac{a_{m-1}}{\sigma_{m-1}})$, where σ_i^2 are the eigenvalues after PCA is performed on the shape data matrix.

Using the new set of notations, the 2D points ($y_{mod2D,i}$) obtained by projecting the 3D model points ($\mathbf{s}_{3D}^{mod}$), given by the shape parameter $\mathbf{a}$, to the 2D space is

$$y_{mod2D,i} = \mathbf{P}_i \cdot (\hat{\mathbf{S}}_h \mathbf{a} + \bar{\mathbf{s}}) \tag{74}$$

where $\mathbf{P}_i$ is the ith row of $\mathbf{P}$.

The next step is to discuss the error functional to be minimized in only a single step. This can be done by differentiating the functional, setting it to 0, and solving for $\mathbf{c}_s$. The error functional is

$$E = \sum_{i=1}^{3N} \frac{[y_{mod2D,i} - y_i]^2}{\sigma_{2D,i}^2} + ||\mathbf{c}_s||^2 \tag{75}$$

where $\sigma_{2D,i}^2$ is the 2D point error variance that explains the difference between the observed and modeled feature point positions in the input image. The value of $\sigma_{2D,i}^2$ is determined after performing some offline training.

Substituting the expanded form of $y_{mod2D,i}$ in (74) to (75) and applying the second binomial theorem [72] and expanding yields

$$E = \sum_{i=1}^{3N} \frac{[\mathbf{P}_i \cdot \hat{\mathbf{S}}_h \mathbf{a} + \mathbf{P}_i \cdot \bar{\mathbf{s}}]^2 - 2[\mathbf{P}_i \cdot \hat{\mathbf{S}}_h \mathbf{a} + \mathbf{P}_i \cdot \bar{\mathbf{s}}]y_i + y_i^2}{\sigma_{2D,i}^2} + ||\mathbf{c}_s||^2 \tag{76}$$

For notational convenience, let $\mathbf{R}_i = \mathbf{P}_i \hat{\mathbf{S}}_h$ and $k_i = 2\mathbf{P}_i \cdot \bar{\mathbf{s}}$. Expanding according to the first binomial theorem leads to

$$E = \sum_{i=1}^{3N} \frac{(\mathbf{R}_i \mathbf{a})^2 + k_n(\mathbf{R}_i \mathbf{a}) + (\mathbf{P}_i \cdot \bar{\mathbf{s}})^2 - 2y_i \mathbf{R}_i \mathbf{a} + k_i y_i + y_i^2}{\sigma_{2D,i}^2} + ||\mathbf{c}_s||^2 \tag{77}$$

The next procedure is to minimize the error E by differentiating it with respect to $\mathbf{a}$ and setting the result to 0, i.e.,

$$0 = \nabla E = \sum_{i=1}^{3N} \frac{2\mathbf{R}_i^T \mathbf{R}_i \mathbf{a} + k_i \mathbf{R}_i^T - 2y_i \mathbf{R}_i^T - 2y_i \mathbf{R}_i^T}{\sigma_{2D,i}^2} + 2\mathbf{c}_s \tag{78}$$

However, as stated earlier, the functional is to be minimized with respect to $\mathbf{c}_s$, not $\mathbf{a}$. Hence, the matrix $\mathbf{R}_i$ will be multiplied with the shape eigenvalues, i.e., $\mathbf{Q}_i = \mathbf{R}_i diag(\sigma_i^2)$, which leads to

$$\sum_{i=1}^{3N} \frac{2\mathbf{Q}_i^T \mathbf{Q}_i \mathbf{c}_s}{\sigma_{2D,i}^2} + 2\mathbf{c}_s = \sum_{i=1}^{3N} \frac{2y_i \mathbf{Q}_i^T}{\sigma_{2D,i}^2} - \sum_{i=1}^{3N} \frac{k_i \mathbf{Q}_i^T}{\sigma_{2D,i}^2} \tag{79}$$

For clarity, let

$$\mathbf{T}_1 = \sum_{i=1}^{3N} \frac{2\mathbf{Q}_i^T \mathbf{Q}_i \mathbf{c}_s}{\sigma_{2D,i}^2}, \quad \mathbf{T}_2 = \sum_{i=1}^{3N} \frac{2y_i \mathbf{Q}_i^T}{\sigma_{2D,i}^2} - \sum_{i=1}^{3N} \frac{k_i \mathbf{Q}_i^T}{\sigma_{2D,i}^2} \tag{80}$$

and get the simplified version of (79), $\mathbf{T}_1 + 2\mathbf{c}_s = \mathbf{T}_2$. This equation can be solved by applying a *Cholesky Decomposition* [73] to $\mathbf{T}_1$ and further decomposing the result with *Singular Value Decomposition*, i.e.,

$$\begin{aligned}
\mathbf{M}^T\mathbf{M}\mathbf{c}_s + 2\mathbf{c}_s &= \mathbf{T}_2, \text{ where } \mathbf{T}_1 = \mathbf{M}^T\mathbf{M} \\
\mathbf{V}\mathbf{W}^2\mathbf{V}^T\mathbf{c}_s + 2\mathbf{c}_s &= \mathbf{T}_2, \text{ where } \mathbf{M} = \mathbf{U}\mathbf{W}\mathbf{V}^T \\
diag(w_i + 2)\mathbf{V}^T\mathbf{c}_s &= \mathbf{V}^T\mathbf{T}_2 \\
\mathbf{c}_s &= [diag(w_i + 2)\mathbf{V}^T]^{-1}\mathbf{V}^T\mathbf{T}_2 \qquad (81)
\end{aligned}$$

Therefore, using only a sequence of matrix operations, the normalized shape parameters ($\mathbf{c}_s$) can be computed given the location of the 2D feature points from the input image, as well as the camera projection matrix $\mathbf{C}$. It is straightforward to compute the actual shape parameter $\mathbf{a}$, i.e., $a_i = c_{s,i}\sigma_i$.

C. Proposed Approach

This section presents the proposed method to solve the problem of extracting 3D information from single images of unknown pose using only a small number of feature points (Figure 75). Figure 75 shows two output cases, namely: (a) 3D sparse shape and (b) 3D dense shape. The discussion in this section will focus first on the simpler case of 3D sparse shape and move on to 3D dense shape, as an extension.

1. Preliminaries

As previously discussed that the USF database used in this work contains both albedo and dense shape (Figure 78 (**Upper Left**)). Both albedo and dense shape are expressed as Monge patches, i.e., $(x, y, Z(x, y), T(x, y))$. Also, the image data of the USF database samples are manually annotated with 76 points, related to the important features of the face, as shown in Figure 37. Since both image and dense shape data are in correspondence with each other, the annotation points can also be applied to the height maps, which results into 3D sparse shapes, as illustrated in (Figure 78 (**Upper Right**)). Since there are multiple USF subjects, there is a series of dense shapes together with corresponding sparse shapes, as illustrated in (Figure 78 (**Lower**)). This series of dense and sparse shapes is integral to the proposed method in this chapter.

2. Model-based Framework (Case I: Output 3D Sparse Shape)

Suppose we have an input 2D sparse shape and the goal is to find the camera projection matrix **C** from its unknown (and yet to be solved) actual 3D sparse shape. A good substitute for this unknown 3D shape is the mean shape. A camera projection matrix can be computed between the mean 3D sparse shape and the input 2D sparse shape (Figure 79(**Left**)). Further, this projection matrix can be used to project a sample USF 3D sparse shape to the 2D space. This is illustrated in Figure 79(**Right**), where the projections of the mean shape, sample USF 3D sparse shape, and the input image feature points are plotted together. The projection matrix **C** can be used to project all USF database samples to the 2D space, as illustrated in Figure 80. Notice that they are rigidly registered together about the input 2D shape.

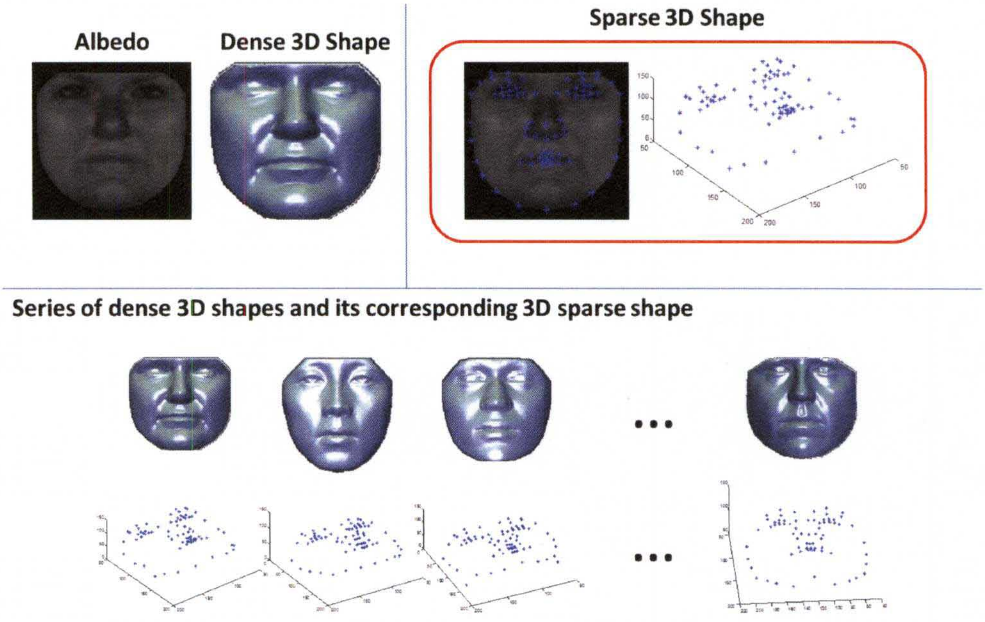

Albedo
Dense 3D Shape
Sparse 3D Shape
Series of dense 3D shapes and its corresponding 3D sparse shape
...
...

FIGURE 78 – (**Upper Left**) Albedo (texture) and 3D dense shape of a USF database sample, expressed as Monge patches, i.e., $(x, y, Z(x, y), T(x, y))$. (**Upper Right**) 3D sparse shape corresponding to the annotated positions of the albedo image. (**Lower**) A series of 3D dense shape with its corresponding 3D sparse shapes.

The next step is to build two models related to the 3D USF sparse shapes and the corresponding projected 2D shapes, i.e., $\mathbf{s}_{3D} = \bar{\mathbf{s}}_{3D} + \mathbf{P}_{s_{3D}}\mathbf{b}_{s_{3D}}$ and $\mathbf{s}_{2D} = \bar{\mathbf{s}}_{2D} + \mathbf{P}_{s_{2D}}\mathbf{b}_{s_{2D}}$.

Notice that Figure 80 is an example of a coupled model, similar to that of [1] and Fig. 43. The diagram in Figure 80 can be cast as a regression framework, where the independent data are the 2D shapes ($\mathbf{x}_i$) and the dependent data are the 3D shapes ($\mathbf{X}_i$).

Similar to the combined models in the previous chapter, Principal Component Regression (PCR) is used to model the relationship between the dependent and independent data. The basic idea is to decompose both 2D and 3D shapes into a low-dimensional subspace, i.e., replace $\mathbf{x}_i$ and $\mathbf{X}_i$ by their respective PCA coefficients $\mathbf{b}_{s_{2D},i}$ and $\mathbf{b}_{s3D,i}$. Standard multivariate linear regression (MLR) is then performed between the low-dimensional representations of $\mathbf{x}_i$ and $\mathbf{X}_i$. For reference, Figure 44 explains the difference between MLR and PCR.

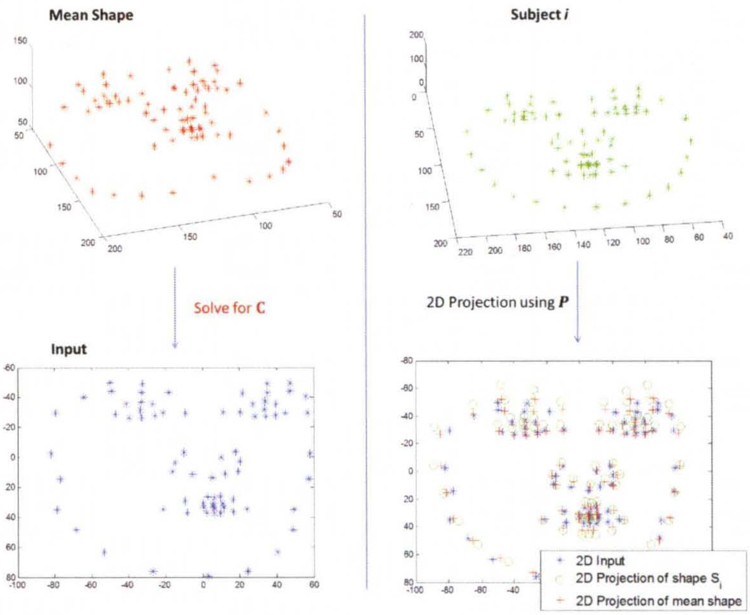


FIGURE 79 – (**Left**) Compute the camera projection matrix **C** between the 3D sparse mean shape and the input 2D shape ($\mathbf{x}_{inp}$). The mean shape substitutes for the unknown 3D sparse shape where $\mathbf{x}_{inp}$ comes from. (**Right**) Using the computed matrix **C**, the 3D mean shape and a sample USF subject are projected to the 2D space. The results are plotted together with the input 2D feature points.

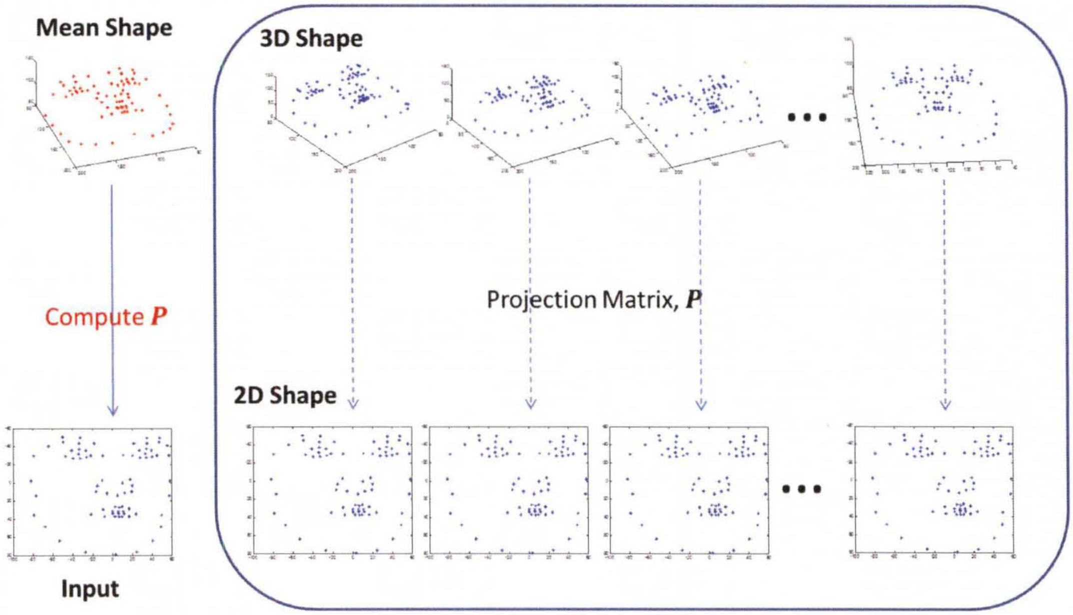


FIGURE 80 – Given the input 2D feature points ($\mathbf{x}_{inp}$), a camera projection matrix **P** can be estimated using the 3D mean sparse shape. This camera projection matrix can be used to project all USF database samples to the 2D space.

Let $\mathbf{T} = [\mathbf{b}_{s_{2D},1}, \cdots, \mathbf{b}_{s_{2D},m-1}]$ and $\mathbf{U} = [\mathbf{b}_{s_{3D},1}, \cdots, \mathbf{b}_{s_{3D},m-1}]$ be the low-dimensional representations of $\mathbf{x}_i$ and $\mathbf{X}_i$, respectively. Performing MLR yields

$$\mathbf{U} = \mathbf{T}\mathbf{C}_R + \mathbf{F} \tag{82}$$

where $\mathbf{C}_R$ is the matrix of regression coefficients and F is the matrix of random noise errors. The least squares method then provides

$$\tilde{\mathbf{C}}_R = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U} \tag{83}$$

There are two remaining steps before the 3D sparse shape can be recovered. The shape coefficient of the 2D input feature points need to be solved, i.e., $\mathbf{b}_{s_{2D},inp} = P_{s_{2D}}^T(\mathbf{x}_{inp} - \bar{\mathbf{s}}_{2D})$. Using the PCR model above, the 3D sparse shape coefficient can be inferred with the following equation, $\tilde{\mathbf{b}}_{s_{3D}} = (\mathbf{b}_{s_{2D},inp}\tilde{\mathbf{C}}_R)^T$. The solved shape coefficient $\tilde{\mathbf{b}}_{s_{3D}}$ can be substituted to the 3D shape model, i.e., $\mathbf{x}_r = \bar{\mathbf{s}}_{3D} + \mathbf{P}_{s_{3D}}\tilde{\mathbf{b}}_{s_{3D}}$, to get the desired output. The algorithm listing below summarizes these steps.

3. Experimental Results

This section will compare the proposed approach above to recent methods that deal with the same problem mentioned in Figure 75, namely, the iterative approach of [5] and the linear (non-iterative) contribution of [4].

The face models are built using the USF 3D Face Database [67], which contains 100 subjects of diverse gender and ethnicity. Out of these 100 samples, 80 subjects were deemed to be acceptable and were subsequently chosen to build the shape and texture models. To perform comparisons, the 3D sparse shape for 80 out-of-training USF samples are recovered. The input images are generated with a random pose, with pan angle range of ($-20°$ to $20°$), as illustrated in Figure 81.

The 2D feature points in both training and test images are derived from the control points mentioned in Figure 37, which are annotated by a single person, with over five years of experience in annotating faces at the time of writing of this

Algorithm 7 Principal Component Regression (PCR) Framework for 3D Sparse Shape Recovery

```
INPUT: (a) Input image feature points, $\mathbf{x}_{inp}$ (b) USF sparse shape samples: $(\mathbf{X}_1, \cdots, \mathbf{X}_n)$ (c) Sparse mean shape, $\mathbf{X}_m$
OUTPUT: (a) Recovered 3D sparse shape, $\mathbf{x}_r$
Solve for the camera projection matrix: Determine $\mathbf{C}$ such that $\mathbf{x}_{inp} = \mathbf{C}\mathbf{X}_m$.
Project all 3D sparse shapes to the 2D space using the computed projection matrix: Solve for $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$, such that $\mathbf{x}_i = \mathbf{C}\mathbf{X}_i$
Build the 3D sparse shape model from the USF samples using PCA: Construct $\mathbf{s}_{3D} = \bar{\mathbf{s}}_{3D} + \mathbf{P}_{s_{3D}}\mathbf{b}_{s_{3D}}$.
Build the 2D sparse shape model from the projected 2D USF samples $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$: Construct $\mathbf{s}_{2D} = \bar{\mathbf{s}}_{2D} + \mathbf{P}_{s_{32}}\mathbf{b}_{s_{2D}}$.
Replace the 3D shape samples $(\mathbf{X}_1, \cdots, \mathbf{X}_n)$ with its coefficients: Solve for $\mathbf{b}_{s_{3D},i} = P_{s_{3D}}^T(\mathbf{X}_i - \bar{\mathbf{s}}_{3D})$
Replace the projected 2D shape samples $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ with its coefficients: Solve for $\mathbf{b}_{s_{2D},i} = P_{s_{2D}}^T(\mathbf{x}_i - \bar{\mathbf{s}}_{2D})$
Setup matrices for Principal Component Regression (PCR): Let $\mathbf{T} = [\mathbf{b}_{s_{2D},1}, \cdots, \mathbf{b}_{s_{2D},m-1}]$, and $\mathbf{U} = [\mathbf{b}_{s_{3D},1}, \cdots, \mathbf{b}_{s_{3D},m-1}]$
Build the PCR model: Construct $\tilde{\mathbf{C}}_R = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U}$
Solve for the shape coefficients of the 2D input feature points $(\mathbf{x}_{inp})$: Solve for $\mathbf{b}_{s_{2D},inp} = P_{s_{2D}}^T(\mathbf{x}_{inp} - \bar{\mathbf{s}}_{2D})$
Solve for the shape coefficients: Get $\tilde{\mathbf{b}}_{s_{3D}} = \mathbf{b}_{s_{2D},inp}\tilde{\mathbf{C}}_R$
Solve for the recovered 3D sparse shape: $\mathbf{x}_r = \bar{\mathbf{s}}_{3D} + \mathbf{P}_{s_{3D}}\tilde{\mathbf{b}}_{s_{3D}}$
```

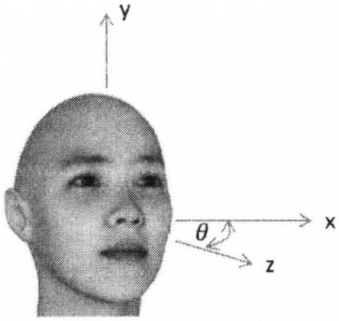


FIGURE 81 – Pan angle illustration. The face moves left-to-right or right-to-left sideways.

document. These control points correspond to 3D feature points in a USF subject. Given a random pose angle, the 2D feature points are the projection to the image plane of the 3D feature points.

Figure 82 shows the recovered 3D sparse shape of the same input using the three algorithms, together with the ground truth. Figure 82 (**View 2**) represents the projection of the recovered 3D shapes to the *x*-*y* plane. Notice that the results of the proposed method and that of Aldrian et al. [4] are similar.

The next point of comparison will be the timing results of the three algorithms under the same computational conditions. Fig. 83 presents the side-by-side stem plot of the time (in seconds) needed to recover the 3D sparse shape for inputs generated from 80 samples of the USF database. Notice that the proposed method is computationally faster than the others, due to its simplistic regression framework. It is worth mentioning also that the method of [4] requires significant time in offline training for the 2D error variance $\sigma^2_{2D,i}$ in (75). This offline training time is not included in Fig. 83.

The 2D shape projection error is another way to compare the proposed method with respect to the state-of-the-art. Let $\mathbf{s}^{im}_{2D}$ be the feature points annotated on the input image, $\mathbf{S}^{r}_{3D}$ be the recovered 3D shape, and $\mathbf{s}^{r}_{2D}$ be the projected 2D shape after applying the computed camera matrix $\mathbf{C}$. The 2D shape projection error is defined as

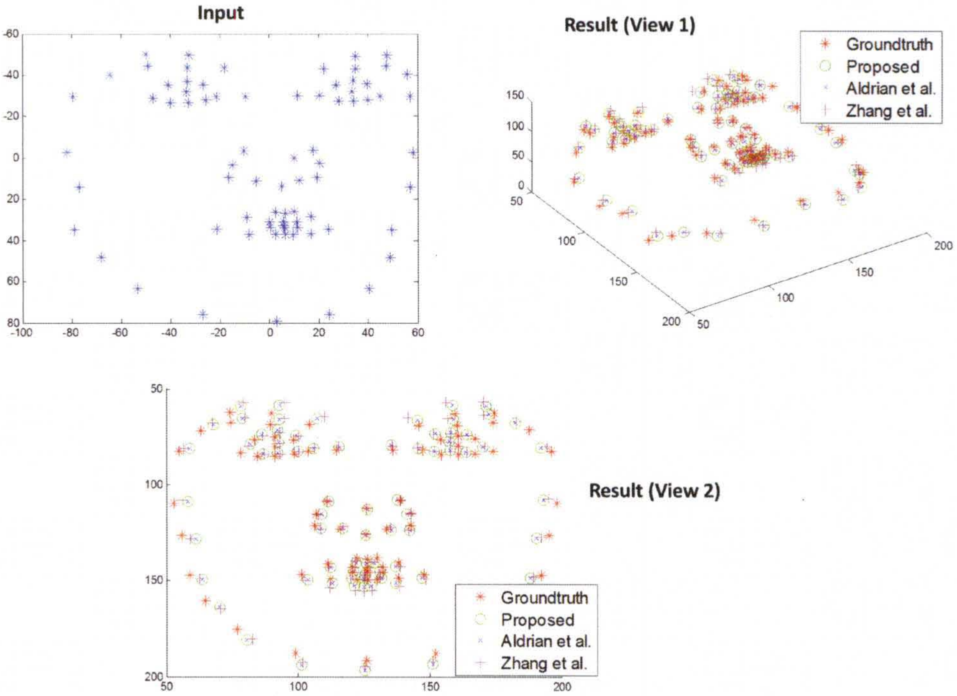


FIGURE 82 – Recovered 3D sparse shape of the same input using the three algorithms (proposed, [4], and [5]), together with the ground truth 3D sparse shape. (View 2) represents the projection of the recovered 3D shapes to the x-y plane. Notice that the results of the proposed method and that of Aldrian et al. [4] are similar.

$$s_{2D}^{err} = \|\mathbf{s}_{2D}^{im} - \mathbf{s}_{2D}^{r}\| \tag{84}$$

Figure 84 presents the side-by-side stem plot of the 2D shape projection error (84) for inputs generated from 80 samples of the USF database. Notice that both the proposed method and that of Aldrian et al. [4] perform better than Zhang et al. [5].

The ultimate goal in the previously discussed methods is to recover 3D shape given only 2D input points. The last point of comparison for this section is the 3D shape error. Let $\mathbf{S}_{3D}^{r}$ be the recovered 3D shape and $\mathbf{S}_{3D}^{gnd}$ be the ground truth shape. The 3D shape error is simply the norm of the difference between the recovered and true shapes, i.e.,

$$s_{2D}^{err} = \|\mathbf{S}_{3D}^{r} - \mathbf{S}_{3D}^{gnd}\| \tag{85}$$

Fig. 85 presents the side-by-side stem plot of the 3D shape error (85) for inputs generated from 80 samples of the USF database. Similar to the timing results and 2D shape projection error, both the proposed method and that of Aldrian et al. [4] outperform than Zhang et al. [5].

There are five conclusions that can be drawn from these experiments, the proposed: (a) is competitive due to its linear and non-iterative nature, (b) does not need explicit training, as opposed to [4], (c) has comparable results to [4], at a shorter computational time, (d) better in all aspects than Zhang and Samaras [5], and (e) has the limitation, together with [4] and [5], in terms of the need to manually annotate the input 2D feature points.

4. Model-based Framework (Case II: Output 3D Dense Shape)

To extend the proposed 3D face recovery framework here, from sparse shapes to dense shapes as output, the 3D sparse shape model is simply replaced with its dense model equivalent. Note that the 3D shape samples are a subset of the 3D dense samples, i.e., the USF dense sample represent the global shape of the face, while the sparse shape indicates the locations of the important feature points in 3D. The 3D sparse shapes are still retained in the framework because they will be projected to 2D space using the computed camera projection matrix **C**. Figure 86 illustrates these ideas. The modified algorithm pseudocode for dense shapes is listed below. Notice the similarity to its sparse predecessor, i.e., there are lines where sparse variables are simply replaced with dense versions.

5. Experimental Results

This section shows experiments to evaluate the performance of the proposed method in recovering 3D dense facial shapes. The face models are built using the USF 3D Face Database [67], which contains 100 subjects of diverse gender and ethnicity. Out of these 100 samples, 80 subjects were deemed to be acceptable and were subsequently chosen to build the shape and texture models.

The 2D feature points in both training and test images are derived from the control points mentioned in Figure 37, which are annotated by a single person, with over five years of experience in annotating faces at the time of writing of this document. These control points correspond to 3D feature points in a USF subject. Given a random pose angle, the 2D feature points are the projection to the image plane of the 3D feature points.

Experiment A To quantify the reconstruction accuracy, the author recovers the 3D shape for 80 out-of-training USF samples. The input images are generated with a random pan angle within the range of (-20° to 20°), as illustrated in Fig. 81. Two approaches to recover 3D shape from the input images will be used: (a) Case I - refers to 3D dense shape recovery approach using 2D feature points only and (b) Case II - refers to the 3D shape recovery method from the previous chapter that requires both shading and 2D shape in formation. For each input image, the two error measures, mean height error (62) and mean surface orientation error (63), are measured.

Figure 87 shows the recovered shapes together with the input and ground-truth shape. The results are very close visually. The next step is to show a side-by-side visualization of the mean height (Figure 88) and surface orientation (Figure 89) error stem plots, to compare against the proposed 3D shape recovery method using both texture and shape data from the last chapter. The actual mean and standard deviation of the mean height error and mean surface orientation error of the 80 out-of-training USF samples, for the above experiments, are shown in Tables 13 and 14. The numerical values indicate that even if texture information is not available, a decent reconstructed 3D shape is still possible.

Experiment B To investigate the sensitivity of the proposed approach with respect to pose, a similar experiment is performed as described, except that for a specific pose, the recovered 3D shapes of all 80 out-of-training input images are analyzed. Specifically, at pan angle of x°, generate 80 out-of-sample input images and compute the average mean height error and surface orientation error across all 80 input images, where $x \in \{-20^\circ, -15^\circ, \cdots, 15^\circ, 20^\circ\}$.

Figures 90 and 91 plot the average mean height error and average mean surface orientation error, respectively, with respect to a pan angle range of (-20° to 20°). The bar graphs indicate that the proposed method is not sensitive to pose changes.

TABLE 13

SPARSE-TO-DENSE EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN HEIGHT ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II
$\mu_{\bar{s}_{err}}$ (%)	2.51	2.34
$\sigma_{\bar{s}_{err}}$ (%)	0.77	0.73

It is interesting to note that in Figures 90 and 91, input images with close-to-frontal pose angles have slightly larger errors than inputs with non-frontal pose

TABLE 14
SPARSE-TO-DENSE EXPERIMENTS: MEAN AND STANDARD DEVIATION OF THE MEAN SURFACE ORIENTATION ERROR FOR 80 OUT-OF-TRAINING USF SAMPLES

	Case I	Case II
$\mu_{\bar{\theta}_{err}}$ (rad)	0.044	0.040
$\sigma_{\bar{\theta}_{err}}$ (rad)	0.010	0.010

angles. This phenomenon is not present in the proposed approach from the previous chapter (Figures 67 and 69), which requires both shading and 2D feature points information. The reason behind this is that at frontal pose (Figure 81), the x-y axis information playes a larger role in the location of the projected 2D feature points; z-axis information is virtually non-existent in close-to-frontal pose angles. Similarly, when there is significant pose (non-frontal) in the input image, all three axes (x, y, z) information contribute to the projected 2D feature points. Since the proposed approach is trying to reconstruct depth (z) from the 2D input image, it helps when the input is non-frontal.

Experiment C Similar to the previous chapter, the last experiment does not show at what point pose begins to have an effect. The next obvious step is to increase the pan angle range, i.e., $x \in \{-90°, -85°, \cdots, 85°, 90°\}$, and perform Experiment B. However, this experiment will take a considerable amount of time to perform.

A quick solution is to choose only one subject (sample index #1 in this experiment), instead of all 80 subjects, from the USF database and then perform Experiment B but with a pan angle range of $x \in \{-90°, -85°, \cdots, 85°, 90°\}$.

Figures 92 and 93 show both mean height error and surface orientation error across various pan angles. Figures 94 and 95 show the forward-difference numerical differentiations of the mean height error and surface orientation error plots to highlight the change in error per pan angle incremental change (5°). It appears that there is no breaking point of pose invariance, unlike Figures 71 and 72 from the previous chapter.

D. Conclusion

This chapter addressed the situation in which only 2D shape information is available from the input image. Recall that both texture (shading) and shape information were needed for the previous 3D face recovery methods (from the last chapter) to work. This chapter showed that both sparse and dense 3D shapes can be estimated from 2D shape information alone at an acceptable quality. The ideas in this chapter are published in [74].

The chapter started with the simpler case of output 3D sparse shapes. The proposed method was compared with state-of-the art algorithms, showing decent performance. There were five conclusions that drawn from the sparse experiments, namely, the proposed approach: (a) is competitive due to its linear and non-iterative nature, (b) does not need explicit training, as opposed to [4], (c) has comparable results to [4], at a shorter computational time, (d) better in all aspects than Zhang and Samaras [5], and (e) has the limitation, together with [4] and [5], in terms of the need to manually annotate the input 2D feature points.

The proposed method was then extended to output 3D dense shapes simply by replacing the sparse model with its dense equivalent, in the regression framework inside the 3D face recovery approach. The numerical values of the mean height and surface orientation error indicate that even if shading information is unavailable, a decent 3D dense reconstruction is still possible.

Interestingly, input images with close-to-frontal pose angles have slightly larger errors than inputs with non-frontal pose angles. The reason behind this observation is that at frontal pose (Figure 81), the x-y axis information contributes more to location of the projected 2D feature points. When there is significant pose (non-frontal) in the input image, all three axes (x, y, z) information influence the projected 2D feature points. Since the goal of the proposed approach is to reconstruct depth (z) from the 2D input image, it is beneficial if the input is non-frontal.

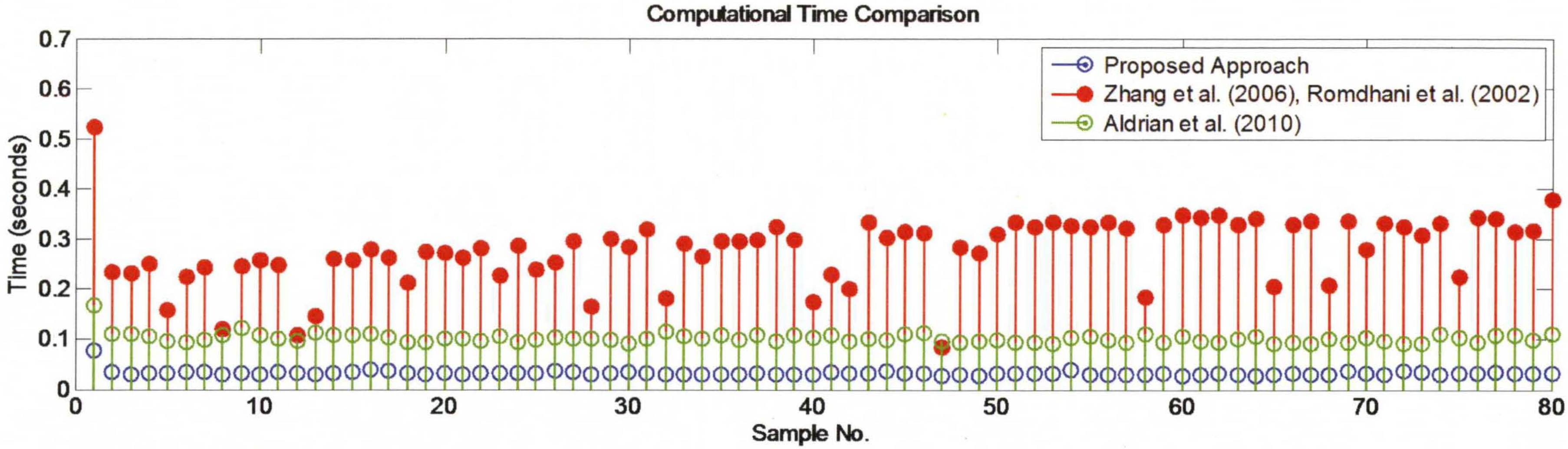


FIGURE 83 – Side-by-side stem plot of the time (in seconds) needed to recover the 3D sparse shape for inputs generated from 80 samples of the USF database. The x-axis refers to the sample index of the USF Database. Notice that the proposed method is computationally faster than the others, due to its simplistic regression framework. Average time is 0.03, 0.10, and 0.28 seconds for the proposed, Aldrian et al., and Zhang et al. approaches, respectively. Note that offline training time is not included in Aldrian et al.

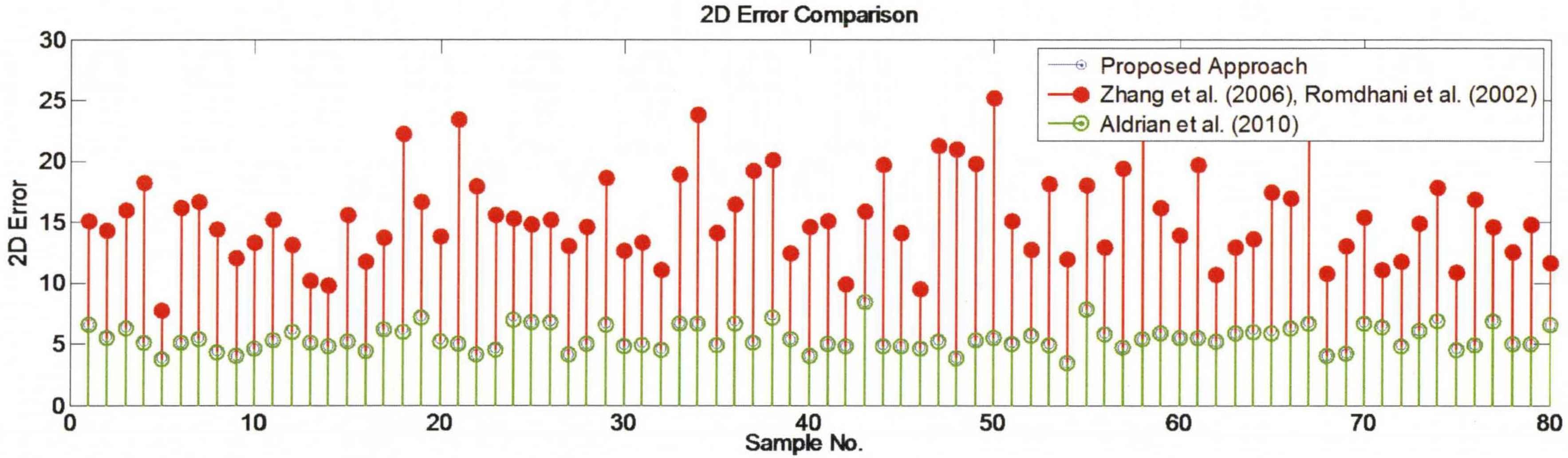


FIGURE 84 – Side-by-side stem plot of the 2D shape projection error (84) for inputs generated from 80 samples of the USF database. Average 2D shape projection error is 5.55, 5.48, and 15.43 seconds for the proposed, Aldrian et al., and Zhang et al. approaches, respectively.

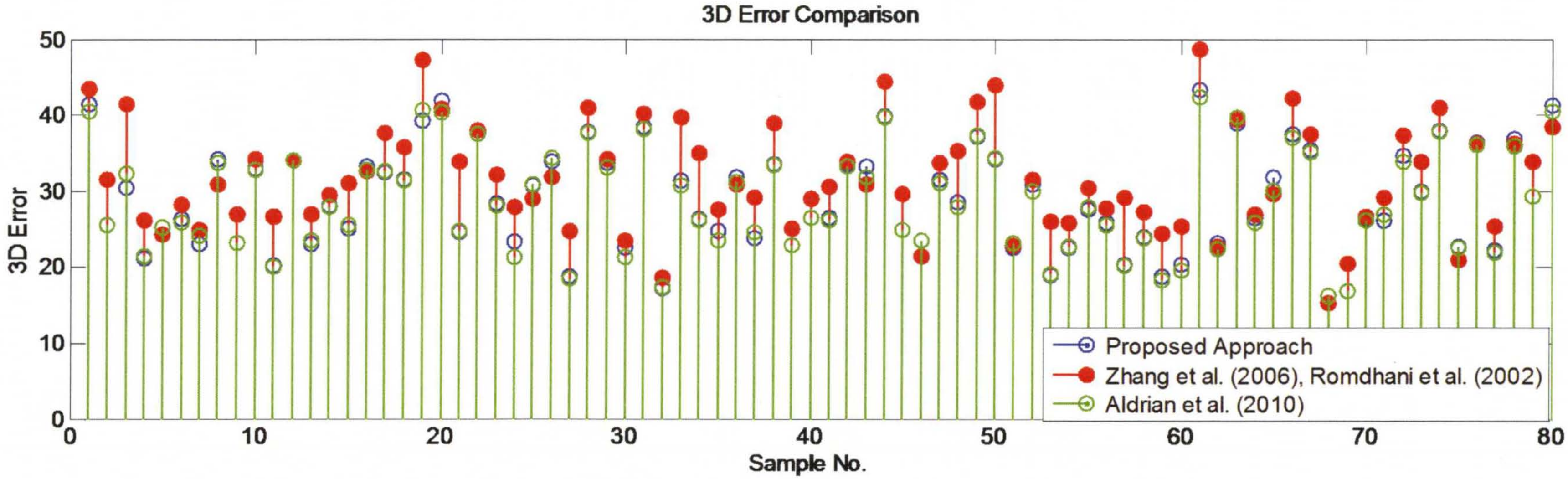


FIGURE 85 – Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. Average 3D shape error is 29.00, 28.79, and 31.80 seconds for the proposed, Aldrian et al., and Zhang et al. approaches, respectively.

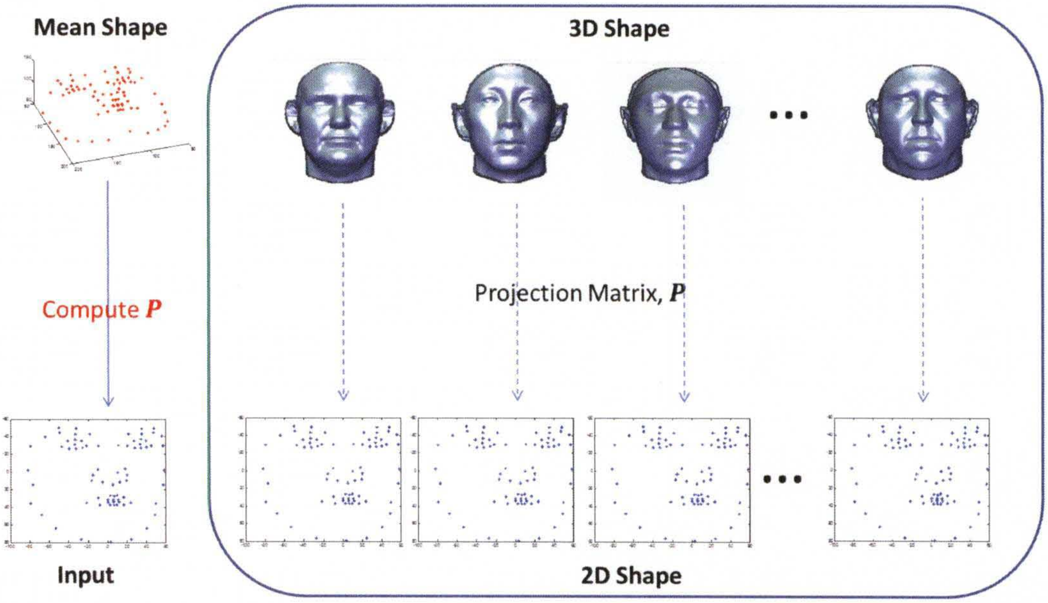


FIGURE 86 – Given the input 2D feature points ($\mathbf{x}_{inp}$), a camera projection matrix $\mathbf{P}$ can be estimated using the 3D mean sparse shape. This camera projection matrix can be used to project all USF database samples to the 2D space.

Algorithm 8 Principal Component Regression (PCR) Framework for 3D Dense Shape Recovery

```
INPUT: (a) Input image feature points, $\mathbf{x}_{inp}$ (b) USF dense $(\mathbf{X}_1^d, \cdots, \mathbf{X}_n^d)$ and sparse shape samples $(\mathbf{X}_1, \cdots, \mathbf{X}_n)$ (c) Sparse mean shape, $\mathbf{X}_m$
OUTPUT: (a) Recovered 3D dense shape, $\mathbf{x}_r^d$
Solve for the camera projection matrix: Determine $\mathbf{C}$ such that $\mathbf{x}_{inp} = \mathbf{C}\mathbf{X}_m$.
Project all 3D sparse shapes to the 2D space using the computed projection matrix: Solve for $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$, such that $\mathbf{x}_i = \mathbf{C}\mathbf{X}_i$
Build the 3D dense shape model from the USF samples using PCA: Construct $\mathbf{s}_{3D} = \bar{\mathbf{s}}_{3D} + \mathbf{P}_{s_{3D}}\mathbf{b}_{s_{3D}}$.
Build the 2D sparse shape model from the projected 2D USF samples $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$: Construct $\mathbf{s}_{2D} = \bar{\mathbf{s}}_{2D} + \mathbf{P}_{s_{32}}\mathbf{b}_{s_{2D}}$.
Replace the 3D dense shape samples $(\mathbf{X}_1, \cdots, \mathbf{X}_n)$ with its coefficients: Solve for $\mathbf{b}_{s_{3D},i} = P_{s_{3D}}^T(\mathbf{X}_i - \bar{\mathbf{s}}_{3D})$
Replace the projected 2D shape samples $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ with its coefficients: Solve for $\mathbf{b}_{s_{2D},i} = P_{s_{2D}}^T(\mathbf{x}_i - \bar{\mathbf{s}}_{2D})$
Setup matrices for Principal Component Regression (PCR): Let $\mathbf{T} = [\mathbf{b}_{s_{2D},1}, \cdots, \mathbf{b}_{s_{2D},m-1}]$, and $\mathbf{U} = [\mathbf{b}_{s_{3D},1}, \cdots, \mathbf{b}_{s_{3D},m-1}]$
Build the PCR model: Construct $\tilde{\mathbf{C}}_R = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U}$
Solve for the shape coefficients of the 2D input feature points ($\mathbf{x}_{inp}$): Solve for $\mathbf{b}_{s_{2D},inp} = P_{s_{2D}}^T(\mathbf{x}_{inp} - \bar{\mathbf{s}}_{2D})$
Solve for the shape coefficients: Get $\tilde{\mathbf{b}}_{s_{3D}} = \mathbf{b}_{s_{2D},inp}\tilde{\mathbf{C}}_R$
Solve for the recovered 3D dense shape: $\mathbf{x}_r^d = \bar{\mathbf{s}}_{3D} + \mathbf{P}_{s_{3D}}\tilde{\mathbf{b}}_{s_{3D}}$
```

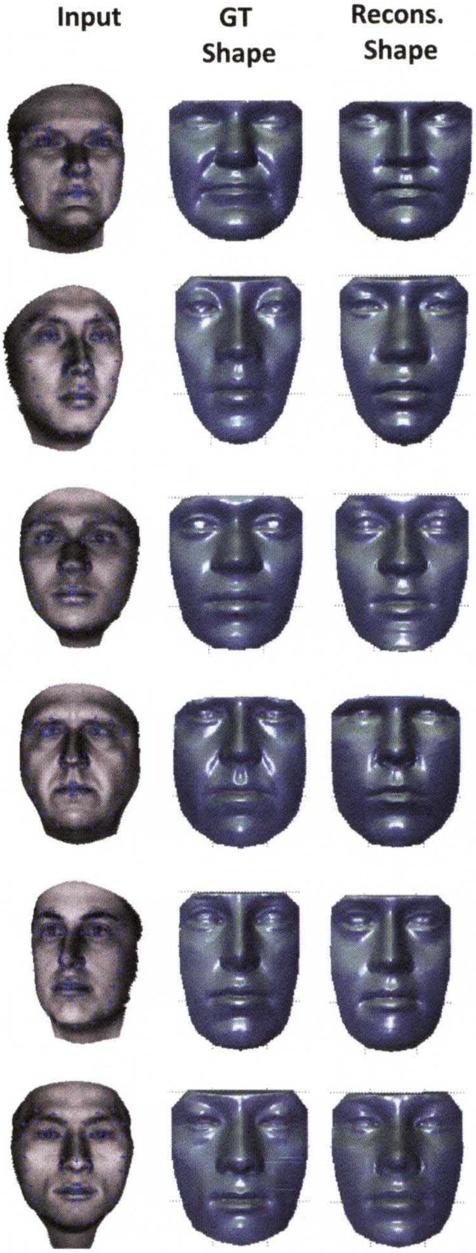


FIGURE 87 – Recovered shapes, together with the input image and ground-truth (GT) shape, for the model-based 3D shape recovery framework in Algorithm **8**.

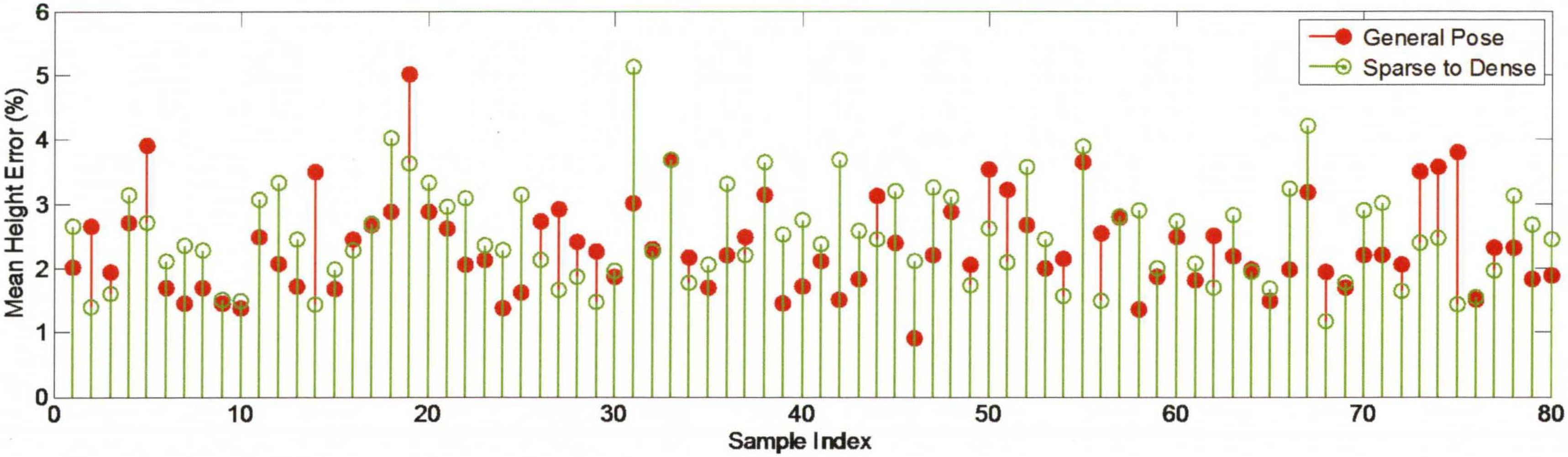


FIGURE 88 – Side-by-side visualization of the mean height error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are Sparse-to-Dense (Case I) and General Pose (Case II). Average error for all 80 out-of-training samples is 2.51% and 2.34% for the Case I and Case II experiments, respectively.

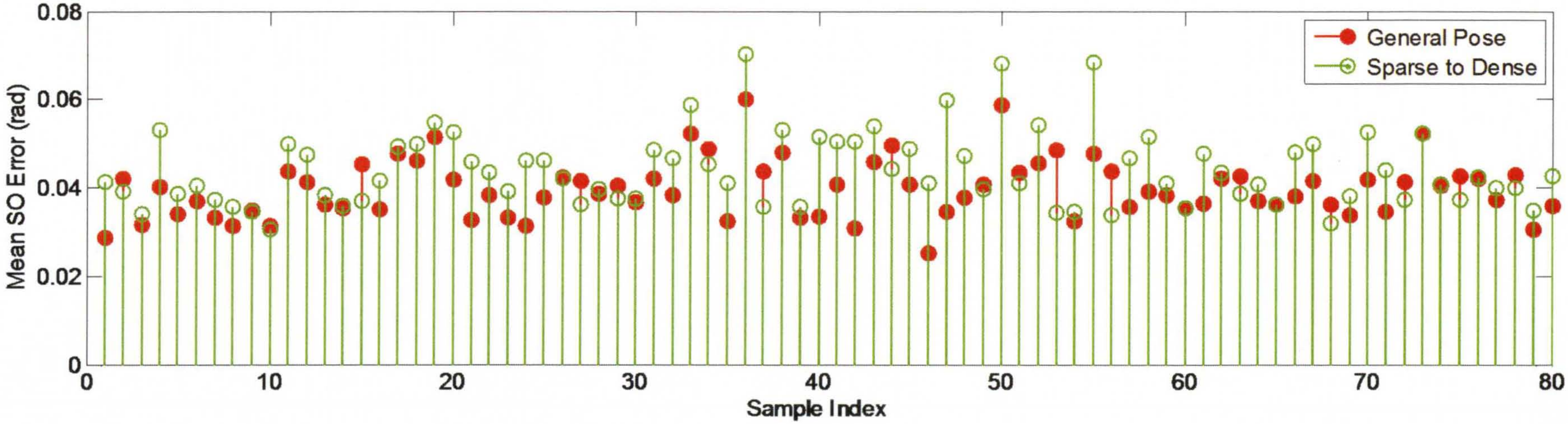


FIGURE 89 – Side-by-side visualization of the mean surface orientation error, in terms of a stem plot. The x-axis refers to the sample index and the y-axis is the actual error value. The plot legends are Sparse-to-Dense (Case I) and General Pose (Case II). Average error for all 80 out-of-training samples is 0.04 rad and 0.04 rad for the Case I and Case II experiments, respectively.

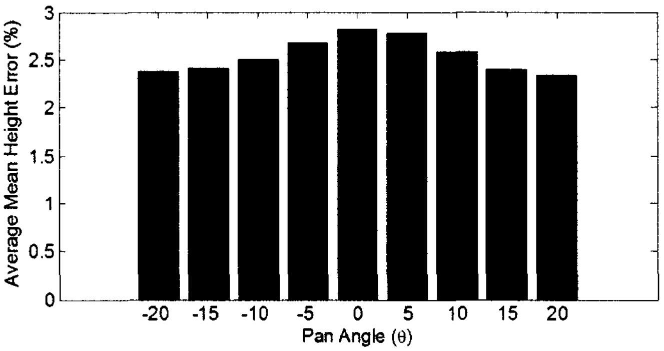


FIGURE 90 – 2D-Sparse to 3D-Dense Experiments. Bar graph of the average mean height error with respect to pose changes, i.e., pan angle range of (−20° to 20°). The graph indicates that the proposed approach is insensitive to pose changes.

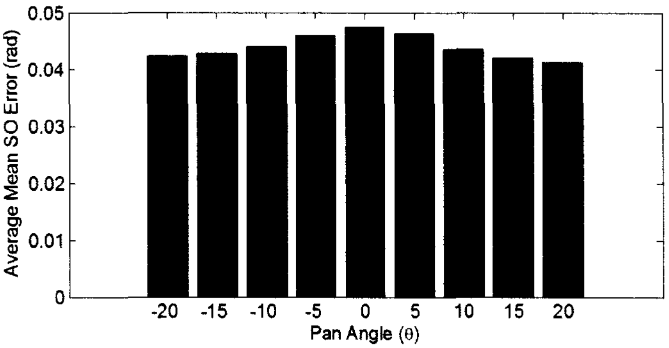


FIGURE 91 – 2D-Sparse to 3D-Dense Experiments. Bar graph of the average mean surface orientation error with respect to pose changes, i.e., pan angle range of (−20° to 20°). The graph indicates that the proposed approach is insensitive to pose changes.

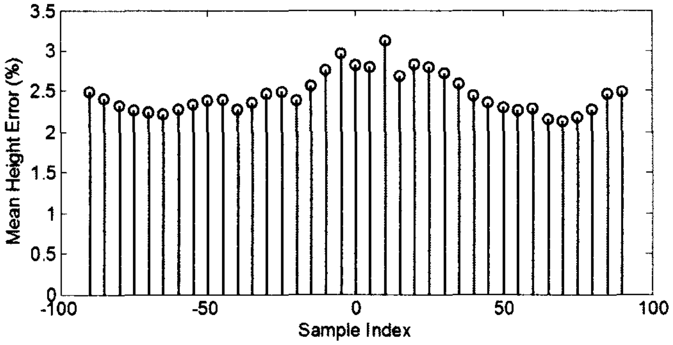


FIGURE 92 – 2D-Sparse to 3D-Dense Experiments. Stem plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (−90° to 90°). The plot indicates that the proposed approach is insensitive to pose changes, with frontal-pose images having slightly higher errors than non-frontal ones.

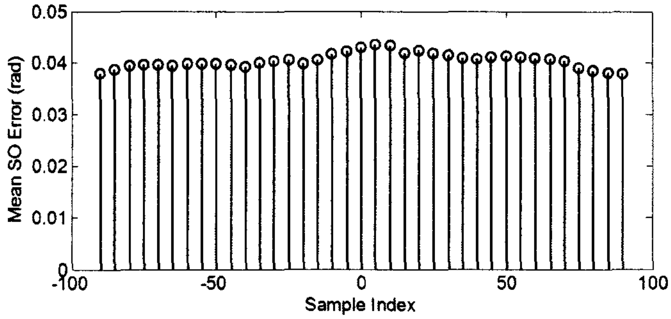


FIGURE 93 – 2D-Sparse to 3D-Dense Experiments. Stem plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (−90° to 90°). The plot indicates that the proposed approach is insensitive to pose changes, with frontal-pose images having slightly higher errors than non-frontal ones.

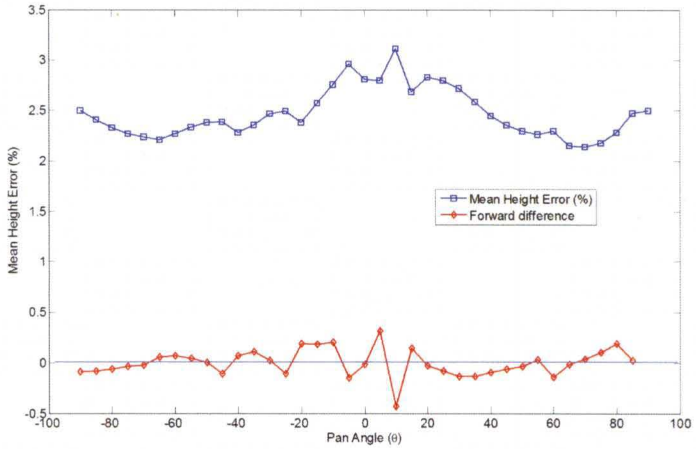


FIGURE 94 – Plot of the average mean height error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (−90° to 90°). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change (5°). It appears that there is no breaking point of pose invariance since the change in error per pan angle incremental change (5°) is close to 0% as the pan angle approaces ±90°.

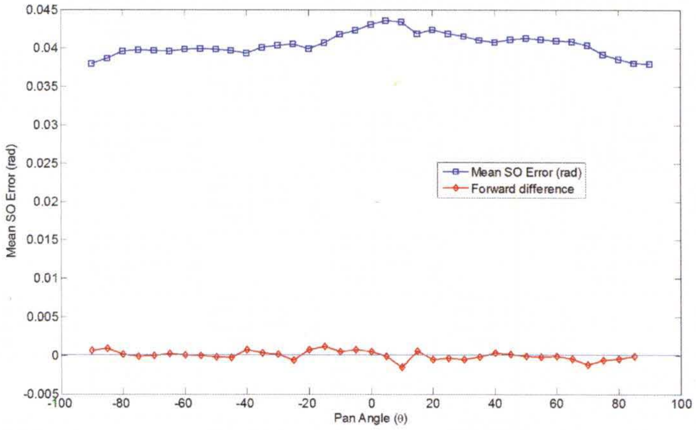


FIGURE 95 – Plot of the average mean surface orientation error of recovered shapes of input images coming from a single subject, with respect to pose changes, i.e., pan angle range of (−90° to 90°). Superimposed in this plot is the forward-difference numerical differentiation of the former to highlight the change in error per pan angle incremental change (5°). It appears that there is no breaking point of pose invariance since the change in error per pan angle incremental change (5°) is close to 0 rad as the pan angle approaces ±90°.

CHAPTER V
CONCLUSIONS AND FUTURE DIRECTIONS

The main purpose of this work is to extract 3D facial shape information from a single image of arbitrary and unknown pose and illumination. This dissertation starts first with the simple case of general illumination and fixed frontal pose. The classical shape-from-shading iterative equation is cast as a regression framework, which can then be solved efficiently using Principal Component Regression (PCR). Before coming up with the PCR-based method, there were two approaches considered, namely, an iterative approach and one based on combined models. The PCR-based algorithm was deemed to be the best after numerous tests and simulations.

General pose is added into the framework by incorporating multiple-view geometry concepts, specifically the computation of the camera projection matrix. The end result is a framework that can deal with both unknown pose and illumination.

The next major part of this work is the development of 3D facial shape recovery methods given only input 2D shape information, instead of both texture and shape. The proposed method can extract both 3D sparse and dense shapes, with acceptable results. The development of this approach was due to the gained insights from the previous proposed methods that deal with both shape and texture. Results show very acceptable performance, i.e., there is no huge difference between the recovered 3D shape from 2D shape information alone, compared to that, which was extracted using both shape and texture information.

A. Future Directions

There are several future directions identified for this dissertation:

- Use the recovered 3D shape (sparse or dense) from single 2D images to perform face recognition at-a-distance. Initial recognition results using sparse shape are already promising in [46] [75]. The USF database can be used to create the shape model. The stereo pairs at the 3-meter range can also be utilized for the shape model.
- Incorporate the framework in Chapter IV.D to act as a local update constraint for the Constrained Local Model (CLM) face alignment approach [76]. The current local update of the CLM approach works only with close-to-frontal face. Adding the proposed framework in this chapter will hopefully help it deal with non-frontal faces.
- The 2D input feature points are considered to be manually annotated in this dissertation. The next step is to make it automatic, using methods such as Active Shape Models (ASM) and Active Appearance Models (AAM). The automatic annotation algorithms should be modified to handle pose; current algorithms are sensitive to pose in input images. The work in Chapter IV.D may help in dealing with pose.
- Apply the proposed methods in this paper to objects (e.g., ears and teeth) other than its intended purpose, which are facial images.

REFERENCES

[1] M. Castelan, W. A. P. Smith, and E. R. Hancock. A coupled statistical model for face shape recovery from brightness images. *IEEE Transactions on Image Processing*, 16(4):1139–1151, 2007.

[2] Abdelrehim Ahmed and Aly Farag. A new statistical model combining shape and spherical harmonics illumination for face reconstruction. In *Proceedings of the 3rd international conference on Advances in visual computing - Volume Part I*, ISVC'07, pages 531–541, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[4] Oswald Aldrian and William Smith. A linear approach of 3d face shape and texture recovery using a 3d morphable model. In *Proceedings of the British Machine Vision Conference*, pages 75.1–75.10. BMVA Press, 2010. doi:10.5244/C.24.75.

[5] Lei Zhang and Dimitris Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):351–363, 2006.

[6] Jon Sullivan. Barns grand tetons mountains. `http://www.public-domain-image.com/nature-landscape/mountain/slides/barns-grand-tetons-mountains.html`.

[7] Gustave Caillebotte. Paris street in rainy weather. `http://en.wikipedia.org/wiki/Texture_gradient`.

[8] Fog landscape. `http://www.copyright-free-pictures.org.uk/landscapes/05-picture-fog.htm`.

[9] Johannes Vermeer. The lacemaker. `http://en.wikipedia.org/wiki/The_Lacemaker_(Vermeer)`.

[10] Abdelrehim H. Ahmed and Aly A. Farag. A new formulation for shape from shading for non-lambertian surfaces. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1817–1824, Washington, DC, USA, 2006. IEEE Computer Society.

[11] Abdelrehim H. Ahmed and Aly A. Farag. Shape from shading under various imaging conditions. In *Computer Vision and Pattern Recognition*, 2007.

[12] Robin Green. Spherical harmonic lighting: The gritty details. In *Archives of the Game Developers Conference*, 2003.

[13] Sudeep Sarkar. *USF HumanID 3D Face Database.* University of South Florida, Tampa, FL, USA.

[14] M. Castelan and J. V. Horebeek. 3d face shape approximation from intensities using partial least squares. *Computer Vision and Pattern Recognition Workshop,* 0:1–8, 2008.

[15] OBJ Files. A 3d object format. In *http://people.scs.fsu.edu/ burkardt/data/obj/obj.html.*

[16] E. Bruce Goldstein. *Sensation and Perception.* Wadsworth Publishing, 8th edition, 2009.

[17] F. Allard. Information processing in human perceptual motor performance. In *Working Paper,* Department of Kinesiology, University of Waterloo, Ontario, 2001.

[18] W. Epstein. Nonrelational judgments of size and distance. *American Journal of Psychology,* 78:120123, 1965.

[19] Jeremy M. Wolfe, Keith R. Kluender, Dennis M. Levi, Linda M. Bartoshuk, Rachel S. Herz, Roberta L. Klatzky, Susan J. Lederman, and Daniel M. Merfeld. *Sensation and Perception.* Sinauer Associates, 2nd edition, 2008.

[20] Harvey Richard Schiffman. *Sensation and Perception: An Integrated Approach.* Wiley, 5th edition, 2001.

[21] George Mather. *Foundations of Perception.* Psychology Press, 1st edition, 2006.

[22] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 21:690–706, 1999.

[23] George B. Arfken and Hans J. Weber. *Mathematical Methods for Physicists: A Comprehensive Guide.* Academic Press, 6th edition, 2005.

[24] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[25] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision,* 47:7–42, April 2002.

[26] Berthold K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View.* PhD thesis, Massachusetts Inst. of Technology, Cambridge, Massachusetts, USA, 1970.

[27] Abdelrehim H. Ahmed. *Shape From Shading under Various Imaging Conditions.* PhD in Electrical Engineering, CVIP Laboratory, University of Louisville, Louisville, KY, USA, 2008.

[28] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 680–702, 1991.

[29] Emmanuel Prados and Olivier Faugeras. Unifying approaches and removing unrealistic assumptions in shape from shading: Mathematics can help. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume 3024 of *Lecture Notes in Computer Science*, pages 141–154. Springer, 2004.

[30] Wen Yi Zhao and Rama Chellappa. Symmetric shape-from-shading using self-ratio image. *Int. J. Comput. Vision*, 45(1):55–75, 2001.

[31] David C. Knill. Mixture models and the probabilistic structure of depth cues. *Vision Research*, 43:831–854, 2003.

[32] Joseph J. Atick, Paul A. Griffin, and A. Norman Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Comput.*, 8(6):1321–1340, 1996.

[33] R. Dovgard and R. Basri. Statistical Symmetric Shape from Shading for 3D Structure Recovery of Faces. In *Proceedings of the 8^{th} European Conference on Computer Vision (volume II)*, volume 3022 of *Lecture Notes in Computer Science*, pages 99–113, Prague, Czech Republic, May 2004.

[34] William A. P. Smith and Edwin R. Hancock. Recovering facial shape using a statistical model of surface normal direction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1914–1930, 2006.

[35] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, 2003.

[36] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685, 2001.

[37] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[38] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:218–233, 2003.

[39] Ravi Ramamoorthi. Analytic pca construction for theoretical analysis of lighting variability in images of a lambertian object. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(10):1322–1333, 2002.

[40] Ham M. Rara, Shireen Y. Elhabian, Thomas Starr, and Aly A. Farag. Model-based shape recovery from single images of general and unknown lighting. In *IEEE International Conference on Image Processing (ICIP) '09*, pages 517–520, 2009.

[41] Ham M. Rara, Shireen Y. Elhabian, Thomas Starr, and Aly A. Farag. 3d face recovery from intensities of general and unknown lighting using partial least squares (pls). In *IEEE International Conference on Image Processing (ICIP) '10*, 2010.

[42] Shireen Elhabian, Ham Rara, Asem Ali, and Aly Farag. Illumination-invariant statistical shape recovery with contiguous occlusion. In *Eighth Canadian Conference on Computer and Robot Vision (CRV 2011)*, 2011.

[43] Michael J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.*, 63(1):75–104, 1996.

[44] John Bustard and Mark Nixon. 3d morphable model construction for robust ear and face recognition. *Computer*, pages 2582–2589, 2010.

[45] Volker Blanz, Albert Mehl, Thomas Vetter, and Hans-Peter Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, 3DPVT '04, pages 293–300, Washington, DC, USA, 2004. IEEE Computer Society.

[46] Ham M. Rara, Asem A. Ali, Shireen Y. Elhabian, Thomas L. Starr, and Aly A. Farag. Face recognition at-a-distance using texture, dense- and sparse-stereo reconstruction. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pages 1221–1224, Washington, DC, USA, 2010. IEEE Computer Society.

[47] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH '01*, pages 117–128, 2001.

[48] Frank Hampel, Elvezio Ronchetti, Peter Rousseeuw, and Werner Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley-Interscience, New York, April 2005.

[49] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: a review. *Int. J. Comput. Vision*, 6(1):59–70, 1991.

[50] Gerard Medioni and Sing Bing Kang. *Emerging Topics in Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[51] Peter Huber. *Robust Statistics*. Wiley, New York, 1981.

[52] Michael J. Black. *Robust Incremental Optical Flow*. PhD thesis, Yale University, New Haven, CT, 1992.

[53] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.

[54] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole, February 1988.

[55] Andrew Blake and Andrew. Zisserman. *Visual reconstruction / Andrew Blake and Andrew Zisserman.* MIT Press, Cambridge, Mass. :, 1987.

[56] A. Patel and W.A.P. Smith. 3d morphable face models revisited. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on,* 0:1327–1334, 2009.

[57] Edward Angel. *Interactive Computer Graphics: A Top-Down Approach with OpenGL.* Addison Wesley Longman, Inc., 2000.

[58] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1999.

[59] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer graphics: principles and practice (2nd ed.).* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[60] Iain Matthews and Simon Baker. Active appearance models revisited. *Int. J. Comput. Vision,* 60(2):135–164, 2004.

[61] M. B. Stegmann. Active appearance models: Theory, extensions and cases, August 2000.

[62] T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, U.K., March 2004.

[63] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 11:567–585, 1989.

[64] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* Wiley, New York, 2. edition, 2001.

[65] I. T. Jolliffe. *Principal Component Analysis.* Springer, second edition, October 2002.

[66] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing.* Cambridge University Press, New York, NY, USA, 1992.

[67] *http://www.cse.usf.edu/ sarkar/, USF DARPA Human-ID 3D FaceDatabase, Courtesy of Prof. Sudeep Sarkar, University of South Florida, Tampa, FL.*

[68] B. Jrgensen and Y. Goegebeur. Multivariate Data Analysis and Chemometrics. Available from: <http://statmaster.sdu.dk/courses/ST02/index.html>.

[69] Zihan Zhou, Andrew Wagner, Hossein Mobahi, John Wright, and Yi Ma. Face recognition with contiguous occlusion using markov random fields. *Computer Vision, IEEE Computer Society International Conference on,* pages 1050–1057, 2009.

[70] Sami Romdhani, Volker Blanz, Curzio Basso, and Thomas Vetter. Morphable models of faces. In Stan Z. Li and Anil K. Jain, editors, *Handbook of Face Recognition*. Springer, 2004.

[71] Sami Romdhani, Volker Blanz, and Thomas Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, ECCV '02, pages 3–19, London, UK, UK, 2002. Springer-Verlag.

[72] E. Kreyszig. *Advanced Engineering Mathematics*. Wiley, 9th edition, November 2005.

[73] Gilbert Strang. *Introduction to Linear Algebra, Third Edition*. Wellesley Cambridge Pr, March 2003.

[74] Ham Rara and Aly Farag. Model-based 3d shape recovery from single images of unknown pose and illumination using a small number of feature points. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–6, Washington, DC, USA, October 2011.

[75] Ham Rara, Aly Farag, Shireen Elhabian, Asem Ali, William Miller, Thomas Starr, and Todd Davis. Face recognition at-a-distance using texture and sparse-stereo reconstruction. In *2010 Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*, pages 1–7, September 2010. 978-1-4244-7580-3.

[76] Y. Wang, S. Lucey, and J. Cohn. Non-rigid object alignment with a mismatch template based on exhaustive local search. In *IEEE Workshop on Non-rigid Registration and Tracking through Learning (NRTL)*, 2007.

[77] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99*, pages 187–194, 1999.

APPENDIX I

2D/3D Deformable Models

Deformable face models are integral to computer vision applications such as face recognition, expression recognition, lip-reading, head-pose estimation, and gaze estimation. The face model is usually constructed from a training data of either 2D images or 3D range scans. The model is fitted to an input image and the fitting process results to model parameters. These parameters can be used with a variety of applications including face recognition. Deformable face models can be classified into two classes: (a) 2D models such as Active Appearance Models (AAM) and (b) 3D models like Morphable Models (MM). AAMs and MMs are similar such that both consist of a linear shape and texture model. The primary difference comes in the nature of the shape component of the model, with AAMs being 2D while MMs are 3D. The next sections will discuss AAMs and MMs in greater detail.

A. Statistical Shape/Appearance Models: AAM

An Active Appearance Model is an integrated statistical model composed of a shape variation model and an appearance model. The model is said to generalize to almost any valid example image. Fitting the model to an input image involves finding the model parameters that minimize the difference between the input and the synthesized model example.

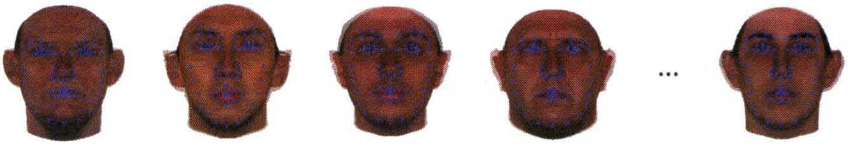


FIGURE 96– Sample annotations of the USF database. There is a total of 76 manually annotated landmark points, including both anatomical and pseudo-landmarks.

1. Model Construction

The AAM is constructed from a training set of labeled images, where key landmark points are marked on each example object. For a face, the landmark points outline the main features as shown in Figure 96.

The shape of the object is described by a vector with the 2D position of the landmarks as elements, e.g. $\mathbf{s}_{2D} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)^T$, where (x_i, y_i) refers to the image coordinates of the *i*th landmark point. For consistency, all shape vectors are normalized to a common coordinate system using Generalized Procrustes Analysis [62]. Principal component analysis (PCA) is applied to the shape data and the resulting shape model is of the following form: $\mathbf{s}_{2D} = \bar{\mathbf{s}}_{2D} + \mathbf{P}_s\mathbf{b}_s$, where $\mathbf{s}_{2D}$ is a shape vector, $\bar{\mathbf{s}}_{2D}$ is the mean vector, $\mathbf{P}_s$ is the set of orthogonal modes of shape variation and $\mathbf{b}_s$ is the set of shape parameters.

The construction of the appearance model requires that each example image such that its control points match that of the mean shape. The appearance instance for each training sample is the warped image region inside the mean shape. Warping can be performed using the piecewise-affine warp (PAW) or thin-plate splines (TPS) [62]. Similar to the construction of the shape model, the appearance vector is represented as $\mathbf{g} = (I_1, I_2, \ldots, I_n)^T$, where I_i denotes the intensity of the sampled pixel in the warped image. PCA is applied to construct the linear appearance model $\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{b}_g$, where $\mathbf{g}$ is a shape vector, $\bar{\mathbf{g}}$ is the mean vector, $\mathbf{P}_g$ is the set of orthogonal modes of appearance variation and $\mathbf{b}_g$ is the set of appearance parameters.

2. Combined Shape/Appearance Models (Combined AAM)

One of the assumptions with the AAM proposed by Cootes et al. [36] is that there may be correlations between the shape and appearance variations. Therefore, a combined model is constructed by concatenating the $\mathbf{b}_s$ and $\mathbf{b}_g$ vectors for each training sample

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s\mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s\mathbf{P}_s^T(\mathbf{x} - \bar{\mathbf{x}}) \\ \mathbf{P}_g^T(\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix} \tag{86}$$

where $\mathbf{W}_s$ is a diagonal matrix of weights for each shape parameter that takes into account the difference in units between the shape and appearance models. PCA is applied on the concatenated vectors, resulting to a combined model

$$\mathbf{b} = \mathbf{Q}\mathbf{c} \tag{87}$$

where $\mathbf{Q}$ are the eigenvectors and $\mathbf{c}$ is a vector of appearance parameters controlling both shape and texture. Because of the linear nature of the combined model, the shape and appearance can be expressed directly as functions of $\mathbf{c}$

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s\mathbf{W}_s\mathbf{Q}_s\mathbf{c} \qquad \mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{Q}_g\mathbf{c} \tag{88}$$

where $\mathbf{Q} = (\mathbf{Q}_s, \mathbf{Q}_g)^T$. Therefore, a new image can be synthesized for any given $\mathbf{c}$ by generating the shape-free image from the vector $\mathbf{g}$ and warping it using the control points describe by $\mathbf{x}$.

3. Independent Shape and Appearance Models (Independent AAM)

The original AAM formulation (combined AAMs) by [36] parameterized shape and appearance with a single set of parameters (88). Matthews and Baker [60] considered the independence of shape and appearance (independent AAMs) in their AAM algorithm. The model construction for both combined and independent AAMs are similar in nature. Independent AAMs do not go the extra step of concatenating the shape ($\mathbf{b}_s$) and appearance ($\mathbf{b}_g$) parameters to get a combined model controlled by the parameter $\mathbf{c}$ (87). Faces in the independent AAM model are instead described by both $\mathbf{b}_g$ and $\mathbf{b}_s$.

To generate a model instance of the independent AAM, the approximated shape described by $\mathbf{b}_s$ is solved by using the linear shape model, $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}_s\mathbf{b}_s$. Similarly, the approximated appearance described by $\mathbf{b}_g$ is determined from the linear appearance model, $\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{b}_g$. Just as in the combined model approach, the approximated appearance is warped according to the 2D points of the approximated shape.

4. AAM Model Search (Fitting)

Given an input 2D image, a fully constructed AAM model, and a sufficient starting condition, the goal of the AAM search (fitting) process is to adjust the model parameters efficiently, such that when a synthetic example is constructed, it matches the input image as closely as possible according to a distance measure.

5. Combined Model

The AAM model fitting process can be considered as an optimization problem, in which the difference the input and synthetic image generated by the AAM model is minimized. The image difference vector δI is defined as

$$\delta I = I_{inp} - I_m \tag{89}$$

where I_{inp} is the vector of grey-level values in the input image and I_m is the vector of grey-level values for the current model parameters.

To locate the best match between the synthetic image from the model and the input image, the magnitude of the image difference vector, $\Delta = |\delta I|^2$, is minimized by varying the model parameters $\mathbf{c}$ in (87).

The simplest model for the relationship between δI and the model parameter error δc is linear

$$\delta c = A\delta I \tag{90}$$

According to [36], this linear relationship is enough to achieve acceptable results. To find A, multivariate linear regression is performed on a sample of known model displacements, δc, and the corresponding difference images, δI. The sets of random displacements can be generated by perturbing the *true* model parameters for images in which they are known. The images can be original or synthetic, with the latter having the advantage of knowing the exact parameters.

The next step is the actual iterative method for the optimization problem. It starts with the initial estimate of the model parameters, $\mathbf{c}_0$, and that of the normalized image sample, $\mathbf{g}_0$. The procedure can be found in [36]. This iterative algorithm is repeated until no improvement is made to the error, $|\delta\mathbf{g}|^2$, and convergence is declared. The authors in [36] used a multiresolution approach to speed up the process and help ensure convergence.

6. Independent AAM Model:

The fitting process for the independent AAM model is similar to that of the combined model. Following the notation in [60], let the image $A(\mathbf{x})$ be the appearance of an AAM defined over the pixels $\mathbf{x}\in\mathbf{s}_0$, where $\mathbf{s}_0$ is the mean shape (base mesh). Let $\mathbf{p} = (p_1, p_2, \ldots, p_n)^T$ be the shape parameters that generate the AAM shape $\mathbf{s}$. Let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^T$ be the appearance parameters that generate the AAM appearance $A(\mathbf{x})$. The AAM model (M) instance with parameters $\mathbf{p}$ and $\boldsymbol{\lambda}$ is created by warping the appearance A from the base mesh $\mathbf{s}_0$ (e.g., the mean shape) to the model shape $\mathbf{s}$. The warping process is denoted by $\mathbf{W}(\mathbf{x}, \mathbf{p})$. Formally, this process can be expressed in equation form as

$$M(\mathbf{W}(\mathbf{x}, \mathbf{p})) = A(\mathbf{x}) \tag{91}$$

The goal of the fitting process is to minimize the error between input $I(\mathbf{x})$ and $M(\mathbf{W}(\mathbf{x}, \mathbf{p})) = A(\mathbf{x})$. The error can be computed from two choices of coordinate frames, the image I coordinate frame and the AAM coordinate frame. The algorithm in [60] used the AAM coordinate frame, which is the base mesh $\mathbf{s}_0$.

Suppose $\mathbf{x}$ is a pixel in $\mathbf{s}_0$ and the corresponding pixel in the input image I is $\mathbf{W}(\mathbf{x}, \mathbf{p})$. At the pixel $\mathbf{x}$, the AAM has the appearance $A(\mathbf{x}) = A_0(\mathbf{x}) + \Sigma_{i=1}^{m} \lambda_i A_i(\mathbf{x})$, where $A_i(\mathbf{x})$ is similar to $\mathbf{P}_g$ in the combined model formulation. At the corresponding pixel $\mathbf{W}(\mathbf{x}, \mathbf{p})$ in the I coordinate frame, the input image has the intensity $I(\mathbf{W}(\mathbf{x}, \mathbf{p}))$. Therefore, the error to be minimized is

$$\sum_{\mathbf{x} \in \mathbf{s}_0} [A_0(\mathbf{x}) + \Sigma_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}))]^2 \tag{92}$$

where the sum is performed over all pixels in $\mathbf{s}_0$. The goal of the fitting process is to minimize (92), with respect to the shape parameters p and the appearance parameters $\boldsymbol{\lambda}$. The authors in [60] used the efficient *Inverse Compositional Image Alignment* (ICIA) algorithm, which is fully explained in their paper.

B. 3D Morphable Models (3DMM)

The morphable face model is based on a vector space representation of faces that is constructed such that any convex combination of shape and texture vectors S_i and T_i of a set of examples describe a realistic human face [77] [35]. Formally,

$$S = \sum_{i=1}^{m} a_i S_i \qquad T = \sum_{i=1}^{m} b_i T_i \tag{93}$$

Dense point-to-point correspondence between each face in the database and a reference face is a crucial step in building the morphable face model. The laser scans are parameterized in cylindrical coordinates. The facial surface is expressed in 2D using the h and ϕ variables of the cylindrical coordinate system. Correspondence is achieved when each point in a sample scan, $I_1(h, \phi)$, corresponds to the point at the same location $I_2(h, \phi)$ in another scan. Figure 97 illustrates the dense correspon-

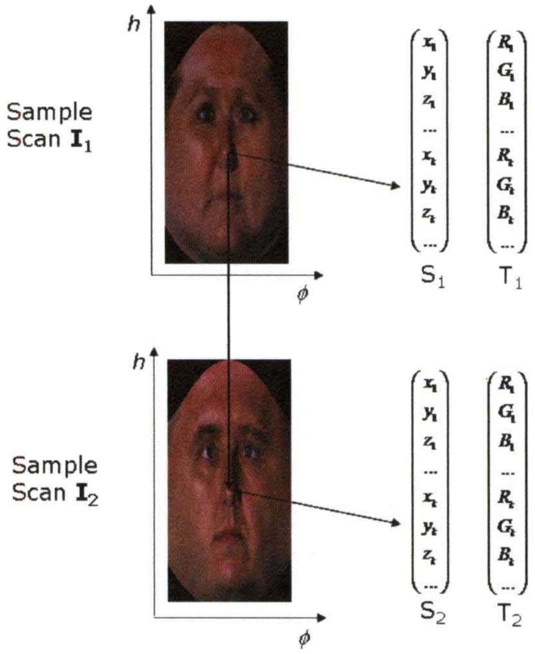
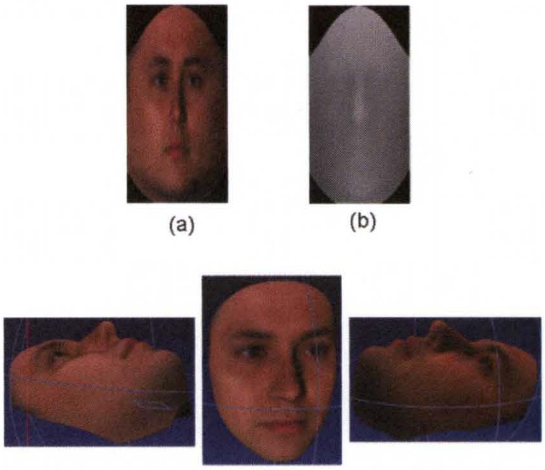

Sample Scan $\mathbf{I}_1$
h
φ
x_1 y_1 z_1 ... x_k y_k z_k ...
R_1 G_1 B_1 ... R_k G_k B_k ...
S_1
T_1
Sample Scan $\mathbf{I}_2$
h
φ
x_1 y_1 z_1 ... x_k y_k z_k ...
R_1 G_1 B_1 ... R_k G_k B_k ...
S_2
T_2

FIGURE 97 – Dense correspondence between two 3D scans in the cyclograph (parameterized by h and ϕ) domain.

dence between two sample scans. [77] uses a modified optical flow to determine this correspondence. Moreover, since scans of different individuals may differ in overall brightness and size, Laplacian pyramids are utilized, instead of the raw data. This work uses the USF HumanID 3D database [67] to build the morphable face model, as well as similar models for other face processing tasks. The database samples are already in dense correspondence with each other, using the algorithm in [77]. Figure 98 shows the mean face of the USF database, with shape and texture information parameterized by h and ϕ variables, in 2D.

1. Face Vectors

The shape and texture vectors are based on a certain reference face, which is commonly the average (mean) face. The reference face has m faces composed of n vertices. The vertices $k \in 1, \ldots, n$ are located at $(h_k, \phi_k, r(h_k, \phi_k))$ and (x_k, y_k, z_k) in cylindrical and Cartesian coordinate systems, respectively. Each vertex has a



(a) (b)

(c)

FIGURE 98 – The mean face of the USF database. (a) mean face in RGB format parameterized by h and ϕ, (b) radius data of the mean face parameterized by h and ϕ, (c) mean face expressed in Cartesian coordinates, visualized as an OBJ file [15].

corresponding color of (R_k, G_k, B_k). The *reference* shape and texture vectors, concatenated versions of all shape and texture vertices, are defined as

$$S_0 = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)^T \qquad T_0 = (R_1, G_1, B_1, \ldots, R_n, G_n, B_n)^T \tag{94}$$

The database samples, represented by S_i and T_i, take the same form as the reference face with each vertex k corresponding to that of the same vertex k in the reference face.

Principal Component Analysis (PCA) is applied to both shape and texture values. The analysis is done separately since it is assumed that there is no correlation between shape and texture. The preliminary step of PCA is to compute the mean of the sample vectors and subtract it to each sample, resulting to a mean-centered data. The next step is to compute the eigenvectors of the covariance matrix, e.g. $C = AA^T$, where $A = (a_1, a_2, \ldots, a_n)$ and a_i is a mean-centered column vector. After PCA, (94) becomes

$$S = \bar{S} + \sum_{i=1}^{m-1} \alpha_i s_i \qquad T = \bar{T} + \sum_{i=1}^{m-1} \beta_i t_i \tag{95}$$

where s_i and t_i are shape and texture eigenvectors, respectively. α_i and β_i are the shape and texture coefficients, respectively.

2. Image Synthesis

The three-dimensional positions and color values of the vertices are controlled by the coefficients α_i and β_i in (95). The synthesis of 2D images from the model consists of two steps: (a) solving for the image positions of vertices and (b) determining the color and illumination of each 3D face in the mesh.

$$(w_{x,k}, w_{y,k}, w_{z,k})^T = R_\gamma R_\theta R_\phi x_k + t_w \tag{96}$$

The angles ϕ, θ, and γ refer to the rotations around the vertical axis, horizontal axis, and the camera axis, respectively. t_w is a spatial shift. The next step is to perform a perspective projection that maps the vertices to image coordinates (p_x, p_y)

$$p_{x,k} = P_x + f\frac{w_{x,k}}{w_{z,k}} \qquad p_{y,k} = P_y - f\frac{y_{x,k}}{w_{z,k}} \tag{97}$$

The variable f refers to the focal length of the camera and (P_x, P_y) defines the image-plane position of the optical axis. A similar formulation is used in [5], but with orthographic projection.

$$s_{2D} = fPR(\bar{s} + S\alpha + t_{3D}) + t_{2D} \tag{98}$$

where f is a scale parameter (focus), P is the orthographic projection matrix, R is the rotation matrix, t_{3D} and t_{2D} are translation vectors in 3D and 2D, respectively. It has the advantage of being expressed in matrix formulation, in its current form as well as its derivative with respect to the shape coefficients.

The illumination of surfaces depends on the direction of the surface normal n. Since the triangles in a 3D face mesh are minute in size (e.g., 0.2 mm^2 in area), it is satisfactory to consider only the triangle centers. The 3D coordinate and color of the center is determined by solving for the mean of the corner's values. The face can be illuminated using a simple Lambertian model or by more elaborate lighting models (e.g., the Phong model [57]).

3. Fitting the 3D Morphable Model to an Image

The fitting method optimizes the shape parameters α_i and texture parameters β_i, together with the rendering parameters (e.g. parameter f, angles ϕ, θ, and γ, etc.), such that when the 3D model is rendered, it is close to the input image using a certain cost function. Given an input image

$$I_{inp}(x,y) = (I_r(x,y), I_g(x,y), I_b(x,y))^T \tag{99}$$

the goal is to minimize the sum of square differences over all pixels and color channels between the input and synthetic image, with the following equation for the error

$$E_I = \sum_{(x,y)} \|I_{inp}(x,y) - I_{mod}(x,y)\|^2 \tag{100}$$

The first iterations need to roughly align the synthetic reconstruction to the input image, using manually defined feature points $(q_{x,j}, q_{y,j})$ and the positions $(p_{x,j}, p_{y,j})$ of the corresponding vertices, in an additional cost function

$$E_F = \sum_j \| \begin{pmatrix} q_{x,j} \\ q_{y,j} \end{pmatrix} - \begin{pmatrix} p_{x,j} \\ p_{y,j} \end{pmatrix} \|^2 \tag{101}$$

The overall cost function is a weighted sum of E_F and E_I, together with additional terms related to the PCA coefficients [35]. At the start of the optimization, E_F is weighted high. The term E_I dominates the final iterations and the optimization process no longer depends on E_F. The original morphable model implementation uses a stochastic version of Newton's method to minimize the overall cost function [35]. The authors in [5] developed a semi-automatic method for fitting the input 2D image to the morphable model, where the shape and texture coefficients are optimized separately.

CURRICULUM VITAE

A. CONTACT INFORMATION

Ham M. Rara
July 1, 1981.

4748 Creek Tree Ct
Louisville, Kentucky, 40219 USA.
(502) 415-3784
hmrara01@{louisville.edu,gmail.com}

B. RESEARCH INTERESTS

Computer vision, image processing, pattern recognition, machine learning, remote biometrics, stereo reconstruction, shape-from-shading (3D-from-2D), illumination modeling

C. EDUCATION

University of Louisville, Louisville, Kentucky USA

Ph.D., Electrical Engineering, August, 2008,

- Dissertation Topic: "3D Facial Shape Estimation from a Single Image under Arbitrary Pose and Illumination"
- Advisor: Aly A. Farag

University of Louisville, Louisville, Kentucky USA

M.S., Electrical Engineering, May, 2006

- Thesis Topic: "Dimensionality reduction techniques in face recognition"
- Advisor: Aly A. Farag

University of the Philippines Diliman, Quezon City, Philippines

B.S., Electrical Engineering, April, 2003

D. HONORS AND AWARDS

- Grosscurth University Fellowship (2006-2008)
- University of Louisville Dean's Citation Award (2006)
- Romago Award for Excellence in Electrical Engineering (2003)

E. PUBLICATIONS

Model-based 3D Shape Recovery

Ham Rara and Aly Farag, "Model-based 3D Shape Recovery from Single Images of Unknown Pose and Illumination using a Small Number of Feature Points," Proc. of the International Joint Conference on Biometrics (IJCB 2011), Washington D.C., USA, pp. 1221-1224, 11-13 October 2011

Shireen Elhabian, **Ham Rara**, Asem Ali, and Aly Farag. Illumination-invariant statistical shape recovery with contiguous occlusion. In Eighth Canadian Conference on Computer and Robot Vision (CRV 2011), St. John's, Newfoundland, 25-27 May 2011

Ham Rara, Shireen Elhabian, Thomas Starr, and Aly Farag, "3D Face Recovery from Intensities of General and Unknown Lightning Using Partial Least Squares," Proc. of 2010 IEEE International Conference on Image Processing (ICIP), pp. 4041-4044, 2010.

Ham Rara, Shireen Elhabian, Thomas Starr, and Aly A. Farag, "Model-Based Shape Recovery From Single Images Of General And Unknown Lighting," 2009 IEEE International Conference on Image Processing (ICIP), Nov. 7 - Nov. 10, 2009, Cairo, Egypt.

Biometrics

Jon Parris, Michael Wilber, Brian Heflin, **Ham Rara**, Ahmed El-barkouky, Aly Farag, et al., "Face Detection on Hard Datasets," Proc. of the International Joint Conference on Biometrics (IJCB 2011), Washington D.C., USA, pp. 1221-1224, 11-13 October 2011

Ham Rara, Aly Farag, Shireen Elhabian, Asem Ali, Mike Miller, Thomas Starr and Todd Davis, "Face Recognition at-a-Distance using Texture and Sparse-Stereo Reconstruction," Proc. of IEEE Fourth International Conference on Biometrics: Theory, Applications and Systems (BTAS), pp. 1221-1224, 2010

Ham Rara, Asem Ali, Shireen Elhabian, Thomas Starr, and Aly A. Farag, "Face Recognition at-a-Distance using Texture, Dense- and Sparse-Stereo Reconstruction," Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 1221-1224, 2010.

Ham Rara, Shireen Elhabian, Asem Ali, William Miller, Thomas Starr, and A. A. Farag, "Face recognition at-a-distance based on sparse-stereo reconstruction," IEEE CVPR Biometrics Workshop, 2009.

Illumination Modeling

Shireen Elhabian, **Ham Rara**, and Aly Farag, "Towards Accurate and Efficient Representation of Image Irradiance of Convex-Lambertian Objects Under Unknown Near Lighting," Proc. of IEEE International Conference on Computer Vision (ICCV 2011), Barcelona, Spain, Nov. 2011

Shireen Elhabian, **Ham Rara** and Aly Farag, "Towards Efficient and Compact Phenomenological Representation of Arbitrary BRDF Reflectance," Proc. of the British Machine Vision Conference (BMVC), University of Dundee, 2011

Shireen Elhabian, **Ham Rara**, and Aly Farag, "On The Use of Hemispherical Harmonics For Modeling Images of Objects Under Unknown Distant Illumination,"Proc. of 2011 IEEE International Conference on Image Processing (ICIP), Brussels, Belguim, Sept. 2011

Medical Imaging

Melih S. Aslan, Asem Ali, **Ham Rara**, and Aly A. Farag, "An Automated Vertebra Identification and Segmentation in CT Images," Proc. of 2010 IEEE International Conference on Image Processing (ICIP), pp. 233-236, 2010.

Melih S. Aslan, Asem Ali, Aly A. Farag, **Ham Rara**, Ben Arnold, and Ping Xiang, "3D Vertebral Body Segmentation Using Shape Based Graph Cuts," Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 3951-3954, 2010.

F. PROFESSIONAL ACTIVITIES

- IEEE SoutheastCon 2010, 2011 (Technical Program Committee)
- IET Computer Vision (Reviewer)
- Computerized Medical Imaging and Graphics (Reviewer)

G. SOFTWARE PROGRAMMING

C/C++, Matlab, Python, Java, CUDA C (GPU), OpenCV, OpenMP

H. LANGUAGES

- **Filipino** (Native)
- **English** Fluent (Reading/Writing/Speaking)

I. MEMBERSHIP

- Eta Kappa Nu
- Golden Key International Honour Society